

T.C.  
İNÖNÜ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

**BİNA İÇİ LOJİSTİK FAALİYETLER İÇİN  
BULUT TABANLI OTONOM ROBOT YÖNETİMİ**

**DOKTORA TEZİ**

**Fatih OKUMUŞ**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Tez Danışmanı: Doç. Dr. Adnan Fatih KOCAMAZ**

**ARALIK 2020**

**T.C  
İNÖNÜ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**BİNA İÇİ LOJİSTİK FAALİYETLER İÇİN  
BULUT TABANLI OTONOM ROBOT YÖNETİMİ**

**DOKTORA TEZİ**

**Fatih OKUMUŞ  
(23616190009)**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Tez Danışmanı: Doç. Dr. Adnan Fatih KOCAMAZ**

**ARALIK 2020**

## TEŞEKKÜR VE ÖNSÖZ

Doktora eğitimimde ve hayatımın her anında bana destek olan akademik bilgi ve tecrübelerinin yanı sıra hayat tecrübelerini ve bilgi birikimini sabır ve cömertlikle benimle paylaşan kıymetli danışman hocam Doç.Dr. Adnan Fatih KOCAMAZ'a,

Lisans eğitimimden bu yana bana destek olan, öneri, bilgi ve tecrübesiyle yol gösteren kıymetli hocam Prof.Dr. Ali KARCI'ya, yol gösterici önerilerini ve yardımlarını esirgemeyen Doç.Dr. Barış Baykant ALAGÖZ'e,

Tez süresince beraber yaptığımız projeler ve yayınlardaki katkıları ve gayretleriyle yanımda olan Dr. Öğr. Üyesi Emrah DÖNMEZ'e, bilgi ve birikimlerini benimle paylaşan ekip arkadaşlarım Dr. Fahrettin Burak DEMİR ve Yahya ALTUNTAŞ'a,

İnönü Üniversitesi Bilgisayar Mühendisliği Bölümünde görevli olan tüm hocalarıma ve idari personellerine,

Beni yetiştiren, bugünlere gelmemi sağlayan, tez çalışmam süresince benden desteklerini esirgemeyen ve üzerimde büyük hakları olan anneme ve babama,

Tez çalışması süresince ihmal ettiğim, sıkıntılarımı, mutluluklarımı ve hayatımı paylaştığım sevgili Eşim Hatice OKUMUŞ'a, oğlum Hamza Miraç'a ve kızım Asya'a,

Tezin uygulaması için vermiş olduğu proje destekleriyle maddi ve manevi katkı sağlayan, İnönü Üniversitesi BAP birimine,

Çalışma ortamının modellenmesinde bizlere üretim tesislerinin kapılarını açan, araştırmalara maddi-manevi büyük katkı ve uzmanlık sağlayan, ihtiyaçları doğrultusunda teze ilham kaynağı olan ve yarı zamanlı Ar-Ge personeli olarak çalıştığım Çalık Denim A.Ş. Ar-Ge Merkezi'ne,

*Şükranlarımı sunar, teşekkür ederim.*

## ONUR SÖZÜ

Doktora veya yüksek lisans tezi olarak sunduđum “Bina İçi Lojistik Faaliyetler İin Bulut Tabanlı Otonom Robot Yönetimi” başlıklı bu alıřmanın bilimsel ahlak ve geleneklere aykırı düřecek bir yardıma bařvurmaksızın tarafımdan yazıldıđına ve yararlandıđım bütün kaynakların hem metin iinde hem de kaynakada yöntemine uygun biçimde gösterilenlerden oluřtuđunu belirtir, bunu onurumla dođrularım.

Fatih OKUMUŐ



## İÇİNDEKİLER

TEŞEKKÜR VE ÖNSÖZ .....	i
ONUR SÖZÜ .....	ii
İÇİNDEKİLER.....	iii
ÇİZELGELER DİZİNİ.....	v
ŞEKİLLER DİZİNİ.....	vi
SEMBOLLER VE KISALTMALAR.....	viii
ÖZET .....	ix
ABSTRACT .....	x
<b>1. GİRİŞ .....</b>	<b>1</b>
1.1 İlişkili Çalışmalar.....	6
1.1.1 Haritalama ve konumlama üzerine yapılan çalışmalar.....	7
1.1.2 Yol planlama çalışmaları.....	9
1.1.3 Çoklu robot görev dağılımı çalışmaları.....	10
1.1.4 Bulut robotik çalışmaları.....	14
1.1.5 Tez çalışmasının benzer çalışmalarla karşılaştırılması.....	16
1.2 Tezin Amacı.....	18
1.3 Tez Çalışmasının Gerekçeleri.....	18
1.4 Tezin Yenilikçi Yönü ve Ar-Ge Niteliği .....	19
1.5 Tezin Çıktıları.....	20
1.6 Tezin Organizasyonu .....	20
<b>2. ENDÜSTRİYEL İŞLETMELERDE BİNA İÇİ LOJİSTİK .....</b>	<b>22</b>
2.1 Bina İçi Lojistikte Etken Unsurlar .....	23
2.2 Bina İçi Lojistik Süreçleri.....	23
2.3 Otonom Bina İçi Lojistiğin Önemi .....	24
2.4 Vaka Örneği Olarak Tekstil Terbiye İşletmeleri .....	26
2.5 Vaka Örneğindeki Bina İçi Lojistik Problemleri.....	30
<b>3. ENDÜSTRİYEL LOJİSTİK AMAÇLI OTONOM MOBİL     ROBOTLARDA TEMEL PROBLEMLER ve ÇÖZÜM     YAKLAŞIMLARI .....</b>	<b>31</b>
3.1 Eş Zamanlı Haritalama ve Konumlama.....	31
3.1.1 Min-Max ile konum tahmini .....	32
3.1.2 Trilaterasyon (üçleme) ile konum tahmini .....	34
3.1.3 Maksimum benzerlik ile konum tahmini.....	35
3.1.4 Odometri ile konum tahmini .....	36
3.1.5 Parmak izi tabanlı konum tahmin teknikleri .....	38
3.1.6 Kalman filtresi tabanlı konum tahmini.....	38
3.2 Yol Planlama .....	39
3.2.1 A* algoritması .....	40
3.2.2 D* algoritması .....	41
3.2.3 D* Lite algoritması.....	41
3.2.4 Dijkstra algoritması .....	43
3.2.5 PSO algoritması.....	43
3.3 Çoklu Robot Görev Dağılımı.....	44
3.4 Robot Hareket Modelleme ve Robot İşletim Sistemi (ROS).....	46
3.5 Otonom ürün/malzeme yükleme bırakma .....	47
<b>4. MATERYAL VE METOT .....</b>	<b>49</b>
4.1 Sistem Materyalleri ve Çalışma Modeli .....	49
4.2 Geliştirilen Bulut Sistem Mimarisi .....	52
4.2.1 Haritalama ve konum takibi .....	54

4.2.1.1 RESTful Web Servisler .....	55
4.2.1.2 WebSocket Metotları.....	56
4.2.2 Lojistik görevi oluşturma .....	58
4.2.3 Robot batarya optimizasyonu.....	59
4.2.4 Yol Planlama .....	60
4.2.5 Taşıma görevleri için en uygun yük arabasının seçimi.....	61
4.2.6 Önerilen optimum çoklu robot görev dağılımı yöntemi.....	63
4.2.7 Dinamik engellerden kaçınma.....	68
4.3 Robot Kontrol Mimarisi .....	69
4.3.1 Konumlama Sistemi .....	70
4.3.2 Otonom hareket modelleme ve navigasyon .....	72
<b>5. DENEYSEL SONUÇLAR.....</b>	<b>74</b>
5.1 Simülasyon Sonuçları.....	74
5.1.1 Yol planlama algoritmalarının karşılaştırılması.....	80
5.2 Laboratuvar (Gerçek Ortam) Sonuçları.....	82
5.2.1 Konumlama sistemi sonuçları .....	83
5.2.2 Bulut tabanlı navigasyon sonuçları .....	84
<b>6. SONUÇLAR VE ÖNERİLER.....</b>	<b>88</b>
6.1 Sonuçların Değerlendirilmesi .....	90
6.2 Öneriler.....	91
<b>KAYNAKLAR.....</b>	<b>93</b>
<b>ÖZGEÇMİŞ.....</b>	<b>104</b>

## ÇİZELGELER DİZİNİ

<b>Çizelge 1.1</b> : Tez kapsamında geliştirilen otonom çoklu robot yönetim sisteminin öne çıkan ilişkili çalışmalarla karşılaştırılması.....	<b>17</b>
<b>Çizelge 5.1</b> : Tanımlanan ürünler ve görevleri.....	<b>77</b>
<b>Çizelge 5.2</b> : Örnek deney için gerçekleşen lojistik yönetiminin sonuçları.....	<b>77</b>
<b>Çizelge 5.3</b> : Önerilen görev atama yönteminin rastgele atama yöntemiyle karşılaştırılması. ....	<b>80</b>
<b>Çizelge 5.4</b> : Gerçek ortam konumlama hassasiyeti. ....	<b>83</b>
<b>Çizelge 5.5</b> : Teker kinematığı ve önerilen sistem ile oluşan toplam sapmaların karşılaştırılması.....	<b>85</b>



## ŞEKİLLER DİZİNİ

Şekil 2.1 :	Bina içi genel ürün/malzeme akış çizelgesi. ....	24
Şekil 2.2 :	Kumaş yüklenmiş dok arabası. ....	26
Şekil 2.3 :	Elektrikli dok çekici. ....	27
Şekil 2.4 :	İnsan gücüyle çalışan dok çekici. ....	27
Şekil 2.5 :	Dok çekicinin dok arabasına bağlanması. ....	28
Şekil 2.6 :	Dok çekici seyir koridoru. ....	28
Şekil 2.7 :	Terbiye makinesi örneği. ....	29
Şekil 2.1 :	Örnek kumaş terbiye işletmesi krokisi. ....	29
Şekil 3.1 :	Min-Max metodu ile lokasyon tahmini. ....	33
Şekil 3.2 :	Trilaterasyon metodu ile lokasyon tahmini. ....	34
Şekil 3.3 :	Lineer Odometri. ....	36
Şekil 3.4 :	Dönüş Odometrisi. ....	37
Şekil 3.5 :	Sola Dönüş Teker Kinematığı. ....	37
Şekil 3.6 :	A* algoritması akış diyagramı. ....	40
Şekil 3.7 :	D* Lite algoritması akış diyagramı. ....	42
Şekil 3.8 :	Dijkstra algoritması akış diyagramı. ....	43
Şekil 4.1 :	Otonom Lojistik için modellenmiş kumaş terbiye işletmesi. ....	51
Şekil 4.2 :	İki farklı kumaşın 9 terbiye makinesinin bulunduğu ortamda taşınma senaryosu. ....	52
Şekil 4.3 :	Bulut tabanlı otonom mobil robot yönetim sistemi mimarisi. ....	53
Şekil 4.4 :	Modellenmiş bir ortam haritası örneği. ....	55
Şekil 4.5 :	Başlangıç görevi örneği. ....	58
Şekil 4.6 :	Kumaşın M <sub>1</sub> makinesinde işlenmesi için oluşturulan giriş portu görevi ve çıkış portu görevi örnekleri. ....	59
Şekil 4.7 :	Teslimat görevi örneği. ....	59
Şekil 4.8 :	R ve G kümeleri için permütasyon matrisinin oluşturulması. ....	64
Şekil 4.9 :	Sınıflandırılmamış permütasyon hesabı ile atama süreleri. ....	65
Şekil 4.10 :	$\varphi = 8$ olduğu durumda k-NN ile sınıflandırılmış permütasyon hesabı ile elde edilen atama süreleri. ....	65
Şekil 4.11 :	Aynı yarıçap içine giren robotların tespit edilmesi. ....	69
Şekil 4.12 :	Güvenli alana giren robotlar için yerel rota oluşturulması ve alandan çıkıldığında tekrar eski rotaya girilmesi. ....	69
Şekil 4.13 :	Bulut tabanlı robot kontrol mimarisi. ....	70
Şekil 4.14 :	Etiket okuma modeli. ....	71
Şekil 5.1 :	Simülasyon yazılımında oluşturulmuş sanal bir ortam haritası görüntüsü. ....	74
Şekil 5.2 :	Sanal ortamın bulut sistemdeki harita izleme ekranı ile takip edilmesi. ....	75
Şekil 5.3 :	10-adet kumaşın, 9 adet terbiye makinesinin, 20 adet AGV'nin ve 30 adet dok arabasının bulunduğu örnek bir simülasyon haritası. ....	76
Şekil 5.4 :	K <sub>1</sub> kumaşını teslimat noktasına ulaştıracak 10 adet görev. ....	78
Şekil 5.5 :	Ürünlerin teslimat-zaman çizelgesi. ....	79
Şekil 5.6 :	İşlem sürelerinin karşılaştırılması. ....	81
Şekil 5.7 :	Optimal yol uzunluklarının karşılaştırılması. ....	81
Şekil 5.8 :	TurtleBot 2. ....	82
Şekil 5.9 :	Önerilen modelin laboratuvar ortamına uyarlanması. ....	83
Şekil 5.10 :	Test bazında hata dağılımı. ....	84

