

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

**ENDÜSTRİYEL OTONOM MOBİL ROBOT SÜPÜRGE İÇİN TAM
KAPSAMA YOL PLANLAMA ALGORİTMASI GELİŞTİRİLMESİ**

EYLÜL ÖZER

KOCAELİ 2025

KOCAELİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

**ENDÜSTRİYEL OTONOM MOBİL ROBOT SÜPÜRGE İÇİN TAM
KAPSAMA YOL PLANLAMA ALGORİTMASI GELİŞTİRİLMESİ**

EYLÜL ÖZER

Dr. Öğr. Üyesi Alparslan Burak İNNER
Danışman, Kocaeli Üniversitesi

.....

Dr. Öğr. Üyesi Levent BAYINDIR
Jüri Üyesi, Kocaeli Üniversitesi

.....

Doç. Dr. Adnan Fatih KOCAMAZ
Jüri Üyesi, İnönü Üniversitesi

.....

Tezin Savunulduğu Tarih: 21.01.2025

ETİK BEYAN VE ARAŞTIRMA FONU DESTEĞİ

Kocaeli Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında,

- Bu tezin bana ait, özgün bir çalışma olduğunu,
- Çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı,
- Bu çalışma kapsamında elde edilen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi,
- Bu çalışmanın Kocaeli Üniversitesi'nin abone olduğu intihal yazılım programı kullanılarak Fen Bilimleri Enstitüsü'nün belirlemiş olduğu ölçütlere uygun olduğunu,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı,

beyan ederim.

Bu tez çalışmasının herhangi bir aşaması hiçbir kurum/kuruluş tarafından maddi/alt yapı desteği ile desteklenmemiştir.

Bu tez çalışması kapsamında üretilen veri ve bilgiler tarafından no'lu proje kapsamında maddi/alt yapı desteği alınarak gerçekleştirilmiştir.

Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçları kabul ettiğimi bildiririm.

Eylül ÖZER

YAYIMLAMA VE FİKRİ MÜLKİYET HAKLARI

Fen Bilimleri Enstitüsü tarafından onaylanan lisansüstü tezimin tamamını veya herhangi bir kısmını, basılı ve elektronik formatta arşivleme ve aşağıda belirtilen koşullarla kullanıma açma izninin Kocaeli Üniversitesi'ne verdiğimi beyan ederim. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet hakları bende kalacak, tezimin tamamının ya da bir bölümünün gelecekteki makale, kitap, tebliğ, lisans, patent gibi çalışmalarda kullanımı, danışmanımın isim hakkı saklı kalmak koşuluyla ve her iki tarafın bilgisi dâhilinde bana ait olacaktır.

Tezin kendi özgün çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanılması zorunlu metinlerin yazılı izin alarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

Yükseköğretim kurulu tarafından yayınlanan **“Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge”** kapsamında tezim aşağıda belirtilen koşullar haricinde YÖK Ulusal Tez Merkezi/ Kocaeli Üniversitesi Kütüphaneleri Açık Erişim Sisteminde erişime açılır.

- Enstitü yönetim kurulu kararı ile tezimin erişime açılması mezuniyet tarihinden itibaren 2 yıl ertelenmiştir.
- Enstitü yönetim kurulu gerekçeli kararı ile tezimin erişime açılması mezuniyet tarihinden itibaren 6 ay ertelenmiştir.
- Tezim ile ilgili gizlilik kararı verilmemiştir.

.....
(İmza)
Eylül ÖZER

ÖNSÖZ VE TEŞEKKÜR

Bu tez çalışması, endüstriyel temizlik robot süpürge için kapsam yol planlama algoritması geliştirilmesi amacıyla gerçekleştirilmiştir. Çalışmanın amacı, enerji verimli bir şekilde endüstriyel alanlardaki atıkların etkili bir şekilde temizlenmesini sağlayarak hem işletme maliyetlerinin düşürülmesine hem de çevresel sürdürülebilirliğe katkı sağlamaktır. Algoritmanın uygulanabilirliğini göstermek için simülasyon ve gerçek ortam testleri gerçekleştirilmiştir.

Tez çalışmam süresince bilgi, deneyim ve görüşleri ile çalışmalarına katkıda bulunan, danışmanım Sayın Dr. Öğr. Üyesi A. Burak İNNER'e teşvikleri ve destekleri için teşekkür ederim.

Bu tez çalışması kapsamında geliştirilen kapsam yol planlama algoritmalarının testleri, Birfen Elektrik Elektronik San. Tic. Ltd. Şti. atölyesinde gerçekleştirilmiştir. Bu süreçte sağladığı destek ve katkılarından dolayı firma genel müdürü Sayın Seçkin ÖZER'e teşekkürlerimi sunarım.

Tüm tez çalışma süresince beni destekleyen ve her zaman bana umutlu bir bakış açısı kazandıran aileme en içten dileklerle teşekkür ederim.

Ocak - 2025

Eylül ÖZER

İÇİNDEKİLER

ETİK BEYAN VE ARAŞTIRMA FONU DESTEĞİ.....	i
YAYIMLAMA VE FİKRİ MÜLKİYET HAKLARI	ii
ÖNSÖZ VE TEŞEKKÜR.....	iii
İÇİNDEKİLER.....	iv
ŞEKİLLER DİZİNİ	vi
TABLOLAR DİZİNİ.....	viii
SİMGELER VE KISALTMALAR DİZİNİ	ix
ÖZET	x
ABSTRACT	xi
1. GİRİŞ.....	1
1.1. Kapsam Yol Planlama ve Endüstriyel Uygulamaları.....	3
1.2. Kapsam Yol Planlama Algoritmaları	4
1.3. Önerilen Yöntem	7
2. MATERYAL VE METODLAR.....	9
2.1. ROS (Robot İşletim Sistemi).....	9
2.2. SLAM.....	10
2.3. Robot	11
2.4. Gazebo Ortamı	12
2.5. Atölye Ortamı.....	14
2.6. Global ve Yerel Planlayıcı	14
2.7. Hücresel Ayrıştırma	17
2.8. Serbest Hücresel Ayrıştırma.....	18
2.8.1. Boustrophedon Hücresel Ayrıştırma	20
2.8.2. Hiyerarşik Hücresel Ayrıştırma	21
2.8.3. Kare Hücresel Ayrıştırma	22
2.8.4. Trapez Hücresel Ayrıştırma.....	23
2.8.5. Altıgen Hücresel Ayrıştırma.....	23
2.8.6. Voronoi Hücresel Ayrıştırma	24
2.8.7. Spiral Hücresel Ayrıştırma	25
2.8.8. Dışbükey Poligon Hücresel Ayrıştırma	26
2.9. Kapsama Yol Planlama	26
2.9.1. Boustrophedon Kapsama Yol Planlama	28
2.9.2. Boustrophedon Yayılan Ağaç Kaplama (YAK) Yol Planlama	30
2.9.3. Izgara Tabanlı Gezgin Satıcı Problemi ile Kapsama Yol Planlama	31
2.9.4. Sinir Ağı Tabanlı Kapsama Yol Planlama.....	33
2.9.5. Dışbükey Sensör Yerleştirme Kapsama Yolu Planlaması.....	35
2.9.6. Izgara Tabanlı Yerel Enerji Minimizasyonu Kapsama Yol Planlama	36
2.9.7. Kontur Hattı Tabanlı Kapsama Yol Planlama	37
2.9.8. Enerji Fonksiyonel Optimizasyonu Yöntemi ile Kapsam Yol Planlama	38
2.10. Oda Segmentasyonu	40
2.11. Önerilen Yöntem.....	41
3. DENEYSEL BULGULAR VE TARTIŞMA	45
3.1. Haritalama.....	46
3.1.1. Sanal Ortam Haritalama	46

3.1.2. Atölye Test Ortamı Haritalama	47
3.1.3. Fabrika Ortamı Haritalama	48
3.2. Performans Metrikleri	49
3.3. Yöntemlerin Deneysel İncelemesi ve Performans Karşılaştırmaları	52
3.3.1. Boustrophedon Hücresel Ayırıştırma Yöntemi ile Deney Sonuçları.....	52
3.3.2. Izgara Tabanlı Gezin Satıcı Problemi Yaklaşımı ile Deney Sonuçları	56
3.3.3. Sinir Ağı Tabanlı Kapsama Yol Planlayıcısı Yöntemi ile Deney Sonuçları	59
3.3.4. Enerji Fonksiyonel Optimizasyonu Yöntemi ile Deney Sonuçları.....	62
3.3.5. Ortalama Kayma Kümeleme Yöntemi ile Deney Sonuçları.....	66
3.3.6. Fields2Cover ve nav2_coverage Paketlerinin İncelenmesi	68
3.3.7. Önerilen Yöntemin Performans Değerlendirmesi	73
4. SONUÇLAR VE ÖNERİLER.....	78
KAYNAKLAR.....	82
KİŞİSEL YAYIN VE ESERLER.....	86
ÖZGEÇMİŞ.....	87

ŞEKİLLER DİZİNİ

Şekil 2.1.	ROS 2 Mimarisi Genel Bakış	10
Şekil 2.2.	Cartographer Metodu ile Harita Çıkarılması	12
Şekil 2.3.	Robot Görseli.....	12
Şekil 2.4.	3 Boyutlu Test Ortamı	13
Şekil 2.5.	Gazebo Test Ortamı	14
Şekil 2.6.	Birfen Atölye Ortamı	15
Şekil 2.7.	Hücresel Ayrıştırma.....	17
Şekil 2.8.	Serbest C-Uzay Hücresel Ayrıştırma ile Alt Uzaylara Bölünmesi.....	19
Şekil 2.9.	Boustrophedon Hücresel Ayrıştırma	20
Şekil 2.10.	Hiyerarşik Hücresel Ayrıştırma.....	22
Şekil 2.11.	Trapez Hücresel Ayrıştırma.....	24
Şekil 2.12.	Boustrophedon Algoritması Görünümü	29
Şekil 2.13.	Boustrophedon YAK Algoritması Gösterimi	30
Şekil 2.14.	Boustrophedon YAK Algoritması Akış Şeması	32
Şekil 2.15.	Önerilen Yöntem Aşamaları	41
Şekil 2.16.	Önerilen Algoritma Akış Şeması.....	43
Şekil 3.1.	Sanal Ortam Haritası.....	46
Şekil 3.2.	Atölye Test Ortamı	47
Şekil 3.3.	Atölye Ortamı Haritası	48
Şekil 3.4.	Fabrika Ortamı Haritası	49
Şekil 3.5.	Boustrophedon Algoritması Simülasyon Ortamı Yol Planı	53
Şekil 3.6.	Boustrophedon Algoritması Atölye Ortamı Yol Planı	53
Şekil 3.7.	Boustrophedon Algoritması Fabrika Ortamı Yol Planı	54
Şekil 3.8.	Izgara Tabanlı Gezgin Satıcı Problemi Algoritması Simülasyon Ortamı Yol Planı.....	56
Şekil 3.9.	Izgara Tabanlı Gezgin Satıcı Problemi Algoritması Atölye Ortamı Yol Planı.....	57
Şekil 3.10.	Izgara Tabanlı Gezgin Satıcı Problemi Algoritması Fabrika Ortamı Yol Planı.....	57
Şekil 3.11.	Sinir Ağı Tabanlı Kapsama Yol Planlama Algoritması Simülasyon Ortamı Yol Planı.....	60
Şekil 3.12.	Sinir Ağı Tabanlı Kapsama Yol Planlama Algoritması Atölye Ortamı Yol Planı.....	61
Şekil 3.13.	Sinir Ağı Tabanlı Kapsama Yol Planlama Algoritması Fabrika Ortamı Yol Planı.....	62
Şekil 3.14.	Enerji Fonksiyonel Optimizasyonu ile Kapsama Yol Planlama Algoritması Simülasyon Ortamı Yol Planı.....	63
Şekil 3.15.	Enerji Fonksiyonel Optimizasyonu ile Kapsama Yol Planlama Algoritması Atölye Ortamı Yol Planı.....	64
Şekil 3.16.	Enerji Fonksiyonel Optimizasyonu ile Kapsama Yol Planlama Algoritması Fabrika Ortamı Yol Planı	64
Şekil 3.17.	Ortalama Kayma Kümeleme Yöntemi ile Kapsama Yol Planlama Algoritması Simülasyon Ortamı Yol Planı	67
Şekil 3.18.	Ortalama Kayma Kümeleme Yöntemi ile Kapsama Yol Planlama Algoritması Atölye Ortamı Yol Planı	68
Şekil 3.19.	Fields2Cover Kütüphanesi Akış Şeması	69

Şekil 3.20. Atölye Ortamı Haritasının Sınırlarının Poligon Formatında Gösterimi	71
Şekil 3.21. Opennav_coverage Paketi Atölye Ortamı Yol Planı	72
Şekil 3.22. Önerilen Algoritma Simülasyon Ortamı Yol Planı	74
Şekil 3.23. Önerilen Algoritma Düşük Şişirme Yarıçapı Uygulanması	74
Şekil 3.24. Önerilen Algoritma ile Robotun Geçtiği Yerlerin Gösterimi	75
Şekil 3.25. Önerilen Algoritma Fabrika Ortamı Yol Planı	76

TABLolar DİZİNİ

Tablo 3.1	Boustrophedon algoritması ile elde edilen performans metrikleri	54
Tablo 3.2	Izgara tabanlı gezgin satıcı algoritması ile elde edilen performans metrikleri	58
Tablo 3.3	Sinir ağı tabanlı kapsama yol planlama ile elde edilen performans metrikleri	60
Tablo 3.4	Sinir ağı tabanlı kapsama yol planlama ile elde edilen performans metrikleri	65
Tablo 3.5	Ortalama kayma kümeleme kapsama yol planlama algoritması ile yapılan çalışmada elde edilen performans metrikleri	66
Tablo 3.6	Önerilen Yöntem Test Sonuçları	77
Tablo 4.1	Performans Metriklerine Göre Yöntemlerin Karşılaştırılması	78

SİMGELER VE KISALTMALAR DİZİNİ

Kısaltmalar

BINN	: Bio-Inspired Neural Networks (Biyolojik Esinli Sinir Ağı Algoritmaları)
CCD	: Convex Coverage Decomposition (Konveks Kapsama Dekompozisyonu)
C-UZAYI	: Configuration Space (Yapılandırma Uzayı)
DDS	: Dynamic Domain Selection (Dinamik Alan Seçimi)
DQN	: Deep Q Network (Derin Q Ağı)
DWB	: Dynamic Window Approach (Dinamik Pencere Yaklaşımı)
FOV	: Field of View (Görüş Alanı)
ILP	: Integer Linear Programming (Tam Sayılı Doğrusal Programlama)
IMU	: Inertial Measurement Unit (Eylemsizlik Ölçüm Birimi)
ROS	: Robot Operating System (Robot İşletim Sistemi)
ROS RMW	: ROS Middleware (Robot İşletim Sistemi Ara Yazılım Arayüzü)
SLAM	: Simultaneous Localization and Mapping (Eşzamanlı Yerelleştirme ve Haritalama)
STC	: Spanning Tree Coverage (Yayılan Ağaç Kapsama)
TSP	: Traveling Salesman Problem (Gezgin Satıcı Problemi)
UAV	: Unmanned Aerial Vehicle (İnsansız Hava Araçları)
USV	: Unmanned Surface Vehicle (İnsansız Yüzey Araçları)
YAK	: Yayılan Ağaç Kaplama

ENDÜSTRİYEL OTONOM MOBİL ROBOT SÜPÜRGE İÇİN TAM KAPSAMA YOL PLANLAMA ALGORİTMASI GELİŞTİRİLMESİ

ÖZET

Bu tez çalışması, endüstriyel otonom mobil robot süpürgeler için tam kapsama yol planlama algoritması geliştirmeyi amaçlamaktadır. Çalışma, endüstriyel alanlarda enerji verimliliğini artırmaya odaklanırken, robotun belirlenen alanları tam olarak kaplayacak şekilde optimum hareket güzergâhını oluşturmayı hedeflemektedir. Geliştirilen algoritmanın performansı, simülasyon ortamı, firmanın test atölyesi ve endüstriyel fabrika ortamında yapılan kapsamlı deneylerle değerlendirilmiştir.

Önerilen yöntem, hücresel ayrıştırma ve boustrophedon-temelli bir hibrit yapı kullanarak, alanın geometrik özelliklerine uygun bir yol planlama stratejisi sunmaktadır. Algoritma, robotun navigasyon yeteneklerini ROS 2 platformu üzerinde tümleştirerek, engel tespiti ve karmaşık ortamlarda hareket planlaması yapabilmektedir. Endüstriyel ortamlarda sıklıkla karşılaşılan dar geçitler, yoğun engel bölgeleri ve geniş açık alanlar gibi zorluklara çözüm getirecek şekilde tasarlanmıştır.

Çalışma kapsamında altı farklı kapsama yol planlama yöntemi (Izgara Tabanlı Gezgin Satıcı Problemi, Boustrophedon, Sinir Ağları, Enerji Fonksiyonel, Ortalama Kayma Kümeleme ve önerilen yöntem) üç farklı ortamda (simülasyon, Birfen atölyesi ve fabrika) test edilerek kapsamlı bir karşılaştırmalı analiz sunulmuştur. Deneysel sonuçlar, önerilen algoritmanın kapsama yüzdesi, dönüş sayısı, seyahat mesafesi ve hesaplama süresi gibi kritik performans metrikleri açısından mevcut yöntemlere kıyasla üstünlük sağladığını göstermiştir. Simülasyon ortamında %98,3, atölye ortamında %95,4 ve fabrika ortamında %94,7 kapsama oranı elde edilmiştir. Algoritmanın enerji verimliliğini artırmak için dönüş sayısını minimize etme özelliği, özellikle endüstriyel temizlik uygulamaları için önemli bir avantaj sunmaktadır.

Bu çalışma, endüstriyel mobil robot süpürgelerin tam kapsama yol planlama problemine yönelik etkin bir çözüm sunarak, endüstriyel otomasyon ve robotik alanına önemli bir katkı sağlamaktadır.

Anahtar Kelimeler: Endüstriyel Temizlik Robotu, Kapsam Yol Planlama, Otonom Mobil Temizlik Robotu, Tam Kapsama Yol Planlama.

ALGORITHM DEVELOPMENT FOR COMPLETE COVERAGE PATH PLANNING IN AUTONOMOUS INDUSTRIAL CLEANING ROBOTS

ABSTRACT

This thesis aims to develop a complete coverage path planning algorithm for industrial autonomous mobile cleaning robots. The study focuses on enhancing energy efficiency in industrial environments while creating an optimal movement path that ensures the robot completely covers designated areas. The performance of the developed algorithm has been evaluated through comprehensive experiments conducted in simulation environments, test workshops, and industrial factory settings.

The proposed method presents a path planning strategy adapted to the geometric characteristics of the area, using a hybrid structure based on cellular decomposition and boustrophedon approaches. The algorithm integrates the robot's navigation capabilities on the ROS 2 platform, enabling obstacle detection and motion planning in complex environments. It is designed to address challenges frequently encountered in industrial settings such as narrow passages, densely populated obstacle regions, and large open spaces.

This study provides a comprehensive comparative analysis by testing six different coverage path planning methods (Grid-Based Traveling Salesman Problem, Boustrophedon, Neural Networks, Energy Functional, Mean Shift Clustering, and the proposed method) across three different environments (simulation, Birfen atelier, and factory). Experimental results demonstrate that the proposed algorithm outperforms existing methods in terms of critical performance metrics including coverage percentage, number of turns, travel distance, and computation time. Coverage rates of 98.3% in simulation environments, 95.4% in workshop settings, and 94.7% in factory environments were achieved. The algorithm's ability to minimize the number of turns to improve energy efficiency presents a significant advantage, particularly for industrial cleaning applications.

This study makes a significant contribution to the field of industrial automation and robotics by providing an effective solution to the complete coverage path planning problem for industrial mobile cleaning robots.

Keywords: Autonomous Mobile Cleaning Robot, Coverage Path Planning, Full Coverage Path Planning, Industrial Cleaning Robot.

1. GİRİŞ

Endüstriyel temizlik robotları, geniş ve karmaşık alanların düzenli, etkin ve eksiksiz bir şekilde temizlenmesini sağlamak amacıyla fabrikalar, depolar, havaalanları, büyük perakende mağazaları ve hastaneler gibi geniş alanlara sahip ortamlarda giderek daha yaygın bir şekilde kullanılmaktadır. Bu robotlar, genellikle yoğun insan ve makine trafiğinin olduğu dinamik ortamlarda çalışır ve temizlik görevlerini yerine getirirken çevresindeki engellerden kaçınma, dar alanlarda etkili hareket edebilme ve karmaşık yapıları temizleme gibi zorluklarla karşılaşır. Bu sebeple, endüstriyel temizlik robotlarının güvenli, kesintisiz ve verimli çalışabilmesi için gelişmiş yol planlama algoritmalarına olan ihtiyaç kaçınılmaz hale gelmiştir (Chen ve diğ., 2019).

Endüstriyel temizlik robotlarının temel amacı, belirlenmiş bir alanı eksiksiz bir şekilde temizlemek ve bu süreçte minimum enerji tüketerek maksimum verim elde etmektir. Endüstriyel ortamların çok geniş olması ve makineler gibi statik engellere yaklaşılmaması ile insan ve forklift gibi dinamik engellerden sakınma gerekliliği zorluklar yaratmaktadır. Endüstriyel temizlik robotlarının çevresel koşullara uyum sağlayabilmesi ve karşılaşabilecekleri her türlü engelin etrafından dolaşarak veya bu engelleri aşarak temizlik işlemlerini sorunsuz bir şekilde tamamlayabilmesi büyük önem taşır.

Tam kapsama yol planlama, bir robotun belirli bir alanı tamamen kapsamaları için oluşturulan bir yol planlama stratejisidir. Bu tür planlama, robotun her bir noktayı en az bir kez ziyaret ederek görevini eksiksiz yerine getirmesini sağlar. Özellikle temizlik robotları, tarım robotları, mayın tarama robotları gibi otonom sistemler için bu planlama türü kritik öneme sahiptir. Tam kapsama yol planlama problemi, literatürde engellerin mevcut olduğu ve olmadığı iki ana senaryo kapsamında incelenmiştir. Engellerin olmadığı alanlarda veya az sayıda ve basit şekilli engellerin olduğu alanlarda robotun yapılandırılmamış ve sezgisel algoritmalar kullanarak tüm alanı kapsamaları mümkündür. Bu tür algoritmalar, alanın geometrisine bağlı kalmaksızın robotun sistematik bir şekilde ilerlemesine olanak tanır. Seçilen alanlarda, robot rastgele veya sistematik hareketlerle hedefe ulaşabilir; random walk, spiral, s-shape, snake gibi algoritmalar robotun serbest hareket ederek alanı kapsamalarına olanak tanır(Choset & Pignon, 1998). Engeller bulunmadığı için alanın tamamını kapsama, basit ileri-geri hareketlerle veya spiral yollar izleyerek gerçekleştirilebilir.

Engellerin bulunduğu alanlarda tam kapsama sağlamak daha karmaşıktır ve bu durumda hücrel ayrıştırma yöntemine başvurulması gerekmektedir. Hücrel ayrıştırma, karmaşık yapıdaki engellerin bulunduğu alanların daha küçük, basit hücrelere ayrılması işlemidir (Barraquand & Latombe, 1991). Bu hücrelerin her biri, robot tarafından sistematik bir şekilde kapsanır, böylece robotun zigzag veya ileri-geri hareketlerle tüm alanı gezmesi sağlanır. Engellerin bulunduğu düzensiz ortamlarda, robotun her hücreyi kapsamaları için kullanılan bu yöntem, kapsama işlemini kolaylaştırır ve robotun engellerle çarpışmasını önler (Choset & Pignon, 1998). Özellikle tarımsal uygulamalarda yapılan çalışmalar, bu yöntemin karmaşık alanlardaki etkinliğini ortaya koymuş ve daha optimal algoritmaların geliştirilmesi için önemli bir adım olduğunu vurgulamıştır (Oksanen & Visala, 2007).

Bu çerçevede, endüstriyel temizlik robotları için sıklıkla tercih edilen Boustrophedon algoritması, özellikle geniş ve karmaşık alanlarda robotun tüm alanı eksiksiz bir şekilde tarayabilmesi için etkili bir çözüm sunar. Boustrophedon algoritması, alanı paralel hatlarla tarayarak robotun minimum dönüş ve hareket ile kapsama alanını tamamlamasını sağlar (Bähnemann ve diğ., 2019). Bu yöntem, robotun gereksiz enerji tüketiminden kaçınmasına ve daha kısa sürede daha geniş alanları temizleyebilmesine olanak tanır. Endüstriyel temizlik robotları, bu algoritma sayesinde iş gücü maliyetlerini düşürerek, operasyonel verimliliği artırmakta ve temizlik süreçlerinde daha hızlı ve etkin sonuçlar elde etmektedir (Rodrigues ve diğ., 2018).

Ancak, endüstriyel alanların yapısı gereği makinelere çok fazla yanaşmama gerekliliği olduğundan ve çoğunlukla dinamik engeller fazla olduğundan, bu alanlarda daha gelişmiş algoritmaların uygulanması gerekmektedir. Boustrophedon gibi algoritmaların endüstriyel temizlik robotlarında kullanımı, alanın yapısına ve o anki duruma bağlı olarak optimize edilmelidir (Mahajan & Rock, 2020). Dinamik engeller, dar alanlar ve karmaşık yapıların varlığı, temizlik robotlarının yol planlamasında esneklik ve hassasiyet gerektirir. Bu tez çalışması, endüstriyel temizlik robotları için enerji verimliliğini artıran ve dinamik koşullara uyum sağlayabilen kapsam yol planlama algoritmalarının geliştirilmesine odaklanmıştır. Çalışmanın temel amacı, robotların endüstriyel temizlik alanında daha verimli ve maliyet etkin bir şekilde çalışmasını mümkün kılan algoritmaların iyileştirilmesidir. Özellikle, temizlik robotlarının rotalarını optimize

edilmesiyle, engellere çarpmaların önüne geçilmesi ve gereksiz enerji tüketiminin azaltılarak tüm alanı eksiksiz temizlemesi sağlanmış, böylece endüstriyel temizlik süreçlerine yenilikçi ve daha etkin çözümler sunulmaktadır.

1.1. Kapsam Yol Planlama ve Endüstriyel Uygulamaları

Kapsam yol planlama algoritmaları, robotik, madencilik ve askeri uygulamalardaki alanların etkili bir şekilde taranmasını sağlamak amacıyla sürekli bir evrim sürecinden geçmiştir. Bu algoritmalar, rastgele yürüyüş, duvarı takibi, yılan ve sarmal gibi basit geometrik yaklaşımlarla sınırlıyken, zamanla daha karmaşık ve verimli yöntemlere dönüşmüştür. Günümüzde, endüstriyel uygulamalarda daha karmaşık yapıların ve yoğun engel ortamlarının varlığı, bu algoritmaların doğruluğunu, hızını ve esnekliğini artırmayı amaçlayan yeni yaklaşımlar geliştirilmesini zorunlu kılmıştır. Özellikle yapay zekâ, makine öğrenimi ve nöral ağlar gibi teknolojilerle desteklenen kapsam yol planlama algoritmaları, geniş alanların daha verimli bir şekilde kapsanmasını sağlamış ve enerji tüketimini en aza indirmeyi başarmıştır. Bu gelişmeler, özellikle endüstriyel temizlik robotları, tarım robotları, tarımsal ve askeri insansız hava araçları gibi alanlarda önemli uygulamalara kapı açmıştır.

Endüstriyel temizlik robotları, modern üretim süreçlerinin vazgeçilmez unsurları arasında yer alarak, büyük üretim tesislerinden depolara, alışveriş merkezlerinden lojistik merkezlerine kadar geniş uygulama alanlarına sahiptir. Bu robotlar, insan müdahalesini azaltırken, karmaşık ortamlarda verimliliği artırıp işçilik maliyetlerini düşürmekte ve iş süreçlerinin optimize edilmesine katkıda bulunmaktadır. Ayrıca, enerji verimliliğini artıran kapsam yol planlama algoritmaları sayesinde maksimum alan kapsaması sağlanırken, minimum enerji tüketimiyle işletme maliyetlerini azaltarak hem de sürdürülebilirliğe önemli bir katkı sunulmaktadır.

Özellikle karmaşık endüstriyel ortamlarda, engellerin yoğun olduğu alanlarda ve geniş yüzeylerde etkin çalışabilmeleri için geliştirilen kapsam yol planlama algoritmaları, bu robotların performansını doğrudan etkilemektedir. Örneğin, endüstriyel temizlik robotları için geliştirilen enerji verimli algoritmalar hem zamandan hem de maliyetten tasarruf sağlarken, minimum dönüş hareketleriyle operasyonel süreçlerde süreklilik ve verimlilik sağlamaktadır. Temizlik işlemlerinin düzenli, güvenilir ve standart bir şekilde

gerçekleştirilmesi, iş yerlerindeki hijyen standartlarını yükseltmekte ve işçi sağlığını korumaya yardımcı olmaktadır. Bu bağlamda, endüstriyel temizlik robotları, sadece otomasyonun bir unsuru olmaktan öte, modern endüstriyel yönetim stratejilerinin ve operasyonel mükemmellik anlayışının temel bir bileşeni haline gelmiştir.

1.2. Kapsam Yol Planlama Algoritmaları

Kapsam yol planlama algoritmaları, otonom robotların belirli bir alanı eksiksiz bir şekilde tarama görevlerini yerine getirmesini sağlayan stratejik yöntemlerdir. Bu algoritmalar, enerji verimliliği sağlamak, yol uzunluğunu en aza indirmek ve hesaplama süreçlerini optimize etmek gibi kritik hedeflere yönelik olarak geliştirilmiştir. Algoritmaların tasarımı, kullanılacakları alanın spesifik gereksinimlerine göre özelleştirilebilme esnekliği sunar. Bu özellik, farklı uygulama alanlarında, örneğin tarım, endüstriyel temizlik, insansız hava araçları (UAV) ve arama-kurtarma operasyonları gibi senaryolarda etkinliğin artırılmasını sağlar. Böylece, operasyonel süreçlerde enerji ve zaman gibi kaynakların daha verimli kullanılması mümkün olurken, bu yöntemler hem pratik hem de teorik düzeyde önemli bir araştırma alanı haline gelmektedir.

Son yıllarda kapsam yol planlama algoritmalarında kaydedilen gelişmeler, bu alandaki yenilikçi uygulamaların etkileyici örneklerini ortaya koymuştur. Örneğin, insansız yüzey araçları (USV) için önerilen bir takviyeli öğrenme algoritması, harita ön işleme ve derin Q ağı (DQN) kullanımıyla kapsama oranını artırırken tekrarlama oranını azaltmayı başarmıştır (Xing ve diğ., 2023). Çoklu robot sistemlerinde kullanılan TMSTC* (Multirobot Spanning Tree Coverage Star) algoritması ise dönüş sayısını azaltarak kapsama görevlerinin daha hızlı tamamlanmasını sağlamıştır (Lu ve diğ., 2023). Sensor ağları için geliştirilen enerji verimli algoritmalar, düğüm hareketlerini optimize ederek ağın ömrünü uzatmayı mümkün kılmıştır. Bu yaklaşımlar, özellikle enerji sınırlamalarının önemli olduğu uygulamalarda dikkat çekmektedir (Liu & Zhou, 2023). Tarımsal alanlarda ise, çoklu robot sistemleri için önerilen poligonal hücre bazlı yöntemler, alanın alt bölümlere ayrılmasını sağlayarak her robotun kendi bölgesinde optimal kapsama gerçekleştirmesini sağlamıştır (Choton & Prabhakar, 2023). Kapalı alanlarda kullanılan Boustrophedon hücresel ayrıştırma algoritması, mobilyalı ve mobilyasız ortamlarda yüksek kapsama oranlarına ulaşmış, özellikle düşük mobilyalı odalarda en verimli yöntemlerden biri olarak öne çıkmıştır (Bormann ve diğ., 2018).

Benzer şekilde, SCCPP (Smooth Complete Coverage Trajectory Planning) algoritması gibi yeniden planlama ve yol düzleştirme tekniklerini birleştiren yöntemler, kapsama yollarını optimize ederek robot hareketlerini iyileştirmiştir (Şelek ve diğ., 2022). Fotovoltaik enerji santrallerinde derin öğrenme tabanlı semantik segmentasyon ve kapsama yol planlamasının entegrasyonu, insansız hava araçlarının bu alanlarda etkili bir şekilde çalışmasını sağlamıştır (Pérez-González ve diğ., 2021). Dört rotorlu UAV'ler için geliştirilen planlama algoritmaları ise görev süresini optimize ederken tam kapsama sağlamayı başarmıştır (Nama ve ark., 2016). Neural network tabanlı yöntemler, karar verme süreçlerini iyileştirerek kapsama yüzdesini artırmış ve enerji tüketimini optimize etmiştir (Yang & Luo, 2004a). Ayrıca, B-spline eğrilerini kullanan algoritmalar hem konveks hem de konveks olmayan bölgelerde etkili kapsama yolları sunmuştur (Rodrigues ve diğ., 2018).

