T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

KLASİK YOL PLANLAMA ALGORİTMALARININ SENTEZİYLE OLUŞTURULAN DAİRESEL DÜĞÜM ALGORİTMASININ OTONOM MOBİL ROBOT ÜZERİNDE UYGULANMASI

Anıl Gökhan GÖKTAŞ

YÜKSEK LİSANS TEZİ

Makine Mühendisliği Anabilim Dalı

Makine Teorisi ve Kontrol Programı

Danışman

Prof. Dr. Semih SEZER

Temmuz, 2023

T.C.

YILDIZ TEKNİK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

KLASİK YOL PLANLAMA ALGORİTMALARININ SENTEZİYLE OLUŞTURULAN DAİRESEL DÜĞÜM ALGORİTMASININ OTONOM MOBİL ROBOT ÜZERİNDE UYGULANMASI

Anıl Gökhan GÖKTAŞ tarafından hazırlanan tez çalışması 25.07.2023 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Makine Mühendisliği Anabilim Dalı, Makine Teorisi ve Kontrol Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Prof. Dr. Semih SEZER
Yıldız Teknik Üniversitesi
Danışman

Jüri Üyeleri

Prof. Dr. Semih SEZER, Danışman
Yıldız Teknik Üniversitesi _____

Prof. Dr. Vasfi Emre ÖMÜRLÜ, Üye
Yıldız Teknik Üniversitesi _____

Prof. Dr. Erdinç ALTUĞ, Üye
İstanbul Teknik Üniversitesi _____

Danışmanım Prof. Dr. Semih SEZER sorumluluğunda tarafımca hazırlanan “Klasik Yol Planlama Algoritmalarının Senteziyle Oluşturulan Dairesel Düğüm Algoritmasının Otonom Mobil Robot Üzerinde Uygulanması” 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.

Anıl Gökhan GÖKTAŞ

Bu çalışma, “Yıldız Teknik Üniversitesi Bilimsel Araştırma Proje Koordinatörlüğü’nün FYL-2022-4880” numaralı projesi ile desteklenmiştir.



Aileme

TEŞEKKÜR

Bilgisi, tecrübeleri ve yönlendirmeleriyle her zaman yanımda olan, gerektiği zaman esneklik sağlayan değerli danışman hocam Prof. Dr. Semih SEZER'e içten duygularımla teşekkürlerimi sunuyorum.

Bugünlere kadar gelmeme katkı sağlayan, her zaman arkamda olan, işlerimin kolaylaşması için bana hep destek olan babam Servet GÖKTAŞ, annem Güler GÖKTAŞ ve kardeşim İlayda GÖKTAŞ'a teşekkür ediyorum.

Anıl Gökhan GÖKTAŞ

İÇİNDEKİLER

TEŞEKKÜR vi
SİMGE LİSTESİ viii
KISALTMA LİSTESİ ix
ŞEKİL LİSTESİ xi
TABLO LİSTESİ xiv
ÖZET xvi
ABSTRACT xviii
1 GİRİŞ 1
1.1 Tezin Amacı 2
1.2 Literatür Özeti 2
2 ROBOTUN TASARIMI 22
2.1 Robotun Tasarımı 22
2.2 Malzeme 25
2.3 Elektronik Malzemeler 32
2.4 Mobil Robotun Hareket Konfigürasyonları 37
2.5 Mobil Robotun Tekerlek Konfigürasyonları 38
3 YOL PLANLAMA YÖNTEMLERİ 43
3.1 Çevre Bilgisine Dayalı Yol Planlama 43
3.2 Bütünlüğe Dayalı Yol Planlama 44
4 ROBOTİK İŞLETİM SİSTEMİ 56
4.1 ROS Paket Yapısı 56
4.2 ROS Dosya Yapısı 58
4.3 Koordinat Sistemleri 61
4.4 Haritalama 62
4.5 AMCL 64
5 YOL PLANLAMA ALGORİTMALARININ SİMÜLASYON ÇALIŞMALARI 66
5.1 PRM Algoritması Simülasyon Çalışmaları 66
5.2 RRT* Algoritması Simülasyon Çalışmaları 76
5.3 Dairesel Düğüm Algoritması 83
6 SONUÇLAR VE ÖNERİLER 102
KAYNAKÇA 105
TEZDEN ÜRETİLMİŞ YAYINLAR 112

SİMGE LİSTESİ

a	Acceleration
℃	Celcius
F_r	Frictional Force
F_s	Shear Force
*	Star
μ	Static Friction Coefficient
F_x	Translational Force

KISALTMA LİSTESİ

3D	3 Dimensional
A	Amper
ABD	Amerika Birleşik Devletleri
AGV	Autonomous Ground Vehicle
Ah	Amper.hour
AMCL	Adaptive Monte Carlo Localization
AMR	Autonomous Mobile Robot
CM	Centimeter
CNC	Computer Numerical Control
CPP	C++ Kaynaklı Kod
DC	Direct Circuit
DK	Dakika
G	Gram
GPa	GigaPascal
ISO	International Organization for Standardization
K	Kelvin
KG	Kilogram
KJ	Kilojoule
LIDAR	Light Detection and Ranging
M	Meter
M.Ö.	Milattan Önce
MAh	MiliAmper
MM	Milimeter
MPa	MegaPascal
MSG	Message
N	Reaction Force
NM	Nanometer
PLA	Poliaktik Asit
PRM	Probabilistic Roadmap
ROS	Robotic Operating System
RPM	Revolutions per Minute
RRT*	Rapidly Exploring Random Tree Star

RUR	Rosumovi Umell Roboti
RViz	Robotic Operating System Visualization
S	Second
SLAM	Simultaneous Localization and Mapping
SRC	Source
SRV	Service
USB	Universal Serial Bus
V	Voltage
W	Watt

ŞEKİL LİSTESİ

Şekil 1.1 "Rossum'un evrensel robotları" oyunundaki robot [2] **3**

Şekil 1.2 a)Stanford robot kolu [10], b) Sojourner [15], c) Sprit and Oppurtunity [16], d) StarFish [17] e) Robonaut [18] **7**

Şekil 1.3 Robotların sınıflandırılması [20], [21] **9**

Şekil 1.4 KUKA mobil robot [22] **12**

Şekil 1.5 Uranus çok yönlü mobil robot [29] **14**

Şekil 1.6 MEGAN-1 mobil robotu [30] **14**

Şekil 1.7 Küpik ve beşli eğri segment hesabı için kullanılan mobil robot [33] **15**

Şekil 1.8 90 kilogram yük taşıyabilen mobil robot [34] **16**

Şekil 1.9 Markom'un ışıkla uzaklık algılayıcıya sahip mobil robotu [38] **17**

Şekil 1.10 Mobil robot slam uygulaması [25] **17**

Şekil 1.11 Kavrayıcılı mobil robot [40] **18**

Şekil 1.12 Nagowah'ın mobil robotu [41] **18**

Şekil 1.13 Xi'nin çalışmasındaki mobil robot [42] **19**

Şekil 1.14 Rivai'nin mobil robotunun engel tespit çalışması [43] **19**

Şekil 1.15 Palanisamy'nin mobil robotunun gui arayüzü [44] **20**

Şekil 1.16 Hartono'nun mobil robot platform [45] **20**

Şekil 1.17 a) Aldair'in simulasyon testleri, b) Aldair'in deneysel testleri [46] **21**

Şekil 2.1 Tasarlanan mobil robot **22**

Şekil 2.2 Robotun parçaları **23**

Şekil 2.3 Robotun 2D ölçüleri **23**

Şekil 2.4 Ön bölümün 3D görünümü ve birleştirilmiş hali **24**

Şekil 2.5 Gövdenin üretilmiş ve birleştirilmiş hali **24**

Şekil 2.6 Yük taşıyıcı bölümün 3D ve 2D görünümü **25**

Şekil 2.7 Porima PLA filamentinin malzeme özellikleri [49] **26**

Şekil 2.8 Doluluk oranı değişiminde parçadaki ağırlık değişimi **27**

Şekil 2.9 Doluluk oranı değişiminde baskı süresinde ağırlık değişimi 28

Şekil 2.10 a) ISO-527 numune ölçüleri, b) Numunelerin 3D baskıları **28**

Şekil 2.11 a) Çekme testi için kullanılan teçhizat, b) Numenin kopması, c) Bilgi sayar üzerinden sonuçların gözlemlenmesi **29**

Şekil 2.12 Kopma sonrası tüm numunelerin görünümleri **30**

Şekil 2.13 %100 dolu numunenin gerilme-gerinim grafiği **30**

Şekil 2.14 %60 dolu numunenin gerilme-gerinim grafiği **31**

Şekil 2.15 %40 dolu numunenin gerilme-gerinim grafiği **31**

Şekil 2.16 %30 dolu numunenin gerilme-gerinim grafiği **32**
Şekil 2.17 Mobil robotun donanımsal yapısı **33**
Şekil 2.18 Mecanum tekerleklere sahip robotun hareket konfigürasyonları[54] .. **37**
Şekil 2.19 Mecanum tekerlek konfigürasyonu **38**
Şekil 2.20 Sol ön teker konfigürasyonu **39**
Şekil 2.21 Mobil robotun robot düzleminde konfigürasyonu **40**
Şekil 2.22 Mobil robotun eğik yüzeyde durma pozisyonu **41**
Şekil 2.23 Eğik düzlem üzerinde etki eden kuvvetler **41**
Şekil 3.1 Klasik algoritma yöntemleri **45**
Şekil 3.2 Engellerin olduğu statik haritanın hücrelere bölünmesi **46**
Şekil 3.3 Yapay potansiyel alanlar algoritma yöntemi[61] 46
Şekil 3.4 Görünürlük grafiği [62] **47**
Şekil 3.5 Voronoi yöntemi [63] **48**
Şekil 3.6 Olasılıksal yol planlama algoritması **49**
Şekil 3.7 Hızlı keşfedilen gelişigüzel ağaçlar yol planlaması **50**
Şekil 3.8 Dijkstra algoritması [66] **51**
Şekil 3.9 Yapay sinir ağları yapısı [68] **52**
Şekil 3.10 Gri kurtların hiyerarşik yapısı [72] **53**
Şekil 3.11 Karınca kolonisinin yeni yol oluşturması [73] **54**
Şekil 3.12 Genetik algoritma ile yol planlaması örneği [74] **55**
Şekil 4.1 ROS paket yapısı **56**
Şekil 4.2 ROS haberleşme yöntemi **57**
Şekil 4.3 ROS master çalışma şeması **57**
Şekil 4.4 Roscore komutu ile ROS yöneticisinin aktifleştirme **58**
Şekil 4.5 ROS çalışma alanı **59**
Şekil 4.6 ROS dosya yapısı **59**
Şekil 4.7 RPLidar düğüm kodları **60**
Şekil 4.8 Koordinat sistemi yapısı [79] **61**
Şekil 4.9 Rviz üzerinden çalışma ortamının haritalaması **62**
Şekil 4.10 a) AMCL algoritması ilk tahmin, b) Hareketin başlamasıyla konum tah minindeki iyileşme, c) Hareketin bitişi sırasındaki konum tahmini ... **64**
Şekil 5.1 Simülasyon çalışmalarının yapıldığı harita **66**
Şekil 5.2 a) Engellerin şişkinlik yarıçapının 0.2 metre olduğu simülasyon denemesi, b) Engellerin şişkinlik yarıçapının 0.3 metre olduğu simülasyon denemesi **68**

Şekil 5.3 a) 1. durum, b) 2. durum, c) 3. durum, d) 4. durum, e) 5. durum, f) 6. Durum, g) 7. durum, h) 8. durum, i) 9. durum, j) 10. durum, k) 11. durum..... **71**

Şekil 5.4 a) 1. durumda gerekli olan düğüm noktaları, b) Kırmızı bölgedeki düğüm noktalarının sayısı **72**

Şekil 5.5 a) 500 iterasyon, b) 1000 iterasyon, c) 2000 iterasyon, d) 3000 iterasyon, e) 4000 iterasyon, f) 5000 iterasyon, g) 6000 iterasyon, h) 7000 iterasyon, i) 8000 iterasyon, j) 9000 iterasyon..... **78**

Şekil 5.6 a) 9000 iterasyona sahip çalışmada gerekli olan düğüm noktaları, b) Kırmı zı bölgedeki düğüm noktalarının sayısı **79**

Şekil 5.7 a) Başlangıç ve bitiş noktalarının belirlenmesi b) Düğüm noktalarının atanması, c) Engele yakın olan noktaların tespit edilmesi d) Tespit edilen noktaların silinmesi, e) Kalan noktalar arasından hedefe en yakın olanın bulunması ve hareket edilmesi, f) Hedefe yaklaşana kadar aynı adımların tekrarlanması **85**

Şekil 5.8 Hedef düğüm noktasının tespiti..... **86**

Şekil 5.9 Laboratuvar ortamının simülasyon çalışmaları için haritalanması..... **86**

Şekil 5.10 Laboratuvar ortamında algoritmanın denemeleri **87**

Şekil 5.11 ROS simülasyon çalışmalarının gerçekleştirildiği harita **87**

Şekil 5.12 ROS simülasyonunda dairesel düğüm noktaların oluşturulması..... **88**

Şekil 5.13 Robot merkezinden 0.5 metre uzaklıkta 10 düğüm noktasına sahip örnek yol planlaması **88**

Şekil 5.14 Robot merkezinden 0.3 metre uzaklıkta 10 düğüm noktasına sahip örnek yol planlaması **89**

Şekil 5.15 Deneysel çalışmalarda uygun olmayan düğüm noktalarının silinmesi **97**

Şekil 5.16 a) Noktaların oluşturulması ve ara hedefe yönelim, b) Ara hedefe ulaşım, c) Yeni ara hedefin belirlenmesi **99**

Şekil 5.17 Robotun merkeziyle düğüm noktaları arasındaki uzaklığın 0.3 metre olduğu test denemesi..... **99**

Şekil 5.18 Engelin olduğu durumda deneysel çalışma **100**

Şekil 5.19 a) İlk ara hedefe doğru yönelme, b) İlk ara hedefe gelme, c) Engelden kaçınarak oluşturulan 2. ara hedefe yönelme, d) 3. Ara hedefe yönelim **101**

TABLO LİSTESİ

Tablo 1.1 Robotların tarihçesi [9]–[13] **5**

Tablo 1.2 Robotların Kullanımındaki Avantajlar ve Dezavantajlar [18] **8**

Tablo 1.3 Robotların tekerlek dizilimine göre yapıları [20] **10**

Tablo 1.4 Tablo 1.3'de gösterilen robotların tekerlek dizilim modelleri [20] **11**

Tablo 1.5 AMR ve AGR arasındaki farklar [21]–[23] **13**

Tablo 2.1 Alüminyum 5754 alaşımının fiziksel ve mekanik özellikleri [48] **25**

Tablo 2.2 3D yazıcı baskı ayarları **26**

Tablo 2.3 Belirlenen bir parçanın doluluk oranlarına karşılık baskı süresi ve ağırlık değerleri **27**

Tablo 2.4 Raspberry Pi 4 Model B özellikleri [51] **34**

Tablo 2.5 Kullanılan motorun özellikleri [52] **35**

Tablo 2.6 RPlidar A1 M8 özellikleri [53] **36**

Tablo 3.1 Yol planlama yöntemi oluşturulabilecek durumlar **43**

Tablo 3.2 Küresel ve yerel yol planlaması [60], [61] **44**

Tablo 5.1 PRM algoritması simülasyon parametreleri **67**

Tablo 5.2 Simülasyonların gerçekleştirildiği durumlar **68**

Tablo 5.3 Yol uzunluk değerleri **71**

Tablo 5.4 Gerekli düğüm sayıları ve noktaların verimlilik verileri **73**

Tablo 5.5 Engellerin şişkinlik yarıçapı 0.3 olduğunda toplam yol uzunlukları **74**

Tablo 5.6 Engelleri şişkinlik yarıçapı değeri 0.3 olduğunda gerekli düğüm sayıları ve noktaların verimlilik verileri **75**

Tablo 5.7 RRT* algoritması simülasyon parametreleri **76**

Tablo 5.8 RRT* algoritmasında engel yarıçapının 0.2 metre olduğu simülasyon çalışmalarında toplam yol uzunluk değerleri **79**

Tablo 5.9 RRT* algoritmasında engel yarıçaplarının 0.2 metre olduğu simülasyon çalışmalarında gerekli düğüm sayıları ve noktaların verimlilik verileri **80**

Tablo 5.10 RRT* algoritmasında engel yarıçapının 0.3 metre olduğu simülasyon çalışmalarında toplam yol uzunluk değerleri **81**

Tablo 5.11 RRT* algoritmasında engel yarıçaplarının 0.2 metre olduğu simülasyon çalışmalarında gerekli düğüm sayıları ve noktaların verimlilik verileri **82**

Tablo 5.12 Dairesel düğüm algoritması için simülasyon durumları **90**

Tablo 5.13 Test 1 sonucu elde edilen yol uzunluğu ve düğüm noktalarının verimlilik sonuçları **90**

Tablo 5.14 Test 2 sonucu elde edilen yol uzunluğu ve düğüm noktalarının verimlilik sonuçları **91**

Tablo 5.15 Test 3 sonucu elde edilen yol uzunluğu ve düğüm noktalarının verimlilik sonuçları **92**

Tablo 5.16 Test 4 sonucu elde edilen yol uzunluğu ve düğüm noktalarının verimlilik sonuçları **94**

Tablo 5.17 Test 5 sonucu elde edilen yol uzunluğu ve düğüm noktalarının verimlilik sonuçları **95**

Tablo 5.18 Test 6 sonucu elde edilen yol uzunluğu ve düğüm noktalarının verimlilik sonuçları **96**

ÖZET

Klasik Yol Planlama Algoritmalarının Senteziyle Oluşturulan Dairesel Düğüm Algoritmasının Otonom Mobil Robot Üzerinde Uygulanması

Anıl Gökhan GÖKTAŞ

Makine Mühendisliği Anabilim Dalı

Makine Teorisi ve Kontrol Programı

Yüksek Lisans Tezi

Danışman: Prof. Dr. Semih Sezer

Bu tez çalışmasında endüstriyel sektörde üretimin, teslimatın ve tedarik işlemlerin hızlı yapılabilmesine yönelik, kapalı alanlarda kullanılabilecek bir mobil robot tasarlanmıştır. Robotun hareketleri, çok yönlü çalışabilir olmasından dolayı hareket kabiliyeti kazandıran çok yönlü tekerlekler ile sağlanmıştır. Mobil robotun ana kontrolcüsü için içerisine ROS sisteminin kurulabildiği Raspberry Pi 4 cihazı kullanılmıştır. ROS üzerinden mobil robotun mevcut konum tahmini, gidilecek olan hedef nokta seçimi, başlangıç ve bitiş noktaları arasındaki yolun planlaması ve kullanılan ışıkla uzaklık algılayıcı sensörü ile dışarıdan gelen etkenlere karşı anlık önlem alma işlemleri yapılabilmektedir. Mobil robotun gideceği hedef noktanın seçilmesinin ardından Arduino'ya gönderilen komutlarla birlikte motorlar tahrik edilmektedir.

Robotun bulunduğu konumdan hedef konuma gitmesini sağlamak için yol planlama algoritmalarına ihtiyaç bulunmaktadır. Literatürde kullanılmakta olan birçok yol planlama algoritmasının birbirlerine göre avantaj ve dezavantajları bulunmaktadır. Statik haritalarda kullanılmakta olan PRM ve geliştirilmiş olan RRT* algoritmaları

kullanılarak MATLAB üzerinden simülasyon denemeleri gerçekleştirilmiş, algoritmaların çalışma yöntemlerindeki verimlilik incelenmiş ve belirlenen başlangıç ve hedef noktaları arasındaki yol uzunlukları hesaplanarak karşılaştırma yapılmıştır. Statik durumda kullanılan bu algoritmalar birlikte sentezlenerek düğüm noktalarının kullanıldığı, dinamik engellere karşı önlem alınabildiği yeni bir algoritma geliştirilmiştir. Aynı harita üzerinde geliştirilen algoritmanın kullanılmasıyla birlikte elde edilen simülasyon sonuçları değerlendirilmiştir.

Elde edilen sonuçlarda; PRM algoritmasında 1300 adet düğüm noktası kullanıldığında, düğüm noktası verimliliğinin 0.295 oranında olduğu ve optimum yol uzunluğunun 1,49 metre olduğu; RRT* algoritmasında 8000 iterasyon kullanıldığında, düğüm noktası verimliliği 0.198 oranında olduğu ve optimum yol uzunluğunun 1,44 metre olduğu görülmüştür. Geliştirilen algoritmayla 3 adımda 0,688 oranında düğüm noktası verimliliğiyle 1,42 metre uzunluğunda optimum yola ulaşılmıştır.

Anahtar Kelimeler: Işıklı uzaklık algılayıcı, mobil robot, PRM, ROS, RRT*

ABSTRACT

Application of the Circular Node Algorithm Created by the Synthesis of Classical Path Planning Algorithms on an Autonomous Mobile Robot

Anıl Gökhan GÖKTAŞ

Department of Mechanical Engineering

Master of Science

Supervisor: Prof. Dr. Semih SEZER

In this thesis study, a mobile robot that can be used indoors has been designed for fast production, delivery and supply processes in the industrial sector. The robot's movements have multi-directional wheels that provide mobility due to its versatility. Raspberry Pi 4, in which the ROS system can be installed, is used as the main controller of the mobile robot. The mobile robot's current location estimation, selection of the target point to be reached, planning the path between the start and end points, and instant measures can be taken against external factors with the LIDAR sensor used via ROS. After the target point of the mobile robot is selected, its motors are driven with the commands sent to the Arduino. There is a need for path planning algorithms where you want the robot to go from its location to the target location. Many path planning algorithms used in the literature have advantages and disadvantages compared to each other. Simulation experiments were carried out on MATLAB using the PRM used in static maps and the developed RRT* algorithms, the efficiency in the working methods of the algorithms was examined. The path lengths between the determined starting and target points were calculated and compared. These algorithms used in the static situation were

synthesized together and a new algorithm was developed, which uses nodal points and can take precautions against dynamic obstacles. The simulation results were evaluated using the algorithm developed on the same map. In the results obtained, it has been observed that the optimum path length in the PRM algorithm is 1.49 meters when 1300 points are used with a nodal efficiency of 0.295 and 1.44 meters when 8000 iterations are used with a nodal efficiency of 0.198 in the RRT* algorithm. With the developed algorithm, the optimum path of 1.42 meters has been reached with a nodal efficiency of 0.688 in 3 steps.

Keywords: Light detection and ranging, mobile robot, PRM, ROS, RRT*

1
GİRİŞ

İnsanoğlu yaşadığı hayatı daha güzel ve rahat kılabilmek için yenilikler arayışı içine girmiştir. Bu yenilikler arasından popüler ve insan dışı bir varlık olan robotlar, bunun en büyük örneklerinden birisidir. Farklı amaçlar doğrultusunda eklenen yenilikler ve gelişmelerle hayatımızın birçok yerinde kullanılmaktadır. Günümüzde yapılan bu yeniliklerin birçoğuna farkında varılmadan rastlanılmaktadır.

M.Ö. başlayan ve 20. yüzyılda hızlanarak günümüze kadar devasa bir hale gelmiş olan robot teknolojisi sınır tanımamaktadır. Robot araçlarının kullanımın artmasıyla birlikte firmalar arasında rekabet ortamı ortaya çıkmıştır. Sanayi firmalarında üretim ve verimlilik arttırıp, insan gücünü ve maliyeti azaltma düşüncesiyle kullanılmaktadır. Robotların da kullanımındaki asıl amaç bunları sağlamaktır. Dolayısıyla bu faktörler de piyasa rekabetini etkilemektedir.

Robotlar, programlama vasıtasıyla doğrudan ve dolaylı olarak kendisine verilmiş olan görevi yerine getirmek için sahip olduğu mekanik ve elektronik aksamlar kullanan, çok işlevsel özelliği sahip aygıtlardır. Robotların en fazla kullanıldığı alan ise endüstriyel alandır. Endüstriyel robot olarak adlandırılan robot kolları kaynak, montaj, boyama, taşıma, paletleme gibi işlemler için tercih edilir. Hem endüstriyel hem de lojistik alanında kullanılabilen mobil robotlar ise disiplinler arası robotlar sınıfına girmektedir. Lojistik alanda kullanılabilir olmasından dolayı sabit bir noktada kalmaz ve sürekli hareket halindedir. İnsan ile etkileşimli çalışabilen mobil robotlar, robot kol vasıtasıyla da insansız görev robotu haline gelebilmektedir.

İmalat sanayisinde üretilen ürünler birçok işlemden geçerek son halini almaktadır. Bu işlemler ise sanayi kuruluşunun farklı bölgelerinde gerçekleşir. Hammadde, imalat bölümüne taşınır. İşleme sonrası montaj bölümüne alınır. Ardından paketleme yapılır ve ürün lojistik bölümüne gönderilir. Bu taşıma işlemleri ise forklift, transpalet gibi yardımcı elemanlar kullanılarak insanlar tarafından yapılır. Robot teknolojisinde daha önce de bahsedildiği gibi amaç verimliliği arttırmak ve insan gücünü azaltmaktır. Bu sebeple taşıma işlemi mobil robotların kullanılmasıyla gerçekleştirilebilir.

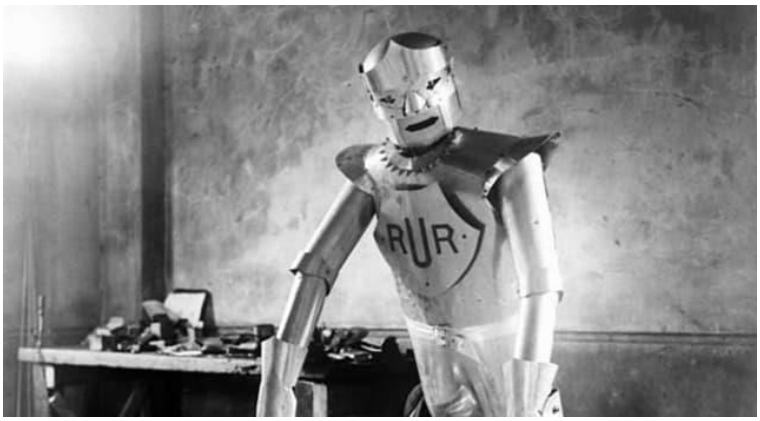
1.1 Tezin Amacı

Mobil robotların bir konumdan başka bir konuma otonom olarak hareketi için kullanılmakta olan birçok algoritma bulunmaktadır. Bu algoritmaların bazı durumlarda birbirlerine göre avantajları ve dezavantajları oluşmaktadır. Bu çalışmadaki literatürde mevcut olan fakat eksik yanları bulunan bazı yöntemlere karşı olarak, benzer çalışma mantığıyla yöntemi değiştirerek, daha etkin bir yol planlaması gerçekleştirilmiştir. Geçmişte kullanılan ve hala kullanılmakta olan algoritmaların simülasyon sonuçlarının karşılaştırılıp, sonuçların değerlendirilmesiyle oluşturulabilecek yeni bir algoritma ile daha verimli sonuçların alınabileceği ön görülmektedir. Sonuçların daha verimli olduğuna ulaşılmasının ardından deneysel olarak algoritmanın doğruluğu, farklı senaryolarda çalışma şekli ve sonuçların değerlendirilmesi amaçlanmıştır. Deneysel çalışmalarla birlikte karşısına çıkabilecek engelleri görebilen, yük taşıyabilen, düşük bütçeli otonom bir mobil robot geliştirilmesi ve bu robotun kontrol çalışmalarına, üretimine, kullanılan donanımına ve programlarına çalışma içinde yer verilmesi hedeflenmiştir.

1.2 Literatür Özeti

1.2.1 Robot Nedir

Slav dilinde “hizmet eden” anlamında gelen “robota” kelimesinin 1921 yılında Çek oyun yazarı Karel Čapek tarafından Rossum'un Evrensel Robotları (RUR, Rossum’s Universal Robots) isimli tiyatro oyununda kullanmasıyla başlayan yolculuk günümüzde bu kavramın robot olarak kullanılmasıyla devam etmektedir [1]. Oyunun temasında ise “İnsan makineyi yapar, makine insanı öldürür.” teması işlenmiştir. Tiyatro oyununda kullanılan robot Şekil 1.1’de gösterilmiştir.



Şekil 1.1 "Rossum'un evrensel robotları" oyunundaki robot [2]

Rus bilim kurgu yazarı Isaac Asimov tarafından 1940 yıllarında başlangıçta insanlığa hizmet etmek amacıyla kullanılan ardından insanlığı yok etme çabası içerisinde olan robotlara karşı robot yasaları oluşturulmuştur.

Sıfırıncı Yasa: Bir robot bir insana zarar veremez ve hareketsiz kalarak bir insana zarar gelmesine izin veremez [3].

Birinci Yasa: Bir robot insanlığa zarar veremez ve hareketsiz kalarak insanlığa zarar gelmesine izin veremez.

İkinci Yasa: Bir robot birinci yasa ile çatışmama koşulu ile kendisine verilen emirleri yerine getirmek zorundadır.

Üçüncü Yasa: Bir robot birinci ve ikinci yasa ile çatışmama koşulu ile kendi varlığını korumak zorundadır [4].

ABB robot firmasında baş mühendis olarak çalışmakta olan Stig Moberg, bu yasalara 2 yasa daha eklemiştir [5]. Bunlar:

1. Bir robot 3 yasa ile çatışmama koşulu ile yönlendiren kişinin belirlediği yörüngede hareket etmek zorundadır.
2. Bir robot diğer yasalar ile çatışmama koşulu ile yönlendiren kişinin belirlediği hız ve ivme ile hareket etmek zorundadır [5].

Fizikçi olan Mark W, Tilden robotların geliştirilmesiyle birlikte 3 yasa oluşturmuştur [6].

1. Bir robot ne olursa olsun varlığını korumalıdır.
2. Bir robot kendi güç kaynağına erişim sağlamalı ve bunu sürdürmelidir.
3. Bir robot sürekli olarak daha iyi güç kaynakları aramalıdır.

Robot kavramı için farklı kişi ve kurumlar tarafından farklı kelimeler kullanılarak tanımlamalar yapılmıştır.

Bilimsel olarak robotlar; mekanik sistemler, sensör sistemleri ve algoritmalarla birlikte çalışan akıllıca makinelerdir [7]. Öğrenebilen, yeniden programlanabilen, hareket edebilecekleri yolu değiştirebilen makinelerdir. Robotların bir kısmı insan modeline benzemese de insan özelliklerini taşırlar. Örneğin, endüstriyel robotlar insan modelinin dışında gerçekleştirilmesi zor olan görevleri yerine getirir. Günümüzde otonom insansız araçlar ve otomatik kontrollü makineler gibi cihazlar robot olarak tanımlanabilmektedir.

Bir makinenin robot olarak tanımlanabilmesi için 4 önemli faktör [8] gereklidir. Bunlar;

1. Algılama için gerekli olan sensörler,
2. Veri kontrolü için gerekli elektronik devreler,
3. Karar verebilmesi için programlama,
4. Gerekli hareketlerin yapılabilmesi için mekanik sistem.

1.2.2 Robotların Tarihçesi

Otomasyonun ilk kullanımı M.Ö.'ye dayanmaktadır. İkinci Dünya Savaşı sırasında araştırma sırasında yeni sistemler geliştirilmiş ve askeri teçhizatları geliştirilmesi için CNC (Bilgisayarlı Sayısal Kontrol) teknolojisi kullanılmıştır. Robot üretiminde bu teknoloji ön plana çıkmıştır. Teknolojinin zamanla birlikte ilerlemesiyle yapılan çalışmalar da değer kazanmıştır. Sanayi, askeri, ulaşım, keşif, yazılım gibi birçok alanda kullanılan teknolojilerle artık insanın tarafından yapılanların insan dışı varlıklar tarafından da gerçekleştirebileceğine ulaşılmış; insanın yapmasının uygun olmadığı durumlarda robotların kullanılmasıyla iş akışının ve planının değişmesi sağlanmıştır. Tablo 1.1'de robotların yaklaşık son 70 yılda kayda değer şekilde geliştiğini gösteren bazı çalışmalar verilmiştir.

Tablo 1.1 Robotların tarihçesi [9]–[14]

Yıl	Gelişme
1949	İlk mekanik tele manipülatör üretildi. (Raymond Goertz)
1954	İlk elektrik sürüşlü tele manipülatör üretildi. (Argonne National Laboratory)
1954	İlk endüstriyel robot patenti uygulaması gerçekleştirildi. (George C. Devol)
1961	İlk endüstriyel robot "Unimate" üretildi.
1962	İlk robot üretici firması "Unimation Inc." kuruldu.
1969	İlk elektrik sürüşlü ve 6 serbestlik derecesine sahip robot "Stanford Arm" üretildi. (Victor Scheinmann)
1978	SCARA robot kolu "Hiroshi" üretildi.
1996	"Honda" humanoid robotu üretildi.
1997	"Sojourner" mobil robotun Mars'a iniş gerçekleştirdi.
2001	Uzaktan robotlarla hasta ameliyatı (ABD'den Fransa'ya) yapıldı.
2001	Global Hawk insansız hava aracı Amerika'dan Avusturalya otonom uçuş gerçekleştirdi.
2002	iRobot temizlik robotu üretildi.
2003	"Sprit and Oppurtunity" mobil robotunun Mars'a iniş gerçekleştirdi.
2005	İlk çift kollu ticaret robotu "Yaskawa Motoman" üretildi.
2006	Cornell Üniversitesi'nde 4 ayaklı "Starfish" robot üretildi.
2006	Ağırlığı ile taşıma kapasite oranı 1 olan, 7 eksenli robot kolu "KUKA" üretildi.
2008	İnsansı bir robotun seri üretimi gerçekleştirildi. (NAO, Aldebaran)

Tablo 1.1 Robotların tarihçesi (devamı)

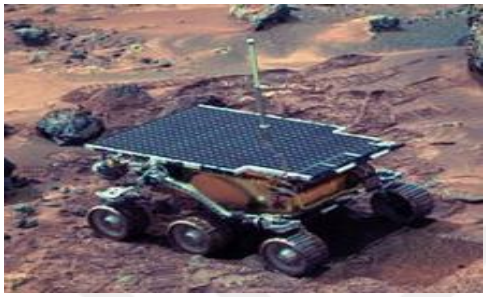
2011	Uluslararası Uzay İstasyonu'nda ilk humanoid robot "Robonaunt" imal edildi.
2016	Endüstri 4.0'a geçiş yapıldı.
2017	"Sophia" adında vatandaşlığı olan kadın insansı robot üretildi.
2017	Müzik orkestrası yöneten ilk çift kollu robot kolu "ABB YuMi" üretildi.
2017	Boston Dynamics tarafından insansız hava aracı benzeri hareket kabiliyetine sahip "SpotMini" geliştirildi.
2018	Yapay zekâ ile donatılan evcil hayvan robot projesi olan "Aibo" CES 2018'de tanıtıldı.
2018	Boston Dynamics tarafından "Atlas" otonom robot geliştirildi.
2019	Evde yardımcı olması amacıyla prototip olarak üretilen "BotCare" CES 2019'da tanıtıldı.
2019	Service Robots tarafından geliştirilen düğün fotoğrafçısı Eva kullanılmaya başlandı.
2020	"UBTECH" insansı robotu tanıtıldı.
2020	Boston Dynamics firmasının "Spot" robotu endüstriyel ve araştırma alanlarında kullanılmaya başlandı.
2021	Lojistik sektörü için taşıma robotu olan "Stretch" duyuruldu.
2021	Paket dağıtımı için tasarlanan "Digit" kullanılmaya başlandı.
2022	Summer Berry Company tarafından geliştirilen özel hasat robotları kullanılmaya başlandı.
2022	Bear Robotics tarafından Servi isimli otonom servis robotu tasarlandı.
2022	Volkswagen tarafından geliştirilen ilk yolcu insansız hava aracı prototipi Çin'de tanıtıldı.



a)



b)



c)



d)



e)

Şekil 1.2 a)Stanford robot kolu [10], b) Sojourner [15], c) Sprit and Oppurtunity [16], d) StarFish [17] e) Robonaut [18]

1.2.3 Robotların Kullanılma Nedenleri

Günümüzde endüstriyel alanda kullanılan robotlar hem fiziksel hem de karakteristik anlamda insanlara benzetilmeye çalışılmaktadır. Her ne kadar bir benzetme amacı içinde olunsa da insanın sahip olduğu his, düşünce, duyguya göre değişen davranış ve hissiyat sonrası vücutta gerçekleşen biyoloji değişimlerden dolayı bire bir benzerlik sağlanması mümkün görünmemektedir.

Robotlar sahip oldukları sensörler ve algılayıcılar sayesinden insanların yapamayacağı veya yapmasında tehlike barındıran görevlerin yüksek hassasiyetle

gerçekleştirilmesinde rol oynamaktadır. İşlevsel olarak insandan pek bir farkı bulunmamaktadır. Fakat karşılaştırma yapıldığında birçok avantaj ve dezavantajlar oluşmaktadır. Tablo 1.2'de robotların endüstriyel sanayide kullanılmasıyla birlikte sağladığı avantajlar ve dezavantajlar belirtilmiştir.