<b>Şekil 5.11</b> : Odometrik konum tahmini ile navigasyon.....	<b>84</b>
<b>Şekil 5.12</b> : Önerilen yöntem ile navigasyon.....	<b>85</b>
<b>Şekil 5.13</b> : Bulut yazılım ile yol planı oluşturma.....	<b>86</b>
<b>Şekil 5.14</b> : Bulut yazılımdan gelen yol planı emirin uygulanması.....	<b>86</b>



## SEMBOLLER VE KISALTMALAR

<b>CPS</b>	: Siber Fiziksel Sistem (Cyber Physical System)
<b>AGV</b>	: Otomatik Güdümlü Araç (Automated Guided Vehicle)
<b>IoT</b>	: Nesnelerin İnterneti (Internet of Things)
<b>API</b>	: Uygulama Programlama Arayüzü (Application Programming Interface)
<b>APF</b>	: Yapay Potansiyel Alan (Artificial Potential Field)
<b>ROS</b>	: Robot İşletim Sistemi (Robot Operating System)
<b>PSO</b>	: Parçacık Sürü Optimizasyonu (Particle Swarm Optimization)
<b>GA</b>	: Genetik Algoritma (Genetic Algorithm)
<b>k-NN</b>	: k-En Yakın Komşuluk (k-Nearest Neighbour)
<b>GPS</b>	: Küresel Konumlama Sistemi (Global Positioning System)
<b>ÜSİP</b>	: Üniversite-Sanayi İşbirliği Projesi
<b>ACO</b>	: Karınca Kolonisi Optimizasyonu (Ant Colony Optimization)
<b>AACO</b>	: Yaşlanan Karınca Kolonisi Optimizasyonu (Aging-based Ant Colony Optimization)
<b>memEAPF</b>	: Membran Evrimsel Yapay Potansiyel Alanı (Membrane Evolutionary Artificial Potential Field)
<b>ABC</b>	: Yapay Arı Kolonisi (Artificial Bee Colony)
<b>FPA</b>	: Çiçek Tozlaşma Algoritması (Flower Pollination Algorithm)
<b>SIFT</b>	: Ölçek-Değişimsiz Özellik Dönüşümü (Scale-Invariant Feature Transform)
<b>SLAM</b>	: Eş Zamanlı Konumlama ve Haritalama (Simultaneous Localization and Mapping)
<b>MLE</b>	: Maksimum Benzerlik Tahmini (Maximum Likelihood Estimation)
<b>YSA</b>	: Yapay Sinir Ağları
<b>SWSF-FP</b>	: Sıkı Zayıf Üstün Fonksiyon – Sabit Nokta (Strict Weak Superior Function – Fixed Point)
<b>LPA</b>	: Uzun Süreli A* Planlama (Life-long Planning A*)
<b>MRTA</b>	: Çoklu Robot Görev Dağılımı (Multi Robot Task Allocation)
<b>SOAP</b>	: Basit Nesne Erişim Protokolü (Simple Object Access Protocol)
<b>REST</b>	: Temsili Durum Transferi (Representational State Transfer)
<b>HTTP</b>	: Hiper Metin Transfer Protokolü (Hyper Text Transfer Protocol)
<b>SMTP</b>	: Basit Posta Transfer Protokolü (Simple Mail Transfer Protocol)
<b>XML</b>	: Genişletilebilir İşaretleme Dili (Extensible Markup Language)
<b>JSON</b>	: JavaScript Nesne Notasyonu (JavaScript Object Notation)
<b>RSSI</b>	: Alınan Sinyal Gücü Göstergesi (Received Signal Strength Indication)
<b>OAP</b>	: Optimal Atama Problemi (Optimal Assignment Problem)

# ÖZET

Doktora Tezi

## BİNA İÇİ LOJİSTİK FAALİYETLER İÇİN BULUT TABANLI OTONOM ROBOT YÖNETİMİ

FATİH OKUMUŞ

İnönü Üniversitesi  
Fen Bilimleri Enstitüsü  
Bilgisayar Mühendisliği Anabilim Dalı

105 + X sayfa

2020

Danışman: Prof. Dr. Adnan Fatih KOCAMAZ

Endüstriyel üretim sahalarında, süreçlerin otonom olarak yürütülmesine dayanan dördüncü sanayi devrimi için yakın zamanda “Endüstri 4.0” başlığı altında bir dijitalleşme süreci başladı. Bu süreç ile birlikte, iç içe geçmiş ve uyum içinde çalışan fiziksel ve dijital etmenleri kontrol altına alan ve izleyen siber fiziksel sistemlere olan ilgi kayda değer bir şekilde artmaktadır. Bina içinde sorunsuz ve sürekli malzeme-ürün akışı, üretim odaklı siber fiziksel sistemlere sahip işletmeler için hayati önem taşır. Bu durumda, lojistik faaliyetlerin esnek, güvenilir, hesaplanabilir ve izlenebilir olması için otonom lojistik sistemler ön plana çıkmaktadır. Bina içi otonom lojistikte, ürünleri taşımak için, sağladığı avantajlar ve kullanım kolaylığı sebebiyle, endüstriyel mobil lojistik robotu olan AGV’ler sıklıkla kullanılır. Bazı işletmelerde eş zamanlı çoklu taşımacılık gerektiği için çoklu AGV’lerin işbirliği içinde kullanımı ihtiyacı doğmaktadır. Çoklu mobil robotlarla işbirliği içinde otonom taşımacılık yapmak için şu temel problemlerin çözülmesi gerekmektedir: 1) Haritalama ve konumlama, 2) Optimum yol planlama, 3) Çoklu robot görev dağılımı, 4) Gerçek zamanlı navigasyon, 5) Çarpışmadan kaçınma. Bu temel problemler için üretilen çözüm yöntemlerindeki yaklaşımlar lojistik kalitesini doğrudan etkilemektedir.

Tez çalışması kapsamında, endüstriyel işletmelerdeki bina içi lojistik faaliyetlerin yürütülmesi amacıyla bulut tabanlı otonom çoklu robot yönetim sistemi geliştirilmiştir. Geliştirilen sistemde, haritalama, yol planlaması, çoklu robot işbirliği ve konum takibi işlemleri bulut sunucu tarafında yer almaktadır. Konumlama sistemi ve optimum yol planı üzerindeki gerçek zamanlı navigasyon ise, robot üzerinde koşturulan kontrol sistemiyle gerçekleştirilmiştir. Tez çalışması, Çalık Denim A.Ş. bünyesindeki kumaş terbiye işletmesinin bina içi lojistik faaliyetlerinde kullanılmak üzere, ilgili kuruluşun Ar-Ge Merkezi ve İnönü Üniversitesi BAP biriminin işbirliğinde gerçekleştirilen ÜSİP projesi ile desteklenmiştir. Geliştirilen bulut tabanlı otonom çoklu robot yönetim sistemi, benzer süreçlere sahip endüstrilere kolayca uygulanabilecek bir alt yapıya sahiptir. Yapılan bu çalışma, bina içi lojistik süreçlerini güçlendirmek ve tedarik zincirini otonom sistemlerle iyileştirmek isteyen işletmelerin yanı sıra konu üzerinde akademik çalışma yapan araştırmacılar için de aydınlatıcı ve yol gösterici niteliktedir.

**Anahtar Kelimeler:** Akıllı Üretim, Lojistik Planlama, Çoklu AGV İşbirliği, Bulut Robotik, Çoklu Görev Tahsisi, Optimum Yol Planlama, Konumlama.

## ABSTRACT

Phd. Thesis

### CLOUD-BASED AUTONOMOUS ROBOT MANAGEMENT FOR INDOOR LOGISTICS ACTIVITIES

Fatih Okumuş

Inonu University  
Graduate School of Nature and Applied Sciences  
Department of Computer Engineering

105 + X pages

2020

Supervisor: Doç. Dr. Adnan Fatih Kocamaz

For the fourth industrial revolution based on the autonomous execution of processes in industrial production areas, a digitalization process has recently started under the title of “Industry 4.0”. With this process, interest in cyber-physical systems that control and monitor intertwined and harmonious physical and digital factors is increasing significantly. Indoor material-product transportation, which is smooth and continuous, is vital for businesses with production-oriented cyber-physical systems. In this case, autonomous logistics systems come to the fore in order to make indoor logistics activities flexible, reliable, calculable and traceable. In indoor autonomous logistics, AGVs, which are industrial mobile logistics robots, are frequently used to transport products because of their advantages and ease of use. Since simultaneous multiple transportation is required in some enterprises, there is a need to use multiple AGVs in cooperation. In order to perform autonomous transportation in cooperation with multiple mobile robots, the following basic problems need to be solved: 1) Mapping and positioning, 2) Optimum path planning, 3) Multi-robot task allocation, 4) Real-time navigation, 5) Collision avoidance. The approaches in the solution methods produced for these basic problems directly affect the logistics quality.

Within the scope of the thesis, a cloud-based autonomous multi-robot management system has been developed for indoor logistics in industrial enterprises. In the developed system, mapping, path planning, multi-robot collaboration and position tracking processes are located on the cloud server-side. The positioning system and real-time navigation on the optimum path plan are carried out by the control system on the robot. The thesis study was supported by the ÜSİP project, carried out in cooperation with Çalık Denim A.Ş. and İnönü University BAP unit, to be used in indoor logistics activities of fabric finishing businesses. The developed cloud-based autonomous multi-robot management system has an infrastructure that can be easily applied to industries with similar processes. This study has enlightening and guiding characteristics for researchers who conduct academic studies on the subject, as well as enterprises that want to strengthen their indoor logistics processes and improve their supply chain with autonomous systems.

**Keywords:** Intelligent Manufacturing, Logistics Planning, Multiple AGV Collaboration, Cloud Robotics, Multi-Task Allocation, Optimum Path Planning, Positioning.

## 1. GİRİŞ

Dördüncü sanayi devrimi olarak nitelendirilen Endüstri 4.0 ile birlikte üretim süreçlerinde, gerçek zamanlı veri işleme, makine öğrenmesi, otomasyon ve birbirine bağlı uçlar, odak noktası haline gelmiştir. Bu bağlamda ortaya atılan siber-fiziksel sistemler (CPS) kavramı, endüstriyel işletme içi varlıkların yazılım sistemleri ile iç içe geçtiği, çoklu davranış modellerinin yönetildiği ve varlıkların birbiriyle iletişime geçebileceği bir sistemi ifade eder. Bu haliyle nesnelerin interneti (IoT) ile benzer mimariye sahip olsa da, CPS'lerde fiziksel ve yazılımsal unsurlar arasında yüksek bir koordinasyon ve kontrol aranır. Endüstride CPS'lere olan ilginin artmasıyla birlikte, akıllı üretimde önemli yapı taşlarından biri olan otonom lojistik sistemlerin kullanımı da kaçınılmaz hale gelmiştir (Llopis-Albert, Rubio, & Valero, 2019; Okumus & Kocamaz, 2019). Otonom lojistik, bina içinde ve dışında, mekânsal farklılıklardan dolayı farklı süreçlere sahiptir. İşletme içindeki kapalı alanlarda (fabrika, depo vs.) ürün ve malzeme akışını sağlayan bina içi lojistik sistemleri için mobil robotların kullanımında hatırı sayılır bir artış görülmektedir. Bina içi lojistikte iş gücünün insan odaklı yapılanmadan robotik odaklı yapılanmalara evrilmesi, üretimde kalite, hız, verimlilik, izlenebilirlik, ölçülebilirlik ve güvenlik gibi parametrelerde iyileştirmeler sağlamaktadır. Mobil robotlar, bu parametrelerin iyileştirilmesi ve sayısallaştırılmasını yapay zekâ, optimizasyon, sensor füzyonu ve görüntü işleme yöntemleriyle desteklemektedir.

Otomatik GÜdümlü Araçlar (AGV), akıllı lojistik uygulamalarında sağladığı avantajlar ve kullanım kolaylığı sebebiyle sıklıkla tercih edilen endüstriyel mobil robotlardır (Fellan, Schellenberger, Zimmermann, & Schotten, 2018). Yakın gelecekte çoklu AGV yönetim sistemlerinin birçok sektör için ilgi odağı haline geleceği öngörülmektedir. Çevre algılama, yol planlama ve davranış modelleme gibi özelliklere sahip olabilen AGV'lerin, raylı veya telli kılavuz araçlarına nazaran üretim ortamlarına adaptasyonu oldukça esnek ve düşük maliyetlidir (Mellado vd., 1999). İnsan gücüyle çalışılan bina içi lojistik süreçlerinde, taşımacılık görevleri herhangi bir operatöre atanır. Ancak bu operatörler görevleri icra ederken, yol, enerji veya zaman gibi önemli kaynakları optimize edemezler. İşletme ve üretim kapasitesi büyüdükçe, eş zamanlı olarak yürütülmesi gereken çok fazla sayıda lojistik görevi oluşacaktır. Bu nedenle, bu tip işletmelerde otonom lojistik için çoklu AGV'lere ihtiyaç vardır.