Statik ve dinamik engellerle başa çıkabilen CCD* (Complete Coverage D*) algoritması, temizlik robotları için etkili çözümler sunmuştur (Dakulović ve diğ., 2011). A* algoritması tabanlı yaklaşımlar ise özellikle engellerin yoğun olduğu bölgelerde üstün performans sergilemiştir (Cai ve diğ., 2014). Çevrimiçi kapsama planlama yöntemleri hem statik hem de dinamik ortamlardaki esnekliği artırarak uygulama alanlarını genişletmiştir. Spiral yol planlaması ve sınırlı ters mesafe dönüşümü gibi yenilikçi yöntemler, çevrimdışı planlama süreçlerine güçlü bir alternatif olarak öne çıkmıştır (Choi ve diğ., 2009)

Öne çıkan klasik tekniklerden biri olan Boustrophedon Hücresel Ayrıştırma, çalışma alanını daha küçük hücrelere ayırarak alanın tamamının sistematik bir şekilde taranmasını sağlar. Bu algoritma, enerji verimliliği ve dönüş sayısının azaltılması gibi avantajlarıyla endüstriyel temizlik ve tarım robotlarında sıklıkla kullanılmaktadır (Mier ve diğ., 2023). Özellikle tarımsal uygulamalarda, bu yöntem başlık oluşturma ve swath düzenleme gibi işlevlerle entegre edilerek robotların optimize edilmiş yollar izlemesini mümkün kılar (Mier ve diğ., 2023).

Grid tabanlı yaklaşımlar ise ortamı eşit boyutlu hücrelere bölerek, hesaplama açısından daha verimli ve esnek çözümler sunar. Fields2Cover gibi kütüphaneler, bu yöntemleri modüler bir yapıda sunarak kapsama yollarını planlama ve optimizasyon için esnek araçlar sağlar (Mier ve diğ., 2023). Özellikle Boustrophedon desenleri ve Dubins eğrileri

gibi yolları kullanan bu sistemler hem alanın tam kapsanmasını hem de gereksiz hareketlerin azaltılmasını mümkün kılar (Mier ve diğ., 2023), (Pérez-González ve diğ., 2021).

Klasik kapsama yol planlama yöntemlerinin teknik açıdan öne çıkma sebepleri, düşük hesaplama maliyetleri, deterministik ve öngörülebilir sonuçlar sunmalarıdır. Özellikle önceden bilinen çevrelerde, bu algoritmaların sunduğu güvenilirlik ve basitlik, dinamik ve karmaşık ortamlarda daha ileri yöntemler geliştirilmesi için bir temel oluşturur (Carvalho & Aguiar, 2023). Ayrıca, açık kaynaklı platformlarla kolay entegrasyonu ve modüler yapısı, bu yöntemlerin robotik uygulamalardaki geçerliliğini daha da artırmaktadır (Mier ve diğ., 2023).

Klasik tam kapsama yol planlama algoritmalarının bir diğer örneği, Trapezoidal ayrıştırma ve Grid tabanlı ayrıştırma yöntemleridir. Bu yöntemler, alanı daha küçük ve düzenli hücrelere bölerek, robotların her noktayı sistematik bir şekilde ziyaret etmesine olanak tanır. Özellikle, robotik temizleme ve tarım uygulamalarında sıklıkla kullanılan bu teknikler, gereksiz dönüşleri en aza indirerek operasyonel verimliliği artırır. Ancak, bu yöntemler genellikle statik haritalar ve önceden bilinen çevrelerde etkili olup, dinamik ortamlar için sınırlı bir uygulama sunar (Lu ve diğ., 2023).

Son yıllarda, yapay zekâ ve öğrenme tabanlı yöntemler klasik tekniklere kıyasla daha karmaşık ve dinamik ortamlar için önemli avantajlar sağlamıştır. Biyolojik esinli sinir ağı algoritmaları (BINN), robotların çevresel verileri işleyerek engellerden kaçınmalarını ve kapsama yollarını optimize etmelerini sağlar. Bu algoritmalar, özellikle karmaşık çevresel yapılar için daha esnek ve enerji verimli çözümler sunar (Cai ve diğ., 2014).

Enerji verimliliği ve hesaplama maliyetleri açısından ise, iyileştirilmiş A* algoritmaları gibi yöntemler öne çıkmaktadır. Bu algoritmalar, hareket maliyetlerini hesaba katarak en kısa ve en az enerji harcayan yolu belirler. Özellikle robotik temizleme ve lojistik operasyonlarında etkin bir şekilde kullanılmaktadır (Cai ve diğ., 2014b).

Bu çalışmalar, kapsam yol planlama algoritmalarının geniş bir uygulama yelpazesi için etkili çözümler sunduğunu göstermektedir. Gelecekteki araştırmalar, bu algoritmaların daha karmaşık ve dinamik ortamlara uyarlanabilirliğini artırmayı hedeflemelidir.

1.3. Önerilen Yöntem

Önerilen yöntem, boustrophedon ve yayılan ağaç kapsama (STC) (Spanning Tree Coverage) stratejilerinin birleşimine dayanan hibrit bir kapsama planlaması yaklaşımıdır. Bu iki yöntemin bir araya gelmesi, kapsama planlamasında hem düzenli alanlar hem de engellerle dolu karmaşık topolojiler için etkili bir çözüm sunmayı hedefler. Literatürde bu iki yöntem genellikle ayrı ayrı ele alınmış olmakla birlikte, yöntemlerin birleştirilmesi fikri, alanın topolojik özelliklerini göz önünde bulundurmaya ve kapsama esnasında esneklik sağlamaya amaçlayan çalışmalarla ilişkilendirilebilir.

Boustrophedon yöntemi, kapsanacak alanı yatay veya dikey şeritlere bölerek robotun bu şeritler üzerinde ileri-geri hareket etmesini sağlar. Bu yöntemin temel avantajı, basit ve düzenli alanlarda yüksek bir verimlilik sağlamasıdır. Ancak engelli ve karmaşık haritalar söz konusu olduğunda, boustrophedon yöntemi eksiksiz kapsama sağlayamayabilir. Bu bağlamda, daha esnek bir kapsama stratejisi sunan STC yöntemi devreye girmektedir. STC, alanı topolojik olarak anlamlı alt bölgelere ayırarak her bölge için bir yayılan ağacı (spanning tree) oluşturur. Bu ağaçlar, robotun karmaşık ve engellerle dolu alanlarda dahi kapsama yapmasını sağlar. STC'nin bu yönü, alan içinde karşılaşılabilecek çıkmaz durumlarına veya engellerin oluşturduğu kısıtlamalara alternatif yollar sunarak üstün bir kapsama başarımı sağlamaktadır.

Önerilen yöntem, boustrophedon ve STC yaklaşımlarının avantajlarını birleştirerek hibrit bir yapı sunar. Öncelikle alan, boustrophedon yöntemiyle düzenli şeritlere ayrılır; sonrasında, her bir şerit üzerinde kapsama stratejisi geliştirir. Bu sayede, yöntem hem basit hem de karmaşık topolojilere uyum sağlama potansiyeline kavuşur. Literatürde, bu iki yöntem genellikle ayrı başlıklar altında ele alınırken, hibrit yaklaşım her iki yöntemin güçlü yönlerini bir araya getirir. Örneğin, boustrophedon yöntemi, kapsama planlamasında Morse teorisi gibi topolojik analiz araçları ile ilişkilendirilmiş; STC ise, sürekli alanların kapsanmasında bir yayılan ağacı (spanning tree) yaklaşımını temel alan çalışmalarda öne çıkmaktadır.

Özellikle son yıllarda bu iki yöntemin kombinasyonunu içeren hibrit yaklaşımlar, karmaşık ve değişken ortamların yanı sıra daha fazla esneklik ve adaptasyon gerektiren uygulamalarda öne çıkmıştır. Önerilen yöntemin literatürdeki konumu, kapsama

planlaması alanında tanınan iki temel yöntemin güçlü yönlerini bir araya getirerek daha etkin bir strateji oluşturulmasıyla belirginleşmektedir. STC algoritması, sürekli alan kapsama problemini çözmek için hem çevrimdışı hem de çevrimiçi yöntemler sunarak, optimal kapsama yollarını düşük zaman ve bellek karmaşıklığında üretmiştir. Çevrimdışı STC, önceden tanımlanmış bir ortamda optimal kapsama sağlar, çevrimiçi STC, harita bilgisi olmadan sensör verilerine dayalı olarak çalışır ve karınca tipi STC, minimum bellek kullanımı ile tam kapsama gerçekleştirir. Bu algoritmalar, kompleks alanlar ve değişken ortam koşulları için etkili çözümler sunmaktadır. Bu yaklaşım, robotik sistemlerin operasyonel etkinliğini artırmayı ve mevcut yöntemlerin sınırlamalarını gidermeyi hedeflemektedir.(Ranjitha & Guruprasad, 2016)

2. MATERYAL VE METODLAR

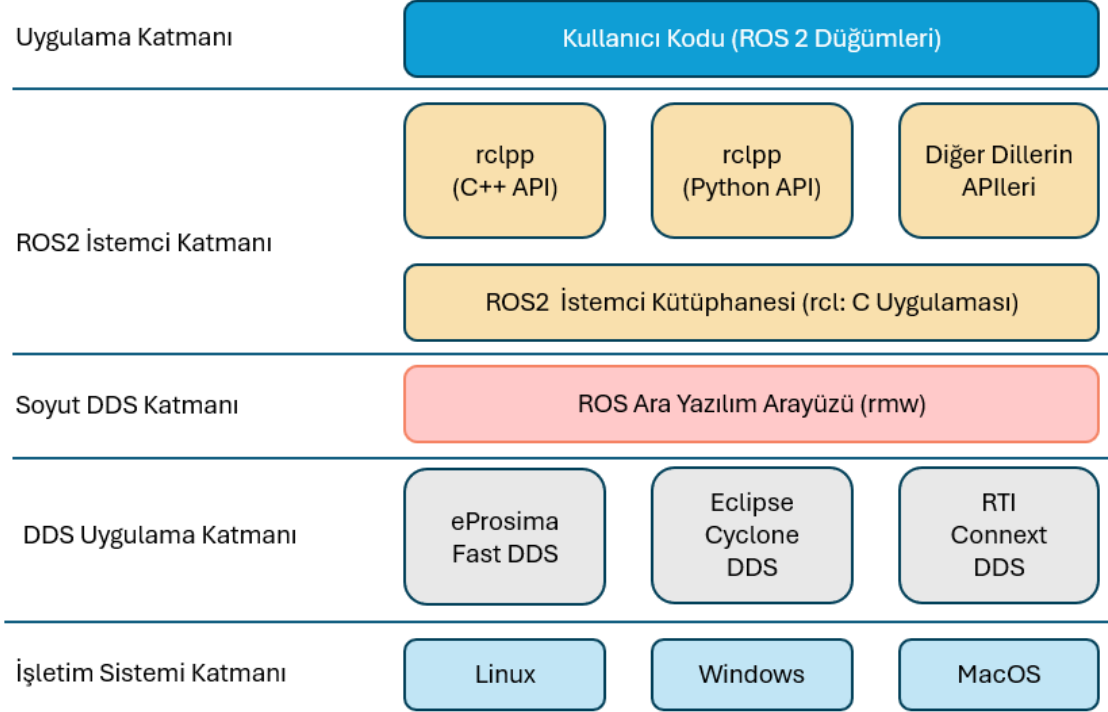
2.1. ROS (Robot İşletim Sistemi)

Robot İşletim Sistemi (ROS), robotik sistemlerin geliştirilmesi ve entegrasyonu için kullanılan açık kaynaklı bir yazılım çerçevesidir (URL-1). ROS, robot yazılımlarının karmaşıklığını yönetmek için bir araç ve kütüphaneler bütünü sunarak, modüler ve genişletilebilir bir yapı sağlar. Temel amacı, robot yazılımı geliştiren araştırmacılar ve mühendisler için bir standart platform oluşturmaktır. ROS, dağıtık bir yapı sunarak robotik sistemin farklı bileşenleri arasında iletişimi sağlar; sensörlerden gelen veriler, işleme algoritmaları ve motor kontrolü gibi robotik sistemin farklı parçaları arasında sorunsuz bir veri akışı sağlar. ROS'un bu özellikleri, robotların çok çeşitli ortamlarda esnek bir şekilde çalışmasını sağlar ve "dağıtık sistem yapısıyla robotların karmaşık görevleri gerçekleştirme kapasitesini artırır" (Quigley ve diğ., 2009). ROS ayrıca, otonom sistemlerin yerel ve küresel navigasyonunu sağlayacak çeşitli araçlar içerir (Macenski ve diğ., 2020).

ROS 2, ROS'un gelişmiş bir versiyonu olarak, özellikle daha karmaşık ve büyük ölçekli uygulamalara ihtiyaç duyan sistemler için optimize edilmiştir. ROS 2, ROS'a kıyasla gerçek zamanlı uygulamalar için daha uygun bir platform sunar. Gerçek zamanlılık desteği ROS 2'nin en önemli farklarından biri olarak kabul edilir; gerçek zamanlı işletim sistemleriyle uyumlu olması, ROS 2'nin endüstriyel robotlar gibi kritik uygulamalarda kullanılmasını sağlar (Maruyama ve diğ., 2016). Ayrıca ROS 2, Data Distribution Service (DDS) kullanarak daha güvenilir ve hızlı bir iletişim altyapısı sunar. Bu, özellikle birden fazla robotun birbiriyle koordinasyon içinde çalıştığı uygulamalarda önemli avantajlar sağlar.

ROS 2'nin katmanlı yapısı Şekil 2.1.'de açıklanmaktadır. Bu mimari, ROS 2'nin nasıl çalıştığını ve robot uygulamalarının farklı seviyelerde nasıl yapılandırıldığını gösterir. En üst katmanda, kullanıcı kodu yer almakta ve bu katman, C++ API'si (rclcpp) ve Python API'si (rclpy) gibi farklı dillerdeki istemci katmanları ile desteklenmektedir. İstemci katmanı, ROS 2'nin daha düşük seviyelerdeki DDS tabanlı mesajlaşma protokollerini soyutlayarak kullanıcının robotik uygulamaları kolayca geliştirmesini sağlar. Aynı

zamanda, ROS ara yazılım arayüzü (rmw) katmanı, farklı DDS uygulamaları ile çalışabilme esnekliği sunar.



Şekil 2.1. ROS 2 Mimarisi Genel Bakış

Bu çalışmada ROS 2 Humble dağıtımı kullanılmıştır ve ana hedef, robotun fabrika ortamındaki engellerden kaçınarak enerji verimli bir şekilde tüm alanı kapsayabilmesidir. ROS 2 Humble dağıtımının Nav 2 modülü, robotun dinamik çevrelerde engelleri algılayıp, bu engellerden güvenle kaçınarak ilerlemesine olanak tanır. Böylece, robot karmaşık ve değişken ortamlarda dahi etkin bir şekilde yol planlaması yapabilir. Nav 2 sayesinde kapsam yol planlamada robotun navigasyon yetenekleri büyük ölçüde geliştirilmiştir. ROS 2, ayrıca robotların çevre algılama, nesne tanıma ve haritalama gibi görevleri etkin bir şekilde yerine getirmesini sağlayan birçok gelişmiş araç sunar (Macenski ve diğ., 2020). Bu sayede ROS 2 Humble, robotun tüm alanı kapsaması ve görevlerini verimli bir şekilde gerçekleştirmesi için gerekli olan altyapıyı sağlar.

2.2. SLAM

SLAM, robotun hem kendi konumunu belirleyip hem de çevresinin haritasını çıkarma sürecidir. Yani, bir robot ya da insansız araç bir alan içinde hareket ederken, o alanın

haritasını eş zamanlı olarak oluşturur ve bu harita üzerinden kendi konumunu sürekli olarak günceller (URL-2). Bu süreç, robotun sürekli olarak sensör verilerini kullanarak çevresini öğrenmesi ve engellerden kaçınmasını mümkün kılar. Robotlar, bilmedikleri bir ortamda hareket ederken dinamik çevrelere anında tepki verebilir, bu sayede güvenli ve verimli bir şekilde gezinmeleri sağlanır.

Bu çalışmada, robotun SLAM işlemi kapsamında kullanılan Cartographer haritalama metodu, Google tarafından geliştirilen açık kaynaklı bir kütüphanedir (URL-3). Cartographer, özellikle LiDAR ve IMU sensörlerinden gelen verileri işleyerek son derece hassas haritalar oluşturur; özellikle odometri verilerinin kullanımı, hareket modellemesini güçlendirerek haritalama doğruluğunu artırmaktadır. Ayrıca, kütüphanenin 2D ve 3D haritalama verileri, robotun farklı çevre koşullarında etkili bir şekilde çalışmasına olanak tanımaktadır.

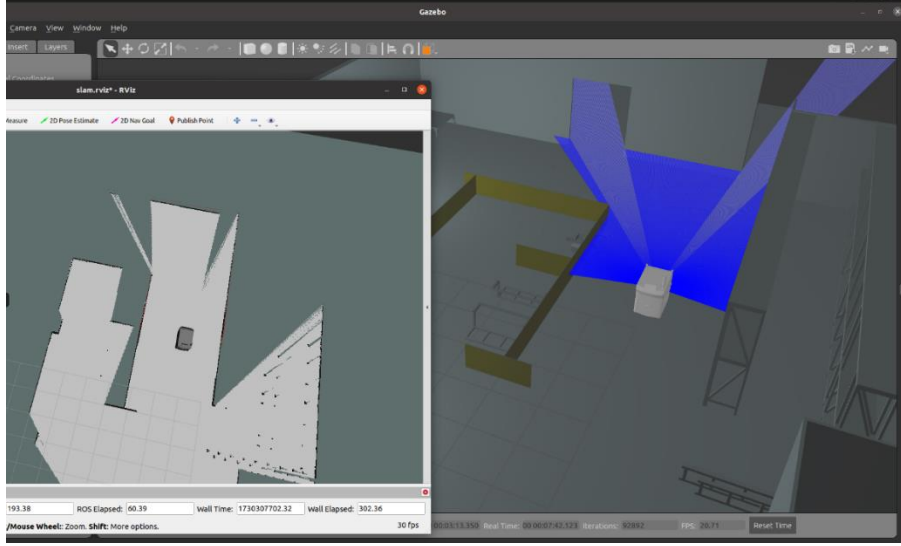
Cartographer algoritması, SLAM (Simultaneous Localization and Mapping) süreçlerini kullanarak robotların çevresel verilerden hem haritalama hem de lokalizasyon yapmasını sağlar. Robot üzerinde bulunan LiDAR ve IMU gibi sensörler aracılığıyla çevreden veri toplanan veriler, ortamın geometrisinin çıkarılması ve robotun pozisyonunun tahmin edilmesi için kullanılır. Algoritmanın önemli aşamalarından biri olan tarama eşleştirme algoritması, robotun LiDAR verilerini çevredeki nesnelere karşılaştırarak sürekli günceller. Ayrıca, grafik tabanlı optimization tekniği ile robotun hareket verileri grafiksel bir yapıya dönüştürülüp küresel optimizasyon gerçekleştirilir; bu yapıda düğümler, robotun belirli zamanlardaki konumlarını, kenarlar ise hareket ilişkilerini ifade eder. Loop closure tekniği sayesinde, robot daha önce geçtiği yerleri tanıyarak haritayı optimize eder ve hataları düzeltir. Bu yöntem, dinamik ve değişken ortamlarda robotun doğru bir şekilde haritalama yapmasını ve güvenilir lokalizasyon sağlamasını mümkün kılar.

Şekil 2.2, robotun sanal ortamda harita çıkarma sürecini görsel olarak sunarak, SLAM yönteminin nasıl işlediğini göstermektedir.

2.3. Robot

Bu çalışmada, Birfen Elektrik & Elektronik Ltd. Şti tarafından üretilen endüstriyel otonom temizlik amacıyla üretilmiş ve otonom navigasyon kabiliyetine sahip bir otonom

mobil robot kullanılmıştır (URL-4). Robotun görseli Şekil 2.3.'te gösterilmiştir. 600 x 900 taban alanına sahiptir. Robot üzerinde sensör olarak güvenli lazer alan tarayıcı ve derinlik kameraları bulunmaktadır.



Şekil 2.2. Cartographer Metodu ile Harita Çıkarılması



Şekil 2.3. Robot Görseli

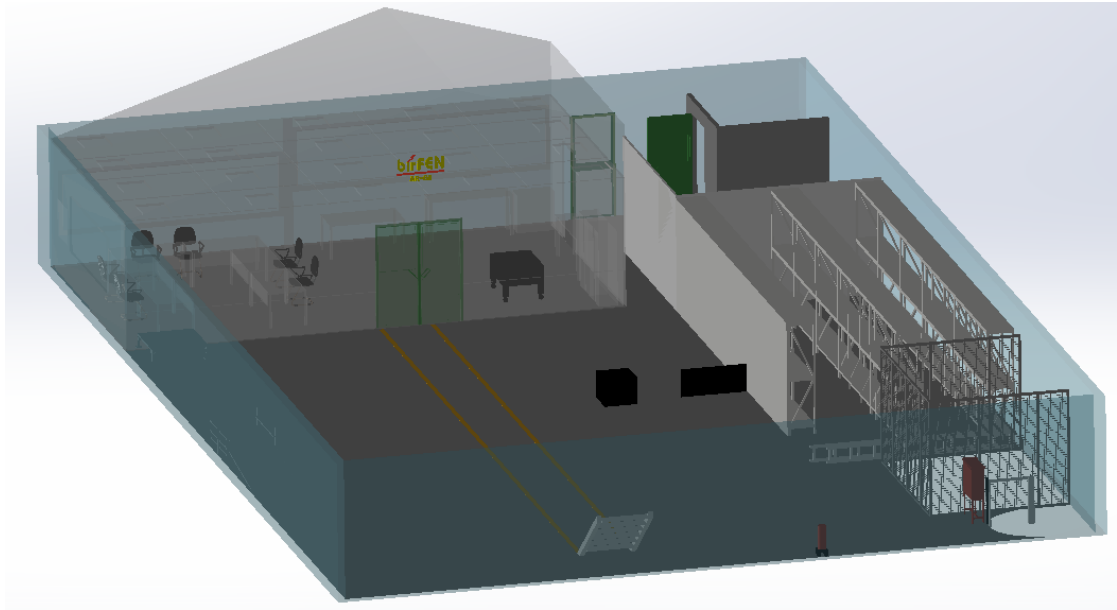
2.4. Gazebo Ortamı

Algoritma testlerinin gerçek ortam koşullarından ayrıştırılması amacıyla Gazebo simülasyon ortamı kullanılmaktadır. Gazebo, sensör verilerine istenilen miktarda gürültü

ekleyebilme özelliđi sayesinde, gerek dnya kořullarına olduka yakın bir simlasyon sunabilmektedir. Bu zellik, robotun evresel algılamalarının gerek ortamda karřılařabileceđi belirsizlikleri ve hataları ierecek řekilde modellenmesini sađlar, bu da algoritmaların daha gvenilir ve kapsamlı bir řekilde test edilmesine olanak tanır.

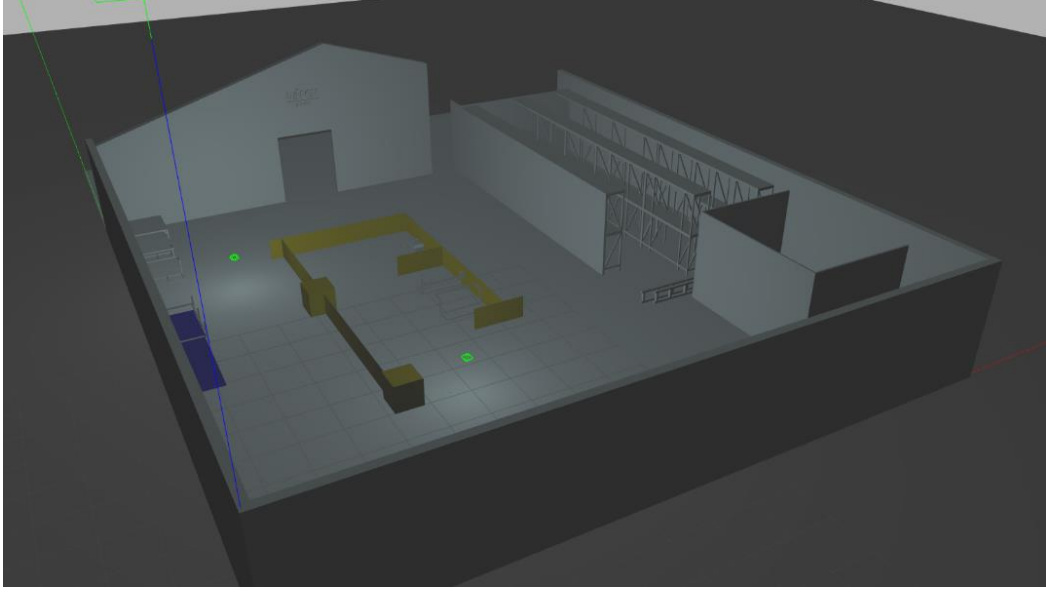
Gazebo ortamında robot donanım bileřenleri de tanımlanarak, donanım arabirimleri sanal ortamda simle edilmektedir. Bylece, donanım kaynaklı veri akıřı ve sensr tepkimeleri, fiziksel testler yapılmadan nce sanal ortamda gzlemlenebilir hale gelmektedir. Bu yaklařım, robotun farklı algılama senaryolarına ve dinamik evre kořullarına karřı gsterdiđi performansın detaylı analizini mmkn kılar.

Bu alıřmada testlerin gerekleřtirileceđi Birfen Elektrik & Elektronik'e ait atlye alanı, gerek boyutlarıyla  boyutlu olarak modellenmiřtir. Model řekil 2.4.'te gsterilmektedir.



řekil 2.4. 3 Boyutlu Test Ortamı

Daha sonra model Gazebo dosyaları oluřturularak ortamında simlasyon iin uygun hale getirilmiřtir. Ortam modelinin Gazebo dosyası řekil 2.5.'te sunulmaktadır. Bu model sayesinde, robotun alıřma alanındaki hareketleri, engellerle etkileřimi ve temizlik grevindeki verimliliđi gerek ortam kořullarına benzer řekilde deđerlendirilebilmektedir.



Şekil 2.5. Gazebo Test Ortamı

2.5. Atölye Ortamı

Testlerin gerçekleştirildiği ortam, Birfen Elektrik & Elektronik San. Tic. Ltd. Şti.'ye ait, endüstriyel bir depo alanıdır. Şekil 2.6.'da gösterilmiştir. Toplamda 13,5 metre uzunluğunda ve 11,7 metre genişliğinde olan bu alan, otonom mobil robotların navigasyon ve kapsam planlama algoritmalarının performansının değerlendirilmesi için kullanılmıştır. Depo içerisinde farklı boyutlardaki karton kutular düzensiz bir şekilde yerleştirilmiş olup, bu sayede endüstriyel ortam koşullarına benzer engeller oluşturulmuştur. Testler sırasında, robotun bu engellerle başa çıkma yetenekleri değerlendirilmiş ve gerçek dünya senaryolarına uygun veriler elde edilmiştir. Bu düzenlemeler, robotun çevresel algılama, yol planlama ve kapsamlı temizlik performansını test etmek amacıyla yapılmıştır.

2.6. Global ve Yerel Planlayıcı

ROS 2 tabanlı otonom mobil robotlarda, navigasyon sürecinde kullanılan genel ve yerel planlayıcılar, robotun belirlenen hedefe güvenli, verimli ve kesintisiz bir şekilde ulaşabilmesi için birbiriyle uyum içinde çalışan iki temel bileşendir (URL-5). Bu planlayıcıların her biri, robotun çevresindeki statik ve dinamik engellerle etkileşimini farklı düzeylerde yönetir ve hedefe ulaşmak için gerekli yönlendirmeleri sağlar. Robotun

genel rotasını ve dinamik çevreye tepkisini düzenleyen bu iki planlayıcı, çevresel verilerle etkileşime girerek robotun rotasını optimize eder.



Şekil 2.6. Birfen Atölye Ortamı

Genel planlayıcı, robotun başlangıç noktasından hedef noktaya kadar izlemesi gereken geniş çaplı güzergâhı belirler. Bu güzergâh belirleme işlemi, haritanın tamamına yayılmış bir analiz sürecini içerir ve statik bir harita üzerinde gerçekleştirilir. Statik engellerin bulunduğu bu haritada, robot için optimal bir yol seçebilmek adına A* veya Dijkstra gibi graf tabanlı algoritmalar kullanılmaktadır. Genel planlayıcının birincil amacı, harita üzerinde en kısa veya en güvenli rotayı seçerek robotun hedefe hızlı ve güvenilir bir şekilde ulaşmasını sağlamaktır. Graf tabanlı algoritmalar, genellikle haritanın tamamını kapsayan bir analiz süreci yürütür ve robotun çevresindeki statik engelleri dikkate alır. Bu sayede, robotun yol boyunca karşılaşılabileceği engellerden kaçınarak optimal rotayı oluşturur. Özellikle yapılandırılmış, kapalı ortamlar ve uzun mesafeli hareket gerektiren senaryolarda, genel planlayıcının sağladığı geniş ölçekli rehberlik, robotun hedefe ulaşması için idealdir.

Yerel planlayıcı, genel planlayıcı tarafından belirlenen ana güzergâh çerçevesinde, robotun çevresindeki dinamik engelleri dikkate alan daha kısa vadeli ve reaktif bir yol planlaması yapar. Bu planlama türü, robotun çevresindeki değişen durumlara hızla uyum sağlayabilmesi için sensör verilerine dayanmaktadır. Robotun hareket sırasında sürekli olarak çevresindeki verileri analiz eden yerel planlayıcı, aniden beliren veya hareket eden

engelleri algılayarak çarpışmaları önlemek için gereken hız ve yön değişikliklerini anında gerçekleştirir. Örneğin, robotun önüne bir insan ya da başka bir hareketli nesne çıkması durumunda, yerel planlayıcı robotun hızını düşürerek veya engelden uzaklaşmasını sağlayacak alternatif bir rota belirleyerek güvenliği sağlar. ROS 2 ortamında yaygın olarak kullanılan Dynamically Weighted Band (DWB) gibi yerel planlama algoritmaları, robotun genel planda belirlenmiş güzergâhından sapmadan güvenli bir şekilde ilerlemesine katkıda bulunur (URL-6). Bu algoritmalar, robotun çevresel değişikliklere tepki vermesini sağlarken genel planda belirlenen rotayı takip etmesine yardımcı olur.

Genel ve yerel planlayıcıların entegre bir biçimde çalışması, ROS 2 tabanlı robotlar için kesintisiz, güvenli ve verimli bir navigasyon sağlanmasında kritik bir rol oynar. Genel planlayıcı tarafından belirlenen ana güzergâh, robotun hedefe ulaşması için genel bir rehberlik sunarken; yerel planlayıcı, bu güzergâh üzerinde ilerlerken karşılaşılabileceği dinamik engelleri anlık olarak izleyip analiz ederek güvenli bir yol sağlar. Genel ve yerel planlayıcılar arasındaki bu etkileşim, robotun hem statik hem de dinamik engellerle dolu karmaşık ortamlarda daha etkin bir şekilde yönlendirilmesine olanak tanır. Bu bağlamda, genel planlayıcı, geniş ölçekli kapsama sağlarken; yerel planlayıcı ise robotun çevresinde meydana gelen anlık değişiklikleri değerlendirir ve robotun rotasında ihtiyaç duyulan uyarlamaları gerçekleştirir.

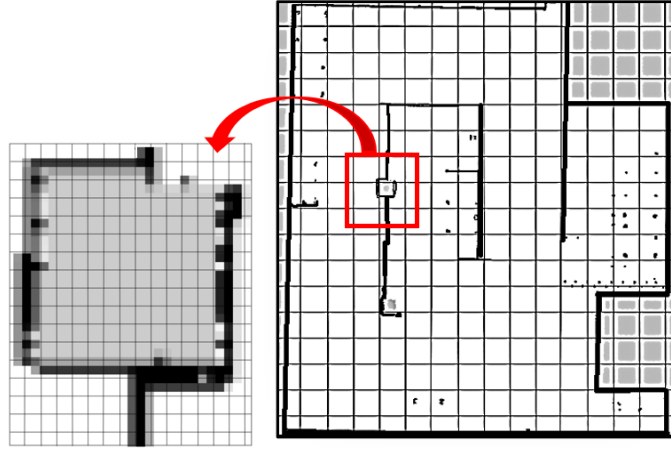
Bu iki planlayıcı arasındaki koordinasyon, ROS 2 platformunda otonom mobil robotların yüksek doğruluk ve güvenilirlikle görevlerini yerine getirmelerine imkân tanır. Örneğin, endüstriyel otomasyon, sağlık sektörü ve lojistik gibi alanlarda faaliyet gösteren robotların, karmaşık çevre koşullarında güvenli bir şekilde yönlendirilmeleri gerekmektedir. Genel planlayıcı, bu tür senaryolarda genel bir rota belirleyerek robotun hedefe ulaşmasını sağlarken, yerel planlayıcı çevresel koşullardaki değişikliklere anında yanıt verir ve robotun güvenliğini ön planda tutarak çarpışmaları engeller. Genel plan, harita üzerinde uzun mesafeli bir rehberlik sunarken, yerel planlayıcı robotun çevresindeki engellerin konumuna göre anlık tepki geliştirmesine olanak tanır. Böylelikle, robotlar hem önceden planlanmış rotalarını koruyarak ilerler hem de çevresel değişikliklere dinamik bir şekilde uyum sağlar.

ROS 2 tabanlı robotlarda genel ve yerel planlayıcıların entegrasyonu, robotların çevresel değişimlere karşı esnek ve çok katmanlı bir navigasyon stratejisi geliştirmesine olanak

tanır. Genel planlayıcı, ortamın geniş perspektifini sunarken; yerel planlayıcı, anlık çevresel değişikliklere uyum sağlayarak güvenli hareket yollarını belirler. Bu iki planlayıcının sinerjisi, robotların karmaşık ve dinamik ortamlarda optimal ve güvenilir bir şekilde ilerlemesine önemli katkılar sağlar.