Tablo 1.2 Robotların kullanımındaki avantajlar ve dezavantajlar [19]

Avantajları	**Dezavantajları**
Tekrarlanabilirlik sayesinde üretim hızının artması	Bazı görevlerde yavaş olması
İşçilik maliyetinin azalması	Maliyetli olması
Operatör kaynaklı hata payının azalması	Karar verebilme yapısının düşük olması
Çalışan personele göre yorulma gibi bir durumun olmaması	Bakım gerektirmeleri
Tehlikeli ve insanın yapamayacağı görevleri yerine getirmesi	
Planlanan zamanda işin bitirilmesi	
Aynı anda birden fazla işin yapılması	

1.2.4 Robotların Sınıflandırılması

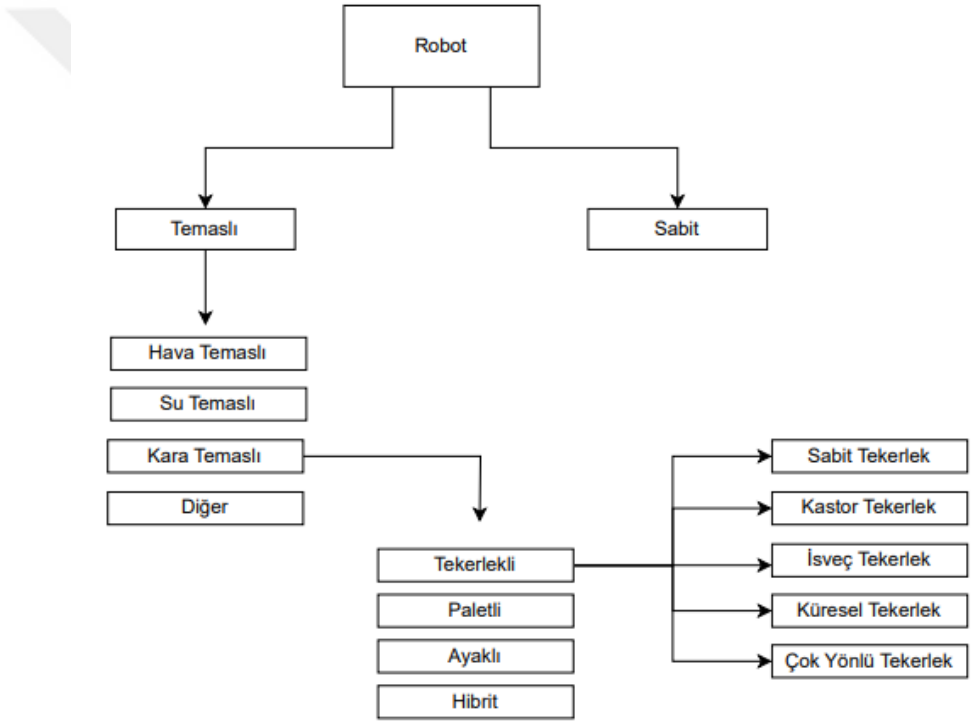
Robotlarda yapılması gereken görevin belirlenebilmesi için sahip olduğu yetenekleri değerlendirmede kullanılan 4 temel işlevi bulunmaktadır [20]. Bunlar;

1. Hareket,
2. Algılama,
3. Anlama ve karar verme,
4. Yörünge

Robotun hareket işlemiyle mekanik, dinamik ve kontrol yönteminin anlaşılması hedeflenir. Robotum tasarımı, çalışma ortamı ve farklı hareketlerinde kullanılacak olan kontrol edilebilirlik ve manevra kabiliyeti gibi teknik konuları içermektedir. Algılama işlemi, görev anında sensör teknolojisinin kullanılmasıyla robotun

etrafındaki nesnelere karşı tepki yapısının gerçekleştirebilmesi için gerekli olan sinyal analizini gerçekleştirir. Anlama işlemi, sensörlerden alınan verilerin ve gerçekleştirilecek olan hareketin yorumlanmasını sağlar. Yörünge işlemi algoritma planlanması ve yazılıma göre gerçekleşmektedir. Robotlar; yürüme, dönme, zıplama, koşma, yüzme gibi birçok hareketi gerçekleştirebilir yapıdadır.

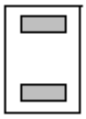
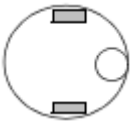
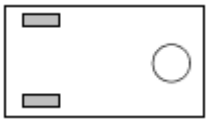
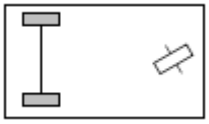
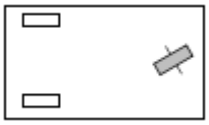
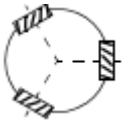
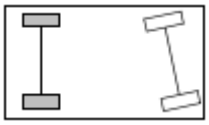
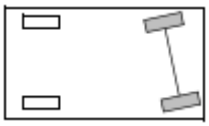
Geçmişten günümüzü hayatımızın içinde yer almakta olan robotları kullanım alanlarına ve yapılarına göre sınıflandırmak mümkündür. Genel olarak robotları endüstriyel ve mobil robotlar başlıkları altında sınıflandırmak mümkündür [21]. Yapılarına ve işlevsel özelliklerine göre ayrıştırma yapılarak birbirlerinden ayrılırlar. Şekil 1.3'te robotların temel sınıflandırılması gösterilmiştir.



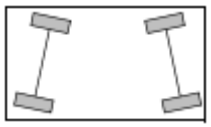
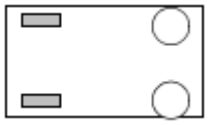
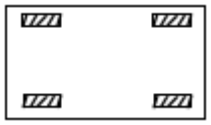
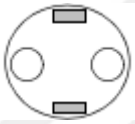
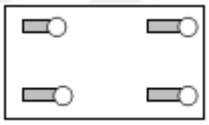
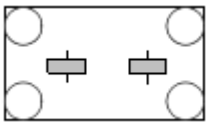
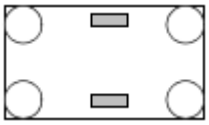
Şekil 1.3 Robotların sınıflandırılması [20], [21]

Tekerlek yapılarına göre sınıflandırılan robotlar, sahip oldukları tekerlerin konumlarına göre düzenlenerek gruplandırılabilir. Aynı robot üzerinde birden farklı tekerlek kullanılarak çeşitli konfigürasyonlarda yapılar oluşturulabilir. Minimum 2, maksimum 6 tekerlek kullanılarak robotun hareket ve manevra sınırlandırılmasına göre şasi yapıları şekillenebilir. Tablo 1.3 ve Tablo 1.4'te robotların tekerlek dizilimine göre değişen yapıları gösterilmiştir.

Tablo 1.3 Robotların tekerlek dizilimine göre yapıları [21]

Tekerlek Sayısı	Yapı Modeli	Kullanılan Tekerlekler
2		Önde yönlendirilebilen tekerlek; arkada motora sahip standart tekerlek
2		Solda ve sağda motora sahip standart tekerlek
3		Önde elektrik gücü olmayan, çok yönlü; yanlarda motora sahip standart tekerlekler
3		Önde elektrik gücü olmayan, çok yönlü; arkada motora sahip standart tekerlekler
3		Önde yönlendirilebilen tekerlek; arkada motora sahip bağlantılı standart tekerlekler
3		Önde motora sahip tekerlek; arkada elektrik gücü olmayan standart tekerlekler
3		120 derecelik açılarla motora sahip İsveç tekerlekler
4		Önde elektrik gücü olmayan tekerlekler; arkada elektrik gücü olan bağlantılı standart tekerlekler
4		Önde elektrik gücü olan bağlantılı tekerlekler; arkada elektrik gücü olmayan tekerlekler

Tablo 1.3 Robotların tekerlek dizilimine göre yapıları (devamı)

4		Önde ve arkada elektrik gücü olan bağlantılı standart tekerlekler
4		Önde elektrik gücü olmayan, çok yönlü tekerlekler; arkada elektrik gücü olan standart tekerlekler
4		Önde ve arkada motora sahip İsveç tekerlekler
4		Önde ve arkada elektrik gücü olmayan, çok yönlü tekerlekler; her iki yanda motora sahip standart tekerlekler
4		Önde ve arkada bulunan, motora sahip ve yönlendirilebilen standart tekerlekler
6		Önde ve arkada elektrik gücü olmayan, çok yönlü tekerlekler; ortada motora sahip sıralı standart tekerlekler
6		Önde ve arkada elektrik gücü olmayan, çok yönlü tekerlekler; ortada sol ve sağda motora sahip standart tekerlekler

Tablo 1.4 Tablo 1.3'te gösterilen robotların tekerlek dizilim modelleri [21]

	Elektrik gücüne sahip olmayan çok yönlü tekerlek
	Motora sahip İsveç tekerleği
	Elektrik gücüne sahip olmayan standart tekerlek
	Motora sahip standart tekerlek

Tablo 1.4 Tablo 1.3'te gösterilen robotların tekerlek dizilim modelleri (devamı)

	Motora sahip ve yönlendirilebilen tekerlek
	Yönlendirilebilen standart tekerlek
	Tekerlek bağlantısı

Çok yönlü tekerlekler ile sağlanan manevra kabiliyeti, mobil robotlarda bu tekerlek türünün tercih edilme sebeplerindendir. Endüstriyel, eğitim, sağlık, savunma alanlarında bu tekerlek modelinin kullanıldığı görülmektedir. Endüstriyel sektörde kullanılmakta olan otonom mobil robotlar bir ürünün, nesnenin veya parçanın bir noktadan başka bir noktaya taşınması amacıyla kullanılmaktadır. Yüksek yük taşıma kapasitesinden ve farklı amaçlar için de kullanılabilir yapısından dolayı endüstriyel sektör içerisinde forklift araçlarında, KUKA gibi endüstriyel robot üreticisi firmaların mobil robotlarında, NASA'nın zor koşullarda keşif için ürettiği mobil robotlarda çok yönlü hareket sağlayan tekerlek grubunda bulunan mekanum tekerleklerin kullanıldığı görülebilmektedir. Şekil 1.4'te KUKA firmasının geliştirdiği üzerinden robot kolun da bulunduğu mobil robot gösterilmiştir.



Şekil 1.4 KUKA mobil robot [22]

Mekanum tekerleklere alternatif olarak benzer çalışma prensibinde çalışan ve yanal hareket kabiliyeti sayesinde diğer tekerleklere göre avantajları bulunan omni tekerlekler de endüstriyel sektörde kullanılabilmektedir. Mekanum tekerleklerde silindirik yapılar 45 derecelik açılarla yerleştirilmişken, omni tekerleklerde 90 derecelik açılarla yerleştirilmiş yapılar bulunmaktadır. Mekanum tekerleklerden

farklı olarak bağlantı konfigürasyonunun değiştirilmesiyle 3 tekerlekli yapıya sahip mobil robotlarda kullanılabilmektedir. Mobil robotun savrulmasına yol açtığı için omni tekerlekler genellikle kullanılmamaktadır [23].

Mobil robotlar kendi arasında sahip oldukları özelliklere, kullandıkları teknolojilere ve kullanım amaçlarına göre 2 kategoride incelenir.

1. AMR (Autonomous Mobile Robot)
2. AGV (Autonomous Ground Vehicle)

Tablo 1.5'te AMR ve AGV arasındaki farklar belirtilmiştir.

Tablo 1.5 AMR ve AGR arasındaki farklar [24]–[26]

AMR	**AGV**
Otonom yer araçlarının gelişmiş formlarıdır.	Platforma eklenen robot kollar ile karışık yapıda çalışan mobil robot haline gelebilirler.
Çalışma alanı önceden tahmin edilmeyen alanlarda çalışabilirler.	Herhangi bir karar mekanizması olmadığından mobil robotlarda tanımlanan görevin açık bir şekilde anlatılması gerekir.
Yol planlaması yapabilirler.	Çalışma alanı ve bölgesi önceden tanımlanmıştır.
Çevresindeki varlıklarla etkileşim içinde olarak haritalama gerçekleştirebilirler.	Sürekli olarak tekrarlanabilir olan görevleri gerçekleştirirler.
Yazılımları daha karışık ve ekipman sayısı daha fazladır.	Son zamanlarda endüstride fazla kullanılmamaktadır.

Bu tezdeki mobil robotun düşük bütçeli ve göreve yönelik olmasının yanında çalışma alanı koşulları da dikkate alınarak kara temaslı ve 4 adet mekanum tekerleğine sahip otonom mobil robot tasarımı gerçekleştirilmiştir.

1.2.5 Geçmişte Yapılmış Olan Çalışmalar

Mekanum tekerlek, 1973 yılında "Mecanum Company" sahibi İsveçli mühendis olan Ilon tarafından icat edilmiştir. Genellikle kullanılmakta olan ackerman [27] ve diferansiyel sürüş yöntemleri [28] yerine çok yönlü bir hareket imkanı ortaya

sunmaktadır. Mekanum tekerlek kullanılarak yapılan ilk mobil robot, 1987 yılında Uranus ismiyle gerçekleştirilmiştir.



Şekil 1.5 Uranus çok yönlü mobil robot [29]

Diegel, Badve, Bright, Potgieter ve Tlale tarafından çok yönlü mekanum tekerlere sahip bir mobil robotun, anlık yön değiştirme ve konfigürasyon yöntemlerinin değerlendirilmesi için MEGAN-1 [30] geliştirilmiştir. Mekanum tekerlerin sahip olduğu çevresel silindirik yapının hareket verimliliğini arttırdığı, dar alanlı fabrika ortamlarında hareket açısından avantaj ve taşınım konusunda kolaylık sağladığı vurgulanmıştır.



Şekil 1.6 MEGAN-1 mobil robotu [30]

Koestler ve Bräunl [31] EyeSim simülasyon sistemini kullanarak oluşturdukları mobil robotun test edilmesi, kullanılan elektronik ekipmanlardan kaynaklı gerçekçi hataların modellenmesi ve bu hata modellerinin değerlendirilebilmesi üzerine çalışma gerçekleştirilmiştir.

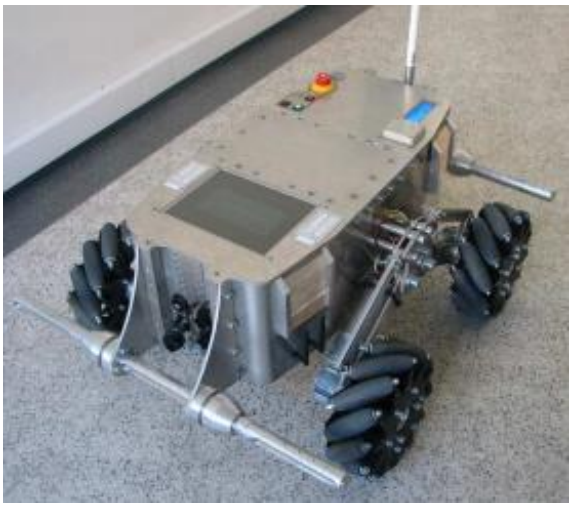
Ismail, Ramli, Ahmad ve Marhaban'in çalışmalarında [32] endüstri uygulamalarında kullanılan pahalı sensörlerin yerine düşük maliyetli bir kamera kullanarak çizgi izleyen mobil robot uygulaması geliştirilmiştir. Mobil robotun alması gereken bilgi, görüntü ayrıştırma metoduyla sağlamıştır. Koyu bir zemin üzerinde beyaz çizgiyle belirtilmiş ve önceden planlanmış yol güzergâhını başarılı bir şekilde tamamladığı görülmüştür.

Künemund ve Roehrig'ın çalışmalarında [33] odak noktası endüstriyel uygulamalar olacak şekilde yol planlayıcı sistemlerinin daha hızlı ve doğru çalışması için kübik eğri segmentleri ve beşli eğri segmentleri kullanmaya yönelik çalışmalar gerçekleştirilmiştir. Önceden yörünge, hız ve ivme bilgilerine sahip olunan robotun kübik eğrilerin kullanılmasıyla bir eğrilik yörüngesi elde edilmiş ve elde edilen verilerle beşli eğriler hesaplanmıştır. Sonuçların endüstriyel uygulamalarda kullanılabilirlik açısından olumlu olduğu gösterilmiştir.



Şekil 1.7 Küpik ve beşli eğri segment hesabı için kullanılan mobil robot [33]

Gulan ve Ilkiv [34] tarafından iç ve dış mekanlar için uygun olan, 90 kilogram ağırlığında bulunan ve ek olarak 90 kilogram daha yük taşıyabilen, alüminyum malzemeden yapılmış bir mobil robot geliştirilmiştir. Bu çalışmada mekanum tekerlek kullanılarak tezde yapılması planlanan yük taşıma görevine yönelik bir yapı oluşturulmuştur. Bir kontrol sistemi yerine, büyük miktarda ve küçük alt sistemler üzerinden kontrole yönelik çalışma yapılmıştır.

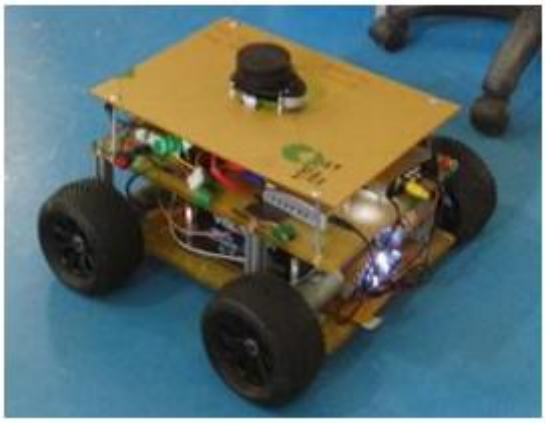


Şekil 1.8 90 kilogram yük taşıyabilen mobil robot [34]

Zohaib, Pasha, Riaz, Javaid, Ilahi, Khan'in çalışmaları [35] engelden kaçınma için kullanılabilecek algoritmaların karşılaştırılması üzerine gerçekleştirilmiştir. Algoritmalar kombine edilerek verimli bir algoritma oluşturulmuştur. Boşluk takip etme metodu [36], zaman ve bellek tüketimi açısından başarılıdır; fakat U ve H tipi engellerde başarısız sonuçlar vermektedir.

Bilinmeyen çalışma alanında, diferansiyel sürüş özelliğine sahip ve PID kullanılan bir mobil robotun engelden kaçınma ve yol planlama algoritmasına yönelik Crenganis ve Bologa [37] tarafından çalışma gerçekleştirilmiştir.

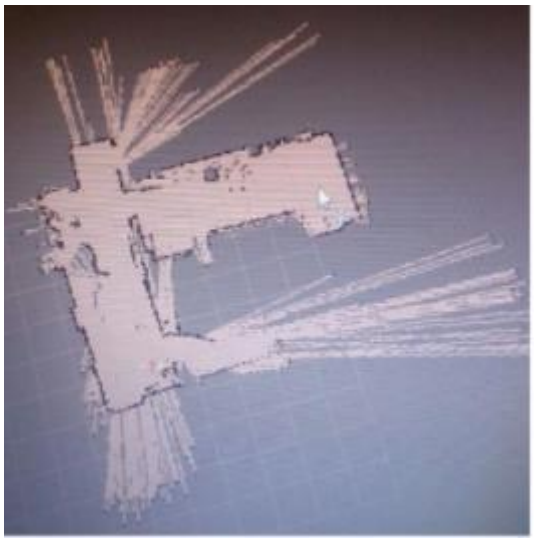
Markom, Adom, Tan, Shukor ve Rahim'in çalışmalarında [38] otonom sürüş sırasında mobil robotun konumlandırılma ve haritalandırılma çalışması gerçekleştirilmiştir. Bu kapsamda bu tezde de kullanılmış olan düşük maliyetli ışıkla uzaklık algılayıcı kullanılmıştır. Robota yerleştirilen sensör, üretici firmanın kendi programının çalıştırılmasıyla 3 farklı parametrede sonuç vermektedir. Elde edilen sonuçların filtreleme yapılmasıyla birlikte Matlab programı üzerinde çalışma alanı haritalanmıştır.



Şekil 1.9 Markom'un ışıkla uzaklık algılayıcıya sahip mobil robotu [38]

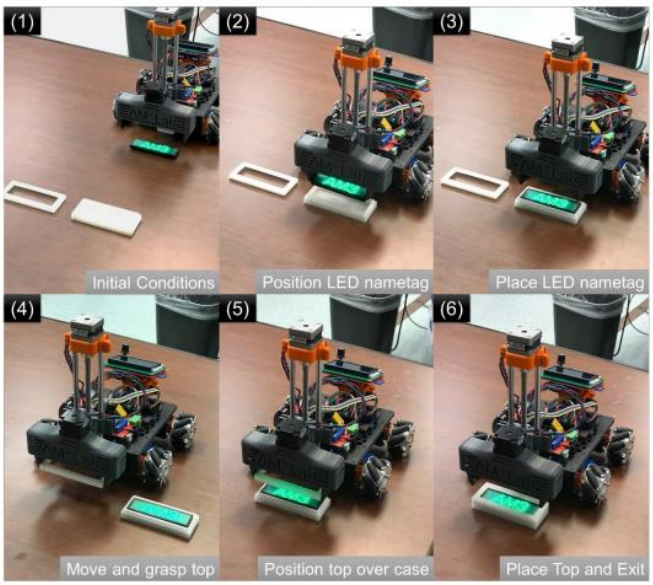
Gümüş, Topaloğlu ve Özçelik'in [39] ulaşımın hızlanması ve yolcu sayısı yoğunluğunun azalmasına yönelik sembolik çizgi izleme çalışması bulunmaktadır. Bilgisayar etkileşimli çalışan otonom robotun prototip denemelerinde konum tespiti ve robot bilgisi alımında başarılı olduğu gözlemlenmiştir. Test sonuçlarında sürücülere göre verilen tepkilerin hızlı oldu görülmüştür.

Köseoğlu, Çelik ve Pektaş'ın çalışmalarında [25] oluşturdukları mobil robot platformuna ışıkla mesafe ölçer yerleştirilmesiyle haritalama gerçekleştirilmiştir. Haritalama sırasında araç hızından kaynaklı ve sayaç tabanlı oluşan hataların giderilmesi yönelik filtreleme, algoritma ve kontrol çalışmaları yapılmıştır.



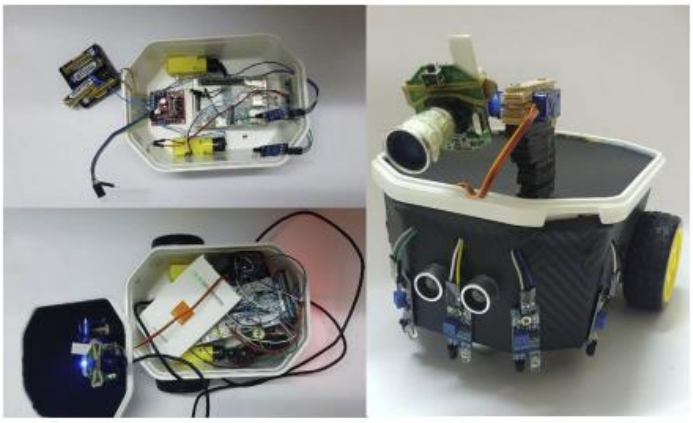
Şekil 1.10 Mobil robot slam uygulaması [25]

Steck, Ortega, Currence ve Zhou'in çalışmalarında [40] mobil robotun ön bölümünde kullanılan tutucu mekanizma sayesinde 3 boyutlu baskı olarak alınan ürünlerin montajı gerçekleştirilmiştir.



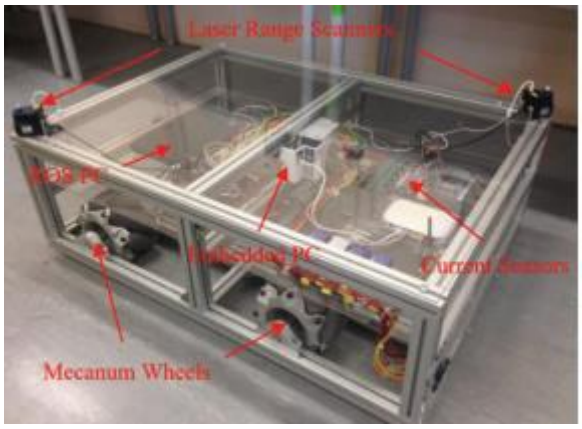
Şekil 1.11 Kavrayıcılı mobil robot [40]

Man, Koonjul ve Nagowah'ın çalışmasında [41] düşük bütçeli mobil robot tasarlanmıştır. Bu mobil robot birçok sensörler ile donatılmıştır. Engellerden kaçabilen, QR kodlarıyla yol planı yapabilen ve algoritmalarla hedefe en yakın yolu bulan mobil robot otonom hareket etmektedir. Elde edilen sonuçlarda araca yön sensörünün eklenmesi, pil tipinin değiştirilmesi ve ek bir servo motor ihtiyaç gerektiği ön görülmüştür.



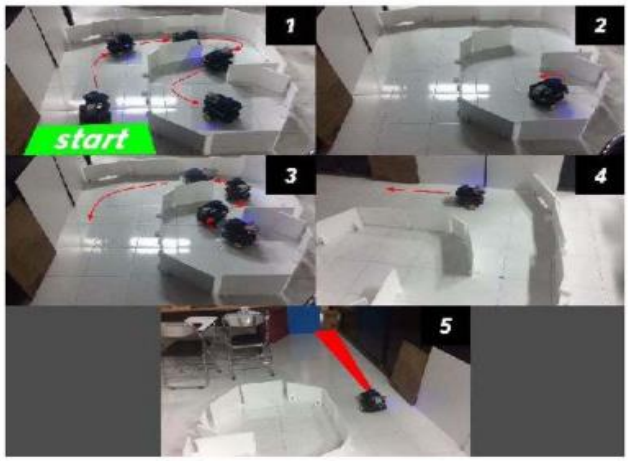
Şekil 1.12 Nagowah'ın mobil robotu [41]

Xie, Henkel, Xu ve Stol'un çalışmalarında [42] kullanımı artmaktan olan mekanum tekerleklerin dinamik yapısını incelenmiştir. Robotun birden fazla hareketinin gözlemlenebileceği bir harita oluşturulmuştur. Tekerlerin birbirinden bağımsız olarak her bir harekette harcadıkları enerji değerleri not edilmiştir. Enerji verimliliğinin geliştirilmesine yönelik testler ve çalışmalar yapılmıştır.



Şekil 1.13 Xi'nin çalışmasındaki mobil robot [42]

Hutabat, Purwanto, Rivai ve Hutomo'nun çalışmalarında [43] üzerinde ışıkla uzaklık algılayıcı bulunan bir mobil robot platformunun navigasyon sırasında karşısına çıkabilecek engellerden kaçınma durumları değerlendirilmiştir. Işıkla mesafe ölçerden alınan ölçümlerde 0.12 metre – 10.5 metre arasında ortalama olarak %9 oranında hatalı mesafe ölçümleri tespit edilmiştir. Farklı renklere sahip ve malzeme yapıları birbirinden farklı engeller üzerinde testler yapılmıştır. Yapılan testlerde renk değişiminin ölçümlerdeki hata payında çok büyük bir fark oluşturmadığı; fakat şeffaf görünüme sahip malzemelerde başarısız olduğu görülmüştür.



Şekil 1.14 Rivai'nin mobil robotunun engel tespit çalışması [43]

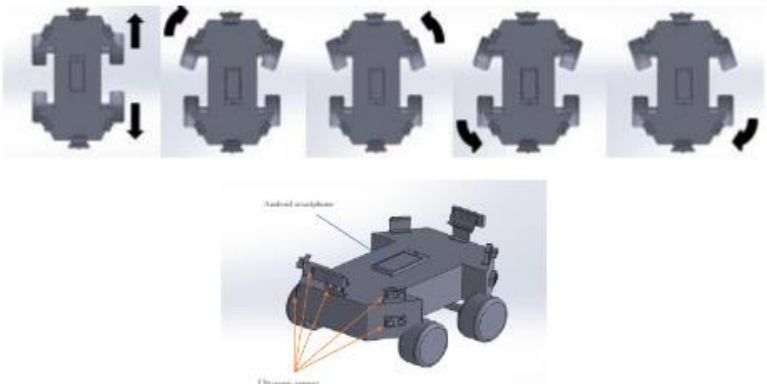
Fernandez, Vijayakumar, Palanisamy, Selvakumar, Karthikeyan, Selvabharathi, Vidyasagar, Kalyanasundhram [44] tarafından insansız ve otonom hareket sağlayan mobil robot platformu oluşturulmuştur. Robotun hareketleri kamera açı kontrolünün yapıldığı GUI üzerinden takip edilmiştir. Kullanılan sensörler ve

yüksek altyapı ile çevre algılamasının başarılı olduğu; fakat tahmin edilemeyen bir durumda otomatik kararlar aldığı ortaya çıkmıştır.



Şekil 1.15 Palanisamy'nin mobil robotunun gui arayüzü [44]

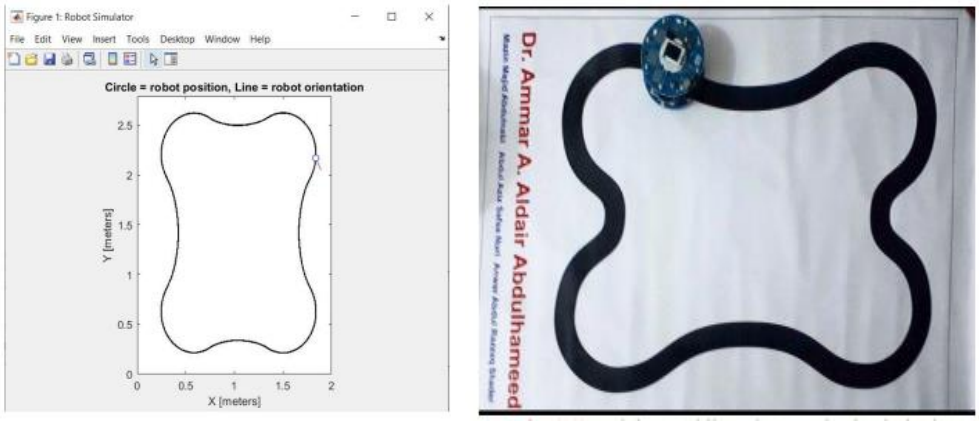
Hartono, Nizar, Robani ve Jatmiko'ın çalışmalarında [45] ultrasonik sensörler ve bluetooth kullanarak mobil robot platformu geliştirilmiştir. Bluetooth haberleşmesi Android cihaz üzerinden belirlenen koordinatların mikrodenetleyiciye iletilmesiyle robot belirlenen rotaya göre hareketini gerçekleştirmektedir.



Şekil 1.16 Hartono'nun mobil robot platform [45]

Alwan, Green, Noori ve Aldair' in çalışmalarında [46] mobil robotların belirlenen yolu en kısa sürede ve doğru şekilde ilerlemesindeki sorunlar hem teorik olarak hem de deneysel olarak incelenmiştir. Kızılötesi sensörlere sahip mobil robot belirlenen güzergâh üzerinden toplanan verilerle ve motorların hızının değiştirilmesiyle kontrol edildiği görülmüştür. Simulink üzerinden PID kontrol kullanılarak yapılan çalışmada yolun kompleksli yapıda olmasına rağmen robotun

hareketinin kararlı olduğu görülmüştür. Mobil robotun deneysel ve simülasyon performanslarının aynı olduğu gözlemlenmiştir.



a) b)

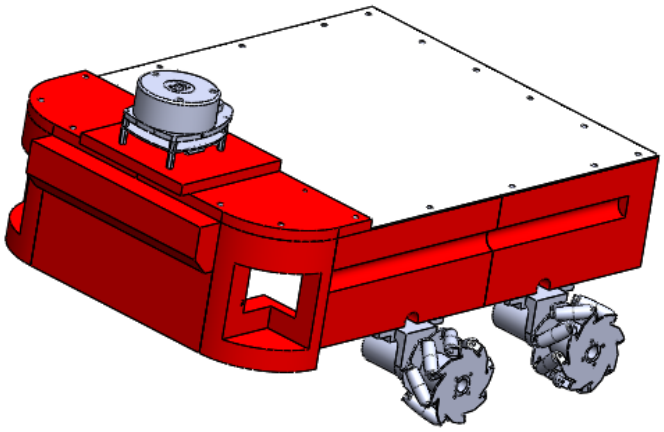
Şekil 1.17 a) Aldair'in simulasyon testleri, b) Aldair'in deneysel testleri [46]

2
ROBOTUN TASARIMI

Bu bölümde mobil robotun 3 boyutlu tasarımı, kullanılan malzemeleri ve elektronik ekipmanları hakkında bilgiler verilmiştir.

2.1 Robotun Tasarımı

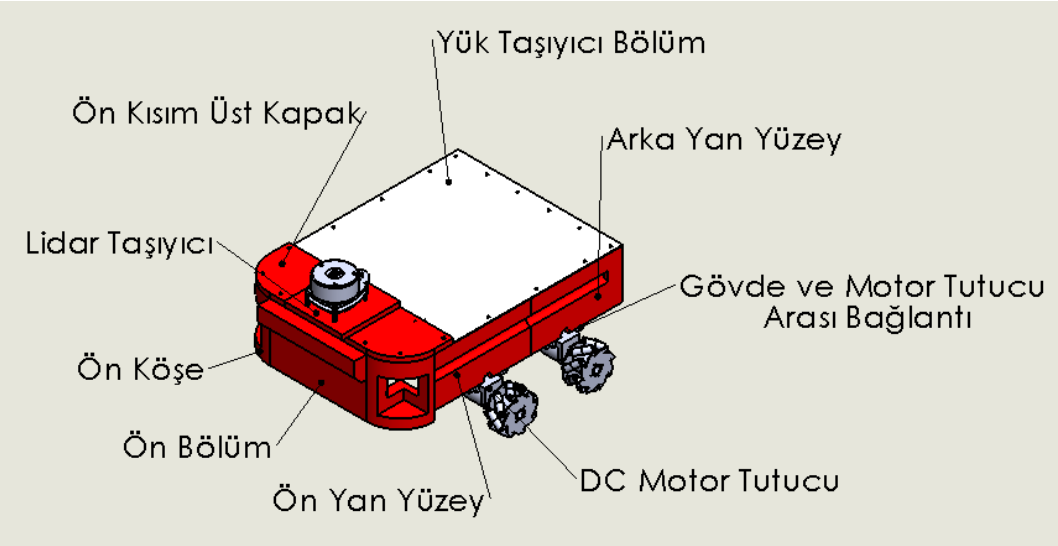
Robotun 3D tasarımı Solidworks programı kullanılarak yapılmıştır. 235x235x280 mm boyutlarına sahip 3D yazıcı ile basıma uygun olabilmesi için uygun ölçülerde çizimler gerçekleştirilmiştir. Şekil 2.1'de robotun genel olarak tasarımı gösterilmiştir.



Şekil 2.1 Tasarlanan mobil robot

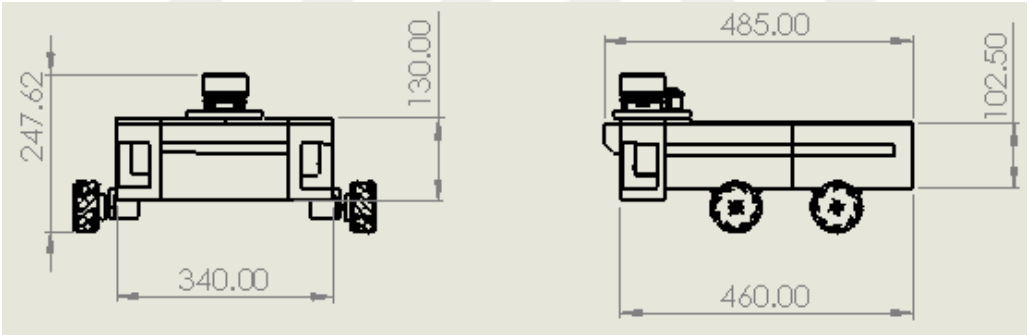
Şekil 2.2 de gösterildiği gibi robot toplamda 9 adet parçadan oluşmaktadır.

1. Ön bölümdeki orta kısım
2. Ön bölümdeki radyuslu köşeler
3. Yan yüzeyler
4. Arka yan yüzeyler
5. Yük taşıyıcı bölüm
6. Ön kısım üst kapak
7. Lidar taşıyıcı
8. Motor tutucu
9. Motor tutucu sabitleyici



Şekil 2.2 Robotun parçaları

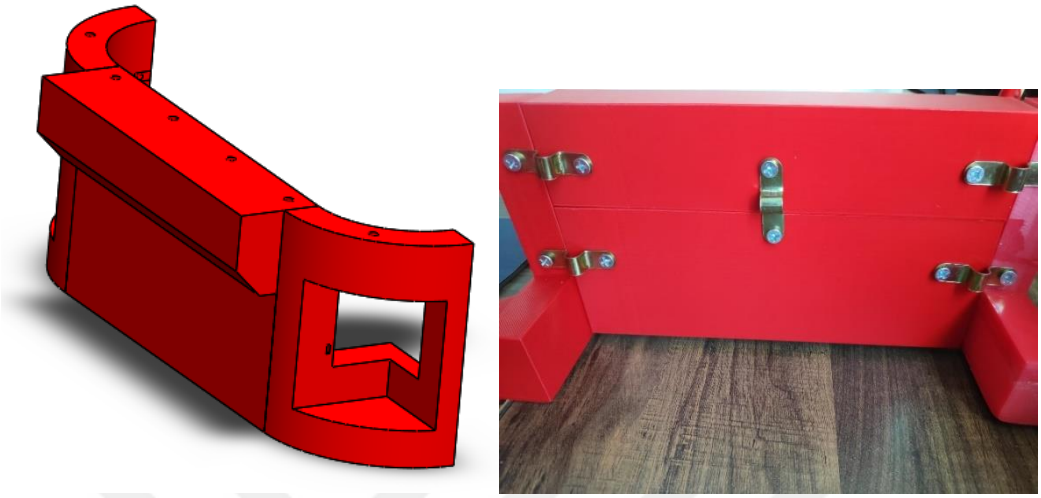
Tasarlanan robotun şasesi Şekil 2.3'te teknik çizimlerde gösterildiği gibi 340x485x130 mm boyutlarındadır. Yük taşıyıcı bölümünde 340x350x1.5 mm boyutlarında alüminyum levha ve şase yapımında PLA filament olmak üzere 2 farklı malzeme kullanılmıştır.