Çok sayıda AGV'nin aynı alanda eş zamanlı olarak kullanılmasında ortaya çıkacak karmaşıklığı engellemek için iki yöntem öngörülebilir: 1) AGV'lerin birbirleriyle doğrudan haberleştirilmesi 2) Haberleşme ve koordinasyonu sağlayacak merkezi bir yönetim mekanizması kurulması. Birinci yöntem çok karmaşık algoritmaların oluşmasına neden olduğu gibi gerçek ortamdaki iletişim, optimizasyon ve koordinasyon problemlerine de sebep olmaktadır. İkinci yöntemde ise daha basit iletişim mekanizması ile yönetim sağlanmakta ve robotların koordinasyonu merkezi birim tarafından sağlandığı için karmaşıklık oluşmamaktadır. Ayrıca ikinci yöntemde robotların donanımsal yükümlülükleri de daha az olduğu için maliyet avantajı bulunmaktadır. Bu tez çalışmasında robot koordinasyonu için ikinci yöntem tercih edilmiştir.

Birden fazla AGV'nin işbirliği içinde özerk ve eş zamanlı olarak çalışması için, başta optimize edilmiş çoklu görev tahsisi, yol planlaması, çarpışmadan kaçınma, haritalama ve konumlandırma gibi temel sorunların çözülmesi gerekmektedir.

Geleneksel (birinci yöntem) uygulamalarda işbirliği, AGV'lerin sahip olduğu donanım ve sensörler kullanılarak birbirleriyle iletişim kurma ilkesiyle sağlanır. Geleneksel çözümlere alternatif olarak, bulut bilişim (ikinci yöntem) mimarisi ile uygulanan işbirliği, çoklu AGV'lerin merkezi bir birim (bulut sunucusu) tarafından kontrol edilmesine olanak tanır. CPS'lerde bağımsız çalışma şartlarına sahip fiziksel etmenlerin ortak akıl ile yönetilmesi büyük önem arz etmektedir. Bulut bilişim ile CPS'ler için hem ortak akla dayalı bir yönetim sağlanmakta hem de tüm CPS'lerde uygulanabilecek esnek bir mekanizma oluşturulabilir. Bulut bilişim mimarisi ile güçlendirilmiş bir otonom lojistik sisteminde görevlerin oluşturulması, AGV'lerin optimum bir şekilde görevlere atanması, optimum yol planlaması ve konum takibi gibi işlemler bulut sunucu üzerinde gerçekleştirilebilir. Bu sayede bu işlemler, hesaplama kapasitesi yüksek sunucular tarafından gerçekleştirilerek, AGV'lerin donanımsal ve yazılımsal yükümlülükleri azaltılabilir. Bu durum aynı zamanda maliyeti de azaltacaktır. Bulut bilişimin getirdiği avantajlar çerçevesinde aynı sistem içindeki birçok işletme farklı fiziksel ortamlarda olsalar dahi ortak haberleşme imkânı kazanmış olacaktır.

Ortam haritalaması, robotların yönlendirilmesinde ve takip edilmesinde önemli bir rol oynar. Haritalama yapılmayan bir sistemde robotlar istenilen şekilde hareket edemezler. Robotik sistemlerde haritalama için üç farklı yaklaşım bulunmaktadır: 1) Bilinmeyen bir ortamda eş zamanlı haritalama 2) Ortamdaki nesnelerin ve konumlarının bilindiği ön tanımlı haritalama 3) Ön tanımlı haritalama ve eş zamanlı haritalamayı birleştiren hibrit yaklaşımlar. Eş zamanlı Konumlama ve Haritalama (SLAM) yöntemlerinde bir robot, gezinti halindeyken ortamı

haritalandırır ve kendini konumlandırır. Ön tanımlı harita yaklaşımlarında ise harita, robotları kontrol eden merkezi bir sistemde bulunur ve bu sistemden bağımsız çalışan bir konumlama sistemine ihtiyaç duyulur. Bu şekilde çalışan bir haritalama yönteminde konumlama sisteminin tespit edebileceği noktaların harita üzerinde bir karşılığının olması gerekir. Böylece elde edilen konum bilgisiyle robot, bilinen harita üzerinde konumlandırılabilir. SLAM uygulamalarında stereo kamera görüş sistemleri, lazer mesafe sensörleri ve ultrasonik sensörler kullanılır. Ön tanımlı haritalama yönteminde ise konumlama sisteminin kullandığı yöntem uygun sensörler kullanılır. Haritalama yapılırken alanda bulunan sabit varlıklar kapladıkları alan, geometrik şekilleri ve konumlarıyla birlikte bu haritaya ön tanımlı olarak yerleştirilir. Bu varlıklar hem çalışma modelinde etkin rol oynarlar hem de robotlar için bir engel olarak yol planlamasında kullanılırlar.

Mobil robotların otonom bir şekilde yönlendirilmesi için gelişmiş bir konumlama sistemine sahip olması gerekmektedir (Guo, Fang, Song, Xi, & Wei, 2018). Endüstriyel bina içi lojistiğinde kullanılan konumlama sistemleri, teker enkoderleri, hareket sensörleri veya kablosuz donanımlar kullanarak robotun bina içindeki konumunu tespit edebilmektedir. Kablosuz konumlama sistemleri, esnek ve kolay uygulanabilir bir navigasyona ihtiyacı olan endüstriyel işletmeler için ön plana çıkmaktadır. Kablosuz konumlama sistemlerinde GPS (Costley & Christensen, 2020; Lajoie, Ramtoula, Chang, Carlone, & Beltrame, 2020), RFID (W. Fu vd., 2020; Shamsfakhr, Palopoli, Fontanelli, Motroni, & Buffi, 2020; Haibing Wu, Tao, Gong, Yin, & Ding, 2019), Bluetooth (Y. Bae, 2019; Giuliano vd., 2020), Wi-Fi (Alves, Silva De Morais, & Yamanaka, 2020; Ismail & Terashima, 2018; Yen, Wang, & Chou, 2020), Zeg-Bee (Hsu, Yeh, & Hsu, 2016) ve Lidar (Al-Hourani & Ristic, 2020; Beokhaimook vd., 2020; M. Fu, Zhu, Yang, Song, & Wang, 2020) gibi özel donanımlar kullanılmaktadır. Bunun yanı sıra robotun teker enkoderlerinden gelen hareket verilerini kullanarak odometri tabanlı konumlama yapan sistemler de bulunmaktadır (Khattak, Papachristos, & Alexis, 2020; W. Liu vd., 2020). Tüm bu çalışmalar temelde varış açısı (AoA), varış zamanı (ToA), varış zamanı farkı (TDoA) veya RSSI teknikleri kullanılarak geliştirilmiştir (Wattananavin, Sengchuai, Jindapetch, & Booranawong, 2020). Bina içi konulamada, AoA ve (TDoA) tekniklerinin yönetilmesi oldukça karmaşık olması ve GPS kullanılmadığı için genellikle RSSI tekniği tercih edilmektedir. RSSI kablosuz erişim aygıtlarının alıcılarındaki sinyal okuma gücü göstergesidir ve sinyalin kalitesini ölçmek için kullanılan bir terimdir. Kablosuz erişim aygıtları RSSI ölçümü yapabildiği için donanımsal tercih alanı geniştir. Bu aygıtlar içerisinde RFID okuyucular ucuzluğu, hızı ve düşük güç

tüketimi ile ön plana çıkmaktadır. RFID teknolojisi diğer teknolojilerle karşılaştırıldığında, kirli havaya, metal olmayan engellere, duvarlara ve zemine nüfuz edebilmesi sayesinde kullanımı daha elverişli hale gelmektedir.

Kapalı alanlarda RSSI tekniği ile konum bilgisinin hesaplanması, sinyallerin düzensizliği ve ortamdaki maddelerin istenmeyen etkileşimlerinden dolayı karmaşık hale gelebilmektedir. Hareketli nesnelere ve insanların bulunduğu ortamlarda da RSSI varyasyonu belirgin bir şekilde oluşmaktadır (Booranawong, Sengchuai, & Jindapetch, 2019). Bu durumda tamamen RSSI üzerine kurulu olan konumlama sistemleri yüksek oranda başarısız olduğu görülmektedir. RSSI tabanlı konumlama sistemleri farklı yöntemler ve donanımlarla desteklenerek, tahmin yöntemlerinin ve uygulama modellerinin iyileştirilmesi gerekmektedir (Jondhale, Deshpande, Walke, & Jondhale, 2017). RSSI ile konum tespiti için genellikle Min-Max ve trilaterasyon (üçleme) gibi tahmin yöntemleri uygulanmaktadır. Yapılan çalışmalar Min-Max yönteminin doğru konum tespitinde diğer yöntemlere nazaran daha iyi sonuçlar verdiğini ve hareketli nesnelere bulunduğu ortamlarda da aynı başarıyı elde ettiğini göstermiştir (Wattananavin vd., 2020).

RFID bazlı konumlama sistemlerinde tahmin yöntemleri doğru bir konum tespiti için tek başına yeterli olmayabilir. Bu durumun oluşmasındaki önemli etkenlerden biri de RF sinyallerinin bozulma problemidir. Ortamda bulunan metal malzemeler, kablosuz erişim noktaları, RF sinyali yayan diğer cihazlar ve canlı varlıklar RFID sinyalini etkileyerek yanlış mesafe okumalarına sebep olabilmektedir (Lu, Xu, Zhong, & Wang, 2018; L. Ma vd., 2019). Sinyal bozulmasından kaynaklanan hatalı mesafe ölçümünden kurtulmak için yansımayı azaltacak şekilde bir okuma ve ortam modeli tasarlanmalıdır.

Çoklu AGV'lerin eş zamanlı olarak işbirliği yapmasının en temel zorluklarından biri de, çoklu görev tahsisi sorunudur. Optimize edilmiş bir çoklu görev tahsisi, basit tasarlanmış robotlarla karmaşık görevleri çözme, yüksek performans, güvenilirlik ve düşük maliyet gibi avantajlara sahiptir (Khamis, Hussein, & Elmogy, 2015). Çoklu görev tahsisiyle, hedef sayısı, görevlerin önceliği ve engellerin konumu gibi parametrelere bağlı olarak maliyet, zaman, enerji ve güvenlik gibi bileşenlerin iyileştirilmesi amaçlanmaktadır.

Birden çok bileşeni iyileştirmek için çok amaçlı optimizasyon yaklaşımları kullanılabilir. Bu çok amaçlı optimizasyonu, görevlerin ve diğer bileşenlerin birbiriyle ilişkili şekilde tasarlamak önemlidir (Korsah, Stentz, & Dias, 2013). AGV'lerin sayısı, konumu, şarj

durumu, taşınacak yükün ağırlığı, ziyaret edilecek istasyon sayısı ve bu istasyonların mesafeleri gibi birçok parametre görev tahsisinde değerlendirilmelidir.

Çoklu görev tahsisi ile ilgili çalışmalar iki başlık altında ele alınabilir: 1) Kesin çözümler (D. H. Lee, 2018; Tsarouchi vd., 2017; Wei, Hindriks, & Jonker, 2016) 2) Sezgisel yaklaşımlar. Kesin çözümlerle yapılan çalışmalar en iyi sonuçları verebilirler, ancak sezgisel metotlar kadar hızlı çalışamazlar (Nunes, Manner, Mitiche, & Gini, 2017). Sezgisel yaklaşımlı görev dağılım yöntemlerinde, parçacık sürüsü optimizasyonu (PSO) (Chen, Yang, & Wu, 2011; Choudhury & Biswal, 2011; Nedjah, De Mendonça, & De Macedo Mourelle, 2015; Wan vd., 2018; Xun & Hong-xu, 2008; Zhu, Tang, & Yuan, 2017), genetik algoritma (GA) (Jose & Pratihar, 2016; M. Mousavi, Yap, Musa, Tahriri, & Md Dawal, 2017; S. Mousavi, Afghah, Ashdown, & Turck, 2019; Suemitsu vd., 2016; Zitouni, Maamri, & Harous, 2019), k-WTA (Jin, Li, La, Zhang, & Hu, 2019), Macar yöntemi (Chopra, Notarstefano, Rice, & Egerstedt, 2017) gibi algoritmalar sıklıkla kullanılır.