2.7. Hücresel Ayrıştırma

Hücresel ayrıştırma, bir alanın daha küçük ve yönetilebilir hücelere bölünmesi yöntemi olarak tanımlanır. Bu yöntem, özellikle robotik uygulamalarda, belirli bir alanın kapsanması veya navigasyonu gibi problemlerin çözümünde yaygın olarak kullanılmaktadır (Guo ve diğ., 2022). Şekil 2.7.'de gösterildiği gibi, kapsanacak alan daha küçük, yönetilebilir hücelere bölünerek robotun izlemesi gereken yolu basitleştirmeyi hedefler. Bu teknik, geometrik olarak karmaşık yapıdaki alanlarda robotun hareket planlamasını kolaylaştırmak, enerji tüketimini optimize etmek ve engellerden kaçınarak görev başarımını artırmak amacıyla sıklıkla tercih edilmektedir.



Şekil 2.7. Hücresel Ayrıştırma

Bu hücresel ayrıştırmanın temel prensibi, hedef alanın geometrik yapısının analiz edilmesi ve robotun geçişine uygun olan bölgelerin, engellerden ve erişilemez alanlardan ayrıştırılmasıdır. Elde edilen geçişe uygun alan, belirli bir geometrik modele (örneğin kare, dikdörtgen, trapez, altıgen ya da dışbükey çokgenler) uygun hücelere bölünür. Hücrelerin boyutları ve yapısı, robotun fiziksel boyutları, manevra kabiliyeti ve görev gereksinimlerine göre belirlenir. Yaklaşık hücresel ayrıştırma ise alanı sabit boyutlu grid hücrelerine bölerek, belirli bir çözünürlükte alanın tamamını analiz etmeyi mümkün kılar.

Ancak çözünürlük arttıkça hesaplama maliyetinin artması, bu yöntemin bir dezavantajıdır.

Hücresele ayırıştırma yöntemleri, genellikle robotun enerji tüketimini optimize etmek ve karmaşık geometrilere sahip alanlarda güvenli bir kapsama sağlamak amacıyla yapılandırılır. Her bir hücre için robotun izlemesi gereken kapsama yolları; enerji tasarrufu, dönüş sayısının azaltılması ve yol uzunluğunun optimize edilmesi gibi parametreler göz önünde bulundurularak belirlenir. Hücresele ayırıştırmanın başarısı, robotun tüm hücreleri eksiksiz kapsaması ve alanın maksimum verimlilikle taranabilmesine bağlıdır.

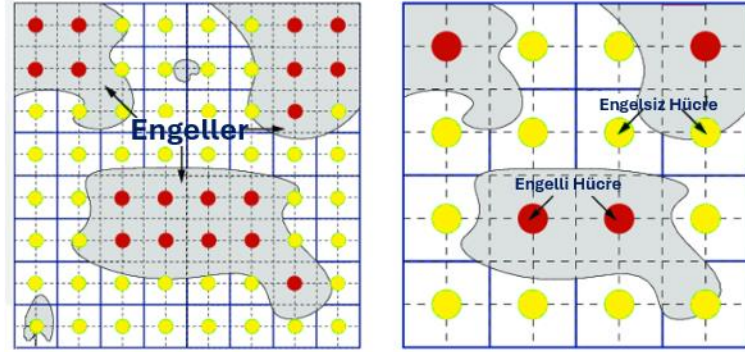
Hücresele ayırıştırma yöntemleri farklı stratejilere dayanarak çeşitlendirilmiştir ve uygulama alanlarına göre seçilmektedir. Bu yöntemler arasında serbest hücresele ayırıştırma, boustrophedon hücresele ayırıştırma, hiyerarşik hücresele ayırıştırma, kare hücresele ayırıştırma, trapez hücresele ayırıştırma, altıgen hücresele ayırıştırma, voronoi ayırıştırma, spiral ayırıştırma ve konveks poligon ayırıştırma gibi yaklaşımlar yer almaktadır. Bu yöntemler, alanın özelliklerine ve robotun görev gereksinimlerine göre seçilerek uygulanır. Her bir yöntemin kendi avantajları ve dezavantajları bulunmakta olup, bu başlıklar altında ayrıntılı şekilde incelenecektir.

Hücresele ayırıştırmanın kapsama verimliliğini artırdığı ve engellerle dolu alanlarda bile etkili olduğu yapılan çalışmalarla kanıtlanmıştır (Choset & Pignon, 1998). Ancak bu yöntemlerin uygulamalarında karşılaşılan hesaplama sürelerinin uzunluğu gibi sorunlar, optimizasyon algoritmalarının geliştirilmesiyle aşılmaktadır. Örneğin, dinamik hücresele ayırıştırma algoritmaları, hücrelerin boyutlarını duruma göre ayarlayarak hesaplama maliyetlerini önemli ölçüde düşürmeyi başarmaktadır (Cai ve diğ., 2014).

2.8. Serbest Hücresele Ayırıştırma

Serbest hücresele ayırıştırma, yapılandırma uzayının (C-uzayı) serbest bölgesini, yani robotun engellerle çakışmaksızın konumlanabileceği tüm noktaların alt kümesini, belirli topolojik ve geometrik kriterlere uygun biçimde daha küçük ve yönetilebilir hücrelere ayırma sürecine dayanmaktadır (Choset & Pignon, 1998). Şekil 2.8.'de, iki boyutlu bir çalışma alanında engellerle tanımlanmış C-uzayının serbest bölgesinin, çokgen benzeri

hücelere ayrılarak nasıl basit alt uzaylara dönüştürüldüğü gösterilmektedir. Genel olarak robotun konfigürasyonları bir C uzayı olarak tanımlanır; bu uzay, tekil bir robot için genellikle \mathbb{R}^n biçiminde, çok eklemli ya da karmaşık yapıları robotlar içinse daha yüksek boyutlu manifoldlar hâlinindedir. Engellerle tanımlanan Q_{obs} uzayı, robotun herhangi bir konum ve yönelimde engel geometrileri ile kesişime uğradığı yapılandırmalar kümesini ifade ederken, serbest uzay Q_{free} ise $Q_{free} = Q / Q_{obs}$ olarak tanımlanır. Burada Q tüm olası konfigürasyonların uzayını göstermek üzere kullanılır. Serbest hücresel ayrıştırma, temelde Q_{free} alt uzayını, topolojik olarak basit, çoğunlukla çokgen veya polieder biçimli hücrelerin ayrık bir bileşimi olarak ifade eden bir fonksiyonel ayrıştırma işlemidir. Bu yaklaşım, Q_{free} alanını sonlu sayıda hücreye ayırarak $Q_{free} = \bigcup_{i=1}^m C_i$ şeklinde bir örtü elde eder; bu durumda her bir C_i hücresi, engellerden arınmış, bağlantılı ve geometrik analizi kolay bir alt bölge olarak tanımlanır (Barraquand & Latombe, 1991). Bu parçalı yapı sayesinde, robotun başlangıç konfigürasyonu $q_{start} \in Q_{free}$ ile hedef konfigürasyonu $q_{goal} \in Q_{free}$ arasında var olan herhangi bir yol, hücresel ayrıştırma sonucu elde edilen komşuluk grafi üzerinden belirlenebilir.



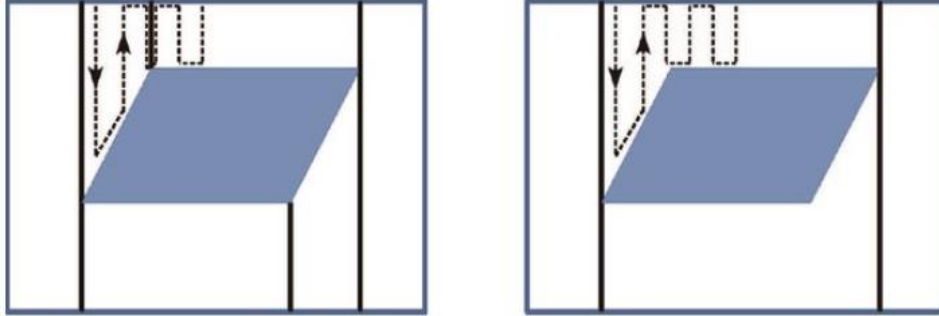
Şekil 2.8. Serbest C-Uzay Hücresel Ayrıştırma ile Alt Uzaylara Bölünmesi

Söz konusu komşuluk grafi, hücreler arası geçişlerin düğümler ve kenarlar düzeyinde modellenmesiyle oluşturulur: Eğer hücreler arasındaki geçiş koşulları sağlanabiliyorsa, bu durum graf üzerinde bir kenar ile temsil edilir. Böylece, karmaşık geometri problemleri, ayrık bir grafik arama problemi haline dönüştürülerek, örneğin A^* ya da Dijkstra gibi standart arama algoritmalarının kullanımı mümkün olur (Bähmann ve diğ., 2019). Ayrıca, hücrelerin boyutları (örneğin hücre kenar uzunluğu δ) sistematik biçimde ayarlanarak hem hesaplama maliyetini hem de elde edilen yolun doğruluğunu dengelemek mümkündür. Daha küçük hücreler, daha ayrıntılı bir yapılandırma sunarken,

hesaplama süresini artırır; daha büyük hücreler ise basitleştirilmiş bir serbest alan yaklaşımı sunsa da bazı dar geçişlerin veya kritik yapılandırmaların atlanmasına neden olabilir. Böylelikle, serbest hücresel ayrıştırma, endüstriyel robotik uygulamalardan, otonom araçların çevresel farkındalığına kadar birçok alanda, serbest hareketin daha sistematik, güvenilir ve ölçeklenebilir bir biçimde ele alınmasına imkân tanımaktadır.

2.8.1. Boustrophedon Hücresel Ayrıştırma

Boustrophedon hücresel ayrıştırmayı, temel olarak bir robotun iki boyutlu bir alanı eksiksiz ve tarla sürme benzeri şekilde sistematik biçimde taraması için geliştirilen bir stratejidir. Şekil 2.9.'da gösterildiği gibi bu yaklaşım, incelenen çalışma uzayını, robotun üzerinde hareket edebileceği uygun şekilli küçük bölgelere, yani hücrelere, bölerek bir yol planı oluşturmaya dayanır (Choset & Pignon, 1998). Burada amaç, robotun serbest bölgede herhangi bir engelle çakışmadan hareket edebilmesi için alanın karmaşık geometrisinin cebirsel-topolojik yöntemlerle anlaşılır, yönetilebilir ve hesaplanabilir parçalara ayrılmasıdır.



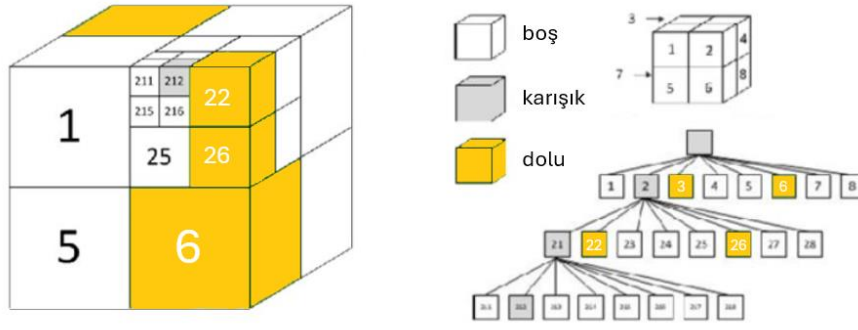
Şekil 2.9. Boustrophedon Hücresel Ayrıştırma

Temel fikir, çalışma alanının dikey doğrultuda adım adım “taranması” ile topolojik yapısının incelenmesidir. Temel fikir, çalışma alanının dikey doğrultuda adım adım taranması ile topolojik yapısının incelenmesidir. Bu işlem genellikle bir izdüşüm fonksiyonu $\phi = W \rightarrow \mathbb{R}$ yardımıyla gerçekleştirilir. Bu izdüşüm, düzlemdeki herhangi bir noktanın x eksenindeki konumunu dikkate alarak, alanın her x değerine karşılık gelen kesitini, yani bir şeritini tanımlar. Böylece, çalışma alanının serbest bölgesi her sabit x değeri için $F_x = \{y \in \mathbb{R} \mid (x, y) \in W\}$ şeklinde bir aralığa indirgenir. Farklı x değerleri boyunca bu aralıkların bağlantı sayılarının (bir ya da birden fazla kesintisiz

aralık gibi) nasıl deđiřtiđi, alıřma alanının topolojik yapısı hakkında bilgi verir. Bu deđiřim noktaları, kritik deđerler olarak adlandırılır ve alanın kesit topolojisinin deđiřtiđi konumları gsterir (Choset & Pignon, 1998). Bu kritik noktalar, alanı, her birinde kesit topolojisi sabit kalan alt aralıklara bler. Bylece, alıřma uzayı $W = \cup_i C_i$ řeklinde, boydan boya gidip gelerek (boustrophedon adı da bu ift ynl tarama tekniđinden gelir) taranabilecek hcrelere ayrılmıř olur. Her bir hcre, sınırları belirli, topolojik olarak basit bir yapıya sahip bir blgedir. Robot, bu hcreleri birer birer geerek alanı taradıđında, tm serbest blgeyi eksiksiz ve sistematik biimde ziyaret etmiř olur. Bu yaklařım, alan ne kadar karmařık olursa olsun, robotun gidiř-dnř hareketleri kullanarak tm alanı dzenli bir řekilde kaplamasını sađlar. Bylece Boustrophedon Hcresel Ayrıřtırma, rota planlama, kaplama ve haritalama konularında son derece iřlevsel ve sayısal olarak etkin bir ara sunar.

2.8.2. Hiyerarřık Hcresel Ayrıřtırma

Hiyerarřık Hcresel Ayrıřtırma, karmařık yapılı alıřma alanlarını ok katmanlı, ynetilebilir paralara ayırarak yol planlama, haritalama ve kapsama problemlerine verimli zmler sunan bir tekniktir. Bu yaklařım, řekil 2.10.'da grselleřtirildiđi gibi, alıřma alanını nce kaba bir lekle sınırlı sayıdaki hcrelere bler; ardından, her bir hcre iin ihtiya duyulan detay seviyesine gre daha ince bir ayrıřmaya gidilerek alt seviye hcreler oluřturulur. Bylece, en st seviyede elde edilen geniř kapsamlı bir harita, alt seviyelere dođru gidildike daha karmařık engeller, geiř yolları ve topografik ayrıntıların belirginleřtiđi katmanlı bir yapı hline gelir. Bu hiyerarřık yapıda, yol planlama algoritmaları nce yksek seviyede genel bir rota planlayabilir, ardından alt seviye hcrelere inerek daha ince lekli, lokal optimizasyonlar gerektiren detayları zebilir (Barraquand & Latombe, 1991). Aynı zamanda, bu yaklařım leklenebilirliđiyle dikkat eker; alıřma alanı ne kadar byr ya da karmařıklařırsa karmařıklařsın, ok katmanlı bir hiyerarřık strateji, hesaplama maliyetlerini kontrol altında tutmaya ve deđiřen kořullara uyum sađlamaya olanak tanır (Bhnmann ve diđ., 2019). Bylece Hiyerarřık Hcresel Ayrıřtırma, otonom sistemlerin deđiřken ortamlarda gvenilir, esnek ve etkin bir biimde grev icra etmesine imkn veren gcl bir yntem olarak ne ıkar.



Şekil 2.10. Hiyerarşik Hücresel Ayrıştırma

2.8.3. Kare Hücresel Ayrıştırma

Kare Hücresel Ayrıştırma, bir çalışma alanının konuma göre düzenli bir grid yapısında basit bileşenlere ayrılmasıyla elde edilen bir yöntemdir. Bu yaklaşım, özellikle robotik ve otonom sistemlerde, karmaşık çevresel özellikleri standart boyutta kare hücrelere bölerek temsili basitleştirme, hareket planlama ve engellerin konumlandırılması açısından pratik bir çözüm sunar. Grid tabanlı temsil, her bir hücrenin serbest veya engelli alan olarak etiketlenmesini olanaklı kılar; böylece mekânın topolojik karmaşıklığı, belirli bir çözünürlük seviyesinde, hesaplanabilir ve güncellenebilir bir formda ifade edilir. Bu durum, yolu tıkanmış veya serbest hücrelerin bir haritasının çıkarılması, rota belirlemede grafik tabanlı arama algoritmalarının (A*, Dijkstra, vb.) kolayca uygulanması ve ek yapısal bilgilerin (eğim, zemin özellikleri, enerji gereksinimi) hücrelerle ilişkilendirilmesi için elverişli bir zemin hazırlar (Bähnemann ve diğ., 2019).

Kare hücrelerle elde edilen eş boyutlu grid yapısı, açıkça tanımlanmış bir mekânsal ayrışma sağlar. Her bir kare hücrenin basit geometrik karakteri, matris veya çok boyutlu dizi gibi veri yapıları aracılığıyla depolanıp işlenebilmesini kolaylaştırır. Bu durum, otomatikleştirilmiş haritalama, konumlandırma, çarpışma önleme ve kapsama planlama gibi problemler için de kritik önemdedir. Örneğin, bir mobil robot, sensör verilerini işleyerek çevresini düzenli aralıklarla güncellenen bir grid haritasına dönüştürebilir ve daha sonra bu harita üzerinden güvenli bir yol bulmak için standart arama yöntemlerini devreye sokabilir. Bununla birlikte, bu yöntem sabit bir çözünürlük seviyesine bağlı kaldığından, çevrenin son derece ince detaylarını yakalamak, çözünürlüğü artırma gerekliliği doğurur. Bu da grid boyutunun büyümesiyle orantılı olarak hesaplama ve bellek maliyetlerini yükseltir. Buna karşın, modern hesaplama kaynaklarının bolluğu ve

bellek kapasitesinin genişlemesi, karmaşık veya dinamik ortamlar için dahi kare hücresel ayrıştırmanın, güvenilir, kolay uygulanabilir ve yüksek oranda yeniden yapılandırılabilir bir strateji olarak tercih edilmesine imkân tanır.

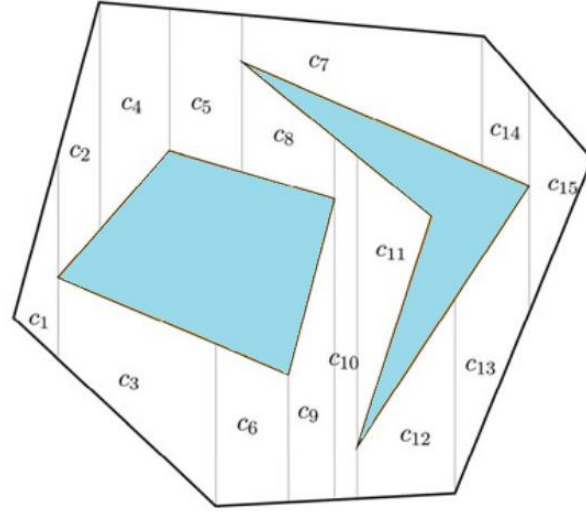
2.8.4. Trapez Hücresel Ayrıştırma

Trapez Hücresel Ayrıştırma, bir çalışma alanının topolojik ve geometrik özelliklerini düzenli, dörtgen hücrelere ayırarak biçimlendiren bir yaklaşım olup, özellikle robotik yol planlama, haritalama ve kapsama alanlarında etkin şekilde kullanılmaktadır. Bu yöntemde, Şekil 2.11.'deki gibi genellikle çokgen biçimli bir çevrenin içinden geçen dikey veya yatay doğrular yardımıyla alan, yamuk benzeri hücrelere bölünür. Her bir hücre, belirli bir çözünürlükte çevresel engellerin, açık alanların veya geçiş imkânlarının temsilini sadeleştirmeye imkân tanır. Bu temsil, bir yandan arama ve optimizasyon algoritmalarının uygulanmasını kolaylaştırırken, diğer yandan da topolojik açıdan karmaşık yapıları, sayısal olarak işlenebilir ve güncellenebilir bir forma dönüştürür. Özellikle serbest bölgenin poligonal sınırlarını kontrol altında tutan, düzenli ve biçimsel bir ayrışma elde edilmesi hem konumlandırma hem de rota belirleme süreçlerinde ciddi bir avantaj sağlar. Sayısal hesaplamanın ve veri yapısı yönetiminin görece basitliği, trapez hücresel ayrıştırmayı pek çok otonom sistem ve insansız araç uygulamasında yaygın bir yöntem hâline getirmektedir (Bähnemann ve diğ., 2019). Böylelikle, karmaşık mekânların topolojik çözümlemesi, trapezoidal biçimli hücrelerle sağlanan sistematik bir çerçeve dâhilinde hızla gerçekleştirilmekte; bu da dinamik ve öngörülemeyen çevrelere uyum sağlamak isteyen robotik platformlar için önemli bir stratejik altyapı sunmaktadır.

2.8.5. Altıgen Hücresel Ayrıştırma

Altıgen Hücresel Ayrıştırma, çalışma alanlarının Şekil 2.12.'deki gibi altıgen hücrelere bölünmesi yoluyla topolojik ve geometrik karmaşıklıkların yönetilebilir bir formda temsil edilmesine imkân tanıyan bir yöntemdir. Bu yaklaşım, özellikle robotik ve otonom sistemlerde, daha verimli kapsama planlaması ve rota belirleme stratejilerinin geliştirilmesi için uygulanır. Kare ya da dikdörtgen hücrelere dayanan grid tabanlı ayrıştırmalara kıyasla, altıgen hücre yapısı, her bir hücrenin daha fazla komşu hücreye sahip olmasına olanak tanır; bu, bağlantı sayısının artmasıyla mekânın topolojik bütünlüğünü daha zengin biçimde yakalamayı mümkün kılar. Böylece, hareket planlama

algoritmaları, alanın topolojisini daha ince detaylarla ilişkilendirerek, hesaplamalı olarak etkin çözümlere ulaşabilir (Almadhoun ve diğ., 2019)



Şekil 2.11. Trapez Hücresel Ayırıştırma

Altıgen biçimli hücreler, kesintisiz bir örtü sağlama, düzgün yakınsar kenar yapısı ve açısal boşlukları daha verimli temsil edebilme özellikleriyle öne çıkar. Bu yapının sunduğu izotropik yaklaşım, özellikle sensör verilerinin bütünleştirilmesinde, konum güncellemelerinde ve lokal rota iyileştirme algoritmalarında performansı artırır. Ayrıca, altıgen hücresel ayırıştırma, geleneksel dörtgen grid temsillerine göre belirli yönlerde oluşan menzıl hatalarını azaltarak kapsama planlaması, alan taraması ve engel tespiti gibi görevlerde daha tutarlı ve kestirilebilir sonuçlar elde edilmesini sağlar. Sonuç itibarıyla, altıgen hücresel ayırıştırma, çok yönlü topolojik bağlantıları ve dengeli hücre geometrisi sayesinde, karmaşık çevrelerde görev yapan otonom sistemlerin güvenilir, uyarlanabilir ve verimli planlama stratejileri geliştirmesine katkıda bulunan etkili bir modelleme çerçevesi sunmaktadır.

2.8.6. Voronoi Hücresel Ayırıştırma

Voronoi Hücresel Ayırıştırma, bir çalışma alanının belirli çekirdek noktalar veya çekirdek bölgeler etrafında hücreleştirilmesi temeline dayanan bir yöntemdir ve özellikle robotik, haritalama, konumlandırma ve kapsama problemlerinde yaygın biçimde kullanılır. Bu yaklaşım, önceden seçilmiş veya belirlenmiş nokta kümelerini temel alarak alanı, her noktanın etkisinin kendi yakın çevresinde en baskın olduğu bölgeler şeklinde parçalara

ayırır; böylelikle, alan topolojik olarak anlamlı bir biçimde bölünmüş, hesaplanabilir ve düzenlenebilir bir forma bürünür. Her bir Voronoi hücresi, kendi çekirdek noktasına uzaydaki herhangi bir diğer çekirdek noktasından daha yakın olan tüm noktaları içerdiğinden, bu ayrıştırma, yakınlık tabanlı bir ayrışma sunar. Özellikle robotik uygulamalarda, sensör verilerinin bütünleştirilmesi, engellerin tespiti ve hareket planlaması süreçlerinde, bu bölgeler üzerinde arama, gezinme veya kapsama algoritmaları kolaylıkla yürütülebilir. Voronoi hücresel ayrıştırmanın bir diğer avantajı, dinamik veya belirsiz çevre koşullarına uyum sağlayabilmesi ve yeniden hesaplanabilmesi, böylece otonom sistemlerin esneklik, etkinlik ve kararlılık gerektiren görevler için güvenilir bir temel sunmasıdır(Bähnemann ve diğ., 2019c). Bu nedenlerle Voronoi hücresel ayrıştırmanın, mekânsal verinin topolojik özelliklerini verimli bir şekilde yakalayarak otonom robotlardan coğrafi bilgi sistemlerine kadar geniş bir yelpazede uygulama alanı bulmaktadır (Şeda, t.y.).

2.8.7. Spiral Hücresel Ayrıştırma

Spiral Hücresel Ayrıştırma, bir çalışma alanının merkezden dışarıya doğru ilerleyen sarmal yapıda alt bölgelere ayrılması esasına dayanan bir yöntemdir. Bu yaklaşım, özellikle kaplama, keşif ve arama görevlerinde, bir robotun ya da otonom sistemin hedef bölgeyi sistematik, hatasız ve tekrarlar olmadan taramasını kolaylaştırmak amacıyla kullanılır. Spiral formdaki hücreleşme, alanın orta noktasından başlayarak, giderek büyüyen bir çember ya da sarmal biçiminde genişleyen hücreler aracılığıyla topolojik kontrol sağlar. Bu sayede, alanın her bir parçasına en az bir kez ulaşılacağı garanti altına alınır ve robotik sistemler, karmaşık geometriye sahip mekânlarda dahi temiz, anlaşılır bir hareket planlaması kurgulayabilir.

Spiral Hücresel Ayrıştırmanın en belirgin avantajı, çizgisel ya da grid tabanlı yaklaşımlara kıyasla merkezden dışarıya doğru yayılma stratejisi ile kesintisiz bir kapsam sunmasıdır. Bu sarmal biçimli yapı, çevredeki engeller, engebeler veya öngörülemeyen durumlar söz konusu olduğunda, bu engelleri dıştan dolaşma ve geride boşluk bırakmaya imkânı tanır. Ayrıca, spiral düzen, sensör verilerinin aşamalı ve kontrollü bir biçimde işlenmesine olanak vererek haritalama, konum belirleme ve kapsama görevlerinde daha yalın bir veri bütünleştirme sürecini destekler. Sonuç olarak, Spiral Hücresel Ayrıştırmanın, özellikle düzenli bir kaplama rutini gerektiren uygulamalarda, otonom

sistemlerin çevrelerini planlı, eksiksiz ve ölçeklenebilir bir şekilde analiz etmelerine olanak sağlayan güçlü bir topolojik çerçeve sunmaktadır (Yue & Jiang, 2018).

2.8.8. Dışbükey Poligon Hücresel Ayırıştırma

Dışbükey Poligon Hücresel Ayırıştırma, çokgen biçimli çalışma alanlarının dışbükey alt bileşenlere ayrılması yoluyla topolojik basitleştirme ve yol planlaması açısından avantaj sağlayan bir yöntemdir. Bu yaklaşımın temelinde, karmaşık yapıdaki bir ortamın dışbükey poligonlara bölünmesi, bu sayede her alt bölgenin geometrik ve topolojik açıdan daha yönetilebilir olması yer alır. Dışbükeylik, elde edilen her hücrenin, konveksite özelliği gereği, herhangi iki noktasını birleştiren doğru parçasının her zaman hücre içinde kalmasını sağlar. Bu durum, özellikle robotik navigasyon, engel aşma, hareket optimizasyonu ve haritalama uygulamaları için güvenli ve etkin bir temel oluşturur (Li ve diğ., 2007).

Dışbükey Poligon Hücresel Ayırıştırmanın en önemli avantajlarından biri, her bir dışbükey hücrede yol planlama algoritmalarının karmaşıklığının belirgin şekilde azalmasıdır. Örneğin, dışbükey bir hücrede, bir robot için herhangi iki nokta arasındaki en kısa yol, engellere takılmadan düz bir doğru parçası olarak elde edilebilir. Bu sayede, her hücre içinde basitleştirilmiş stratejiler, yüksek seviyeli planlamaya entegre edilerek, çalışma alanının genelinde verimli bir navigasyon mekanizması kurulabilir. Ayrıca, dışbükey hücrelerin birleştirilmesiyle elde edilen hiyerarşik planlama yapıları, değişen koşullar altında güncellemeye ve yeniden yapılandırmaya elverişlidir. Böylece, Konveks Poligon Hücresel Ayırıştırma hem istikrarlı hem de esnek bir temsil sunarak, değişken veya öngörülemeyen ortamlarda dahi otonom sistemlerin güvenilir, ölçeklenebilir ve etkili bir biçimde görev icra etmesine olanak tanır.

2.9. Kapsama Yol Planlama

Kapsama yol planlama, bir çalışma alanının her noktasının en az bir kez ziyaret edileceği bir hareket stratejisinin tasarlanmasıdır. Bu noktada hücresel ayırıştırma, kapsama yol planlama sürecinin ön hazırlık aşamalarından birini oluşturur. Hücresel ayırıştırma, çalışma alanının topolojik ve geometrik karmaşıklığını azaltmak amacıyla bu alanı belirli boyut ve biçimlerde hücrelere ayırır ve böylece homojen, hesaplanabilir ve işlenebilir

parçalara dönüştürür. Bu temsil düzeneği kapsama yol planlama için gerekli altyapıyı hazırlar, ancak tek başına bir sonuca varmaz. Diğer bir deyişle, hücresele ayrıştırma yalnızca alanın kavramsal haritalandırılmasını, düzenlenmesini ve ifade edilmesini sağlarken kapsama yol planlama, oluşturulan bu hücresele harita üzerinde alanın tümünü eksiksiz biçimde kapsayacak bir rota ya da tur tasarlama problemidir. Dolayısıyla kapsama yol planlama, hücresele ayrıştırma aracılığıyla elde edilen temsilin üzerine inşa edilen, pratikteki asıl yönlendirme, strateji geliştirme ve etkin görev icra etme sürecini ifade eder (Neumann ve diğ., t.y.).

Bu bağlamda, alanda birçok farklı kapsama yol planlama tekniği geliştirilmiştir. Boustrophedon Yol Planlama ve bunun özel bir uygulaması olan Boustrophedon STC Yol Planlama, alanı satır-sütun benzeri bir yapı içinde kesintisiz bir şekilde dolaşma mantığı üzerine kuruludur. Izgara Tabanlı Gezgin Satıcı Problemi ile Kapsama Yol Planlama, grid yapısına dayalı bir alan temsili kullanırken, en kısa turu bulma mantığını kapsama problemine uyarlayarak etkin bir strateji sunar. Sınır Ağı Tabanlı Kapsama Yol Planlama, yapay zekâ yöntemleriyle kapsama hedefini bir öğrenme sürecine dönüştürerek otonom sistemlere adaptasyon yeteneği kazandırır. Dışbükey Sensör Yerleştirme Kapsama Yolu Planlaması ise, sensörlerin yerleştirilmesi ve bu sensör alanlarının kapsanması sorunsalını dışbükeylik esaslarıyla çözümler.

Izgara Tabanlı Yerel Enerji Minimizasyonu Kapsama Yol Planlama, enerjinin etkin kullanımını merkeze alarak robotun hareket masraflarını minimize etmeye odaklanır. Kontur Hattı Tabanlı Kapsama Yol Planlama, alanı kontur hatları üzerinden bölerek, topografik ya da mekânsal değişkenlikleri daha verimli bir şekilde göz önünde bulundurur. Son olarak, Enerji Fonksiyonel Optimizasyonu Yöntemi ile Kapsama Yol Planlama, karmaşık optimizasyon problemlerini enerjisel fonksiyonlar üzerinden formüle ederek hem kapsama hem de maliyet dengesini kontrol altına alır. Bu çeşitlilik, kapsama yol planlama süreçlerinde çok yönlülük, uyarlanabilirlik ve farklı performans kriterlerine göre özelleştirme olanağı sağlayarak, pek çok uygulama alanında esnek ve etkin çözümler üretmeye imkân tanır.

2.9.1. Boustrophedon Kapsama Yol Planlama

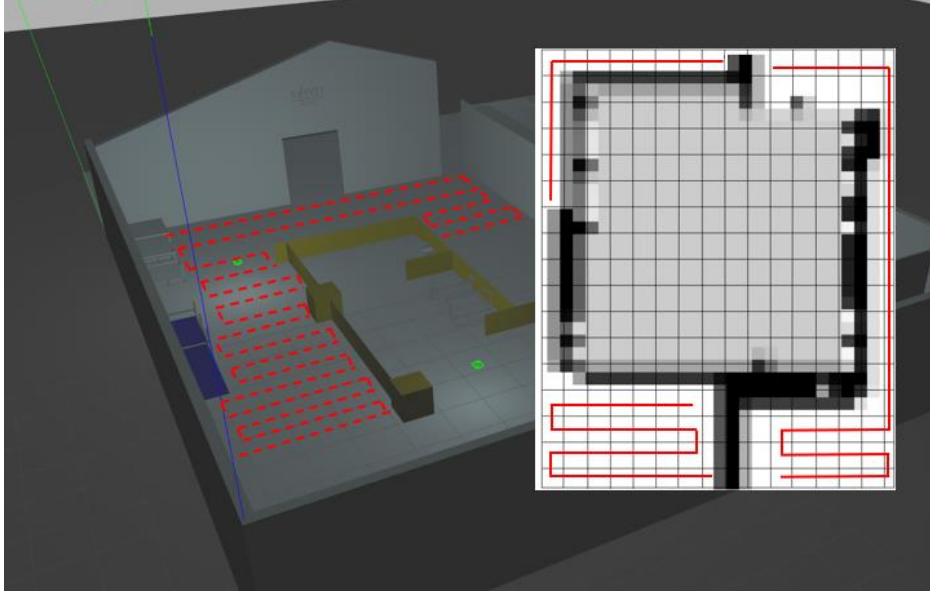
Boustrophedon kapsam yol planlama, bir robotun belirli bir alanı tamamen kaplayacak şekilde bir yol izlemesini sağlamak için kullanılan bir algoritmadır. Boustrophedon hücresel ayrıştırma, robotun alanı kesintisiz yatay ve dikey çizgilerle taramasını sağlar. Bu yöntemde robot, genellikle tarla sürme gibi, satır sonuna geldiğinde yön değiştirerek bir sonraki satıra geçer. Boustrophedon yöntemi, dönüş sayısını minimize ederek robotun enerji tüketimini azaltmayı ve görev süresini kısaltmayı hedefler. Boustrophedon hücresel ayrıştırmanın, özellikle geniş alanlarda ve dönüşlerin sınırlı olduğu durumlarda avantaj sağladığı belirtilmiştir (Choset & Pignon, 1998). Bu yöntemin en büyük avantajlarından biri, engelli alanlarda bile robotun yolunu kesintisiz bir şekilde sürdürebilmesidir. Bu durum, robotun alanı daha verimli bir şekilde kapsamasını ve daha az sayıda dönüş yaparak enerji tüketimini azaltmasını sağlar.

