**T.C.**
**YILDIZ TEKNİK ÜNİVERSİTESİ**
**FEN BİLİMLERİ ENSTİTÜSÜ**

# HETEROJEN ROBOT TAKIMI İÇİN 3B YOL PLANLAMA VE YOL TAKİBİ

**Ömer Faruk KAYA**

YÜKSEK LİSANS TEZİ
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Programı

Danışman
Dr. Öğr. Üyesi Erkan USLU

Ağustos, 2023

Danışmanım Dr. Öğr. Üyesi Erkan USLU sorumluluğunda tarafımca hazırlanan Heterojen Robot Takımı İçin 3B Yol Planlama ve Yol Takibi başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Ömer Faruk KAYA

İmza

*Bana güç veren aileme*
*ve arkadaşlarıma ithaf edilmiştir.*

# TEŞEKKÜR

Heterojen Robot Takımı İçin 3B Yol Planlama ve Yol Takibi başlıklı çalışmamın geliştirilmesinde anlayışı ve yaptığım görüşmeler sonucu verdiği bilgilerle ve gösterdiği yol haritası için danışmanım Dr. Öğr. Üyesi Erkan USLU'ya, sağladığı donanım ve bilgi birikimi açısından Polonom Teknoloji'ye ve Olasılıksal Robotik Grubu'na teşekkürü borç bilirim. Ayrıca bütün dertlerimde, sıkıntılarımda, mutluluklarımda maddi ve manevi yanımda olan aileme ve tüm dostlarıma teşekkür ederim.

Ömer Faruk KAYA

# İÇİNDEKİLER

**KISALTMA LİSTESİ** vii

**ŞEKİL LİSTESİ** viii

**TABLO LİSTESİ** x

**ÖZET** xi

**ABSTRACT** xiii

**1 GİRİŞ** 1

1.1 Literatür Özeti . . . . . 1
1.2 Tezin Amacı . . . . . 5
1.3 Hipotez . . . . . 6

**2 ROS** 7

2.1 Roscore . . . . . 8
2.2 ROS Düğümleri (Nodes) . . . . . 8
2.3 ROS Başlıkları (Topics) . . . . . 10
2.4 ROS Servisleri (Services) . . . . . 12
2.5 ROS Paketleri (Packages) . . . . . 12
2.6 ROS Araçları (Tools) . . . . . 13

**3 SİMÜLASYON** 17

3.1 Gazebo . . . . . 17
3.2 Yükseklik Haritasından Dünya Oluşturma . . . . . 18

**4 HARİTALAMA ve ODOMETRİ** 19

4.1 Eş Zamanlı Konum Belirleme ve Haritalama (EKBH) . . . . . 19
4.2 Yükselti Tabanlı Haritalama . . . . . 21
4.3 Gezilebilirlik Haritası . . . . . 23

**5 NAVİGASYON** 25

5.1 Costmap Yapılandırmaları . . . . . 27

5.2 Planlayıcılar . . . . . 29

**6 DENEYSEL SONUÇLAR** **35**

6.1 Simülasyon Ortamında Kullanılan Robotlar . . . . . 35

6.2 Gerçek Dünya Testlerinde Kullanılan Robotlar . . . . . 39

6.3 Simülasyon Test Ortamı . . . . . 51

6.4 Gerçek Dünya Test Ortamı . . . . . 56

**7 SONUÇ** **61**

**KAYNAKÇA** **65**

**TEZDEN ÜRETİLMİŞ YAYINLAR** **69**

# KISALTMA LİSTESİ

| | |
|---|---|
| API | Application Programming Interface |
| EKBH | Eş Zamanlı Konum Belirleme ve Haritalama |
| GPS | Global Positioning System |
| IMU | Inertial Measurement Unit |
| MPC | Model Predictive Control |
| OSRF | Open Source Robotics Foundation |
| PID | Proportional Integral Derivative |
| ROS | The Robot Operating System |
| RRT | Rapidly-exploring Random Trees |
| SMC | Sliding Mode Control |

# ŞEKİL LİSTESİ


**Şekil 2.1** Örnek bir rviz görseli . . . . . 14
**Şekil 2.2** Örnek bir rqt_tf_tree görseli . . . . . 15
**Şekil 2.3** Örnek bir rqt_plot görseli . . . . . 16
**Şekil 2.4** Örnek bir rqt_graph görseli . . . . . 16
**Şekil 4.1** LIO-SAM sistem yapısı[32] . . . . . 20
**Şekil 4.2** Gerçek ortam testlerinin yapıldığı araziden bir LIO-SAM görseli . 21
**Şekil 4.3** Örnek bir grid_map görseli [35] . . . . . 22
**Şekil 4.4** a) Haritanın çıkarıldığı dünya, b) Çıkarılan yükseklik haritası . . . 23
**Şekil 4.5** Filtre zinciri görseli . . . . . 24
**Şekil 4.6** Gezilebilirlik haritası görseli . . . . . 24
**Şekil 5.1** "Navigation stack" yapısının görseli [36] . . . . . 25
**Şekil 5.2** Önerilen yöntemin "Navigation stack" yapısının görseli . . . . . 26
**Şekil 5.3** Örnek bir drone_costmap görseli . . . . . 28
**Şekil 5.4** Örnek bir global_costmap görseli . . . . . 29
**Şekil 5.5** Örnek bir local_costmap görseli . . . . . 29
**Şekil 5.6** GPS ile ara nokta bazlı takip için oluşturulan potansiyel alan . . . 34
**Şekil 6.1** Clearpath Husky simülasyon görseli . . . . . 35
**Şekil 6.2** Lazer mesafe ölçüm duyargası görseli . . . . . 36
**Şekil 6.3** Simülasyon lazer mesafe ölçüm duyargası örnek verisi . . . . . 37
**Şekil 6.4** Hector Quadrotor simülasyon görseli . . . . . 38
**Şekil 6.5** Clearpath Jackal görseli . . . . . 40
**Şekil 6.6** Clearpath Jackal üzerine duyargalar eklenmiş görseli . . . . . 40
**Şekil 6.7** Cincoze DX1100 endüstriyel bilgisayar . . . . . 41
**Şekil 6.8** Velodyne VLP-16 lazer ölçüm duyargası . . . . . 42
**Şekil 6.9** Velodyne VLP-16 lazer ölçüm duyargası örnek verisi . . . . . 42
**Şekil 6.10** Microstrain 3DM-RQ1-45 iç işleyiş görseli [41] . . . . . 43
**Şekil 6.11** Microstrain 3DM-RQ1-45 görseli . . . . . 44
**Şekil 6.12** Clearpath Jackal kontrol kartı ve motor sürücüsü . . . . . 45
**Şekil 6.13** 26V 11.6Ah Lityum-Ion batarya görseli . . . . . 45
**Şekil 6.14** DJI Matrice 600 Pro görseli . . . . . 46
**Şekil 6.15** DJI Matrice 600 Pro üzerine duyargalar eklenmiş görseli . . . . . 46
**Şekil 6.16** Nvidia Xavier görseli . . . . . 48

**Şekil 6.17** Velodyne VLP-32 lazer ölçüm duyargası . . . . . 49
**Şekil 6.18** DJI A3 Pro görseli . . . . . 49
**Şekil 6.19** DJI A3 Pro TTL bağlantısı detay görseli . . . . . 50
**Şekil 6.20** DJI A3 Pro TTL bağlantısı üstten görseli . . . . . 50
**Şekil 6.21** 22.8V 5700mAh Lityum-Polimer batarya görseli . . . . . 51
**Şekil 6.22** Dünyanın oluşturulduğu yükseklik haritası . . . . . 52
**Şekil 6.23** Oluşturulan Gazebo dünyası . . . . . 52
**Şekil 6.24** Oluşturulan Gazebo dünyası yandan görünüm . . . . . 52
**Şekil 6.25** Simulasyon test senaryo noktaları . . . . . 54
**Şekil 6.26** Planlanan yollar ile gerçek mesafeler(Mavi: Gerçek Mesafe, Yeşil: Hesaplanan Yol) a) Kısa ve düz zemin, b) Uzun ve engebeli zemin, c) Kısa ve engebeli zemin . . . . . 55
**Şekil 6.27** Önerilen yöntem ile simülasyonda hedeflere çizilen yollar a) A* ile Yeşil, b) A* ile Sarı, c) A* ile Mor, d) RRT* ile Sarı . . . . . 57
**Şekil 6.28** Gerçek robot testlerinin yapıldığı arazi . . . . . 58
**Şekil 6.29** Önerilen yöntem ile A* algoritması kullanılarak gerçek dünyada hedeflere çizilen yollar a) Kırmızı, b) Mor, c) Turkuaz, d) Turuncu 59

# TABLO LİSTESİ

**Tablo 6.1** Lazer mesafe ölçüm duyargasının özellikleri . . . . . 36
**Tablo 6.2** Atalet duyargasının özellikleri . . . . . 37
**Tablo 6.3** Atalet duyargasının özellikleri . . . . . 38
**Tablo 6.4** Lazer mesafe ölçüm duyargasının özellikleri . . . . . 39
**Tablo 6.5** Atalet duyargasının özellikleri . . . . . 39
**Tablo 6.6** Yer aracının özellikleri . . . . . 41
**Tablo 6.7** Ana işlem biriminin özellikleri . . . . . 42
**Tablo 6.8** Lazer mesafe ölçüm duyargasının özellikleri . . . . . 43
**Tablo 6.9** Atalet duyargası ve GPS özellikleri . . . . . 44
**Tablo 6.10** Hava aracının özellikleri . . . . . 47
**Tablo 6.11** Hava aracı ana işlem birimi özellikleri . . . . . 47
**Tablo 6.12** Lazer mesafe ölçüm duyargasının özellikleri . . . . . 48
**Tablo 6.13** Uçuş kontrolcüsü özellikleri . . . . . 49
**Tablo 6.14** Yol planlama algoritmaları test sonuçları . . . . . 53
**Tablo 6.15** Yol planlama ve gidilen yol karşılaştırması . . . . . 54
**Tablo 6.16** Simülasyon Test Sonuçları . . . . . 56
**Tablo 6.17** Gerçek Dünya Test Sonuçları . . . . . 60