Mobil robotlarda çoklu görev tahsis problemi, yol planlama problemiyle yakından ilgilidir. Yol planlama adımı, görev atama adımından sonra gerçekleştirilecekse, bu, optimum görev dağılımının faydasını etkileyecektir (Yao, Qi, Liu, Xu, & Du, 2019). Görevleri atarken ana amaç, tüm AGV'lerin kat ettiği yolların toplam uzunluğunu en aza indirebilmektir. Bu, bir kısıt olarak düşünüldüğünde, görevler verilirken yol planlamasının da yapılması gerekmektedir. Yol planlaması, bir başlangıç konumu ile bir veya daha fazla hedef arasında uygun bir yol oluşturmaya dayalı olarak hesaplanır. Yol planlama alanındaki çalışmalar arasında, hücre ayrışması, potansiyel alan ve yol haritasına dayalı geleneksel çalışmaların yanı sıra sezgisel yaklaşımlar da bulunmaktadır (J. Bae & Chung, 2018; Digani, Hsieh, Sabattini, & Secchi, 2018; Dönmez & Kocamaz, 2019; Mac, Copot, Tran, & De Keyser, 2016; J. Zhang, Zhou, & Zhang, 2018). Tüm sezgisel yaklaşımlar tam (completeness) ve optimal (optimality) algoritma olmadıkları için optimum çözüme ulaşmayı garanti etmezler (Okumuş & Kocamaz, 2019). Literatürde yol planlaması için A\*, D\*, Dijkstra, k-NN, GA, APF, PSO gibi algoritmalar kullanılmıştır. Bu çalışmada, yol planlaması için D\* Lite algoritması kullanılmıştır. D\* Lite, en uygun yolu ararken D\* stratejisini uygular, ancak basit bir yöntemdir ve daha yüksek verimlilikle çalışır (Silveira vd., 2012).

Tez çalışmasında, endüstriyel işletmelerde bina içi lojistik için bulut tabanlı bir otonom robot yönetim sistemi geliştirilmiştir. Sistemde, haritalama, yol planlaması, koordinasyon ve konum takibi işlemleri bulut sunucu tarafında yer almaktadır. Konum belirleme sistemi, optimum yol planı üzerindeki gerçek zamanlı navigasyon ve çarpışmadan kaçınma işlemleri

ise robot üzerinde gerçekleştirilmiştir. Haritalama için sistemin çalıştığı ortamın boyutlarının ve ortamdaki tüm sabit nesnelere konumlarının ve kapladıkları alanların tanımlanabildiği bir modül geliştirilmiştir. Harita verilerinin oluşturulması ve güncellenebilmesi için RESTful API'ler oluşturulmuştur. Optimum yol planlaması için literatürde yüksek oranda kullanılan algoritmalarından A\*, D\*, D\* Lite, Diskstra, APF, RRT\* algoritmaları sisteme seçilebilir şekilde entegre edilmiştir. Çoklu robotların çoklu görevlere optimum şekilde dağıtıldığı bir işbirliği algoritması geliştirilmiştir. Görev dağılımı yapılırken, AGV'lerin toplam yolunu, görevler yapılırken geçen zamanı ve harcanan enerjiyi optimize eden çoklu amaç fonksiyonu oluşturulmuştur. Optimum görev dağılımı için k-NN ve permütasyon olasılıklarını temel alan atama matrisi oluşturan ve matristeki her satırın toplam yol miktarlarına bakılarak en uygun görev atamalarını barındıran satırı seçen yeni bir algoritma geliştirilmiştir. Görevlerin, robotların ve ortamdaki diğer nesnelere takibi için harita tabanlı bir görüntüleme modülü geliştirilmiştir. Robot düzeyinde yapılan konumlama sistemi için RFID teknolojilerinden faydalanılarak özgün bir etiket okuma modeli oluşturulmuştur. Düz bir zemin üzerinde grid (ızgara) şeklinde ve belirli aralıklarla yerleştirilen RFID etiketlerden okunan sinyallerin RSSI değerleri kullanılarak Min-Max metodu ile konum tahmini yapılmıştır. ROS etkileşimli ve konumlama sisteminden gelen verilerle karar veren bir navigasyon yönetim modülü geliştirilerek robotun, oluşturulan yol planından sapmadan ilerlemesi sağlanmıştır. Tespit edilen konum bilgilerinin anlık olarak WebSocket aracılığıyla bulut sisteme bildirilmesi sağlanmıştır.

Tez kapsamında ortaya çıkan otonom çoklu robot yönetim sistemi, denim üretimi yapan Çalık Denim A.Ş. bünyesindeki Ar-Ge Merkezi ile işbirliği içinde geliştirilmiş ve ilgili kuruluşun bina içi lojistik faaliyetlerine uyarlanmıştır. Tekstil sektörünün ulusal ve uluslararası pazarda önde gelen kuruluşlarından biri olan Çalık Denim A.Ş.'nin, kumaş terbiye işletmesi tez çalışması için vaka örneği olarak ele alınmıştır. İlgili kuruluşun kumaş terbiye biriminde ürünlerin işlemden geçirileceği makineler arasındaki taşımacılığı hali hazırda insan kontrollü araçlarla gerçekleştirilmektedir. Tez kapsamında geliştirilen otonom çoklu mobil robot yönetim sistemiyle insan müdahalesi olmadan terbiye işletmesindeki tüm taşımacılık işlemlerinin otonom olarak yürütülmesi amaçlanmaktadır.

## **1.1 İlişkili Çalışmalar**

Bu bölümde yukarıda belirtilen temel problemlerin çözümü ile ilgili literatürde yapılan çalışmalar yer almaktadır. Çoklu mobil robotların otonom olarak işbirliği içerisinde hareket

ettirilmesi ve taşımacılık faaliyetlerinin yönetilmesi için bazı temel problemlerin çözülmesi gerekmektedir. Bu temel problemler şu şekilde sıralanabilir:

- 1- Ortamdaki tüm varlıkların, engellerin ve hareket edilebilir alanların yer aldığı ölçeklenebilir bir ortam haritası,
- 2- Robotun nerede olduğunu belirlenmesini ve hareket modelinin bu bilgiye göre oluşturulmasını sağlayacak konumlama sistemi,
- 3- Hareket edecek robotun başlangıç noktası ile hedef noktası arasında engelsiz bir yol oluşturulmasını sağlayacak yol planlama yöntemi,
- 4- Çoklu robotların işbirliği içinde çalışmasını sağlayarak optimum görev atanmasını sağlayacak çoklu görev dağılımı yöntemi,
- 5- Merkezi yönetim sağlayarak çoklu robotlar arasındaki koordinasyonu sağlayan bulut robotik sistemi.

### **1.1.1 Haritalama ve konumlama üzerine yapılan çalışmalar**

Lee vd. (T. J. Lee, Kim, & Cho, 2019) ileriye bakan bir monoküler görsel sensör kullanarak verimli eş zamanlı konumlama ve haritalama için yeni bir uygulama yöntemi sunmuşlardır. Yöntem, kapalı alan servis robotları için düşük maliyetli bir gömülü sistemde gerçek zamanlı olarak uygulanacak şekilde geliştirilmiştir. Yazarlar, önerdikleri yöntemin performansını, bir masaüstü bilgisayar kullanılarak veri kümesi tabanlı deneylerle ve düşük maliyetli bir gömülü sistem kullanılarak gerçek zamanlı deneylerle çeşitli zorlu ortamlarda göstermişlerdir.

Zhang vd. (Q. bin Zhang, Wang, & Chen, 2019) ön tanımlı ortam haritasında konumlama problemine odaklanarak, parçacık filtresi ve parçacık sürüsü optimizasyon algoritmasına dayalı bir hibrit konumlama yaklaşımı sunmuştur. Yazarlar, önerdikleri yaklaşımın performansını, kapalı alan robot lokalizasyonu için değerlendirmiş ve iki kıyaslama algoritması ile karşılaştırmıştır. Deneysel sonuçlar, önerilen yöntemin, kıyaslama yöntemlerinden daha az parçacık gerektiren iç ortamlarda sağlam ve doğru konumlandırma sonuçları elde ettiğini göstermiştir.

Gao vd. (M. Gao, Yu, Guo, & Xu, 2019) tek sensörlü mobil robotlarda düşük navigasyon doğruluğu ve hata birikimi sorunları için, görsel ve eylemsiz sensör kombinasyonuna dayalı bir bina içi mobil robot konumlandırma yöntemi sunmuşlardır. Yazarlar ilk olarak, renkli görüntü ve derinlik görüntüsünü elde etmek için görsel sensör (Kinect) kullanarak,

geliştirilmiş ölçek-değişimsiz özellik dönüşümü (SIFT) algoritması ile özellik eşleştirme gerçekleştirmişlerdir. Daha sonra, iki ardışık görüntü çerçevesinde bir robotun dönme matrisini ve çevirme vektörünü hesaplamak için mutlak oryantasyon algoritması kullanmışlardır.

Wu vd. (Hao Wu, Wu, & Tian, 2018) boylamsal düzleme yapıştırdıkları RFID etiketlerini okuyarak konum bilgisini hesaplamak için bulanık çıkarım algoritmasına dayalı olarak RFID anten açısının olasılık modelini belirlediler ve Bayesian tahmin kurallarını kullanarak robotun yönünü ve global konumunu elde ettiler.

Digiampaolo ve Martinelli (Digiampaolo & Martinelli, 2018) bir deponun raflarındaki nesnelere konumlamak için otonom bir robot kullanmışlar ve robotun konumunu bulmak için RFID sinyallerinden gelen bilgilerle robot odometrisini birleştiren Kalman tabanlı bir algoritma geliştirmişlerdir.

Ma vd. (L. Ma vd., 2019) RFID tabanlı konumlama sistemi olan WallSense'i tanıtmışlardır. Yazarlar, WallSense sistemiyle, Parçacık Sürü Optimizasyonunu (PSO) kullanarak komşu etiketlerden gelen sinyalleri farklılaştırmayı ve bu sayede istenmeyen faktörleri ortadan kaldırmayı hedeflemiştir. Önerdikleri yöntemle 0.24m'lik bir medyan hatasıyla konumlama yapmışlardır.

Zeng vd. (Zeng, Chen, Li, & Tan, 2019) RFID etiketlerin komşuluk ilişkilerini temel alarak faz kaymasını ölçmüş ve mesafe tahminindeki hata oranını düşürmüşlerdir. Yazarlar, hedef ve yardımcı etiketlerden oluşan çift etiket dizisinden gelen sinyal verilerinin, etiketler arasındaki mesafeyi ve okuma açılarına göre hata benzerliklerini ölçmüşlerdir. Bir hiperbolik pozisyon algoritması ve genetik algoritmayı birleştiren konumlama yöntemiyle 0.2217 m'lik bir hata oranı yakalamışlardır.

Shi vd. (Shi vd., 2019) IKULDAS ismini verdikleri geliştirilmiş bir k-NN yöntemi kullanan konumlama algoritması önermişlerdir. Yazarlar ayrıca, RSSI sinyallerinin yansılardan etkilenmemesi için anten eğimlerinin ve dönüş açılarını içeren bir okuma sistemi modellemişlerdir.

Ma vd. (Y. Ma, Tian, & Jiang, 2019) alanda bulunan etiketler arasındaki mesafeyi temel alan bir ağırlıklı çok boyutlu ölçeklendirme (WMDS) yöntemi önermişlerdir. Yazarlar 4 adet antenden alınan RSSI verileri ile etiketlerin konumunu kabaca elde ettikten sonra etiketten etikete iletişim yöntemini kullanarak tahmini konumu iyileştirmişlerdir.

Magnago vd. (Magnago vd., 2020) RSSI sinyal fazlarında yaşanan faz kaymalarının konum tespitini kötü etkilediğini belirterek kokusuz Kalman filtresi kullanılarak bu durumun iyileştirilebileceğini göstermişlerdir.

### 1.1.2 Yol planlama çalışmaları

Fu vd. (B. Fu vd., 2018) endüstriyel alanlarda kullanılmak üzere iyileştirilmiş bir A\* algoritması önermişlerdir. Yazarlar yaptıkları çalışmada, orjinal A\* algoritmasının komşuluk sınırlaması nedeniyle en kısa yolu aramada başarısız olma ihtimaline dikkat çekmişlerdir. Önerdikleri algoritma ile mevcut düğümün çevresinde bir sonraki aramadan önce planlanan mevcut düğüm ile hedef düğüm arasındaki yerel yol güvenli ve çarpışmasız ise doğrudan benimsenmesini sağlamışlardır.

Orozco-Rosas vd. (Orozco-Rosas, Montiel, & Sepúlveda, 2019) mobil robot yol planlama problemini çözmek için bir membran evrimsel yapay potansiyel alanı (memEAPF) yaklaşımı önermişlerdir. Yazarlar, uygulanabilir ve güvenli bir yol üreten parametreleri bulmak için genetik algoritma içeren bir membran hesaplama (tek seviyeli membran yapısına sahip zardan ilham alan evrimsel algoritma) ve yapay potansiyel alan yöntemlerini birleştirmişlerdir.