Şekil 2.3 Robotun 2D ölçüleri

Üretim işlemi için Solidworks üzerinden yapılan çizimler kaydedilip, Cura yardımcı programında gerekli olan baskı ayarlarının yapılmasıyla parçalar elde edilmiştir. 0.4 mm nozul kullanılarak dolgu yoğunluğu parçalara göre değişmekle birlikte %30, %40, %60 ve %100 olarak seçilmiştir.

Ön bölüm toplam 4 farklı parça olarak basılıp klipslerle birleştirme işlemleri yapılmıştır. Sabitleme aparatları kullanılarak farklı parçalar birbirlerine Şekil 2.4'te olduğu gibi bağlanmıştır.



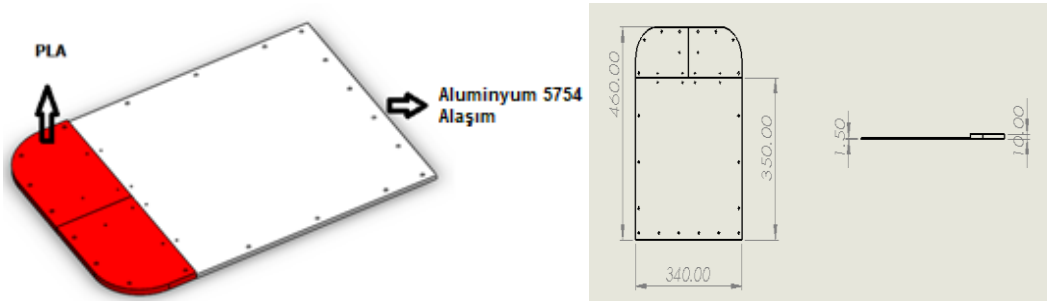
Şekil 2.4 Ön bölümün 3D görünümü ve birleştirilmiş hali

Aynı şekilde klipslerle yan yüzeyler de birleştirilmiştir. Yapıyı daha dayanıklı kılmak amacıyla alt bölümlerden de klipsler kullanılarak olası bir dönme işlemi engellenmiştir. Oluşturulan temel şasi yapısı Şekil 2.5'de gösterilmiştir.



Şekil 2.5 Gövdenin üretilmiş ve birleştirilmiş hali

Üretilen şasinin üzerine yük taşıyıcı kısmın eklenmesi gerekmektedir. Bu kısmın ön ve arka olarak iki parçaya bölünmüştür. Yükün sadece arka kısma koyulması hedeflenmiştir. Ön bölümde desteklemek ve kapatmak için yine PLA baskısı ile üretilmiş parça kullanılmıştır. Şekil 2.6'da gösterilen yük taşıyıcı kısım için 340x350x1.5 mm boyutlarında alaşım alüminyum levha kullanılmıştır.



Şekil 2.6 Yük taşıyıcı bölümün 3D ve 2D görünümü

2.2 Malzeme

Mobil robotun büyük bir bölümü biyolojik olarak parçalanabilen, termoplastik yapıya sahip olan PLA filament kullanılarak oluşturulmuştur [47]. Toplamda 9 önemli kısımdan oluşan robotun yük taşıma bölgesi dışındaki bütün bölgeler PLA malzemeden basılmıştır. Yük taşıma bölgesi için ise yüklenecek olan ağırlığın hesaba katılmasından ve gerekli olan parça boyutunun büyük olmasından dolayı alüminyum kullanılmıştır. Kullanılan alüminyum 5754 alaşım malzemenin özellikleri Tablo 2.1'de ve yerli üretim olan Porima PLA malzemelerinin özellikleri Şekil 2.7'de gösterilmiştir.

Tablo 2.1 Alüminyum 5754 alaşımının fiziksel ve mekanik özellikleri [48]

Özellik	**Değer**
Yoğunluk	$\mathbf{2.66\ g/cm^3}$
Elastisite Modülü	$\mathbf{68\ GPA}$
Zorlama Gerilimi	$\mathbf{160\ MPa}$
Çekme Mukavemeti	$\mathbf{240 - 280\ MPa}$

Özellikler *(Properties)*	**Test Metodu** (Test Methods)	**Değer, Birim** (Value, Unit)
Özkütle *(Density)*	ISO 1183	1.23 g
Eriyik Akış İndeksi *(Melt Flow Index)*	ISO 1133	17,3 g/10 dk
Erime Sıcaklığı *(Melting Temperature)*	-	190-220 C°
Isıda Eğilme Sıcaklığı *(Heat Deflection Temperature)*	ASTM D648	55 C°
Çekme Mukavemeti *(Tensile Strength)*	ISO 527	54,3 MPa
Elastik Modül *(Young Modulus)*	ISO 527	2300 MPa
Kopma Uzaması *(Elongation at break)*	ISO 527	3,4 %
Çentikli Darbe Testi *(Charpy Impact Strength-Notched)*	ISO 179	14,2 k J/m^2

Şekil 2.7 Porima PLA filamentinin malzeme özellikleri [49]

PLA malzemenin baskısı için Two Trees Bluer 3D yazıcı kullanılmıştır. Baskı boyutları 230x230x280 mm olmasından dolayı parçaların boyutları uygun şekilde ayarlanmıştır. Basım sırasında makinenin sıcaklık değerleri, baskı hızı ve kullanılan doluluk oranı verileri Tablo 2.2'de gösterilmiştir.

Tablo 2.2 3D yazıcı baskı ayarları

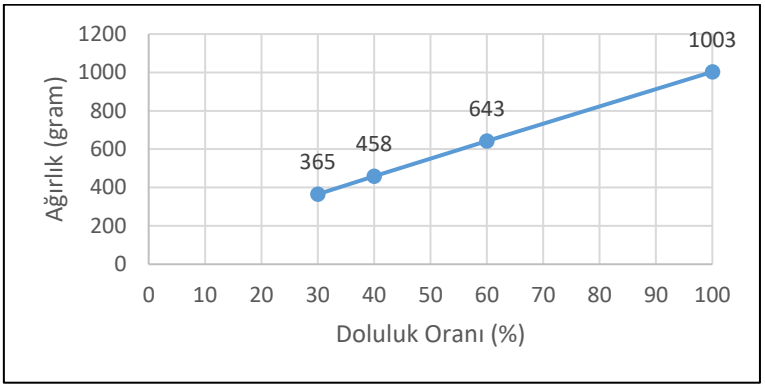
Yazıcı	Two Trees Bluer
Nozul Sıcaklığı	220 ℃
Yatak Sıcaklığı	70 ℃
Baskı Hızı	50 mm/s
Doluluk Oranı	%30-%40-%60-%100

Tablo 2.3'te arka köşe parçasının sadece doluluk oranı parametresinin değiştirilmesiyle oluşan baskı süreleri ve ağırlık değerleri verilmiştir.

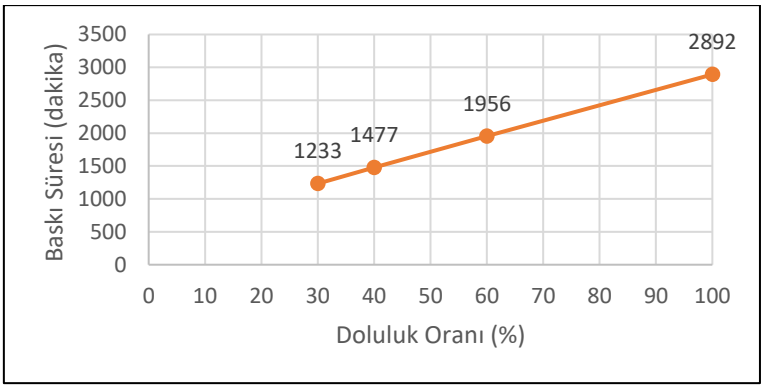
Tablo 2.3 Belirlenen bir parçanın doluluk oranlarına karşılık baskı süresi ve ağırlık değerleri

Doluluk Oranı	Baskı Süresi (dakika)	Ağırlık (gram)
%30	1233 dakika	365 gram
%40	1477 dakika	458 gram
%60	1956 dakika	643 gram
%100	2892 dakika	1003 gram

Bu değerlerin daha net bir şekilde karşılaştırılmasının yapılması için grafikle gösterimiyle birlikte bu parça için %30 doluluk oranı kullanılmıştır. %30 ve %100 baskılarda süre bakımından yaklaşık olarak %135 değerinde artış, ağırlık bazında yaklaşık olarak %175 oranında bir artış ön görülmüştür. %30 doluluk oranı ve %40 doluluk oranındaki parçalar arasında karşılaştırma yapıldığında ağırlık olarak %25 oranında bir artış, süre bakımından yaklaşık olarak %20 oranında bir artış olacağı ön görülmüştür. Şekil 2.8'de doluluk oranı değişiminde parçanın ağırlığında olan değişimi ve Şekil 2.9'da doluluk oranı değişiminde baskı süresinde ağırlık değişimi grafik olarak gösterilmiştir.



Şekil 2.8 Doluluk oranı değişiminde parçadaki ağırlık değişimi

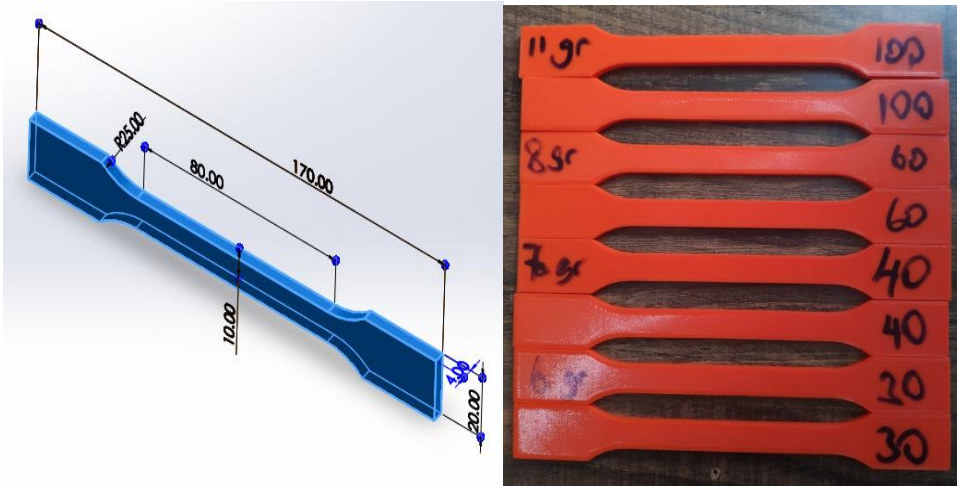


Şekil 2.9 Doluluk oranı değişiminde baskı süresinde ağırlık değişimi

3D yazıcı ile baskı sırasında malzeme özelliğini değiştiren etkenler vardır. Bunlar temelde şu şekilde sıralanabilir.

1. Doluluk oranı
2. Sıcaklık değerleri
3. Baskı hızı
4. Baskı modeli

Baskı süresi ve ağırlık değerlerinin yanı sıra malzeme doluluğu da malzemenin yapısını etkilemektedir. Bunun incelenmesi için robotta kullanılan %30, %40, %60 ve %100 doluluk oranlarına sahip ISO-527-2-1996 [50] standartlarında çekme numuneleri hazırlanmıştır. Şekil 2.10'da 3 boyutlu yazıcı kullanılarak elde edilen çekme numuneleri ve kullanılan standart boyutları gösterilmiştir.



a) b)

Şekil 2.10 a) ISO-527 numune ölçüleri, b) Numunelerin 3D baskıları

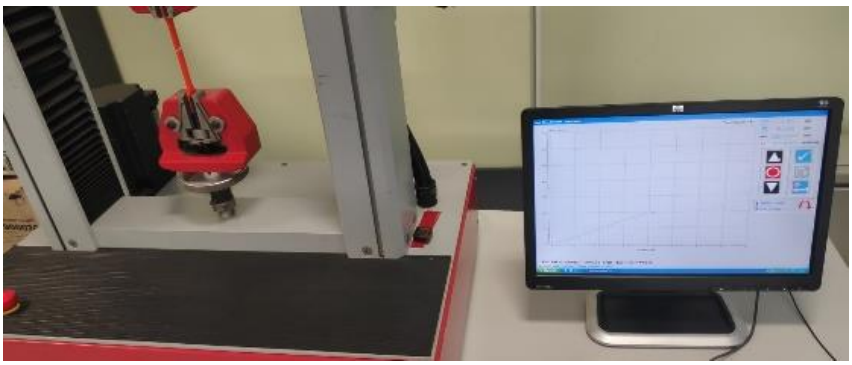
Baskı anında nozul bölümünde sıcaklık yardımıyla eriyen plastik malzemenin eridiği sıcaklık değeri de parçanın malzeme özelliğini etkilemektedir. Doluluk oranı kadar etkili olmasa da her malzemenin uygun sıcaklıkta daha iyi olduğu yapı vardır. Bu özellik filament malzemesinin içeriğinde bulunan karışımlardan dolayı kaynaklanabilmektedir. Modelin oluşturulma sırasında filament akışı belirli bir dolgu şekli ile hareket eder. Her bir dolgu şeklinin harcadığı plastik miktarı da farklıdır. Dolgu şeklinin seçimi konusunda bir diğer önemli husus ise açık sistemlerde basım sırasındaki yapışma sorunudur. Bu sebeple çizgi modeli ile baskı alınması tercih edilmiştir. Numunelerin eksenel çekme işleminin gerçekleştirilip sonuçlarının alınabilmesi için Şekil 2.11'de gösterilmiş olan Testometric cihazı kullanılmıştır.



a)



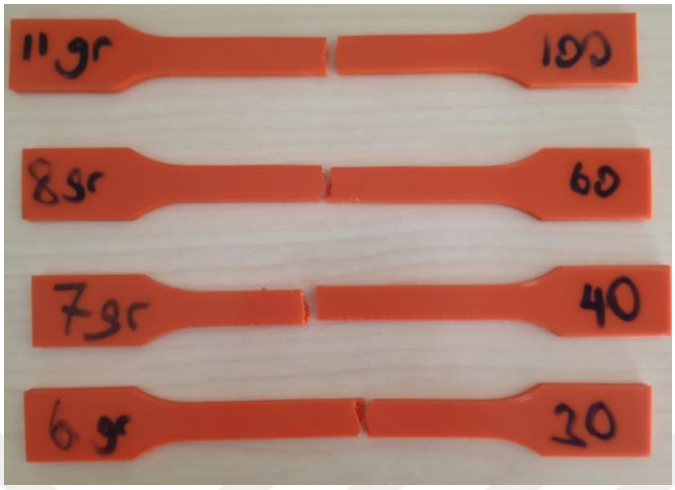
b)



c)

Şekil 2.11 a) Çekme testi için kullanılan teçhizat, b) Numenin kopması, c) Bilgisayar üzerinden sonuçların gözlemlenmesi

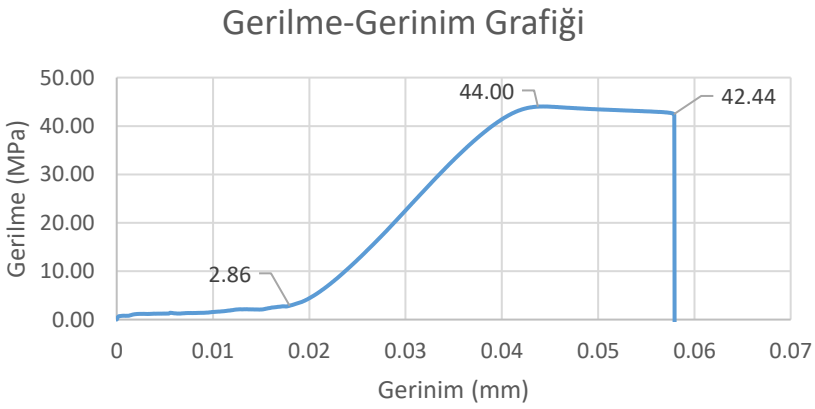
Şekil 2.12'de sırasıyla %100, %60, %40 ve %30 doluluk oranlarına sahip numunelerinin çekme testlerinin sonucunda meydana gelen kopma sonrası görünümleri gösterilmiştir.



Şekil 2.12 Kopma sonrası tüm numunelerin görünümleri

Bilgisayar üzerinden test sonuçlarının alınmasıyla birlikte her bir numune için gerilim-gerinim grafikleri oluşturulmuştur.

%100 doluluk oranına sahip yaklaşık olarak 11 gram ağırlığında olan numunede çekme testi gerçekleştirilmiştir. Kuvvet ve alan sonuçlarının kullanılmasıyla birlikte Excel programı üzerinden gerilme değerlerine ulaşılmıştır. Elde edilen maksimum gerilme değerinin 44 MPa, bu noktadaki gerinim değerinin 0.0438 mm, kopma öncesi gerilme değerinin ise 42.44 MPa olduğu görülmüştür.

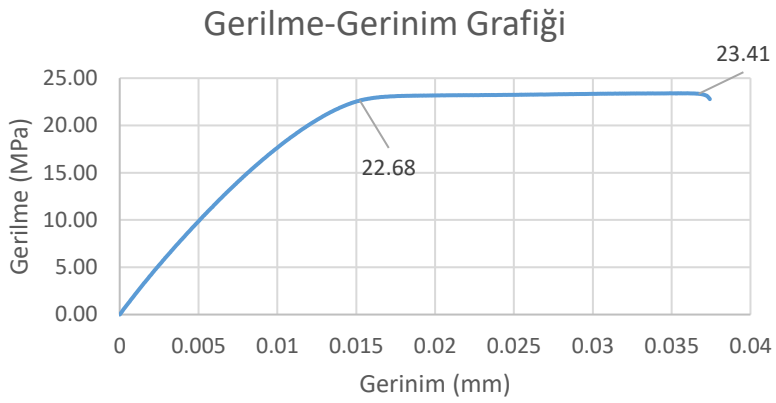


Şekil 2.13 %100 dolu numunenin gerilme-gerinim grafiği

Doluluk oranın azalması durumunda malzemenin yapısında gerçekleşmiş olan değişimle birlikte gerilme değerlerinde azalma görülmesi beklenmektedir. Ağırlığı

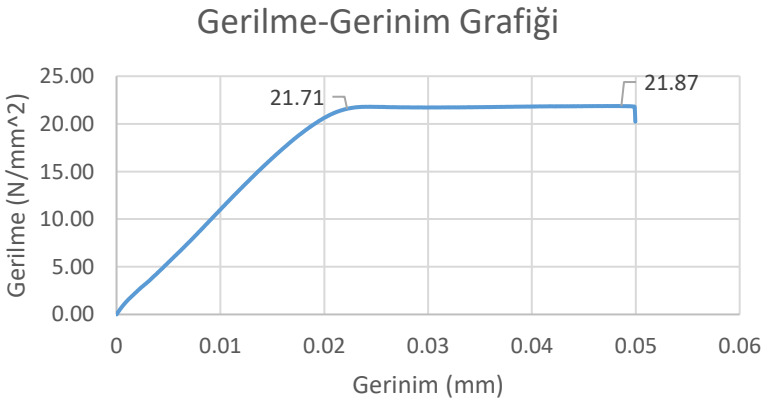
8 gram olan ve %60 doluluk oranına sahip numunenin deneysel sonuçlara göre çizilen gerilim-gerinim grafiği Şekil 2.14'te gösterilmiştir.

Kuvvet değerinin maksimum olduğu noktada gerilme değerinin maksimum olduğu görülmüştür. 0.036 mm gerinim değerinde ulaşılan gerilme değeri 23.41 MPa olarak görülmüştür.



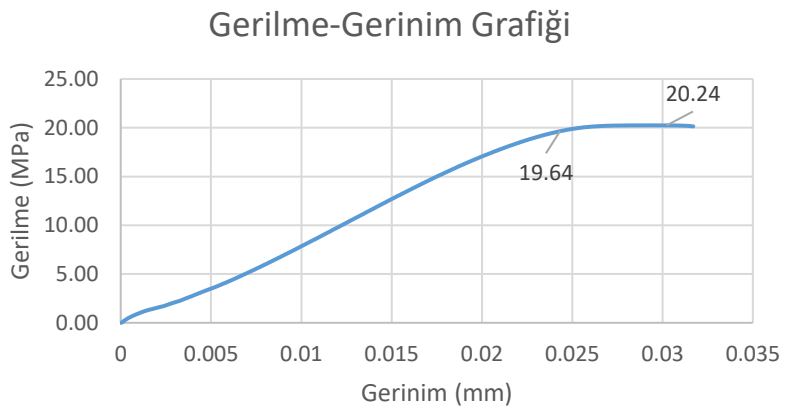
Şekil 2.14 %60 dolu numunenin gerilme-gerinim grafiği

Ağırlığı 7 gram olan ve %40 doluluk oranına sahip numunenin deneysel sonuçları göre gerçekleştirilen gerilim-gerinim grafiği Şekil 2.15'te gösterilmiştir. Büyük kuvvet değerlerinin oluştuğu noktalardaki gerilme değeri incelendiğinde 21.71-21.87 MPa sonuçları ortaya çıkmaktadır. Maksimum gerilme değerinin ise 21.87 MPa olduğu ve ardından numunenin koptuğu görülmüştür.



Şekil 2.15 %40 dolu numunenin gerilme-gerinim grafiği

Ağırlığı 6 gram olan %30 doluluğa sahip numunenin deneysel sonuçlarının ise %40 doluluğa sahip numune ile benzer değerlerde olduğu görüldü. Şekil 2.16'da görüldüğü gibi gerilme-gerinim grafiği incelendiğinde maksimum gerilme değerinin 20.24 MPa olduğu görülmüştür.



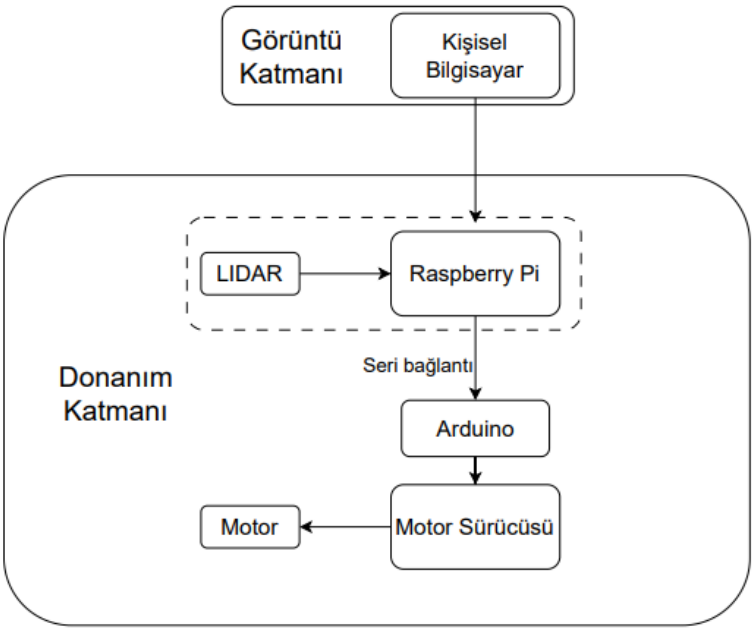
Şekil 2.16 %30 dolu numunenin gerilme-gerinim grafiği

Tüm sonuçlar incelendiğinde doluluk oranının azalması sonucunda uygulanabilir kuvvet, gerilme ve gerinim değerlerinin azaldığı, doluluk oranı değerleri birbirine yakın olan numune sonuçlarının da birbirine yakın olduğu görülmüştür. Baskı süresi, ağırlık ve gerilme değerlerinin göz önünde bulundurulmasıyla parçaların doluluk oranları belirlenmiştir.

2.3 Elektronik Malzemeler

Bu bölümde mobil robotta kullanılan elektronik malzemelerin özellikleri hakkında bilgi verilmiştir. Şekil 2.17'de mobil robotun donanımsal yapısı gösterilmiştir. Bu çalışmada kullanılan elektronik malzemeler 5 temel başlıkta incelenmiştir.

1. Kartlar
2. Motorlar
3. Motor sürücüleri
4. Batarya
5. LIDAR



Şekil 2.17 Mobil robotun donanımsal yapısı

2.3.1 Kartlar

Bu çalışmada toplam 2 farklı kart kullanılmıştır.

1. Arduino MEGA 2560 R3
2. Raspberry Pi 4 Model B

Motor ve motor sürücülerinin kontrollerinin sağlanması için harici bir kart ihtiyacına duyulmuştur. Raspberry Pi cihazının üzerinde bulunan USB portu kullanılarak Arduino beslenir. Arduino pinleri ile motor sürücüsü arasında bağlantı kurulur. Ardından pinlere gönderilen mesajların motor sürücülerine gönderilmesi birlikte motorların tahrik edilmesi sağlanmaktadır.

RPlidar A1 M8 sensörünün kullanılabilmesi için USB port girişine sahip Raspberry Pi 4 Model B kartı kullanılmıştır. Raspberry Pi'nin çalışması için gerekli olan güç, taşınabilir batarya sayesinde gerçekleştirilmiştir. Kullanılan Raspberry Pi 4 kartının özellikleri Tablo 2.4'te belirtilmiştir.

Tablo 2.4 Raspberry Pi 4 Model B özellikleri [51]

Güç Kaynağı	5 VDC
Akım	3 A
Hafıza	Micro hafıza kartı
İşlemci	Broadcom 2711 Dört çekirdekli Cortex-A72
RAM	4 GB
USB	2 x USB 2.0 + 2 x USB 3.0
Wi-Fi	Gigabit Ethernet
Pin Sayısı	40

2.3.2 Motor

Mobil robotun hareketinin sağlanması için tahrik edici sistem olarak DC motorlar kullanılmıştır. İhtiyaç duyulan mekanik özelliklere sahip DC motorun seçimi sırasında dikkat edilen unsurlar şunlardır:

1. Çalışma gerilimi
2. Tekerlek ile uyumlu olabilmesi için mil şaftının çapı,
3. Tork değeri
4. Maksimum çekilecek akım miktarı

DC motorun beslemesinin doğru değerde yapılabilmesi için en uygun ve verimli olduğu voltaj değerinin bilinmesi gerekmektedir. Mobil robotun hareketini sağlayan tekerleklerin tahrik edilmesi sırasında motorlarla olan şaft bağlantının doğru bir şekilde gerçekleştirilmesi gerekir. 12V ile çalışmakta olan bu DC motorların motor sürücüleri tarafından beslenmesinin ardından, ihtiyaç duyulan tork değerine ulaşabilmesi için uygun akımın sağlanması gerekmektedir. Bu durumda motor sürücülerin seçimi de önem arz etmektedir. Seçilen DC motor çap boyutu yaklaşık olarak 37 milimetre, uzunluğu ise 57 milimetredir. Tekerlek

bağlantısını sağlayan şaftın çapı ise 6 milimetredir. Tablo 2.5'te kullanılan DC motorun özellikleri belirtilmiştir.

Tablo 2.5 Kullanılan motorun özellikleri [52]

Yüksüz hızı (12V)	76 RPM
Yüksüz çektiği akım (12V)	150 miliamper
Zorlanma akımı (12V)	5.5 Amper
Zorlanma torku (12V)	45 kg.cm
Yüksüz hızı (6V)	38 RPM
Zorlanma akımı (6V)	3 Amper
Zorlanma torku (6V)	28 kg.cm

2.3.3 Motor Sürücüsü

Kişisel bilgisayar üzerinden verilen komutlar Raspberry Pi cihazına aktarılır. Raspberry Pi ve Arduino arasındaki seri bağlantı sayesinde bilgiler Arduino cihazına taşınır. Ardından komutlara uygun sinyaller motor sürücülerine iletir. Motor sürücülerinden motorlara sinyaller gönderilerek istenilen hareketlenmelerin gerçekleştirmesi sağlanır. Kişisel bilgisayardan gönderilen komutların dışında bu hareketlere dış sensörler de müdahale edebilmektedir. Kullanılan her bir motor sürücüsünde 2 kanal bulunmaktadır. Bu hem 2 farklı motoru tek sürücü ile yönetebilmeyi, hem de beslenecek olan motor sürücüsü sayısının azaltılmasını sağlamaktadır.

2.3.4 Taşınabilir Batarya

Arduino cihazı Raspberry Pi cihazının USB portundan, Raspberry Pi cihazı ise 5V ve 2A ile taşınabilir bataryadan beslenmektedir. Ek olarak kullanılan 2 adet 5V çıkış gerilimine sahip taşınabilir güç bataryaları motor sürücülerini beslemektedir. Bu bataryalar sayesinde motorların çalışma süresi arttırılmış ve motorlardan gerek duyulan tork ve hız değerleri karşılanmıştır.

Uygun voltaj, düşük ağırlık, düşük maliyet ve uzun batarya süresi faktörleri dikkate alınarak seçim yapılmıştır. Kullanılan güç bataryalarının kapasiteleri sayesinde yaklaşık olarak 10 saatlik bir sürüş süresi sağlamaktadır.

2.3.5 LIDAR

Raspberry Pi cihazının slotuna yerleştirilmiş durumda olan hazır kartın içerisine daha önceden bilgisayar üzerinden yüklenmiş olan ve Linux işletim sistemini kullanan Ubuntu 20.04 kurulmuştur. Işıkla uzaklık algılayıcının çalışması için Linux sistemine RPlidar yazılımları ve ROS paketlerinin yüklenmesi gerekmektedir. Yazılımların yüklenmesinin ardından sensörün USB kablosunun Raspberry Pi cihazının uygun portuna takılarak bağlantı sağlanmıştır.

Mobil robot hareket sırasında etrafında birçok engelle karşılaşabilir. Haritalama işleminin yapılabilmesi için bu engellerin tespit edilebilmesi gerekmektedir. 360 derece açıyla engelleri algılayabilen, gelebilecek tehditlerin farkına varılmasını sağlayan bu sensör sayesinde çalışma ortamının haritası oluşturulur ve robotun konumu harita üzerinden gözlemlenebilir.

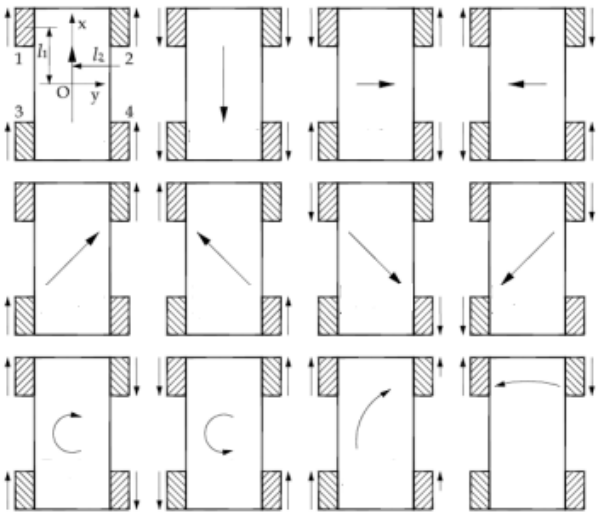
Bu çalışmada mesafe ölçüm aralığının kapalı alanda çalışmalar için uygun olması, Raspberry Pi ile besleyerek ile çalışabilir durumda olması, kolay kullanıma sahip olması, maliyetin düşük olması ve haritalamanın kolay bir şekilde yapılabilmesi sebeplerinden dolayı RPlidar A1 M8 kullanılmıştır. Bu algılayıcının özellikleri Tablo 2.6'da gösterilmiştir.

Tablo 2.6 RPlidar A1 M8 özellikleri [53]

Mesafe Ölçüm Aralığı	0-6 metre
Açısal Ölçüm Aralığı	0-360 derece
Mesafe Çözünürlüğü	<0.5 mm
Lazer Dalga Boyu	775-795 nm
Örnek Üretimi	2000 örnek/sn
Ağırlık	190 gram

2.4 Mobil Robotun Hareket Konfigürasyonları

Mobil robotlardan kullanılan çeşitli tekerlekler bulunmaktadır. Tekerlek yapısına ve sayısına göre de mobil robotun kinematik hareketleri değişiklik göstermektedir. Bu çalışmada kullanılan mekanum tekerleklerde belirli açılarda yerleştirilmiş silindirler bulunmaktadır. Bu silindirik yapılar tekerin dönme yönündeki kuvvetin bir kısmını tekerin normal kuvvetine dönüştürür. Kullanılan tüm tekerleklerdeki silindirlerin bu kuvveti dönüştürmesi sonucunda bir kombinasyon ortaya çıkmaktadır. İstenilen yönde bir kuvvet vektörü meydana getirilir ve robotun tekerleklerinin yönünde herhangi bir değişiklik gerçekleştirmeden kuvvet vektörü doğrultusunda serbest bir hareket oluşur [23]. Bu kuvvetlerin de etkisiyle 4 adet mekanum tekerleklere sahip olan bir mobil robotun gerçekleştirebileceği hareketlerin kombinasyonları Şekil 2.18'de gösterilmiştir.



Şekil 2.18 Mecanum tekerleklere sahip robotun hareket konfigürasyonları [54]

Mekanum sürüş yöntemini gerçekleştirebilen tekerleklerin tercih edilme sebepleri [23], [55] şunlardır:

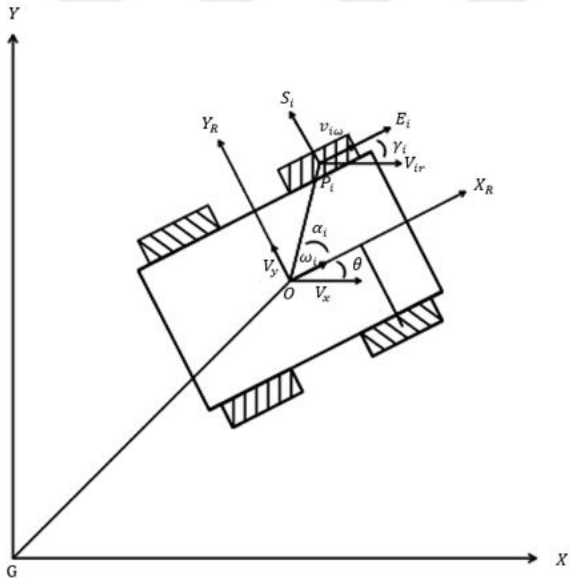
1. Çalışma prensibi bakımından farklı bir tasarıma sahiptir.
2. Açılı silindirik parçaların temas yüzeyleriyle olan aktarımı sebebiyle yük kapasitesi fazladır.
3. Kayma hareketiyle dar alanlarda kullanım kolaylığına sahiptir.

Sahip olduğu olumlu yönlerin dışında dezavantaj getiren durumlar da [56] bulunmaktadır.

1. Kayma hareketini gerçekleştirebilen tekerlekler, eğilimli arazilerde sorun yaratabilmektedir. Robotun bir sonraki pozisyonunun tahmin edilmesi zordur.
2. Silindirik yapıların da dönebiliyor olması kinetik enerji kullanımında verimsizlik oluşturur.
3. Kompleks bir yapıya sahip tasarımı bulunmaktadır.

2.5 Mobil Robotun Tekerlek Konfigürasyonları

Mobil robotların sahip oldukları tekerlek yapısı hareketleri doğrudan etkilemektedir. Şekil 2.19'da dört adet mekanum tekere sahip bir mobil robotun konfigürasyonları verilmiştir.



Şekil 2.19 Mecanum tekerlek konfigürasyonu

Konfigürasyonda belirtilen parametrelerin karşılıkları şu şekildedir:

X, G, Y eylemsizlik referans çerçevesi,

X, Y, θ robotun konumu ve açısı,

X_R, Y_R, θ robotun ayak çerçevesi,

P_i, S_i, E_i P merkezli tekerleklerin koordinat sistemi,

I_{ix}, I_{iy} ön ve arka tekerler arasındaki mesafelerin yarısı,

α_i OP_i ve X_R arasındaki açı,

γ_i E_i ve V_{ir} arasındaki açı,

ω_i tekerlerin açısal hızı,

$v_{i\omega}$ tekerlek dönüşleri sonucu oluşan hız vektörü,

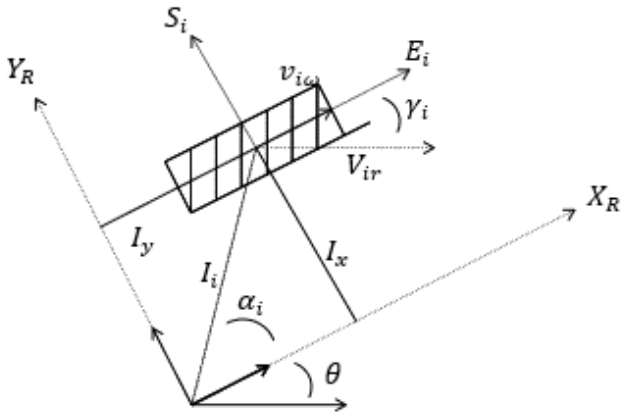
V_{ir} silindirik tekerleklerin hızı,

V_x, V_y robotun lineer hızı

Mobil robot basit bir şekilde düşünülecek olursa konumunu belirleyen 3 temel bileşen vardır. Pozisyon bilgisini x , y ve θ oluşturmaktadır.

$$q = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{2.1}$$

Şekil 2.19'da gösterilen robotun sol ön tekerine ait pozisyon, hız ve açı bilgileri Şekil 2.20'de detaylı olarak gösterilmiştir.