Özellikle, boustrophedon hücresel ayrıştırma adı verilen bir yöntem kullanılarak robotun hareket edebileceği alanlar hücrelere bölünür. Her bir hücre, robot tarafından ileri geri hareketlerle kaplanır ve tüm hücreler kaplandığında, robot alanın tamamını taramış olur. Robotun serbest hareket alanı, birbirine bitişik hücrelerden oluşur ve her hücre bir kere kapsandığında robot tüm serbest alanı kapsar. Şekil 2.12.'de sol kısımda algoritmanın ortamda nasıl gözüktüğü ve hücresel düzeyde nasıl gözüktüğü görselleştirilmiştir. Bu yöntem tamdır; yani, algoritma sınırlı bir süre içinde kapsama yolunu bulur ya da bulunamadığını belirler. Ayrıca bu yöntem, engellerin bulunduğu karmaşık ortamlar için de uygulanabilir. Bu yaklaşımın doğruluğu, belirli bir sürede tüm alanın kapsanacağı veya kapsanamayacağını belirlenmesiyle garanti altına alınmıştır (Choset & Pignon, 1998).

Çalışma alanı poligonlara ayrılarak her bir hücre için süpürme yolları oluşturulmakta ve en uygun süpürme yolu optimize edilmektedir. Böylece engellerin bulunduğu ortamlarda bile alanın verimli şekilde kapsanması sağlanır. Geleneksel kapsama yöntemlerine kıyasla %14 daha kısa yol üretme kapasitesine sahip olduğu belirtilmiştir (Bähmann ve diğ., 2019).

Özellikle otonom araçlar için yapılan çalışmalarda, Boustrophedon yönteminin avantajları arasında düşük enerji tüketimi ve hızlı kapsama süresi yer almaktadır. Yöntemin spiral tabanlı yaklaşımlar ile birleştirildiği ve özellikle dinamik engellerin

bulunduğu ortamlarda yüksek başarı oranı elde edildiği belirtilmiştir (Choi ve diğ., 2009). Bu çalışma, engellerle karşılaşıldığında robotun rotasını değiştirerek alanın tamamını en az sayıda dönüş ile kapsamasını sağlayan bir algoritma geliştirmiştir.



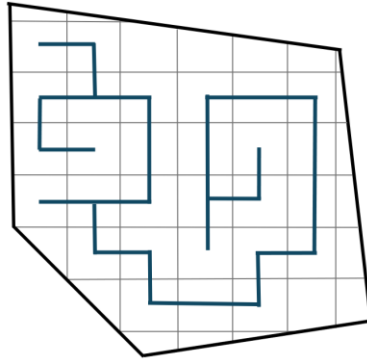
Şekil 2.12. Boustrophedon Algoritması Görünümü

Boustrophedon yöntemi, özellikle kapalı ve yapılandırılmış ortamlarda verimlilik sunarken, düzensiz ve dinamik ortamlarda bazı sınırlamalara sahiptir. Engellerin düzensiz olduğu ortamlarda robotun daha fazla dönüş yapması gerekebilir, bu da enerji ve zaman maliyetini artırabilir. Bu tür sınırlamaların üstesinden gelmek için yol yumuşatma algoritmalarının kullanıldığı ve böylece daha verimli bir kapsama sağlandığı belirtilmiştir (Şelek ve diğ., 2022).

Bu yöntemin avantajlarından biri de özellikle enerji tüketiminin optimize edilmesidir. Özellikle büyük alanlarda, Boustrophedon tarzı sistematik ileri geri hareket, robotun enerjisiyi daha verimli kullanmasını sağlar ve böylece daha az sürede daha fazla alan kapsanmış olur. Enerji verimliliği açısından Boustrophedon benzeri yöntemlerin, öğrenme tabanlı algoritmalarla birleştiğinde daha etkin bir hale geldiği ve geleneksel algoritmalarından daha az enerji tükettiği gösterilmiştir (Noh ve diğ., 2022). Özellikle dinamik ortamlarda, derin pekiştirmeli öğrenme algoritmaları ile güçlendirildiğinde bu yöntem, büyük bir etki yaratabilir.

2.9.2. Boustrophedon Yayılan Ağaç Kaplama (YAK) Yol Planlama

Boustrophedon YAK algoritması, otonom mobil robotların bir alanı eksiksiz ve etkin bir şekilde kapsamasını sağlamak amacıyla geliştirilmiş bir yol planlama yöntemidir ve klasik Boustrophedon yönteminin eksikliklerini gidermek üzere tasarlanmıştır. Algoritma, Şekil 2.13.'de gösterildiği gibi bir otonom mobil robotun belirli alanda bulunan her noktayı tam olarak bir kez kapsayacak şekilde hareket etmesini sağlamaktadır. Klasik Boustrophedon yöntemi, alanı hücresel ayrıştırma ile ayırık bölgelere ayırır ve robotun bu bölgeleri paralel hatlar boyunca ileri-geri hareket ederek kapsamasına dayanır. Ancak, klasik yöntem hücreler arası geçişleri optimize etmediğinden, özellikle engellerin bulunduğu karmaşık veya düzensiz şekilli alanlarda gereksiz dönüşler ve enerji tüketimi ortaya çıkarabilir.



Şekil 2.13. Boustrophedon YAK Algoritması Gösterimi

Boustrophedon YAK algoritması, bu sınırlamaları aşmak için geliştirilmiştir. Bu algoritma, hücresel ayrıştırma sonrası bir yayılan ağaç oluşturur ve robotun tüm alanı tek bir sürekli yol ile kapsamasını sağlar. Yayılan ağaç, hücreler arası geçişleri optimize ederek gereksiz hareketleri ve dönüşleri azaltır. Ayrıca, engellerin bulunduğu ortamlarda robotun hareketini dinamik olarak planlayarak kapsama oranını maksimize eder. Robot, her bir hücre içinde ileri-geri tarama yaparak alanın tamamını kapsar ve bu işlemi hücreler arasında en kısa yolu izleyerek sürdürür. Bu yaklaşım, enerji tüketimini ve çalışma süresini azaltırken kapsama etkinliğini artırır.

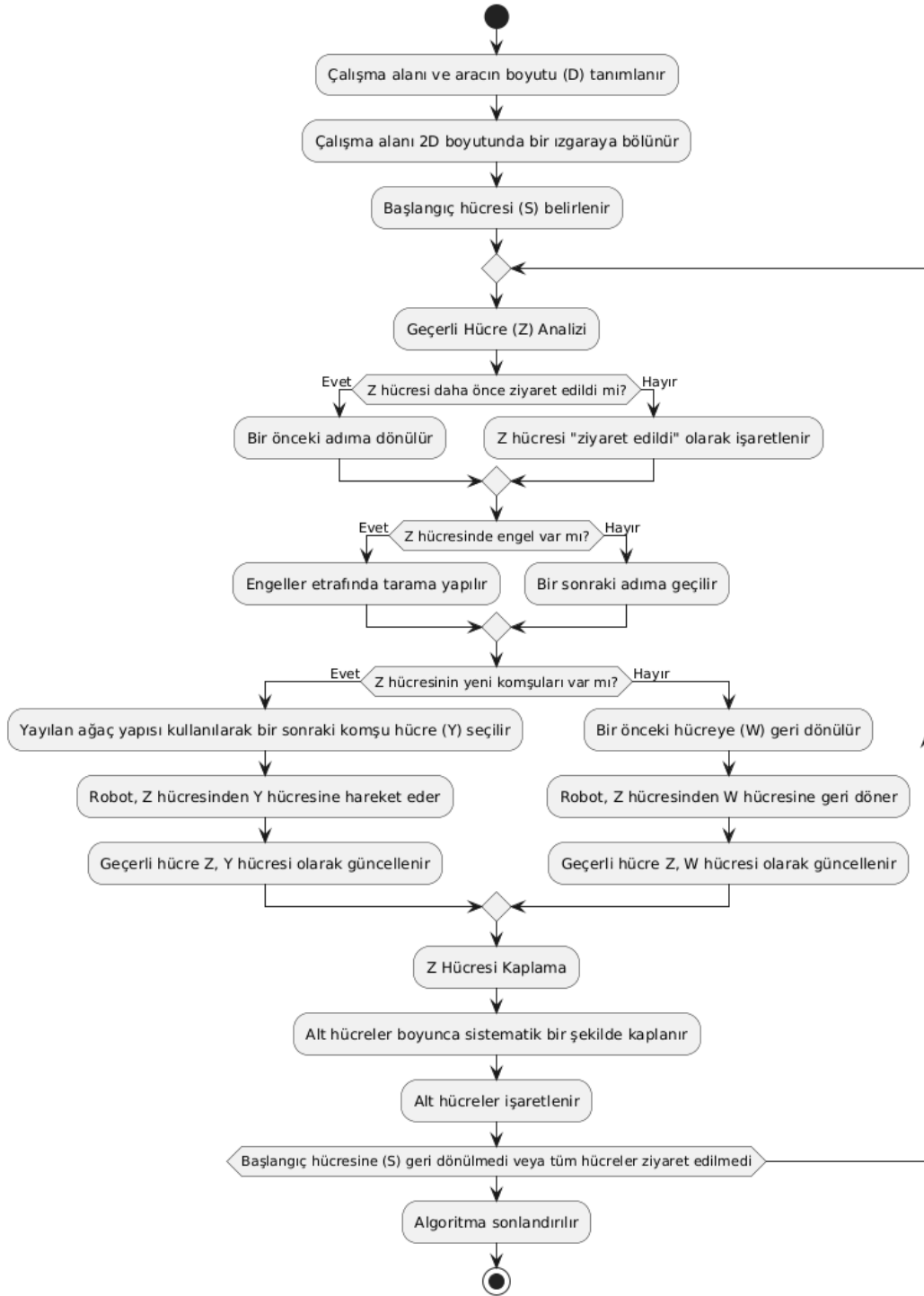
Boustrophedon YAK algoritmasının çevrimiçi versiyonuna yönelik oluşturulan akış diyagramı Şekil 2.14.'de gösterilmiştir. Algoritma, çalışma alanını verimli bir şekilde kapsamak için kullanılan sistematik bir süreci göstermektedir. Algoritma, öncelikle

çalışma alanının ve aracın boyutunun tanımlanmasıyla başlar ve alanı 2D bir ızgaraya böler. Başlangıç hücresi belirlendikten sonra, algoritma iteratif bir süreç içinde çalışır. Her iterasyonda, mevcut hücrenin analizi gerçekleştirilir. Bu süreçte, hücrenin daha önce ziyaret edilip edilmediği ve engel içerip içermediği kontrol edilir. Engel içermeyen ve daha önce ziyaret edilmemiş hücreler, sistematik bir şekilde alt hücreler boyunca kaplanır ve "ziyaret edildi" olarak işaretlenir. Eğer mevcut hücreye komşu yeni hücreler varsa, yayılan ağaç yapısı kullanılarak bir sonraki hücre seçilir ve robot bu hücreye hareket eder. Yeni komşu hücre bulunmadığında ise robot, yayılan ağaçta önceki hücreye geri döner. Bu döngü, başlangıç hücresine dönülüp tüm hücrelerin ziyaret edildiği doğrulanana kadar devam eder. Son olarak, algoritma çalışma alanını tamamen kapsayacak şekilde tamamlanır. Bu diyagram, algoritmanın her bir adımını mantıksal bir sıra içinde sunarak, çevrimiçi kapsama senaryolarında uygulama için net bir rehber sunmaktadır.

Klasik ve YAK yöntemleri arasındaki en belirgin farklardan biri enerji verimliliği ve kapsama yolunun optimizasyonunda ortaya çıkar. Klasik yöntem, her hücrenin bağımsız kapsanmasına odaklanırken, YAK algoritması alanın tamamını kapsayan bir plan sunarak robotun toplam yol uzunluğunu minimize eder. Bu özellik, YAK'ı endüstriyel temizlik robotları gibi enerji tasarrufu ve verimliliğin kritik olduğu uygulamalar için ideal bir yöntem haline getirir. Boustrophedon YAK algoritması, klasik yöntemle kıyasla daha gelişmiş bir yapı sunarak karmaşık ve dar alanlarda yüksek performans sergiler ve otonom mobil robotlar için daha etkin bir yol planlama çözümü sunar.

2.9.3. Izgara Tabanlı Gezgin Satıcı Problemi ile Kapsama Yol Planlama

Bormann ve arkadaşlarının geliştirdiği yöntem (Bormann ve diğ., 2018), H. Choset ve çalışma arkadaşları tarafından önerilen Morse Hücresel Ayrıştırma Yöntemini baz alan (Choset & Pignon, 1998), keşif alanını kritik noktalar kullanarak topolojik özelliklere dayalı hücresel bölümlere ayırır. Bu yöntemde, her hücre, robotun etkili bir hareket modeli izlemesi için optimize edilmiştir. Morse hücresel ayrıştırmanın klasik boustrophedon yöntemlerinden temel farkı, haritayı döndürerek ve kritik noktalardan yararlanarak hücre sınırlarını belirlemesidir. Böylece robotun engellerle dolu, karmaşık alanlarda bile sistematik bir yol izlemesi sağlanır.



Şekil 2.14. Boustrophedon YAK Algoritması Akış Şeması

Yöntemin ilk adımında, harita üzerindeki gradyan yönleri Sobel operatörü ile hesaplanır. Gradyan yönü kullanılarak harita döndürülür ve bu döndürme, hücrelerin en uygun ekseninde ayrılmasına olanak tanır. Daha sonra, Morse fonksiyonları kullanılarak süpürme işlemi gerçekleştirilir. Süpürme hattı boyunca segment bağlantı durumu kontrol edilir ve

bağlantılar arttığında hücreler oluşturulur, azaldığında ise mevcut hücreler genişletilir. Bu aşamalardan sonra her hücre, bir hareket şeması oluşturmak için konturları üzerinden yeniden tanımlanır.

Sonrasında, Morse hücresel ayrıştırma yöntemi ile elde edilen tüm hücreler üzerinden bir global yol planlanır. Bu aşamada gezgin satıcı problemi çözülerek, robotun hücreleri en verimli sırayla ziyaret etmesi sağlanır. Her hücrede, ileri-geri hareket eden bir boustrophedon yol planı uygulanır. Böylece robotun en uzun yollar boyunca hareket etmesi sağlanır ve gereksiz dönüşler minimize edilerek enerji tasarrufu yapılır. Hücrelerin döndürülerek en uzun kenarlarının yatay eksene paralel hale getirilmesi, bu düzenlemeye katkı sağlar. Her hücre içinde tanımlanan yol, işlem sonunda orijinal konumuna döndürülerek global yola entegre edilir.

2.9.4. Sinir Ağı Tabanlı Kapsama Yol Planlama

Sinir ağı tabanlı kapsama yol planlaması, mobil robotların belirli bir alanı kapsarken engellerden kaçınmasını ve en uygun yolu bulmasını sağlayan bir tekniktir. Bu yöntemde, grid tabanlı Gezgin Satıcı Problemi çözümleyicisinin küresel optimizasyon yapmasının aksine, sinir ağı tabanlı kapsama yol planlama, yarı-küresel bir yaklaşım benimseyerek, nöronların aktivasyon dinamiklerinden ilham alır. Bu yöntemde, her bir grid hücresi bir nöron olarak adlandırılır ve her nöron, kendisine komşu sekiz nöronla bağlantı kurar. Nöronların dinamikleri, henüz ziyaret edilmemiş nöronlardan gelen çekici terimleri, engellerden ve duvarlardan gelen itici terimleri ve ziyaret edilmiş hücreler için nötr terimleri içeren bir shunting denklemi ile modellenir (Yang & Luo, 2004). Simon X. Yang ve Chaomin Luo'nun çalışmasında, sinir ağı tabanlı bu modelin hesaplama açısından basit olduğu ve zamanla değişen çevrelerde robotların çarpışma olmadan kapsama yol planlaması yapabildiği belirtilmiştir (Yang & Luo, 2004b).

Sinir ağı tabanlı kapsama yol planlama, robotun çalışma alanını ızgara yapısında temsil eder ve her nöron, robotun hareket edeceği belirli bir konumu gösterir. Bu sistemde nöronların aktivite seviyeleri, robotun en uygun rotayı seçmesini sağlar. Nöronlar arasındaki bağlantılar ve aktivite düzeyleri, robotun engellerden kaçınarak görevini tamamlamasına yardımcı olur. Özellikle biyolojik sinir sistemlerinden ilham alan bu modeller, engellerin robotu yerel olarak itmesi ve kapsanması gereken alanların robotu

küresel olarak çekmesi prensibiyle çalışır. Bu sayede robot, engellerle karşılaştığında yeni bir yol planlamasına gerek duymadan anında karar alabilir ve yoluna devam edebilir. Sinir ağı tabanlı kapsama yol planlama algoritmasına Q-learning eklenerek algoritmanın engeller çevresindeki yol planlama stratejisinin optimize edildiği belirtilmiştir. Böylece, kapsama oranı %100'e ulaşırken, tekrar edilen yol oranı büyük ölçüde azaltılmıştır (Tan ve diğ., 2023).

Sinir ağı tabanlı kapsama yol planlayıcısı, keşif görevlerinde robotun, haritanın tamamını kapsayacak şekilde hareket etmesini sağlamak için kullanılan bir yöntemdir. Bu yöntem, nöron aktivasyon dinamiklerine dayanır ve robotun kapsama yolunu, her bir hücreyi sırayla ziyaret edecek şekilde optimize eder (Lu ve diğ., 2023). Yöntemde, harita üzerinde belirli bir adımla örneklenen her nokta bir nöron olarak adlandırılır ve bu nöronlar, birbirine komşu olan diğer nöronlarla bağlantı kurarak bir hücre ızgarası oluşturur.

Bu kapsama planlamasında, her bir nöronun dinamikleri belirli aktivasyon terimleriyle modellenir. Henüz ziyaret edilmemiş hücelere ait nöronlar çekici terimler olarak belirlenirken, engeller ve duvarlar için itici terimler kullanılır. Ziyaret edilen hücelere ise nötr olarak değerlendirilir. Bu model, robotun hareket sırasında engellerden kaçınarak boş alanları en kısa sürede kapsamasını sağlar. Robotun bulunduğu konumdan en yüksek aktivasyona sahip nöron her adımda seçilir ve robot bu nörona doğru hareket eder. Böylece, algoritma bir noktaya sıkışmışsa ve hareket etmesi gerekiyorsa, daha önce ziyaret edilmemiş hücelere yönelerek aktivasyonun daha yüksek olduğu bölgelere doğru devam eder. Her adımda nöronların aktivasyon durumları yeniden güncellenir, bu da algoritmanın yerel bilgiye göre anlık olarak karar verebilmesini sağlar.

Algoritmanın kapsama alanı sağlamak üzere tasarlanan ilk aşamasında, robotun belirli bir kapsama yarıçapı tanımlanır ve bu kapsama alanı içerisinde kalan tüm nöronların kapsandığı varsayılır. Daha sonra, belirlenen başlangıç konumundan hareket edilerek, belirli bir sabit katsayı kullanılarak robotun yönünü değiştirmesi gereken durumlar minimize edilir. Bu katsayı, robotun yön değiştirme gereksinimini azaltarak daha istikrarlı bir yol izlemesini sağlar ve böylece enerji verimliliği artırılır.

Algoritmanın sonunda, robotun ayak izi ya da görüş alanı için planlanmış bir kapsama yolu elde edilir. Robot ayak izi üzerinden yol planlaması yapılacaksa, bu yol doğrudan

kullanıma sunulur. Ancak, robotun görüş alanı (FOV) için planlama yapılmışsa, görüş alanının merkezi ile robotun merkez konumu arasındaki vektör dönüşümü uygulanarak bu yol robot ayak izine göre yeniden şekillendirilir. Eğer robotun hesaplanan pozisyonu serbest alanda değilse, görüş alanı merkezinin belirli bir yarıçapı etrafında daha erişilebilir bir nokta hesaplanır ve bu nokta, robotun son pozisyonuna en yakın mesafeyi sağlayacak şekilde seçilir.

Sinir ağı tabanlı kapsama yol planlaması yöntemi, özellikle dinamik olmayan ortamlarda, engellerin etrafında dolaşarak tüm alanı kapsayacak etkili bir yol sunar. Robotun dönüş hareketleri azaltılarak daha doğal bir hareket yolu oluşturulur ve böylece hem kapsamayı maksimuma çıkarır hem de enerji verimliliğini artırır.

2.9.5. Dışbükey Sensör Yerleştirme Kapsama Yolu Planlaması

Dışbükey Sensör Yerleştirme Kapsama Yolu Planlaması yöntemi, kapsama planlama problemlerine yönelik etkili bir yaklaşımdır ve gezgin satıcı problemi ile ilişkilidir. Bu problem, belirli bir alanın en az sayıda sensörle tamamen gözlemlenmesini amaçlar ve mobil robotlarla kapsama planlama problemine dönüştürülebilir. Bu yaklaşımda, robotun üzerine yerleştirilen sensörlerin en uygun şekilde yerleştirilmesi ve robotun bu alanı tamamen tarayarak kapsama sağlayabileceği en kısa yolun bulunması esas alınır (Bormann ve diğ., 2018).

İlk aşamada, kapsanacak alanda mümkün olan en az sayıda sensör pozisyonu seçilir. Seçilen bu konfigürasyonlar, haritadaki tüm hücrelerin gözlemlenebilmesini sağlayacak stratejik noktalardan oluşur. Algoritma, sensörlerin algılama kapasitesine göre haritanın her bir bölgesini analiz ederek tüm alanı kapsayacak en az sayıda sensör pozisyonunu belirler. Bu süreçte, gezgin satıcı problemi kullanılarak sensör pozisyonları arasında en kısa yol hesaplanır. Robotun bu pozisyonlar arasındaki en kısa yolu izlemesi, enerji verimliliği ve görev süresi açısından önemlidir.

Sensör yerleştirme problemini çözmek için tam sayılı doğrusal programlama (Integer Linear Programming, ILP) kullanılır. Bu model, tüm alanın gözlemlenmesini sağlayacak pozisyonları en düşük maliyetle bulmayı hedefler. Ancak, bu tür programlar genellikle

zaman alıcı olabilir. Bu nedenle, daha hızlı bir başlangıç çözümü elde etmek için iteratif yeniden ağırlıklandırılmış konveks gevşetme yöntemi uygulanır.

2.9.6. Izgara Tabanlı Yerel Enerji Minimizasyonu Kapsama Yol Planlama

Izgara Tabanlı Yerel Enerji Minimizasyonu Kapsama Yol Planlama algoritması, enerji verimliliğine dayalı kapsama planlama problemlerinde kullanılan etkili bir yöntemdir. Bu algoritmada, robotun hareket ettiği alan, belirli aralıklarla ızgara hücrelerine bölünür. Robot, başlangıç noktasından itibaren hareket ederken her ziyaret ettiği hücreyi işlenmiş hücreler listesine ekler ve bir sonraki adımını seçerken enerji tüketimini minimize eden bir strateji izler. Bu yaklaşım, temizlik robotları gibi enerji sınırlaması olan sistemler için büyük avantaj sağlar (Liu & Zhou, 2023b).

Algoritma, robotun bir sonraki hedef hücrelerini seçerken bir enerji fonksiyonunu minimize etmeyi hedefler. Bu enerji fonksiyonu öteleme mesafesi, dönel mesafe ve komşu hücrelerin durumu olmak üzere üç ana bileşene dayanır. Öteleme mesafe, robotun şu anki pozisyonu ile potansiyel bir sonraki konum arasındaki doğrusal uzaklıktır ve Denklem (2.1)'deki gibi hesaplanır.

$$d_t(p, n) = \sqrt{(p_x - n_x)^2 + (p_y - n_y)^2} / l \quad (2.1)$$

Burada l , grid hücrelerinin mesafesini belirtir ve p , robotun mevcut pozisyonunu, n ise potansiyel hedef hücreyi ifade eder. Rotasyonel mesafe ise robotun yönelimi ile bir sonraki konumun yönelimi arasındaki açısal farkı temsil eder ve Denklem (2.2)'de verildiği gibi hesaplanır.

$$d_r(p, n) = \frac{|p_\theta - n_\theta|_{angle}}{\pi/2} \quad (2.2)$$

Bu iki bileşen, robotun bir sonraki adımında hem doğrusal hem de açısal olarak en düşük enerji tüketimini sağlamayı amaçlar. Bunun yanı sıra, komşu hücrelerin durumu da göz önünde bulundurulur. Komşu hücreler, potansiyel hedef hücrenin 8 komşuluğu içinde yer alır ve daha önce ziyaret edilmemiş olan komşu hücreler robot için daha cazip hale getirilir. Bu durum, Denklem (2.3)'deki gibi hesaplanır.

$$N(n) = 4 - \frac{\sum_{k \in Nb8(n)} |k \cap L|}{2} \quad (2.3)$$

Bu formülde, $Nb8(n)$, hücrenin 8 komşusunu ifade ederken, L , işlenmiş hücrelerin listesini belirtir. Bu cazibe terimi, robotun halihazırda işlenmiş hücrelerin yanına yönelmesini ve komşu bölgeleri en verimli şekilde kapsamasını sağlar.

Algoritma, komşu hücrelerin tümü ziyaret edildikten sonra, işlenmemiş hücrelerden bir sonraki hedefi seçer ve bu süreçte robotun aynı hücreyi tekrar ziyaret etmemesi sağlanır. Algoritma, tüm hücreler ziyaret edildiğinde sona erer ve bu şekilde enerji verimliliği sağlanarak tam kapsama gerçekleştirilir.

2.9.7. Kontur Hattı Tabanlı Kapsama Yol Planlama

Kontur Hattı Tabanlı Kapsama Yol Planlama, kapsama planlama problemlerinde kullanılan ve hücresel ızgara veya harita yönelimi normalizasyonu gerektirmeyen bir yaklaşımdır. Bu yöntem, robotun çalışma alanındaki engellere olan mesafeleri temel olarak kapsama yolları oluşturur ve özellikle geniş, açık alanlar ile dar geçitler arasında geçiş yapmayı gerektiren durumlarda oldukça etkili bir çözüm sunar. Yöntemin ilk adımı, haritadaki engellere en uzak noktaların bulunmasıdır. Bu noktalar, Voronoi grafinin iskeleti kullanılarak hesaplanır ve merkez noktalar olarak adlandırılır. Voronoi grafinin kritik noktaları, iki geniş alan arasında kalan dar geçitlerde ortaya çıkar. Bu kritik noktalar kullanılarak, alan, en yakın engel noktalarına doğrudan bağlantılar oluşturularak bölümlere ayrılır. Haritanın bu şekilde bölünmesi, Voronoi hücre ayrıştırmasına benzer bir süreçtir. Her bir Voronoi hücresi, robotun çalışma alanındaki bağımsız bölgeleri temsil eder. Voronoi hücreleri oluşturulduktan sonra, her hücre içinde bir mesafe dönüşümü hesaplanır. Bu dönüşüm, haritanın her erişilebilir noktasının en yakın engele veya hücre sınırına olan uzaklığını temsil eder (Bormann ve diğ., 2018). Bu hesaplamaların sonucunda, hücre içindeki engellere eşit mesafeli kontur hatları oluşturulur. Robotun kapsama genişliği $wc = 2 \cdot rc$ olarak tanımlandıktan sonra, kontur seviyeleri Denklem (2.4)'e göre belirlenir.

$$ci = rc + i \cdot wc \quad (2.4)$$

Her bir kontur seviyesindeki noktalar, mesafe, yön deęişikliği ve eęer noktalar daha önce temizlendiyse bu durumu dikkate alarak maliyetleri minimize edecek şekilde birbirine bağlanır.

Son olarak, kontur seviyesindeki yollar birbirine bağlanarak, tüm alanın kapsanması sağlanır. Bu yöntem, robotun alanı enerji açısından verimli bir şekilde kapsamasına olanak tanır ve özellikle geniş, açık alanlardan dar geçitlere geçerken robotun yol planlamasını optimize eder. Algoritmanın doğrudan harita üzerinde çalışabilmesi ve hücresel grid yapısı gerektirmemesi, robotik uygulamalar için pratik bir çözüm sunar.

2.9.8. Enerji Fonksiyonel Optimizasyonu Yöntemi ile Kapsam Yol Planlama

Enerji Fonksiyonel Optimizasyonu yöntemi, robotun belirli bir alanı en kısa sürede ve en düşük enerji tüketimi ile kapsamasını sağlamak amacıyla geliştirilmiş bir yol planlama tekniğidir. Bu yöntemde, robotun serbest alan olarak tanımlanan bölgeleri kapsayabilmesi için harita, her biri kapsamı gereken küçük düğümlere bölünür. Her düğüm bir hücre büyüklüğündedir ve bu düğümlerin tamamı kapsanarak alanın tümü taranır. Robotun başlangıç pozisyonuna en yakın düğümden başlayan yol, robotun her adımda bir sonraki düğüme geçmesiyle genişletilir ve belirli bir enerji fonksiyonunu minimize ederek kapsama sağlanır.

Bu yöntemde ilk adım olarak, robotun hareket edebileceği tüm serbest alanlar düğümlere ayrılarak tanımlanır. Bu düğümler, hücre boyutuna göre serbest alanın sol üst köşesinden başlanarak belirlenir. Serbest alanın her beyaz pikseli, bir düğüm olarak işaretlenir ve her düğüm kendi çevresindeki sekiz komşu düğümlerle ilişkilendirilir. Bu komşular, daha sonra enerji fonksiyonunun hesaplanması aşamasında yol optimizasyonu için kullanılacaktır (Bormann ve dię., 2015).

Yol planlama sürecinde, başlangıç pozisyonuna en yakın olan ve haritanın kenarında bulunan bir düğüm başlangıç noktası olarak seçilir. Kenar düğümü, dört komşusundan daha az sayıda komşuya sahip olan düğümdür ve başlangıç olarak seçilmesi, robotun kenar boyunca başlayarak ilerlemesini sağlar. Bir sonraki düğüm ise enerji fonksiyonunu minimize eden komşu düğüm olarak belirlenir. Her bir adımda daha önce ziyaret edilmemiş komşular arasında en düşük enerji maliyetine sahip olan düğüm seçilerek robot

bu düğüme yönlendirilir. Eğer komşular arasında erişilebilir bir düğüm bulunamazsa, algoritma tüm grid üzerinde sonraki düğümü arar ve kapsama yoluna devam eder. Bu işlem, tüm düğümler kapsanarak robotun tüm serbest alanı dolaşmasına kadar tekrarlanır.

Elde edilen kapsama yolunun robotun ayak izine veya görüş alanına uyarlanması istenirse, robotun görüş alanı merkez noktası ile robotun merkez noktası arasındaki bir vektör dönüşümü uygulanır. Bu dönüşüm, robotun ayak izi veya görüş alanı bazında yol planlamasını mümkün kılar. Eğer hesaplanan robot pozisyonu serbest alan dışındaysa, görüş alanı merkezinde belirli bir yarıçap üzerinde en yakın erişilebilir noktaya bir dönüşüm yapılır.

Enerji Fonksiyonel Optimizasyonu yönteminin avantajları arasında, robotun hareket süresini ve enerji tüketimini azaltmaya yönelik optimize edilmiş bir kapsama yolu sunması yer alır. Yöntem, her bir düğüm için minimum enerji maliyetini dikkate aldığı için robotun dönüş hareketlerini azaltır ve daha kısa, verimli bir kapsama yolu oluşturur. Ayrıca, komşu düğümlerin öncelikli olarak değerlendirilmesi sayesinde robot, yakın mesafeli hareketlerle daha hızlı kapsama sağlar. Yöntemin diğer bir avantajı, başlangıç pozisyonuna yakın kenar düğümünü seçerek kenarlardan başlayan bir yol çizmesi, bu sayede robotun sınırlara yakın alanları ilk olarak kapsayabilmesi ve iç alanlara daha düzenli bir şekilde ilerleyebilmesidir.