# ÖZET

## Heterojen Robot Takımı İçin 3B Yol Planlama ve Yol Takibi

Ömer Faruk KAYA

Bilgisayar Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Dr. Öğr. Üyesi Erkan USLU

Günümüzde, askeri operasyonlarda, çevrenin izlenmesinde ve doğal afetlerde arama kurtarma görevlerinde heterojen robot ekiplerinin kullanımı artmaktadır. Heterojen robot ekipleri, madencilik, tarım, çiftçilik, dış mekan gözetimi ve dış mekan haritalama gibi alanlarda da kullanılabilmektedir. Ancak, 2 boyutlu otonom navigasyon sistemleri, engebeli arazilerde başarısız olmaktadır. Bu çalışmada, yükseklik haritasındaki eğimleri dikkate alarak 2.5B navigasyon kullanılarak bu soruna bir çözüm önerilmektedir. Çalışmada önerilen yöntem simülasyon ve gerçek ortam dikkate alınarak geliştirilmiştir. Simülasyon ortamındaki geliştirmede ROS altyapısı ve Gazebo simülasyon ortamı kullanılmıştır. Hava ve kara araçları koordineli bir şekilde çalıştırılmakta ve çevredeki yükseltiler çıkarılan haritalara işlenmektedir. Çalışmada gerçek dünyadaki engebeli bir araziyi yansıtan bir simülasyon dünyası oluşturulmuş ve içerisine bazı kentsel öğeler, binalar ve ağaçlar simülasyon dünyasına eklenmiştir. Gerçek ortamda kullanılan yer aracı ile uyumlu olacak şekilde kara aracı olarak üzerinde 3 boyutlu lazer ölçüm duyargası, GPS ve 9 DOF atalet duyargalarına sahip bir robot kullanılmıştır. Bu duyargalar kullanılarak bir Eş zamanlı Konum Belirleme ve Haritalama (EKBH) algoritması çalıştırılmıştır. EHBH algoritmasının çıktısı olan konum verisi kullanılarak kara aracında bir 2.5B harita oluşturulmuştur. GPS tabanlı nokta bazlı navigasyon ile kara aracını takip edebilen, üzerinde 3 boyutlu lazer mesafe ölçüm, 9 DOF atalet ve GPS duyargalarını bulunduran bir hava aracı kullanılmış ve duyargaları kullanılarak kara aracında 2.5B bir harita katmanı daha oluşturulmuştur. Her iki aracın koordine çalışması ile kara aracı için global bir plan oluşturulmaktadır, bu plan kara aracının

bilgisinin yeterli olduğu durumda sadece kara aracının haritası kullanılarak, yetersiz olması durumunda hava aracının çıkardığı harita bilgileri kullanılarak yapılmıştır. Yapılan bu plan kara aracının yerel planlayıcısı kullanılarak takip edilmiştir. Önerilen yöntem sayesinde, daha kısa yollar ile hedeflere ulaşılmanın yanı sıra yerel minimuma sıkışma sorunları da azaltılmıştır. Çalışmada önerilen yöntem ile literatürdeki yöntemlerin karşılaştırması hem simülasyon hem gerçek ortamda gerçekleştirilmiştir.

**Anahtar Kelimeler:** Yükseklik tabanlı haritalama, konumlandırma, 3B haritalama, 2.5B dış ortam navigasyon, heterojen robot takımı

# ABSTRACT

## 3D Path Planning and Tracking for Heterogenous Robot Team

Ömer Faruk KAYA

Department of Computer Engineering
Master of Science Thesis

Supervisor: Assist. Prof. Dr. Erkan USLU

Today, the use of heterogeneous robot teams is increasing in military operations, monitoring the environment, and search and rescue missions in natural disasters. Heterogeneous robot teams can also be used in fields such as mining, agriculture, farming, outdoor surveillance, and outdoor mapping. However, single 2D autonomous navigation systems fail in rough terrains. In this study, a solution to this problem is proposed using 2.5D navigation that takes into account the slopes on the elevation map. The proposed method in the study was developed by considering simulation and real environment. ROS infrastructure and Gazebo simulation environment were used in the development in the simulation environment. Drone and ground vehicles are operated in a coordinated manner and the elevations in the environment are processed on the map. The simulation world used in the study reflects a real-world rough terrain and also some urban artifacts such as buildings and trees are added to the simulation world. A robot, which compatible with ground vehicle used in the real environment, equipped with a 3D laser measurement sensor, GPS and 9 DOF inertial sensors is used as a ground vehicle. The ground vehicle can output 3D map and 3D localization using 3D SLAM. Using the SLAM localization, a 2.5D map is created on the ground vehicle. Drone, which utilizes 3D LIDAR, 9 DOF IMU, and GPS sensors, is capable of following the ground vehicle with a GPS-based waypoint navigation and can create a 2.5D map using its sensors. A global plan is created for the ground vehicle by the cooperative effort of both robots, using the map information from the ground vehicle where available and using the map information from the drone where the ground vehicle's map is insufficient.

2.5D navigation of the ground vehicle is carried out by the local planner taking into account the calculated cooperative global path. Thanks to the proposed method, it is possible to reach the given target points through shorter routes and reduce the number of situations where path planning is stuck at local minimums. The proposed method in the study was compared with the methods in the literature both in simulation and real environment.

**Keywords:** Elevation mapping, localization, 3D mapping, 2.5D outdoor navigation, heterogeneous robot team

# 1
# GİRİŞ

## 1.1 Literatür Özeti

İnsansız araçlar askeri, ticari ve akademik ortamlarda her geçen gün daha fazla kullanılmaktadır. Bu insansız araçlar tekil veya birtakım olarak insanların girmek istemediği veya giremediği alanlarda çalışmalar yaparak insanların hayatını kolaylaştırmaktadır. İnsansız araçlardan birtakım oluşturulmak istenildiğinde genellikle hava ve kara araçları birlikte kullanılmaktadır. Hava aracı olarak ise sabit kanatlı araçlar daha uzun görev süresine sahip olsa da genellikle kalkış için geniş alanlara ihtiyaç duymakta ve manevra kabiliyetleri az olmaktadır, bu sebeple çok rotorlu araçlar tercih edilmektedir. Yüksek irtifalara çıkarak alanın daha geniş görünümünden duyargaları aracılığıyla aktardığı bilgi sayesinde kara araçları görevlerini daha hızlı ve güvenli tamamlayabilmektedir.

Robotların verilen görevleri gerçekleştirebilmeleri için cevabını bilmek zorunda oldukları sorulardan ikisi "Ben neredeyim?" ve "Çevremdeki ortam neye benziyor?" sorularıdır. Bu soruların bir arada çözüldüğü yöntemlere ise Eş Zamanlı Konum Belirleme ve Haritalama (EKBH) yöntemleri denilmektedir. EKBH yöntemlerinin konum belirleme ve haritalama kısımları birbirini besleyen veya ayrık iki yapı olarak bulunabilmektedir. Genel EKBH yapısında konum belirleme modülünün ürettiği odometri verisi haritalama modülü tarafından duyarga verilerinin harita üzerine işlenmesi sırasında kullanılmaktadır. Literatürde ise birbirinden farklı birçok duyargayı veya duyarga takımlarını kullanarak 2 boyutlu ve 3 boyutlu olarak çözüm üreten birçok yöntem bulunmaktadır. 2 boyutlu düzlemde yapılan çalışmalarda daha çok 2 boyutlu lazer ölçüm duyargaları ile çözümler sunan yöntemler bulunmaktadır. 2 boyutlu düzlemde menzil ve yön duyargası kullanarak ortamdaki referans noktaları algılayan ve genişletilmiş kalman filtresi kullanarak EKFSLAM[1] yöntemi sunulmuştur. 2 boyutlu düzlemde menzil ve yön duyargası kullanan diğer bir yöntem olan FastSLAM[2], genişletilmiş kalman filtresine dayanan yöntemlerindeki sınırlılığına karşın Bayesci bir bakış açısını sunmuştur. Gmapping[3] yönteminde 2 boyutlu konumlandırma sırasında