Şekil 2.20 Sol ön teker konfigürasyonu

Şekil 2.20'ye göre hız denklemleri aşağıdaki formlarda elde edilebilir.

$$V_{ir} = \frac{1}{\cos \gamma_i} v_{i\omega} \quad i = 0,1,2,3 \tag{2.2}$$

$$v_{i\omega} = \omega_i r_i \tag{2.3}$$

$$V_{S_i} = V_{ir} \sin(\gamma_i) \sin(45°) \tag{2.4}$$

$$V_{E_i} = \omega_i r_i + V_{ir} \cos(\gamma_i) \tag{2.5}$$

$$\begin{bmatrix} V_{S_i} \\ V_{E_i} \end{bmatrix} = \begin{bmatrix} \sin(\gamma_i) & 0 \\ \cos(\gamma_i) & r_i \end{bmatrix} \begin{bmatrix} V_{ir} \sin(45°) \\ \omega_i \end{bmatrix} \tag{2.6}$$

$$^{w_i}T_{P_i} = \begin{bmatrix} \sin(\gamma_i) & 0 \\ \cos(\gamma_i) & r_i \end{bmatrix} \tag{2.7}$$

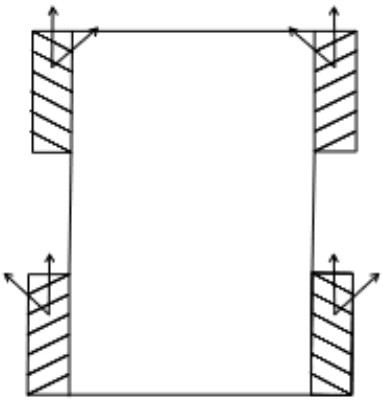
Tekerleğin robotun düzlemindeki hızı incelendiğinde (2.5) ve (2.7) eşitlikleri kullanılarak (2.8) eşitliliği elde edilir.

$$\begin{bmatrix} V_{i_{X_R}} \\ V_{E_{Y_R}} \end{bmatrix} = \begin{bmatrix} \cos(\beta_i) & -\sin(\beta_i) \\ \sin(\beta_i) & \cos(\beta_i) \end{bmatrix} \begin{bmatrix} V_{S_i} \\ V_{E_i} \end{bmatrix} \tag{2.8}$$

$$\begin{bmatrix} V_{i_{X_R}} \\ V_{E_{Y_R}} \end{bmatrix} = \begin{bmatrix} \cos(\beta_i) & -\sin(\beta_i) \\ \sin(\beta_i) & \cos(\beta_i) \end{bmatrix} \begin{bmatrix} \sin(\gamma_i) & 0 \\ \cos(\gamma_i) & r_i \end{bmatrix} \begin{bmatrix} V_{ir} \sin(45°) \\ \omega_i \end{bmatrix} \tag{2.9}$$

$$V_{S_i}\gamma_i = 0 \tag{2.10}$$

Mekanum tekerleklerde bulunan silindir 45 derecelik açılarla yerleştirilmiştir. Bu sebeple θ açısı 45 derecedir. Tekerleklerin ve silindirlerin hareket yönleri Şekil 2.21'de gösterilmiştir. Tekerlek numaralandırılmaları soldan sağa doğru yapılmıştır.



Şekil 2.21 Mobil robotun robot düzleminde konfigürasyonu

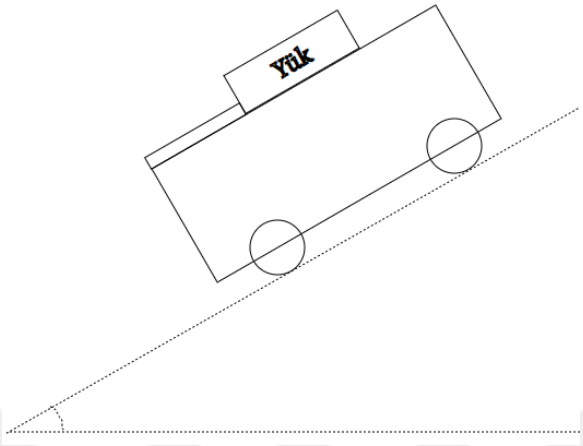
$$V_{x_0} = \frac{V_{r_0}}{\sqrt{2}}, \qquad V_{y_0} = V_{MW_0} + \frac{V_{r_0}}{\sqrt{2}} \tag{2.11}$$

$$V_{x_1} = -\frac{V_{r_1}}{\sqrt{2}}, \qquad V_{y_1} = V_{MW_1} + \frac{V_{r_1}}{\sqrt{2}} \tag{2.12}$$

$$V_{x_2} = -\frac{V_{r_2}}{\sqrt{2}}, \qquad V_{y_2} = V_{MW_2} + \frac{V_{r_2}}{\sqrt{2}} \tag{2.13}$$

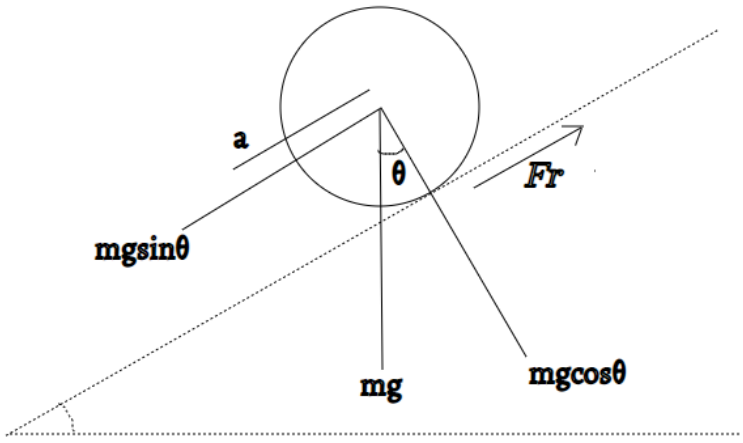
$$V_{x_3} = \frac{V_{r_3}}{\sqrt{2}}, \qquad V_{y_3} = V_{MW_3} + \frac{V_{r_3}}{\sqrt{2}} \tag{2.14}$$

Lineer hareket halinde zemin sürtünmesinden kaynaklı bir kuvvet ve sahip olduğu ivmeden kaynaklı başka bir kuvvet oluşmaktadır. Bunların dışında ürün taşıma amacıyla oluşturulan mobil robot platformunun üzerine 3 kilogramlık yük etki etmektedir.



Şekil 2.22 Mobil robotun eğik yüzeyde durma pozisyonu

Şekil 2.22'de mobil robotun belirli bir açıyla birlikte eğik düzlem üzerindeki durma pozisyonu gösterilmiştir. Robotun hareketi sırasında motorlardan sağlanması beklenen 2 tork değeri bulunmaktadır. Bunlarından birisi yuvarlanma hareketinden, diğeri ise lineer hareketten kaynaklanmaktadır. Şekil 2.23'te eğik düzlem üzerinde etki eden kuvvetler gösterilmiştir.



Şekil 2.23 Eğik düzlem üzerinde etki eden kuvvetler

$$\sum F_x = mgsin(\theta) - F_r = ma \tag{2.15}$$

$$N - mgcos(\theta) = 0 \tag{2.16}$$

$$\sum M_g = I\alpha \tag{2.17}$$

Yuvarlanma hareketinin oluşabilmesi için elde edilen maksimum kuvvetin, kayma kuvvetinden daha büyük olması gerekir.

$$T = I\alpha \tag{2.18}$$

$$F_r = \frac{1}{2}ma_m \tag{2.19}$$

$$F_{max} = ma_{max} \tag{2.20}$$

$$mgsin(\theta) - F_r = ma_{max} \tag{2.21}$$

$$mgsin(\theta) - \frac{1}{2}ma_{max} = ma_{max} \tag{2.22}$$

$$F_{max} = \frac{3}{2}ma_{max} \tag{2.23}$$

$$a_{max} = \frac{2F_r}{m} \tag{2.24}$$

$$a_{max} = 2gcos(\theta)\mu \tag{2.25}$$

$$mgcos(\theta)\mu = \frac{1}{2}ma \tag{2.26}$$

$$mgsin(\theta) - mgcos(\theta)\mu = ma \tag{2.27}$$

$$a = gsin(\theta) - \left(gcos(\theta)\frac{a}{2gcos(\theta)}\right) \tag{2.28}$$

$$a = \frac{2}{3}\ gsin(\theta) \tag{2.29}$$

$$M = \frac{1}{2}mgsin(\theta) \tag{2.30}$$

$$\sum M = \frac{1}{2}mar + mar \tag{2.31}$$

$$M = \frac{3}{2}mar \tag{2.32}$$

Motor seçiminde ihtiyaç duyulan tork değerinin hesaplanabilmesi için gerekli denklemlerin kullanılması sonucunda, elde edilen tork değeri göze alınarak uygun motor seçimi yapılmıştır.

3
YOL PLANLAMA YÖNTEMLERİ

Mobil robotlar için yol planlama, temelde çevre bilgisine dayalı yol planlama ve bütünlüğe dayalı yol planlama olmak üzere 2 ana başlıkta sınıflandırılmaktadır [57]. Bu bölümde literatürde bulunan bazı yol planlama algoritmalarının çalışma yöntemleri hakkında bilgi verilmiş; yöntemlerin avantajları ve dezavantajları arasında karşılaştırılma gerçekleştirilmiştir.

3.1 Çevre Bilgisine Dayalı Yol Planlama

Mobil robotun konumundaki değişiklik durumlarında insan ve insan dışı varlıkların etki durumu bulunmaktadır. Bir ortamda bulunan robotun çevre faktörlerine dair bilgisinin olması planlayacağı yolu değiştirebilmektedir. Çalışılacak olan ortamın biliniyor veya bilinmiyor olması ve ortamda bulunan engellerin statik veya dinamik olması oluşturulacak olan yol planlaması için birer parametrik değer taşımaktadır. Bu faktörler altında Tablo 3.1'de gösterildiği gibi temelde 6 farklı durumda birbirinden farklı planlama yöntemleri elde edilebilir.

Tablo 3.1 Yol planlama yöntemi oluşturulabilecek durumlar

Harita	Engel	Ortam	Durum Sayısı
Biliniyor	Var	Statik	2
		Dinamik	
	Yok	Statik	1
Bilinmiyor	Var	Statik	2
		Dinamik	
	Yok	Statik	1

Çalışılacak olan ortamın bilindiği; ortamda statik ve dinamik engellerin bulunduğu veya ortamda herhangi bir engelin olmadığı durumlarda 3 farklı senaryonun ortaya

çıkmaktadır. Çalışılacak olan ortamın bilinmediği; ortamdaki statik ve dinamik engellerin bulunduğu veya ortamda herhangi bir engelin olmadığı durumlarda 3 farklı senaryonun ortaya çıkmasıyla birlikte toplamda 6 farklı senaryo bulunmaktadır. Her biri birbirinden farklı ortama sahip senaryolar için farklı yol planlama algoritmaları kullanılabilmektedir.

Mobil robotun ortam ve engel bilgilerine sahip olduğu ve bu bilgileri kullanarak oluşturduğu yol planlaması küresel yol planlama olarak değerlendirilirken; robotun ortam ve engel konusunda bilgisinin olmadığı veya kısmen bilgi sahibi olduğu durumlarda yerel yol planlama olarak değerlendirilmektedir. Bu yol planlamalarının birbiriyle olan farklılıkları, avantaj ve dezavantajları Tablo 3.2 'de gösterilmiştir.

Tablo 3.2 Küresel ve yerel yol planlaması [57], [58]

Küresel Yol Planlama	**Yerel Yol Planlama**
Bilinen bir ortamda çalışmaktadır.	Bulunduğu ortamdaki sensör verilerini kullanarak çalışmaktadır.
Tüm harita üzerinde planlama oluşturur.	Anlık olarak belirli bir alanda yol planlaması oluşturur.
Engelle karşılaşması durumunda optimum yol planlaması sağlamaz.	Engelle karşılaşması durumunda yol planlaması anlık olarak değişir.
Tepkileri yavaştır.	Tepkileri hızlıdır.

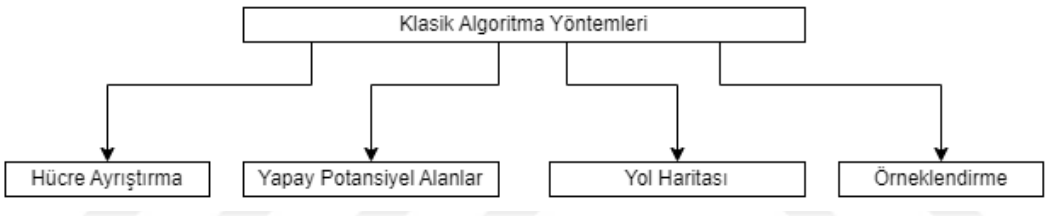
3.2 Bütünlüğe Dayalı Yol Planlama

Yol planlama yöntemlerinin oluşturulabileceği 6 farklı senaryo değerlendirildiğinde farklı temellere dayalı algoritma yöntemleri kullanılabilmektedir. Aynı senaryo için birden fazla yöntemle farklı sonuçlara ulaşılabilir ve sonuçların değerlendirilerek en uygun yöntemin seçilmesi gerekmektedir. Örneğin, haritanın bilindiği ve engelin bulunduğu statik bir harita için kullanılabilecek olan iki farklı algoritma yöntemi arasından optimum sonuçları veren algoritmanın kullanılması verimlilik açısından önemlidir. Temel anlamda yol planlama yöntemleri 4 farklı başlık altında değerlendirilebilir.

1. Klasik algoritma yöntemleri
2. Sezgisel algoritma yöntemleri
3. Yapay zekâ temelli algoritma yöntemleri
4. Biyolojik temelli algoritma yöntemleri

3.2.1 Klasik Algoritma Yöntemleri

Klasik algoritma yöntemleri mobil robotlarda kullanılan yol planlama algoritmalarından birisi olarak kategorize edilmektedir. Çalışma tekniği bakımından statik ortamlarda daha verimli çalışmasından dolayı dinamik bir ortamın oluşacağı durumlarda tercih edilmemektedir. Statik haritada uygun bir yolun olması durumunda mutlaka o yolu bulur; fakat işlem karmaşıklığından dolayı yol planlamasının ortaya çıkarılması uzun sürmektedir. Klasik algoritma yöntemleri Şekil 3.1'de olduğu gibi 4 farklı başlık altında kategorize edilebilmektedir.

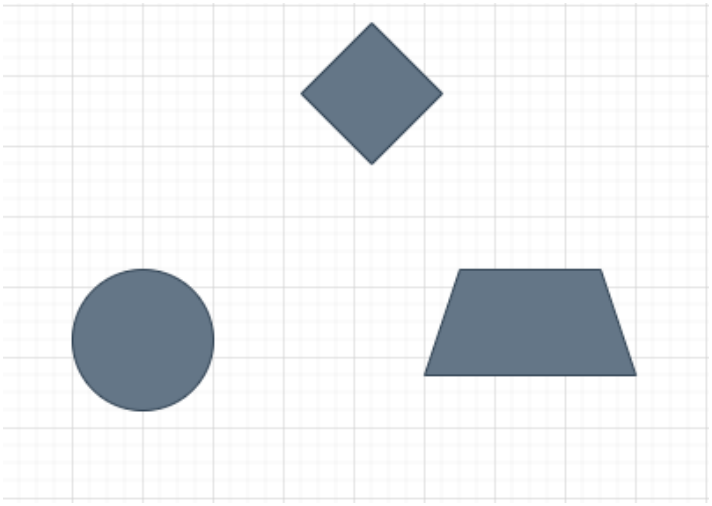


Şekil 3.1 Klasik algoritma yöntemleri

3.2.1.1 Hücre Ayrıştırma Yöntemi

Bu yöntemdeki amaç haritadaki engel bölgeleri küçük parçalara ayırıp yol güzergâhının bu noktalardan geçmesini sağlayarak hedefe ulaşmaktır. Şekil 3.2'de gösterilen bu algoritmanın detaylı olarak çalışma prensibi şu şekildedir [59]:

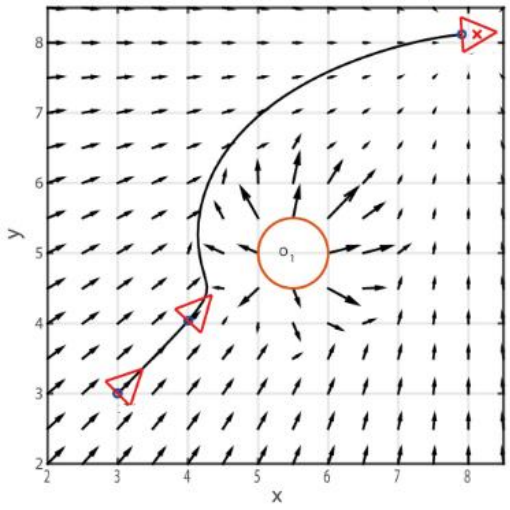
1. Statik harita üzerindeki bölgeler küçük hücrelere ayrılır.
2. Engel parçasının bulunmadığı hücreler tespit edilir.
3. Tespit edilen bu hücrelerin de kullanılmasıyla başlangıç noktasından bitiş noktasına doğru engelle temasın olunmadığı hücrelerden geçen bir yol planlaması elde edilir.



Şekil 3.2 Engellerin olduğu statik haritanın hücrelere bölünmesi

3.2.1.2 Yapay Potansiyel Alanlar Yöntemi

Bu yöntemde mobil robotun çalışma ortamı yapay potansiyel alan olarak düşünülür. Robotun gideceği noktaya ulaşması için başlangıç noktasından itibaren hedef noktaya doğru çekici vektörler oluşturulur. Mobil robotun engele yaklaşması durumunda ise itici vektörlerin oluşturulmasıyla mobil robotun bu noktalara yanaşması engellenmiş olur. Hedef noktanın engele yakın bölgelerde olması durumunda ise engelden gelen itici vektörlerin olması ve hedeften gelen çekici vektörlerin bulunmasından dolayı bu yöntem kullanıldığında sorunlar meydana gelmektedir [60]. Şekil 3.3'te yapay potansiyel algoritma yöntemiyle planlanan yol güzergâhı gösterilmiştir.

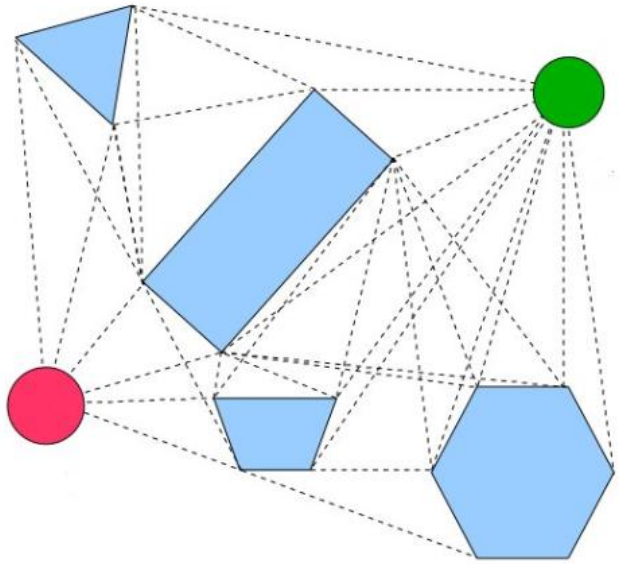


Şekil 3.3 Yapay potansiyel alanlar algoritma yöntemi[61]

3.2.1.3 Yol Haritası Yöntemi

Statik ortamda bulunan engellerin kullanılmasıyla birlikte başlangıç noktasından hedef noktaya küçük yollar oluşturulur. Bu yolların birleştirilmesiyle tam bir yol güzergâhı elde edilir. En çok kullanılan yol haritası yöntemleri görünürlük grafiği ve Voronoi grafiği yöntemleridir.

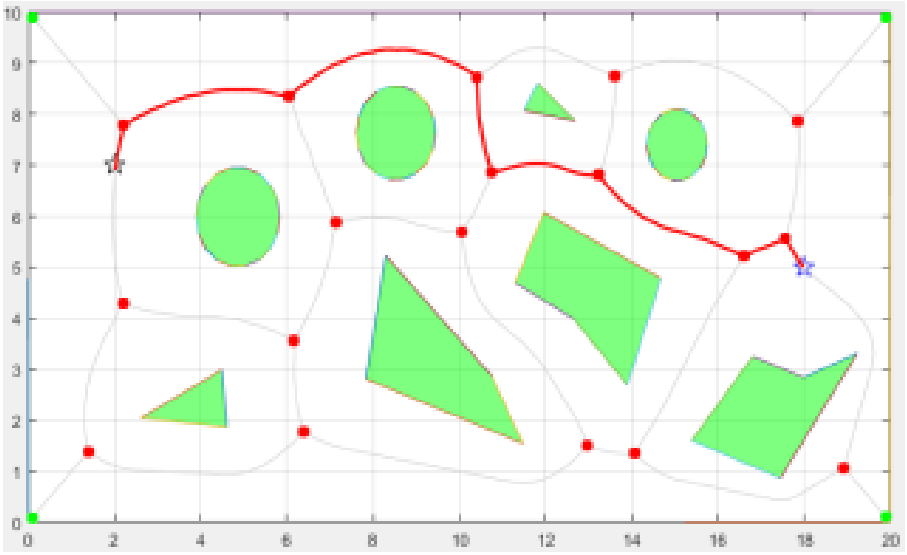
Görünürlük grafiği yönteminde engellerin köşegenlerine noktalar yerleştirilir. Başlangıç noktasından başlayarak gidilecek noktalar arasında olan engellerin noktaları birleştirilerek hedef noktaya ulaşan bir yol güzergâhı oluşturulur. Basit bir yol planlama tekniği olmasından dolayı hızlıdır; fakat engel üzerine yerleştirilen noktaların kullanılmasıyla oluşturulan bu yol mobil robot için kaza riski oluşturabilir. Şekil 3.4'te yeşil noktadan kırmızı noktaya görünürlük grafiği kullanılarak planlanabilen yollar gösterilmiştir.



Şekil 3.4 Görünürlük grafiği [62]

Voronoi yöntemiyle görünürlük grafiği yöntemi birbirleriyle çok benzerlik göstermektedir. Haritada bulunan engellerin kenarlarından belirli bir uzaklıkta olacak ve engelleri çevreleyecek şekilde çizgilerle birleştirme yapılır. Başlangıç ve bitiş noktaların bu çizgilerle birleştirilmesiyle yeni bir yol planlaması elde edilmiş olur. Bu yöntemde oluşturulan yolun engelle arasındaki güvenli mesafesinden dolayı görünürlük grafiğine göre kaza riski daha azdır.

Şekil 3.5'te başlangıç ve bitiş noktaları arasında Voronoi yönteminin kullanılmasıyla oluşturulan yol planlaması gösterilmiştir.



Şekil 3.5 Voronoi yöntemi [63]

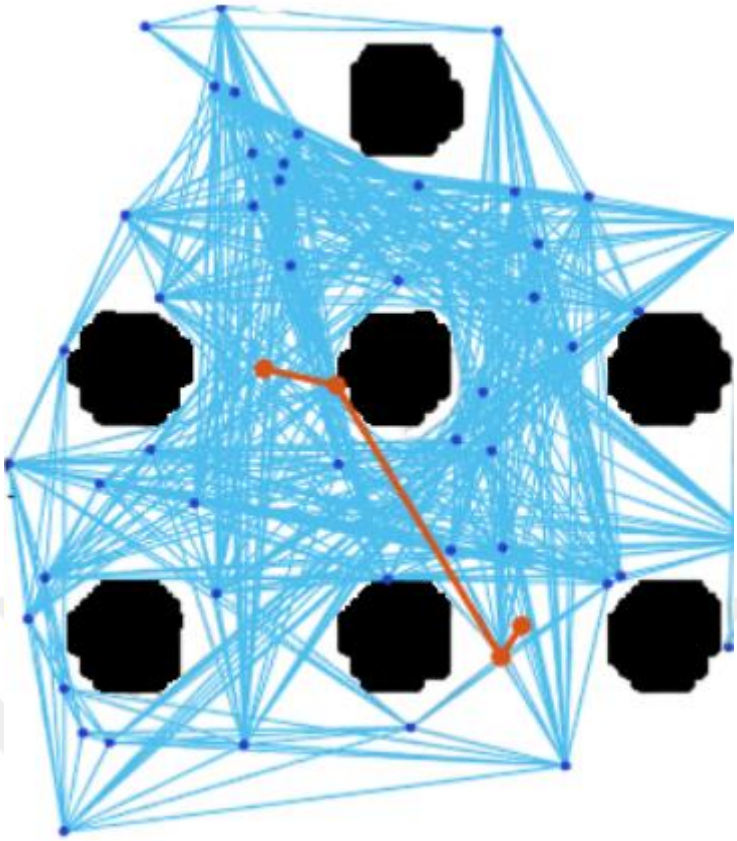
3.2.1.4 Örneklendirme Yöntemi

Örneklendirme yönteminde harita üzerinden belirli şartlar altında noktalar oluşturulur. Bu noktalar başlangıç ve bitiş noktalarıyla birleştirilerek uygun bir yol planlaması gerçekleştirilir. Bu çalışma mantığının en çok kullanıldığı yol planlama algoritmaları olasılıksal yol planlama ve hızlı keşfedilen gelişigüzel ağaçlar yol planlamasıdır.

Olasılıksal yol planlama algoritmasının çalışma prensibi şu şekildedir:

1. Harita üzerine yerleştirilecek olan rastgele düğüm nokta sayısı belirlenir.
2. Yerleştirilen düğüm noktalarının birbirleriyle yapılacak olan birleştirme işlemi için izin verilen maksimum mesafe miktarı belirlenir.
3. Düğüm noktalar arasında birbirleriyle olan mesafesi belirlenen değerden küçük olan noktalar arasında birleştirme işlemi gerçekleştirilir.
4. Başlangıç ve hedef noktası belirlenir.
5. Birleştirilmesi yapılmış düğüm noktaları ile başlangıç ve bitiş noktaları birleştirilerek yol planlaması gerçekleştirilmiş olur.

Bu çalışma prensibine göre örnek bir harita üzerinde olasılıksal yol planlama algoritmasının kullanımıyla elde edilen optimum yol Şekil 3.6'da gösterilmiştir.



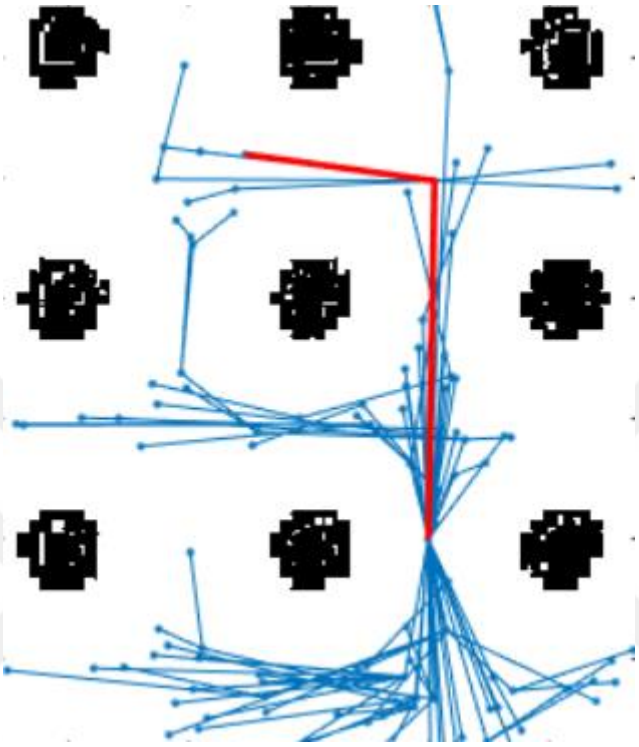
Şekil 3.6 Olasılıksal yol planlama algoritması

Bu planlama algoritması başlangıç ve bitiş noktaları arasında bir yol varsa bu yolu bulur; fakat haritaya yerleştirilen düğüm noktaları rastgele yapıldığı için her denemede farklı bir yol güzergâhı, farklı yol uzunluğu ve farklı yolculuk sürelerine ulaşılmasına sebebiyet vermektedir.

Hızlı keşfedilen gelişigüzel ağaçlar yol planlama algoritması da olasılıksal yol planlama algoritmasına benzer bir şekilde çalışmaktadır. Çalışma prensibi ise şu şekildedir:

1. Gerçekleştirilecek olan iterasyon sayısı belirlenir.
2. Başlangıç noktasına komşu rastgele düğüm noktası belirlenir ve bu nokta seçilir.
3. Ardından yeni mevcut konuma en yakın komşu düğüm noktası belirlenir ve yeni noktaya ilerlenir.
4. Bu işlem hedefe ulaşıncaya kadar devam eder.
5. Seçilen noktalar birleştirilerek başlangıç noktasından bitiş noktasına giden bir yol planlaması gerçekleştirilmiş olur.

Bu çalışma prensibine göre örnek bir harita üzerinde hızlı keşfedilen gelişigüzel ağaçlar yol planlama algoritmasının kullanımıyla elde edilen optimum yol Şekil 3.7'de gösterilmiştir.



Şekil 3.7 Hızlı keşfedilen gelişigüzel ağaçlar yol planlaması

3.2.2 Sezgisel Algoritma Yöntemleri

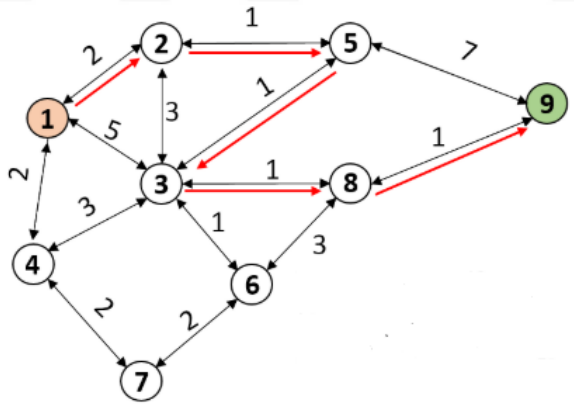
Sezgisel algoritmalar, matematiksel modelleme ve farklı algoritmalar gibi bilimsel verilerek dayandırılarak verimlilik yerine geçen sürenin önemine değinildiği algoritmalar olarak değerlendirilmektedir. Genellikle bu yöntem grubunda bulunan parçacık sürü optimizasyonu, A* ve Dijkstra algoritmaları kullanılmaktadır.

Parçacık sürü optimizasyonu, 1995 yılında Kennedy ve Eberhart tarafından sürü tabanlı olarak öne sürülmüş bir metottur [64]. Temel olarak kuş, balık, böcek gibi hayvanlar aleminde bulunan canlıların sosyal hayatlarında yiyecek bulma ve sürü içerisinde birlikte hareket etme gibi davranışlarından ortaya çıkarılmış bir algoritmadır [65].

Bu algoritmanın çalışma aşamaları şu şekildedir:

1. Harita üzerinde rastgele başlangıç pozisyonuna, hıza ve yöne sahip parçacıklar oluşturulur.
2. Optimum hareketin belirlenebilmesi için bir fonksiyon tanımlaması yapılır.
3. Parçacıkların yeni konumlarına bir önceki konum ve hız verilerinin kullanılmasıyla ulaşılır.
4. En iyi yaklaşımla mevcut partiküllerden en uygunu seçilir, yeni hız ve konum güncellemesi gerçekleştirilir.
5. Doğru noktaya ulaşıldığında durur, ulaşılamazsa aşamalar aynı şekilde tekrardan gerçekleştirilir.

A* ve Dijkstra algoritmaları birbirlerine benzemektedir. Aralarındaki en büyük fark ise, Dijkstra algoritmasında mevcut konum ve bir sonraki hedef konum arasında sadece maliyet analizi yapılırken; A* algoritmasında maliyet değerine ek olarak sezgisel maliyet değeri de eklenerek hesaplama gerçekleştirilir. Şekil 3.8'de 1 numaranın başlangıç noktası, 9 numaranın hedef noktası olduğu durumda Dijkstra algoritmasının kullanımıyla elde edilen yol planlaması gösterilmiştir.



Şekil 3.8 Dijkstra algoritması [66]

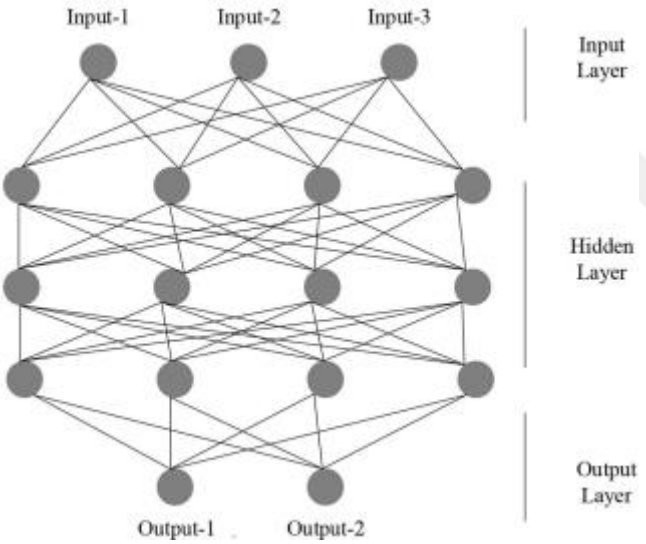
Bu yol planlaması sırasında yol maliyetleri en düşük olan yolların tercih edilmesiyle hedefe ulaşılmıştır. İlerlenen her bir düğümdeki değerlerin maliyete eklenmesi durumunda ise elde edilen algoritma A* algoritması olmaktadır.

3.2.3 Yapay Zekâ Temelli Algoritma Yöntemleri

Yol planlama algoritmalarının birçoğunun içeriğinde yapay zekâ kullanımlarını görmek mümkündür. Temelinde yapay zekâ bulunan ve yol planlama amacıyla en

çok kullanılan yapay sinir ağları ve benzetilmiş tavlama algoritmaları örnek verilebilir.

Yapay sinir ağları temelde giriş katmanı, gizli katman ve çıkış katmanı olmak üzere 3 temel bölümden oluşmaktadır [67]. Yol planlaması için harita yapay sinir ağları algoritmasına giriş olarak yerleştirilir ve çıktı olarak ise oluşturulmuş olan yolun elde edilmesi beklenmektedir. Başlangıç ve bitiş noktaları arasında birden fazla yol olacağı için çıktı olarak olasılık verilerine ulaşılabilmektedir. Yapay sinir ağları bazı durumlarda uygun yolu bulamayabilir veya olasılıksal olarak düşük değerlerle karşılaşmamıza sebep olabilir. Bu gibi durumlarda pekiştirmeli öğrenme metodu kullanılarak başlangıç ve bitiş noktaları arasında önceden oluşturulmuş yol güzergâhının tanıtılması ile elde edilecek sonuçların daha verimli olması mümkündür. Şekil 3.9'da yapay sinir ağlarının genel yapısı gösterilmiştir.

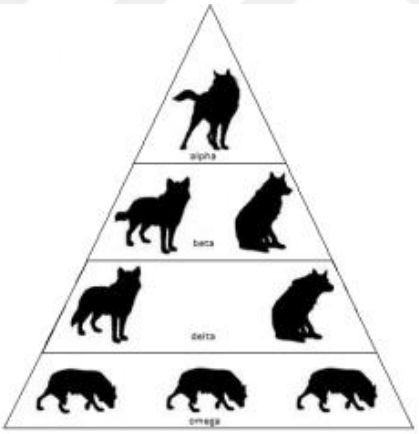


Şekil 3.9 Yapay sinir ağları yapısı [68]

Benzetilmiş tavlama algoritması genellikle enerji ve sıcaklık gibi fonksiyonların uygun seviyelere getirilmesi amacıyla kullanılan bir algoritma [69] olsa da, minimum enerji kullanımın olduğu yol planlama algoritması [70] olarak da kullanılmaktadır. Yolun uzunluğu, yolculuk süresi ve maliyeti gibi faktörlerin hesaba katılmasıyla başlangıç noktasından harekete başlayarak bitişe kadar geçen zaman içerisinde harcanan enerjinin minimum düzeyde olacağı en iyileştirme çalışması gerçekleştirilerek optimum yolun oluşturulmasına olanak sağlamaktadır.

3.2.4 Biyolojik Temelli Algoritma Yöntemleri

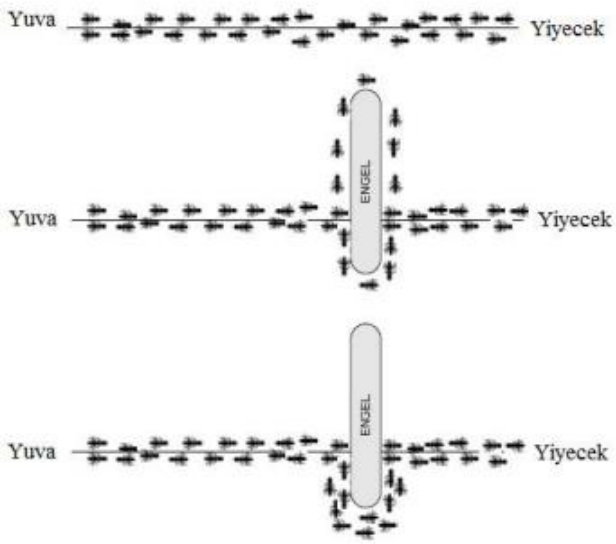
Biyolojik temelli algoritmalar doğal hayatın içerisinde yer alan durum ve davranışlardan veya canlıların hayvanlar alemindeki yaşam şekillerinden esinlenerek oluşturulmuş algoritmaları kapsamaktadır. En çok kullanılan algoritmalar gri kurtlar algoritması, karınca koloni algoritması ve genetik algoritmadır. Gri kurt algoritması doğada yaşamakta olan gri kurtların avlanma, aranma ve saldırma hareketlerinin fonksiyonel olarak ifade edilmesiyle en iyi konum hesabı gerçekleştirilir [71]. Şekil 3.10'da gösterildiği gibi alfa, beta, delta ve omega olmak üzere 4 farklı gri kurt grubu bulunmaktadır. Başlangıçta rastgele kurt sürüsü ataması gerçekleştirilir. Ardından her bir kurt için değer hesaplaması gerçekleştirilir. Bu kurtlar arasında değerlere göre yapılan sıralamada en iyi değere sahip gri kurtlar sırasıyla alfa, beta ve delta olacak şekilde sıralanır. Omega olarak tanımlanan ve grubun geriye kalanını temsil eden gri kurtlar için avlanma denklemleri kullanılarak konum hesaplamaları gerçekleştirilir. Parametre ve fonksiyon değerlerinin güncellenmesiyle alfanın en uygun pozisyonu hesaplanmaya çalışılır.