Rostami vd. (Rostami, Sangaiah, Wang, & Liu, 2019) mobil robotların sabit engellerden kaçınarak güvenli bir yol izlemeleri için iyileştirilmiş bir yapay potansiyel alan yöntemi önermişlerdir. İyileştirilmiş APF yöntemleri, en uygun yolun bulunmasında yerel minimumda sıkışma problemine odaklanarak bu problemi ortadan kaldırmaktadır. Ancak geleneksel APF yöntemine göre daha karmaşık ve yavaş çalışmaktadır.

Bayat vd. (Bayat, Najafinia, & Aliyari, 2018) görsel olarak bilinen ve dağınık engellerin olduğu bir ortamda yol planlama problemini ele almışlar ve bu problemin çözümü için elektrostatik potansiyel alan yaklaşımı önermişlerdir. Önerdikleri bu yöntem yüklü parçacıkların potansiyel alanları teorisinden yararlanarak ve yazarların son çalışmalarının temel bir fikrinden esinlenerek, optimum ve sağlam bir yol planlama çözümü elde etmek için optimizasyona dayalı bir yaklaşımdır.

Ajeil vd. (Ajeil, Ibraheem, Azar, & Humaidi, 2020) sürü zekâsı optimizasyonuna dayalı statik ve dinamik ortamlarda bir mobil robot için akıllı yol planlama algoritmalarının tasarımına odaklanmışlardır. Yaptıkları çalışmada karıncanın yaşına dayalı bir değişiklik uygulanan modifiye edilmiş yaşlanan karınca kolonisi optimizasyonu (AACO) adı verilen standart karınca kolonisi optimizasyonu kullanmışlardır.

Xu vd. (F. Xu vd., 2020) yapay arı kolonisi algoritmasının yavaş yakınsama hızı ve tek boyutlu arama stratejisinin, ayrılabilir işlevlerde avantaj göstermesini sınırladığını belirterek bu probleme odaklanmışlardır. Bu ilgili konuları ele almak için, ABC'ye bir birlikte evrim çerçevesi sunarak yakınsamasını hızlandırmak ve boyutun bağımlılığını ayrı ayrı ele geçirmek için geliştirilmiş bir stratejiye sahip küresel en iyi lider yapay arı kolonisi algoritması tasarlamışlardır. Önerdikleri bu yaklaşımı bir mobil robot yol planlama problemi üzerinde uygulayarak başarılı sonuçlar elde ettiklerini belirtmişlerdir.

Zhong vd. (Zhong, Tian, Hu, & Peng, 2020) büyük ölçekli dinamik ortamlarda mobil robot için küresel yol planlaması, gerçek zamanlı izleme ve engellerden kaçınmak için A\* algoritmasını uyarlanabilir pencere yaklaşımıyla birleştiren yeni bir hibrit yol planlama yöntemi sunmuşlardır. Yazarlar, yaptıkları simülasyon ve pratik deneylerde, küresel yol planlama, izleme ve engellerden kaçınma için önerdikleri hibrit yol planlama yönteminin, karmaşık dinamik ortamlarda mobil robotların uygulama ihtiyaçlarını karşılayabileceğini göstermişlerdir.

Low vd. (Low, Ong, & Cheah, 2019) kendi kendine öğrenme yeteneği nedeniyle son zamanlarda otonom mobil robot yol planlamasında artan popülerlik kazanan Q-öğrenme yönteminin optimum çözüme yavaş yakınsama problemini ele almışlardır. Q-öğrenmenin başlatılmasını iyileştirmek için çiçek tozlaşma algoritmasının (FPA) kullanarak bu problemin çözülmesini sağlamışlardır. Önerdikleri yöntemi üç tekerlekli bir mobil robot kullanarak gerçek dünya deneyinde doğrulamışlardır.

Dai vd. (Dai, Long, Zhang, & Gong, 2019) mobil robot için karmaşık haritalarda yol planlaması için geliştirilmiş bir karınca kolonisi algoritması önermişlerdir. Önerdikleri algoritma, A\* algoritmasının ve MAX-MIN Karınca sisteminin özelliklerini kullanmaktadır. Algoritmanın test edilmesi için simülasyon deneyi yapmışlar ve karşılaştırdıkları yöntemlere nazaran daha hızlı ve daha etkili olduğunu göstermişlerdir.

### **1.1.3 Çoklu robot görev dağılımı çalışmaları**

Schillinger vd. (Schillinger, Bürger, & Dimarogonas, 2018), bir robot grubu için otonom eylem davranışını optimize eden bir framework sunmuşlardır. Gerçekleştirdikleri sistemle, görevlerin önceden açık bir temsili veya tüm görev yürütme maliyetlerinin hesaplanmasını gerektirmeden, ayrılabilir görevleri mevcut robotlara en uygun şekilde tahsis ettiklerini iddia etmişlerdir.

Turner vd. (Turner, Meng, Schaefer, Whitbrook, & Soltoggio, 2018), yaptıkları çalışmada çoklu robot sisteminde görev dağılımını en üst düzeye çıkarma problemini ele almışlardır. Bu problemin çözümü için geliştirdikleri yeni yöntem, araçların mevcut zamanlama alanından yararlanarak ayrılmamış görevler için uygulanabilir bir alan oluşturmak üzere görevleri yeniden tahsis etmesine izin vermektir. Yazarlar, geliştirdikleri bu yöntemi bir simülasyon ile test etmişler ve toplam tahsis edilen görev sayısı olarak ölçülen performansta kayda değer bir artış gösterdiğini iddia etmişlerdir.

Du vd. (Du vd., 2019), dokuma atölyeleri için yaptıkları uygulamada iki rastgele nokta arasındaki en uygun yolu aramak için Warshall-Floyd'u kullanmışlardır. Genetik parçacık sürüsü optimizasyon algoritmasını (GA-PSO) kullanarak yol planlama probleminin matematiksel modelini oluşturarak zaman önceliğine dayalı parçacık iterasyon mekanizması ile algoritmanın yakınsama hızını hızlandırmışlardır.

Afrin vd. (Afrin, Jin, Rahman, Tian, & Kulkarni, 2019), endüstride sıklıkla kullanılmaya başlanan çoklu robotların yönetilmesi için bir bulut mimarisi önermişlerdir. Önerdikleri Edge Cloud tabanlı sistemle eş zamanlı üretim, enerji tüketimi ve maliyet optimizasyonu problemlerinin çözümü üzerine yoğunlaşmışlardır. Çoklu görev atama probleminde bulut üzerinde çalışan yeniden tasarladıkları NSGA-II algoritmasını kullanmışlardır.

Liu vd. (H. Liu, Liu, & Zheng, 2018), çalışmalarında bulut bilişim yeteneğinin çoklu robotların yönetimi üzerindeki etkisine dikkat çekmiştir. Takviye öğrenmesine (RL) dayalı bir görev atama algoritmasının geliştirdikleri bulut sisteme entegre olan çoklu robotların yönetilmesini sağlamışlardır. Önerdikleri sistemi Greedy Search algoritmasıyla karşılaştırarak faydasını ortaya koymuşlardır.

Chowdhury ve Maier (Chowdhury & Maier, 2017), robotun konumu, kalan enerji, görev gerçekleştirme süresi gibi parametrelere dayalı bir görev tahsis stratejisi önermişlerdir. Bu stratejiyi bir bulut sistemden ve komşu robottan alınan verilere dayalı bir karar mekanizması kurmuşlardır.

Yan vd. (H. Yan, Hua, Wang, Wei, & Imran, 2017), akıllı üretim ortamların temelini, bulut bilişim ve endüstriyel robotlar arasındaki entegrasyonların oluşturduğunu belirtmişlerdir. Bununla beraber bu ortamlarda yük paylaşımlarının esnek olmaması ve mevcut algoritmaların aşırı kaynak tüketmelerini birer problem olarak görmüşlerdir. Yaptıkları çalışmada grup öğrenmesine yönelik bir bulut mimarisi geliştirmişler ve elde ettikleri sonuçları paylaşmışlardır.

Wang ve Liu (L. Wang, Liu, & Meng, 2017), çoğu çoklu robotik uygulamalarda işbirliğinin merkezi bir operatör yerine robotlar arasında etkileşim yapılarak sağlandığını ve bulut robotik uygulamalarının merkezi kontrol odaklı alternatif bir yaklaşım sunduğunu belirtmişlerdir. Böyle bir merkezi uygulamanın kaynakları etkin bir şekilde kullanabilmesi için bir kaynak dağıtım stratejisi gerektiğini belirterek, bu problem için bağlantı kalitesi matrisi (LQM) açık artırması yöntemi önermişlerdir. Önerdikleri bu yöntemin son teknoloji algoritmalarından daha iyi performans sergilediğini iddia etmişlerdir.

Clark (Clark, 2005), limitli algılama ve robotlar arası iletişim ile çalışan, robotlar için çok robotlu bir koordinasyon platformu olan; Dinamik Robot Ağları temelli bir Olasılıklı Yol Haritası (PRM) hareket planlama algoritmasını sunmaktadır. Elde edilen sonuçlara göre DRN platformunun, sık ağ birleşmesi veya kopma meydana geldiğinde bile iyi çalıştığını gösterdiği ifade edilmektedir.

Solovey ve Halperin (Solovey & Halperin, 2013) robotların gruplara (renklere) ayrıldığı çok robotlu hareket planlama probleminin basit ve doğal bir uzantısını sunmaktadırlar, böylece her grupta robotların birbirleriyle değiştirilebilir olduğunu ifade etmektedirler. Geliştirilen algoritmada, k-renk probleminin çok sayıda ayrı çoklu robot hareket planlama problemine indirildiği yeni bir yaklaşım vardır. Geliştirdikleri yöntemi de k-renk çoklu-robot hareket planlama olarak isimlendirmişlerdir.

Ma vd. (X. Ma, Jiao, Wang, & Panagou, 2016) 3D poligonal engeller arasında hareket eden hava robotları için bir merkezi olmayan çok ajanlı hareket planlama yöntemi sunmaktadır. Algoritma, düşük seviyeli (yerel) koordinasyon ve kontrol için bariyer fonksiyonlarına dayalı bir yöntem ile birlikte yüksek seviyeli (global) planlama için öncelikli bir A\* algoritmasını birleştirmektedir. Geliştirilen algoritmanın, ajanların çarpışmasız hareketlerini sağlarken, uygun olduğunda ara noktaları yeniden tanımlamak yoluyla yolları hesapladığı ve poligonal bir engel ortamında tek başına gradyan bazlı bir çözümün performansını artıran yolları koruduğu iddia edilmektedir.

Draganjac vd. (Draganjac, Miklic, Kovacic, Vasiljevic, & Bogdan, 2016) çalışmalarında, endüstriyel ve depolama ortamlarında nakliye işleri yapan çoklu otomatik güdümlü araçların merkezi olmayan kontrolü için bir algoritma sunmaktadır. Çalışmada, önerilen kontrol stratejisinin sadece çoklu AGV sistemleri ile sınırlı olmadığı, aynı zamanda çeşitli diğer çoklu robot uygulamalarında mobil robotların merkezi olmayan koordinasyonu için de uygulanabilir olduğu söylenmektedir.

Mousavi vd. (M. Mousavi vd., 2017) çalışmalarında AGV'lerin görev planlamasını en aza indirme hedefiyle optimize etmek için bir matematiksel model geliştirdiler ve evrimsel algoritmalar (Genetik Algoritma – GA, Parçacık Sürüsü Optimizasyonu – PSO ve melez GA – PSO) ile entegre ettiler. Elde ettikleri sonuçlara göre, üç algoritmanın tamamlama zamanı ve AGV sayılarını düşürmedeki uygulanabilirliğinin kanıtlandığını öne sürmüşlerdir. En iyi sonucun ise melez sistemle elde edildiği vurgulanmaktadır.

Dewangan vd. (Dewangan, Shukla, & Godfrey, 2017) araştırmalarında, çoklu robot öncelikli yol planlama tekniklerinin seçimini belirleyen yöntemlerden bazılarını irdelemektedirler. Çoklu robot sistemlerinde var olan zorluklara değinilerek yol planlama yaklaşımının çeşitli konfigürasyonlara göre değişimlerini incelemişlerdir.

Digani vd. (Digani, Hsieh, Sabattini, & Secchi, 2019) önceden tanımlanmış geçici yol haritalarında seyahat eden bir AGV filosunu koordine etmek için bir optimizasyon stratejisi sunmaktadır. Çalışmada spesifik amacın, AGV'lerin diğer birbirleriyle çarpışmalarını önlemek için karmaşık trafik modellerini aşmak için harcadıkları zamanı en aza indirerek otomatik bir depoda trafik akışını en üst düzeye çıkarmak olduğu ifade edilmektedir.