Rao-Blackwellized parçacık filtrelerinin kullanımı önerilmiş ve doğru bir harita oluşturmak için gereken parçacık sayıları tartışılmıştır. 2 boyutlu lazer ölçüm duyargası ile ataletsel algılama temelli 3B tutum tahmini siteminin birleştirilmesinin önerildiği HectorMapping[4] yöntemi, düşük hesaplama gereksinimleri ile esnek ve ölçeklenebilir bir sistem sunmaktadır. Literatürde geliştirilmiş 2 boyutlu yöntemlerin bir kısmının büyük alanlarda yaşadığı kaybolma ve harita bozulma problemlerini graf yapısı kullanarak ve bu graf düğümleri arasında döngü kapatma ile çözen SlamToolbox[5] yöntemi, gerçek zamanlı ve çevrimdışı çalıştırma seçenekleri de sunmaktadır. İki boyutlu fabrika, ofis, ev gibi ortamlarda 2 boyutlu algoritmalar uygun bir şekilde konum kestirimi yapıp harita çıkarabilse de, yükseltilerin ve çukurların olduğu arazi şartlarında konum kestirimi ve haritalama yapamamaktadır. Arazi şartlarında konum verisi çıkarabilmek veya iki boyutlu düzlem olsa da 2 boyutlu lazer ölçüm duyargaları ile kesit alındığında konum kestiriminin zor olduğu ortamlarda 3 boyutlu lazer ölçüm duyargası, tekil ve stereo kameralar, derinlik kameraları kullanarak konum kestirim yapıp harita çıkaran EKBH yöntemleri geliştirilmiştir. Bu yöntemlerden bir tekil kamera ve bir düşük maliyetli atalet duyargası kullanarak çözüm üreten VINS-Mono[6] 6 serbestlik derecesinde konum üretmek için doğrusal olmayan optimizasyon temelli bir yöntem kullanmaktadır. ORB-SLAM3 [7] kamera duyargasından gelen verilerden "ORB" algoritması ile özellik çıkarmakta ve bu özellikleri eşlemektedir. Bu eşleştirmelerden oluşan çıktılar ile atalet duyargasının verilerini de birleştirerek harita ve odometri verisi üretmektedir. 3 boyutlu lazer ölçüm duyargasını kullanan LIO-SAM (Lidar Inertial Odometry via Smoothing and Mapping) [8] yöntemi ise LeGO-LOAM yönteminin de temel aldığı LOAM (Lidar Odometry And Mapping) [9] paketini temel almaktadır. LOAM paketi aracılığıyla üretilen "LiDAR-Inertial Odometry" verisini alıp üzerine atalet duyargasından gelen verileri kullanarak bir faktör graf üzerinde optimizasyonlar yaparak harita ve konum verisi üretmektedir. Sadece lazer ölçüm duyargası veya kamera verileri ile EKBH yapan yöntemlerin yanı sıra iki farklı duyargayı bir arada kullanarak konum iyileştirmesi yapan yöntemler de bulunmaktadır. LeGO-LOAM (LightwEight and Ground Optimized Lidar Odometry And Mapping) [10] yöntemi 3 boyutlu lazer ölçüm ve stereo kamera duyarga ikilisini kullanarak yüksek doğruluğa sahip gerçek zamanlı harita ve odometri üreten bir algoritmadır. Bir görsel-ataletsel sistem (VIS) ve bir lidar-ataletsel sistem (LIS) olmak üzere iki alt sistemden oluşan LVI-SAM[11], konum doğruluğunu arttırmak amacıyla bu iki alt sistemi birbirine sıkı olarak bağlamaktadır. Literatürde geleneksel yöntemlerin yanı sıra problemi çözmek amacıyla derin öğrenme yöntemleri de kullanılmaktadır. 3DRegNet [12] yöntemi nokta bulutlarının kaydı amacıyla regresyon yerine derin öğrenme yaklaşımını önermektedir. Nokta bulutları yerine kamera verilerini kullanan LIFT-SLAM [13]

yönteminde ise kamera verilerini eşlemek amacıyla derin öğrenme temelli özellik tanımlayıcılar önerilmektedir.

Robotların görevi gerçekleştirebilmesi için konum ve harita bilgisini elde ettikten sonra "Hedefe nasıl gideceğim?" sorusunun yanıtını bulması gerekmektedir. Global planlama, robotun bulunduğu konumdan hedef konuma harita bilgisini de kullanıp engellere çarpmadan gidilebileceği en optimal yolları bulmayı sağlayarak bu sorunun cevabını vermektedir. Hedef noktaya giden yolun bulunması için çalıştıracak algoritmalar çeşitli ortam temsil şekillerine ihtiyaç duyabilmektedir. Örneğin, algoritmaların bazıları ortamın graf ile oluşturulmuş bir temsiline ihtiyaç duymaktadır. Hücre Ayrışımı (Cell Decomposition) ve Yol Haritaları (Roadmaps) graf yapısıyla oluşturulan, düğümler arası gezinmenin maliyetinin az olduğu ve bellekte depolamanın kolay olduğu ortam temsil yöntemleridir [14]. Hücre ayrışımı yöntemlerinde ortam eşit ve karesel alanlara bölünebileceği gibi, farklı poligonlar ile ve dinamik boyutlarla da alt parçalara bölünebilmektedir. Yol haritaları yöntemlerinde ise ortam Görünürlük (Visibility) [15], Voronoi [16] ve State-Lattice yöntemleri ile temsil edilebilmektedir. Ortam temsili sürecinde ortamdaki noktalar arasında bulunan mesafelerin temsil edilmesinin yanı sıra ortamdaki noktaların ne kadar yüksekliğe sahip olduğunun belirtilebildiği Dijital Yükseklik Haritası (Digital Elevation Map) [17] yöntemi de bulunmaktadır. Bu ortam temsil yöntemleri kullanılarak yol planlaması yapan algoritmalar ise C-Space arama, reaktif hesaplama ve optimal kontrol olmak üzere üç ana başlık altında bulunmaktadır. Reaktif yol planlama ortam temsilinin sadece mevcut engellerin yer ve şekillerini gösterdiği durumlarda kullanılmaktadır. Reaktif yol planlama altında ise iki ana başlık bulunmaktadır. Bu ana başlıklardan ilki ortamdaki engellerin itme ve gidilmek istenilen hedefin çekme kuvveti olarak tanımlandığı yapay potansiyel alan (Artificial Potential Fields) yöntemleridir [18]. Diğer başlıkta ise uygulanacak kuvvetin sanal kuvvet toplamları yerine elektrostatik potansiyel alan olarak tanımlandığı vektör alan histogramı (Vector Field Histogram) yöntemleridir [19]. Literatürdeki en popüler C-Space arama ile yol bulma algoritmalarından biri A* [20] algoritmasıdır. A* algoritması haritayı bir graf olarak ele almakta ve sezgisel aramalar yaparak en kısa yolu bulmaktadır. Bir diğer yöntem olan Dijkstra algoritması [21] da A* yöntemi gibi haritayı bir graf olarak ele almakta ve en kısa yolu hesaplamaktadır, ancak yayılımlı olarak ilerleyen bu yöntem arazi şartlarına bağlı olarak daha uzun hesaplama süreleri ile sonuç verebilmektedir. D* Lite [22] yöntemi ise ortamların sıklıkla değişebildiği veya düzenli olarak yol planlaması yapılması gereken ortamlar için üretilmiştir. Hesaplanan yolu sadece değişen alanlarda güncelleyen bu yöntem D* yönteminin artımlı bir sürümüdür. Klasik yöntemler hızlı ve deterministik olarak yol planlaması yapabilse de, arama uzayının

arttığı durumlarda çok uzun hesaplama ihtiyaçları olmaktadır. Bu arama süresini kısaltmak amacıyla optimal kontrol başlığı altında örnekleme tabanlı teknikler önerilmiştir. Örnekleme tabanlı yöntemlerden olan RRT* [23] yöntemi, RRT [24] yönteminin en kısa yolu bulma konusunda optimize edilmiş versiyonudur.

Robotların görevlerini gerçekleştirebilmeleri amacıyla son bir soruya cevap vermeleri gerekmektedir, bu soru "Nasıl yolu takip edeceğim?" sorusudur. Bu sorunun cevabı için de literatürde bir yöntem bulunmaktadır. Bu yöntemlerden biri olan Model Predictive Control (MPC) [25], doğrusal olmayan sistemleri, robot kısıtlarını ve kinematiklerini işleyebilmesi sayesinde aracın yola uygun bir biçimde gidebilmesi için gereken tüm manevraları hesaplayabilmektedir. Bu kullanıcıya tüm tanımları yapma özgürlüğü sunması sebebiyle otonom otomobiller, insansız hava araçları ve mobil robotlar gibi geniş bir kullanım alanı bulmaktadır. Proportional Integral Derivative (PID) [26] yöntemi ise, robotların kinematiklerinin çok karmaşık olmadığı sistemlerde ise bilinirliği, uygulama kolaylığı ve basitliği sebebiyle literatürde sıklıkla kullanılmaktadır. Sliding Mode Cotnrol (SMC) [27] yöntemi ise robot modeli bağımsız bir şekilde geri bildirim eylemlerini dikkate alarak kontrol komutu üretebilmektedir.

Hava ve kara araçlarının işbirlikçi şekilde çalışması ve bu sayede görevlerin daha hızlı ve doğru bir şekilde tamamlanabilmesi için literatürde birçok yöntem önerilmiştir. Dragonfly [28] yönteminde nöron topolojik olarak görevlerin gerçekleştirildiği arazinin topolojik bir modeli çıkarılmış ve bu model kara aracının haraketlerini optimize etmek için kullanılmıştır. Önerilen diğer bir yöntemde [29] ise, her biri bireysel olarak kamera duyargaları ile bireysel haritalar oluşturan robot sisteminin haritasını birleştirerek ortamın daha doğru bir haritasının çıkarılması hedeflenmiştir. Simülasyon ortamında gerçeklenen diğer bir yöntemde [30] ise, bir sonlu durum makinası kullanılarak koordineli bir robot ekibinin engellerden kaçınarak bir formasyonu koruması sağlanmıştır.

Gerçek robot sistemleri ile çalışmalar ve testler yapmak zor ve maliyetli olmaktadır. Gerçek robotları birebir simüle eden sistemlerin kullanılması hem algoritma geliştirme sürecinde yaşanabilecek kırımları engellemekte, şarj kısıtından kurtarmakta ve uç durum testlerinin gerçekleştirilmesine olanak sağlamaktadır. Ayrıca bu robot sistemleri arasında haberleşme için bir standart arayüz olması, simülasyonda geliştirilen algoritmaların minimum maliyetle gerçeğe aktarılması ihtiyacı doğmaktadır. Bu problemlerin hepsini çözmeyi sağlayan The Robot Operating System (ROS), her ne kadar isminde işletim sistemi geçse de gerçek manada bir çerçevedir (framework) ve içerisinde birçok hazır metod ve yöntemler barındırmaktadır. [31] Üretilen veya kullanılan algoritmaların

arasındaki haberleşme katmanını sağlayarak, kullanıcıları işlemler arası iletişim, soket programlama gibi gereksinimlerden kurtarmaktadır.