Şekil 3.10 Gri kurtların hiyerarşik yapısı [72]

Karınca kolonisi algoritmasında gri kurtlar algoritmasına benzer bir şekilde karıncaların günlük aktivitelerinden birisi olan yiyecek bulma davranışlarından ortaya çıkarılmış bir algoritmadır. Günlük hayatta koloni olarak yiyecek taşıma görevi gerçekleştiren karıncalar aldıkları besinlere en kısa yolla belirledikleri bölgeye götürmeye çalışırlar. Çok sayıda karınca farklı yollar deneyerek bunu gerçekleştirmeye çalışır. En kısa yolun bulunmasıyla karıncaların artık tek bir yoldan hareket ettiği gözlemlenebilir. Bu davranıştan esinlenerek ortaya çıkarılan

karınca koloni algoritmasında karıncalar başlangıç noktasına yerleştirilir ve hedef noktaya gitmeleri beklenir. İlk hareket esnasında farklı sayıda rotalar elde edilir. Karıncalar geçtikleri rotalara feromon hormonu bırakırlar. Feromon yoğunluğunun artması o güzergâhtan geçen karınca sayısının arttığı ve Şekil 3.11'de görüldüğü gibi en kısa yolun o yol olduğuna dair bilginin ulaşmasını sağlar.



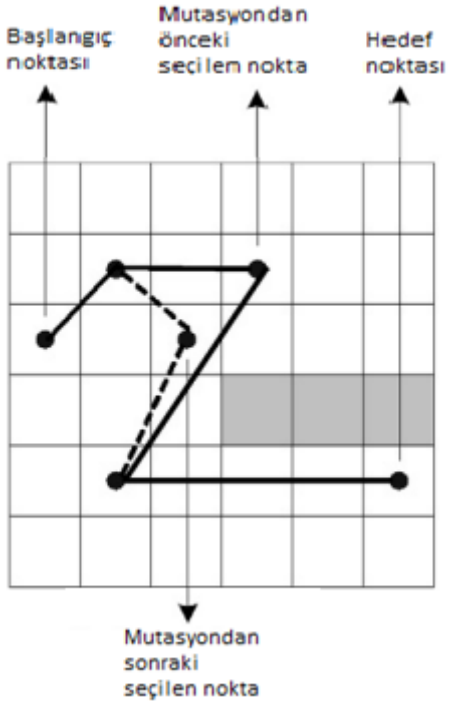
Şekil 3.11 Karınca kolonisinin yeni yol oluşturması [73]

Biyolojik temelinin yanında yapay zekâ temeli de bulunan kromozom ve gen yapılarının kullanıldığı genetik algoritma güçlü ve daha iyi olanın hayatta kaldığı, zayıf ve daha kötü olanın hayatta kalamadığı bir çalışma temelinden oluşmaktadır. Genetik algoritmanın çalışma mantığı şu şekildedir:

1. Başlangıç ve bitiş noktaları arasında rastgele rotalar elde edilir ve bunlar çözüm kümesini oluşturur.
2. Uygun yolun bulunması için bir fonksiyon oluşturulur ve rastgele oluşturulan rotalar bu fonksiyonun için yerleştirilir.
3. Fonksiyon sonucu yolun optimum yol değeri alınır. Eğer elde edilen değerler doğrultusunda çözüme ulaşıldıysa, değerler sabit kalıyorsa veya iterasyon sayısı sonlandırıldıysa çözümü arama işlemi biter. Aksi durumda çözümü arama işlemi devam eder.
4. Optimum yol değerleri yüksek olan veya çözüme yakın olduğu düşünülen rotalar için mutasyon ve çaprazlama işlemi yapılarak yeni bireyler elde edilir.

5. Elde edilen yeni bireyler başlangıçta belirlenen fonksiyona yerleştirilir ve sonuçlar tekrar değerlendirilir.
6. En uygun yola ulaşıldığında veya iterasyon sayısı sona erdiğinde algoritmanın çalışması durdurulur.

Şekil 3.12'de genetik algoritma ile yenilenen örnek bir yol planlaması gösterilmiştir.



Şekil 3.12 Genetik algoritma ile yol planlaması örneği [74]

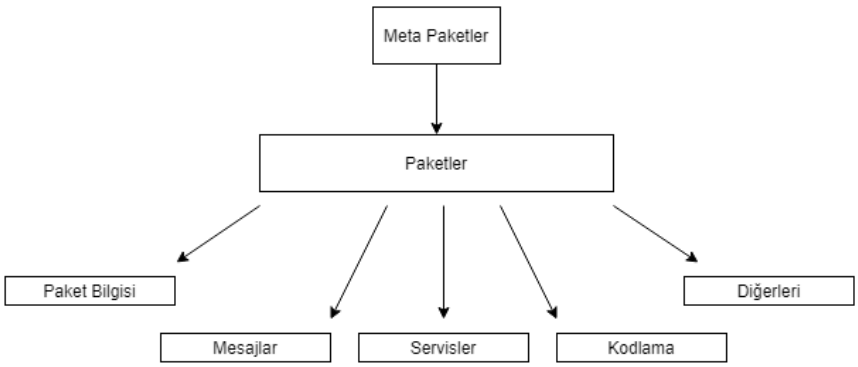
4
ROBOTİK İŞLETİM SİSTEMİ

Bu çalışmada robotun yazılım sisteminde Willow Garage tarafından geliştirilmiş olan ve robot yazılımlarının gerçekleştirilebilmesini sağlayan robotik operasyon sistemi kullanılmıştır [78],[79]. Bu sistemin tercih edilmesindeki sebebi ise içerisinde birçok programlama dili desteği bulunduran, küçük veya büyük çaplı tüm projeler için kullanılabilen, özgün bir şekilde içeriğinde değişiklikler yapılabilen, uçtan uca iletişimi sağlayabilen ve açık kaynak kodlu bir sistem yapısına sahip olmasından dolayıdır [75]–[78].

Mobil robotun kontrol çalışmaları ROS üzerinden gerçekleştirilmiştir. Çalışma alanının haritalanması, otonom yönelimin ve kontrolünün sağlanması bu sistem üzerinden yapılmıştır.

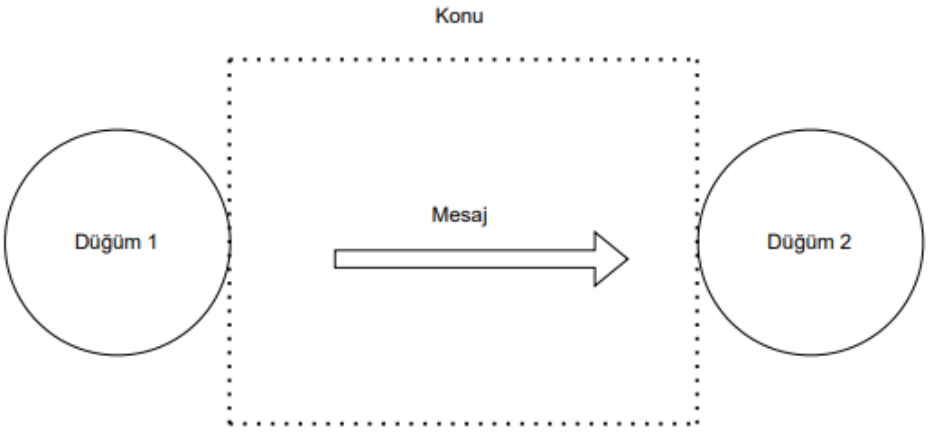
4.1 ROS Paket Yapısı

Ubuntu işletim sisteminde çalışmakta olan ROS içerisinde oluşturulan paketlerin birbirleriyle haberleşme esasıyla çalışmaktadır. Paket yapısı Şekil 4.1'de gösterildiği gibi temelde 4 başlıkta incelenebilir.



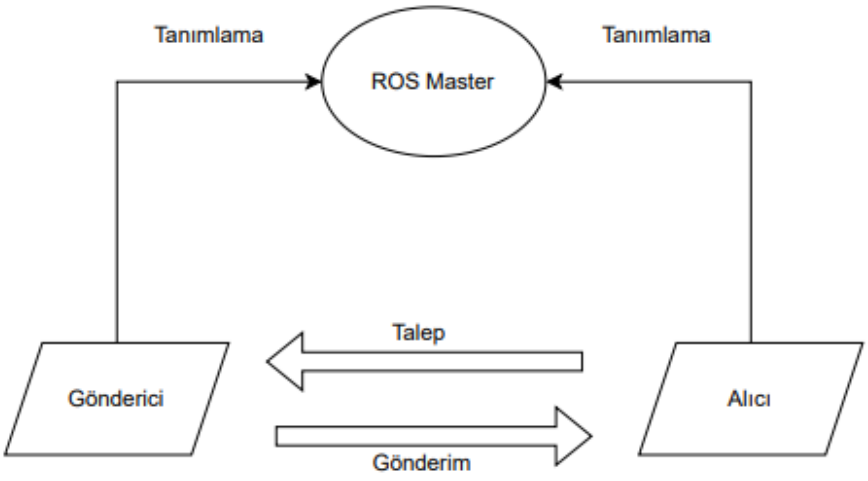
Şekil 4.1 ROS paket yapısı

- Temelde paketler üzerinde çalışmaktadır. Birden fazla paketin birlikte kullanılması durumunda ise meta paket olarak adlandırılmaktadır [78].
- Her bir paketin isim, versiyon, bağlı olduğu kaynaklar gibi bilgileri içeren "package.xml" dosyasına sahiptir.
- Düğümler arasındaki haberleşme "mesaj (msg)" ile sağlanmaktadır.
- Veri gönderim ve veri alımı hizmetleri "servis (srv)" ile sağlanmaktadır.



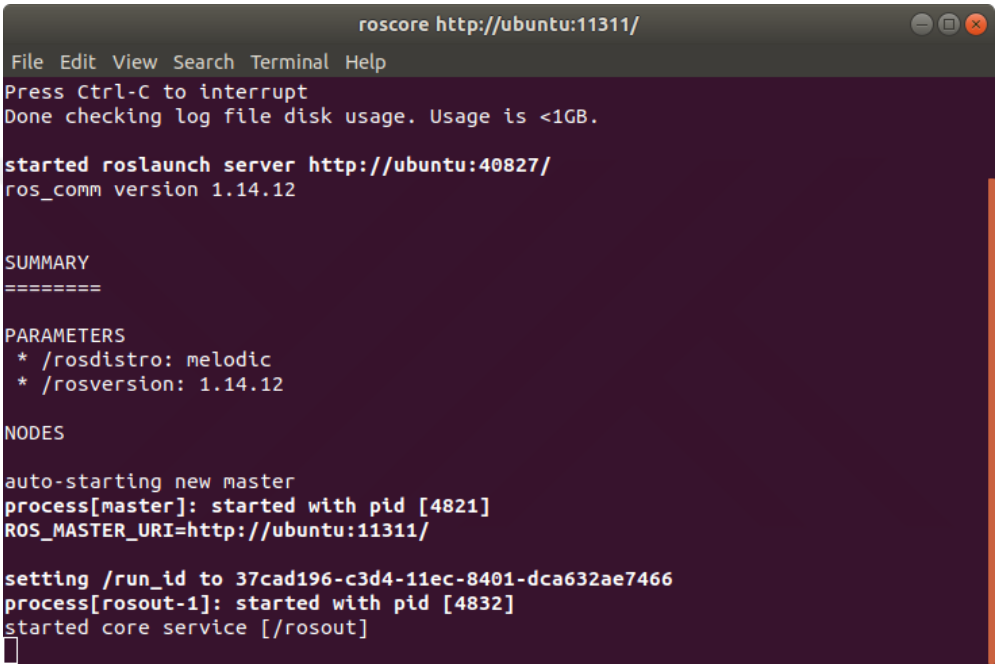
Şekil 4.2 ROS haberleşme yöntemi

Şekil 4.2'de basit bir iletişim örneği gösterilmiştir. Sistemde büyük bir dosya bulundurmak yerine küçük parçalar halinde sistem içinde birden fazla düğüm bulunmaktadır. Birbirleri arasındaki haberleşme ise mesaj yoluyla gerçekleşir ve gerçekleşen mesaj işlemi için bir konu oluşturulur. Haberleşme işleminde bir gönderici ve bir alıcı bulunur. Şekil 4.3'te düğüm 1 için gönderici, düğüm 2 için alıcı tanımlamaları yapılabilir. Göndericinin mesajı iletmesi için alıcıdan istek gelmesi gerekmektedir. Bu işlem için ise "servis" ihtiyacı bulunmaktadır. Düğümler arası iletişimlerde mesaj, konu, servis adlandırılmalarına göre sınıflandırılma yapılabilir.



Şekil 4.3 ROS master çalışma şeması

Gönderici ve alıcı arasında mesaj transferinin olabilmesi için birbirleri arasında bağlantı kurulması gerekir. Bu bağlantı ROS yöneticisi ile sağlanmaktadır. Göndericinin iletiyi göndermeye, alıcının ise iletiyi almaya gereksinimi bulunmaktadır. Bu sebepler aradaki talep ve gönderim işlerinin yapılması için ROS yöneticisi kullanılır. ROS yöneticisinin bu iletişimi sağlayabilmesi için ise hem göndericiyi hem de alıcıyı tanımlaması gerekir. Tanımlamanın ardından kayıtlar tutulur ve bağlantı kurularak iletişim gerçekleşir. Terminale yazılan roscore komutu ile Şekil 4.4'de gösterildiği gibi ROS yöneticisi çalıştırılır.

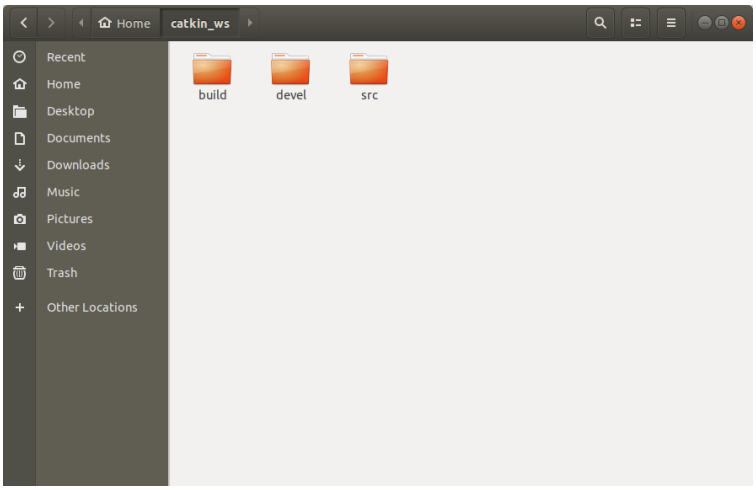


Şekil 4.4 Roscore komutu ile ROS yöneticisinin aktifleştirme

4.2 ROS Dosya Yapısı

Bu tez çalışmasının ana karakterini ROS oluşturmaktadır. Ubuntu işletim sistemine kurulan bu sistemin çalışma ortamı olarak adlandırılan ana başlığın altında 3 adet temel dosyası bulunmaktadır. Şekil 4.5'te Ubuntu içerisinde oluşturulan dosyalar gösterilmiştir.

1. src
2. build
3. devel

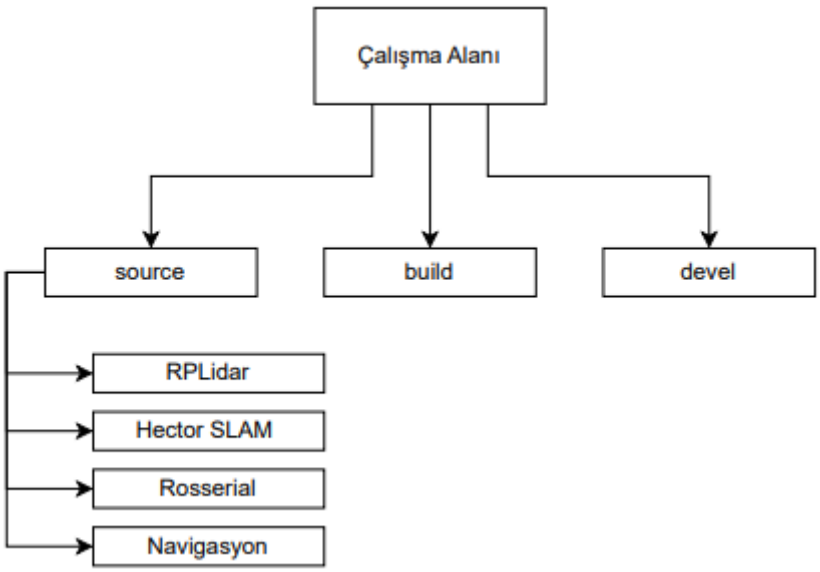


Şekil 4.5 ROS çalışma alanı

Uygulama çalışmalarında CMake sistemi adı verilen catkin kullanılmaktadır. Çalışma alanında paketlerin hazırlanmasının ardından catkin_make komutu ile kurulumlar gerçekleştirilir.

$ ~/catkin_ws$ catkin_make

Paketin içinde bulunan kaynak kodlarını içeren dosyalar src dosyasının içine gönderilmektedir. Verilecek olan komutların, yapılacak olan görevlerin tanımlamalarını içeren kodlanmış dosyalar, kütüphane dosyaları bu kısımda bulunmaktadır. Bu tez çalışmasında kullanılacak olan paketlerin listesi ve ROS dosya yapısı Şekil 4.6'da gösterilmiştir.



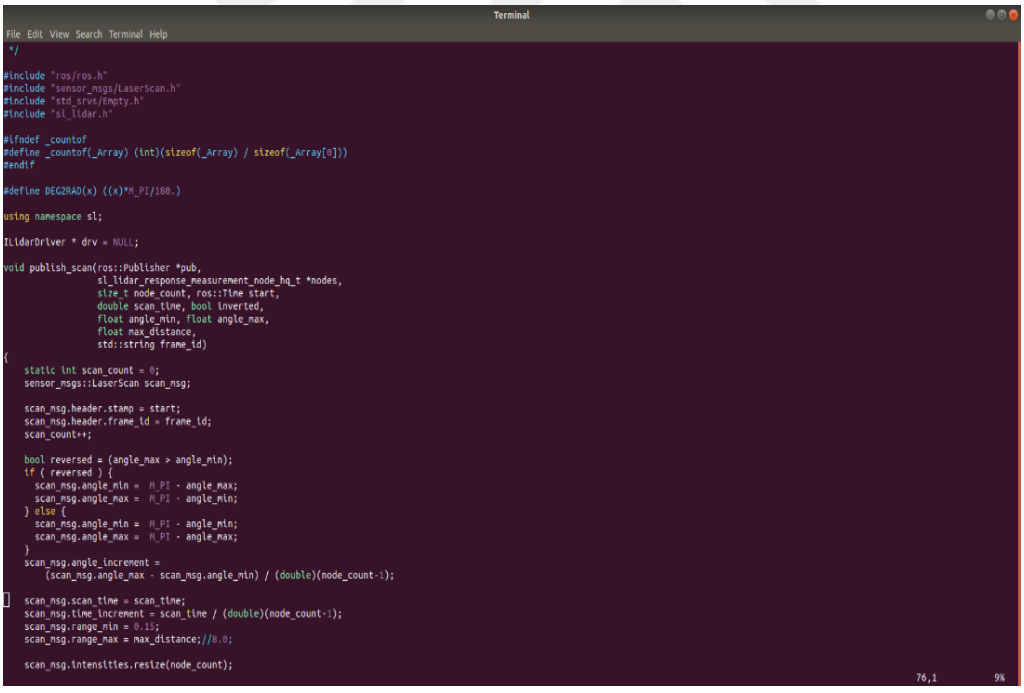
Şekil 4.6 ROS dosya yapısı

Build dosyasının içerisinde ise gerçekleştirilen çalışmaların ve paketlerin önbellek verileri ve bazı dosyaları saklanmaktadır. catkin_make komutu ile paketlerinin yüklenmesinin ardından yürütülebilir dosya oluşturulur ve bu dosya devel dosyasının içerisinde depo edilir.

Bu tez çalışması için Ubuntu işletim sisteminin yüklenmesinin ardından ROS Melodic sürümü yüklenmiştir. Bu sayede kullanılacak olan paketlerle birlikte sistem aktif şekilde kullanılabilecektir. Otonom sürüşün sağlanabilmesi için ana başlıkta 4 farklı paket kullanılmıştır. Bu paketlerin kurulması için ROS kütüphanesinden ek paketler de gerekmektedir.

1. RPLidar
2. Hector SLAM
3. Rosserial
4. Navigasyon

Şekil 4.7'de RPlidar paketinden örnek bir düğüm kodları gösterilmiştir.



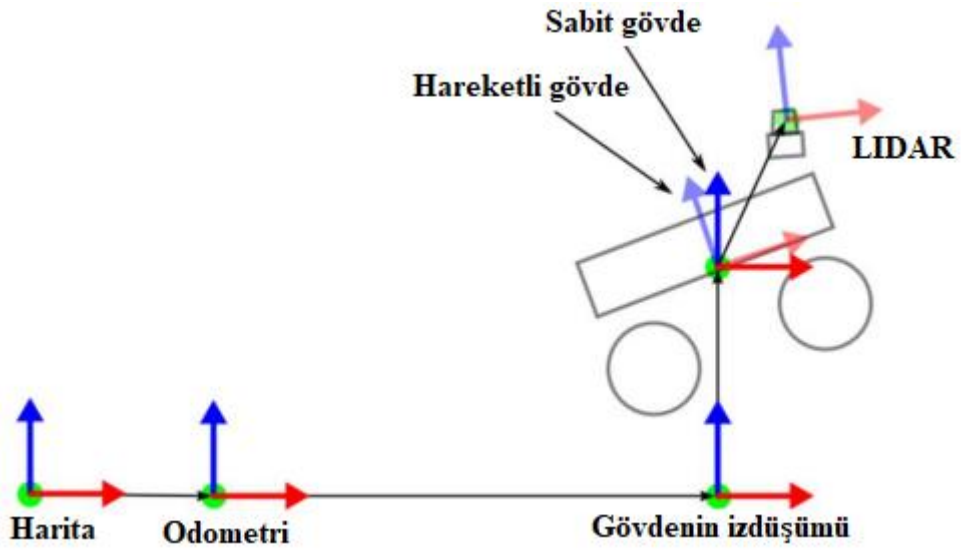
Şekil 4.7 RPLidar düğüm kodları

4.3 Koordinat Sistemleri

ROS altyapısında objelerin x, y ve z eksenlerinden oluşan koordinat sistemleri bulunmaktadır. Koordinat sistemlerini belirleyen faktörler arasında neyi referans aldığı ve neye göre sistemin oluşturulduğu yer almaktadır. Bu tez çalışmasında temelde 4 farklı koordinat sistem bulunmaktadır.

1. Hareketli gövde
2. Odometri
3. Harita
4. Işıklı mesafe ölçer

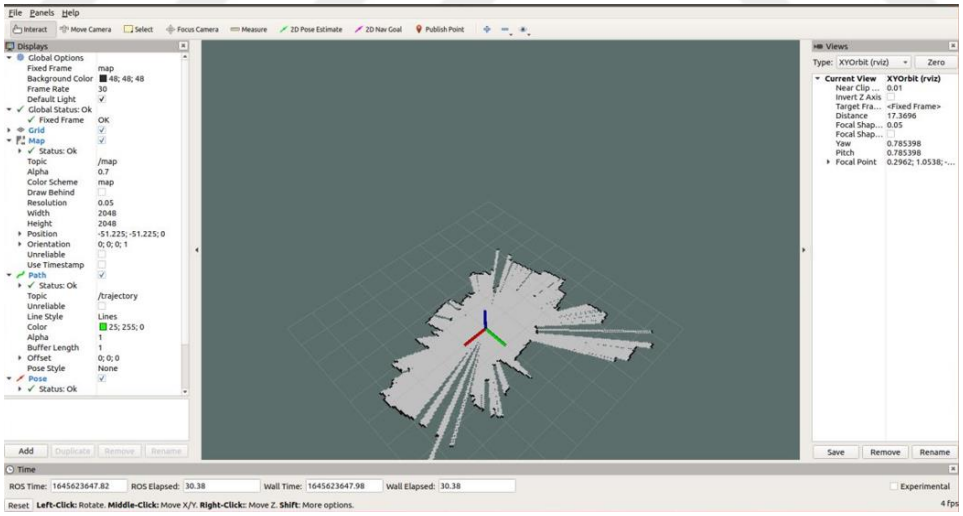
Robotun tabanının referans noktası kabul edilmesiyle hareketli gövde sistemi oluşturulur. Bu sisteme TF dönüşümlerin etkisiyle birlikte odometri, harita ve ışıklı mesafe ölçer sistemleri oluşturulur. Odometri koordinat sistemi ROS üzerinde /odom kısaltmasıyla kullanılmaktadır. Kullanıcıya robotun Dünya üzerinde bulunduğu konumunu gösteren koordinat sistemini oluşturur. "Harita" koordinat sistemi ise mobil robotun kullandığı sensörler vasıtasıyla robotun Dünya üzerinde bulunduğu konumu belirten koordinat sistemi oluşturur. Şekil 4.8'de koordinat sistemleri gösterilmiştir.



Şekil 4.8 Koordinat sistemi yapısı [79]

4.4 Haritalama

Kapalı alanlarda mobil robotların otonom olarak hareketlerinin sağlanabilmesi için çevresel faktörlerin sensörler tarafından alınıp yorumlanmasına ihtiyaç bulunmaktadır. Bu verilerin yorumlanması sonucu oluşturan haritalama ve robotun bulunduğu konumun tespit çalışması SLAM olarak adlandırılmaktadır [80]. Doğal afetlerin veya insanların meydana getirdiği değişimler robotların çalışma alanlarını etkilemektedir. Doğru bir hareket planlanmasının sağlanabilmesi için ortaya çıkan çevreyi tanıma sorununun çözülmesi gerekir. Bu tez çalışmasında Stefan Kohlbrecher tarafından geliştirilmiş olan açık kaynaklı Hector Slam yöntemi kullanılmıştır [81]. SLAM algoritması parçacık filtresini esas alarak bilinmeyen alanlarda çalışabilme kolaylığı sağlamaktadır [77]. Engellerin tespit edilerek harita oluşturabilmesi için bir arayüz ihtiyacı bulunmaktadır. Bu sebeple haritalamanın, konumlandırmanın, hareket planlanmasının yapılabilmesi için Rviz paketi [82], [83] kullanılmıştır. Işıklı mesafe sensörünün USB bağlantının yapılmasının ardından Rviz çalıştırılmış ardından kullanılacak olan paketlerinin devreye girmesi sağlanmıştır. Şekil 4.9'da bir odada gerçekleştirilen haritalama denemesi gösterilmiştir.



Şekil 4.9 Rviz üzerinden çalışma ortamının haritalaması

Yapılan haritalama işleminde siyah, gri ve beyaz renklerinde alanların olduğu görülmektedir. Haritada oluşan beyaz alanlar herhangi bir engelin olmadığı ve hareket edilebilen güvenli bölgeyi, gri alanlar bölge hakkında bilgiye sahip olunmadığı bölgeyi, siyah alanlar ise engellerin tespit edildiğini belirtmektedir [84].

Mobil robotun gerçek zamanlı kontrolünün sağlanabilmesi ve çevresinde olan durumların yorumlanabilmesi için sensörlerle donatılması gerekmektedir. Bu bağlamda kamera, GPS, radar, ışıklı mesafe ölçer gibi araçlar kullanılmaktadır [85]. Bu araçlar sayesinde robotun bulunduğu konum, gidebileceği konum ve karşılaştığı engellerin değerlendirilebilmesini sağlayan harita oluşturulmaktadır. Haritalama için noktasal haritalama, doluluk ızgara yöntemiyle haritalama, topolojik haritalama gibi yaklaşımlar bulunmaktadır [38].

Son zamanlarda LIDAR (Light Detection and Ranging) teknolojisinin kullanımı geniş bir yelpaze içine konumlandırılmış bulunmaktadır. Bazı dezavantajlarına rağmen tercih edilmesini sağlayan birçok neden [86], [87] bulunmaktadır. Şeffaf cisimlere karşı elde edilen verilerin hatalı olmasına [43] karşın; engellerin tespit edilebilmesi, haritalama yapılabilmesi, uzaktan kontrolle bağlı olduğu robotun hareketinin sağlanabilmesi, robotun hareket tahmininin gerçekleştirilebilmesi, nesne takibinin yapılabilmesi, robota yerleştirilmesiyle konumunun bilinebilmesi, geniş bir çalışma alanına sahip olması ve kullanım kolaylığının bulunmasından dolayı tercih edilmektedir.

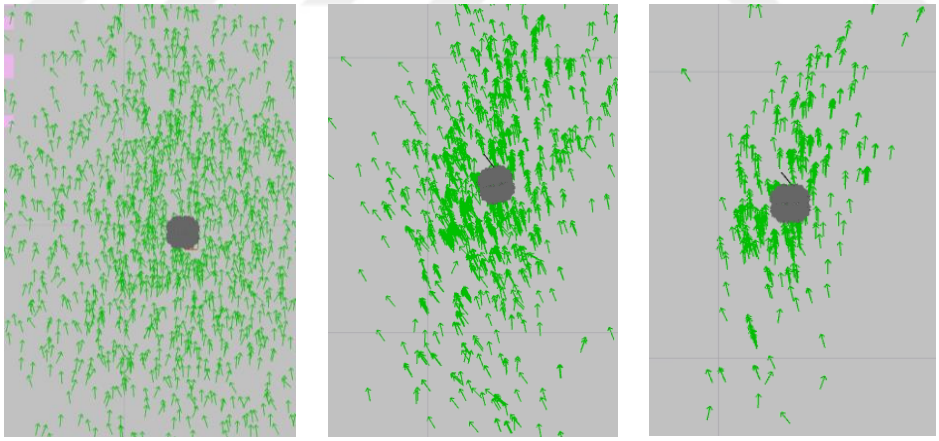
Tablo 4.1'de farklı üretici firmalara ait LIDAR özellikleri ile HC-SR04 ultrasonik sensörün özellikleri verilmiştir. HC-SR04'ün sensörünün engel tespiti yapabilmesinden dolayı LIDAR yerine de kullanılabilmektedir [88].

Tablo 4.1 Lidar ve mesafe sensör özellik karşılaştırmaları [53], [89]–[91]

Sensör Modeli	**Maksimum Mesafe (m)**	**Açı (°)**	**Tolerans (mm)**	**Fiyat ($)**
Hokuyo UTM-30LX	30	270	10-30 m:± 50 mm	3500
RPLidar A1 M8	6	360	1.5–6 m: %1	100
YDLIDAR X2	8	360	1–6 m: %3.5	70
HC-SR04	4	15	2-400cm:3 cm	1.1

4.5 AMCL

Mobil robotların konumlarını değiştirdikleri sırada çevresindeki hareketli veya hareketsiz nesnelere dair bilgiye sahip olması önemlidir. Bu sebeple bir noktadan başka bir noktaya olan hareket sırasında robotun konumunun doğru bir şekilde yansıtılması gerekir. Robotların konum tahmini yapabilmeleri için en çok kullanılan algoritmalar istatistiksel yöntemlerin kullanıldığı genişletilmiş kalman filtresi [92] ve sensor verilerinin kullanıldığı AMCL (Adaptif Monte Carlo Lokalizasyonu) [93] algoritmalarıdır. Mobil robotun haritaya yerleştirilmesinden sonra öncelikli olarak bulutsal bir görünümde oklar haritanın tamamına rastgele olarak yayılır. Bu durumda robotun konumuyla ilgili net bir tahmin yoktur, sadece genel bir olasılıksal tahmin gerçekleştirilir. Ardından robotun ufak hareketleri sonucunda bu okların yenileme ile konumlarında değişmeler olur. Robotun hareketinin biraz da anlaşılır olmasıyla oklar robota doğru yanaşarak mevcut konumu hakkında bilgi sahibi olunmasını sağlar. Okların birbirlerine yakın ve sıkı olmaları robotun konumunun o bölge üzerinde olduğunu göstermektedir. Şekil 4.10'da simülasyon çalışmasında mobil robotun AMCL algoritmasıyla konum tahminlerinin sağlanması gösterilmiştir.



a) b) c)

Şekil 4.10 a) AMCL algoritması ilk tahmin, b) Hareketin başlamasıyla konum tahminindeki iyileşme, c) Hareketin bitişi sırasındaki konum tahmini

ROS içerisinde kurulumu yapılan ve kullanılan AMCL algoritmasının sonuçları doğrudan etkileyecek parametreleri bulunmaktadır. Parçacık sayısının düşük seviyelerde olması tahminin doğruluğunu azaltırken, yüksek seviyelerde olması işlemci gücünü arttırması ve kompleks bir yapıya bürünmesinden dolayı minimum

300, maksimum 5000 parçacık sayısı olacak şekilde kullanılmıştır. Kullanılan parametrelere dair bilgiler Tablo 4.2'de gösterilmiştir.

Tablo 4.2 Kullanılan bazı AMCL parametreleri

Parametre	**Parametrenin değeri**
Minimum parçacık sayısı	300
Maximum parçacık sayısı	5000
D parametresi minimum güncelleme	0.1
A parametresi minimum güncelleme	0.2
Yeniden örnekleme aralığı	1.0
Dönüşüm toleransı	2.0

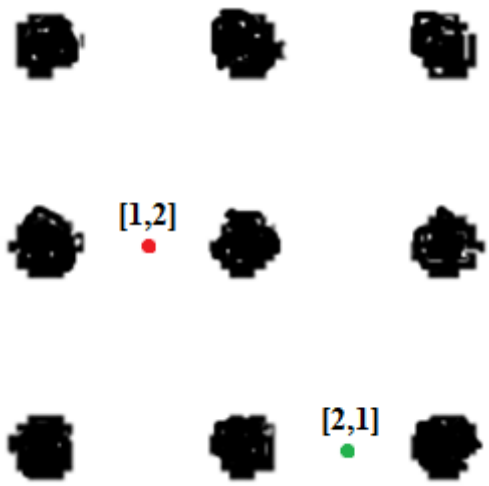
Parametrik değerlerin birbirleriyle uyuşmaması veya uygun olmaması durumunda konum doğrulaması yapılamayan mobil robot uygun hareketler göstermeyerek kontrolden çıkabilmektedir.

5
YOL PLANLAMA ALGORİTMALARININ SİMÜLASYON ÇALIŞMALARI

Bu bölümde mobil robotun manuel olarak sürüşü ve ışıklı mesafe sensörü kullanılarak elde edilen haritanın MATLAB programına aktarılması ve bu harita üzerinde belirlenen iki nokta arasında klasik yöntem olarak kullanılmaktan olan PRM ve RRT* algoritmalarıyla yol güzergâhlarının oluşturulması ve elde edilen planlama sonuçlarının karşılaştırılması gerçekleştirilmiştir. Bu algoritmalardan yola çıkarak geliştirilen yeni algoritmanın simülasyon sonuçları ile PRM ve RRT algoritmalarının simülasyon sonuçları karşılaştırılmış ve deneysel çıktıları gösterilmiştir.

5.1 PRM Algoritması Simülasyon Çalışmaları

PRM algoritması, örnekleme temelli olarak çalışan klasik yöntemlerden birisidir. Harita üzerine rastgele yerleştirilmiş noktaların birleştirilmesiyle en kısa yolu bulma amacıyla kullanılmaktadır. Bu tez çalışmasında farklı konumlara yerleştirilmiş 6 adet engelden oluşan haritanın MATLAB ortamına aktarılması ile simülasyon çalışmaları gerçekleştirilmiştir.



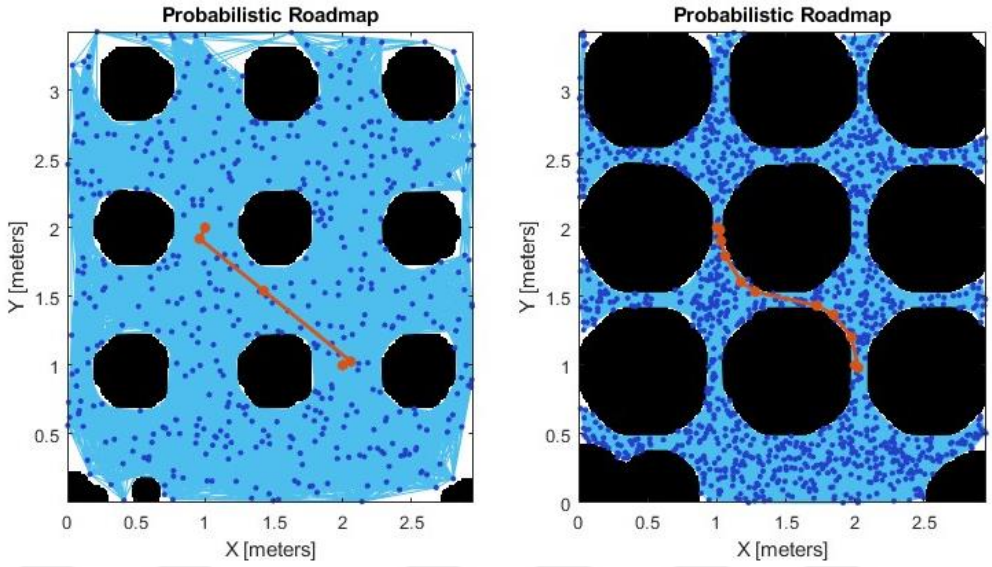
Şekil 5.1 Simülasyon çalışmalarının yapıldığı harita

Mobil robotun başlangıç noktası (2,1) ve bitiş noktası (1,2) koordinatları olarak Şekil 5.1'de olduğu gibi kabul edilmiştir. Bu iki nokta arasında herhangi bir engelin bulunmadığı düşünüldüğünde noktalar arası uzaklık 1,41 birim olarak bulunmaktadır. Fakat noktaların arasında engellerin bulunmasından veya robotun boyutlarından kaynaklı engele çarpma durumlarından dolayı yol güzergâhının uzunluğu değişebilmektedir. PRM algoritmasında oluşturulacak olan düğüm sayısı ve engellerin şişkinlik yarıçap değerlerinin girilmesiyle farklı yol uzunluklarına sahip yol güzergâhlar elde edilmiştir. Tablo 6.1'de PRM algoritmasıyla gerçekleştirilen simülasyon çalışmaları için kullanılan parametrik değerler gösterilmiştir.