Baek vd. (Baek, Hwang, Kim, & Kwon, 2018) Olasılıksal Yol Haritası (Probabilistic Roadmap – PRM) ve Takviyeli Öğrenme (Reinforcement Learning – RL) kullanarak cerrahi robotun otomasyonu için dinamik durumda çarpışmadan kaçınma yolu planlaması önermektedirler. PRM ve RL üzerinden çarpışma önleme yolunu bulmuş, koordinasyon sisteminin haritalama algoritmasını pikselden dünya koordinasyonuna kullanmış ve koordinasyon sistemini ters kinematik kullanarak kartezyen uzaydan eklem uzayına dönüştürmüşlerdir.

Oh vd. (Oh, Kim, Ahn, & Choi, 2016) parçacık sürüsü optimizasyonuna (PSO) dayanan bir optimal görev tahsisi algoritmasını, işbirliği zamanlaması görevleri için çoklu ajanlarla çalışacak biçimde bir sistem önermişlerdir. Çalışmalarında, konvansiyonel Parçacık Sürüsü Optimizasyonu (PSO) tabanlı algoritmayı, çizge teorisi altyapısı kullanarak değiştirmişlerdir. Önerilen algoritmanın performansını göstermek için sayısal simülasyonlar yürütmüşlerdir.

Guerrero vd. (Guerrero, Valero, & Oliver, 2017) yürütülecek en iyi görevi seçmenin (görev atama sorunu), çok robotlu sistemlerde ana sorunlardan biri olduğunu ifade etmektedirler. Çalışmalarında, bir görev tahsisi problemiyle ilişkili bulanık Markov zincirlerinin sınırlı

sayıda adımda durağan bir aşamaya gelebileceğini göstermektedirler. Bulanık Markov zincirlerinin hareket tahminini modelleme de daha iyi olduğunu söylemektedirler.

Trigul vd. (Trigui, Cheikhrouhou, Koubaa, Zarrad, & Youssef, 2018) afet yönetimi senaryosu bağlamında hedef bölgelere özerk robotlardan oluşan bir ekip atama problemini ele almaktadırlar. Bu problemin, birkaç robotun belirlenmiş yerleri ziyaret etmesi gereken birden fazla gezgin satıcı problemi (GSP) olarak ortaya çıkabileceği ifade edilmiştir. Araştırmanın, bu soruna analitik bir hiyerarşi süreci (AHP) temelli bir yaklaşım sağlayarak, toplam seyahat mesafesi, maksimum tur ve sapma oranı olmak üzere üç hedefi en aza indirdiği iddia edilmektedir.

Dönmez ve Kocamaz (DÖNMEZ & KOCAMAZ, 2019) çalışmalarında çoklu hedefler için çoklu robotlara görev paylaştırılması amacıyla algoritma geliştirmişlerdir. Geliştirilen algoritma var olan görevlerin sayısı, dağılım türü ve hedeflerin mesafelerine göre mevcut robot ajanlarına görevlerin dağılımını gerçekleştirmektedir. Çalışmada genetik algoritma ve en yakın komşu algoritmasını hedefler arası gezinti için kullanmışlardır.

Mevcut çoklu robot planlamacıları, tipik olarak, tüm robotların ortak durum alanındaki birleştirilmiş arama araştırmasına veya ayrıştırılmış planlamaya dayanmaktadır. Birleştirilmiş yaklaşımlar optimal çözüm bulabilir (Narayanan, Phillips, & Likhachev, n.d.; Velagapudi, Sycara, & Scerri, n.d.), ancak en kötü zaman karmaşıklığı ortamda yer alan çatışma potansiyeli olan robot sayısı ile katlanarak artar, bu durum ise birkaç robottan fazla robot olması halinde robotlar arası koordinasyon sağlanmasını pratik olmayan hale getirir.

#### **1.1.4 Bulut robotik çalışmaları**

Riazuelo vd. (L. Riazuelo, Civera, & Montiel, 2014) SLAM yeteneklerine sahip bir robotun en yeni bilgisayarlarla donatılması gerektiğine odaklanarak, pahalı harita optimizasyonu ve depolamanın Bulut'ta bir hizmet olarak tahsis edildiği, hafif kamera izleme istemcisinin yerel bir bilgisayarda çalıştığı, dağıtılmış bir çerçeveye dayalı görsel bir SLAM sistemi sunmuşlardır. Geliştirdikleri sistem, bir harita veritabanına arayüz sağlamaktadır ve depolanan haritalar tüm robotlar tarafından yeniden kullanılabilir. Ayrıca bir robot, haritasını çevrimiçi olarak hâlihazırda yer alan bir harita ile birleştirebilmektedir.

Wan vd. (Wan vd., 2016) yaptıkları çalışmada endüstri 4.0 kapsamında mobil hizmetleri ve bulut bilişim teknolojisini akıllı üretim ortamında kullanmayı amaçlayan bulut tabanlı bir yönetim sistemi sunmuşlardır. Sistem, bir üretim cihazı katmanı, bulut servis sistemi katmanı ve mobil servis katmanı olmak üzere üç katmandan oluşmaktadır. Üretim cihazı

katmanında endüstri içinde bulunan makine ve teçhizatlar yer almaktadır. Bu katmandaki aygıtların bulut servis sistemi katmanı ile iletişimi bir kablosuz modül ile sağlanmaktadır. Mobil servis katmanında ise sipariş işlemleri ve üretim bilgilerinin takip edilmesi sağlanmaktadır.

Riazuelo vd. (Luis Riazuelo vd., 2015) yaptıkları çalışmada, basit bir robotu akıllı bir robota dönüştürebilen web ve bulut hizmetleri sağlamak için bilgiye dayalı bir sistem olan RoboEarth projesinde mantıksal haritalamaya odaklanmışlardır. RoboEarth, aşağıdakileri sağlayarak haritalamayı güçlendirir: 1) hesaplamayı ve yanlış pozitif oranı azaltarak tanımayı geliştiren, anlamsal akıl yürütme ile elde edilen, eldeki görevle ilgili nesne modellerinin bir alt veritabanı; 2) robotlar arasında anlamsal haritaların paylaşılması; 3) robotun zorunlu gerçek zaman kısıtlamalarını karşılarken bulutta daha yoğun haritalama hesaplamalarını yapmak.

Wang vd. (Z. Wang vd., 2021) insansız yüzey araçları (USV) filosu arasında esnek uzaktan erişim ve koordineli görev kontrolü sağlamak için bulut tabanlı bir görev kontrol mimarisi önermişlerdir. Yaptıkları çalışmada robotlar ve bulut sunucusunu birbirine bağlayan Robot İşletim Sistemi (ROS) ile sorunsuz entegrasyon sağladıklarını belirtmişlerdir. Önerilen bulut tabanlı görev kontrol mimarisini gerçek araçlardan oluşan bir filoda uygulayarak deneysel performansını sunmuşlardır.

Ghazal vd. (Ghazal vd., 2020) yaptıkları çalışmada alüminyum fabrikalarında termal anormalliklerin tespiti için otonom robotlarda taşınan mobil IoT uç düğümlerine sahip bir kenar sis bulutu mimarisi önermişlerdir. Geliştirdikleri bulut sistem ile ilk müdahale dronlarının kontrolünü sağlayan bir yapı oluşturmuşlardır. Ayrıca kendi kendini yöneten bir derin öğrenme mimarisi ve termal anormallik algılama ve görselleştirme algoritması önermişlerdir. Yaptıkları sistemle düşük maliyetli robot filosu ile daha kısa yanıt süresi elde etmişler ve klasik uygulamalarda insan operatörlere kıyasla anormallikleri daha doğru bir şekilde tespit ettiklerini belirtmişlerdir.

Akbar vd. (Akbar, Lewis, & Wanner, 2020) mevcutta bulunan mobil-bulut hibrit (MCH) robotik uygulamaların, hem pil gücü tüketimini hem de ağ kullanımını en aza indirmek gibi eş zamanlı olarak çakışan hedefler arasındaki görev değişimlerini optimize etme açısından verimsiz olduğunu belirtmişlerdir. Bu sorunu çözmek için, yalnızca bir MCH robotik görevini yerine getirmek için değil, aynı zamanda hedefler arasında uzlaşmacı çözümü temsil eden verimli yapılandırmaları aramak için de kullanılacak yeni bir yaklaşım

önermişlerdir. Önerdikleri bu yaklaşımı bir simülasyon uygulamasıyla test etmişler ve sonuçlarını sunmuşlardır.

Varma vd. (Varma, Karjee, Rath, & Pal, 2020) çoklu robot uygulamalarında, uçtan uca iletişim ile koordinasyonun robotlar tarafından sağlanabilse de, bu işlemin robotlar için karmaşık bir görev olduğunu ve bazen düşük hesaplama ve iletişim güçleri nedeniyle pratikte mümkün olmadığını belirtmişlerdir. Bu sorunun üstesinden gelmek için, Bulut tabanlı Çok Atlamalı Çoklu Robot (DPS-CMM) modelini kullanan Dinamik Yol Seçimi adını verdikleri çok sekmeli bir iletişim modeli önermişlerdir. Bu modelde uçtan uca yapılan iletişim robotlar yerine bulut tarafından yönetilmektedir. Yazarlar yaptıkları modeli doğrulamak için, NS-3 ve Matlab tabanlı simülasyonların kombinasyonunu kullanmışlar ve uygulamayı mevcut son çözümlerle karşılaştırmışlardır.

Keung vd. (Keung, Lee, Ji, & Ng, 2020) bulut hizmetlerinin ve IoT iyileştirmesinin bir analizini sağlayarak, robotik mobil uygulama sisteminin (RMFS) daha iyi sipariş karşılama sunmasının operasyonel verimlilik ve sistem güvenilirliği açısından nasıl avantajlar sağlayabileceğine ilişkin araştırma kuyruklarını çözmek için literatürlerdeki teorik kavramları birleştirmişlerdir. Ayrıca bulut tabanlı bir CPS mimarisi önermişler ve çok katmanlı çok derinlikli depo düzeninde çarpışmadan kaçınma stratejisi hakkında kapsamlı bir anlayış geliştirmişlerdir. Geliştirdikleri sistemi bir vaka üzerinde uygulayarak başarılı olmuşlardır.

### **1.1.5 Tez çalışmasının benzer çalışmalarla karşılaştırılması**

Tez çalışmasında, yukarıda verilen ilişkili çalışmalardaki yaklaşımlar dikkate alınarak çoklu robot işbirliği için bulut tabanlı bir yönetim sistemi ve robot kontrol sistemi geliştirilmiştir. Geliştirilen bu sistem oldukça kapsamlı yöntemler içermekte ve konu bir bütün olarak ele alınmaktadır. Sistem her biri ayrı çalışma alanı olan haritalama, konumlama, yol planı, çoklu robot görev dağılımı, robot navigasyonu ve çarpışmadan kaçınma gibi karmaşık modüler yapıları bir arada barındırmaktadır. Sistemdeki modüler özellikler oldukça fazla olmakla birlikte bu modüler özellikler, literatürde öne çıkan benzer çalışmalarla karşılaştırılmış ve karşılaştırma sonuçları Çizelge 1.1’de verilmiştir.

**Çizelge 1.1** : Tez kapsamında geliştirilen otonom çoklu robot yönetim sisteminin öne çıkan ilişkili çalışmalarla karşılaştırılması

	(Schillinger vd., 2018)	(Turner vd., 2018)	(Du vd., 2019)	(Afrin vd., 2019)	(Chowdhury & Maier, 2017)	Tez Çalışması
Ortam Varlıklarının Belirlenmesi	✓	✓	✓	✓	✓	✓
Robot Şarj Ünitesi	✓	-	-	-	-	✓
Haritalama	✓	-	-	-	✓	✓
Çoklu Robot	✓	✓	-	✓	-	✓
Çoklu Görev Dağılımı	✓	✓	✓	✓	✓	✓
Eş zamanlı Görevler	✓	✓	-	✓	✓	✓
Merkezi Koordinasyon	✓	-	-	✓	✓	✓
İzleme	-	-	-	-	-	✓
Simülasyon	✓	✓	✓	✓	-	✓
Laboratuvar Deneyi	-	-	-	-	-	✓
Yol Planlama	✓	-	✓	-	-	✓
Çoklu Robot İşbirliği	✓	-	-	✓	-	✓
Batarya Optimizasyonu	✓	✓	-	-	-	✓
Çarpışmadan Kaçınma	✓	-	-	-	-	✓
Sözde Kod	✓	✓	-	✓	✓	✓
Maliyet Optimizasyonu	✓	✓	✓	✓	✓	✓
Bulut Tabanlı Yaklaşım	-	-	-	✓	✓	✓
Konumlama Sistemi	-	-	-	-	-	✓
Robot Hareket Modelleme	-	-	-	-	-	✓

## 1.2 Tezin Amacı

Üniversite-Sanayi İşbirliği Projesi (ÜSİP) kapsamında birlikte çalışılan ve tez konusuna ilham kaynağı olan Çalık Denim Ar-Ge Merkezinin kumaş terbiye işletmesinde hâlihazırda yürütülen bina içi lojistik faaliyetler insan kontrollü taşıma araçları kullanılarak icra edilmektedir.