Ancak, yöntemin bazı dezavantajları da vardır. Enerji fonksiyonunu minimize etmek için geliştirilen algoritma, özellikle karmaşık ve çok sayıda engel içeren haritalarda hesaplama yükü yaratabilir. Bu durumda, algoritmanın her adımda komşu düğümler arasında enerji maliyetini kıyaslaması, işlem süresini artırabilir ve gerçek zamanlı uygulamalarda performans sorunlarına neden olabilir. Ayrıca, düğüm ağının sadece yakın komşular üzerinden planlanması nedeniyle, bazı durumlarda robotun döngüsel hareketler yaparak aynı bölgeyi tekrar tekrar ziyaret etme riski vardır. Bu durum özellikle düğümlerin dar koridorlarda veya sıkışık bölgelerde yer aldığı haritalarda ortaya çıkabilir. Algoritma her seferinde en düşük enerji maliyetine sahip düğümü seçtiği için bazen küresel bir yol planlamasından saparak kısa mesafede döngüler yaratabilir.

Enerji Fonksiyonel Optimizasyonu yöntemi, belirli bir enerji verimliliği gerektiren ve serbest alanların tamamının kapsanmasını hedefleyen yol planlamalarında oldukça etkili

bir stratejidir. Yöntem, özellikle açık alanlarda verimli bir kapsama sağlarken, karmaşık ve engellerle dolu ortamlarda işlem süresini artırarak bazı sınırlamalar sergileyebilir.

2.10. Oda Segmentasyonu

Oda segmentasyonu, iç mekan haritalarını ayrıştırarak odalara bölme sürecini ifade eden, robotik, yapay zeka ve bina yönetimi gibi alanlarda önemli bir yere sahip olan bir yöntemdir. Bu süreç, genellikle 2 boyutlu haritalardan veya 3 boyutlu nokta bulutu verilerinden yapılır ve hem robotların hem de insanların iç mekanları daha verimli bir şekilde anlamalarını ve kullanmalarını sağlar. Oda segmentasyonu, bir ortamın semantik olarak analiz edilmesi, navigasyon rotalarının optimize edilmesi ve bina yönetiminin daha etkin hale getirilmesi gibi birçok kritik uygulama alanına sahiptir. Bir temizlik robotu odaları bağımsız birimler olarak tanıyarak her bir odanın temizliğini optimize edebilir. Benzer şekilde, güvenlik sistemleri belirli alanları önceliklendirebilir ve oda bazlı izleme yapabilir. İnsan-robot etkileşimi bağlamında ise, segmentasyon, robotların doğal dilde verilen komutları anlamalarını ve uygulamalarını mümkün kılar. Örneğin, bir robota "mutfığa git" komutunun verilebilmesi için, robotun odaları tanımlayabilmesi gereklidir. Bu hem robotların işlevselliğini hem de insanlarla olan etkileşimlerini güçlendiren bir unsurdur. Oda segmentasyonu aynı zamanda sanal gerçeklik uygulamalarında ve mimari tasarımlarda, bina modellerinin daha etkili bir şekilde oluşturulmasına olanak tanır (Bormann ve diğ., 2016).

Oda segmentasyonu, çeşitli yöntemlerle gerçekleştirilebilir ve bu yöntemler farklı veri türleri ve uygulamalar için uyarlanabilir. Voronoi grafiği tabanlı segmentasyon, engeller arasındaki boşlukları analiz ederek odaları belirlemek için kullanılır. Bu yöntem, daha düzenli ve simetrik mekanlarda etkili bir çözüm sunar. Morfolojik segmentasyon ise görüntü işleme tekniklerinden türetilmiş olup, morfolojik operatörler kullanarak odaları ayırır. Bu yöntem, düzensiz şekilli veya karmaşık yapılara sahip mekanlarda avantaj sağlar. Uzaklık dönüşümü tabanlı segmentasyon, her noktanın en yakın engele olan mesafesini hesaplayarak odaları tanımlar ve büyük ölçekli iç mekanlarda uygulanabilirliği ile öne çıkar. Özellik tabanlı segmentasyon ise lazer tarayıcı veya diğer sensörlerden elde edilen verilerden faydalanarak odaları sınıflandırır. Bu yöntem, odaların geometrik özelliklerinin yanı sıra semantik bilgilerin de kullanılmasını mümkün kılar.

Her bir segmentasyon yöntemi, belirli bir uygulama alanında daha avantajlı olabilir. Örneğin, Voronoi tabanlı yöntemler hızlı ve verimli çözümler sunarken, özellik tabanlı yaklaşımlar daha karmaşık verilerin analizi için idealdir. Ancak, tüm yöntemlerdeki temel amaç, odaların doğru bir şekilde ayrıştırılması ve bu ayrıştırmanın belirli uygulamalara göre optimize edilmesidir.

Endüstriyel alanlarda ise oda segmentasyonunun uygulanabilirliği sınırlı olabilir. Bu tür ortamlarda genellikle geniş, açık alanlar bulunur ve engeller belirli bölgelerde yoğunlaşır. Bu nedenle, oda segmentasyonu yerine alanların belirlenmesi veya bölümlenmesi kullanıcı tercihlerine bağlı olarak gerçekleştirilir. Örneğin, bir fabrikanın üretim sahasında, segmentasyon yapılmaksızın tüm alan bir bütün olarak ele alınabilir. Bununla birlikte, belirli görevler için segmentasyon kullanılabilir. Temizlik robotları için yoğun engellerin bulunduğu bölgeler ayrı birer bölüm olarak tanımlanabilirken, geniş açık alanlar bir bütün halinde bırakılabilir. Bu esnek yaklaşım, endüstriyel ortamların dinamik yapısına uygun bir planlama yapılmasını sağlar. Segmentasyonun seçime bağlı olması, robotların enerji verimliliği, görev planlaması ve çalışma alanlarının doğru yönetimi açısından büyük bir avantaj sunar.

2.11. Önerilen Yöntem

Önerilen yöntem, kapsama yol planlama sürecinin sistematik bir şekilde yürütülmesini sağlamak amacıyla belirli aşamalardan oluşmaktadır. Bu süreç, harita verisinin işlenmesinden başlayarak alanın hücresele bölümlenmesine, hücre içi hareket planlamasına, hücreler arası geçiş stratejilerine ve nihai kapsama yol planlamasına kadar çeşitli adımları içermektedir. Bu sürece ilişkin şema Şekil 2.15.'te gösterilmiştir.



Şekil 2.15. Önerilen Yöntem Aşamaları

Robotun faaliyet göstereceği ortamın dijital temsili, ızgara doluluk (occupancy grid) formatında işlenmiş bir harita dosyası olarak sisteme sunulmaktadır. ROS platformunda standart olarak kullanılan bu format, her bir grid hücresinin doluluk durumunu belirten

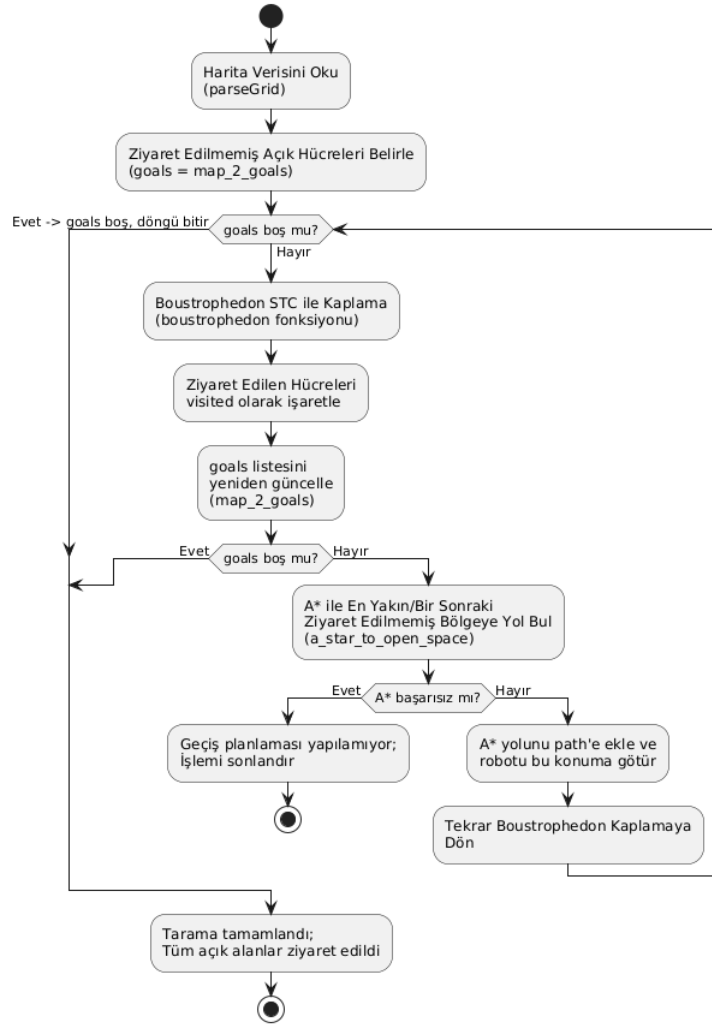
bir deęer (örneğin 0–100 arası) ile ifade edilmektedir. Bu haritada belirli bir eşik deęerin (örneğin >50 veya >65) üzerinde olan hücreler engel olarak kabul edilmekte, geri kalan alanlar ise gezilebilir bölgeler olarak işaretlenmektedir. Böylece, robotun hareket planlaması için sadeleştirilmiş ve ölçeklenmiş bir "harita matrisi" oluşturulmaktadır.

Haritanın kapsama yol planlaması açısından yönetilebilir bir formata dönüştürülebilmesi için grid tabanlı hücresel dekompozisyon yöntemi uygulanmaktadır. Bu süreçte, harita belirli bir çözünürlüğe baęlı olarak eşit boyutlu hücrelere ayrılmakta ve her hücre, gezilebilir veya engel olarak kategorize edilmektedir. Bu yöntem, literatürde Grid-Based Approximate Cellular Decomposition olarak adlandırılmaktadır. Hücresel dekompozisyonun ardından, her bir hücre içinde Boustrophedon Spanning Tree Coverage (BSTC) yöntemi uygulanarak kapsama yol planlaması gerçekleştirilmektedir. Hücre içi hareket planlaması, robotun her bir hücre içerisindeki mikro hareketlerini ve hücre sınırlarına olan geçiş noktalarını belirlemektedir. Önerilen yöntemde, robotun bir hücreyi merkez noktasından geçerek ziyaret ettiği kabul edilmekte ve hücre içi hareket tek bir adım olarak modellenmektedir. Robotun belirli bir yönlendirme ile hareket edebilmesi için hücreler içerisindeki hareket yönlerini (örneğin doęu, batı, kuzey, güney) belirlenerek rota oluşturulmaktadır. Hücre içi gezinti gereklilięi durumunda, ek yörünge planlaması ihtiyacı doęacaktır. Ancak, önerilen yöntemde her hücre tekil bir konum olarak deęerlendirildięinden hesaplama yükü azaltılmaktadır.

Hücreler arası geçiş aşaması, robotun kapsama sürecinde bir hücreden dięerine nasıl geçeceęini belirlemektedir. Bu süreçte Boustrophedon hareket stratejisi uygulanarak, robot bir yönde (örneğin doęu yönünde) ilerlemekte, engel veya harita sınırına ulařınca yön deęiřtirerek kapsama devam etmektedir. Robotun hareketini yöneten boustrophedon algoritmasına göre, komřu hücrelere geçiři kontrol etmektedir. Eęer hedef hücreye doęrudan erişim mümkün deęilse veya engeller nedeniyle geçiş saęlanamıyorsa, A* arama algoritması devreye girmektedir. Harita üzerinde en kısa ve uygun geçiş yolunu belirleyerek robotun engelleri aşmasını saęlamaktadır.

Son aşama olan kapsama yol planlaması, robotun sistematik bir şekilde tüm alanı ziyaret etmesini saęlamaktadır. Robot başlangıç konumundan itibaren belirlenen yönde zigzag hareket ile kapsama yapmaktadır. Kapsanmamış hücreler kaldıęında, A* arama algoritması kullanılarak en yakın ziyaret edilmemiş hücreye ulařılmakta ve kapsama

devam edilmektedir. Kapsama tamamlandığında, robotun izlediği yol dizisi gerçek dünya koordinatlarına dönüştürülerek zaman damgalı poz verisi formatında bir yol planı oluşturulmaktadır. Bu yol planı, ROS navigasyon sistemleri ile uyumlu olup, robotun belirlenen yolu takip etmesini sağlamaktadır.



Şekil 2.16. Önerilen Algoritma Akış Şeması

Önerilen yönteme ait algoritma akış şeması Şekil 2.16.'da sunulmuştur. Kapsama sürecinin tamamlanmasını sağlamak üzere, sistematik bir hücrel geçiş mekanizması oluşturulmuş ve eksik kapsama alanlarını minimize etmek için A* arama tabanlı bir yeniden hedefleme mekanizması entegre edilmiştir.

Önerilen yöntem, ilk olarak çevrenin harita verisini işler ve robotun operasyon alanını belirler. Daha sonra, kapsama planlamasının başlangıç noktası olarak, ziyaret edilmemiş açık hücrelerin belirlenmesi sağlanır ve gezilebilir ancak henüz ziyaret edilmemiş hücreler bir hedefler belirlenir. Kapsama süreci, hedef listesinin boş olup olmadığının kontrol edilmesiyle yönetilir; eğer tüm alan kapsanmışsa algoritma sonlandırılır.

Boustrophedon STC yöntemi, sistematik kapsama sağlayarak belirlenen hücreleri gezmekte ve bu süreci boustrophedon fonksiyonu ile yürütmektedir. Robotun ilerleme stratejisi, belirli bir yönde doğrusal hareketle başlayıp, engel veya harita sınırına ulaştığında yön değiştirerek devam eden bir mekanizmaya dayanmaktadır. Kapsanan hücreler ziyaret edilmiş olarak işaretlenerek algoritmanın ilerleyişi takip edilmektedir. Kapsama devam ederken, yeni ziyaret edilen hücreler ile güncellenerek hedef listesi yeniden oluşturulmaktadır. Eğer bu aşamada hedef listesi boşalır, kapsama süreci tamamlanmış sayılır.

Özellikle endüstriyel fabrika ortamlarında, geniş açık alanlar ve uzun koridorlar kadar, dar geçitler ve engellerle ayrılmış alanlar da bulunmaktadır. Klasik Boustrophedon yöntemi, geniş açık alanlarda etkin bir kapsama sağlarken, dar veya bölünmüş alanlarda boşluklar bırakabilmektedir. Bu sorunu gidermek amacıyla Yayılan Ağaç Kapsama (Spanning Tree Coverage) yaklaşımı uygulanmakta ve eksik kapsama bölgelerine ulaşım için A* arama algoritması devreye sokulmaktadır. Robotun mevcut konumundan en yakın ziyaret edilmemiş hücreye en uygun rotayı oluşturmasını sağlamaktadır. Eğer A* arama algoritması başarılı bir şekilde bir yol oluşturursa, bu rota takip edilerek robot ilgili konuma yönlendirilmekte ve kapsama süreci kaldığı yerden devam etmektedir. Ancak, A* arama algoritması başarısız olur ve hedef hücreye ulaşamazsa, geçiş planlaması yapılamadığı için kapsama süreci sonlandırılmaktadır.

Bu metodoloji sayesinde, Boustrophedon STC yöntemi yan geçiş stratejileri ve hedef bulma mekanizmaları ile entegre edilerek, dar koridorlar, engel içeren alanlar ve bölünmüş odalar arasında kapsamanın sürekliliğini sağlamaktadır. Böylece, robot tüm kapsama alanlarını verimli bir şekilde ziyaret ederken, eksik kapsama bölgeleri en aza indirilmekte ve sistemin endüstriyel ortamlara uygunluğu artırılmaktadır.

3. DENEYSEL BULGULAR VE TARTIŞMA

Bu tez çalışmasının temel amacı, endüstriyel otonom mobil robot süpürge için kapsam yol planlama algoritmalarını enerji verimliliği odaklı optimize etmektir. Günümüzde kapsama algoritmaları çeşitli alanlarda etkinlik sağlamak üzere geliştirilmiş olsa da endüstriyel temizlik robotlarının geniş fabrika alanlarında enerji verimliliğini önceliklendiren bir çözüme ihtiyaç duyulmaktadır. Bu robotlar, geniş ve açık alanların yanı sıra dar veya karmaşık bölgelere de erişim sağlamalıdır; bu da hem etkin navigasyonu hem de enerji tüketimini minimize eden kapsama algoritmalarını gerektirir. Bu kapsamda çalışmada, endüstriyel temizlik robotlarının karşılaştığı zorluklara uygun bir çözüm sunmak amacıyla, Boustrophedon hücresel ayrıştırma yöntemine dayanan ve Robot İşletim Sistemi (ROS) platformu üzerinde çalışacak şekilde özelleştirilmiş bir algoritma geliştirilmiştir. Bu algoritma, her bir hücreyi yalnızca bir kez ve minimum dönüş sayısı ile ziyaret ederek robotun sürekli ve kesintisiz bir kapsama sağlamasını amaçlamaktadır.

Geliştirilen algoritma, enerji verimliliğini artırmanın yanı sıra endüstriyel ortamlarda geniş alanların verimli bir şekilde temizlenmesini sağlamak için kapsam yüzdesi, dönüş sayısı, seyahat mesafesi ve seyahat süresi gibi çeşitli performans metrikleri ile değerlendirilmiştir. Bu metrikler doğrultusunda yapılan analizler, algoritmanın sanal simülasyon, test ortamı ve fabrika koşullarında uygulanarak incelenmesini içermektedir.

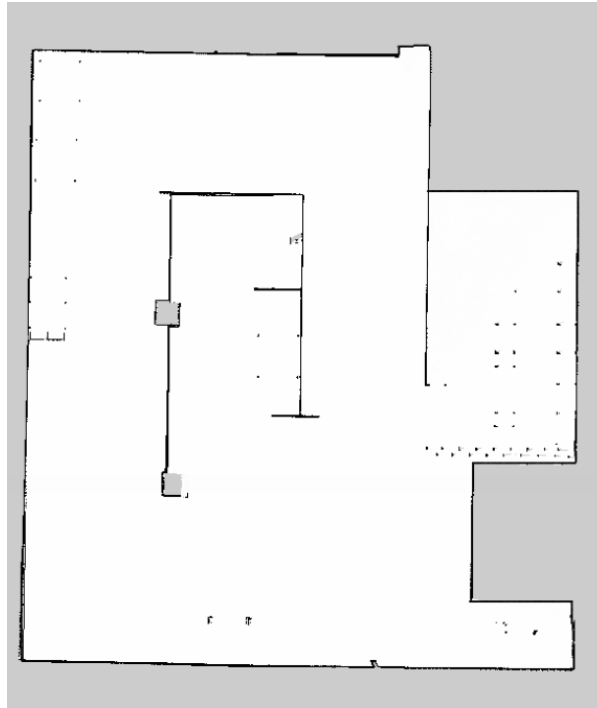
Bu çalışmanın başarılı bir şekilde yürütülebilmesi için, önerilen algoritma ve benzer çalışmalar farklı ortamlarda test edilmiştir. Bu ortamlar sanal ortam, atölye ortamı ve fabrika ortamıdır. Ortaya çıkan sonuçlar performans metriklerine göre değerlendirilmiştir. Bölüm 3.1'de kullanılan test ortamlarının oluşturulması ve haritaların hazırlanması süreci açıklanmıştır. Benzer algoritmaların belirtilen ortamlar ve haritalar üzerinde çalıştırılması ise Bölüm 3.2.'de detaylandırılmıştır. Ayrıca, önerilen yönteme dair algoritma bilgileri, test sonuçlarına dair bilgiler Bölüm 3.3.'te sunulmuştur. Tam kapsam yol planlama algoritmasının takip edilmesi Bölüm 3.4.'te, deneysel çalışmaların ve önerilen algoritmanın performans metriklerine göre sonuçları Bölüm 3.5.'te incelenmiştir.

3.1. Haritalama

3.1.1. Sanal Ortam Haritalama

Bu bölümde, sanal ortamda harita oluşturma sürecinde kullanılan yöntemler ve adımlar açıklanmaktadır. Tam kapsama yol planlama algoritmalarının, gerçek ortamın fiziksel koşullarından bağımsız olarak test edilebilmesi amacıyla, deneyler Gazebo simülasyon ortamında gerçekleştirilmiştir. Bu kapsamda, çalışma alanının 3B modelleri oluşturulmuştur. Daha sonra Gazebo ortamı için model dosyaları oluşturularak 3B modeller ROS'a entegre edilmiştir.

Haritalama sürecinde cartographer metodu kullanılarak Gazebo ortamında harita oluşturulmuştur. Cartographer, robotun hareketi sırasında elde edilen sensör verilerini kullanarak ortamın ayrıntılı bir haritasını çıkaran bir SLAM (Simultaneous Localization and Mapping) yöntemidir. Bu yöntem, özellikle dinamik ve karmaşık ortamlarda yüksek doğrulukta haritalama yapması ile öne çıkmaktadır. Gazebo ortamında, Cartographer'ın LiDAR verilerini kullanarak harita oluşturması sağlanmıştır. Elde edilen harita Şekil 3.1.'de gösterilmektedir.

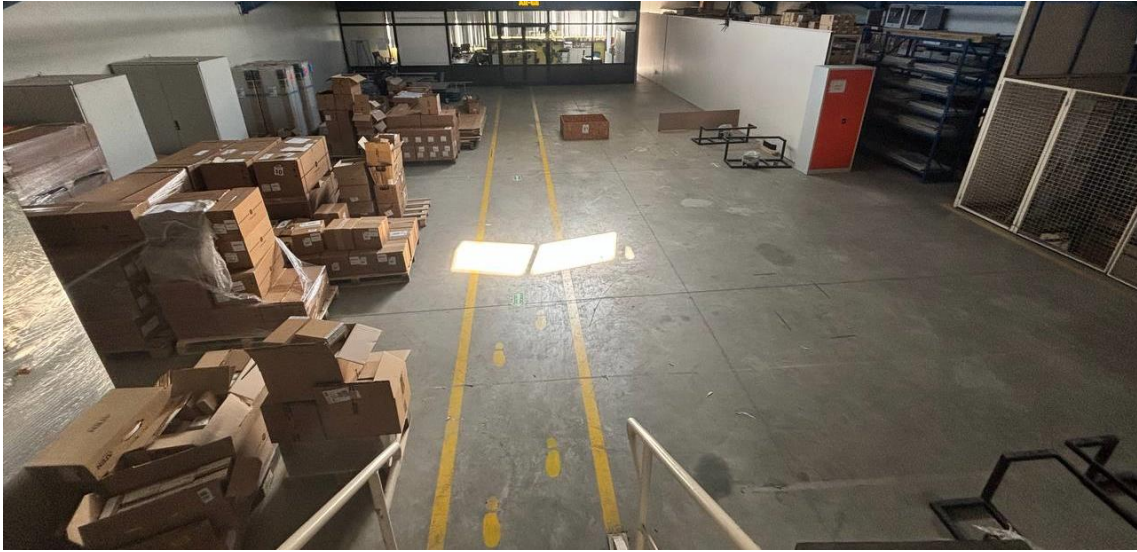


Şekil 3.1. Sanal Ortam Haritası

3.1.2. Atölye Test Ortamı Haritalama

Bu bölümde, atölye test ortamında gerçekleştirilen haritalama süreci ele alınmaktadır. Gerçek ortam koşullarını yansıtacak şekilde hazırlanan bu test ortamında, robotun kapsama yol planlama algoritmasının gerçek dünyadaki performansı değerlendirilmiştir. Gazebo simülasyon ortamının aksine, atölye test ortamı, robotun fiziksel hareketlerini, algılayıcı hassasiyetini ve çevresel değişkenleri göz önünde bulundurarak daha kapsamlı bir değerlendirme imkânı sunmuştur.

Atölye test alanı Şekil 3.2.'de gösterilmiştir. Alan, 14 x 12 metre ölçülerinde olup, robotun geniş alanlarda ve karmaşık yerleşim düzeni bulunmaktadır. Bu kapsama sağlama yeteneğini değerlendirmek üzere düzenlenmiştir. Bu fiziksel ortamda gerçekleştirilen testler, simülasyon ortamında yaşanabilecek kısıtlamaları aşarak robotun gerçek engellerle etkileşimini ve enerji tüketimini daha doğru bir şekilde ölçmeyi sağlamaktadır.



Şekil 3.2. Atölye Test Ortamı

Atölye ortamında gerçek robot ile Cartographer metodu kullanılarak çıkarılan harita, Şekil 3.3.'te detaylı olarak sunulmaktadır. Bu harita, robotun atölye alanında gerçekleştirdiği kapsama yol planlaması sırasında elde ettiği çevresel verileri ve fiziksel engelleri kapsamlı bir şekilde göstermektedir. Haritada, robotun etrafındaki nesnelere, dar

alanlar ve karmaşık bölgeler dahil olmak üzere, test alanının tüm detayları doğru ve hassas bir şekilde yansıtılmıştır.



Şekil 3.3. Atölye Ortamı Haritası

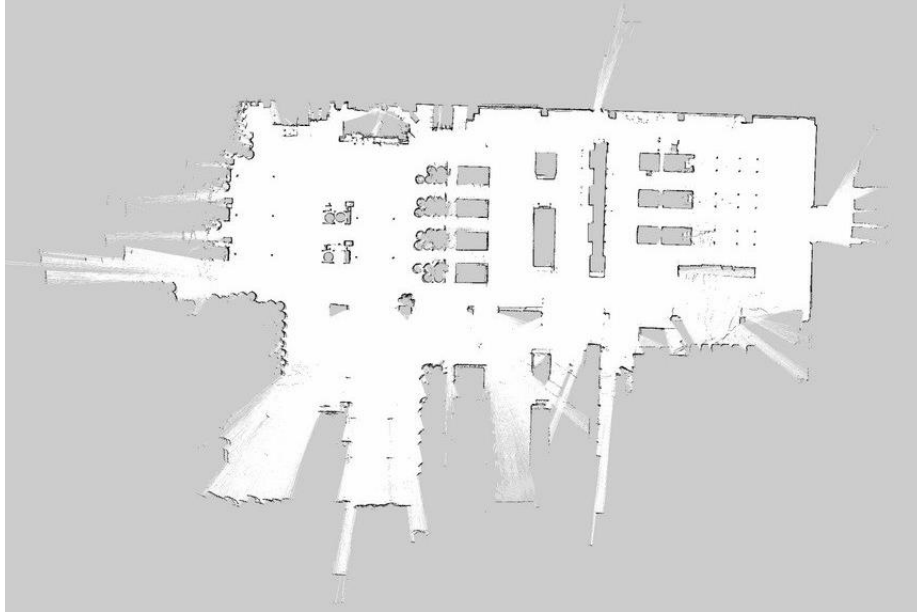
Haritalama işlemi sırasında robotun konumlandırma ve çevre algılama yetenekleri sayesinde elde edilen bu veri, algoritmanın gerçek dünya koşullarında performansını değerlendirmek amacıyla önemli bir temel sunmaktadır. Şekil 3.3.'te gösterilen harita, robotun geçiş yollarını planlaması ve engelleri güvenli bir şekilde yönetmesi için ihtiyaç duyduğu bilgileri içermektedir. Ayrıca, haritanın doğruluğu, robotun enerji verimli bir kapsama sağlama hedefini destekleyecek şekilde, çevresel engelleri ve alan sınırlarını başarılı bir şekilde tanımlamaktadır. Bu bağlamda, Şekil 3.3.'teki harita, algoritmanın atölye ortamındaki performansını analiz etmek için temel bir referans niteliğindedir.

3.1.3. Fabrika Ortamı Haritalama

Bu bölümde, 45 x 22,5 metre (1.012,5 metrekare) ölçülerindeki fabrika ortamında gerçekleştirilen haritalama süreci ele alınmaktadır. Bu ortam, robotun kapsama yol planlama algoritmasının endüstriyel bir alanda sergilediği performansı değerlendirmek amacıyla kullanılmıştır. Atölye test ortamından farklı olarak, fabrika ortamı, robotun daha büyük ve karmaşık bir alan içinde navigasyon yapmasını, farklı engellerle etkileşimini ve enerji verimliliğini test etme fırsatı sunmuştur.

Fabrika test alanı, robotun çeşitli görev bölgeleri arasında hareket ederken karşılaşılabileceği tipik fabrika engellerini (makineler, depolama alanları, geçitler) içerecek şekilde düzenlenmiştir. Bu düzenleme, robotun geniş alanları kapsama yeteneğini ve yol planlama algoritmasının karmaşık endüstriyel bir ortamda enerji verimliliğini gözlemlemek amacıyla yapılmıştır.

Gerçek robot ile Cartographer metodu kullanılarak çıkarılan fabrika ortam haritası, Şekil 3.4.'te detaylı olarak sunulmaktadır. Bu harita, robotun fabrika alanında gerçekleştirdiği kapsama yol planlaması sürecinde elde ettiği çevresel verileri ve fiziksel engelleri detaylı bir biçimde yansıtmaktadır. Haritada, robotun fabrika içindeki nesnelere, geniş geçiş alanları ve dar geçitler dahil olmak üzere, test alanının tüm detayları yüksek doğrulukla sunulmuştur.



Şekil 3.4. Fabrika Ortamı Haritası

3.2. Performans Metrikleri

Temizlik robotları yazılımında performans metriklerinin hesaplanması, robotun etkinliği ve verimliliğini değerlendirmek amacıyla yapılır. Bu metriklerin hesaplanmasında kullanılan yöntemler ve hesaplama süreçleri bu bölümde açıklanmıştır.

Kaplama yüzdesi, bir harita üzerindeki kapsanabilir alanın ne kadarının kaplandığını belirleyen önemli bir performans metriğidir. Bu oran, robotun temizlik görevini ne derece

etkin bir şekilde gerçekleştirdiğini ölçer. Hesaplama sürecinde öncelikle harita, grid tabanlı bir modellemeye dönüştürülür ve her bir grid belirli bir alanı temsil eder. Izgara doluluk formatında temsil edilen bu harita üzerinde robotun ziyaret ettiği hücreler işaretlenir ve bu işaretleme genellikle binary bir matrisle yapılır. Haritalar, her hücreyi belirli bir durumla ilişkilendirir ve bu durumlar sayısal değerlerle ifade edilir. Örneğin, -1 değeri genellikle bilinmeyen veya geçersiz alanları, 0 değeri serbest veya henüz kaplanmamış alanları, 100 değeri engelleri temsil eder. Kaplama yüzdesinin hesaplanması, harita üzerindeki kaplanmış hücre sayısının, kapsanabilir toplam hücre sayısına oranlanmasıyla gerçekleştirilir. Kapsanabilir toplam hücreler, haritada -1 dışındaki hücrelerden oluşur ve bu hücreler, serbest ve kaplanmış hücrelerin toplamıdır. Denklem 3.1'de, kaplama yüzdesi matematiksel formülle ifade edilmiştir (Bormann ve diğ., 2018).

Kaplama Yüzdesi(%)

$$= 100x \frac{\text{Kapanmış Hücre Sayısı}}{\text{Kapanmış Hücre Sayısı} + \text{Kapanmamış Hücre Sayısı}} \quad (3.1)$$

Dönüş sayısı, bir robotun hareket rotası boyunca yön değiştirdiği noktaların toplam sayısını ifade eder ve genellikle rotanın optimizasyon seviyesini değerlendirmek için önemli bir metrik olarak kullanılır. Bu metriğin hesaplanmasında robotun izlediği yol, x ve y koordinatlarıyla kaydedilir. Ardışık hareketler arasındaki açı farkı radyan cinsinden hesaplanır; bu fark belirli bir eşik değeri aştığında, ilgili hareket bir dönüş olarak kabul edilir. Bu yaklaşım, robotun tüm yön değiştirme noktalarının sayılarak toplam dönüş sayısının elde edilmesini sağlar. Robotun tüm hareket rotası boyunca bu dönüşler sayılarak toplam dönüş sayısı elde edilir. Dönüş sayısı Denklem (3.2)'de ki gibi hesaplanır.

$$\text{Dönüş Sayısı}(T) = \sum_{i=1}^{n-1} \delta(\theta_i, \theta_{i+1}) \quad (3.2)$$

Denklem 3.2'de, n rotadaki toplam adım sayısını, θ_i robotun i . adımda izlediği yön açısını, $\delta(\theta_i, \theta_{i+1})$ ise iki ardışık yön açısı arasındaki farkı temsil etmektedir. Belirlenen

eşik değeri ile karşılaştırılarak dönüş olup olmadığını kontrol eder. Eğer, $|\delta(\theta_i, \theta_{i+1})| > \epsilon_1$ ise dönüş kabul edilir. Burada ϵ_1 eşik değerini temsil eder.

Dönüş sayısının fazla olması, enerji tüketimini artırabilir ve robotun görev süresini kısaltabilir. Bu bağlamda, dönüşlerin azaltılmasının enerji tüketimi üzerindeki etkisi, özellikle enerji verimliliği odaklı yol planlama çalışmalarında ele alınmıştır. Zhang ve arkadaşlarının (2014) geliştirdiği enerji tasarruflu yol planlama algoritması, dönüş sayısını azaltarak robotların enerji tüketimini %28,24'e varan oranlarda düşürebilmiştir. Bu yöntem, geleneksel A* arama algoritmasına göre daha düşük enerji tüketimi sağlayan küresel olarak optimize edilmiş yollar oluşturmuştur. Dolayısıyla, dönüş sayısını azaltmaya yönelik stratejiler, enerji verimliliğini artırmada ve görev süresini uzatmada önemli bir rol oynamaktadır (Cai ve diğ., 2014).

Seyahat mesafesi, robotun başlangıç noktasından itibaren kat ettiği toplam mesafeyi ifade eder. Bu hesaplama için robotun izlediği yol üzerindeki x ve y koordinatları dikkate alınır. Ardışık iki nokta arasındaki mesafe Öklid mesafesi formülü ile hesaplanır. Bu mesafelerin toplamı, robotun toplam seyahat mesafesini verir. Seyahat mesafesi, robotun rotasının uzunluğunu ve enerji tüketimini değerlendirmek için önemli bir göstergedir. Seyahat mesafesi Denklem (3.3)'deki gibi hesaplanır.

$$D = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad (3.3)$$

D toplam seyahat mesafesi olmak üzere, x_i ve y_i i. adımda robotun olduğu koordinatlardır.