Tablo 5.1 PRM algoritması simülasyon parametreleri

Engellerin Şişkinlik Yarıçapı	**Düğüm Sayısı**
0.2	500,600,700,800,900,1000 1100,1200,1300,1400,1500
0.3	500,600,700,800,900,1000 1100,1200,1300,1400,1500

Engellerin şişkinlik yarıçapı, engelin bulunduğu noktadan belirtilen yarıçap değerinde bir daire çizilmesiyle dairenin tamamının engel olarak algılanmasını sağlamaktadır. Engeller 0.2 ve 0.3 metre yarıçaplarında büyültülmüştür; harita üzerine rastgele olarak yerleştirilecek düğüm sayısı ise 500'den başlayarak 1500'e kadar 100 adet artacak şekilde seçilmiştir. Rastgele atanan noktaların her biri arasında birleştirme işlemi yapılarak bir yol oluşturulmaktadır. Parametrik değerlerinin arttırılmasıyla yol uzunluklarında değişimlerin gözlenebileceği beklenmektedir. Kullanılan mobil robotun 2. bölümde gösterildiği gibi 340x485x130 mm boyutlarındadır. Robotun merkez noktası referans alındığında yan yüzeyleriyle engel arasındaki mesafenin minimum 170 milimetre, ön ve arka yüzeylerinde ise engelle arasındaki mesafenin en az 243 milimetre olması gerekmektedir. Şekil 5.2'de farklı şişkinlik yarıçaplarına sahip harita üzerinde gerçekleştirilen simülasyon denemeleri gösterilmiştir



a) b)

Şekil 5.2 a) Engellerin şişkinlik yarıçapının 0.2 metre olduğu simülasyon denemesi, b) Engellerin şişkinlik yarıçapının 0.3 metre olduğu simülasyon denemesi

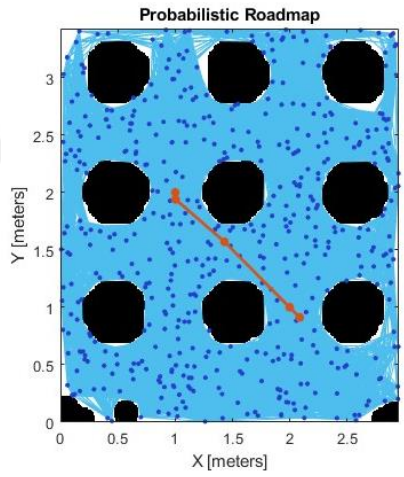
Haritada engeller üzerine uygulanan şişkinlik yarıçapı ile engellerin boyutları arttırılmıştır. Lacivert noktalar rastgelen atanan düğüm noktalarını temsil etmektedir. Mavi renkli çizgiler ise noktaların birleştirilmesiyle elde edilen yolları göstermektedir. Turuncu yol güzergâhı ise en uygun düğüm noktaların ve oluşturulan küçük yolların seçilmesiyle hedefe ulaşılan yolu göstermektedir. Simülasyonları gerçekleştirilen durumlar Tablo 5.2'de gösterilmiştir. Gerçekleştirilen durumlar sonucu elde edilen yol planlamaları Şekil 5.3'te gösterilmiştir.

Tablo 5.2 Simülasyonların gerçekleştirildiği durumlar

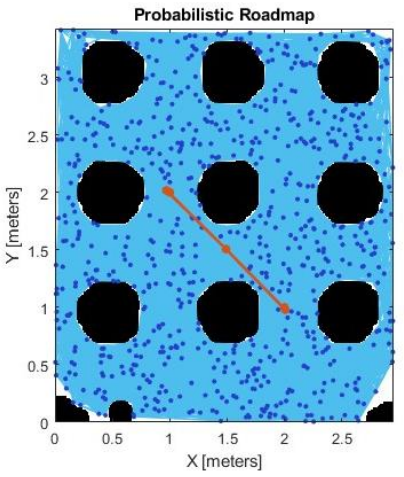
Durum Numarası	**Düğüm Sayısı**
Durum 1	500 düğüm
Durum 2	600 düğüm
Durum 3	700 düğüm
Durum 4	800 düğüm

Tablo 5.2 Simülasyonların gerçekleştirildiği durumlar (devamı)

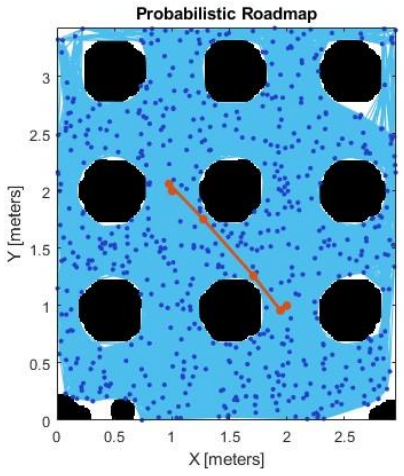
Durum 5	900 düğüm
Durum 6	1000 düğüm
Durum 7	1100 düğüm
Durum 8	1200 düğüm
Durum 9	1300 düğüm
Durum 10	1400 düğüm
Durum 11	1500 düğüm

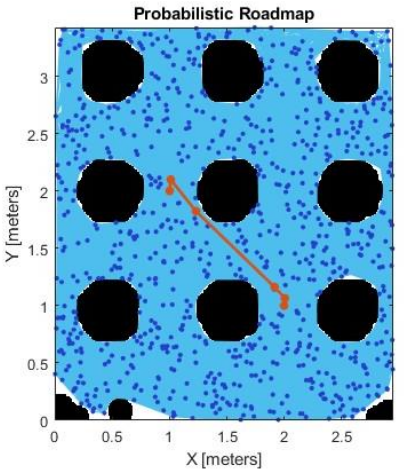


a)

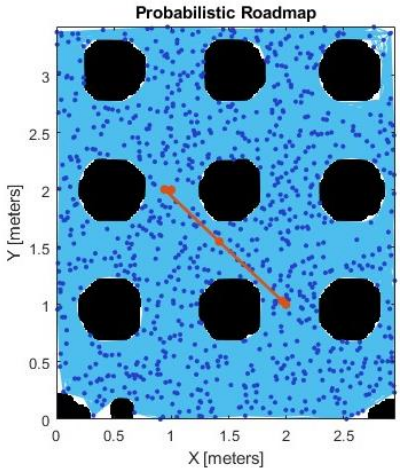


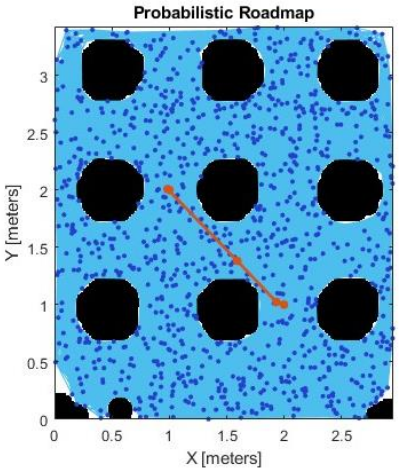
b)



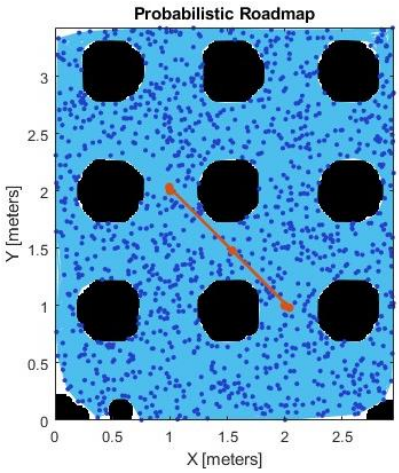


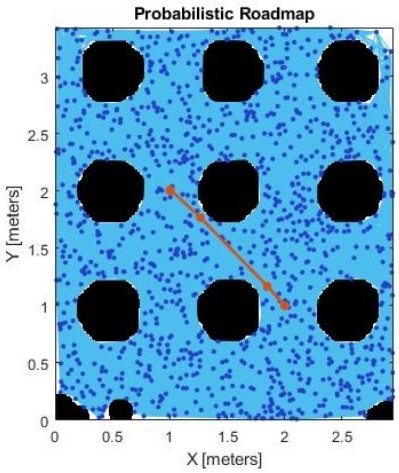
c) d)


Probabilistic Roadmap
Y [meters]
X [meters]

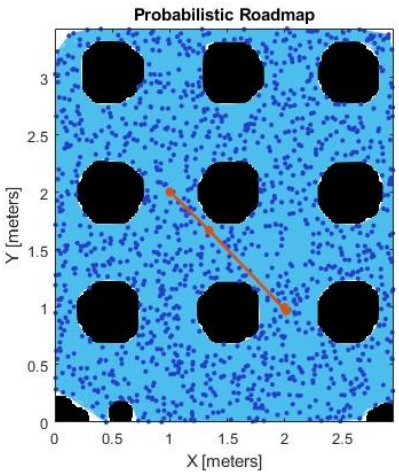

Probabilistic Roadmap
Y [meters]
X [meters]

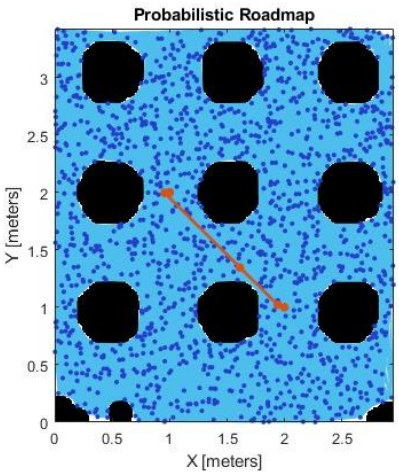
e) f)


Probabilistic Roadmap
Y [meters]
X [meters]

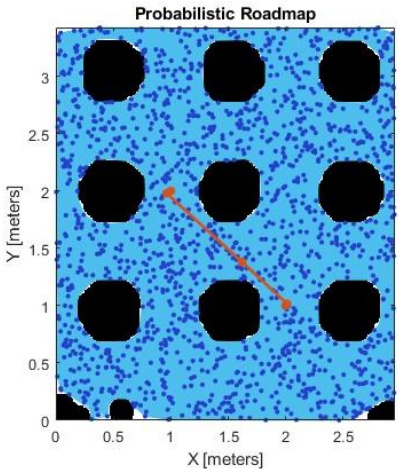

Probabilistic Roadmap
Y [meters]
X [meters]

g) h)


Probabilistic Roadmap
Y [meters]
X [meters]


Probabilistic Roadmap
Y [meters]
X [meters]

i) j)



h)

Şekil 5.3 a) 1. durum, b) 2. durum, c) 3. durum, d) 4. durum, e) 5. durum, f) 6. Durum, g) 7. durum, h) 8. durum, i) 9. durum, j) 10. durum, k) 11. durum

İlk simülasyon denemelerinde düğüm sayısının artmasıyla işlem süresinin uzadığı görülmüştür. Alınan tüm sonuçlar denemelerin ilk sonuçları olarak kullanılmıştır. Düğüm noktalarının rastgele atanmasından dolayı aynı durum birden fazla gerçekleştirildiğinde yol planlaması ve elde edilen optimum yol değişmektedir. Yol uzunluk değerleri robotun yük taşıyıcı kısmının bitiş noktası üzerinde olduğu duruma göre hesaplanmıştır. Tablo 5.3'te oluşturulan yol planlamaları sonucu ulaşılan yol uzunluk değerleri gösterilmiştir.

Tablo 5.3 Yol uzunluk değerleri

Düğüm Sayısı	Toplam Yol Uzunluğu (metre)
500	1.460
600	1.552
700	1.430
800	1.529
900	1.432
1000	1.470
1100	1.445

Tablo 5.3 Yol uzunluk değerleri (devamı)

1200	1.423
1300	1.540
1400	1.454
1500	1.456

Sonuçlar incelendiğinde düğüm sayısının artması veya azalmasıyla yol uzunluğu arasında bir ilişkinin olmadığı görülmüştür. 1. durumda 500 düğüm sayısı ile 1.46 metre uzunluğunda yol planlanmasına rağmen, 9. durumda 1300 düğüm sayısı ile 1.54 metre uzunluğunda yol planlaması gerçekleştirilmiştir. Ortalama yol uzunluğu 1.472 metre olduğu görülmüştür. Yol planlaması sırasında başlangıç ve bitiş noktaları arasında olmayan noktaların gereksiz güç tüketimi oluşturduğu ve atanan düğüm noktalarının verimini düşürdüğü görülmüştür. Şekil 5.4'te örnek bir planlamanın gerekli düğüm noktaları ve düğüm nokta sayısı gösterilmiştir.



Number of Brushed Points	Type	Tag
5	line	
156	scatter	
15540	line	

a) b)

Şekil 5.4 a) 1. durumda gerekli olan düğüm noktaları, b) Kırmızı bölgedeki düğüm noktalarının sayısı

Bu yol planlaması için önemli olan düğüm noktaları kırmızı bölgede bulunan düğüm noktaları olmalıdır. Bu bölgenin dışındaki noktaların yol planlamasına bir etkisi bulunmamaktadır. Tüm durumlar için bu bölgede bulunan nokta sayısı ve bu noktaların oluşturulan toplam düğüm noktalarına oranı Tablo 5.4'te gösterilmiştir.

Tablo 5.4 Gerekli düğüm sayıları ve noktaların verimlilik verileri

Toplam Düğüm Sayısı	Gerekli Düğüm Sayısı	Noktaların Verimliliği
500	156	0.312
600	194	0.323
700	221	0.316
800	232	0.29
900	284	0.316
1000	292	0.292
1100	327	0.297
1200	386	0.322
1300	398	0.306
1400	404	0.288
1500	418	0.279

Toplamda 600 düğümün kullanıldığı 2. durumda düğüm noktalarının %32.3 oranında yol planlamasının bulunduğu bölgeye atandığı görülmüştür. Yol uzunluğu değerleri incelendiğinde ise 1.552 metre ile en uzun yolun bu durumda oluştuğu görülmektedir. Engellerin şişkinlik değeri 0.2'den 0.3'e yükseltilerek aynı işlemler tekrar gerçekleştirilmiştir. Planlanan yolların uzunluk değerleri parametrik değerinin değişimiyle tekrardan ölçülmüştür. Toplam yol uzunluklarına ait veriler Tablo 5.5'de gösterilmiştir.

Tablo 5.5 Engellerin şişkinlik yarıçapı 0.3 olduğunda toplam yol uzunlukları

Düğüm Sayısı	Toplam Yol Uzunluğu (metre)
500	1.830
600	1.600
700	1.560
800	1.600
900	1.580
1000	1.550
1100	1.640
1200	1.520
1300	1.490
1400	1.620
1500	1.580

Engellerin şişkinlik yarıçapının %50 oranında arttırılması yol uzunluk değerlerini arttırdığı görülmüştür. Engellerin boyutunun büyümesiyle birlikte hedefe gidilecek olan yolda kavislerin artmasına ve gidilen düğüm noktası sayısının artması neden olmuştur. Bu sebeplerden ötürü yol uzunluğu artmıştır. Şişkinlik yarıçapı 0.2 olduğunda ortalama yol uzunluk değerinin 1.477 metre olduğu, yarıçapın 0.3 olduğu durumlarda simülasyon gerçekleştirildiğinde ise ortalama yol uzunluğunun 1.60 metreye yükseldiği görülmüştür. Gerekli düğüm nokta sayısının ve noktaların verimlilik değerleri Tablo 5.6'da gösterilmiştir.

Tablo 5.6 Engelleri şişkinlik yarıçapı değeri 0.3 olduğunda gerekli düğüm sayıları ve noktaların verimlilik verileri

Toplam Düğüm Sayısı	Gerekli Düğüm Sayısı	Noktaların Verimliliği
500	135	0.270
600	174	0.290
700	187	0.267
800	221	0.276
900	244	0.271
1000	285	0.285
1100	295	0.268
1200	344	0.287
1300	348	0.268
1400	384	0.274
1500	392	0.261

Engellerin şişkinlik yarıçapının 0.3 değeri yükseltilmesiyle birlikte düğüm noktalarının atanabileceği bölgenin engel olarak tanımlanması gerekli olan düğüm noktası sayısında düşüşe sebebiyet vermiştir. Gerekli olan düğüm nokta sayısının azalması sonucunda verim sayısında azalma olduğu gözlemlenmiştir. Verimlilik ve planlanan yolun uzunluğu bakımından engellerin şişkinlik yarıçapı değeri 0.2 olduğunda sonuçları daha uygun olduğu görülmüştür. Fakat mobil robotun x ve y ekseni boyunca uzunluk değerleri dikkat alındığında engel yarıçap değerinin 0.243 değerinden büyük olması gerekmektedir. Bu sebeple engeller şişkinlik yarıçapının değeri 0.3 olduğunda elde edilen verilerek kullanılarak karşılaştırmak yapılmaktadır.

5.2 RRT* Algoritması Simülasyon Çalışmaları

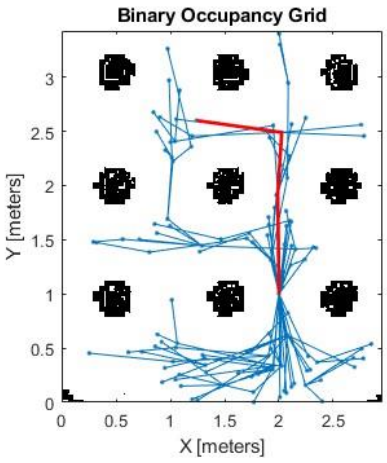
RRT algoritması dallanma tabanlı olarak çalışan klasik yöntemlerden birisidir. Harita üzerine rastgele yerleştirilmiş noktaların dallandırma yöntemiyle birleştirilerek en kısa yolu bulma amacıyla kullanılmaktadır. PRM algoritmasının simülasyon çalışmalarında kullanılan haritanın MATLAB ortamına aktarılması ile RRT algoritmasının gelişmiş versiyonu olan RRT* algoritmasının simülasyon çalışmaları gerçekleştirilmiştir.

RRT* algoritmasında oluşturulacak olan iterasyon sayısı değerlerinin değiştirilmesiyle farklı yol uzunluklarına sahip yol güzergâhları elde edilmiştir. Tablo 5.7'de RRT* algoritmasıyla gerçekleştirilen simülasyon çalışmaları için kullanılan parametrik değerler gösterilmiştir.

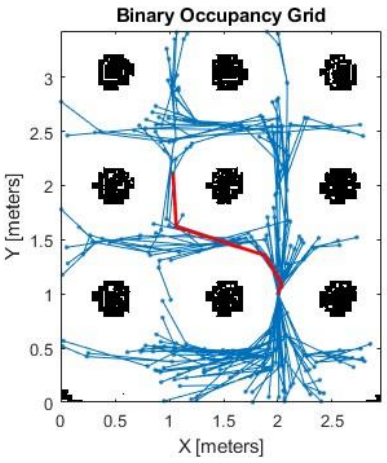
Tablo 5.7 RRT* algoritması simülasyon parametreleri

Maksimum Bağlantı Mesafesi (metre)	**Engellerin Şişkinlik Yarıçapı (metre)**	**İterasyon Sayısı**
1	0.2,0.3	500,1000,2000,3000,4000 5000,6000,7000,8000,9000

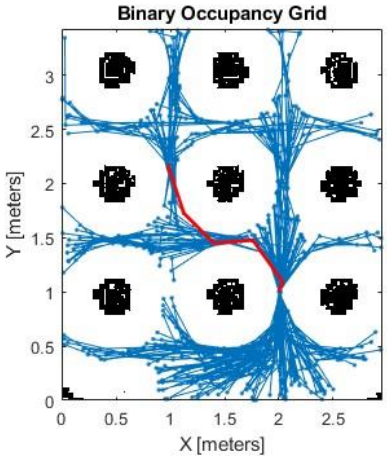
Noktaların birbirleri arasında bağlantı kurması ve yol planlamasının gerçekleşmesi için maksimum izin verilen bağlantı mesafesi 1 metre olarak kabul edilmiştir. Engellerin şişkinlik yarıçap değerleri 0.2 ve 0.3 metre olarak seçilmiştir. Harita üzerinde gerçekleştirilecek olan yol planlaması için iterasyon sayısı 500'den başlayarak 1000, 2000 şeklinde artarak ilerleyen ve 9000 iterasyonda son bulan simülasyon çalışmaları gerçekleştirilmiştir. Engellerin şişkinlik yarıçapının 0.2 metre olduğu durumlarda gerçekleştirilen simülasyon çalışmaları Şekil 5.5'te gösterilmiştir.


Binary Occupancy Grid
Y [meters]
X [meters]

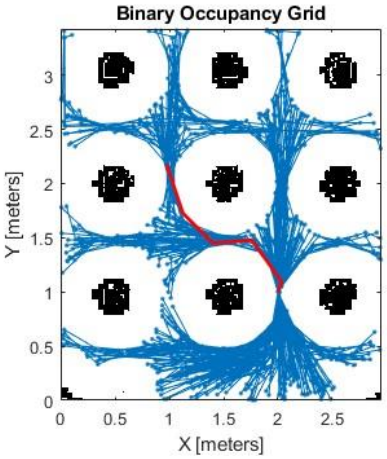
a)


Binary Occupancy Grid
Y [meters]
X [meters]

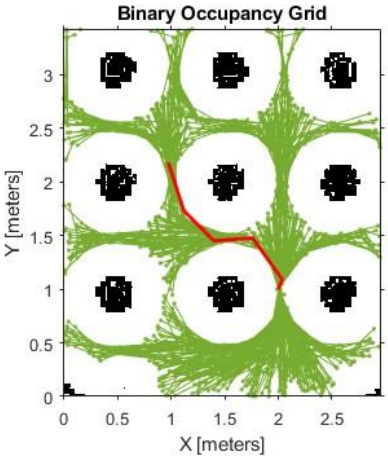
b)


Binary Occupancy Grid
Y [meters]
X [meters]

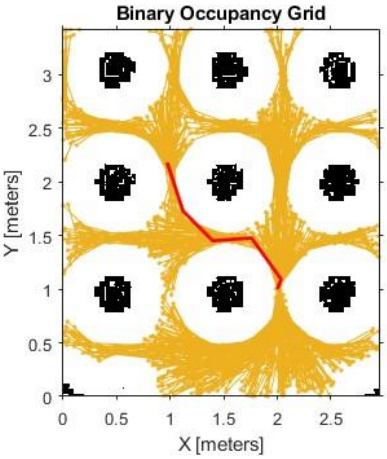
c)


Binary Occupancy Grid
Y [meters]
X [meters]

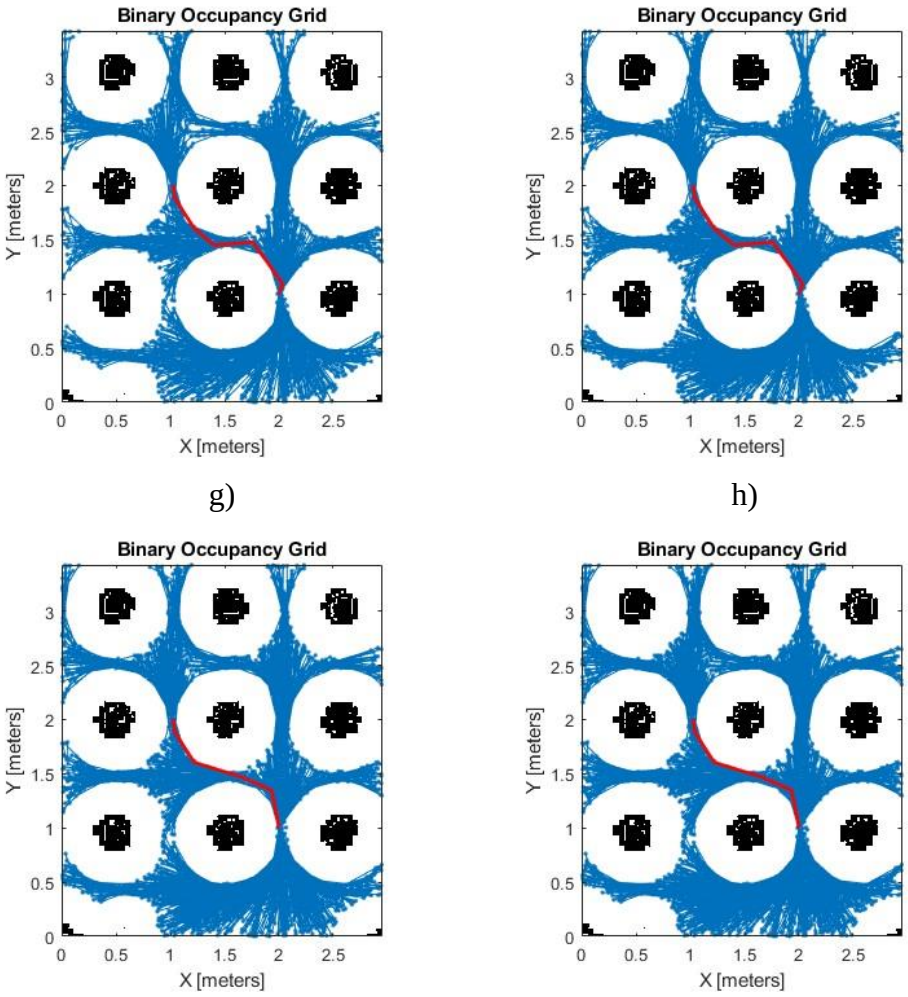
d)


Binary Occupancy Grid
Y [meters]
X [meters]

e)


Binary Occupancy Grid
Y [meters]
X [meters]

f)



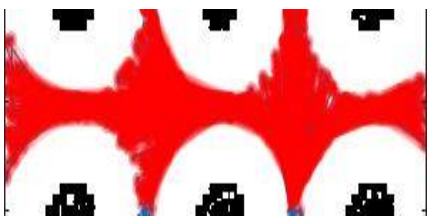
Şekil 5.5 a) 500 iterasyon, b) 1000 iterasyon, c) 2000 iterasyon, d) 3000 iterasyon, e) 4000 iterasyon, f) 5000 iterasyon, g) 6000 iterasyon, h) 7000 iterasyon, i) 8000 iterasyon, j) 9000 iterasyon

Simülasyon sonucu oluşturulan yol güzergâhlarında bulunan düğüm noktaları kullanılarak toplam yol uzunluğu hesaplanmış ve Tablo 5.8'de gösterilmiştir. İterasyon sayısının arttırılmasının toplam yol uzunluğunu kısalttığı; fakat iterasyon sayısındaki artıştan dolayı yol planlama süresinin uzamasına yol açtığı görülmüştür. 2000, 3000 ve 4000 iterasyonun kullanıldığı çalışmalarda iterasyon sayısında ciddi bir artış olmasına karşın toplam yol uzunluğunun her bir iterasyon için de 1.8 metre değerinde olduğu ve bir değişiklik olmadığı görülmüştür. 6000 ve 7000 iterasyona sahip 1.665 metre yol uzunluğuna ulaşıldığı çalışmalar ve 8000 ile 9000 iterasyona sahip 1.444 metre yol uzunluğuna ulaşıldığı çalışmalarda da aynı duruma rastlanmaktadır.

Tablo 5.8 RRT* algoritmasında engel yarıçapının 0.2 metre olduğu simülasyon çalışmalarında toplam yol uzunluk değerleri

İterasyon Sayısı	Toplam Yol Uzunluğu (metre)
500	2.015
1000	1.761
2000	1.800
3000	1.800
4000	1.800
5000	1.705
6000	1.665
7000	1.665
8000	1.444
9000	1.444

PRM algoritmasında olduğu gibi RRT* algoritmasında da yol planlamasının dışında düğüm noktaları ve yol güzergâhları oluşturulmuştur. Bunlar algoritmanın çalışmasını yavaşlatmakta, optimum yolun bulunmasını zorlaştırmakta ve gereksiz işlem hacmi oluşturmaktadır. Şekil 5.6'da gösterildiği gibi önemli sayılabilecek düğüm noktaları kırmızı bölgedeki alan içinde kalan noktalardır.



Number of Brushed Points	Type	Tag
6	line	
1833	line	

a) b)

Şekil 5.6 a) 9000 iterasyona sahip çalışmada gerekli olan düğüm noktaları, b) Kırmızı bölgedeki düğüm noktalarının sayısı

Tüm iterasyon değerleri için gerekli düğüm sayıları ve düğüm noktalarının verimliliği Tablo 5.9'da gösterilmiştir. Elde edilen verilere göre 3000 iterasyon ve üstündeki düğüm atamalarında verimlilik değerlerinde çok fazla değişiklik olmadığı görülmüştür. Aynı yol uzunluk değerlerine sahip 2000, 3000 ve 4000 iterasyonun kullanıldığı çalışmalar kıyaslandığında en verimli noktala atamalarının 3000 iterasyonda gerçekleştiği, 6000 ve 7000 iterasyona sahip çalışmalarda verim açısından değişiklik olmadığı, 9000 iterasyona sahip çalışmada ise 8000 iterasyona sahip çalışmaya göre verimlilik açısından yaklaşık olarak %1 oranında artış olduğu görülmektedir.

Tablo 5.9 RRT* algoritmasında engel yarıçaplarının 0.2 metre olduğu simülasyon çalışmalarında gerekli düğüm sayıları ve noktaların verimlilik verileri

İterasyon Sayısı	Gerekli Düğüm Sayısı	Noktaların Verimliliği
500	70	0.140
1000	166	0.166
2000	373	0.187
3000	660	0.220
4000	873	0.218
5000	1116	0.223
6000	1238	0.206
7000	1440	0.206
8000	1586	0.198
9000	1833	0.200

Optimum yol uzunluğu ve noktaların verimliliği hesaba katıldığında en uygun yol planlamasının 9000 iterasyonun kullanıldığı simülasyon çalışmasında gerçekleştiği saptanmıştır.

Engellerin şişkinlik yarıçapının 0.2 metre olduğu durumda yapılan işlemlerin aynısı engellerin 0.3 metre olduğu durumda da gerçekleştirilmiştir. Engellerin şişkinlik yarıçapının arttırılması sonucunda ortalama yol uzunluğunun azaldığı, fakat optimum yol uzunluğuna ulaşıldığı 8000 ve 9000 iterasyona sahip simülasyon çalışmalarında yol uzunluğu bakımından herhangi bir değişiklik meydana gelmesini sağlamamıştır. Tablo 5.10'de engellerin şişkinlik yarıçapının 0.3 metre olduğu duruma ait yol uzunluk değerleri gösterilmiştir.

Tablo 5.10 RRT* algoritmasında engel yarıçapının 0.3 metre olduğu simülasyon çalışmalarında toplam yol uzunluk değerleri

İterasyon Sayısı	Toplam Yol Uzunluğu (metre)
500	2.015
1000	1.650
2000	1.726
3000	1.735
4000	1.801
5000	1.801
6000	1.501
7000	1.501
8000	1.444
9000	1.444

Engellerin şişkinlik yarıçapının 0.2 metre olduğu ve 1000 iterasyona sahip simülasyon çalışmasında 500 iterasyonun kullanıldığı durumdaki yol uzunluğundan kısa; fakat 2000 iterasyonun kullandığı durumdan da kısa olduğu görülmüştür. Yol uzunluğu dikkate alındığında iterasyon sayısının 8000 ve 9000 olduğu durumda optimum değerlere ulaşılmıştır. Tablo 5.11'de simülasyon sonucu ulaşılan noktaların verimlilik değerleri gösterilmiştir.

Tablo 5.11 RRT* algoritmasında engel yarıçaplarının 0.2 metre olduğu simülasyon çalışmalarında gerekli düğüm sayıları ve noktaların verimlilik verileri

İterasyon Sayısı	Gerekli Düğüm Sayısı	Noktaların Verimliliği
500	71	0.142
1000	201	0.201
2000	345	0.173
3000	498	0.166
4000	658	0.165
5000	901	0.180
6000	1465	0.244
7000	1194	0.171
8000	1603	0.200
9000	1568	0.174

1.444 metre yol uzunluğuna ulaşıldığı 8000 iterasyona sahip simülasyon sonucunda noktalarının maksimum verimliliğinin 0.200 olduğu, aynı yol uzunluğu sonucuna ulaşıldığı 9000 iterasyona sahip simülasyon sonucunda 0.174 değerine ulaşıldığı görülmüştür. Engellerin şişkinlik yarıçapının 0.2 metre olduğu durumla karşılaştırıldığında verimlilik ortalamasının düştüğü görülmüştür.

PRM ve RRT* algoritmalarının kullanıldığı simülasyon sonuçları birlikte değerlendirildiğinde ve engellerin şişkinlik yarıçapının 0.3 metre olduğu durumlar dikkate alındığında nokta verimliliğinin 0.4 değerinden düşük ve en kısa yol mesafesinin 1.444 metre olarak bulunduğu tespit edilmiştir.

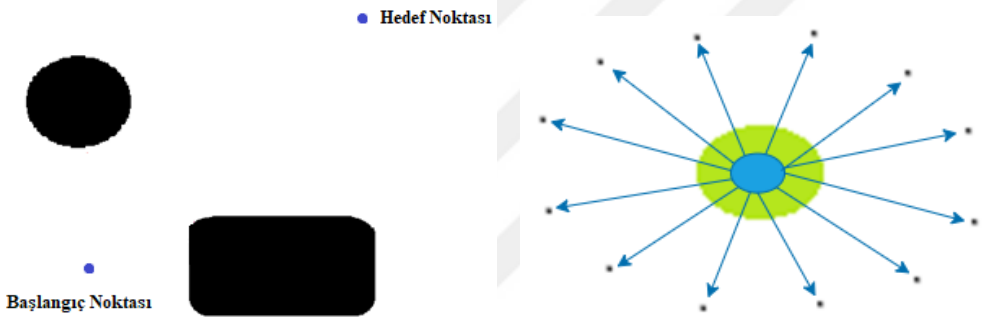
5.3 Dairesel Düğüm Algoritması

PRM ve RRT* algoritmalarından yola çıkarak hem engellerden kaçınabilen hem de hedefe daha kısa yoldan ulaşabilecek yeni bir algoritma oluşturulmaya çalışılmıştır. Oluşturulan dairesel düğüm algoritmasında hem PRM hem de RRT* algoritmalarının çalışma yöntemleri sentezlenerek kullanılmıştır. PRM ve RRT* algoritmalarında olduğu gibi harita üzerine mobil robotun hareket edebileceği düğüm noktaları yerleştirilmiştir. Bu noktaların diğer algoritmalarda olduğu gibi engellerle bir teması bulunmamaktadır. Diğer algoritmalarda robotun olası bir kaza ihtimaline karşı engele şişkinlik yarıçapı değeri verilerek boyutlarının arttırılması sağlanmıştı; fakat bu kullanılan algoritmada bu işlem engel üzerinde değil gidilecek olan düğüm noktası üzerinde gerçekleştirilmiştir. Eğer mobil robotun hareketini gerçekleştireceği düğüm noktası ile noktanın etrafındaki engel arasındaki mesafe mobil robotun boyutlarına uygun değilse, robotun bu noktaya olabilecek olan olası bir yönelme hareketi engellenmektedir. Bu engelleme sayesinde robotun hareket edebileceği noktalar kümesi belirlenmiş olmaktadır. Dairesel düğüm algoritmanın çalışma prensibi şu şekildedir:

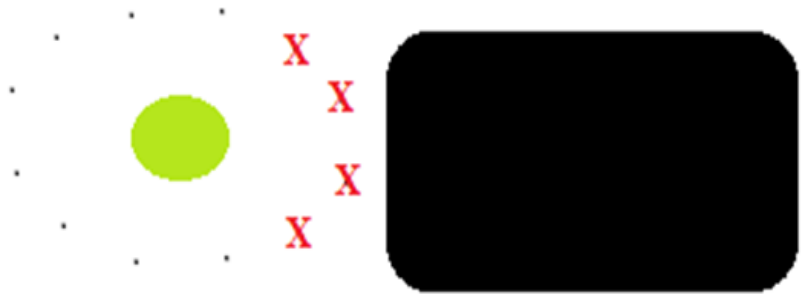
1. Robotun mevcut konumu etrafında belirlenen yarıçap değerinde ve belirlenen düğüm sayısı değerinde noktalar oluşturulur.
2. Oluşturulan bu noktalar arasında etrafındaki herhangi bir engelle arasındaki mesafesi belirlenen limit değerinden küçük olan noktalar silinir.
3. Geri kalan noktalar arasından hedefe en yakın olan noktaya doğru hareket edilir.
4. İlk hedefine ulaştıktan sonra 1. adımdan tekrar başlayarak belirlenen hedef noktaya ulaşana kadar veya hedef nokta ile ulaşılan nokta arasındaki mesafe belirlenen limit değerinin altında olana kadar bu işlemler devam eder.

Düğüm noktalarının atanması için gerek duyulan yarıçap değeri, düğüm noktası sayısı, engelle nokta arasında izin verilen mesafe değeri ve algoritmanın devam edip etmeyeceği kararı için hedef nokta ile ulaşılmış nokta arasında mesafe değeri mobil robot ve haritaya göre değişebilmektedir. Bu yöntem PRM ve RRT* algoritmalarında da olduğu gibi hedef nokta belirlenir, hedefe gidilir ve yeni hedef belirlenir şeklinde çalışmaktadır.