Bu tezin amacı; Çalık Denim Ar-Ge Merkezinin kumaş terbiye işletmesi örnek alınarak, endüstriyel üretimdeki bina içi ürün/malzeme lojistiği için çoklu taşıyıcı robotların (AGV) otonom olarak yönetilmesini sağlamaktır. Bu bağlamda AGV'lerin hassas bir şekilde konumlandırılmasını sağlayan, eş zamanlı çoklu görevler oluşturan, bu görevleri optimum bir şekilde, öncelikleri hesaba katarak, optimum yol planları oluşturarak, enerji yönetimi ve çalışma sahası değişkenlerine göre AGV'lere atayan, AGV'lerin ilerleme, dönüş ve statik/dinamik engellerden kaçınma davranışlarını eş zamanlı olarak modelleyen bulut tabanlı çoklu robot kontrol ve yönetim sisteminin geliştirilmesi hedeflenmiştir.

## 1.3 Tez Çalışmasının Gerekçeleri

Endüstriyel kapalı ortamlarda gerçekleştirilen lojistik faaliyetlerin yeni teknolojilerle desteklenmesi son birkaç yılda yükselen bir ivmeyle artmaktadır. Küresel otonom mobil robot pazarı 2019'da 29,3 milyar dolar gelir elde etti ve bu gelirin 2030'da 220,6 milyar dolara ulaşması beklenmektedir. Ayrıca, pazarın 2020-2030 yılları arasında %18,3'lük bir yıllık bileşik büyüme oranı ile hızlı ilerlemeye tanık olacağı tahmin edilmektedir (*Autonomous Mobile Robots Market Research Report: By Offering, End User - Global Industry Size, Share and Trends Analysis, Forecast to 2030, 2020*). Bu alanda yapılacak çalışmalar geleceği inşa etme konusunda önemli yapı taşlarına sahip olacaktır. Bu bağlamda, tez kapsamında geliştirilen bulut tabanlı otonom endüstriyel lojistik robot yönetim sisteminin geliştirilmesindeki temel gerekçeler şunlardır:

- Otonom mobil robot pazarında ulusal ve uluslararası kalkınmanın sağlanması ve Ar-Ge birikiminin oluşturulması,
- Bina içi lojistik süreçlerinden kaynaklı işçi sağlığı ve iş güvenliği risklerinin azaltılması,
- Lojistik amaçlı personelin başka işlere kaydırılması ile işçi maliyetlerinin düşürülmesi,

- Taşımacılıkta optimal değerleri esas alan algoritmalar ile taşıma maliyetlerinin düşürülmesi, dolayısıyla üretim süresinin kısaltılması
- Endüstriyel üretim standartlarının artırılması ile insan faktöründen kaynaklanan tedarik risklerinin önüne geçilmesi,
- İşletme içi lojistik faaliyetlerin ölçülebilir olmasıyla maliyet analizlerinin daha gerçekçi yapılması,
- Bina içi lojistik süreçlerinin izlenebilir hale gelmesiyle birlikte hızlı ve doğru karar alabilme imkânının sağlanması,
- Çoklu görevlerin çoklu robotlara atanması için özgün bir görev dağılım algoritmasının oluşturulması,
- Mobil robotlarda önemli bir konu olan konumlama problemine özgün bir bakış açısı getirerek hassas konumlama yapılmasının sağlanması.

#### **1.4 Tezin Yenilikçi Yönü ve Ar-Ge Niteliği**

Tez kapsamında geliştirilen sistem, literatürde ve pazarda yer alan muadil uygulamalarla kıyaslandığında aşağıdaki yenilikçi özellikleri öne çıkmaktadır:

- Tez kapsamında geliştirilen çoklu robotlara çoklu görev atama algoritması yüksek bir başarımlık düzeyine sahiptir ve optimum verimlilik sağlamaktadır. Yapılan simülasyon ve laboratuvar deneyleriyle bu başarımlık sunulmuştur.
- Konumlama sisteminde önerilen RFID tabanlı okuma modeli konumlama hassasiyetini oldukça yükseltmektedir. Bu sayede hata oranı oldukça düşük olan bir mobil robot navigasyon sistemi gerçekleştirilmiştir. Yapılan simülasyon ve laboratuvar deneylerinden elde edilen sonuçlar geleneksel yöntemlerle kıyaslandığında önerilen okuma modelinin oldukça etkili olduğu görülmektedir.
- Geliştirilen sistem bulut tabanlı bir yönetim sistemine sahiptir. Bulut tabanlı olmasının getirdiği avantajla, geliştirilen sistem eş zamanlı olarak birden fazla ortamda uygulanabilmektedir.
- Bulut tabanlı olmasının getirdiği avantajlarla mobil robot üzerindeki donanımsal ve yazılımsal yük hafifletilmiştir. Böylece mobil robotların üretim maliyetlerine doğrudan etki etmektedir.

## 1.5 Tezin Çıktıları

Tez çalışması kapsamında, endüstriyel üretim sahalarındaki bina içi lojistik faaliyetlerin mobil robotlar kullanılarak yapılmasını hedefleyen, bulut tabanlı otonom çoklu robot yönetim sistemi geliştirilmiştir. Sistem bulut sistem koordinasyon yazılımı ve robot kontrol sistemi olmak üzere iki aşamalı olarak ele alınmıştır.

Tez kapsamında geliştirilen sistemin çıktılarını ölçebilmek için simülasyon ve gerçek ortam deneyleri uygulanmıştır. Simülasyon testi için tüm ortam nesnelere, özellikleri ile birlikte sanal olarak simüle edebilen bir yazılım geliştirilmiştir. Simülasyon yazılımı üzerinden harita verilerinin oluşturulması ve güncellenmesi, robotların (kumaş taşıyıcı yeni AGV robotlarının) dinamik olarak ortama dâhil edilmesi, işletme içi lojistik ürün ve süreçlerin tanıtımı (kumaş), taşıyıcı yük aygıtlarının (dok arabaları) tanıtımı, yerleşik istasyonların (terbiye makineleri) tanıtımı, konum bildirim (AGV'ler üzerine yerleştirilen RFID konumlama sistemi), navigasyon gibi adımlar başarıyla uygulanarak bulut sisteme aktarılması sağlanmıştır. Simülasyon yazılımında oluşturulan ortam şartlarında, bulut sistemin tüm görevleri optimum kazançla ve yüksek başarıyla uyguladığı gözlenmiştir.

Sistem son olarak, laboratuvar ortamında ve deneysel amaçlı mobil robot üzerinde test edilmiştir. Laboratuvar zeminine, önerilen okuma modelindeki düzene göre RFID etiketler yerleştirilmiştir. Deneysel amaçlı mobil robot olarak Turtlebot 2 kullanılmış, robot üzerine Raspberry Pi bağlanarak ROS (Robot İşletim Sistemi) etkileşimli çalışması sağlanmıştır. Robotun bulut sisteme tanıtılarak verilen emirleri başarıyla yerine getirdiği gözlenmiştir. Gerçekleştirilen deneylerde, işbirliği algoritmasının atanan tüm görevleri optimum düzeyde gerçekleştirdiği görülmüştür. Yapılan laboratuvar deneylerinde konumlama sisteminin, gerçek konum bilgisine göre ortalama  $\pm 12,26$  cm'lik hassasiyetle çalıştığı gözlenmiştir. Konum tahmininde en yüksek hata miktarı 27,56cm olarak ölçülürken en düşük hata miktarı 0,5cm olarak ölçülmüştür. Yol planından sapma miktarında ise odometri (enkoder tabanlı konumlama) kinematiğine göre ortalama %90,46 daha iyi bir sonuç elde edilmiştir.

## 1.6 Tezin Organizasyonu

Tez çalışması altı bölümden oluşmaktadır. Birinci bölümde tez içerisindeki temel konularla ilgili literatürde yer alan ilişkili çalışmalara yer verilmiştir. Ayrıca tezin amacı, gerekçeleri, yenilikçi yönü çıktıları da bu bölümde ifade edilmiştir.

İkinci bölümde endüstriyel işletmelerde bina içi lojistik kavramının detaylıca irdelenmesi sağlanmıştır. Bina içi lojistik kavramıyla birlikte taşımacılıktaki etken unsurlar ve bu unsurların lojistik yönetimine etkisi vurgulanmıştır. Etken unsurların müdahil olduğu lojistik süreçleri akış diyagramlarıyla verilmiştir. Ayrıca bir vaka örneği olarak tekstil terbiye işletmelerinde bina içi lojistik faaliyetler anlatılmıştır. Son olarak otonom lojistiğin önemi vurgulanmıştır.

Üçüncü bölümde otonom robot yönetiminde var olan temel problemler ve bu problemlerin çözümü için kullanılan yöntemler açıklanmıştır. Literatürde yer alan konumlama sistemleri ve konum tahmin yöntemleri, yol planlama algoritmaları, çoklu robot görev dağılım yöntemleri ve robot kontrolü ile ilgili kullanılan yeni nesil çözümler bu bölümde detaylıca açıklanmıştır.

Dördüncü bölümde tez kapsamında geliştirilen sistemin anlatıldığı materyal ve metot bölümü yer almaktadır. Geliştirilen sistemde yer alan materyaller, bulut sistem mimarisi ve içeriği, robot kontrol sistemi, önerilen konumlama sistemi ve önerilen çoklu robot görev dağılım yöntemi bu bölümde detaylıca açıklanmıştır.

Beşinci bölümde tez kapsamında geliştirilen bina içi endüstriyel çoklu robot yönetim sisteminin simülasyon ve gerçek ortamda elde edilen deneysel sonuçlarına yer verilmiştir. Simülasyon bulut sistem ve konumlama sistemi olmak üzere iki aşamalı olarak geliştirilmiş ve uygulanmıştır. Laboratuvar deneylerinin açıklandığı bölümde ise konumlama sisteminin test edilmesi ve bulut tabanlı navigasyon işlemlerinin gerçekleştirilmesi ve elde edilen veriler sunulmuştur.

Altıncı bölümde tez çalışması sonrası sonuç ve öneriler verilmiştir. Geliştirilen sistemin genel değerlendirilmesi, çalışmanın uygulama alanları ve gelecekte planlanan çalışmalara dair bilgiler bu bölümde yer almaktadır. Son olarak altıncı bölümü müteakiben kaynaklar verilmiştir.

## 2. ENDÜSTRİYEL İŞLETMELERDE BİNA İÇİ LOJİSTİK

Lojistik; ticari, askeri, medikal ve daha birçok alanda malzeme ve teçhizat tedarikinde kilit rol oynayan bir disiplindir. Ticari anlamda lojistik, müşterilerin veya kurumların ihtiyaçlarını karşılamak için menşe noktası ile tüketim noktası arasındaki fiziksel öge ve bilgi akışının yönetilmesidir (Y. Yan & Qiang, 2008). Lojistikte yönetilen kaynaklar; malzeme, ekipman, yarı mamul ve sarf malzemelerinin yanı sıra gıda ve diğer tüketim maddelerini içerebilir. Fiziksel öğelerin lojistiği, genellikle bilgi akışı, malzeme taşıma, üretim, paketleme, envanter, nakliye, depolama ve güvenliğin entegrasyonunu içerir. Menşe ve tüketim noktası arasındaki ürün/malzeme lojistiği sürecinde birçok unsur rol oynamaktadır. Bu unsurların her birinin birbirini tamamlaması tam anlamıyla tedarik zincirini ifade eder. Zincirdeki her bir halka, yani her bir unsur, güvenilir ve sağlam olmak zorundadır. Bu sayede kusursuz bir tedarik yönetimi sağlanabilir.

Ticari lojistikte tedarik zincirinin en önemli halkalarından biri de ham ve yarı mamul ürünlerin işletme içindeki (bina içi) lojistiğidir. İşletme içindeki ürün/malzeme taşımacılığını ifade eden bina içi lojistik, bu ürünlerin/malzemelerin belirli işlemlerden geçirilmesi ve teslimat noktasına iletilmesi için kendi içinde bir akış sürecine sahiptir.