## 1.2 Tezin Amacı

Robotlar insanların yapmakta zorlandığı veya giremedikleri alanlarda kullanılmak üzere geliştirilmektedir. Hava, kara ve suda kullanılmak üzere birçok farklı robot türü bulunmaktadır. Bu robotların farklı yetenekleri, amaçları ve sürüş dinamikleri olsa da ortak olarak hepsinde bulunması gereken birtakım özellikler mevcuttur. Bunlar, aracın haraket modeli, bulunduğu konumun ve yönün tespiti, görev yapacağı ortamı haritalama ve otonom yol planlama ve takibidir. Bu problemlerin çözümü için araçlara motor sinyal üretici, iki veya üç boyutlu tarama yapabilen lazer mesafe ölçüm duyargaları, atalet duyargası, Global Positioning System (GPS) ve kamera sistemleri entegre edilmesi gerekmektedir. Bu duyargalar kullanılarak ortama dair bilgiler edinilmekte ve tasarlanan algoritmalarla işlenebilir veriler üretilmektedir.

Robotlara atanan görevler ve ortam karmaşıklaştıkça bireysel olarak problemlerin çözülmesi zorlaşmakta veya çözülememektedir. Bu problemin çözülmesi için homojen veya heterojen robot takımları kullanılabilmektedir. Homojen robot takımları aynı platform ve duyargaları kullanmakta olduğu için genellikle görevin gerçekleştirileceği ortamda bir arama veya keşif olması durumunda görev süresinin kısaltılmasında etkili olmaktadır. Heterojen robot takımlarında ise farklı düzlemlerde hareket edebilen araçlar kullanılarak görev süresi kısaltmanın yanı sıra başarımı da arttırmaktadır.

Yapılan bu çalışmada kara aracının görev ortamının tamamını göremediği koşullarda hava aracının kara aracı ile koordineli çalışmasıyla görev başarımının artması, görev süresinin azaltılması ve ROS içerisinde bulunan 2 boyutlu navigasyon sisteminin engebeli dış ortamlarda da optimal bir şekilde kullanılması için yöntem önerilmiştir. ROS Melodic sürümü üzerinde gerçeklenen yönteme ilişkin deneysel sonuçların elde edilmesi amaçlanmıştır.

Tez çalışmasında Bölüm 2 ile algoritmaların geliştirildiği ve testlerinin yapıldığı aynı zamanda sistem için bir haberleşme katmanı olan ROS ortamının özelliklerine, tarihçesine ve araçlarına yer verilmiştir. Bölüm 3 ile testlerin bir kısmının gerçekleştirildiği Gazebo simülasyon ortamına ve bu ortamda nasıl bir dünya oluşturulduğuna yer verilmiştir. Bölüm 4 ile robotların görevlerini gerçekleştirebilmeleri için kritik öneme sahip EKHB sistemi, yükseklik tabanlı haritalama ve gezilebilirlik analizinden bahsedilmiştir. Bölüm 5 ile robotun

bulunduğu konumdan istenilen konuma yol planlaması ve yol takibi aşamaları, maliyet haritaları ve bunları içerisinde bulunduran "Navigation Stack" yapısından bahsedilmiştir. Bölüm 6 ile testler için kullanılan robotlar, test ortamı ve test ölçüm metriklerinden bahsedilmiş ve test sonuçlarına yer verilmiştir. Bölüm 7 ile deneylerin sonuçları irdelenmiş ve çıktıların sebepleri analiz edilmiştir.

## 1.3 Hipotez

Tez çalışması kapsamında koordineli olarak çalışan bir robot takımı tasarlanmış, kara aracı üzerinde haritalama, navigasyon ve konum kestirim problemleri için algoritmalar kullanılmıştır. Kara aracının navigasyon sürecini iyileştirmek amacıyla kara aracını GPS tabanlı navigasyon ile takip eden hava aracının duyargaları kullanılarak ek bir harita katmanı oluşturulmuştur. Yol planlama ve yol takibi başarımlarının ölçülmesi amacıyla bir model ortaya atılmıştır. Bu model doğrultusunda literatürde bulunan bir takım yöntemler ile karşılaştırmalar yapılmış ve önerilen yöntemin başarımı ölçülmüştür.

"Sisteme eklenen bir hava aracının duyargalarının katkılarıyla, kara aracının yol planlama ve görev tamamlama başarımı artacaktır." hipotezi tez çalışmasını yönlendirmiştir.

# 2
# ROS

ROS, robotik yazılım geliştirme için kapsamlı bir çerçeve sağlayan bir meta işletim sistemi (meta-operating system)'dir. Düşük seviye cihaz kontrolünden, yüksek seviye algoritmaların uygulanmasına kadar geniş yelpazede işlevsellik sağlamasının yanı sıra robot yazılımları geliştirilirken ihtiyaç duyulabilecek bir dizi araç ve kütüphane de sunmaktadır.

Farklı platformlar üzerinde çalışabilecek, farklı diller kullanılarak geliştirme yapılabilecek ve bu geliştirilen algoritmaların birbiriyle sorunsuz bir şekilde haberleşmesini sağlamak zor olmaktadır. İnsanlar için çok basit gibi görünen birçok problem robot perspektifinden bakıldığında birçok zorluk içermektedir. Çözülmesi gereken bu problemlerin hepsinin tek bir birey veya kurum tarafından çözülmesi güç olmaktadır. ROS, robotik yazılımları açık kaynak olarak geliştirilen ve bu sayede insanları iş birliği yapmaya teşvik eden sıfırdan inşa edilmiş bir yapıdır. Gönüllü bireysel ve kurumsal geliştiricilerin iş birliği ile bir çok problem hali hazırda çözülmüş bulunmaktadır. Örneğin, robotların navigasyon alt yapısını sağlayan yapı oluşturulmuş ve paylaşılmıştır. Bu sayede diğer kurumlar veya bireyler aynı eforu sarfetmeden üretilmiş bilgi üzerine yeni bilgiler ekleyerek robotik biliminin ilerlemesine daha kolay katkıda bulunabilmektedirler.

Kısaca ROS'un tarihçesinden bahsetmek gerekirse, Stanford Üniversitesi robotik grubunda çalışan iki doktora öğrencisi olan Eric Berger ve Keenan Wyrobek tarafından temelleri atılmıştır. Açık kaynaklı olarak geliştirilen bu yazılım çerçevesi robotik uygulamalarının geliştirilmesini, test edilmesini ve dağıtılmasını kolaylaştırmak amacıyla tasarlanmıştır. İlk sürümü olan "Box Turtle" 2010 yılında yayımlanmış ve robotik topluluğu tarafından büyük ilgi görmüştür. 2014'te Willow Garage'ın kapatılmasının ardından Open Source Robotics Foundation (OSRF) tarafından devralınmıştır. Gerçek zamanlı işlem, güvenlik ve daha iyi çok platform desteği sunarak ROS'un tasarım sınırlarını aşmayı hedefleyen ROS 2 sürümü 2018 yılında piyasaya sürülmüştür. Daha önce de bahsedildiği üzere bu tez kapsamında ROS framework'ünün "Melodic" sürümü kullanılmıştır.

Bu aşamadan sonra ROS'un genel işleyişini ve işlevselliğini sağlayan çekirdek bileşenlerine detaylı olarak değinilmiştir.

## 2.1 Roscore

ROS önceki kısımlarda da bahsedildiği gibi üretilen algoritmaların ve paketlerin haberleşmesini sağlamak konusunda bir arayüz sunmaktadır. Bu arayüz TCP/IP standardı üzerine kurulmuş ve ileriki başlıklarda bahsedilecek olan başlıklar üzerinden abone ve yayınlayıcı mantığıyla haberleşme sağlamaktadır. Haberleşmenin sağlanabilmesi ve bir düğümün paylaştığı mesajın tüm düğümlere ulaşabilmesi için ana bir yöneticiye ihtiyaç vardır. ROS içerisinde bu yönetim sürecini roscore sağlamaktadır.

Robotlar sürü veya koordine bir şekilde çalıştırılmak istendiği durumda robotların birbirinin duyargalarının veya düğümlerinin ürettiği verilere ulaşabilmeleri için ortak bir roscore kullanmaları gerekmektedir. Bu sürecin işletilebilmesi için araçlardan birinin efendi(master) olarak seçilmesi gerekmektedir. Bu seçimin etkili olabilmesi için köle(slave) olarak seçilen robotlarda *ROS_MASTER_URI* sistem değişkeninin efendi olan robotun IP si olarak ayarlanmalıdır.

Bu tez çalışmaları kapsamında simülasyon ortamında tüm robotlar aynı bilgisayar üzerinde simüle edildikleri için bir efendi-köle ayarlaması yapılması gerekmemiş ancak gerçek robot çalışmalarında kara aracı efendi olarak seçilmiştir.

## 2.2 ROS Düğümleri (Nodes)

ROS yapısı itibariyle dağıtık olarak çalışmayı ve olabildiğince işlem birimlerini birbirinden ayırmayı teşvik etmektedir. Çeşitli mesajların, parametrelerin, servislerin ve bu girdi çıktıyı işleyen algoritmanın bir araya geldiği parça ROS düğümü olarak adlandırılmaktadır. Her biri bir işle ilgilenen bu düğümler bir araya geldiğinde anlamlı bir sistem oluşturmaktadır. Örneğin sistem içerisinde haritalama, navigasyon, duyargaların okunması ve motorların sürülmesi için farklı birer düğüm bulunmakta ve her biri ayrı birer işlem olarak çalışmaktadır.

Dağıtık olarak çalışılmasını sağlayan bu yapı sistemin bütünlüğü, çalışması ve güvenliği için oldukça önemlidir. Her bir düğüm diğer tüm düğümlerden izole olarak çalıştığı için düğüm içerisinde yaşanan bir problem sadece o düğüm içerisinde kalacağı için sistemin geri kalanı işlevlerini devam ettirebilmektedir. Monolotik sistemlerde oluşacak kod karmaşıklığı, sistemin sürdürülebilirliğinin zor olması ve hata tolerans azlığı gibi problemler bu şekilde aşılmaktadır. Ayrıca kara

kutu haline gelen bu düğümlerin parametreleri, girdi ve çıktılarının öğrenilmesi kullanılabilmesi için yeterli olmaktadır. Girdi ve çıktıların roscore üzerinden yönetiliyor olması kod ve algoritmalarını açık olarak paylaşmak istemeyen geliştiricilere derlenmiş düğüm dağıtma ayrıcalığı da tanımaktadır.