5 hertz tarama frekans değerinde çalışan RPLidar, her 0.2 saniyede bir döngüsünü tamamlamaktadır. Bulunduğu konuma göre kendisinden 6 metre uzaklıktaki engelleri tespit edebilen ışıklı mesafe algılayıcı ile saniyede 8000 adet örneklem sağlanabilmektedir. Bu sayede mobil robotun hareketi sırasında engellerin güncellemesi yapılarak algoritmanın dinamik engellere karşı verebileceği tepki ön görülebilir. Mobil robot dairesel düğüm algoritması kullanılarak mevcut konumundan harekete başlayıp, hedefe en yakın düğüm noktasına vardıktan 0.1 saniye sonra yeni noktalar kümesi oluşturmaktadır. Ardından oluşturulan düğüm noktalarından hedefe en yakın olanı seçerek 0.1 m/s araç hızıyla o noktaya doğru hareket eder. Robotun her 0.1 metre hareketinde lidarın döngüyü tamamlamasıyla engellerin konumları tekrardan yenilenir. Şekil 5.7'de algoritmanın çalışma prensibi gösterilmiştir.



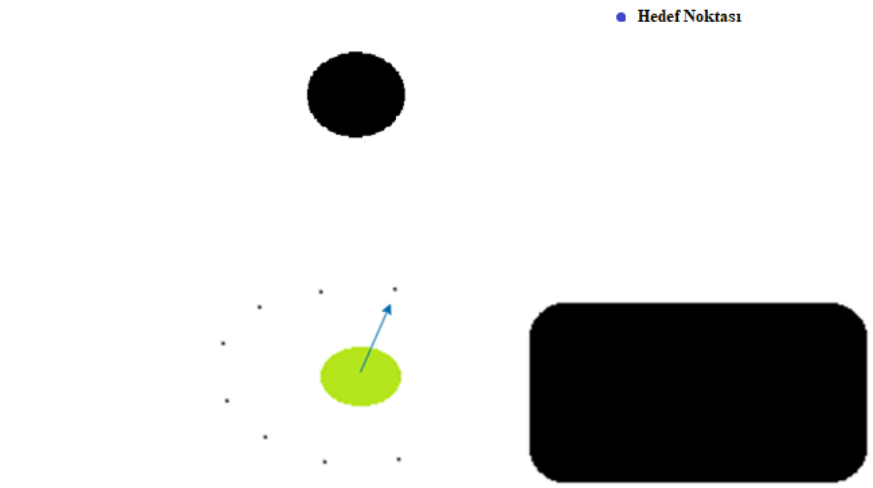
a) b)



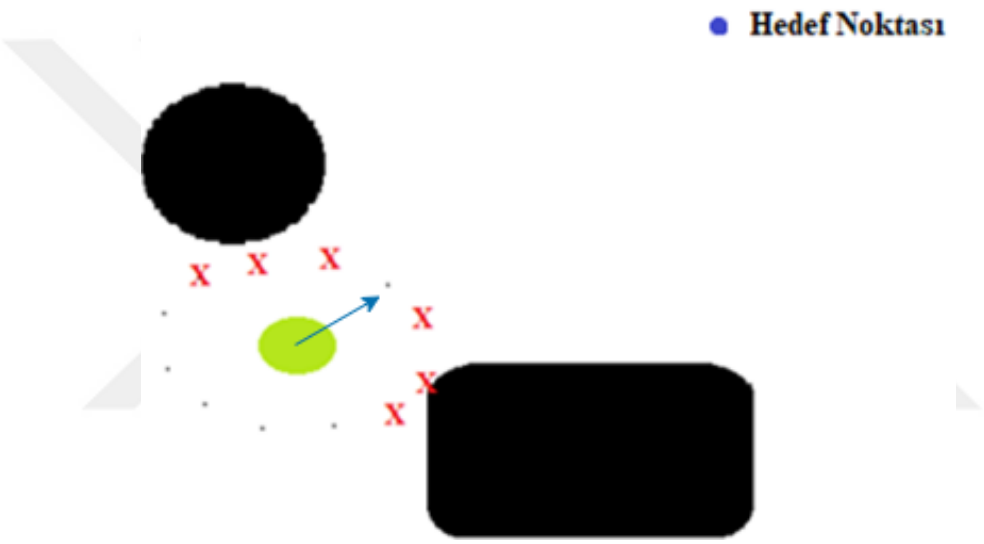
c)



d)



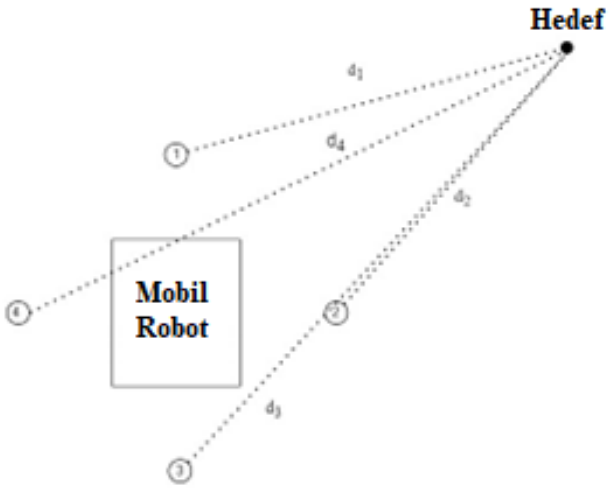
e)



f)

Şekil 5.7 a) Başlangıç ve bitiş noktalarının belirlenmesi b) Düğüm noktalarının atanması, c) Engele yakın olan noktaların tespit edilmesi d) Tespit edilen noktaların silinmesi, e) Kalan noktalar arasından hedefe en yakın olanın bulunması ve hareket edilmesi, f) Hedefe yaklaşana kadar aynı adımların tekrarlanması

Şekil 5.8'de mobil robot etrafında oluşturulacak olan düğüm noktalarının hedef olan uzaklıkları gösterilmiştir. d_1, d_2, d_3 ve d_4 uzunlukları hedef ile düğüm noktalarının x ve y koordinat bilgileri kullanılarak bulunur. En küçük d değerinin olduğu noktaya giderek hedefe yaklaşılır.



Şekil 5.8 Hedef düğüm noktasının tespiti

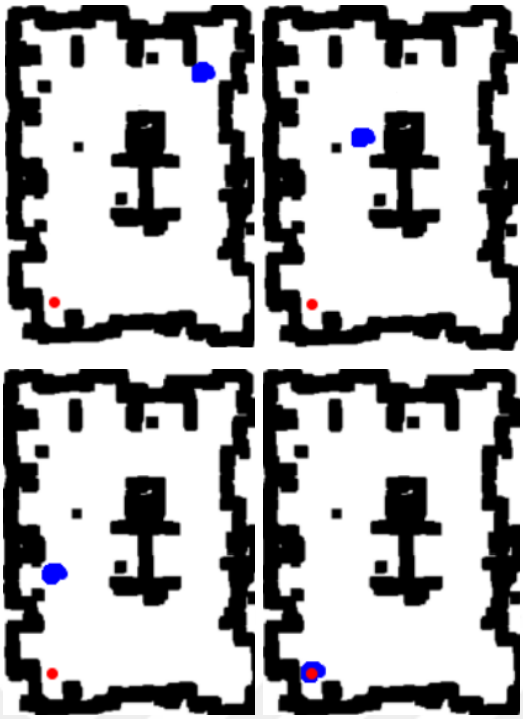
5.3.1 Dairesel Düğüm Algoritması Simülasyon Çalışmaları

Tez çalışması için oluşturulan mobil robot, ışıklı mesafe ölçer algılayıcısı ile manuel sürüş gerçekleştirilerek Şekil 5.9'da gösterildiği gibi laboratuvar ortamının haritalaması gerçekleştirilmiştir.



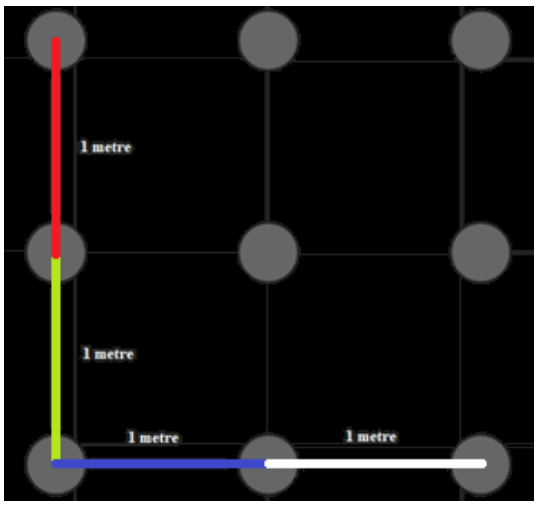
Şekil 5.9 Laboratuvar ortamının simülasyon çalışmaları için haritalanması

Yerleştirilen bu harita üzerinde başlangıç noktasında noktalar kümesi oluşturularak kırmızı noktaya ulaşmak istenmektedir. Mavi renkte olan dairesel noktalar her aşamada hedefe en yakın olan noktayı seçer ve o noktanın etrafında yeni mavi noktalar oluşturarak hedefe ulaşmaya çalışır.



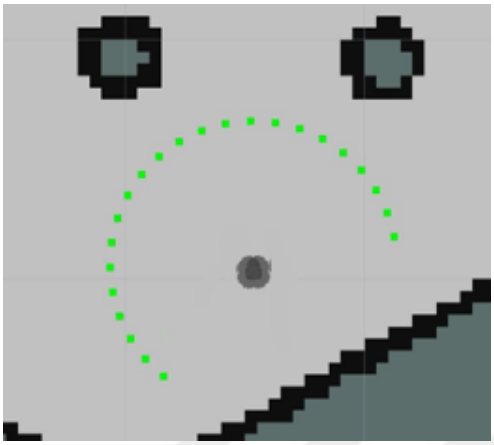
Şekil 5.10 Laboratuvar ortamında algoritmanın denemeleri

Şekil 5.10'daki haritada herhangi dinamik bir engel bulunmamaktadır. Bu durum PRM ve RRT* algoritmalarının da kullanıldığı statik durumdaki haritalar için kullanılabilir durumdadır. Python kodlaması ile oluşturulmuş algoritmanın ana temasının çalışabilir durumda olmasının ardından Robot Operasyon Sistemi (ROS) içeresinde PRM ve RRT* algoritmalarının kullanıldığı harita üzerinde simülasyon denemeleri gerçekleştirilmiştir. Şekil 5.11'de simülasyon çalışmalarının yapıldığı harita gösterilmiştir.



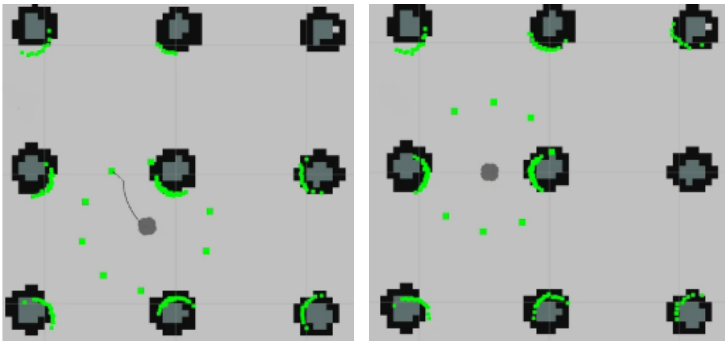
Şekil 5.11 ROS simülasyon çalışmalarının gerçekleştirildiği harita

İlk olarak etrafı engelle çevrili olan bir noktada dairesel düğüm noktalarının denemesi yapılmıştır. Engelle arasındaki mesafenin belirli bir limit değerinde az olduğu durumlarda noktaların silindiği Şekil 5.12'de görülmektedir.



Şekil 5.12 ROS simülasyonunda dairesel düğüm noktaların oluşturulması

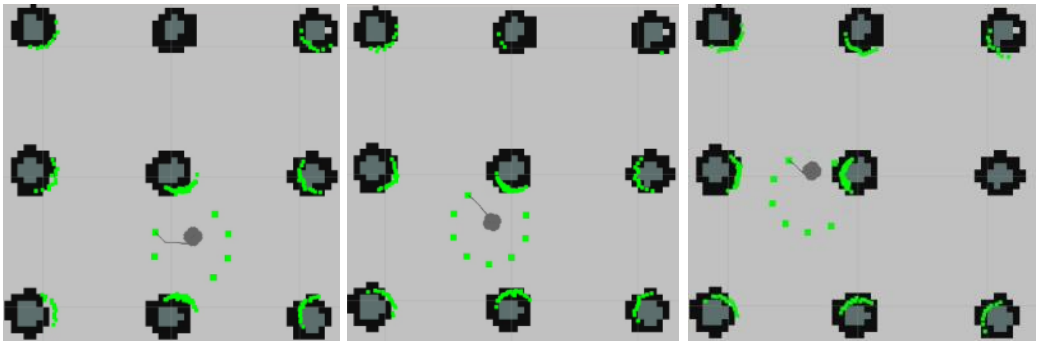
Dairesel düğüm algoritmasında bulunan düğüm noktaları oluşturma yarıçapı, düğüm nokta sayısı ve engelle ile düğüm nokta arasında izin verilen mesafe parametreleri sonuçları doğrudan etkilemektedir. Şekil 5.13'te her bir adımda 10 adet düğüm noktasını robot merkezinden 0.5 metre uzaklıkta oluşturmakta olan bir yol planlama örneği gösterilmiştir.



Şekil 5.13 Robot merkezinden 0.5 metre uzaklıkta 10 düğüm noktasına sahip örnek yol planlaması

Mobil robotun üzerinde bulunan ışıklı mesafe ölçer sayesinde engellerin robotun açısından görünen bölgeleri ve oluşturulan düğüm noktaları yeşil renkli noktalar kümesiyle gösterilmektedir. Engel noktalarının düğüm noktalarıyla çakışması veya aralarındaki mesafenin limit değerinin altında olması durumunda bu noktalar silinmektedir. Kalan noktalar arasında hedefe en yakın nokta hedef nokta olarak belirlenir ve hareket o noktaya doğru yapılır. Düğüm noktaların robot merkeziyle

olan uzaklık değerinin 0.3 metreye düşürülmesiyle Şekil 5.14'teki planlamalar meydana gelmektedir.



Şekil 5.14 Robot merkezinden 0.3 metre uzaklıkta 10 düğüm noktasına sahip örnek yol planlaması

Robot merkeziyle oluşturulacak olan düğüm noktaları arasındaki mesafenin arttırılması, robotun merkezden uzakta bir işlem süreci geçirmesine sebebiyet vermektedir. Bu parametrik değerin büyük olması durumunda robotun çok yakınlarında olan dinamik engeller için sorun teşkil edebilmekte ve hedefe ulaşma noktasında tam koordinat değerine ulaşılamama sorununu ortaya çıkarabilmektedir. Bir diğer taraftan, dairesel düğüm noktalarının oluşturulduğu yarıçap değerinin azalması işlem süresinin uzamasına, oluşturulan toplam düğüm nokta sayısının artmasına yol açtığı gözlemlenmiştir. Optimum yol planlamasının ortaya çıkarılabilmesi için tüm parametrik değerlerin analiz edilerek en uygun değerlerin bulunması gerekmektedir. Uygun parametrik değerler mobil robotun yapısına ve çalışılacak olan ortamın durumuna göre değişiklik gösterebilmektedir.

Örnek yol planlamalarının ardından parametrik değerlerinde incelenebileceği 6 farklı durumun incelendiği simülasyon çalışmaları gerçekleştirilmiş ve sonuçları değerlendirilmiştir. Mobil robotun başlangıç noktası (1.5, 0) ve bitiş noktası (0.5, 1) koordinatları olacak şekilde seçilmiştir. Her bir durumda oluşturulan yol güzergâhlarının toplam uzunluğu ve kullanılabilecek düğüm noktalarının oluşturulan tüm düğüm noktalarına göre oranı kullanarak verimlilik değerleri hesaplanmıştır. Tablo 5.12'da dairesel düğüm algoritması için simülasyon durumları gösterilmiştir.

Tablo 5.12 Dairesel düğüm algoritması için simülasyon durumları

Test Numarası	Düğüm Noktalar Yarıçapı	Engel Nokta Arası Mesafe Limiti	Düğüm Noktası Sayısı
Test 1	0.3	0.2	5,10,20,30,40,50,60,70,80,90,100
Test 2	0.3	0.3	5,10,20,30,40,50,60,70,80,90,100
Test 3	0.4	0.2	5,10,20,30,40,50,60,70,80,90,100
Test 4	0.4	0.3	5,10,20,30,40,50,60,70,80,90,100
Test 5	0.5	0.2	5,10,20,30,40,50,60,70,80,90,100
Test 6	0.5	0.3	5,10,20,30,40,50,60,70,80,90,100

1 numaralı test çalışmasında dairesel düğüm noktaları ile robot merkezi arasındaki mesafe sabit olarak 0.3 metre, noktalar ile engel arasındaki mesafe limitinin sabit olarak 0.2 metre ve düğüm nokta sayılarının 5, 10, 20, 30, 40, 50, 60, 70, 80, 90 ve 100 olduğu 11 farklı yol planlaması gerçekleştirilmiştir. Elde edilen sonuçlar Tablo 5.13'te gösterilmiştir.

Tablo 5.13 Test 1 sonucu elde edilen yol uzunluğu ve düğüm noktalarının verimlilik sonuçları

Simülasyon Numarası	Düğüm Noktası Sayısı	Yol Uzunluğu	Düğüm Noktası Verimliliği
Simülasyon 1	5	1.477	0.933
Simülasyon 2	10	1.587	0.820
Simülasyon 3	20	1.463	0.860
Simülasyon 4	30	1.440	0.560
Simülasyon 5	40	1.673	0.892

Tablo 5.13 Test 1 sonucu elde edilen yol uzunluğu ve düğüm noktalarının verimlilik sonuçları (devamı)

Simülasyon 6	50	1.449	0.880
Simülasyon 7	60	1.514	0.907
Simülasyon 8	70	1.520	0.911
Simülasyon 9	80	1.423	0.760
Simülasyon 10	90	1.457	0.938
Simülasyon 11	100	1.514	0.938

Test 1 sonucunda ortalama yol mesafelerinin ortalama değeri 1.502 metre olduğu görülmüştür. En kısa yol uzunluğuna 1.412 metre ile 80 düğüm noktasının kullanıldığı 9 numaralı simülasyon denemesinde ulaşılmıştır. Düğüm noktaları 90 düğüm noktasının kullanıldığı simülasyon 10 ve 100 düğüm noktasının kullanıldığı simülasyon 11 çalışmalarında 0.938 değeri ile en yüksek verimliliğe ulaşmıştır.

Test 2 çalışmasında dairesel düğüm noktaları ile robot merkezi arasındaki mesafe sabit olarak 0.3 metre, noktalar ile engel arasındaki mesafe limitinin sabit olarak 0.3 metre ve düğüm nokta sayılarının 5, 10, 20, 30, 40, 50, 60, 70, 80, 90 ve 100 olduğu 11 farklı yol planlaması gerçekleştirilmiştir. Elde edilen sonuçlar Tablo 5.14'te gösterilmiştir.

Tablo 5.14 Test 2 sonucu elde edilen yol uzunluğu ve düğüm noktalarının verimlilik sonuçları

Simülasyon Numarası	**Düğüm Noktası Sayısı**	**Yol Uzunluğu**	**Düğüm Noktası Verimliliği**
Simülasyon 1	5	1.749	0.567
Simülasyon 2	10	1.568	0.700
Simülasyon 3	20	1.610	0.670
Simülasyon 4	30	1.529	0.667

Tablo 5.14 Test 2 sonucu elde edilen yol uzunluğu ve düğüm noktalarının verimlilik sonuçları (devamı)

Simülasyon 5	40	1.579	0.554
Simülasyon 6	50	1.598	0.620
Simülasyon 7	60	2.030	0.693
Simülasyon 8	70	1.941	0.779
Simülasyon 9	80	1.870	0.815
Simülasyon 10	90	1.448	0.833
Simülasyon 11	100	1.516	0.860

Test 2 sonucunda ortalama yol mesafelerinin ortalama değeri 1.676 metre olduğu ve noktalar ile engel arasındaki mesafe limitinin 0.2'den 0.3'e yükseltilmesiyle oluşturulan toplam yol mesafesinin ortalama olarak %12 oranında arttığı görülmüştür. En kısa yol uzunluğuna 1.448 metre ile 90 düğüm noktasının kullanıldığı 9 numaralı simülasyon denemesinde ulaşılmıştır.

3 numaralı test çalışmasında dairesel düğüm noktaları ile robot merkezi arasındaki mesafe sabit olarak 0.4 metre, noktalar ile engel arasındaki mesafe limitinin sabit olarak 0.2 metre ve düğüm nokta sayılarının 5, 10, 20, 30, 40, 50, 60, 70, 80, 90 ve 100 olduğu 11 farklı yol planlaması gerçekleştirilmiştir. Elde edilen sonuçlar Tablo 5.15'te gösterilmiştir.

Tablo 5.15 Test 3 sonucu elde edilen yol uzunluğu ve düğüm noktalarının verimlilik sonuçları

Simülasyon Numarası	**Düğüm Noktası Sayısı**	**Yol Uzunluğu**	**Düğüm Noktası Verimliliği**
Simülasyon 1	5	1.920	0.633
Simülasyon 2	10	1.791	0.780
Simülasyon 3	20	3.543	0.800

Tablo 5.15 Test 3 sonucu elde edilen yol uzunluğu ve düğüm noktalarının verimlilik sonuçları (devamı)

Simülasyon 4	30	2.354	0.856
Simülasyon 5	40	1.996	0.717
Simülasyon 6	50	3.507	0.829
Simülasyon 7	60	2.145	0.923
Simülasyon 8	70	3.929	0.897
Simülasyon 9	80	2.327	0.913
Simülasyon 10	90	1.498	0.756
Simülasyon 11	100	2.862	0.939

Test 3 sonucunda yol mesafelerinin ortalama değeri 2.533 metre değerine ulaştığı görülmüştür. Test 1 ile karşılaştırıldığında yol mesafesinde ortalama olarak %69 oranında artış olmuştur. En kısa yol uzunluğuna 1.498 metre ile 90 düğüm noktasının kullanıldığı 9 numaralı simülasyon denemesinde ulaşılmıştır. Düğüm noktaları 100 düğüm noktasının kullanıldığı simülasyon 11 çalışmalarında 0.939 değeri ile en yüksek verimliliğe ulaşmıştır.

4 numaralı test çalışmasında dairesel düğüm noktaları ile robot merkezi arasındaki mesafe sabit olarak 0.4 metre, noktalar ile engel arasındaki mesafe limitinin sabit olarak 0.3 metre ve düğüm nokta sayılarının 5, 10, 20, 30, 40, 50, 60, 70, 80, 90 ve 100 olduğu 11 farklı yol planlaması gerçekleştirilmiştir. Elde edilen sonuçlar Tablo 5.17'de gösterilmiştir.

Tablo 5.16 Test 4 sonucu elde edilen yol uzunluğu ve düğüm noktalarının verimlilik sonuçları

Simülasyon Numarası	Düğüm Noktası Sayısı	Yol Uzunluğu	Düğüm Noktası Verimliliği
Simülasyon 1	5	2.417	0.600
Simülasyon 2	10	1.620	0.520
Simülasyon 3	20	3.849	0.646
Simülasyon 4	30	4.710	0.628
Simülasyon 5	40	7.730	0.665
Simülasyon 6	50	2.386	0.700
Simülasyon 7	60	4.040	0.738
Simülasyon 8	70	2.081	0.771
Simülasyon 9	80	3.204	0.820
Simülasyon 10	90	3.621	0.820
Simülasyon 11	100	3.303	0.824

Test 4 sonucunda yol mesafelerinin ortalama değeri 3.542 metre değerine ulaştığı görülmüştür. Test 4 simülasyonlarında 8'den fazla düğüm noktasına gidildiği gözlemlenmiştir. En kısa yol uzunluğuna 1.620 metre ile 10 düğüm noktasının kullanıldığı 2 numaralı simülasyon denemesinde ulaşılmıştır.

5 numaralı test çalışmasında dairesel düğüm noktaları ile robot merkezi arasındaki mesafe sabit olarak 0.5 metre, noktalar ile engel arasındaki mesafe limitinin sabit olarak 0.2 metre ve düğüm nokta sayılarının 5, 10, 20, 30, 40, 50, 60, 70, 80, 90 ve 100 olduğu 11 farklı yol planlaması gerçekleştirilmiştir. Elde edilen sonuçlar Tablo 5.18'de gösterilmiştir.

Tablo 5.17 Test 5 sonucu elde edilen yol uzunluğu ve düğüm noktalarının verimlilik sonuçları

Simülasyon Numarası	Düğüm Noktası Sayısı	Yol Uzunluğu	Düğüm Noktası Verimliliği
Simülasyon 1	5	4.264	0.817
Simülasyon 2	10	1.499	0.625
Simülasyon 3	20	1.525	0.725
Simülasyon 4	30	1.460	0.692
Simülasyon 5	40	1.415	0.688
Simülasyon 6	50	1.500	0.715
Simülasyon 7	60	1.505	0.721
Simülasyon 8	70	1.470	0.725
Simülasyon 9	80	1.443	0.719
Simülasyon 10	90	1.466	0.742
Simülasyon 11	100	1.515	0.753

Test 5 simülasyonlarında ortalama yol mesafesi bakımından 1.733 metre değerine ulaşıldığı görülmüştür. Simülasyon 1 çalışmasın dışında diğer yol uzunluk değerlerinin iyi olduğu söylenebilir. En kısa yol uzunluğuna 1.415 metre ile 40 düğüm noktasının kullanıldığı 5 numaralı simülasyon denemesinde ulaşılmıştır.

6 numaralı test çalışmasında dairesel düğüm noktaları ile robot merkezi arasındaki mesafe sabit olarak 0.5 metre, noktalar ile engel arasındaki mesafe limitinin sabit olarak 0.3 metre ve düğüm nokta sayılarının 5, 10, 20, 30, 40, 50, 60, 70, 80, 90 ve 100 olduğu 11 farklı yol planlaması gerçekleştirilmiştir. Elde edilen sonuçlar Tablo 5.19'da gösterilmiştir.

Tablo 5.18 Test 6 sonucu elde edilen yol uzunluğu ve düğüm noktalarının verimlilik sonuçları

Simülasyon Numarası	Düğüm Noktası Sayısı	Yol Uzunluğu	Düğüm Noktası Verimliliği
Simülasyon 1	5	6.840	0.560
Simülasyon 2	10	1.490	0.733
Simülasyon 3	20	1.534	0.700
Simülasyon 4	30	1.477	0.711
Simülasyon 5	40	1.446	0.700
Simülasyon 6	50	1.499	0.720
Simülasyon 7	60	1.581	0.722
Simülasyon 8	70	1.430	0.781
Simülasyon 9	80	1.488	0.788
Simülasyon 10	90	1.483	0.826
Simülasyon 11	100	1.527	0.843

Test 6 sonucunda ortalama yol mesafelerinin ortalama değeri 1.981 metre değerine ulaştığı görülmüştür. En kısa yol uzunluğuna 1.430 metre ile 70 düğüm noktasının kullanıldığı 8 numaralı simülasyon denemesinde ulaşılmıştır.

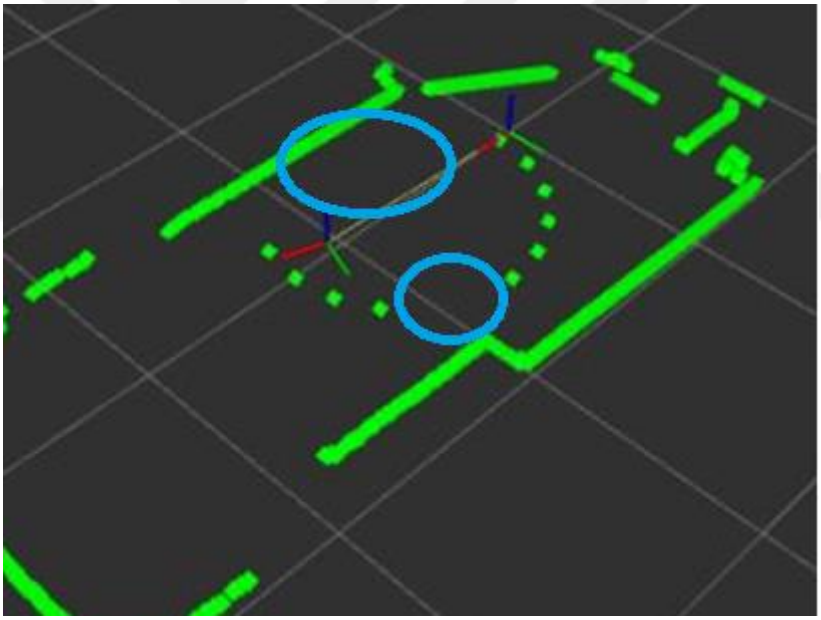
Genel olarak sonuçlar değerlendirildiğinde dairesel düğüm noktaları ile robot merkezi arasındaki mesafenin 0.5 metre olduğu test çalışmalarında 1.415 metre ve 1.43 metre değerleri ile en kısa yol uzunluğuna ulaşıldığı görülmüştür.

Simülasyon çalışmalarında elde edilen düğüm verimliliği sonuçlarının PRM ve RRT* algoritmalarına göre daha başarılı olduğu ve konumlara göre değişebilecek olan planlamış yolların uzunluk değerleri göz önüne alındığında uygun parametrik değerlerin seçilmesiyle daha başarılı sonuçlara ulaşılması beklenmektedir.

5.3.2 Dairesel Düğüm Algoritması Deneysel Çalışmaları

Simülasyon çalışmalarının ardından tasarlanan algoritmanın gerçek hayatta çalışabilir olup olmadığının kontrol edilmesi için yazılan Python kodları ROS içerisine entegre edilmiştir ve navigasyon paketlerinin açılmasının ardından çalıştırılmıştır.

RVIZ üzerinden haritanın açılmadığı durumda sadece ışıklı mesafe algılayıcı ile ortamda bulunan engellerin ve mobil robotun arasındaki mesafe verileri alınmış ve bu verilerin x ve y düzlemlerinde düzenlenmesinin ardından ortamın haritası oluşturulmuştur. Ardından mobil robotun etrafında belirlenen uzaklıkta ve sayıda noktalar oluşturulmuştur. Hedefle oluşturulan noktalar arasındaki mesafenin belli bir limitin altında olması durumunda bu noktaların silindiği mavi dairesel şekillerle gösterilmiştir. Hedef noktanın seçilmesiyle birlikte Şekil 5.15'te robotun hedefe doğru yöneldiği görülmektedir.

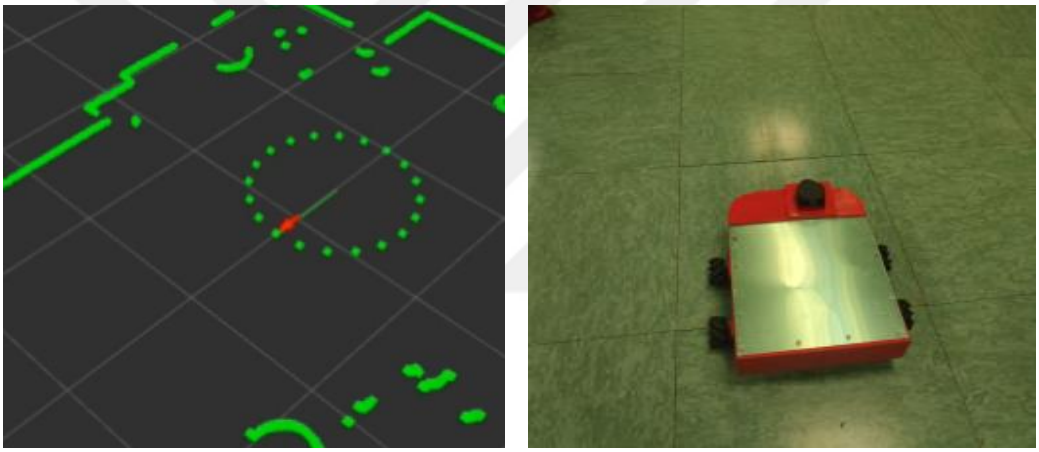


Şekil 5.15 Deneysel çalışmalarda uygun olmayan düğüm noktalarının silinmesi

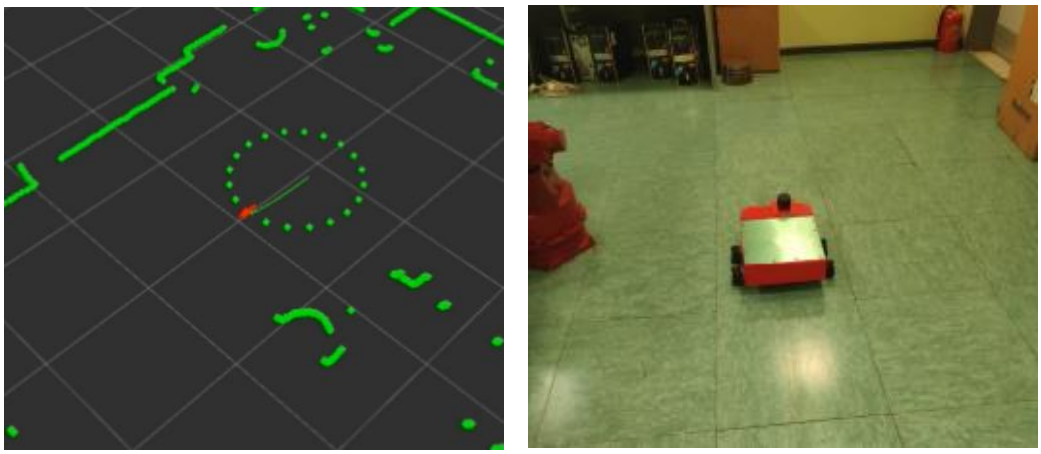
Algoritma üzerinde ilk yapılan denemelerde algoritmanın istenildiği şekilde çalıştığının görülmesinin ardından bir laboratuvar ortamında test çalışmaları gerçekleştirilmiştir. Aynı ortam içerisinde birbirinden farklı 3 durumda robotun tepkileri gözlemlenmiştir.

1. Herhangi bir engelin olmadığı durum
2. Statik engelin olduğu durum
3. Statik ve dinamik engelin olduğu durum

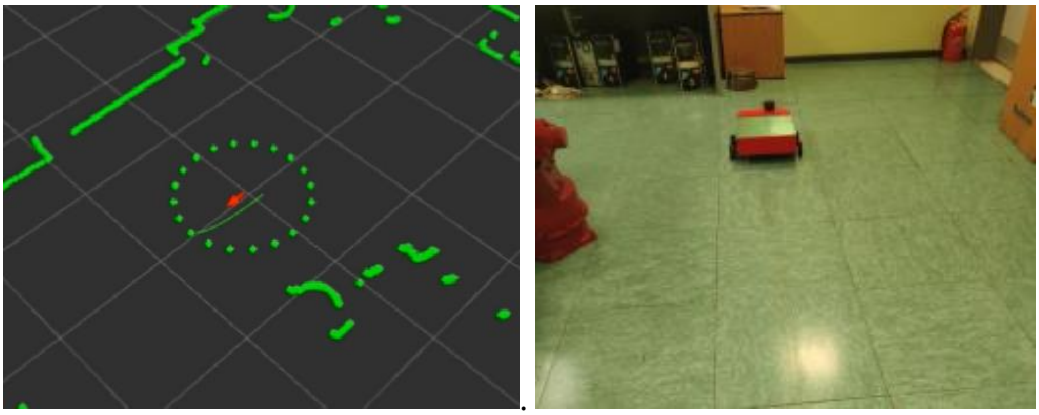
Mobil robotun önünde herhangi bir engelin olmadığı durumda hedef noktanın robot ekseninde herhangi bir nokta seçilmesi durumunda, robotun düz bir hareketle en kısa yolu bulacağı ön görülmektedir. Öncelikle robotla engel arasındaki mesafeler kullanılarak ortam haritalaması gerçekleştirilir. Ardından robotun etrafında düğüm noktaları oluşturulur ve oluşturulan bu hedef noktalar arasından engelle arasındaki mesafenin minimum olduğu düğüm noktası ara hedef nokta olarak belirlenerek bu noktaya hareket edilir. Robotun önünde bir engelin bulunmadığı durumda ve 0.6 metre yarıçap değerinde oluşturulan düğüm noktaları durumunda robotun hareketi Şekil 5.16'de sırasıyla gösterilmiştir.



a)



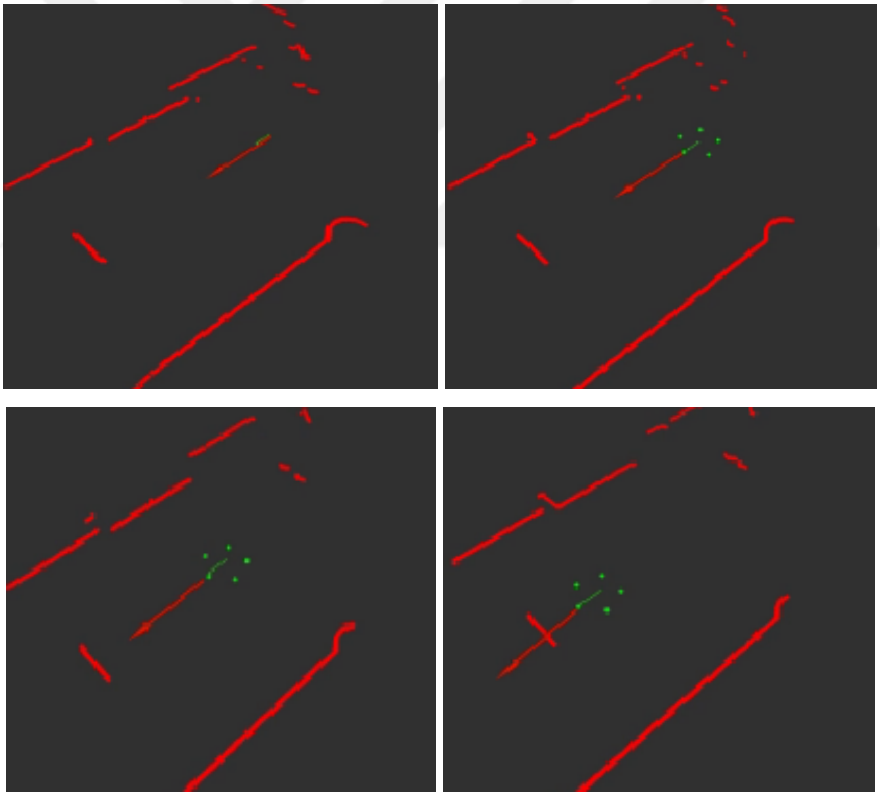
b)



c)

Şekil 5.16 a) Noktaların oluşturulması ve ara hedefe yönelim, b) Ara hedefe ulaşım, c) Yeni ara hedefin belirlenmesi

Şekil 5.17'de gerçekleştirilen test çalışmasında robotun merkeziyle düğüm noktaları arasındaki uzaklık 0.3 metre olacak şekilde ayarlanmıştır.