Seyahat süresi, robotun başlangıç noktasından görevin bitimine kadar geçen toplam süredir. Bu metrik, robotun hız ve mesafe verilerine dayanarak hesaplanır. Her bir segment için mesafenin hıza bölünmesiyle segment bazında süre hesaplanır. Segment sürelerinin toplanmasıyla toplam seyahat süresi elde edilir. Seyahat süresi, görev tamamlanma süresinin optimizasyonunda önemli bir parametredir ve zaman verimliliğini değerlendirir.

3.3. Yöntemlerin Deneysel İncelemesi ve Performans Karşılaştırmaları

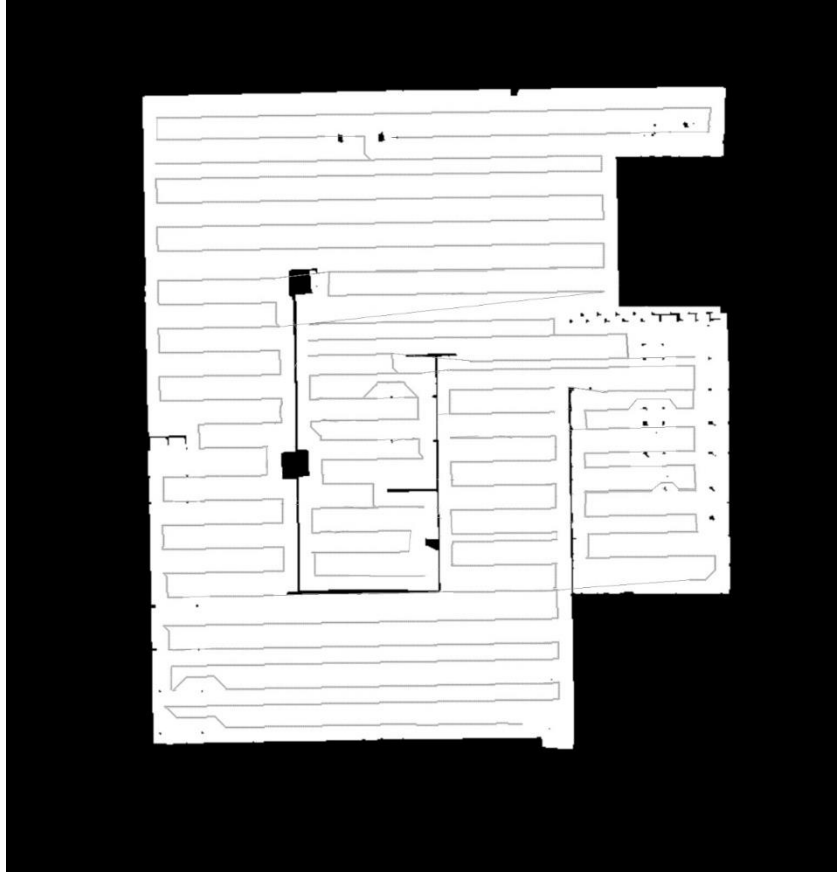
Bu çalışmada, kapalı alan kapsama yol planlama yöntemlerinin performanslarını değerlendirmek amacıyla, çeşitli algoritmalar sanal ortam, atölye ortamı ve fabrika ortamında test edilmiştir. Test edilen yöntemler arasında; Izgara Tabanlı Gezgin Satıcı Problemi Yaklaşımı, Sinir Ağı Tabanlı Yöntemi, Ortalama Kayma Kümeleme Yöntemi, Enerji Fonksiyonel Optimizasyonu Yöntemi ve Ortalama Kayma Kümeleme Yöntemi bulunmaktadır. Bu yöntemler, Bormann ve diğ. (2016) tarafından yapılan çalışmada, kapsama algoritmaları çerçevesinde ele alınmıştır .

Her bir yöntem, test ortamlarına göre farklı parametreler ile yapılandırılarak uygulanmış ve kapsama yüzdesi, enerji verimliliği, yol uzunluğu, dönüş sayısı ve hesaplama süresi gibi kriterler açısından performansları kıyaslanmıştır. Elde edilen sonuçlar, kapalı alanlardaki yol planlama verimliliğini optimize etmek için kullanılan yöntemlerin uygulama ortamına göre nasıl değişkenlik gösterdiğini ortaya koymakta ve bu sayede belirli uygulamalar için en uygun yöntemin seçilmesine olanak sağlamaktadır.

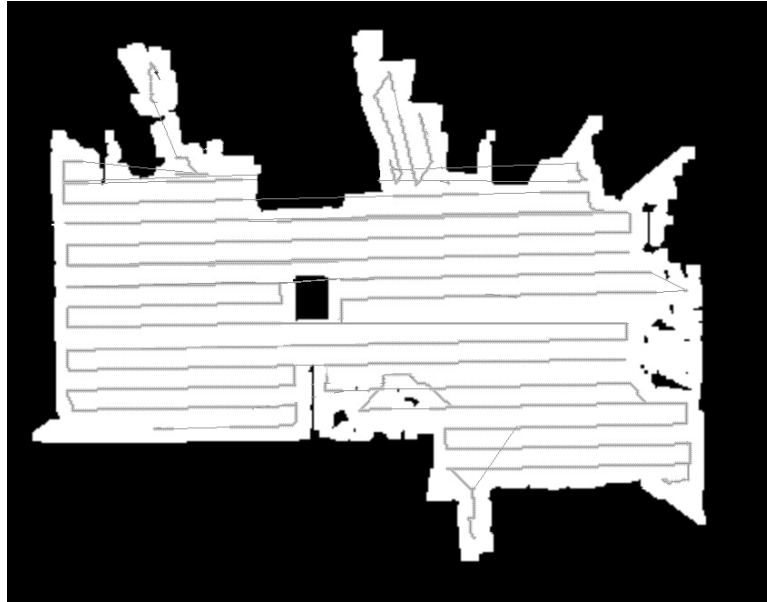
Deneysel simülasyon, atölye ve fabrika ortamları haritaları üzerinde test edilmiştir. Daha sonra simülasyon üzerinde ve atölye ortamında sırasıyla sanal ve gerçek ortamda robotun kaplama yapacak şekilde gezdirilmesi sağlanmıştır. Deneysel sonuçları, her yöntemin spesifik ortamlarda gösterdiği avantajları ve dezavantajları vurgulamakta olup, kapsamlı bir analiz sunmaktadır.

3.3.1. Boustrophedon Hüresel Ayırıştırma Yöntemi ile Deneysel Sonuçları

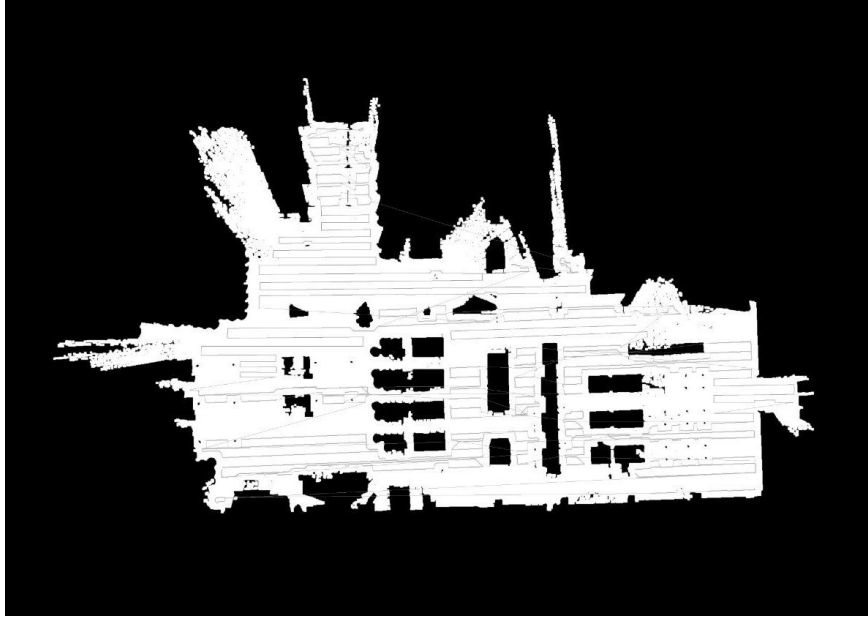
Bir haritayı hüresel parçalara ayırarak her hücrede kapsama yolu planlayan ve bu süreçte Morse yönteminden faydalanılmıştır; simülasyon, atölye ve fabrika ortamlarına ait haritalar üzerinde test edilerek farklı ortamlardaki performansları değerlendirilmiştir. Simülasyon ortamında elde edilen kapsama yolu Şekil 3.5.'te, atölye ortamında elde edilen kapsama yolu Şekil 3.6.'da, fabrika ortamında elde edilen kapsama yolu Şekil 3.7.'de sunulmuştur.



Şekil 3.5. Boustrophedon Algoritması Simülasyon Ortamı Yol Planı



Şekil 3.6. Boustrophedon Algoritması Atölye Ortamı Yol Planı



Şekil 3.7. Boustrophedon Algoritması Fabrika Ortamı Yol Planı

Şekil 3.5.'te simülasyon ortamında ideal sonuçlar veren Boustrophedon hücresel ayrıştırma yöntemi, engellerin daha fazla yer aldığı fabrika ortamında aşırı ayrıştırma sorunu yaşamıştır. Şekil 3.6.'da atölye ortamında ise Boustrophedon yöntemi, engelin olduğu geniş alanlarda oldukça etkili sonuçlar sunarken dar alanlarda aynı performansı göstermemiştir. Bu sonuçlar, yöntemin avantajlarını ve sınırlarını açıkça göstermektedir. Örneğin, daha basit veya nispeten statik ortamlarda Boustrophedon yöntemi yüksek verim sağlarken, karmaşık ve dinamik ortamlarda hesaplama yükü ve ayrıştırma gereksinimleri artmıştır. Yöntemin performans metriklerine göre sonuçları Tablo 3.1.'de gösterilmiştir.

Tablo 3.1. Boustrophedon algoritması ile elde edilen performans metrikleri

Ortam Haritaları	Kapsama Yüzdesi [%]	Seyahat Süresi [ss:dd:sn]	Seyahat Mesafesi [m]	Dönüş Sayısı [rad]	Hesaplama Süresi [ss:dd:sn]
Simülasyon	97,5	00:02:50	341,5	188,4	00:00:04
Atölye	93,2	00:01:33	187,2	81,64	00:00:04
Fabrika	76,8	00:06:18	578,7	317,14	00:00:10

Tablo 3.1.'deki veriler teknik açıdan değerlendirildiğinde, yöntemin güçlü yönleri kadar sınırlamaları da belirgin hale gelmektedir. Algoritmanın kapsama yüzdesi, seyahat süresi, seyahat mesafesi, dönüş sayısı ve hesaplama süresi gibi metriklerdeki performansı, uygulama senaryolarına göre farklılık göstermektedir.

Algoritmanın en dikkat çekici avantajlarından biri, hedef alanı kapsama başarısıdır. Hücrelerin belirlenmesi için sobel operatörüyle gradyan hesaplaması, kritik noktaların bulunması ve hücre sınırlarının çıkarılması gibi adımlar kapsama yüzdeliğini artırmıştır. Ancak bu durum fazladan zaman gerektireceğinden, gerçek zamanlı uygulamalarda sorun yaratabilir. Simülasyon ortamında %98,1, atölye ortamında ise %97,5'lik kapsama yüzdesi elde edilmiştir. Bu, yöntemin alanın neredeyse tamamını tarama kapasitesine sahip olduğunu ve özellikle eksiksiz kapsama gerektiren temizlik veya tarama uygulamalarında etkili bir çözüm sunduğunu göstermektedir. Fabrika ortamında kapsama yüzdesinin %76,8'e düşmüştür. Bunun sebebi, algoritma haritayı kritik noktalar ve gradyan yönüne göre hücrelere ayırmasıdır. Bu süreç genellikle küçük veya orta büyüklükteki alanlarda verimli bir şekilde çalışır. Ancak büyük alanlarda yapılan bu ayırıştırma, hücreler arasında dar veya düzensiz şekilli bölgelerin oluşmasına neden olur. Bu bölgeler algoritmada erişilemez olarak işaretlenebilir ve kapsamada zorluk yaratır. Bu sınırlama, kapsama yüzdesinin düşmesine yol açabilir.

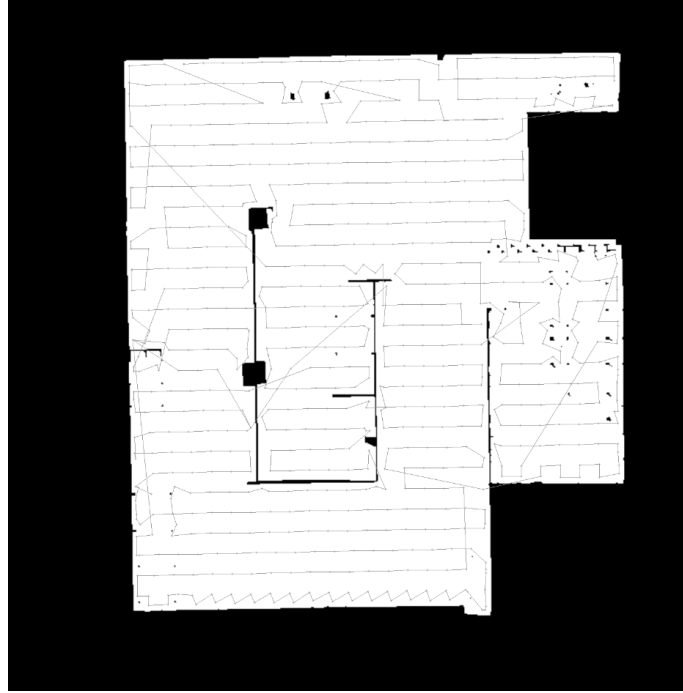
Seyahat süresi ve mesafesi açısından algoritma, düzenli ve kompakt yapılar için oldukça verimli sonuçlar sunmaktadır. Şekil 3.6.'da atölye ortamında seyahat süresi 93,6 saniye ve seyahat mesafesi 187,2 metre olarak ölçülmüştür. Bu, algoritmanın engelleri düzenli olarak böldüğü ve hücresel kapsama yönteminin kompakt ortamlarda enerji ve zaman açısından verimli bir şekilde çalıştığını göstermektedir. Ayrıca, atölye ortamında dönüş sayısının 81,64 gibi düşük bir değerde kalması, boustrophedon yol planlamasının bu tür ortamlarda optimizasyon sağlayabildiğine işaret etmektedir. Ancak Şekil 3.7.'de fabrika ortamında seyahat süresi 655,2 saniye ve seyahat mesafesi 1122,4 metre gibi yüksek değerlere ulaşmıştır. Bu durum, daha geniş ve düzensiz ortamlarda hücre bazlı ayırıştırmanın yol planlamada verimlilik açısından sınırlı kalabileceğini göstermektedir. Dönüş sayısının 317,14'e yükselmesi, yöntemin dönüş optimizasyonunda karmaşık yapılar için ek geliştirmeler gerektirdiğini ifade etmektedir.

Hesaplama süresi açısından bakıldığında, simülasyon ortamında yalnızca 4 saniye süren işlem, algoritmanın bu tür kontrollü ve düzenli alanlarda gerçek zamanlı uygulamalara uygun olduğunu göstermektedir. Ancak, atölye ortamında bu süre 8 saniyeye çıkarken, fabrika ortamında 10 dakikaya ulaşmıştır. Bu artış, geniş ve karmaşık alanlarda algoritmanın gradyan hesaplama ve kritik noktaların tespit edilmesi gibi aşamalarda

yüksek işlem maliyeti gerektirdiğini ortaya koymaktadır. Özellikle fabrika gibi düzensiz yapılarda, hücre sınırlarının çıkarılması ve kapsama yollarının belirlenmesi yoğun bir hesaplama yükü yaratmaktadır.

3.3.2. Izgara Tabanlı Gezgin Satıcı Problemi Yaklaşımı ile Deney Sonuçları

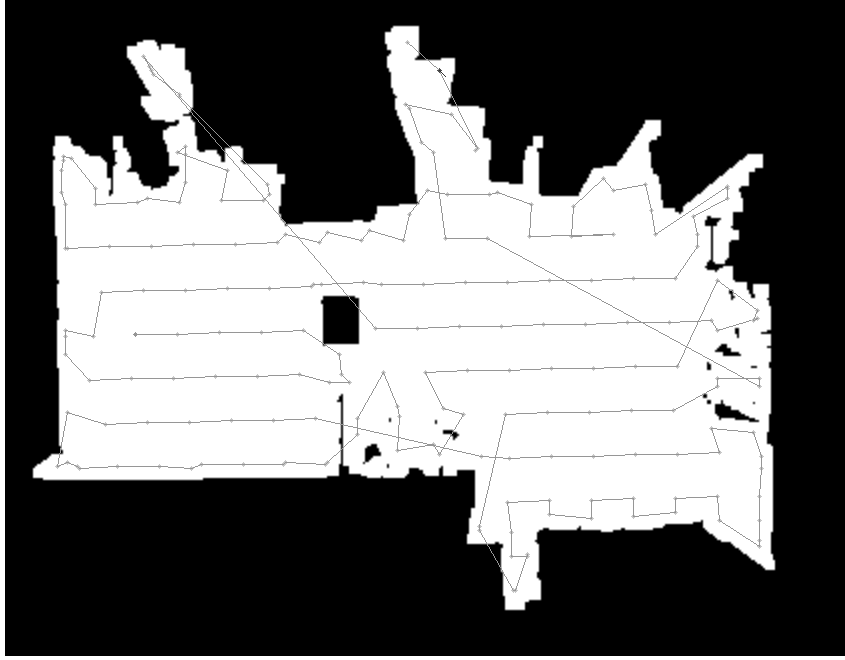
Bir haritayı grid tabanlı bir yapıya ayırarak her düğümde kapsama yolu planlayan ve bu süreçte Gezgin Satıcı Probleminden (TSP) faydalanan bu yöntem, simülasyon, atölye ve fabrika ortamlarına ait haritalar üzerinde test edilerek farklı ortamlardaki performansı değerlendirilmiştir. Simülasyon ortamında elde edilen kapsama yolu Şekil 3.8.'de, atölye ortamında elde edilen kapsama yolu Şekil 3.9'da, fabrika ortamında elde edilen kapsama yolu ise Şekil 3.10.'da sunulmuştur.



Şekil 3.8. Izgara Tabanlı Gezgin Satıcı Problemi Algoritması Simülasyon Ortamı Yol Planı

Hesaplanan kapsama rotası, robotun hareketini kontrol etmek için kullanılmakta ve belirlenen düğüm noktalarının sırasıyla ziyaret edilmesiyle haritanın tamamının kapsanması sağlanmaktadır. Algoritmanın kapsama oranı, düzenli yapıya sahip haritalarda oldukça yüksek bir başarı sergiler. Grid tabanlı yapının sağladığı düzenlilik, eksiksiz kapsama sağlaması açısından avantajlıdır. TSP algoritmasının optimal çözüm kabiliyeti, düğüm noktaları arasında en kısa yolun belirlenmesine olanak tanıyarak

robotun enerji tüketimini ve hareket süresini minimize eder. Ayrıca, farklı grid boyutları kullanılarak farklı boyut ve karmaşıklıkta haritalar için uyarlanabilir bir yapı sunar.



Şekil 3.9. Izgara Tabanlı Gezgin Satıcı Problemi Algoritması Atölye Ortamı Yol Planı



Şekil 3.10. Izgara Tabanlı Gezgin Satıcı Problemi Algoritması Fabrika Ortamı Yol Planı

Bununla birlikte, algoritmanın bazı sınırlamaları mevcuttur. Özellikle geniş ve karmaşık haritalarda TSP çözümünün yüksek işlem gücü gerektirmesi, hesaplama maliyetini artırmıştır. Bu durum, fabrika gibi büyük ve düzensiz ortamlarda algoritmanın

performansını düşüren önemli bir etkidir. Ayrıca, algoritma statik haritalar için tasarlandığından, dinamik ortamlarda meydana gelen değişiklikler rotanın yeniden hesaplanmasını gerektirir. Bu durum, gerçek zamanlı uygulamalar için ek bir zorluk oluşturabilir. Ek olarak, düğüm noktalarının engellere çok yakın olması, kapsama alanının bazı kısımlarının eksik kalmasına yol açmıştır. Yöntemin performans metriklerine göre sonuçları Tablo 3.2.'de gösterilmiştir.

Tablo 3.2. Izgara tabanlı gezgin satıcı algoritması ile elde edilen performans metrikleri

Ortam Haritaları	Kapsama Yüzdesi [%]	Seyahat Süresi [ss:dd:sn]	Seyahat Mesafesi [m]	Dönüş Sayısı [rad]	Hesaplama Süresi [ss:dd:sn]
Simülasyon	99,8	00:02:47	335,2	271,61	00:00:12
Atölye	98,7	00:00:51	103,9	102,05	00:00:18
Fabrika	94,8	00:06:03	748,3	794,42	00:00:30

Izgara tabanlı Gezgin Satıcı Problemi (TSP) algoritması, farklı ortam haritalarında elde edilen performans sonuçlarıyla hem güçlü yönlerini hem de sınırlamalarını teknik açıdan net bir şekilde ortaya koymaktadır. Simülasyon ortamında %99,8'lik kapsama oranı, 167,6 saniyelik seyahat süresi ve 335,2 birimlik mesafe ile düzenli ve küçük ölçekli alanlarda algoritmanın etkili bir çözüm sunduğu görülmektedir. Dönüş sayısının 271,61 gibi nispeten yüksek bir değerde olması dönüş optimizasyonunda sınırlı bir performansı gösterse de 12 saniyelik hesaplama süresi bu tür ortamlar için düşük işlem maliyetini ifade etmektedir.

Şekil 3.9.'de atölye ortamında algoritma %98,7 kapsama oranı sağlamış, seyahat süresi 51,95 saniye ve seyahat mesafesi 103,9 metre olarak ölçülmüştür. Bu sonuçlar, algoritmanın kompakt alanlarda daha verimli bir yol planlama sunduğunu ortaya koymaktadır. Dönüş sayısının 102,05'e düşmesi, düzenli ortamlarda dönüşlerin başarılı bir şekilde optimize edildiğini göstermektedir. Ancak, 18 saniyelik hesaplama süresi simülasyon ortamına kıyasla işlem maliyetinin arttığını işaret etmektedir. Bu artış, engellerin grid yapısına etkisi ve düğüm noktalarının belirlenmesindeki ek hesaplama yükünden kaynaklanabilir.

Şekil 3.10.'da fabrika ortamında %94,8'lik kapsama oranı ile geniş ve karmaşık haritalarda da alanı kapsama başarısını koruduğu görülmektedir. Ancak, seyahat süresi 363,6 saniye, seyahat mesafesi 748,3 birim ve dönüş sayısı 794,42 gibi yüksek değerlere

ulaşmıştır. Bu durum, algoritmanın geniş ve düzensiz alanlarda artan dönüş sayıları ve mesafelerle enerji ve zaman maliyetlerinin ciddi şekilde yükseldiğini göstermektedir. 30 saniye hesaplama süresi ise büyük ölçekli haritalarda yüksek işlem gücü gereksinimini açıkça ortaya koymaktadır.

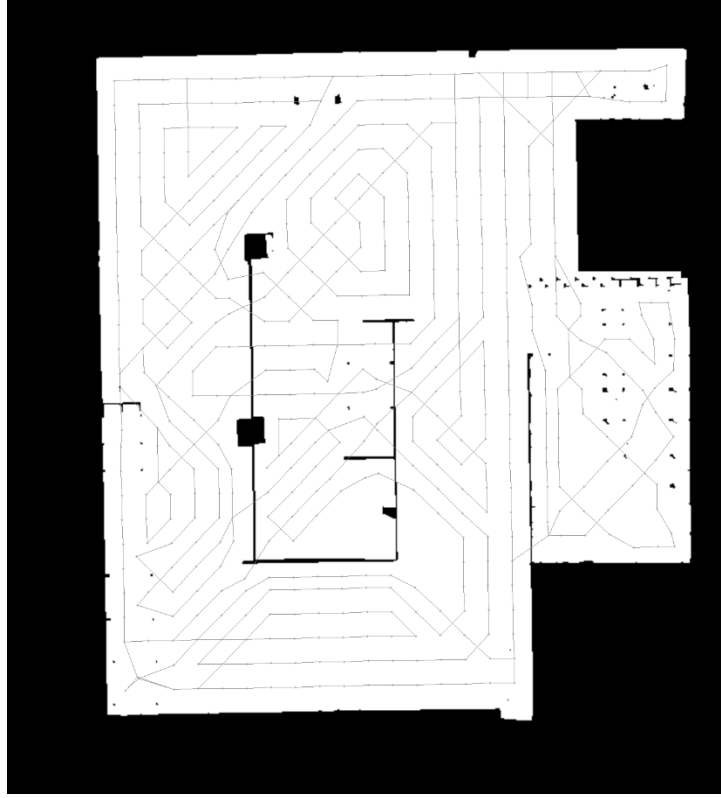
Genel olarak, algoritma düzenli ve orta büyüklükteki alanlarda kapsama başarısı ve verimlilik açısından etkili sonuçlar sunarken, dönüş sayısı ve hesaplama süresi gibi metriklerde geniş ve karmaşık alanlarda sınırlamalar göstermektedir. Bu bulgular, algoritmanın farklı ortam türlerindeki güçlü ve zayıf yönlerini net bir şekilde ortaya koymaktadır.

3.3.3. Sinir Ağı Tabanlı Kapsama Yol Planlayıcısı Yöntemi ile Deney Sonuçları

Sinir tabanlı kapsama yol planlayıcısı yöntemi, bir haritayı grid tabanlı bir yapıya ayırarak yapay sinir ağı modeli kullanmakta ve kapsama yolunu bu ağ üzerinden planlamaktadır. Haritanın her noktasını kapsamak amacıyla, sinir ağı düğümleri robotun hareketini optimize edecek şekilde hesaplanmıştır. Farklı ortamlardaki performansı değerlendirmek için yöntem, simülasyon, atölye ve fabrika ortamlarına ait haritalar üzerinde test edilmiştir. Simülasyon ortamında elde edilen kapsama yolu Şekil 3.11.'de, atölye ortamında elde edilen kapsama yolu Şekil 3.12.'da ve fabrika ortamında elde edilen kapsama yolu ise Şekil 3.13.'te sunulmuştur.

Sinir ağı tabanlı yol planlama algoritmalarının geniş, orta ve dar alanlardaki performansı üzerine yapılan analizler, algoritmanın farklı çevre koşullarında etkin bir şekilde çalışabildiğini göstermektedir. Fabrika ortamı gibi geniş ve karmaşık alanlarda, algoritma, engellerin çevresinden başarılı bir şekilde yönlendirme yaparak beyaz alanın tamamını kapsamış ve etkili bir kapsama oranı sunmuştur. Bununla birlikte, yüksek sayıda dönüş içeren karmaşık yol yapısı enerji tüketiminin artmasına neden olabileceği için bu durum önemli bir dezavantaj olarak öne çıkmaktadır. Orta karmaşıklığa sahip atölye ortamında ise grid tabanlı yapı üzerinden oluşturulan yol planlamasının, engelleri dikkate alarak geniş alan kullanımına imkân tanıdığı gözlemlenmiştir. Ancak, düğümler arası bağlantıların uzunluğu enerji verimliliğini olumsuz etkileyebilecek bir unsurdur ve yolun daha düzenli hale getirilmesi gerektiği düşünülmektedir. Küçük alanlar üzerindeki simülasyon sonuçları, algoritmanın yüksek hassasiyetle tüm alanı kapsayabildiğini ortaya

koymuştur. Ancak, sık dönüşlerin enerji verimliliğini düşürdüğü ve robotun hareket akıcılığını kısıtladığı dikkat çekmektedir. Yöntemin performans metriklerine göre sonuçları Tablo 3.3.'te gösterilmiştir.



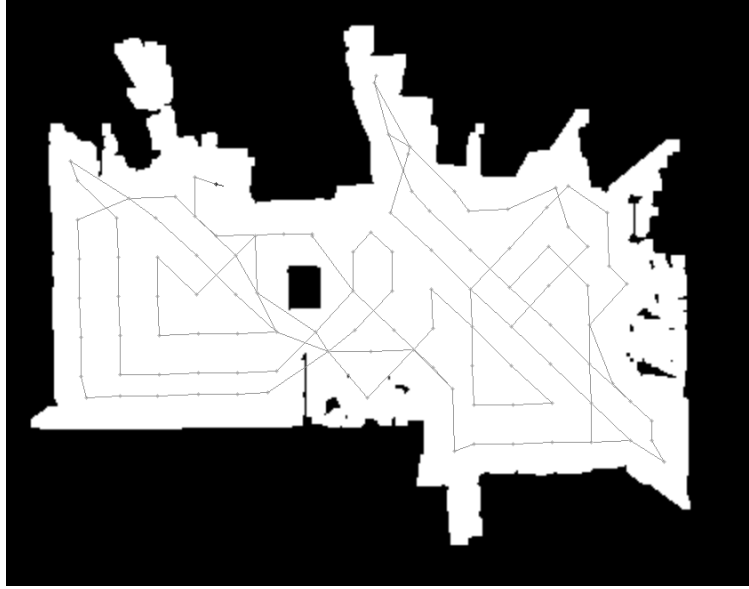
Şekil 3.11. Sinir Ağı Tabanlı Kapsama Yol Planlama Algoritması Simülasyon Ortamı Yol Planı

Tablo 3.3. Sinir ağı tabanlı kapsama yol planlama ile elde edilen performans metrikleri

Ortam Haritaları	Kapsama Yüzdesi [%]	Seyahat Süresi [ss:dd:sn]	Seyahat Mesafesi [m]	Dönüş Sayısı [rad]	Hesaplama Süresi [ss:dd:sn]
Simülasyon	96,9	00:10:04	302,4	182,12	00:01:18
Atölye	94,4	00:03:16	98,5	73,79	00:01:52
Fabrika	90,7	00:26:12	786,7	616,6	00:14:21

Sinir ağı tabanlı yol planlama algoritmasının performans metrikleri, farklı ortam türlerinde gösterdiği başarıyı ve sınırlılıkları kapsamlı bir şekilde değerlendirme imkânı sunmaktadır. Kapsama yüzdesi, seyahat süresi ve mesafesi, dönüş sayısı ve hesaplama süresi gibi metrikler üzerinden yapılan analizler, algoritmanın farklı boyut ve karmaşıklık seviyelerindeki ortamlar için uygunluğunu ortaya koymaktadır. Genel olarak, algoritma tüm ortam türlerinde yüksek bir kapsama oranı (>95%) sağlayarak kapsama görevinde

etkili olduğunu kanıtlamıştır. Özellikle atölye ortamında %96,9 ile en yüksek kapsama oranına ulaşılmış, bu başarının daha düzenli ve az karmaşık yapıdaki ortam koşullarından kaynaklandığı düşünülmektedir. Buna karşılık, fabrika ortamında %90,7 kapsama oranı, büyük alanın ve karmaşık engel yapısının etkisini yansıtmaktadır.



Şekil 3.12. Sinir Ağı Tabanlı Kapsama Yol Planlama Algoritması Atölye Ortamı Yol Planı

Seyahat süresi ve mesafesi açısından, simülasyon ortamı küçük bir alan olması nedeniyle kısa seyahat süreleri 10 dakika ve mesafeleri, 302,4 metre, ile verimli bir performans sergilemiştir. Atölye ortamı ise 3 dakika seyahat süresi ve 98,5 metre mesafe ile en düşük değerlere ulaşmış, bu da algoritmanın bu tür ortamlarda hız ve enerji verimliliği sağladığını göstermektedir. Ancak fabrika ortamında, alanın genişliği ve karmaşıklığı nedeniyle seyahat süresi 26 dakika 12 saniyeye, mesafe ise 786,7 metreye çıkmıştır.

Dönüş sayısı performans değerlendirmesinde önemli bir faktör olarak öne çıkmaktadır. Simülasyon ortamında dönüş sayısının (182,12) küçük bir alan için yüksek olduğu, bunun da enerji verimliliğini olumsuz etkileyebileceği görülmüştür. Atölye ortamında dönüş sayısı (73,79) en düşük seviyede gerçekleşmiş ve bu durum algoritmanın düzenli yol planlama yeteneğini vurgulamıştır. Öte yandan, fabrika ortamında dönüş sayısının (616,6) oldukça yüksek olduğu, bunun da büyük ve karmaşık alanlarda sıklıkla yön değiştirmenin kaçınılmaz olduğunu gösterdiği söylenebilir.



Şekil 3.13. Sinir Ağı Tabanlı Kapsama Yol Planlama Algoritması Fabrika Ortamı Yol Planı

Hesaplama süreleri açısından ise simülasyon ortamında elde edilen 1 dakika süre, küçük ve az karmaşık alanlarda algoritmanın hızlı bir şekilde işlem yapabildiğini göstermiştir. Atölye ortamında bu süre yaklaşık 2 dakika olarak gerçekleşmiş ve kabul edilebilir bir düzeyde kalmıştır. Ancak fabrika ortamında 14 dakikaya ulaşan hesaplama süresi, büyük ve karmaşık ortamların algoritmanın hesaplama maliyetini önemli ölçüde artırdığını ortaya koymaktadır.