ROS içerisinde düğümlerin üç farklı özelliği bulunmaktadır. Bunlar düğümün hangi paket içerisinde bulunduğu, adı ve tipidir. Düğümler başlatılırken düğümün bulunduğu paket ve düğümün tipi kullanılarak başlatılmaktadır. Her düğüm parametreleri ayarlanıp ayağa kaldırıldığında ise sistem tarafından o düğüm için eşsiz bir id tanımlanmaktadır. Sistem çalıştırılırken eğer aynı düğüm iki veya daha çok miktarda çalıştırılmak istenirse her birine farklı düğüm isimleri verilmesi gerekmektedir. Sistem içerisinde koşturulan düğümlerin yayınladıkları ve abone olduğu mesajlar, paket ismi gibi düğüme dair özellikler "rosnode" aracı ile öğrenilebilmektedir.

Sistem içerisindeki çalışan düğümler "rosnode list" komutu çalıştırılarak öğrenilebilmektedir. Aşağıdaki listede komut çıktısında oluşan düğümlerden bir kısmına ve sistemdeki görevlerine değinilmiştir.

- **/drone/elevation_mapping:** Hava aracının duyargalarını kullanarak yükselti tabanlı harita oluşturmaktadır.
- **/drone/traversability_estimation:** Hava aracının duyargaları ile oluşturulan yükselti haritası üzerinde gezilebilirlik analizi yapmaktadır.
- **/drone_pose_publisher:** tf ağacından hava aracının konum verisini alarak yükselti tabanlı haritalama düğümünün istediği mesaj tipinde konum yayınlamaktadır.
- **/gazebo:** Simülasyon aracı düğümüdür.
- **/gazebo_gui:** Simülasyon aracının görsel arayüzü düğümüdür.
- **/global/elevation_mapping:** Kara aracının duyargalarını kullanarak global yükselti tabanlı harita oluşturmaktadır.
- **/global/traversability_estimation:** Kara aracının duyargaları ile oluşturulan global yükselti haritası üzerinde gezilebilirlik analizi yapmaktadır.
- **/local/elevation_mapping:** Kara aracının duyargalarını kullanarak yerel yükselti tabanlı harita oluşturmaktadır.
- **/local/traversability_estimation:** Kara aracının duyargaları ile oluşturulan yerel yükselti haritası üzerinde gezilebilirlik analizi yapmaktadır.

- **/locomotor:** Yol planlama ve yol takibi için gerekli hız komutlarını üretmektedir.
- **/uav_follower:** Hava aracının kara aracını takip etmesi için gerekli hız komutlarını üretmektedir.

## 2.3 ROS Başlıkları (Topics)

ROS içerisinde düğümler arasında ana haberleşme mesajlar aracılığıyla yapılmaktadır. Bir düğüm duyargalar, sistemden edindiği veya algoritmaları aracığıyla ürettiği verileri ROS arayüzü içerisinde bulunan "publish" fonksiyonu aracılığıyla belirlenen başlık üzerinden yayınlar. Bu veri sistem içerisinde "roscore" sayesinde ihtiyaç duyan tüm düğümler tarafından erişebilir hale gelir. Bu veriye erişmek isteyen düğümle yayınlandığı mesajın adını da kullanarak ROS arayüzü tarafından sağlanan "subscribe" metodu ile verileri edinebilmektedir. Yayınlanan başlıkların isimleri ve türleri sistemin doğru çalışabilmesi ve karışıklıkların yaşanmaması amacıyla birbirinden farklı olması gerekmektedir. Sistem içerisinde engellenmeyen bu durumu oluşturacak şekilde iki farklı düğüm aynı isim ve türde veri paylaşması durumunda o veriye ihtiyaç duyan düğümler sistemde o anda en son hangi veri var ise onu kullanmaktadır. Bu durum oluştuğunda veriye ihtiyaç duyan düğümün hangi veriyi edineceğinin ve kullanacağının garantisi olmadığı için algoritmaların çıktılarının beklenenin dışında olması ile sonuçlanabilmektedir. Bazı durumlarda ise aynı isimde veri paylaşılması sistemin sağlığı için avantajlı olabilmektedir. Örneğin navigasyonun düğümünün ürettiği komutları motorlara uygun bir şekilde üretip ileten paketin dinlediği başlığa acil bir durumda durma komutu verilip robotun kaza yapması engellenebilmektedir. Ancak bu durumun en sağlıklı olarak çözümü ara bir düğüm oluşturup bu düğümün gelen mesajların öncelik sıralamasına göre ana başlığa bu veriyi iletmesidir. Sistem içerisinde aynı sürücüyü kullanan birden fazla duyarga olması durumunda başlık isimleri sürücü içerisinde aynı olmaktadır. Bu tez kapsamında da kara ve hava aracında aynı sürücüyü kullanan duyargalar bulunmaktadır ve bunlar aynı başlığa kendi duyarga çıktılarını paylaşmaktadır. Bu durumun sistem içerisinde bir karmaşa yaratmaması için sürücü düğümleri çalıştırılırken ad alanı(namespace) eklenerek çalıştırılmaktadır. Eklenen ad alanları paylaşılan başlık isimlerinin önüne ek olarak gelmekte ve bu sayede verileri kullanan düğümler tarafından ayırt edilebilir olmaktadır.

Sistem içerisindeki yayınlanan başlıklar "rostopic list" komutu çalıştırılarak öğrenilebilmektedir. Aşağıdaki listede komut çıktısında oluşan başlıklardan bir kısmına ve sistemdeki görevlerine değinilmiştir.

- **/clock:** Simülasyon sistem saati olarak zamanı simüle etmektedir.
- **/cmd_vel:** Kara aracının motorlarına uygulanacak hız komutudur.
- **/drone/cmd_vel:** Hava aracının motorlarına uygulanacak hız komutudur.
- **/drone/fix:** Hava aracının GPS duyargası verileridir.
- **/drone/raw_imu:** Hava aracının atalet duyargası verileridir.
- **/drone/velodyne_points:** Hava aracının lazer ölçüm duyargası verileridir.
- **/drone/elevation_mapping/elevation_map:** Hava aracı duyargaları ile oluşturulmuş yükselti haritasıdır.
- **/global/elevation_mapping/elevation_map:** Kara aracı duyargaları ile oluşturulmuş global yükselti haritasıdır.
- **/imu/data:** Kara aracının atalet duyargası verileridir.
- **/local/elevation_mapping/elevation_map:** Kara aracı duyargaları ile oluşturulmuş yerel yükselti haritasıdır.
- **/locomotor/GlobalPlanner/plan:** Navigasyon modülü tarafından hava aracı maliyet haritasından üretilmiş hedefe olan yolun alt noktalarına üretilen yoldur.
- **/locomotor/drone_costmap/costmap:** Hava aracı gezilebilirlik haritasından oluşturulmuş maliyet haritasıdır.
- **/locomotor/drone_path:** Navigasyon modülü tarafından tarafından hava aracı maliyet haritasından üretilmiş yoldur.
- **/locomotor/global_costmap/costmap:** Kara aracı global gezilebilirlik haritasından oluşturulmuş maliyet haritasıdır.
- **/locomotor/local_costmap/costmap:** Kara aracı yerel gezilebilirlik haritasından oluşturulmuş maliyet haritasıdır.
- **/navsat/fix:** Kara aracının GPS duyargası verileridir.
- **/tf:** Sistemdeki dinamik dönüşüm bilgilerini içermektedir.
- **/tf_static:** Sistemdeki statik dönüşüm bilgilerini içermektedir.
- **/velodyne_points:** Kara aracının lazer ölçüm duyargası verileridir.

## 2.4 ROS Servisleri (Services)

ROS içerisinde düğümler arasında başlıklar üzerinden haberleşme sağlandığı durumda bir geri dönüt mekanizması direkt olarak bulunmamaktadır ve bir kere yapılması istenilen emirlerin bir kere gerçekleştirildiğinin veya başarısının durumu edinilememektedir. ROS bu probleme karşın servis yapısına sahiptir. Servisler sunucu ve istemci mantığıyla çalışmaktadır. Bir düğüm eğer diğer düğümler tarafından tetiklenmek istiyorsa sunucu olarak bir servis oluşturmaktadır. Belirlenen isim ile sistemde görünür hale gelen servis istemci düğümler tarafından çağırılabilmekte ve dönütleri alınabilmektedir. Başlıklardan farklı olarak ise sunucu tarafının emir almaya hazır hale geldiği garanti altına alınması gerekmektedir. Bu garanti ise servis çağrısı yapılmadan önce veya düğümün başlangıç aşamasında ROS arayüzü tarafından sunulan "wait_for_service" metodu ile servisin belirlenen süre içerisinde aktif hale gelmesi beklenerek sağlanabilmektedir.

Sistem içerisindeki servisler "rosservice list" komutu çalıştırılarak öğrenilebilmektedir. Aşağıdaki listede komut çıktısında oluşan servislerden bir kısmına ve sistemdeki görevlerine değinilmiştir.

- **/drone/elevation_mapping/clear_map:** Üretilen yükselti haritasının sıfırlanmasını sağlar.
- **/drone/elevation_mapping/get_submap:** Yükselti haritası içerisinden istenilen alanın kesilip döndürülmesini sağlar.
- **/drone/elevation_mapping/save_map:** Yükselti haritasının sisteme kaydedilmesini sağlar.
- **/drone/enable_motors:** Hava aracının motorlarının aktive edilmesini sağlar.
- **/locomotor/GlobalPlanner/make_plan:** Global maliyet haritası üzerinde istenilen noktaya bir plan yapıp planı döndürür.

## 2.5 ROS Paketleri (Packages)

ROS paketleri projelerde geliştirilen ve bir bütün olarak çalışan veya tek bir yerde durması istenilen düğümleri, mesaj tiplerini, servis tiplerini, parametre dosyalarını ve kütüphaneleri içeren temel yapı taşlarından biridir. ROS'un yapısı gereği sadece bir düğüm bile oluşturulacak olsa onun bir paket içerisinde yer alması gerekmektedir. Düğüm eğer derlenen bir dil ile geliştirildiyse nasıl derleneceğini belirleyen adımları içeren dosyaların da paket içerisinde bulunması gerekmektedir.