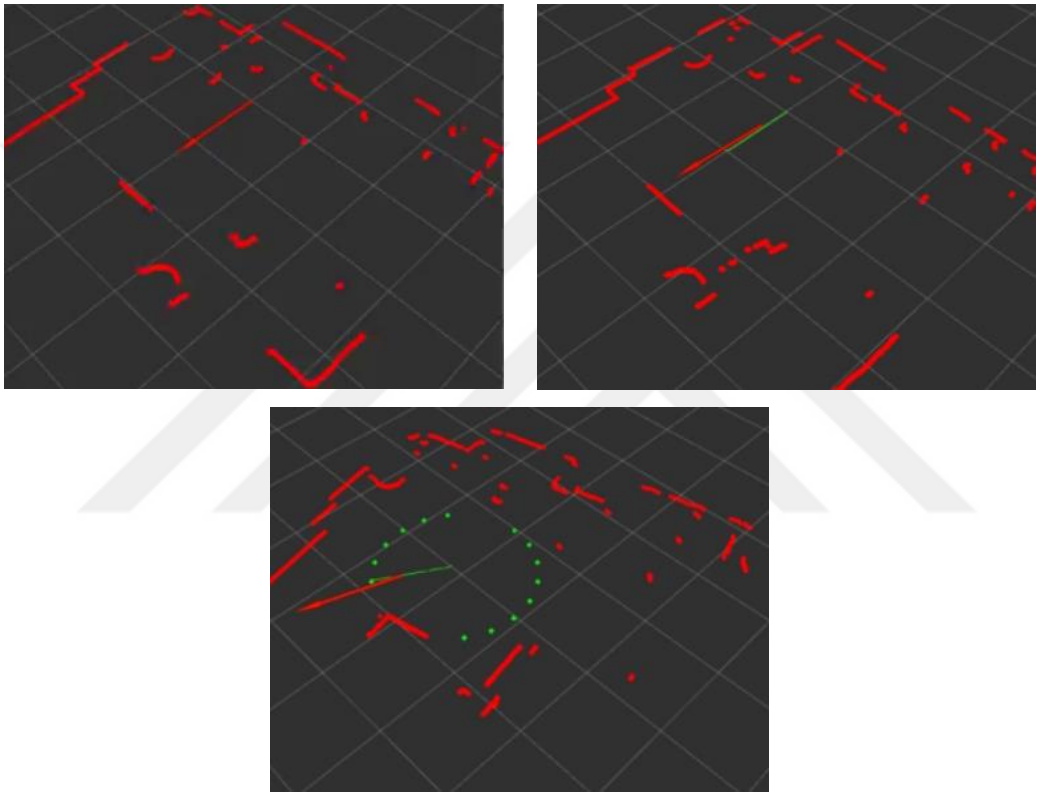


Şekil 5.17 Robotun merkeziyle düğüm noktaları arasındaki uzaklığın 0.3 metre olduğu test denemesi

Robotun merkeziyle düğüm noktaları arasındaki mesafenin ve düğüm sayısının değiştirilebilirliğinin deneysel olarak da görülmesini ardından aynı ortam içerisinde robotun önünde engelin de bulunduğu yeni bir durum için test yapılmıştır. Yol

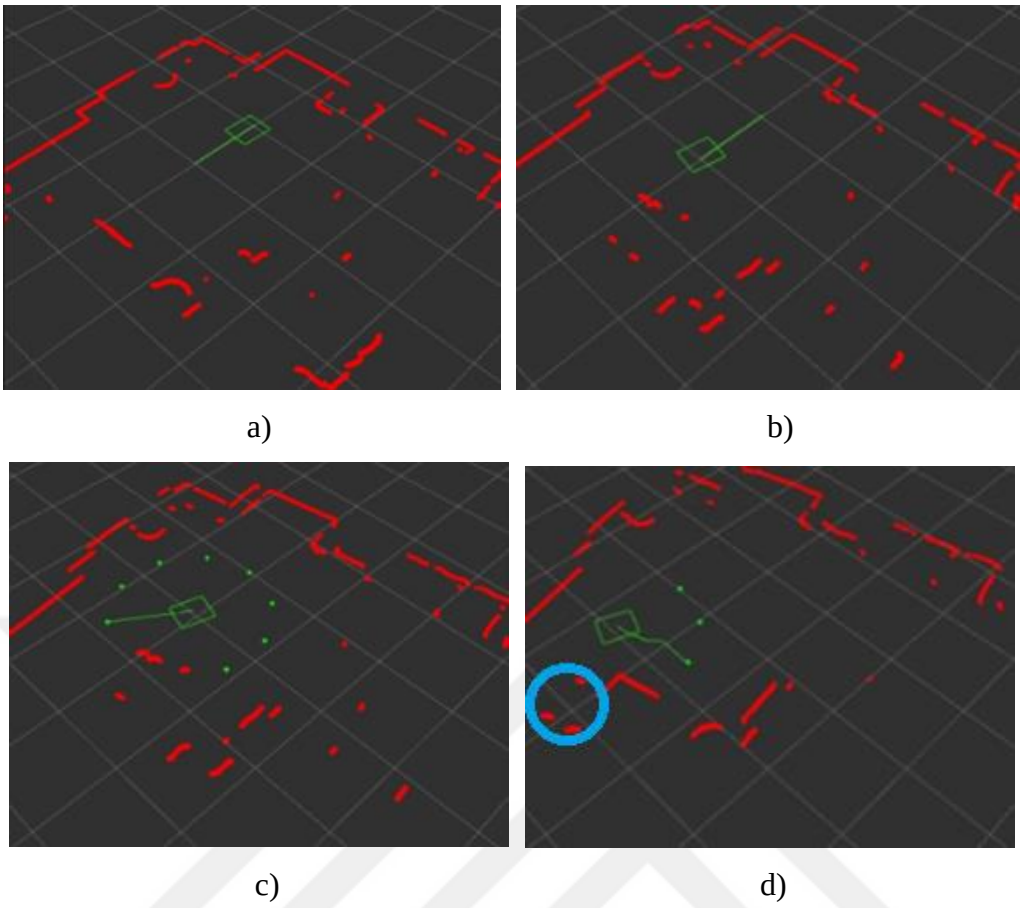
üzerinde bir engelin yerleştirilmesiyle birlikte robotun engelle temasının bulunmadığı noktaya hareket etmesi beklenmektedir.

Şekil 5.18'te gerçekleştirilen deneysel çalışmada robotun merkezinden 1 metre uzaklıkta 20 adet düğüm noktasının kullanılmıştır. Mobil robotun ilk aşamada düz gitmesinin ardından engelle karşılaşmıştır. Düğüm noktalarıyla engel arasında belirlenen limit değerin altına olan noktalar silinmiş ve kalan noktalar arasından hedefe en yakın noktaya doğru hareket edilmiştir. Bu deneysel çalışmada da tasarlanan algoritmanın engellerden kaçınabildiği sonucuna ulaşılmıştır.



Şekil 5.18 Engelin olduğu durumda deneysel çalışma

Birçok yol planlama algoritmaları engelden kaçınabilme özelliğine sahiptir. Halihazırda kullanılmakta olan algoritmalardan farklı olarak tasarlanan bu algoritma ile dinamik engellerden de kaçınma özelliği sayesinde ön plana çıkması ön görülmektedir. Bu sebeple aynı ortamda robotun ilk hareketinin beklenip, engelle karşılaşmasının ardından bir sonraki harekette gidebileceği noktaya canlı bir engelin yaklaşmasının ardından mobil robotun ve algoritmanın verdiği tepki Şekil 5.19'da gösterilmiştir.



Şekil 5.19 a) İlk ara hedefe doğru yönelme, b) İlk ara hedefe gelme, c) Engelden kaçınarak oluşturulan 2. ara hedefe yönelme, d) 3. Ara hedefe yönelim

Bu deneysel çalışmada belirli bir noktaya kadar robota dinamik engel etki etmemektedir. Mobil robotun statik olan engelden kaçınmasının ardından bir sonraki harekette robotun düz gitmesi ön görülerek robotun önüne mavi renkte görülmekte olan dinamik engelin yaklaşmasıyla birlikte, robotun bulunduğu son noktada düğüm noktaları oluşturduğu, dinamik engellerin yakınında herhangi bir noktanın oluşturulmadığı ve oluşturulan noktalar arasından engelin arkasındaki hedefe en yakın olan nokta seçilerek bu noktaya hareket edildiği görülmüştür. Mobil robot ile hedef nokta arasında herhangi bir engelin olmadığı, statik engelin olduğu ve hem statik hem de dinamik engelin olduğu birbirinden farklı durumlarda tasarlanan algoritmanın temelindeki yapının başarılı olduğu ve bu algoritma ile mobil robotun hem statik hem de dinamik engellerden kaçınarak bir yol planlaması oluşturabildiği deneysel çalışmalarla görülmüştür.

6
SONUÇLAR VE ÖNERİLER

Literatürde bulunan ve kullanılmakta olan klasik yöntem algoritmalarının içerisinde yer alan PRM ve RRT* algoritmaları, mobil robotun belirli iki nokta arasında engellerle temas etmeden hareket etmesini sağlayan ve yol planlamasını gerçekleştiren algoritmalardandır. Bu algoritmalar harita üzerinde düğüm noktalarını oluşturmasının ardından bu noktalara hareket edilmesiyle birlikte hedef noktaya gitmeyi amaçlamaktadır. Fakat bu algoritmalarının çalışabilmeleri için çalışma ortamının önceden belirlenmiş şekilde olması ve herhangi bir hareketli nesnenin olmaması gerekmektedir.

Tez çalışması kapsamında sabit engellerin bulunduğu bir harita üzerinden başlangıç ve bitiş noktalarının belirlenmiş; PRM ve RRT* algoritmaları kullanılarak noktalar arasında yol planlamaları oluşturulmuştur. PRM algoritmasının noktaları rastgele yerleştirilmesinden dolayı aynı parametrelerin kullanıldığı birden fazla durumda birbirinden farklı yollar oluşturduğu ve bu sebeple yol uzunluğunun değiştiği, haritanın tüm alanını kullanmasından dolayı ihtiyaç duyulmayan noktalar oluşturarak işlem süresini uzatabileceği ve noktalarının verimini düşürdüğü, nokta sayısının arttırılması sonrası daha optimum yolun kesin olarak bulunabilir düşüncesinin uygun olmadığı görülmüştür. RRT* algoritmasında PRM algoritmasında olduğu gibi ihtiyaç duyulmayan noktaların verimi düşürdüğü fakat iterasyon sayısının arttırılmasının daha optimum yollar oluşturabileceği görülmüştür. Bu iki algoritmanın daha önceden bilinen bir harita üzerinden düğüm nokta atamaları gerçekleştirmesi sebebiyle ortamda bulunan nesnelerin hareket etmesi veya yeni bir nesnenin mobil robotun önüne çıkması gibi dinamik durumlarda sorunlar oluşturduğu ön görülmüştür. Bu algoritmalarının çalışma mantıkları kullanılarak daha optimum yolun bulunabildiği, ortamdaki nesnelerin konumlarının değişmesi veya yeni nesnelerin ortama girmesi durumunda yeni bir yol planlaması gerçekleştirebilen bir algoritma tasarlanmıştır. Bu tasarlanan algoritma PRM algoritmasında olduğu gibi harita üzerine noktalar yerleştirmektedir; fakat PRM algoritmasından farklı olarak bu noktalar mobil robotun etrafında belirli mesafe uzaklıktadır. RRT algoritmasında olduğu gibi ilk noktaya hareketin ardından ikinci noktaya gidiş planlaması yapılmaktadır.

Tasarlanmış bir harita üzerinde mobil robotun başlangıç ve bitiş noktaları belirlenmiş ve bu noktalar arasında PRM, RRT* ve tasarlanan algoritmaların yol uzunluğu ve noktaların verimliliği konusunda simülasyon çalışmaları yapılmış ve sonuçlar karşılaştırılmıştır. Simülasyon çalışmaları MATLAB üzerinden gerçekleştirilen PRM algoritmasında, mobil robot ile engel arasında izin verilen mesafe limitinin 0.2 metre olduğu ve 1200 düğüm noktasının kullanıldığı durumda 1.423 metre yol uzunluğu ve %32.2 nokta verimliliğine ulaşıldığı, mobil robot ile engel arasında izin verilen mesafe değerinin 0.3 metre olduğu ve 1300 düğüm noktasının kullanıldığı durumda 1.490 metre yol uzunluğu ve %26.8 nokta verimliliğe ulaşıldığı görülmüştür. Aynı harita üzerinde RRT* algoritmasıyla gerçekleştirilen yol planlamalarında mobil robot ile engel arasında izin verilen mesafe limitinin 0.2 metre olduğu ve 8000 iterasyonun kullanıldığı durumda 1.444 metre yol uzunluğu ve %20 nokta verimliliğine ulaşıldığı, mobil robot ile engel arasında izin verilen mesafe limitinin 0.3 metreye yükseltilmesi sonrasında ve yine 8000 iterasyonun kullanıldığı durumda 1.444 metre yol uzunluğu ve %20 nokta verimliliğine ulaşıldığı görülmüştür. Tasarlanan dairesel düğüm noktası algoritmasında mobil robot ile engel arasında izin verilen mesafe limitinin 0.2 metre olduğu, her bir aşamada 80 adet düğüm noktasının kullanıldığı ve 5 aşamada hedef noktaya ulaşıldığı durumda %76 nokta verimliliği ile 1.423 metre yol uzunluğu sahip bir yol planlaması gerçekleştirilmiştir. Mobil robot ile engel arasındaki mesafe limitinin 0.3 metreye arttırıldığı durumda her aşamada 70 düğümün kullanıldığı, 3 aşamada hedefe ulaşıldığı %78.1 nokta verimliliğiyle en kısa yol uzunluğunun 1.430 metre olduğu yol planlaması gerçekleştirilmiştir. PRM, RRT* ve tasarlanan algoritma sonuçları karşılaştırıldığında PRM ve RRT* algoritmalarına göre işlem hacminin daha düşük, noktasal verimlilik açısından daha yüksek ve yol uzunluğu bakımından daha optimum yol sonuçlarına ulaşıldığı görülmüştür. Dairesel düğüm algoritmasında 1.415 metre ile en kısa yol uzunluğuna mobil robot ile engel arasında izin verilen mesafe limitinin 0.2 metre, dairesel düğüm yarıçapının 0.5 metre ve her aşamadaki 40 düğüm noktasının kullanıldığı durumda ulaşılmıştır. Bu durumdaki yol planlamasında noktaların verimlilik oranının %68.8 olduğu görülmüştür.

Tasarlanan algoritmanın PRM ve RRT* algoritmalarından farkları şu şekilde sıralanabilir:

1. Dinamik engellerden kaçınabilir.
2. PRM algoritmasındaki nokta oluşturma ve uygun yolu bulma fazları tek fazla gerçekleştirilir.
3. Başlangıç ve hedef noktaları arasındaki bölge harici herhangi bir işlem yapılmamaktadır.
4. Her aşamada oluşturulan noktalara maksimum 1 adet yol oluşturulur ve kısa olan tercih edilir.
5. Başlangıç ve bitiş noktalarının sabit olduğu ve dinamik bir engelin bulunmadığı durumlarda her çözümde farklı yollar bulmamaktadır.

Gelecek çalışmalar için algoritmanın hedefe yaklaştıkça düğüm noktalarının sıkılaşması hedef nokta koordinatlarına daha net bir şekilde ulaşılmasına ve nokta verimliliğinin artmasını sağlayabileceği düşünülmektedir. Deneysel çalışmalarda görüldüğü gibi mobil robotun engelle karşılaşması durumda tekrar noktalar oluşturarak engelden kaçma hareketi yaptığı görülmüştür. Dinamik engelle karşılaşması durumunda mobil robotun dinamik engelin durumuna göre hareket edilecek şekilde ayarlanması algoritmanın daha da gelişmiş versiyonu olabileceği düşünülmektedir. Algoritmanın bir Linux uygulaması içerisine yerleştirilmesiyle daha etkin bir şekilde kullanım kolaylığı sağlanabilir. Koordinata yönelik hareket veya önceden kaydedilmiş alana yönelik hareket şeklinde sınıflandırmayla birlikte, gerçek hayat içerisinde kullanım kolaylığı da sağlanabilir.

KAYNAKÇA

[1] K. J. Kyriakopoulos, S. G. Loizou, "Section 2.4 Robotics: Fundamentals and Prospects," in CIGR Handbook of Agricultural Engineering Volume VI Information Technology. Edited by CIGR--The International Commission of Agricultural Engineering, pp. 93-107, 2006.

[2] Rare Book of the Day – R.U.R., Karel Capek. Featuring the first appearance of the word 'Robot.' First American Printing, https:// steemit.com / books / @rarebooksleuth/rare-book-of-the-day-r-u-r-karel-capek-featuring-the-first-appearance-of-the-word- robot - first-american-printing, (accessed May 02, 2021).

[3] I. Asimov, Robots and Empire. Grafton Books, 1985.

[4] I. Asimov, Runaround. Astounding Science Fiction, 1942.

[5] J. Wallén, The history of the industrial robot. Linköping University Electronic Press, 2008.

[6] Chaotic robotics, https://www.wired.com/1994/09/tilden, (accessed May 10, 2021).

[7] Robot Nedir, http://yapbenzet.kocaeli.edu.tr/robot-nedir, (accessed May 10, 2021).

[8] T. Oates, "Introduction to Autonomous Mobile Robots, R. Siegwart, I. Nourbakhsh, MIT Press (2004)," Elsevier, 2005.

[9] A. Gasparetto, L. Scalera, "A brief history of industrial robotics in the 20th century." Advances in Historical Studies 8, vol. 8, pp. 24–35, 2019.

[10] A. Gasparetto, L. Scalera, "From the Unimate to the Delta robot: the early decades of Industrial Robotics," in Explorations in the History and Heritage of Machines and Mechanisms: Proceedings of the 2018 HMM IFToMM Symposium on History of Machines and Mechanisms, Springer International Publishing, pp. 284–295, 2019.

[11] S. Shigemi, A. Goswami, and P. Vadakkepat, "ASIMO and humanoid robot research at Honda," Humanoid robotics: A reference, pp. 55–90, 2018.

[12] M. Ben-Ari, F. Mondada, "Robots and their applications," Elements of robotics, Springer, pp.1-20, 2018.

[13] Global Hawk Makes Historic First Unmanned Flight To Australia. https://www.spacedaily.com/news/uav-01d.html, (accessed May 06, 2021).

[14] Teknoloji, https://bigumigu.com/teknoloji, (accessed Jul. 28, 2023).

[15] Sojourner (rover), https: / / e n. w i k i p e d i a. o r g / w i k i / S o j o u r n e r _ (rover), (accessed May 06, 2021).

[16] Mars Exploration Rovers, https : / / mars.nasa.gov/ mars-exploration/missions / mars-exploration-rovers, (accessed May 06, 2021).

[17] Cornell robot discovers itself and adapts to injury when it loses one of its limbs,https:// news.cornell.edu / stories / 2006 / 11 / cornell -robot-discovers-itself-and-adapts-injury (accessed May 07, 2021).

[18] Robonaut, https:// commons . wikimedia . org / wiki / File : Robonaut _ 2.png (accessed May 07, 2021).

[19] P. Sutcliffe, C. Czoski-Murray, M. Chattle, L. Ayiku, and G. Parry, "Emerging technology briefing paper on the use of robots in surgery," tech. rep., ReBIP Project, University of Sheffield, 2006.

[20] F. Rubio, F. Valero, and C. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," International Journal of Advanced Robotic Systems, vol. 16, no. 2, pp. 1–22, 2019.

[21] G. Gürgöze, İ. Türkoğlu, "Kullanım Alanlarına Göre Robot Sistemlerinin Sınıflandırılması," Fırat Üniversitesi Mühendislik Bilimleri Dergisi, vol. 31, no. 1, pp. 53–66, 2019.

[22] M. Caireta, Inigo, L. Ros, and E. Celaya, "Model Predictive Control for a Mecanum-wheeled Robot Navigating among Obstacles," IFAC-Papers OnLine, vol. 54, no. 6, pp. 119-125, 2021.

[23] S. Soni, T. Mistry, and J. Hanath, "Experimental Analysis of Mecanum wheel and Omni wheel," International Journal of Innovative Science, Engineering & Technology, vol. 1, no. 3, pp. 292–295, 2014.

[24] Material Handling With Automated Guided Vehicles [Online], https://www.peaklogix.com/materialhandling–with–automated-guided-vehicles, (accessed Jul. 17, 2022).

[25] M. Köseoğlu, O. M. Çelik, and Ö. Pektaş, "Design of an autonomous mobile robot based on ROS," in 2017 International Artificial Intelligence and Data Processing Symposium (IDAP), IEEE, pp. 1–5, 2017.

[26] R. Bostelman, T. Hong, and S. Legowik, "Mobile robot and mobile manipulator research towards ASTM standards development," in Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2016, vol. 9872, pp. 111–120, 2016.

[27] M. S. H. Achmad, N. A. Murtdza, N. A. A. Lokman, M. R. Daud, S. Razali, and D. Pebrianti, "Exploration of Unknown Environment with Ackerman Mobile Robot using Robot Operating System (ROS)," ARPN Journal of Engineering and Applied Sciences, vol. 10, no. 23, pp. 17573-17579, 2016.

[28] K. Lee, C. Jung, and W. Chung, "Accurate calibration of kinematic parameters for two wheel differential mobile robots," Journal of mechanical science and technology, vol. 25, no. 6, pp. 1603–1611, 2011.

[29] P. F. Muir, C. P. Neuman, "Kinematic modeling for feedback control of an omnidirectional wheeled mobile robot," Autonomous robot vehicles, pp. 25–31, 1990.

[30] O. Diegel, A. Badve, G. Bright, J. Potgieter, and S. Tlale, "Improved mecanum wheel design for omni-directional robots," in Proc. 2002 Australasian conference on robotics and automation, Auckland, pp. 117–121, 2002.

[31] A. Koestler, T. Bräunl, “Mobile robot simulation with realistic error models,” in International Conference on Autonomous Robots and Agents, ICARA, vol. 1, pp. 46–51, 2004.

[32] A. H. Ismail, H. R. Ramli, M. H. Ahmad, and M. H. Marhaban, “Vision-based system for line following mobile robot,” in 2009 IEEE Symposium on Industrial Electronics & Applications, vol. 2, pp. 642–645, 2009.

[33] F. Künemund, C. Kirsch, D. Heß, and C. Röhrig, “Fast and accurate trajectory generation for non-circular omnidirectional robots in industrial applications,” in ROBOTIK 2012; 7th German Conference on Robotics, VDE, pp. 1–6, 2012.

[34] F. Tóth, P. Krasňanský, and M. Gulan, “Control systems in omni-directional robotic vehicle with mecanum wheels,” in 2013 International Conference on Process Control (PC), pp. 516–521, 2013.

[35] M. Zohaib, M. Pasha, R. A. Riaz, N. Javaid, M. Ilahi, and R. D. Khan, “Control strategies for mobile robot with obstacle avoidance,” arXiv preprint arXiv :1306.1144, 2013.

[36] V. Sezer, M. Gokasan, “A novel obstacle avoidance algorithm:‘Follow the Gap Method,” Robotics and Autonomous Systems, vol. 60, no. 9, pp. 1123–1134, 2012.

[37] M. Crenganis, O. Bologa, “Implementation PID controller for a mobile platform,” Technical sciences of Romania, Lucian Blaga, University of Sibiu, 2015.

[38] M. A. Markom, A. H. Adom, E. S. M. M. Tan, S. A. Abdul, and A. Y. M. Shakaff, “Mobile Robotics Mapping using RP Lidar Scanner,” in 2015 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS), pp. 1–7, 2015.

[39] O. Gumus, M. Topaloglu, and D. Ozcelik, “The use of computer controlled line follower robots in public transport,” Procedia Computer Science, vol. 102, pp. 202–208, 2016.

[40] J. Steck, R. Morales-Ortega, J. Currence, and W. Zhou, “A mobile robot gripper for cooperative 3d printing,” 2017 International Solid Freeform Fabrication Symposium, University of Texas at Austin, 2017.

[41] C. K. Y. L. L. Man, Y. Koonjul, and L. Nagowah, “A low cost autonomous unmanned ground vehicle,” Future Computing and Informatics Journal, vol. 3, no. 2, pp. 304–320, 2018.

[42] L. Xie, C. Henkel, K. Stol, and W. Xu, “Power-minimization and energy-reduction autonomous navigation of an omnidirectional Mecanum robot via the dynamic window approach local trajectory planning,” International Journal of Advanced Robotic Systems, vol. 15, no. 1, 2018.

[43] D. Hutabarat, M. Rivai, D. Purwanto, and H. Hutomo, “Lidar-based obstacle avoidance for the autonomous mobile robot,” in 2019 12th International Conference on Information & Communication Technology and System (ICTS), pp. 197–202, 2019.

[44] S. G. Fernandez, K. Vijayakumar, R. Palanisamy, K. Selvakumar, D. Karthikeyan, D. Selvabharathi, S. Vidyasagar, and V. Kalyanasundhram, "Unmanned and autonomous ground vehicle," International Journal of Electrical and Computer Engineering, vol. 9, no. 5, 2019.

[45] R. Hartono, T. N. Nizar, I. Robani, and D. A. Jatmiko, "Motion and Navigation Control System of a Mobile Robot as A Prototype of An Autonomous Vehicle," in IOP Conference Series: Materials Science and Engineering, vol. 879, no. 1, 2020

[46] M. M. A. Alwan, A. A. Green, A. S. Noori, and A. A. Aldair, "Design and Implementation of Line Follower Arduino Mobile Robot Using Matlab Simulink Toolbox," Iraqi Journal for Electrical And Electronic Engineering, vol. 17, no. 2, 2021.

[47] S. M. Mirkhalaf, M. Fagerström, "The mechanical behavior of polylactic acid (PLA) films: fabrication, experiments and modelling," Mechanics of Time-Dependent Materials, vol. 25, no. 2, pp. 119–131, 2021.

[48] Aluminium Alloy - Commercial Alloy - 5754 - H24 Sheet and Plate, https://www.aalco.co.uk/datasheets/Aluminium-Alloy-5754-H24-Sheet-and-Plate _340.ashx, (accessed Nov. 22, 2022).

[49] PLA Filament TDS, https://www.porima3d.com, (accessed Apr. 15, 2021).

[50] T. J. Gordelier, P. R. Thies, L. Turner, and L. Johanning, "Optimising the FDM additive manufacturing process to achieve maximum tensile strength: a state-of-the-art review," Rapid Prototyping Journal, vol.25, no.26, pp: 953-971, 2019.

[51] Raspberry Pi 4 - 4GB, www.robotistan.com/raspberry-pi-4-4gb, (accessed Apr. 10, 2021).

[52] 12 V 37 mm 76 RPM 131:1 Redüktörlü DC Motor, https:// www.robotistan. com/12v-37mm-80-rpm-1311-reduktorlu-dc-motor-1311-metal-gearmotor-37dx57l-m, (accessed Apr. 10, 2021).

[53] RPLIDAR A1 Introduction and Datasheet, https://www.generationrobots. com /media/rplidar-a1m8-360-degree-laser-scanner-development-kit-datash eet -1.pdf, (accessed Apr. 05, 2022).

[54] I. Doroftei, V. Grosu, and V. Spinu, "Omnidirectional mobile robot-design and implementation," INTECH Open Access Publisher, 2007.

[55] T. Hamid, Q. Bing, and N. Ghaeminezhad, "Kinematic Model of a Four Mecanum Wheeled Mobile Robot," International journal of computer applications, vol. 113, no. 3, 2015.

[56] F. Adăscăliţei and I. Doroftei, "Practical applications for mobile robots based on mecanum wheels-a systematic survey," The Romanian Review Precision Mechanics, Optics and Mechatronics, vol. 40, pp. 21–29, 2011.

[57] A. Koubaa, H. Bennaceur, I. Chaari, S. Trigui, A. Ammar, M. F. Sriti, M.F. Alajlan, O. Cheikhrouhou, Y. Javed, A. Koubaa, and H. Bennaceur, "Introduction to mobile robot path planning," Robot path planning and cooperation: foundations, algorithms and experimentations, pp. 3–12, 2018.

[58] R. A. Saeed, M. Omri, S. Abdel-Khalek, E. S. Ali, and M. F. Alotaibi, “Optimal path planning for drones based on swarm intelligence algorithm,” Neural Computing and Applications, vol. 34, no. 12, pp. 10133–10155, 2022.

[59] M. N. Zafar and J. C. Mohanta, “Methodology for path planning and optimization of mobile robots: A review,” Procedia computer science, vol. 133, pp. 141–152, 2018.

[60] S. S. Ge and Y. J. Cui, “New potential functions for mobile robot path planning,” IEEE Transactions on robotics and automation, vol. 16, no. 5, pp. 615–620, 2000.

[61] G. Fedele, L. D’Alfonso, F. Chiaravalloti, and G. D’Aquila, “Obstacles Avoidance Based on Switching Potential Functions,” Journal of Intelligent & Robotic Systems, vol. 90, pp. 387–405, 2018.

[62] H. Niu, Y. Lu, A. Savvaris, and A. Tsourdos, “An energy-efficient path planning algorithm for unmanned surface vehicles,” Ocean Engineering, vol. 161, pp. 308–321, 2018.

[63] E. Magid, R. Lavrenov, and I. Afanasyev, “Voronoi-based trajectory optimization for UGV path planning,” 2017 International Conference on Mechanical, System and Control Engineering (ICMSC), pp. 383–387, 2017.

[64] R. Poli, J. Kennedy, and T. Blackwell, “Particle swarm optimization An overview,” Swarm intelligence, pp. 33–57, 2007.

[65] D. Wang, D. Tan, and L. Liu, “Particle swarm optimization algorithm : an overview,” Soft Computing, vol. 22, no. 2, pp. 387–408, 2018.

[66] A. U. Rehman, K. Awuah-offei, D. A. Baker, and D. Bristow, “Emergency Evacuatıon Guıdance System For Underground Mıners,” SME Annual Meeting, pp. 19-100, 2019.

[67] Z. Boger and H. Guterman, “Knowledge extraction from artificial neural networks models,” 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, vol. 4, pp. 3030–3035, 1997.

[68] J. Yu, Y. Su, and Y. Liao, “The Path Planning of Mobile Robot by Neural Networks and Hierarchical Reinforcement Learning,” Frontiers in Neurorobotics, vol. 14, pp. 1–12, 2020.

[69] C. El Mokhi and A. Addaim, “Optimization of wind turbine interconnections in an offshore wind farm using metaheuristic algorithms,” Sustainability, vol. 12, no. 14, pp. 1–24, 2020.

[70] S. Xiao, X. Tan, and J. Wang, “A simulated annealing algorithm and grid map-based uav coverage path planning method for 3d reconstruction,” Electronics, vol. 10, no. 7, 2021.

[71] F. A. Şenel, A. S. Yüksel, F. Gökçe, and T. Yiğit, “Gri kurt optimizasyon algoritması ile iki boyutlu dizilim yazılımının geliştirilmesi,” Balıkesir Üniversitesi Fen Bilim. Enstitüsü Dergisi, vol. 20, no. 2, pp. 293-306, 2018.

[72] M. Karakas, “Sınıflandırma Problemlerinde Özellik Seçimi için Karşıtlık Tabanlı Gri Kurt Algoritması,” Bilecik Şeyh Edebali Üniversitesi, Fen Bilimleri Enstitüsü, 2020.

[73] A. E. Senaras and I. Sahin, “Gsp Çözümü İçin Karınca Kolonisi Optimizasyonu,” Sosyal Bilimler Metinleri, no. 2, pp. 58–67, 2017.

[74] A. Tuncer and M. Yildirim, “Dynamic path planning of mobile robots with improved genetic algorithm,” Computers & Electrical Engineering, vol. 38, no. 6, pp. 1564–1572, 2012.

[75] “ROS / Introduction,” http:// w i k i. r o s. org / ROS / Introduction, (accessed Jun. 10, 2022).

[76] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: an open-source Robot Operating System,” in ICRA workshop on open source software, vol. 3, no. 3.2, p. 5, 2009.

[77] Y. Li and C. Shi, “Localization and navigation for indoor mobile robot based on ROS,” in 2018 Chinese automation congress (CAC), pp. 1135–1139, 2018.

[78] E. Fernandez, L. S. Crespo, A. Mahtani, and A. Martinez, “Learning ROS for robotics programming,” Packt Publishing Ltd, 2015.

[79] A. A. Boshlyakov and A. V Maltsev, “Development of a Navigation System for an Educational Mobile Robot,” in ITM Web of Conferences, vol. 35, p. 4007, 2020.

[80] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: part I,” IEEE robotics & automation magazine, vol. 13, no. 2, pp. 99–110, 2006.

[81] S. Kohlbrecher, J. Meyer, T. Graber, K. Petersen, U. Klingauf, and O. von Stryk, “Hector open source modules for autonomous mapping and navigation with rescue robots,” in RoboCup 2013: Robot World Cup XVII, pp. 624–631, 2013.

[82] S. Gatesichapakorn, J. Takamatsu, and M. Ruchanurucks, “ROS based Autonomous Mobile Robot Navigation using 2D LiDAR and RGB-D Camera,” 2019 First international symposium on instrumentation, control, artificial intelligence, and robotics (ICA-SYMP), pp. 151–154, 2019.

[83] A. J. C. Santos, J. M. M. Castellanos, and A. M. R. Duke, “ROS: An Autonomous Robot Operating System For Simultaneous Localization and Mapping Using a 2d LIDAR Sensor,” in 19th LACCEI International Multi-Conference for Engineering, pp. 3–4, 2021.

[84] C. Lu, J. Wang, and X. Cui, “Research and Implementation of Unknown Environment Map Construction Method with Lidar-based Robot,” in 2020 5th International Conference on Automation, Control and Robotics Engineering (CACRE), pp. 26–30, 2020.

[85] N. Baras, G. Nantzios, D. Ziouzios, and M. Dasygenis, “Autonomous obstacle avoidance vehicle using lidar and an embedded system,” in 2019 8th International Conference on Modern Circuits and Systems Technologies (MOCAST), pp. 1–4, 2019.

[86] P. Denysyuk, V. Teslyuk, and I. Chorna, “Development of mobile robot using LIDAR technology based on Arduino controller,” in 2018 XIV-th International Conference on Perspective Technologies and Methods in MEMS Design (MEMSTECH), pp. 240–244, 2018.

[87] M. A. Markom, S. A. A. Shukor, A. H. Adom, E. Tan, and A. Y. M. Shakaff, “Indoor scanning and mapping using mobile robot and rp lidar,” International Journal of Advances in Mechanical and Automobile Engg.(IJAMAE), vol. 3, no. 1, 2016.

[88] S. Fakhri, R. Mardiati, E. Mulyana, and T. Priatna, “Prototype Design for Object Coordinate Detection using RP LIDAR Concept,” in 2020 6th International Conference on Wireless and Telematics (ICWT), pp. 1–6. 2020.

[89] HC-SR04 Ultrasonic Distance Sensor, http:// www.robotistan.com / hc-sr04-ultrasonic-distance-sensor, (accessed Sep. 11, 2022).

[90] Compare Scanning Lasers & Lidar, https:// www. r o b o t s h o p . com /en/ compare-scanning-laser-rangefinders.html, (accessed Sep. 11, 2022).

[91] YDLIDAR X2 Data Sheet, https://www . ydlidar .com / Public / upload / files /2021-08-20/YDLIDAR X2 Data sheet V2.0.pdf, (accessed Sep. 11, 2022).

[92] G. Klančar, L. Teslić, and I. Škrjanc, “Mobile-robot pose estimation and environment mapping using an extended Kalman filter,” International Journal of Systems Science, vol. 45, no. 12, pp. 2603–2618, 2014.

[93] P. E. Peralta, M. A. Luna, P. D. le Puente, P. Campoy, H. Bavle, A. Carrio, and C. C. Ulloa, “Performance Analysis of Localization Algorithms for Inspections in 2D and 3D Unstructured Environments Using 3D Laser Sensors and UAVs,” Sensors, vol. 22, no. 14, pp. 1–26, 2022.

TEZDEN ÜRETİLMİŞ YAYINLAR

Projeler

1. Otonom Yer Aracı Modellenmesi, Yürütücü Prof. Dr. Semih Sezer, FYL-2022-4880, 05.04.2022-05.10.2023, Araştırmacı Anıl Gökhan Göktaş.