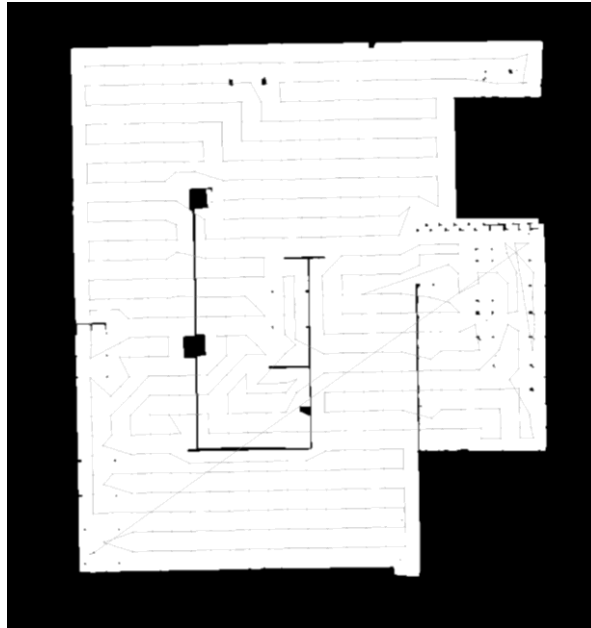
Genel olarak, küçük ve orta karmaşıklıkta ortamlarda algoritma yüksek performans sergilerken, büyük ve karmaşık alanlarda enerji tüketimi ve hesaplama süresi gibi konularda optimizasyon gereksinimi ortaya çıkmaktadır. Bu nedenle, algoritmanın büyük ve karmaşık ortamlar için dönüş sayısını azaltmaya ve hesaplama süresini optimize etmeye yönelik iyileştirilmesi önerilmektedir. Böyle bir optimizasyon, sinir ağı tabanlı yol planlama algoritmasının geniş bir uygulama yelpazesinde daha etkili ve verimli bir çözüm sunmasını sağlayabilir.

3.3.4. Enerji Fonksiyonel Optimizasyonu Yöntemi ile Deney Sonuçları

Enerji fonksiyonel optimizasyonu yönteminde, bir harita grid tabanlı bir yapıya ayrılarak kapsama yol planlaması gerçekleştirilmiştir. Haritanın her noktasını kapsamak amacıyla,

düğümlemler belirlenmiş ve bu düğümlemler robotun hareketini optimize edecek şekilde enerji fonksiyonel yaklaşımı ile sıralanmıştır. Serbest alan üzerindeki her düğüm, çevresindeki komşularla olan bağlantılar ve enerji fonksiyonu kullanılarak değerlendirilmiş, robotun bir sonraki adımı bu değerlendirmeler sonucunda seçilmiştir. Farklı ortamlardaki performansı değerlendirmek için yöntem, simülasyon, atölye ve fabrika ortamlarına ait haritalar üzerinde test edilmiştir. Simülasyon ortamında elde edilen kapsama yolu Şekil 3.14.'te, atölye ortamında elde edilen kapsama yolu Şekil 3.15.'te ve fabrika ortamında elde edilen kapsama yolu ise Şekil 3.16.'da sunulmuştur.

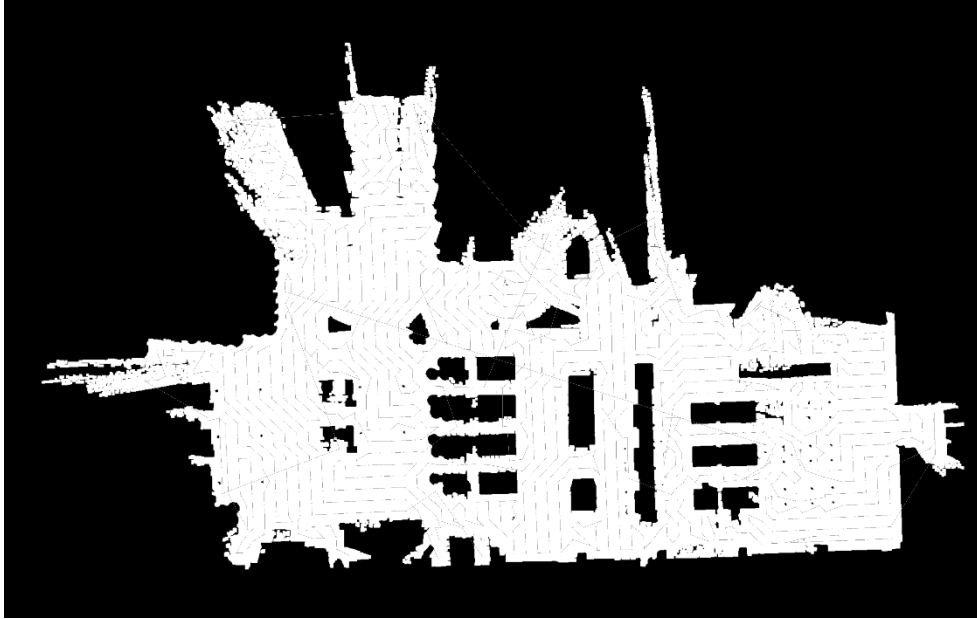
Enerji fonksiyonel optimizasyon yöntemiyle yapılan kapsama planlaması, düğüm yerleşiminden rota seçimine kadar birçok boyutta etkili sonuçlar sunmakla birlikte, çeşitli alanlarda iyileştirme potansiyelleri barındırmaktadır. Algoritmanın performansı üzerinde yapılan analizler, üç farklı görsel üzerinden detaylandırılarak incelenmiştir.



Şekil 3.14. Enerji Fonksiyonel Optimizasyonu ile Kapsama Yol Planlama Algoritması Simülasyon Ortamı Yol Planı



Şekil 3.15. Enerji Fonksiyonel Optimizasyonu ile Kapsama Yol Planlama Algoritması Atölye Ortamı Yol Planı



Şekil 3.16. Enerji Fonksiyonel Optimizasyonu ile Kapsama Yol Planlama Algoritması Fabrika Ortamı Yol Planı

Şekil 3.14.'teki simülasyon ortamı, algoritmanın küçük ve düzenli ortamlarda oldukça verimli çalıştığını göstermektedir. Yolların paralel ve düzenli bir yapıda olması, sade ortamlarda enerji tüketiminin minimum seviyede tutulduğunu ortaya koymaktadır.

Bununla birlikte, dönüşlerin az olması, algoritmanın engellerin bulunmadığı ortamlarda üstün bir performans sergilediğini doğrulamaktadır. Yöntemin gereksiz yol uzatmalarını önleme çabası dikkat çekmekte ve enerji fonksiyonunun daha az maliyetli yolları tercih etmek üzere tasarlandığını ortaya koymaktadır. Bu doğrultuda, algoritmanın daha karmaşık haritalarda test edilmesi ve düğüm seçimindeki önceliklerin genişletilmesi gerektiği sonucuna varılmıştır.

Şekil 3.15.'teki atölye ortamında enerji fonksiyonunun etkili bir şekilde kullanıldığı ve rota seçiminde mantıksal bir yaklaşım sergilendiği gözlemlenmiştir. Düğümler arası bağlantılar genel olarak iyi düzenlenmiş olsa da uzak düğümler arasında yapılan sıçramaların gereksiz enerji harcamasına yol açabileceği tespit edilmiştir. Bazı bölgelerde yol çizgilerinin serbest alanı tam olarak kapsamadığı fark edilmiştir ve bu durum, büyük alanların optimizasyon sırasında ağırlıklı olarak tercih edilmesiyle ilişkilendirilmektedir.

Şekil 3.16.'daki fabrika ortamındaki düğüm dağılımı, serbest alanların etkili bir şekilde kapsandığını göstermekle birlikte, küçük ve engellerin yoğun bulunduğu bölgelerde düğüm yoğunluğunun performansı etkileyebileceği gözlemlenmiştir. Yolların düzgün ve belirgin bir şekilde oluşturulması, enerji fonksiyonunun rotaları oluşturmadaki başarısını ortaya koymaktadır. Ancak, dar alanlarda ve keskin dönüş gerektiren noktalarda enerji tüketiminin arttığı değerlendirilmektedir. Görseldeki küçük boşluklar, tam kapsama sağlanamadığı durumlara işaret etmekte ve bu eksikliğin enerji fonksiyonunun komşu düğümleri değerlendirirken bu bölgeleri göz ardı etmesinden kaynaklanabileceği düşünülmektedir. Bu bağlamda, dönüş sayılarını ve küçük hücrelerin kapsama oranını artırmak amacıyla enerji fonksiyonuna "kapsanmamış alanların önceliği" gibi ek ağırlıklandırmalar yapılması önerilmektedir. Yöntemin performans metriklerine göre sonuçları Tablo 3.4.'te gösterilmiştir.

Tablo 3.4. Sinir ağı tabanlı kapsama yol planlama ile elde edilen performans metrikleri

Ortam Haritaları	Kapsama Yüzdesi [%]	Seyahat Süresi [ss:dd:sn]	Seyahat Mesafesi [m]	Dönüş Sayısı [rad]	Hesaplama Süresi [ss:dd:sn]
Simülasyon	96,2	00:02:36	313,5	175,84	00:02:18
Atölye	94,4	00:01:08	134,5	75,36	00:04:17
Fabrika	89,2	00:09:05	1076,4	602,88	00:42:25

3.3.5. Ortalama Kayma Kümeleme Yöntemi ile Deney Sonuçları

Ortalama Kayma (Mean Shift) Kümeleme yöntemi, veri noktalarını belirli bir yoğunluk fonksiyonunu maksimize edecek şekilde gruplayarak kümeler oluşturmak için kullanılan bir optimizasyon tekniğidir. Şekil 3.17.'de simülasyon ortamında, Şekil 3.18.'de atölye ortamında algoritmanın sonuçları gösterilmektedir. Robotik kapsama alanı planlamasında, bu yöntem, kapsanması gereken belirli alanları kümelere ayırarak, her bir küme merkezinde robotun hareket edip gözlem yapabileceği stratejik pozisyonları belirler. Ortalama Kayma algoritması, başlangıçta rastgele bir noktadan başlar ve ardından her bir adımda daha yüksek yoğunluklu noktalara doğru kayarak kümelerin merkezlerini bulur.

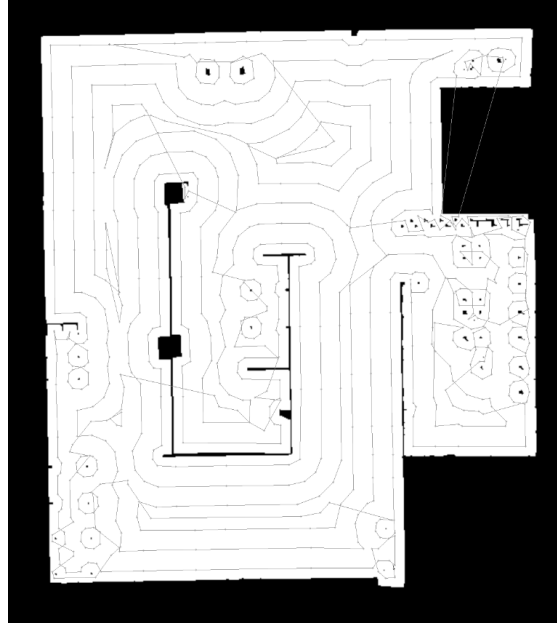
Bu yöntemde veri noktaları, yoğunluk fonksiyonuna dayalı olarak kümeler halinde gruplanır. Her veri noktasının belirli bir komşuluk alanı vardır ve her iterasyonda yoğunluk fonksiyonu yardımıyla bu nokta kümelerin yoğunluk merkezlerine doğru kaydırılır. Kümeler yeterince yoğun hale geldiğinde, iterasyon durur ve son noktalar kümelerin merkezleri olarak tanımlanır. Bu merkez noktalar, robot için gözlem yapılması gereken pozisyonlar olarak belirlenebilir. Yöntemin performans metriklerine göre sonuçları Tablo 3.5.'te gösterilmiştir.

Tablo 3.5. Ortalama kayma kümeleme kapsama yol planlama algoritması ile yapılan çalışmada elde edilen performans metrikleri

Ortam Haritaları	Kapsama Yüzdesi [%]	Seyahat Süresi [ss:dd:sn]	Seyahat Mesafesi [m]	Dönüş Sayısı [rad]	Hesaplama Süresi [ss:dd:sn]
Simülasyon	95,1	00:05:16	363,5	329,7	00:014:17
Atölye	94,9	00:10:04	76	155,43	00:018:17
Fabrika	X	X	X	X	X

Ortalama Kayma Kümeleme yönteminin avantajları arasında, parametresiz bir yöntem olması ve küme sayısının önceden belirtilmesine gerek olmaması yer alır. Bu durum, robotik kapsama alanı gibi karmaşık ortamlarda, gözlem yapılacak alanların otomatik olarak belirlenmesini sağlar. Ayrıca, bu yöntem, herhangi bir sınırlayıcı varsayım olmadan verinin doğal yoğunluk yapısını ortaya çıkarır ve kümelerin sayısı, veri yoğunluğuna göre otomatik olarak belirlenir. Bu nedenle, dinamik ve yoğunluk farklılıkları içeren ortamlarda robotun konum planlamasını esnek ve uyarlanabilir bir hale

getirir. Bu sebeple, fabrika ortamında hesaplamaların aşırı uzun sürmesi nedeniyle uygulama verimli bir şekilde gerçekleştirilememiştir. Bu yüzden Tablo 3.5.'te değerler X olarak gösterilmiştir.



Şekil 3.17. Ortalama Kayma Kümeleme Yöntemi ile Kapsama Yol Planlama Algoritması Simülasyon Ortamı Yol Planı

Yöntemin dezavantajları arasında özellikle karmaşık haritalarda çok sayıda iterasyon gerektirmesi nedeniyle hesaplama yükünün artması bulunur. Ortalama Kayma algoritmasının her iterasyonda yüksek yoğunluk noktalarına doğru kaydırma işlemi, büyük veri setlerinde veya karmaşık ortam haritalarında hesaplama süresini uzatabilir. Bu durum, gerçek zamanlı uygulamalarda performans sorunlarına yol açabilir. Ayrıca, algoritma her ne kadar otomatik olarak yoğunluk merkezlerini belirlese de özellikle çok düşük veya çok yüksek yoğunluklu bölgelerde algoritmanın başarısı düşebilir. Bu gibi durumlarda, kümelenmenin belirgin olmadığı bölgelerde gözlem noktalarının belirlenmesinde zorluk yaşanabilir.

Ortalama Kayma Kümeleme yöntemi, robotik kapsama planlamasında özellikle gözlem yapılacak noktaların belirlenmesinde etkili bir yöntemdir. Ancak, karmaşık ve yoğun engel bulunan ortamlarda hesaplama maliyeti artabileceğinden, gerçek zamanlı kapsama uygulamalarında kullanımına dikkat edilmelidir.

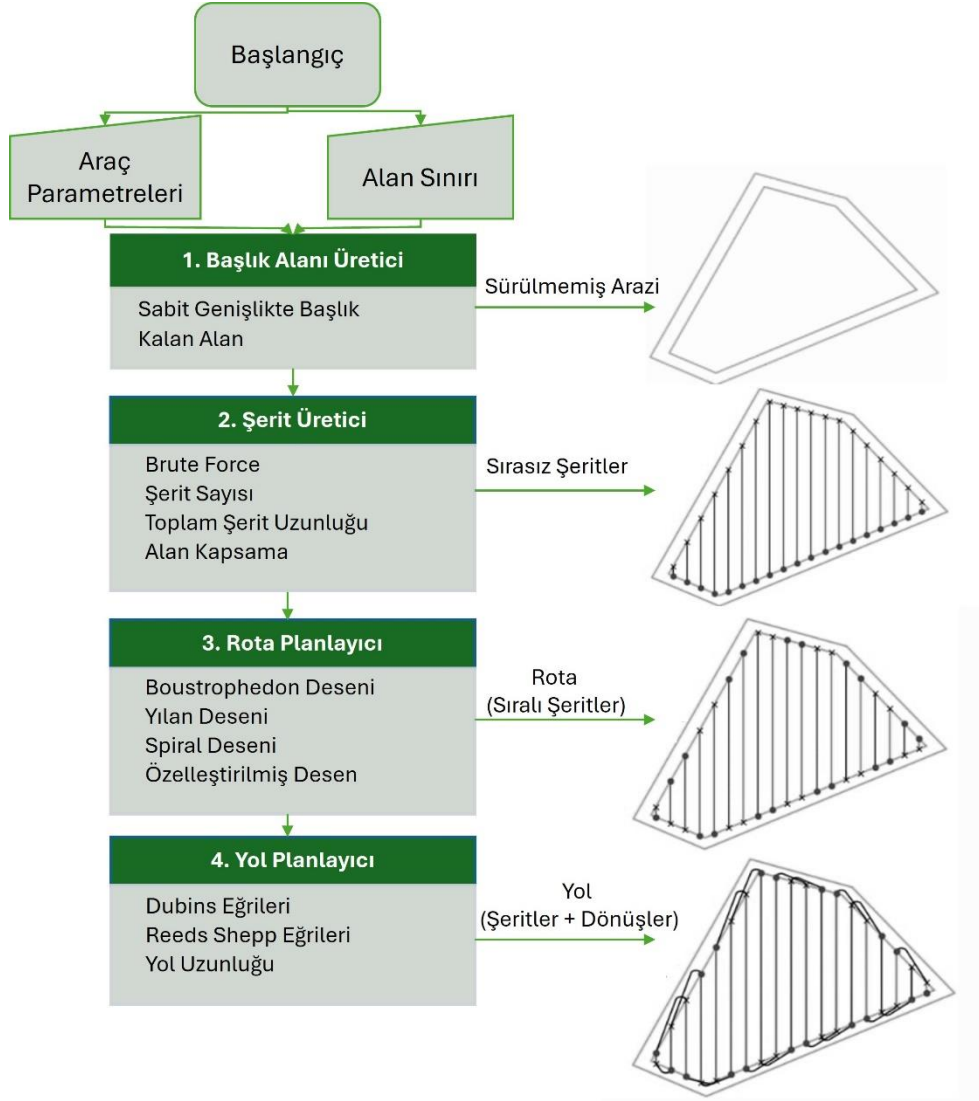


Şekil 3.18. Ortalama Kayma Kümeleme Yöntemi ile Kapsama Yol Planlama Algoritması Atölye Ortamı Yol Planı

3.3.6. Fields2Cover ve nav2_coverage Paketlerinin İncelenmesi

Fields2Cover, insansız tarım araçları için kapsama yolu planlaması problemini ele alan açık kaynaklı bir C++ kütüphanesi olarak tasarlanmıştır. Bu kütüphane, özellikle tarım uygulamaları için modüler ve genişletilebilir bir mimari sunarak, çeşitli araç türlerini ve arazi yapısını destekleme kapasitesine sahiptir (Mier ve diğ., 2023).

Şekil 3.19.'de akış şeması verilen kütüphane, farklı araç gereksinimlerine ve planlama problemlerine uygun olarak başlık alanı üretimi (headland generator), şerit üretimi (swath generator), rota planlayıcı (route planner) ve yol planlayıcı (path planner) gibi temel modüller sunmaktadır. Başlık alanı üretimi modülü, alanın sınırını belirli bir genişlikte tamponlayarak sınır çizgilerini tanımlar. Hat üretimi modülü, başlıklar hariç alanın iç kısmını araçların çalışma genişliğiyle uyumlu paralel hatlara böler. Rota planlayıcı, bu hatların sıralanmasını gerçekleştirirken, yol planlayıcı araç parametreleri ve rota bilgileri doğrultusunda dönüş hesaplamalarını yapar.



Şekil 3.19. Fields2Cover Kütüphanesi Akış Şeması

Algoritma, öncelikli olarak tarımsal alanları geometrik olarak işler. Bu işlem, alanın sınırlayıcı poligonlarının çıkarılması, iç engellerin tespiti ve alanın kapsama yollarını kolaylaştırmak için alt bölümlere ayrılmasıyla gerçekleştirilir. Alanın sınırları genellikle dışbükey zarf (Convex Hull) veya döndürülmüş sınırlayıcı kutu (Rotated Bounding Box) yöntemleriyle modellenir. Bu işlemin ardından, alanın tamamını kapsayacak şekilde paralel doğrusal yollar oluşturulur. Paralel doğruların açıları, alanın topolojik özelliklerine ve minimum dönüş açısını sağlayacak şekilde optimize edilmiştir.

Fields2Cover, yol planlama aşamasında düz ve eğrisel yolları birleştirir. Düz çizgi temelli planlama, engellerden arındırılmış alanlarda kullanılırken, eğrisel yollar dar veya düzensiz şekilli alanlarda tercih edilir. Eğrisel yol planlama sürecinde Bezier eğrileri veya

spline tabanlı yöntemler uygulanarak, dönüşlerin daha yumuşak olması sağlanır. Yol planlama algoritmaları, enerji ve yakıt tüketimini en aza indirmek için optimize edilirken, gereksiz tekrarlar ve boş hareketler minimize edilir. Alanın kapsanma süreci sırasında Dijkstra veya A* arama algoritması gibi grafik temelli algoritmalar, en kısa ve en verimli yolları hesaplamak için kullanılır.

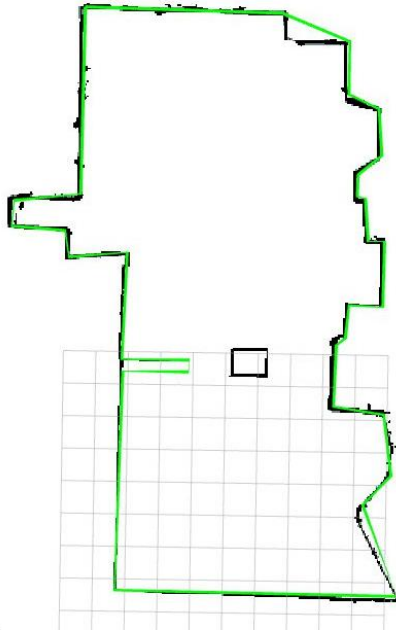
Başlık (Headland) yönetimi, Fields2Cover algoritmasının önemli bir bileşenidir. Tarımsal alanların dış sınırları boyunca tampon bölgeler oluşturularak, bu bölgelerde dönüşlerin etkili bir şekilde gerçekleştirilmesi sağlanır. Başlık bölgelerinde, araçların dönüş hareketleri Dubins ve Reeds-Shepp eğrileri gibi kinematik tabanlı yöntemlerle planlanır. Dubins eğrileri, ileri hareket kısıtlaması olan araçlar için tercih edilirken, Reeds-Shepp eğrileri ileri ve geri hareketlerin gerektiği durumlarda kullanılır. Bu yöntemler, dönüşlerin enerji maliyetini azaltırken alanın kapsama sürecini hızlandırır.

Algoritma, alan segmentasyonu aşamasında iki temel yaklaşımı benimser. Grid tabanlı segmentasyon, alanı sabit boyutlu hücrelere ayırarak her hücre için ayrı kapsama yolları oluşturur. Poligon tabanlı segmentasyon ise daha dinamik bir yaklaşımla, alanın geometrik yapısına uygun alt bölümler oluşturur. Bu esneklik, Fields2Cover'ın farklı tarımsal alan tiplerine kolayca uyum sağlamasını sağlar.

Nav2_coverage paketi, ROS 2'nin navigasyon çerçevesine entegre edilerek Fields2Cover'ın özelliklerini kullanır. Böylelikle, ROS 2 tabanlı robotlar için kapsama görevleri daha erişilebilir hale gelir. Fields2Cover, bağımsız bir kütüphane olarak algoritma çeşitliliği sunarken, nav2_coverage bu özellikleri ROS 2 ekosistemine taşımaktadır. Pakette kapsama planlama algoritmalarının sınırlamaları dikkat çekmektedir. Paket içerisinde kullanılan Fields2Cover kütüphanesi, bazı durumlarda iç boşluklar veya iç engeller içeren alanların kapsanmasında yeterli performansı gösterememektedir. Bu durum, özellikle karmaşık veya düzensiz şekilli alanlarda, eksik veya yetersiz kapsama ile sonuçlanabilmektedir. Bu tür sınırlamalar, robotların görev etkinliğini olumsuz yönde etkileyebilmekte ve kapsamlı bir çözüm gerektirmektedir (Mier ve diğ., 2023).

ROS 2 Humble dağıtımında, endüstriyel süpürge robotunun navigasyon parametreleri ile nav2_coverage paketi test edilmiştir. Bu çalışmalar kapsamında, atölye ortamı haritasının

sınırları kapsama yol planlama sürecinde kullanılmak üzere kullanılmıştır. Harita sınırlarının, nav2_coverage paketinin gereksinim duyduğu poligon formatında ifade edilmesini sağlamak amacıyla, özel bir yazılım paketi geliştirilmiştir. Söz konusu paket, haritanın sınırlarını hassas bir şekilde analiz ederek, nav2_coverage alan formatına uyumlu bir çıktı üretmektedir. Geliştirilen bu yaklaşım hem sistemin entegrasyonunu kolaylaştırmış hem de navigasyon testlerinin doğruluğunu artırmıştır. Şekil 3.20.'de, bu paketin ürettiği örnek bir çıktının rviz2 üzerindeki görselleştirilmiş versiyonu gösterilmektedir.

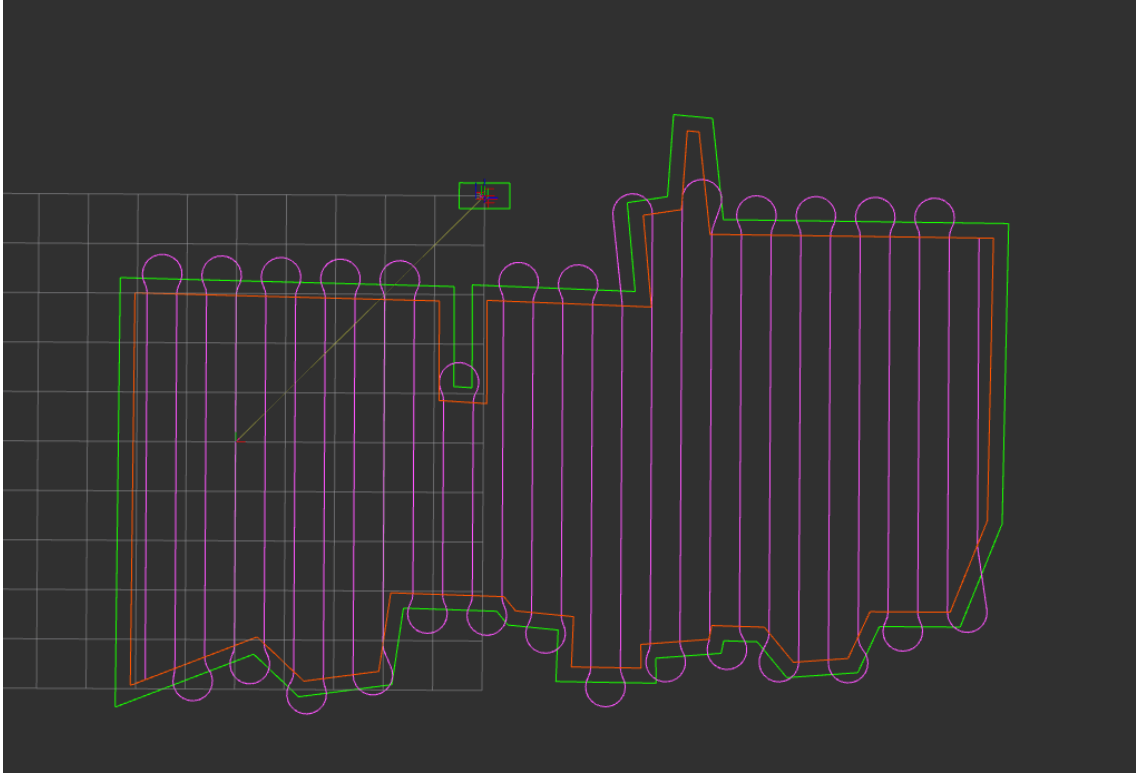


Şekil 3.20. Atölye Ortamı Haritasının Sınırlarının Poligon Formatında Gösterimi

Opennav_coverage paketi içerisinde Python tabanlı bir uygulama geliştirilmiştir. Uygulama, robotun verilen bir alanda kapsama navigasyonu gerçekleştirmesini sağlamak için NavigateCompleteCoverage aksiyon arayüzünü kullanmaktadır (URL-7) . Kod, bir alanın geometrisini poligon olarak tanımlayarak, bu sınırları robotun kapsama hedefi olarak belirlemekte ve navigasyon sürecini başlatmaktadır. Görev sırasında robotun durumu izlenmekte ve süreçle ilgili geri bildirimler alınmaktadır.

Uygulamada, atölye ortamının sınırları bir poligon veri yapısıyla kullanılmıştır. Bu poligon, robotun kapsama görevi sırasında izlemesi gereken alanı temsil etmektedir. Algoritmanın çalıştığı alanın rviz2 üzerindeki görüntüsü Şekil 3.21.'de gösterilmiştir.

Test süreci boyunca robot, verilen sınırlar içinde hareket ederek belirlenen rotayı başarıyla tamamlamış ve kapsama görevini yerine getirmiştir. Görev sırasında alınan geri bildirimler, sürecin tahmini tamamlanma süresi gibi önemli bilgiler sağlamıştır.



Şekil 3.21. Opennav_coverage Paketi Atölye Ortamı Yol Planı

Fields2cover algoritmasının dönüşlerdeki temel dayanağı, Dubins ve Reeds-Shepp eğrileri gibi kinematik yöntemlere dayanır. Dubins ve Reeds-Shepp eğrileri minimum yol uzunluğunu ve dönüş sayısını hedeflerken, endüstriyel süpürge robotlarında öncelikli amaç, bir alanın eksiksiz şekilde temizlenmesidir. Bu tür eğrilerin geometrik kısıtlamalara dayalı olması, dar alanlarda robotun ayak izine göre uygun bir çözüm sunamayabilir.

Opennav_coverage algoritmasının bu hedefe ulaşmadaki zayıflığı, dönüşlerin geometrik olarak optimize edilmesi nedeniyle bazı alanların eksik bırakılmasına veya örtüşmesine neden olabilir. Özellikle karmaşık engellerin bulunduğu düzensiz alanlarda bu algoritmanın kapsama oranı düşebilir ve temizlik süreçleri optimize edilemeyebilir. Geri hareketlerin dahil edilmesiyle birlikte planlama süresinin ve enerji tüketiminin artması da bu yöntemin önemli bir dezavantajıdır.

Bununla birlikte, algoritmanın avantajları arasında belirli geometrik kısıtlamalara sahip alanlarda etkin bir şekilde çalışabilmesi ve kinematik açıdan uygun yollar oluşturabilmesi yer alır. Dubins ve Reeds-Shepp eğrileri, minimum dönüş yarıçapı gerektiren durumlarda robotun manevra kabiliyetini artırabilir. Ayrıca, dinamik engellerin bulunduğu ortamlarda, araçların hareket kabiliyetlerini kısıtlamadan yol planlaması yapabilme kapasitesi, bu algoritmayı belirli uygulamalar için uygun hale getirmektedir.

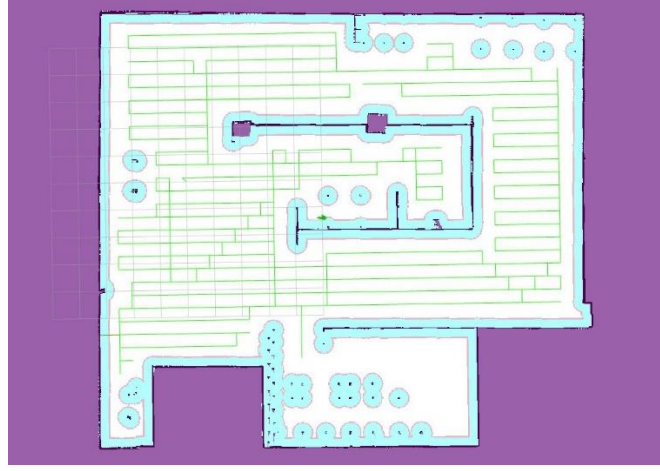
Bir diğer önemli sorun, `opennav_coverage` algoritmasında gidilecek alanın poligon şeklinde tanımlanması gerekliliğidir. Bu durum, haritanın ortasında kalan bölgelerin belirlenmesi için ek kısıtlar oluşturmakta ve algoritmanın esnekliğini sınırlamaktadır. Ayrıca, her yeni harita için ek hesaplamalar yapılmasını gerektiren bu durum, algoritmanın işlem süresini artırmakta ve kullanım kolaylığını azaltmaktadır. Bu özellik, algoritmayı tarım uygulamaları gibi daha düzenli ve geniş alanlarda kullanıma uygun hale getirirse de endüstriyel süpürge robotlar gibi karmaşık iç mekanlarda yüksek verimle çalışmasını sınırlamaktadır. Ancak, bu dezavantajların giderilmesi amacıyla geliştirme çalışmaları aktif olarak devam etmektedir. Gelecekte yapılacak optimizasyonlarla birlikte, `opennav_coverage` algoritmasının endüstriyel süpürge robotları için daha uygun bir çözüm sunması beklenmektedir.