# 3
# SİMÜLASYON

## 3.1 Gazebo

Gazebo, robotik alanında yazılım geliştirenler tarafından sıklıkla kullanılan bir 3B simülasyon ortamıdır. Açık kaynak olarak geliştirilen bu yazılım, robotların karmaşık dünyalarla etkileşimini simüle etme yeteneğine sahip olmasıyla robotların gerçek dünyada nasıl performans göstereceğini analiz etmemizi sağlamaktadır.

3B dünyaları modellemek, simüle etmek ve görselleştirmek için bir dizi araç ve kütüphane sağlayan yazılım, kullanıcıların bir robotun fiziksel özelliklerini (boyut, ağırlık, hareket yetenekleri vb.) ve duyargalarını (kamera, lazer ölçüm duyargası, GPS vb.) modellemelerine olanak sağlamaktadır.

Kullanıcılarına farklı hava ve aydınlatma koşulları, zemin dokuları gibi çevresel faktörleri ayarlamalarına olanak sağlayarak, oluşturulan farklı koşullar altında robotların nasıl performans göstereceğinin analizinin yapılabilmesini sağlamaktadır.

Ayrıca içerisinde bulunan fizik motoru sayesinde gerçek dünyanın fiziksel yasalarını da simüle etmektedir. Fizik motorları robotların gerçek dünyadaki hareketlerinin ve etkileşimlerinin ne kadar doğru bir biçimde yansıtıldığını büyük ölçüde etkilemektedir. Gazebo sunduğu farklı fizik motoru seçenekleri ile kullanıcılara en uygun olanı seçme fırsatı sunmaktadır.

Gazebo sunduğu Application Programming Interface (API) ile simülasyon ortamında özelleştirilmiş robot modelleri, duyargalar ve eklentiler eklemeye olanak sağlamaktadır.

Tez kapsamında ise hava durumu ve robotlara erişim gibi kısıtlardan soyutlanarak testlerin ve geliştirmelerin yapılabilmesi için kullanılmıştır.

## 3.2 Yükseklik Haritasından Dünya Oluşturma

Kullanılan Gazebo simülasyon dünyası üzerinde gerçek dünyayı yansıtabilmek ve istenilen yer yüzü şekillerini oluşturabilmek için yükseklik haritasından dünya oluşturma yöntemi kullanılmıştır. Bu yöntem ile dünya oluşturabilmek amacıyla gri tonlamalı bir resim dosyası oluşturulması gerekmektedir. Bu oluşturulan resim dosyası üzerinde 0 ile 255 arasında değerler olacak şekilde boyama yapılmaktadır. En yüksek olması istenilen noktalar beyaz en alçak olması istenilen noktalar ise siyah renk ile boyanmıştır. Gri renkli bölgeler ise açıklaştıkça yüksekliği artmakta koyulaştıkça yüksekliği azalmaktadır.

İstenilen araziyi oluşturacak şekilde boyanan resmin Gazebo dünyası olarak açılabilmesi için bir dünya dosyası oluşturulması gerekmektedir. Oluşturulan bu dünya dosyası içerisinde yükseklik haritasının yolu ve boyutları genişlik ve yükseklik olarak metre cinsinden girilmelidir. Dünya içerisinde robotların üzerinde durabilmesi ve duyargaların algılayabilmesi için çarpışma olarak da dünya dosyasına aynı boyut ve yükseklikte eklenmesi gerekmektedir.

Gazebo simülasyon aracı dünya dosyasını açarken bu resim dosyasını okuyarak dosya içerisinde girilen yükseklik değerini 255 ile oranlayarak resimden dünyayı oluşturmaktadır. Oluşan bu dünya içerisine tez kapsamında gerçek dünyayı daha iyi yansıtması amacıyla çeşitli objeler eklenecektir.

Tez kapsamında kullanılması amacıyla bahsedilen yöntem ile Şekil 6.22 ile verilen yükseklik haritasından Şekil 6.23 ile verilen dünya oluşturulmuştur.

süreç hedef noktaya ulaşılana kadar devem etmektedir. Bu heuristik yaklaşım sayesinde algoritma diğer yol planlayıcılara göre geniş alanlarda daha hızlı sonuçlar vermektedir. Ancak heuristik fonksiyonun kalitesi algoritmanın performansını ve doğruluğunu yüksek ölçüde etkilemektedir. Seçilen heuristik fonksiyonun azımsaması (underestimate) durumunda gerçek maliyeti daha düşük olarak tahmin etmesi sebebiyle algoritmanın daha geniş bir arama yapmasına sebep olmaktadır. Bu durum algoritmanın hesaplama süresini uzatmaktadır ancak en optimal yolu üretmeyi sağlamaktadır. Seçilen heuristik fonksiyonun abartması (overestimate) durumunda gerçek maliyeti daha yüksek olarak tahmin etmesi sebebiyle hedefe ulaşmanın gerçekte olduğundan daha maliyetli olduğunu varsaymaktadır. Bu durum algoritmanın hesaplama süresini azaltmaktadır, ancak her durumda en optimal yol üretme ihtimalini düşürmektedir. Üretilen sistemde kaynak kısıtı yoksa ve tam doğruluk önemli ise azımsayacak, kaynak kısıtları varsa abartacak bir fonksiyon seçilebilmektedir.

#### 5.2.1.2 Dijkstra Global Planlayıcı

Dijkstra algoritması A* algoritması gibi robotun bulunduğu konumdan hedef konuma maliyet haritalarını kullanarak en kısa yolu bulan bir algoritmadır. Algoritma başlangıç noktasından başlamakta ve henüz ziyaret edilmemiş ve doğrudan bağlantılı tüm noktaları bulmaktadır. Bulduğu bu noktalar için başlangıç noktasından bulunan noktaya yol maliyetini hesaplamakta ve bu bilgiyi saklamaktadır. Bu hesaplama işleminin ardından en düşük maliyetli olan noktayı seçmekte ve aynı adımları hedef noktaya ulaşılana veya gezilmemiş nokta kalmayana kadar tekrarlamaktadır.

#### 5.2.1.3 RRT* Global Planlayıcı

RRT* algoritması graf tabanlı Dijkstra ve A* algoritmalarından farklı olarak rastgele noktalar üretip bu noktalar arasını bağlayarak yol planlaması yapmaktadır. Dijkstra ve A* 2 boyutlu uzayda makul sürelerde yol planlaması yapabilirken uzayın boyutu arttıkça planlama süresi artmaktadır. Her seferinde en optimal yolu garanti etmeyen bu yöntem, yol bulunduktan sonra bile optimizasyon süreçlerine devam edebilmekte ve en optimale yaklaşabilmektedir.

RRT* algoritması RRT (Rapidly-exploring Random Trees) algoritmasının bir çeşididir. Algoritma, her iterasyonda rastgele bir konum örneklemesi yapmakta ve sonrasında bu konuma en yakın noktayı aramaktadır. Bulunan en yakın noktaya örneklenen konumdan bir yol arar ve bu yolda herhangi bir engel yoksa ağaca bu nokta eklenmektedir. Ardından bu yeni eklenen nokta çevresindeki noktaları

kontrol ederek o noktaya daha optimal bir yol olması durumunda ağacı tekrardan yapılandırmaktadır. Sonrasında bu adımları belirlenen iterasyon sayısı, süre veya hedefe ulaşıncaya kadar tekrar etmektedir.

### 5.2.2 Lokal Planlayıcılar

Lokal planlayıcılar global planlayıcıların ürettiği hedef noktasına nasıl gidileceğini gösteren yol bilgisini alıp aracın bu yol üzerinde gitmesi için gereken hız komutlarını üreten algoritmalardır. Kendi içerisinde engel kaçınımı özellikleri barındırabilen lokal planlayıcılar olduğu gibi sadece yolu takip edip engel olması durumunda global planlayıcıyı tekrardan bir yol hesaplaması için tetikleyen planlayıcılar da bulunmaktadır. Bu tez çalışması kapsamında kendi içerisinde bir yol planlaması bulunmayan ve sadece global planlayıcının ürettiği yolu takip eden ve engel çıkması durumunda global planlayıcıya tekrar yol hesaplattıran Proportional Integral Derivative (PID) tabanlı bir lokal planlayıcı kullanılmıştır. PID tabanlı lokal planlayıcı, mevcut hatayı, hatanın zaman integralini ve hatanın türevini kullanarak geri beslemeli hız verisi üreten bir algoritma kullanmaktadır.

$$\upsilon(t) = K_p \cdot e(t) + K_i \cdot \int (e(t)dt + K_d \cdot \frac{(de(t))}{dt}) \tag{5.1}$$

$$K_p = K_u/1.7 \tag{5.2}$$

$$K_i = P_u/2 \tag{5.3}$$

$$K_d = P_u/8 \tag{5.4}$$

Denklem 5.1 ile kullanılan lokal planlayıcının temel aldığı PID formülü verilmiştir. Bu denklemde verilen $u(t)$ çıkış sinyalini, $e(t)$ mevcut hatayı temsil etmektedir ve $K_p$, $K_i$ ve $K_d$ sırasıyla orantısal, integral ve türevsel kat sayıları temsil etmektedir. Kullanılan yöntemde çıkış sinyali aracın uygulaması gereken hız, hata miktarı ise yoldan sapma olarak tanımlanmıştır. Bu yoldan sapma hatasının azaltılması ile global planlayıcının ürettiği yoldan minimum sapmalar gösterecek hız komutları üretilmektedir.

Sistemin kat sayıları belirlenirken Ziegler-Nichols yöntemin önerdiği üzere sistemin integral ve türev kontrolleri kapatılmış, sadece oransal kontrol ile sistem osilasyona girene kadar $K_p$ değeri arttırılmıştır. Sistemi periyodik osilasyonda kontrol eden

# 7
# SONUÇ

Çalışmada, hava aracı ve kara aracının koordine birlikte çalışması ve bu sayede daha kısa sürede daha yüksek başarıma sahip önerilmiştir. Önerilen yöntem ROS Melodic sürümü üzerinde gerçeklenmiştir. Yöntemle ilgili testler Gazebo simülasyon ortamı ve tanıtılan arazide gerçeklemiştir.

Tez kapsamında yapılacak olan navigasyon testlerinde kullanılacak yol planlaması algoritması seçimi sırasında biri deterministik biri olasılıksal iki farklı algoritma eksiksiz bir maliyet haritasında test edilmiştir. Bu testler yapılırken deterministik algoritma olarak graf tabanlı A*, olasılıksal algoritma olarak ise örnekleme temelli RRT* algoritması seçilmiştir. Seçilen bu algoritmaların yol planlama süresi, yol uzunluğu, planlama başarımı ve yolun pürüzsüzlüğü ölçülerek kıyaslama yapılmıştır. Yapılan testlerin sonucunda rastgele olarak atamalar yapıp bunların arasında engel olup olmadığını kontrol eden RRT* algoritmasının probleme yaklaşımı sebebiyle deterministik olan A* algoritmasına göre daha pürüzlü yollar çizdiği görülmüştür. Bu pürüzlü yollar aynı zamanda 10 kere tekrarlı test sonucunda alınan ortalamalara göre de daha uzun olmuştur. Yapılan testlerde A* her denemede yol çizebilse de, daha fazla boyutlu uzaylarda başarılı sonuçlar üretmek için kullanılan RRT* algoritması 2 boyutlu uzayda 0.5 saniye optimizasyon süresinde hedefe ulaşan yollar çizmede başarısız olduğu görülmüştür. Yapılan testlerin sonucunda tez kapsamında deterministik yol planlama algoritmaları ile devam edilmesine karar verilmiştir.

Test ortamlarında öncelikle sadece lokal duyargalar kullanılarak testler gerçekleştirilmiştir. Sadece lokal duyargaların kullanıldığı ve geçmişe dair verilerin tutulmadığı bu yöntemde hedefe gidilen yol üzerinde etrafından dolaşılması gereken engeller olduğu durumlarda osilasyon problemleriyle karşılaşılmış ve aracın istenilen hedefe ulaşamadığı görülmüştür. Ancak ortamda etrafından dolaşılması gereken engeller olmadığı durumlarda aracın önerilen yönteme yakın başarımlar verdiği görülmüştür. Yerel sensör verilerinin basit senaryolarda işe yaradığı ancak geçmişe dair bilgilere ihtiyaç duyulduğu veya karmaşık ortamlarda

çözüm üretmek için yeterli olmadığı görülmüştür.

Yalnızca lokal duyargaların verilerinin kullanıldığı durumda yaşanılan osilasyon problemlerinin ana sebebinin geçmişe dair verilerin tutulmaması ve bu sebeple kaybedilen verilerin yol planlama esnasında problemlere yol açması olduğu görülmüştür. Bu sebeple literatürde uzun yıllardır kullanılan 2 boyutlu haritalama ve 2 boyutlu navigasyon yöntemlerinin test edilmesine karar verilmiştir.

Test edilen 2 boyutlu haritalama ve 2 boyutlu navigasyon testlerinde ise süreç kara aracı üzerinde bulunan 3 boyutlu lazer ölçüm duyargasının 2 boyuta indirgenmesi ve bu 2 boyutlu lazer bilgisiyle 2 boyutlu bir harita üzerinde navigasyon yapması olarak tanımlanmıştır. Ürettiği harita ile geçmişe dair veriler tutabilen bu yöntem yol üzerinde etrafından dolaşılması gereken engelleri aşabilecek kadar veri tutuyor olsa da, arazinin eğimli yüzeyine denk gelen lazer ölçümlerinin engel olarak algılanması ile harita üzerinde eğimli yerler engel olarak işaretlenmektedir. Bunun sonucu olarak aracın aslında gidebileceği bir eğime sahip bir hedef noktası olsa da araç istenilen hedefe yol çizememektedir. Ancak düz zeminlerde yapılan testlerde bu yöntemin de önerilen yönteme yakın sonuçlar verdiği görülmüştür.

Düz ve engebe içermeyen alanlarda optimal sürelerde ve yol uzunluklarında sonuçlar üretebilen ve lokal minimumlara takılıp osilasyonda kalma problemini çözen 2 boyutlu haritalama ve 2 boyutlu navigasyon yöntemi tümsek ve çukurlar olduğu durumlarda başarısız olmuştur. Başarısız olmasının temel sebebi lazer ölçüm duyargasındaki verilerin sadece bir katmanını kullanılması ve bu veri ile sadece dolu, boş ve bilinmez alan olarak haritaya işleme yapılması olduğu görüşmüştür. Lazer mesafe ölçüm duyargasından bir veri geldiğinde yöntem gereği robotun bulunduğu alandan o noktaya kadar alanlar boş, noktanın düştüğü alan ise dolu olarak işaretlenmektedir. Ancak hafif eğimli bir yükseltinin duyarganın bulundu seviyeden yüksek olması durumunda o noktaya çarpacak ve veri döndürecektir, ve o noktaya kadar olan yerler ızgara yapısında boş olarak o nokta ise dolu olarak işlenecektir. Aslında robotun bulunan az eğim sebebiyle gidebileceği bir yükselti gidilemez bir alan olarak işlenmiş olmaktadır. Ofis, fabrika, depo gibi düz zemine sahip alanlarda başarılı olarak çalışabilecek olsa da tez kapsamında tanımlanan arazi şartlarını başarılı olarak tamamlayamamıştır. Arazinin eğiminin tanımlanabileceği ve işlenebileceği yükselti tabanlı bir haritalama yönteminin kullanılmasına karar verilmiştir.

Ortamlardaki eğimleri ve araziyi daha iyi tanımlayabilen 2.5 boyutlu haritaların üretildiği ve bunlar üzerinden navigasyon yapılan yöntem, önerilen yöntemle karşılaştırılacak son yöntem olmuştur. Bu yöntem arazi şartlarını çok iyi

tanımlayabilse de daha büyük bir alan ya da daha yüksek çözünürlük kısıtlarına takılmaktadır. Her ikisini birden daha iyi hale getirdiğimizde hafıza, frekans düşüklüğü, işleme maliyetinin artması gibi problemlerle yüz yüze gelinmiştir. İki parametrenin de optimal olarak ayarlandığı durumlarda ise global haritanın sınırları içerisinde kalan bölgelerde başarılı sonuçlar verdiği gözlemlenmiştir. Ancak arazinin tamamını görememesi sebebiyle hedef noktaya daha uzun yoldan gidebildiği veya yerel minimuma takılarak osilasyonda kalabildiği gözlemlenmiştir.

Kullanılan 2.5 boyutlu haritalama sonuçlarının navigasyon tarafından kullanılabilmesi için 2 boyutlu maliyet haritalarına indirgenip 2 boyutlu yol planlamanın optimal yoldan uzaklaştırıp uzaklaştırmadığına dair testler gerçekleştirilmiştir. Bu testlerde içerisinde hiçbir engebe bulunmayan kısa bir yol planlaması yaptırılmış ve bu yol planlaması sonucunda planlanan yol uzunluğu ve gidilen yol uzunluklarının birbirine çok yakın oldukları görülmüştür. İçerisinde engebe bulunan bir kısa bir alanda yol planlaması yaptırılıp yol takibi yaptırıldığında ise planlanan yol ve gidilen yol arasında fark oluştuğu görülmüştür. Daha uzun ve içerisinde hem düzlük hem de engebe bulunan bir alanda test gerçekleştirildiğinde ise planlanan yol ile gidilen yol arasındaki farkın oransal olarak azaldığı görülmüştür. Yapılan testlerin sonucunda ise 3 boyutlu dünyanın 2.5 boyuta ve sonrasında 2 boyuta indirgenmesinin optimal yoldan uzaklaşma gibi bir problem yaratmadığı görülmüştür.

Önerilen yöntemde ise 2.5 boyutlu global ve lokal haritalama ile navigasyon yöntemi üzerine hava aracının duyargasıyla oluşturulan bir yeni katman eklenmiştir. Bu eklenen yeni katman sayesinde araziyi havadan görme avantajı ve global haritanın başarılı yollar çizebilmesi için kısıt problemi aşılmıştır. Öncelikle hava aracının katkısıyla oluşturulan maliyet haritası üzerinden yol çizmek ortamda istenilen hedefe farklı uzunluklarda iki yol olduğu durumlarda kısa olanın seçilmesini sağladığı gözlemlenmiştir. Hava aracının maliyet haritasının olmadığı durumlarda global haritanın boyut kısıtları sebebiyle yaşanan yerel minimumda takılma problemlerinin çözüldüğü görülmüştür.

Test edilen yöntemlerden harita tutan tüm yöntemlerde haritalar ataletsel merkezli olarak tutulmuştur. Haritaların ataletsel merkezli tutulmayıp dünya merkezli tutulduğu durumlarda araçlar arazi üzerinde ilerledikçe haritanın büyüme ihtiyacı doğmuştur. Bu durumda özellikle 2.5 boyutlu haritalar ile yapılan çalışmalarda haritanın büyümesi işleme, oluşturma ve ağ içerisinde yayma zorluğu oluşturduğu görülmüştür. Daha önceki tüm verilerin tutulması global bir çerçevede daha detaylı planlama yapmayı mümkün kılsa da oluşturduğu zorluklar sebebiyle tercih edilmemiştir. Kullanılan ataletsel merkezli yaklaşımda haritanın merkezi hep

robot olacak şekilde ayarlanmış, yeni gelen veriler eklenmiş, harita dışarısında kalan veriler ise silinmiştir. Bu yöntem hedefin global maliyet haritası dışarısında olduğu durumlarda bilinmeyen alanlardan yol çizilmesi ve yerel çerçevede asıl optimizasyonun yapılmasının zorunlu kılınmasına sebep olmaktadır. Bu da hedefe gitmeye çalışılırken lokal minimumda osilasyon problemleri ile karşılaşılmasına sebep olmaktadır. Önerilen yöntem ile eklenen yeni daha geniş alanları kaplayan yine kara aracına göre ataletsel merkezli haritanın düşük çözünürlük ve geniş alanlı olması sayesinde osilasyon problemleri ile karşılaşılmamıştır.