3.3.7. Önerilen Yöntemin Performans Değerlendirmesi

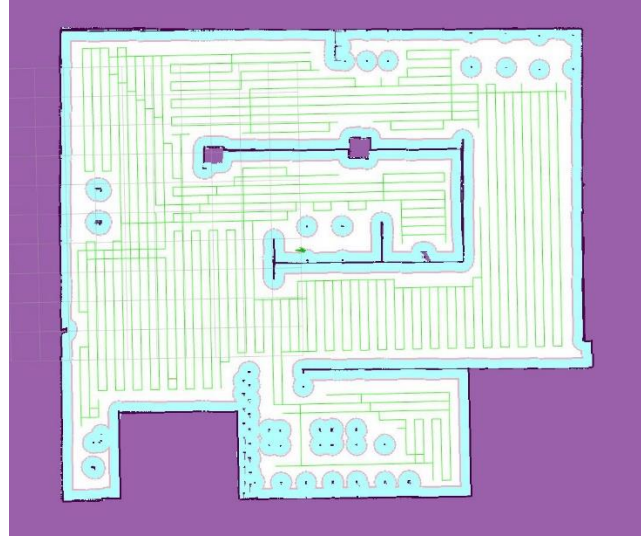
Geliştirilen yöntem, özellikle geniş ve yapılandırılmış endüstriyel alanlarda seçilen poligonlar üzerinde gerçekleştirdiği kapsama planlamasıyla, geleneksel oda segmentasyonuna dayalı planlayıcılara göre daha hedefe yönelik ve pratik bir çözüm sunmaktadır. Bu yaklaşım, endüstriyel uygulamalarda yaygın olarak karşılaşılan dar alanlar, yoğun engel bölgeleri ve karmaşık geometrilerde yüksek doğruluk ve etkinlik göstermiştir.

Yöntemin güvenli ve kararlı çalışma ilkeleri doğrultusunda, ROS 2 tabanlı bir global planlayıcı çerçevesi altında geliştirilen algoritma, engellerin şişirilmesi için belirlenen yarıçap değeriyle optimize edilmiştir. Şekil 3.22.'de simülasyon ortamında önerilen algoritmanın ürettiği yol ve şişirilen engeller mavi ile gösterilmiştir. Bu sayede, robotun çarpışma riskini azaltacak korunaklı mesafeler sağlanırken, uygulanabilir kapsama alanlarının tespiti mümkün kılınmıştır. Ancak çok dar geçitlerin bulunduğu bölgelerde,

şişirilmiş engellerden geriye kalan alan robotun fiziksel boyutlarına uygun olmayabilir. Şekil 3.23.'te şişirme marjı düşürülerek robotun geçebileceği alan bırakılmıştır. Bu sayede, sıkışık veya yoğun biçimde engel barındıran alanlara planlamamaktadır.



Şekil 3.22. Önerilen Algoritma Simülasyon Ortamı Yol Planı

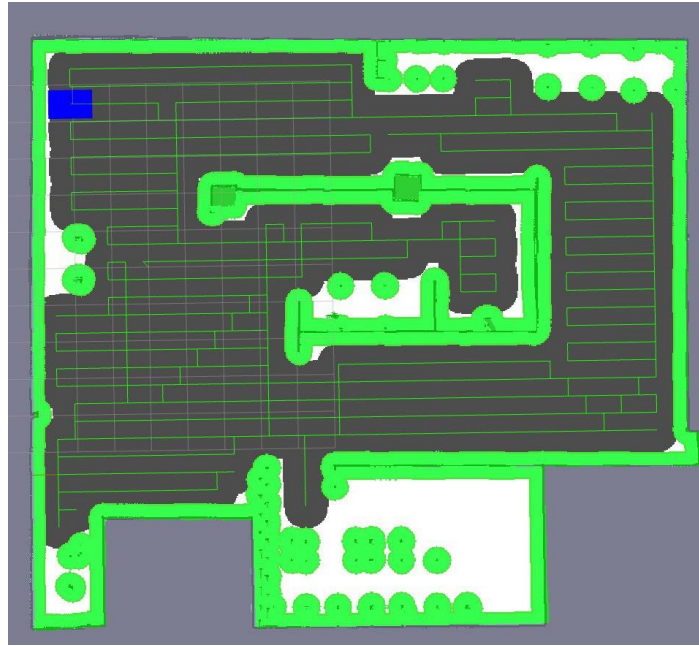


Şekil 3.23. Önerilen Algoritma Düşük Şişirme Yarıçapı Uygulanması

Şişirme yarıçapının artırılması, yol planlamasının uygulanabilirliğini sınırlandırarak kapsama oranını düşürebilmektedir. Bununla birlikte yöntem, dar alanlarda dahi yüksek başarı oranı sergilemekte, endüstriyel temizlik veya taşıma gibi görevlerde robotun güvenli geçişini garanti altına almaktadır.

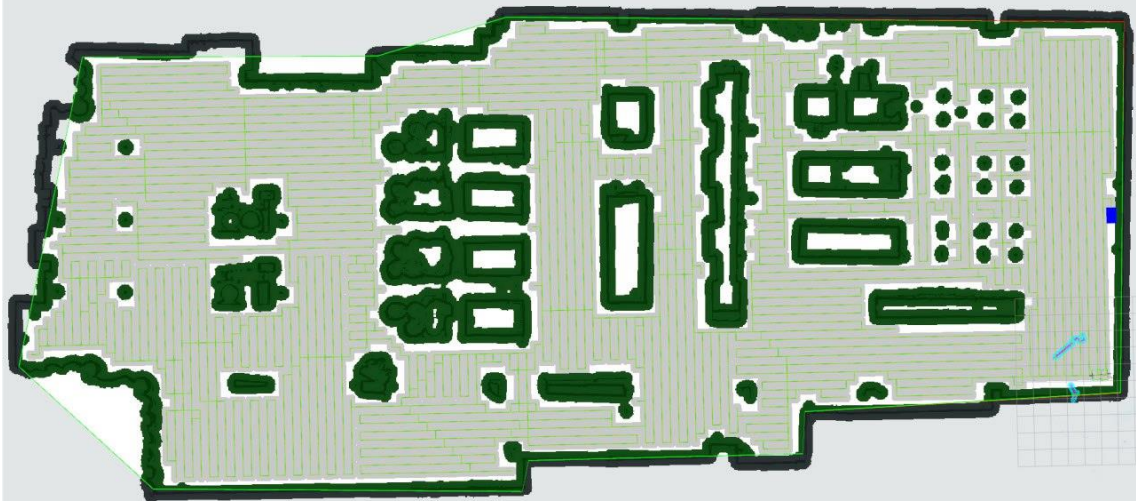
Şekil 3.24.'te robotun geçtiği yerler koyu gri ile gösterilmiştir. Simülasyon ortamında yapılan testlerde, önerilen algoritmanın başlangıç koşullarında yaklaşık %92 oranında

kapsama sağladığı gözlemlenmiştir. Bu oranın nispeten düşük kalması, robotun fiziksel olarak sığamayacağı ve dolayısıyla erişilemeyen çok dar bölgelerin kapsama planından çıkarılmasının doğrudan bir sonucu olarak değerlendirilebilir. İlerleyen aşamalarda, global planlayıcıda engellerin şişirme yarıçapının 0.4'e düşürülmesiyle robotun manevra kabiliyetini artıracak şekilde planlama yapılmış, böylece robotun geçiş yapabileceği tüm alanlar yeniden kapsama altına alınmıştır. Bu düzenleme, kapsama oranında anlamlı bir iyileşme sağlayarak değeri %96'ya yükseltmiştir.



Şekil 3.24. Önerilen Algoritma ile Robotun Geçtiği Yerlerin Gösterimi

Fabrika ortamında gerçekleştirilen testlerde de benzer eğilimler gözlenmiştir. Şekil 3.25.'te önerilen algoritmanın fabrika ortamında oluşturduğu yol planı gösterilmiştir. Dar alanlardaki geçişler için planlanan yolların kapsanması, yöntemin karmaşık ortamlarda dahi güvenilir sonuçlar verdiğini göstermiştir. Bu ortamdaki deneylerde kapsama oranı yaklaşık %92 olarak elde edilmiş, robotun geçtiği konumlar gri renk ile temsil edilerek yöntemin görsel olarak da doğrulanması sağlanmıştır. Elde edilen sonuçlar, önerilen yöntemin hem simülasyon hem de gerçek fabrika ortamlarında güvenli, uygulanabilir ve esnek bir kapsama planlaması sunduğunu ortaya koymaktadır. Böylece robotun fiziksel boyutları, engel konfigürasyonları ve ortamın yapısal özellikleri dikkate alınarak yapılan özelleştirilmiş kapsama yol planlamasının, endüstriyel uygulamalarda başarı ve verimliliği artırma potansiyeli açıkça ortaya konmuştur.



Şekil 3.25. Önerilen Algoritma Fabrika Ortamı Yol Planı

Yöntem, ROS 2 tabanlı bir global planlayıcı çerçevesinde uygulanmış ve alanların etkili bir şekilde kapsanmasını sağlamak için belirli bir şişirme yarıçapı kullanılarak optimize edilmiştir. Şişirme yarıçapı, engellerin güvenli bir mesafeden şişirilmesini sağlayarak, robotun çarpışma riskini azaltırken, geriye kalan kapsanabilir alanların belirlenmesini mümkün kılmıştır. Ancak, bu süreçte, şişirilmiş engellerden kalan alanın robotun fiziksel boyutuna uygun olmaması durumunda yol planlaması yapılamamaktadır. Bu, özellikle dar alanlarda veya engellerin yoğun olduğu bölgelerde, yol planlama algoritmasının geçiş uygunluğu sağlaması açısından önemli bir sınırlama olarak karşımıza çıkmaktadır. Robotun yarıçapına göre dinamik olarak değerlendirilen bu alanlar, başarılı bir kapsama planlaması için temel parametrelerden biridir.

Testler simülasyon, atölye ve fabrika ortamlarında gerçekleştirilmiştir. Simülasyon ortamında kullanılan özel bir paket, robotun kapsama yol planlamasını önce sanal bir ortamda test ederek, algoritmanın doğruluğunu ve etkinliğini değerlendirmek için kullanılmıştır. Simülasyon, farklı poligon boyutları ve engel yoğunlukları için algoritmanın performansını analiz etme fırsatı sunmuştur. Bu ortamda, kapsama yüzdesi, yol uzunluğu, seyahat süresi, hesaplama süresi dönüş sayısı gibi metrikler detaylı bir şekilde analiz edilmiştir.

Fabrika ortamında gerçekleştirilen testlerde ise, gerçek dünya koşullarına uyum sağlama yeteneği ön planda değerlendirilmiştir. Robot, belirtilen poligon içerisinde, şişirilmiş engellerden geriye kalan geçiş uygun alanlarını kullanarak kapsamlı bir yol üretmiştir.

Robotun, yarıçapı dikkate alınarak planlanan yolları, global planlayıcı tarafından güvenli bir şekilde izlenmiştir. Bu süreçte, dinamik engeller ve değişken fabrika koşullarında algoritmanın kararlılığı ve adaptasyonu da incelenmiştir. Tablo 3.6.'da önerilen yöntemin test sonuçları gösterilmiştir.

Tablo 3.6. Önerilen Yöntem Test Sonuçları

Ortam Haritaları	Kapsama Yüzdesi [%]	Seyahat Süresi [ss:dd:sn]	Seyahat Mesafesi [m]	Dönüş Sayısı [rad]	Hesaplama Süresi [ss:dd:sn]
Simülasyon	98,3	00:02:49	339	210,2	00:00:06
Atölye	95,4	00:01:22	145,5	90,4	00:00:07
Fabrika	94,7	00:06:10	698,1	425,3	00:00:14

Elde edilen sonuçlar, önerilen yöntemin hem simülasyon hem de gerçek ortamda etkili bir şekilde çalıştığını göstermektedir. Yöntem, özellikle dar ve karmaşık alanlarda, robotun güvenli geçişini garanti altına alarak, endüstriyel temizlik ve taşıma gibi görevlerde yüksek bir performans sergilemiştir. Bu çalışma, kapsama yol planlamasının, robotun fiziksel boyutları ve fabrika ortamının özellikleri dikkate alınarak özelleştirilmesinin, endüstriyel uygulamalarda başarıyı artırmak için kritik öneme sahip olduğunu ortaya koymaktadır.

4. SONUÇLAR VE ÖNERİLER

Bu çalışmada geliştirilen yöntem, endüstriyel fabrika ortamlarında belirlenen poligonlar üzerinde kapsama yol planlaması için özelleştirilmiş bir yaklaşım sunmaktadır. Global planlayıcıda ayarlanan şişirme yarıçapı, robotun güvenli geçiş alanlarının belirlenmesi ve engellerden uzak mesafelerde çalışmasını sağlamış, böylece çarpışma riskini en aza indirmiştir. Simülasyon ortamında yapılan testler, algoritmanın kapsama yüzdesi, dönüş sayısı ve yol uzunluğu gibi metriklerde başarılı bir performans sergilediğini göstermiştir. Gerçek fabrika ortamında gerçekleştirilen uygulamalar ise yöntemin dinamik koşullara uyum sağlama yeteneğini ortaya koymuştur. Özellikle karmaşık ve dar alanlarda, robotun fiziksel boyutuna uygun kapsama yollarının oluşturulması yöntemin etkinliğini artırmıştır. Her bir yöntemin gerçekleştirilen deneyler sonucunda elde edilmiş performans metrik sonuçları Tablo 4.1.'de verilmiştir.

Tablo 4.1. Performans Metriklerine Göre Yöntemlerin Karşılaştırılması

Yöntem	Ortam Haritaları	Kapsama Yüzdesi [%]	Seyahat Süresi [ss:dd:sn]	Seyahat Mesafesi [m]	Dönüş Sayısı [rad]	Hesaplama Süresi [ss:dd:sn]
Izgara Tabanlı Gezgin Satıcı Problemi	Simülasyon	99,8	00:02:47	335,2	271,61	00:00:12
	Atölye	98,7	00:00:51	103,9	102,05	00:00:18
	Fabrika	94,8	00:06:03	748,3	794,42	00:00:30
Boustrophedon	Simülasyon	97,5	00:02:50	341,5	188,4	00:00:04
	Atölye	93,2	00:01:33	187,2	81,64	00:00:04
	Fabrika	76,8	00:06:18	578,7	317,14	00:00:10
Sinir Ağı Tabanlı	Simülasyon	96,9	00:10:04	302,4	182,12	00:01:18
	Atölye	94,4	00:03:16	98,5	73,79	00:01:52
	Fabrika	90,7	00:26:12	786,7	616,6	00:14:21
Enerji Fonksiyonel	Simülasyon	96,2	00:02:36	313,5	175,84	00:02:18
	Atölye	94,4	00:01:08	134,5	75,36	00:04:17
	Fabrika	89,2	00:09:05	1076,4	602,88	00:42:25
Ortalama Kayma Kümeleme	Simülasyon	95,1	00:05:16	363,5	329,7	00:14:17
	Atölye	94,9	00:10:04	76	155,43	00:18:17
	Fabrika	X	X	X	X	X
Önerilen Yöntem	Simülasyon	98,3	00:02:49	339	210,2	00:00:06
	Atölye	95,4	00:01:22	145,5	90,4	00:00:07
	Fabrika	94,7	00:06:10	698,1	425,3	00:00:14

Bu çalışma, altı farklı kapsama yol planlama yönteminin (Izgara Tabanlı Gezgin Satıcı Problemi, Boustrophedon, Sinir Ağları, Enerji Fonksiyonel, Ortalama Kayma Kümeleme ve önerilen yöntem) üç farklı ortamda (simülasyon, Birfen atölyesi ve fabrika) elde ettiği

performans verilerini karşılaştırarak kapsamlı bir değerlendirme sunmaktadır. Kapsama yüzdesi açısından incelendiğinde, Izgara Tabanlı Gezgin Satıcı Problemi ve önerilen yöntem, özellikle dar veya orta ölçekli ortamlarda (simülasyon ve atölye) oldukça yüksek kapsama oranları sağlayarak başarılı sonuçlar elde etmiştir. Bu iki yöntem, kapsama yüzdesini yüzde 95–99 aralığında tutarak sistematik bir şekilde alanın büyük bir bölümünü kapsamıştır. Boustrophedon yöntemi, atölye ortamında yüzde 93,2 seviyesinde kalmış, ancak simülasyonda yüzde 97,5 gibi yüksek bir kapsama oranına ulaşmıştır. Fabrika ortamında ise, engellerin artışı ve alanın genişlemesi, Boustrophedon ve Sinir Ağları yöntemlerinin kapsama oranlarını sırasıyla yüzde 76,8 ve yüzde 90,7 seviyelerine çekmiştir. Buna karşın, Izgara Tabanlı Gezgin Satıcı Problemi ve önerilen yöntem yüzde 94'ün üzerinde bir kapsama oranı yakalayarak, karmaşık ortamlarda dahi etkin kapsama kabiliyeti göstermiştir.

Seyahat süresi ve mesafesi açısından değerlendirildiğinde, Izgara Tabanlı Gezgin Satıcı Problemi yönteminin fabrika ortamında 6 dakikayı aşan seyahat süresi ve 748 metreye ulaşan toplam mesafesi, yüksek kapsama oranı sağlamak adına daha fazla seyahat maliyetine yol açmaktadır. Boustrophedon yöntemi, nispeten daha kısa sürede ve daha düşük maliyetle orta düzeyde bir kapsama performansı sunmuş, özellikle atölye ortamında yaklaşık 1,5 dakikalık süre ile avantaj sağlamıştır. Ancak, fabrika ortamında seyahat süresi yine 6 dakikanın üzerine çıkarak yöntemler arasında ortalama bir konumda kalmıştır. Önerilen yöntem, simülasyon ve atölye ortamlarında Boustrophedon ve Izgara Tabanlı Gezgin Satıcı Problemi yöntemlerine kıyasla daha dengeli bir performans göstermiş, fabrika ortamında ise seyahat süresi ve toplam mesafe açısından kontrol edilebilir seviyelerde kalarak karmaşık alanlarda kapsama ve süre/maliyet arasında optimal bir denge sağlamıştır.

Dönüş sayısı bakımından yapılan analizde, Izgara Tabanlı Gezgin Satıcı Problemi yönteminin fabrika ortamında 794 gibi yüksek bir dönüş sayısına ulaşması, bu yöntemin rota optimizasyonunu fazla sayıda yön değişikliği yaparak gerçekleştirdiğini göstermektedir. Boustrophedon ve önerilen yöntem, dar alanlarda (atölye ortamı) dönüş sayılarını 80–90 seviyesinde tutabilirken, geniş ve çok engelli ortamlarda dönüş sayıları 300–400 bandına yükselmektedir. Bu durum, kapsama için zigzag veya dallanma

mantığıyla yol izleyen yöntemlerde, ortamın karmaşıklığına bağlı olarak yön değişimlerinin arttığını göstermektedir.

Hesaplama süresi açısından yapılan değerlendirmelerde, simülasyon gibi düzenli ve küçük ölçekli alanlarda yöntemlerin hesaplama sürelerinin genellikle birkaç saniye seviyesinde kaldığı görülmüştür. Örneğin, Boustrophedon yöntemi yaklaşık 4–5 saniyelik bir hesaplama süresi sunar, Izgara Tabanlı Gezgin Satıcı Problemi yöntemi yaklaşık 12 saniyelik bir işlem süresi gerektirmiştir. Bununla birlikte, fabrika ortamı gibi geniş ve karmaşık haritalarda, bazı yöntemlerin 30–40 saniye veya dakikalar süren hesaplama sürelerine ulaştığı gözlemlenmiştir. Önerilen yöntem, fabrika ortamında yaklaşık 14 saniyelik bir hesaplama süresi ile Izgara Tabanlı Gezgin Satıcı Problemi (30 saniye) ve Enerji Fonksiyonel (42 saniye) gibi yöntemlere kıyasla daha hızlı sonuç üretmiş, ancak Boustrophedon yöntemine göre biraz daha uzun bir hesaplama süresi sergilemiştir. Bu durum, büyük ve engelli ortamlarda gerçek zamanlı olmamakla birlikte, pratik anlamda yönetilebilir bir işlem süresi sunduğunu göstermektedir.

Genel değerlendirme yapıldığında, Izgara Tabanlı Gezgin Satıcı Problemi ve önerilen yöntemin çok engelli veya geniş ortamlarda kapsama oranını yüksek tutmayı başardığı gözlemlenmiştir. Boustrophedon yöntemi, daha küçük veya orta ölçekli alanlarda zamansal avantaj sağlamakla birlikte, karmaşık ortamlarda kapsama yüzdesinin düştüğü belirlenmiştir. Sınır Ağları ve Enerji Fonksiyonel yaklaşımları, orta karmaşıklıkta ortamlarda iyi bir performans gösterirken, fabrika gibi daha zorlu alanlarda hesaplama süresi ve kapsama oranı açısından bazı kısıtlarla karşılaşmıştır. Ortalama Kayma Kümeleme yöntemi, simülasyon ve atölye ortamlarında test edilmiştir ve düşük dönüş sayısı ile kısa mesafeler sağlarken, fabrika ortamında yeterli veri bulunmadığı için bu yönteme ilişkin sonuçlar eksik kalmıştır. Bu yüzden, değerler X olarak gösterilmiştir.

Sonuçlar, farklı yöntemlerin avantaj ve dezavantajlarını net bir şekilde ortaya koymaktadır. Daha basit ve dar ortamlarda Boustrophedon ve Sınır Ağları gibi yöntemler, kısa işlem süresi ile yeterli kapsama oranı sağlayabilirken, daha karmaşık ve geniş ortamlarda Izgara Tabanlı Gezgin Satıcı Problemi ve önerilen yöntem, yüksek kapsama oranı ile kabul edilebilir seyahat ve hesaplama süreleri sunarak daha başarılı sonuçlar elde etmektedir.

Bu tez çalışmasının katkıları şu şekilde özetlenebilir: Endüstriyel fabrika ortamlarında belirlenen iş bölgelerine yönelik poligon tabanlı bir kapsama yol planlama yöntemi geliştirilmiş ve robotun fiziksel boyutunu ve geçiş uygunluğunu dikkate alan bir yol planlama mekanizması tanımlanmıştır. ROS 2 tabanlı bir kapsama planlama sistemi hem simülasyon hem de gerçek dünya ortamlarında başarılı bir şekilde uygulanmış; ayrıca, şişirme yarıçapı ile şişirilen engellerin güvenli yol planlamasına etkisi ayrıntılı biçimde analiz edilmiştir. Kapsama yüzdesi, dönüş sayısı, yol uzunluğu ve enerji tüketimi gibi performans metrikleriyle algoritma klasik yöntemlerle karşılaştırılmış; yöntem, dinamik engellerin bulunduğu ortamlarda dahi yüksek kararlılık ve doğrulukla çalışacak şekilde tasarlanmıştır.

Sonuçlar, önerilen yöntemin hem simülasyon hem de gerçek ortamda yüksek bir doğruluk ve kararlılık sergilediğini göstermektedir. Yöntem, poligon bazlı kapsama planlamasının, endüstriyel uygulamalarda hedeflenen alanların verimli bir şekilde taranması için uygun bir çözüm olduğunu ortaya koymaktadır. Bununla birlikte, şişirilmiş engellerin geride robotun geçişine izin vermeyecek kadar dar alanlar oluşturduğu durumlarda yol planlaması yapılamadığı tespit edilmiştir. Bu durum, algoritmanın dar alanlardaki performansını artırmak için ilave optimizasyon gerekliliğini göstermektedir.

Gelecek çalışmalar için, dar alanların daha etkili bir şekilde kapsanmasını sağlamak amacıyla robotun dinamik davranışlarını optimize edecek yeni bir algoritma tasarlanabilir. Ayrıca, yöntem, farklı robot türleri ve daha geniş alanlarda test edilerek geliştirilebilir. Özellikle, gerçek zamanlı dinamik engel tespiti ve bu engellere adaptasyon sağlayan mekanizmalar ile yöntemin uygulanabilirliği daha da artırılabilir. Bu bulgular, önerilen yöntemin endüstriyel temizlik ve taşıma uygulamaları gibi görevlerde güçlü bir çözüm sunduğunu ve gelecekteki iyileştirmeler için geniş bir potansiyele sahip olduğunu göstermektedir.

KAYNAKLAR

- Almadhoun, R., Taha, T., Seneviratne, L., Zweiri, Y. (2019). A survey on multi-robot coverage path planning for model reconstruction and mapping. *Springer Nature Applied Sciences*, 1(8), 847.
- Bähnemann, R., Lawrance, N., Chung, J. J., Pantic, M., Siegwart, R., Nieto, J. (2021). Revisiting Boustrophedon Coverage Path Planning as a Generalized Traveling Salesman Problem. In: *Ishigami, G., Yoshida, K. (Eds.), Field and Service Robotics. Springer Proceedings in Advanced Robotics*, 16(1), 227-290.
- Barraquand, J., Latombe, J.-C. (1991). Robot Motion Planning: A Distributed Representation Approach. *The International Journal of Robotics Research*, 10(6), 628-649.
- Bormann, R., Hampp, J., Hagele, M. (2015). New brooms sweep clean - An autonomous robotic cleaning assistant for professional office cleaning. *Proceedings - IEEE International Conference on Robotics and Automation*, 4470-4477.
- Bormann, R., Jordan, F., Hampp, J., Hagele, M. (2018). Indoor Coverage Path Planning: Survey, Implementation, Analysis. *Proceedings - IEEE International Conference on Robotics and Automation*, 1718-1725.
- Bormann, R., Jordan, F., Li, W., Hampp, J., Hägele, M. (2016). Room segmentation: Survey, implementation, and analysis. *Proceedings - IEEE International Conference on Robotics and Automation, 2016-June*, 1019-1026.
- Cai, Z., Li, S., Gan, Y., Zhang, R., Zhang, Q. (2014). Research on Complete Coverage Path Planning Algorithms based on A* Algorithms. İçinde *The Open Cybernetics & Systemics Journal* (C. 8).
- Carvalho, J. P., Aguiar, A. P. (2023). A Reinforcement Learning Based Online Coverage Path Planning Algorithm. *2023 IEEE International Conference on Autonomous Robot Systems and Competitions, ICARSC 2023*, 81-86.
- Chen, X., Tucker, T. M., Kurfess, T. R., Vuduc, R. (2019). Adaptive Deep Path: Efficient Coverage of a Known Environment under Various Configurations. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3549-3556.
- Choi, Y.-H., Lee, T.-K., Baek, S.-H., Oh, S.-Y. (2009). Online complete coverage path planning for mobile robots based on linked spiral paths using constrained inverse distance transform. *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5788-5793.
- Choset, H., Pignon, P. (1998). Coverage Path Planning: The Boustrophedon Cellular Decomposition. *Field and Service Robotics*, 203-209.

- Choton, J. C., Prabhakar, P. (2023). Optimal Multi-Robot Coverage Path Planning for Agricultural Fields using Motion Dynamics. *Proceedings - IEEE International Conference on Robotics and Automation, 2023-May*, 11817-11823.
- Dakulović, M., Horvatić, S., Petrović, I. (2011). Complete Coverage D* Algorithm for Path Planning of a Floor-Cleaning Mobile Robot. *IFAC Proceedings Volumes*, 44(1), 5950-5955.
- Guo, S., Huang, Y., You, D., Wang, T., Long, Q. (2022). Path planning algorithm for sweeping robot full traversal cleaning area. *Proceedings - 2022 Asia Conference on Advanced Robotics, Automation, and Control Engineering, ARACE 2022*, 127-132.
- Li, J., Wang, W., Wu, E. (2007). Point-in-polygon tests by convex decomposition. *Computers and Graphics (Pergamon)*, 31(4), 636-648.
- Liu, Z., Zhou, W. (2023). Energy-Efficient Algorithms for Path Coverage in Sensor Networks. *Sensors*, 23(11).
- Lu, J., Zeng, B., Tang, J., Lam, T. L., Wen, J. (2023). TMSTC*: A Path Planning Algorithm for Minimizing Turns in Multi-Robot Coverage. *IEEE Robotics and Automation Letters*, 8(8), 5275-5282.
- Macenski, S., Martin, F., White, R., Clavero, J. G. (2020). The Marathon 2: A Navigation System. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2718-2725.
- Mahajan, A., Rock, S. (2020). Completeness seeking probabilistic coverage estimation using uncertain state estimates. *IEEE International Conference on Intelligent Robots and Systems*, 6832-6837.
- Maruyama, Y., Kato, S., Azumi, T. (2016). Exploring the performance of ROS2. *2016 International Conference on Embedded Software (EMSOFT)*, 1-10.
- Mier, G., Valente, J., de Bruin, S. (2023). Fields2Cover: An Open-Source Coverage Path Planning Library for Unmanned Agricultural Vehicles. *IEEE Robotics and Automation Letters*, 8(4), 2166-2172.
- Neumann, R., De Carvalho, I., Vidal, H. A., Vieira, P., Ribeiro, M. I., Pais, A. R. (1997). Complete Coverage Path Planning and Guidance for Cleaning Robots. *IEEE International Symposium on Industrial Electronics*, 2, 677-682.
- Noh, D. K., Lee, W. J., Kim, H. R., Cho, I. S., Shim, I. B., Baek, S. M. (2022). Adaptive Coverage Path Planning Policy for a Cleaning Robot with Deep Reinforcement Learning. *Digest of Technical Papers - IEEE International Conference on Consumer Electronics*,
- Oksanen, T., Visala, A. (2007). Path Planning Algorithms for Agricultural Machines. *Agricultural Engineering International: The CIGR Journal*.

- Pérez-González, A. É., Benítez-Montoya, N., Jaramillo-Duque, Á., Cano-Quintero, J. B. (2021). Coverage path planning with semantic segmentation for UAV in PV plants. *Applied Sciences (Switzerland)*, 11(24).
- Quigley, M., Conley, K., Gerkey, B. P., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A. Y. (2009). ROS: an open-source Robot Operating System. *ICRA Workshop on Open Source Software*.
- Ranjitha, T. D., Guruprasad, K. R. (2016). Pseudo spanning tree-based complete and competitive robot coverage using virtual nodes. *IFAC-PapersOnLine*, 49(1), 195-200.
- Rodrigues, R. T., Aguiar, A. P., Pascoal, A. (t.y.). *A coverage planner for AUVs using B-splines*.
- Šeda, M. (2007). *Roadmap Methods vs. Cell Decomposition in Robot Motion Planning*. Proceedings of the 6th WSEAS International Conference on Signal Processing, Robotics and Automation
- Šelek, A., Seder, M., Brezak, M., Petrović, I. (2022). Smooth Complete Coverage Trajectory Planning Algorithm for a Nonholonomic Robot. *Sensors*, 22(23).
- Tan, X., Han, L., Gong, H., Wu, Q. (2023). Biologically Inspired Complete Coverage Path Planning Algorithm Based on Q-Learning. *Sensors*, 23(10).
- URL-1: <https://wiki.ros.org/ROS/Introduction>, (Ziyaret Tarihi: 04 Kasım 2024).
- URL-2: <https://onnarobotics.com/mapping-ve-slam-nedir/>, (Ziyaret Tarihi: 28 Aralık 2024).
- URL-3: <https://google-cartographer-ros.readthedocs.io/en/latest/#>, (Ziyaret Tarihi: 28.12.2024).
- URL-4: <https://birfenrobotics.com.tr/robovak-temizlik-robotu/>, (Ziyaret Tarihi: 16 Kasım 2024).
- URL-5: <https://docs.ros.org/en/foxy/index.html>, (Ziyaret Tarihi: 11 Kasım 2025).
- URL-6: <https://docs.nav2.org/configuration/packages/configuring-dwb-controller.html>, (Ziyaret Tarihi: 28 Kasım.2024).
- URL-7: https://github.com/open-navigation/opennav_coverage, (Ziyaret Tarihi: 17 Aralık 2024)
- Xing, B., Wang, X., Yang, L., Liu, Z., Wu, Q. (2023). An Algorithm of Complete Coverage Path Planning for Unmanned Surface Vehicle Based on Reinforcement Learning. *Journal of Marine Science and Engineering*, 11(3).

Yang, S. X., Luo, C. (2004). A Neural Network Approach to Complete Coverage Path Planning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34(1), 718-725.

Yue, W., Jiang, Z. (2018). Path planning for UAV to Collect Sensors Data Based on Spiral Decomposition. *Procedia Computer Science*, 131, 873-879.

KİŞİSEL YAYIN VE ESERLER

- Gürkal, A., İner, A.B., **Özer, E.**, Kibar, A. (2021). Yük Kapasitesi Şase Yapısından Bağımsız Bir Otonom Mobil Robot Tasarımı ve Dayanım Analizi. *Mühendis ve Makina*, 62(704), 607-619.
- Kibar, A., Gürkal, A.E., **Özer, E.**, İner, A.B. (2023). Diferansiyel Sürüşlü Otonom Mobil Robotların Avare Teker Kaynaklı Sapmaların İncelenmesi, *Politeknik Dergisi*, 27(3), 947-955.
- Mando, R., **Özer, E.**, İner, B. (2022). Autonomous Mobile Robot Mapping And Exploration Of An Indoor Environment. International Conference on Computing. *Intelligence and Data Analytics*, 643(1), 17-26.
- Özer, E.**, İner, A.B. (2024). Endüstriyel Temizlik Robotları İçin Geliştirilen Yeni Bir Boustrophedon Tabanlı Kapsama Algoritması: Performans İyileştirmeleri ve Uygulama. *Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimleri Dergisi*.
- Özer, E.**, İner, A.B. (2021). Tekstil Sektörü için ROS Destekli Otonom Mobil Robot Geliştirilmesi. *Avrupa Bilim ve Teknoloji Dergisi*, 26(1), 482-487.

ÖZGEÇMİŞ

2012 yılında girdiđi İstanbul Bilgi Üniversitesi Mühendislik Fakültesi Elektrik ve Elektronik Mühendisliđi Bölümü'nden 2017 yılında mezun oldu. 2017 yılından itibaren otonom mobil robot yazılımları geliştiren bir firmanın Ar-Ge biriminde çeşitli pozisyonlarda görev almış, bu kapsamda sektördeki lider firmalarla ortak projeler yürütmüş ve akademik çalışmalar gerçekleştirmiştir.