Gelecek çalışmalarda ise hava aracında da bir EKBH çalıştırması ve GPS bağımsız harita çıkarması ve bu haritayı kullanarak GPS verisinin olmadığı iç ortam, askeri arazilerde navigasyon yapması planlanmaktadır. Ayrıca çıkarılan 3 boyutlu hava ve kara aracı haritalarının çeşitli algoritmalar ile eşleştirilmesi ve bulunan bağıl konum verisi kullanılarak kara aracının takip sürecinin de GPS bağımsız hale getirilmesi planlanmaktadır.

# KAYNAKÇA


[1] T. Bailey, J. Nieto, J. Guivant, M. Stevens, E. Nebot, "Consistency of the ekf-slam algorithm," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 3562–3568. doı: 10.1109/IROS.2006.281644.

[2] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, "Fastslam: A factored solution to the simultaneous localization and mapping problem," Nov. 2002.

[3] G. Grisetti, C. Stachniss, W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007. doı: 10.1109/TRO.2006.889486.

[4] S. Kohlbrecher, O. von Stryk, J. Meyer, U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," in *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, 2011, pp. 155–160. doı: 10.1109/SSRR.2011.6106777.

[5] S. Macenski, I. Jambrecic, "Slam toolbox: Slam for the dynamic world," *Journal of Open Source Software*, vol. 6, p. 2783, May 2021. doı: 10.21105/joss.02783.

[6] T. Qin, P. Li, S. Shen, "Vıns-mono: A robust and versatile monocular visual-ınertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018. doı: 10.1109/TRO.2018.2853729.

[7] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–ınertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021. doı: 10.1109/TRO.2021.3075644.

[8] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, D. Rus, *Lıo-sam: Tightly-coupled lidar ınertial odometry via smoothing and mapping*, 2020. arXiv: 2007.00258 [cs.RO].

[9] J. Zhang, S. Singh, "Loam : Lidar odometry and mapping in real-time," *Robotics: Science and Systems Conference (RSS)*, pp. 109–111, Jan. 2014.

[10] T. Shan, B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4758–4765. doı: 10.1109/IROS.2018.8594299.

[11] T. Shan, B. Englot, C. Ratti, D. Rus, *Lvı-sam: Tightly-coupled lidar-visual-ınertial odometry via smoothing and mapping*, 2021. arXiv: 2104.10831 [cs.RO].

[12] G. D. Pais, P. Miraldo, S. Ramalingam, V. M. Govindu, J. C. Nascimento, R. Chellappa, "3dregnet: A deep neural network for 3d point registration," *CoRR*, vol. abs/1904.01701, 2019. arXiv: 1904.01701. [Online]. Available: http://arxiv.org/abs/1904.01701.

[13] H. M. S. Bruno, E. L. Colombini, "Lıft-slam: A deep-learning feature-based monocular visual slam method," *Neurocomputing*, vol. 455, pp. 97–110, 2021, ıssn: 0925-2312. doı: https://doi.org/10.1016/j.neucom.2021.05.027. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231221007803.

[14] A. Nash, S. Koenig, "Any-angle path planning," *AI Magazine*, vol. 34, no. 4, pp. 85–107, 2013. doı: https://doi.org/10.1609/aimag.v34i4.2512. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1609/aimag.v34i4.2512. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1609/aimag.v34i4.2512.

[15] J. Barraquand, B. Langlois, J.-C. Latombe, "Numerical potential field techniques for robot path planning," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 2, pp. 224–241, 1992. doı: 10.1109/21.148426.

[16] H.-P. Huang, S.-Y. Chung, "Dynamic visibility graph for path planning," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, 2004, 2813–2818 vol.3. doı: 10.1109/IROS.2004.1389835.

[17] S. Choi, J. Park, E. Lim, W. Yu, "Global path planning on uneven elevation maps," in *2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2012, pp. 49–54. doı: 10.1109/URAI.2012.6462928.

[18] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, 1985, pp. 500–505. doı: 10.1109/ROBOT.1985.1087247.

[19] F. Bayat, S. Najafinia, M. Aliyari, "Mobile robots path planning: Electrostatic potential field approach," *Expert Systems with Applications*, vol. 100, pp. 68–78, 2018, ıssn: 0957-4174. doı: https://doi.org/10.1016/j.eswa.2018.01.050. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417418300630.

[20] P. E. Hart, N. J. Nilsson, B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968. doı: 10.1109/TSSC.1968.300136.

[21] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, Dec. 1959, ıssn: 0945-3245. doı: 10.1007/BF01386390. [Online]. Available: https://doi.org/10.1007/BF01386390.

[22] S. Koenig, M. Likhachev, "D*lite," Edmonton, Alberta, Canada: American Association for Artificial Intelligence, 2002, pp. 476–483, ısbn: 0262511290.

[23] S. Karaman, E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011. doı: 10 . 1177 / 0278364911406761. eprint: https : / / doi . org / 10 . 1177 / 0278364911406761. [Online]. Available: https://doi.org/10.1177/0278364911406761.

[24] S. M. LaValle, "Rapidly-exploring random trees : A new tool for path planning," *The annual research report*, 1998.

[25] T. Ming, W. Deng, S. Zhang, B. Zhu, "Mpc-based trajectory tracking control for ıntelligent vehicles," Apr. 2016. doı: 10.4271/2016-01-0452.

[26] K. Åström, T. Hägglund, "The future of pıd control," *Control Engineering Practice*, vol. 9, no. 11, pp. 1163–1175, 2001, PID Control, ıssn: 0967-0661. doı: https : / / doi . org / 10 . 1016 / S0967 - 0661(01) 00062 - 4. [Online]. Available: https : / / www . sciencedirect . com / science/article/pii/S0967066101000624.

[27] K. Oh, J. Seo, "Development of a sliding-mode-control-based path-tracking algorithm with model-free adaptive feedback action for autonomous vehicles," *Sensors*, vol. 23, no. 1, 2023, ıssn: 1424-8220. doı: 10 . 3390 / s23010405. [Online]. Available: https://www.mdpi.com/1424-8220/23/1/405.

[28] J. Ni, X. Wang, M. Tang, W. Cao, P. Shi, S. X. Yang, "An ımproved real-time path planning method based on dragonfly algorithm for heterogeneous multi-robot system," *IEEE Access*, vol. 8, pp. 140 558–140 568, 2020. doı: 10.1109/ACCESS.2020.3012886.

[29] C. Potena, R. Khanna, J. Nieto, R. Siegwart, D. Nardi, A. Pretto, "Agricolmap: Aerial-ground collaborative 3d mapping for precision farming," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1085–1092, 2019. doı: 10.1109/LRA.2019.2894468.

[30] C. Ju, H. Son, "Hybrid systems based modeling and control of heterogeneous agricultural robots for field operations," Jan. 2019. doı: 10.13031/aim.201900316.

[31] M. Quigley *et al.*, "Ros: An open-source robot operating system," vol. 3, Jan. 2009.

[32] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, D. Rus, "Lıo-sam: Tightly-coupled lidar ınertial odometry via smoothing and mapping," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5135–5142. doı: 10.1109/IROS45743.2020.9341176.

[33] F. Dellaert, M. Kaess, "Factor graphs for robot perception," *Foundations and Trends® in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017, ıssn: 1935-8253. doı: 10.1561/2300000043. [Online]. Available: http://dx.doi.org/10.1561/2300000043.

[34] P. Fankhauser, M. Hutter, "A universal grid map library: Implementation and use case for rough terrain navigation," in Jan. 2016, vol. 625, ısbn: 978-3-319-26052-5. doı: 10.1007/978-3-319-26054-9_5.

[35] P. Fankhauser, M. Hutter, "A Universal Grid Map Library: Implementation and Use Case for Rough Terrain Navigation," in *Robot Operating System (ROS) – The Complete Reference (Volume 1)*, A. Koubaa, Ed., Springer, 2016, ch. 5, ısbn: 978-3-319-26052-5. doı: 10.1007/978-3-319-26054-9{_}5. [Online]. Available: http://www.springer.com/de/book/9783319260525.

[36] T. S. 1. author, *Probabilistic Robotics*. The MIT Press, 2005.

[37] J.-M. Lin, C. Lin, "A novel fuel cell system design by using ziegler-nichols-based ıntelligent fuzzy controller," *Applied Mechanics and Materials*, vol. 479-480, pp. 934–937, Dec. 2013. doı: 10.4028/www.scientific.net/AMM.479-480.934.

[38] S. Shimoda, Y. Kuroda, K. Iagnemma, "High-speed navigation of unmanned ground vehicles on uneven terrain using potential fields," *Robotica*, vol. 25, pp. 409–424, Jul. 2007. doı: 10.1017/S0263574706003171.

[39] *Clearpath husky ros wiki*. [Online]. Available: http://wiki.ros.org/Robots/Husky (visited on 07/11/2023).

[40] *Hector quadrotor ros wiki*. [Online]. Available: http://wiki.ros.org/hector_quadrotor (visited on 07/11/2023).

[41] *Microstrain 3dm-rq1-45 datasheet*. [Online]. Available: http://files.microstrain.com/RQ1-datasheet.pdf (visited on 07/11/2023).

# TEZDEN ÜRETİLMİŞ YAYINLAR

## Konferans Bildirisi

1. Kaya, Ö., Uslu E.; "Elevation Based Outdoor Navigation with Coordinated Heterogeneous Robot Team", International Symposium on Intelligent Manufacturing and Service Systems (IMSS), 2023 12th, 26-28 May 2023