Lojistik faaliyetlerin bina içinde ve bina dışında ayrı ayrı değerlendirilmesinde birçok faktör vardır. Teknolojik olarak bakıldığında, özellikle otonom araçların konumlarının belirlenmesi bina içinde ve dışında farklı donanımların kullanılmasını gerektirir. Örnek olarak bina dışında otonom araçların konumlarının belirlenmesi için GPS kullanılırken, aynı sistem bina içinde kullanıma elverişsizdir. Bina içinde konumlamada Wi-Fi, RFID ve Bluetooth gibi kablosuz donanımlar veya raylı ve telli bantlar kullanılırken, bu uygulamaların dış dünyada (işletme içi küçük çaplı dış alanlar hariç) yapılması elverişsizdir. Ayrıca bina içi lojistik daha çok işletme içi tedarik süreçlerini ilgilendirirken dış ortam lojistiği bağımsız unsurlar tarafından yönetilebilmektedir. Bina içindeki lojistik, temel olarak ürünlerin hazır hale getirilmesi ve dış dünyaya açılan kapıya kadar gerçekleşen süreçleri yönetir.

Bu bölümde, bina içi lojistiğin tam olarak anlaşılabilmesi için lojistik yönetiminde etken unsurların belirlenmesi ve bu unsurların hangi süreçlerle ve akışla yönetildiği konularına açıklık getirilmiştir. Ayrıca teze konu olan bir vaka örneği verilerek pratikte uygulanan sistemler somutlaştırılmıştır. Son olarak bina içi lojistiğin neden otonom hale getirilmesi gerektiği ve önemi hakkında bilgiler verilmiştir.

## 2.1 Bina İçi Lojistikte Etken Unsurlar

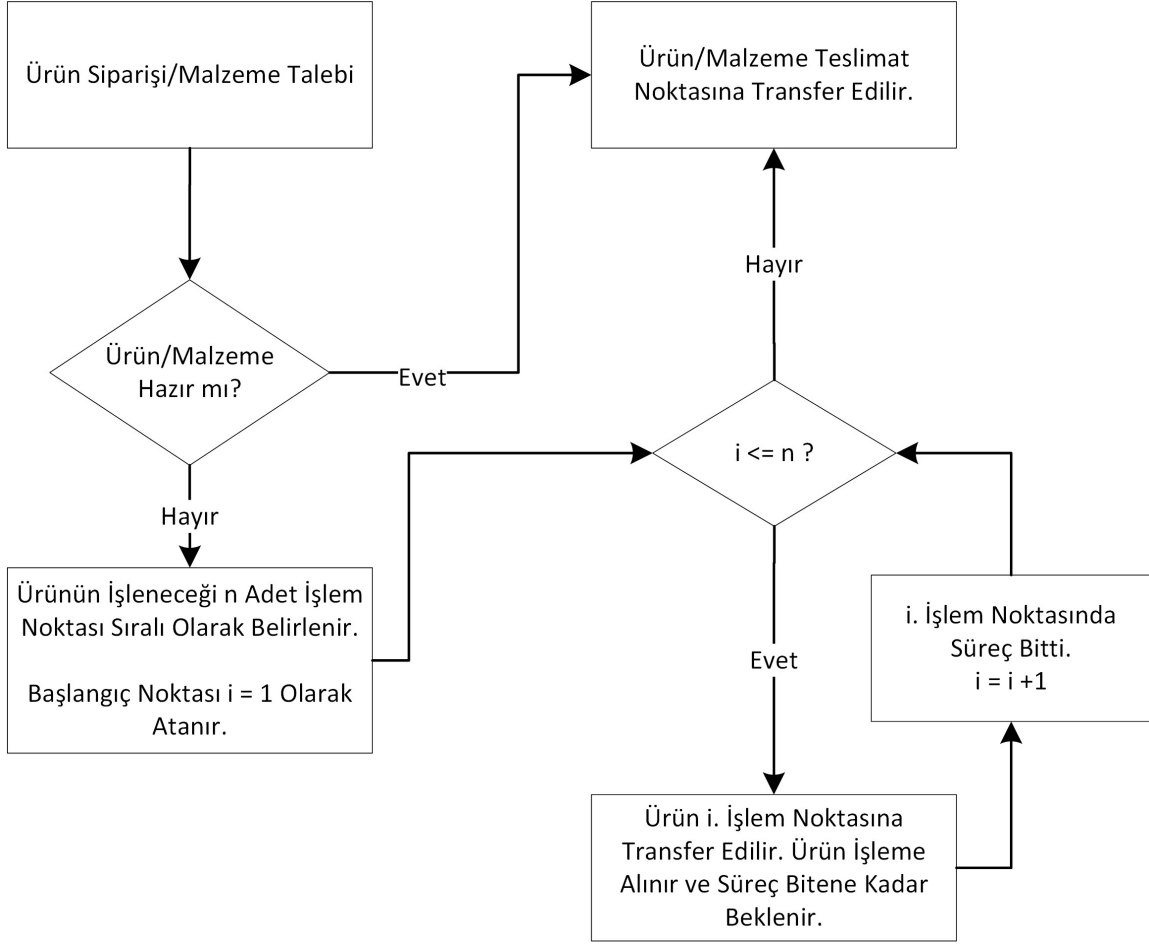
Lojistik farklı materyallere sahip işletmelerde farklı şekillere bürünse de temelde aynı unsurlar kullanılır. Bu unsurların net ve eksiksiz bir şekilde, sistem içindeki rolleri ile birlikte belirlenmesi otonom lojistik için büyük önem arz etmektedir. Bu unsurlar şu şekilde sıralanabilir:

- Taşınan Ürün/Malzeme,
- Ürünün lojistik sisteme girişini belirleyen başlangıç noktaları,
- Ham veya yarı mamul ürünlerin belirli işlemlerden geçirildiği istasyonlar veya makineler,
- Ürünleri taşıyan veya çeken araçlar,
- Çekme tipi taşımacılıkta kullanılan ürünlerin yüklendiği römork tarzı parti arabaları,
- Taşıyıcı araçların hareket alanı,
- Taşıyıcı araçları kullanan operatörler,
- Transfer işlemlerinin ve süreçlerin kontrolünü sağlayan karar alıcı mekanizmalar.
- Elektrikli taşıyıcı araçlar için şarj üniteleri,
- Yarı mamul ve ham malzemeler için bekleme istasyonları,
- Teslimat noktaları

## 2.2 Bina İçi Lojistik Süreçleri

Endüstriyel işletmecilikte bina içi lojistiğe etki eden unsurlar belirli bir akış içinde kullanılırlar. Bu akış aynı zamanda ürünün müşteriye teslim edilebilecek düzeyde hazır hale gelmesi amacını da kapsar. Bazı işletmeler hazır haldeki ürünleri depolardan teslimat noktalarına ulaştırmak için taşımacılık yaparken, bazılarında ham veya yarı mamul ürünler belirli işlemlerden geçirilerek hazır hale getirilir ve ardından teslimat noktasına ulaştırılır. Bazı durumlarda ise sadece işletme içi malzeme taşımacılığı yapılır ve bir ürün teslimatı amacı güdülmez. Ancak tüm bu taşımacılık faaliyetleri belirli bir akış içerisinde yürütülmelidir. Bir ürün veya malzeme talebi geldiğinde, eğer bu ürün/malzeme hazır durumdaysa ve herhangi bir işlemde geçmeden talep edilen noktaya götürülecekse bu işlem için bir transfer aracı atanır ve taşıma işlemi gerçekleştirilir. Eğer bu ürün/malzeme birtakım

işlemlerden (process) geçirilerek hazır hale getirilecekse, bu durumda öncelikle yapılacak bu işlemlerin bir listesi ve sıralaması belirlenir. Taşınan ürün/malzeme sırasıyla bu işlemlerin gerçekleştirileceği noktalara transfer edilir. Yapılan her işlem belirli bir süre içinde tamamlandığı için bu noktalarda ürünün taşınması beklenir. İşlemin bitip bitmediğine uzman operatörler tarafından karar verilir. Son işlem süreci de tamamlandığında ürün/malzeme hazır hale gelir ve teslimat noktasına götürülür. Genel olarak bina içinde, ürün/malzeme lojistiği süreçlerinin akış çizelgesi Şekil 2.1’de gösterilmiştir.



Şekil 2.1 : Bina içi genel ürün/malzeme akış çizelgesi.

### 2.3 Otonom Bina İçi Lojistiğin Önemi

Günümüz dünyasında endüstriyel işletmelerdeki bina içi ürün taşımacılığı daha çok insan kontrolüne ve uzmanlığına dayanıyor. Ancak bununla beraber, akıllı üretim hedeflerinin ön plana çıkmasıyla beraber ürün ve malzeme akışında otonom taşımacılığın kullanımı büyük bir hızla artmaktadır. Elbette ki, bu artış işletmelere birçok açıdan fayda sağlamaktadır. Otonom ve insansız taşımacılığın verimlilik, performans, maliyet ve sağlık alanlarında ciddi etkileri bulunmaktadır. Bu etkileri şu şekilde sıralayabiliriz:

- 1- İnsan koordinesine dayalı sistemlere nazaran daha az hatalı üretim gerçekleşir. Vaka örneği göz önüne alındığında, lojistik operatörlerinin kumaşları hatalı makineye bırakması işletmeye ciddi zarar verme potansiyeli taşır. Otonom sistemlerde ürün ve malzeme talebi net olarak bilinmekte ve tüm işlemler kontrol edilmektedir. Böylece hatalı üretim oranı ciddi oranda düşürülür.
- 2- Otonom sistemlerde daha az personele ihtiyaç duyulması nedeniyle sürdürme maliyeti daha düşüktür. Ürünlerin ve malzemelerin iş akışı süreçlerinin tanımlanması, bu süreçlerin koordinasyonu, araçların noktalar arasındaki transferi için sürüş operatörlüğü, ürünlerin/malzemelerin yüklenmesi veya bırakılması vb. birçok yerde insan desteğine olan ihtiyacı ortadan kaldırmaktadır.
- 3- Otonom olmayan lojistikte, nakil sırasında gerçekleşen kazalar çalışan sağlığını tehdit etmektedir. Bu kazalar, yaralanmalar ve az sayıda da olsa ölümlerle sonuçlanabilmektedir. İnsansız otonom taşımacılık sisteminde bu tür kazalar yaşansa dahi herhangi bir yaralanma söz konusu olmayacaktır.
- 4- Lojistik araçlarının (robotlar) tam bir uyum içerisinde çalışması ile daha az zamanda daha çok üretim yapılabilmektedir. Özellikle otonom sistemiyle, toplam üretim zamanını minimize edecek şekilde optimizasyon sağlanması bu etkiyi daha da artırmaktadır. İnsan koordinasyonu ile yapılan taşımacılıkta optimizasyon mümkün olmamaktadır.
- 5- Otonom lojistik ile yüksek seviyede enerji tasarrufu sağlanabilmektedir. Robotların uyum içinde çalışması ve optimum görev ataması yapılması ile robotlar minimum derecede enerji harcayabilmektedir. Elbette bu görev dağılımı algoritmasının zamanı minimize etmesinin yanı sıra toplam harcanan enerjiyi de minimize etmesi bu konudaki olumlu etkiyi daha da artıracaktır.

Çağımızın önemli yönetsel araçlarından biri de izleme ve gözlem araçlarının zenginleştirilmesidir. Otonom lojistik sistemleri bu açıdan bakıldığında işletme yönetimine önemli avantajlar sağlamaktadır. Ürün/malzeme akışının otonom hale getirilmesi aynı zamanda dijitalleştirilmesi anlamına gelmektedir. Dijitalleşen bu süreçler için geliştirilen izleme sistemleri işletme kârlılığının artırılması hususunda önemli bir etkiye sahiptir. İzleme ve gözlem mekanizmaları aynı zamanda sistemdeki aksayan bölgeleri tespit edebilmeyi de sağlar.

## 2.4 Vaka Örneği Olarak Tekstil Terbiye İşletmeleri

Teze konu olan sistem bir Üniversite Sanayi İşbirliği Projesi (ÜSİP) kapsamında uygulanmıştır. Proje, denim üretimi yapan Çalık Denim A.Ş. bünyesindeki Ar-Ge Merkezi ile işbirliği içinde yürütülmüştür. İlgili kuruluşun tekstil terbiye işletmelerinde yarı mamul halindeki kumaşların bina içi lojistiğinin otonomlaştırılması hedeflenmiştir. Bu işletmelerde yarı mamul kumaşlar bir takım sıralı işlemlerden geçerek son halini almakta ve sonrasında teslimat noktasına ulaştırılmaktadır. Yarı mamullerin işlemden geçme sürecinde de taşımacılık yapılmaktadır. Bu tür üretim stratejisine sahip tesislerde bu işlemler çok fazla sayıda olmakta ve bu nedenle taşımacılık faaliyetleri bir karmaşaya sebep olmaktadır. Müşteri kuruluştan bir sipariş geldiğinde yarı mamulün (kumaşın) teslimat aşamasına kadar birçok işlemde geçmesi gerekebilir.

Kumaşların bina içindeki taşımacılığı dok arabaları ve dok çekiciler kullanılarak sağlanır. Kumaşlar rulo halinde dok arabasına sarılır ve bir operatör vasıtasıyla dok çekicileri ile hareket ettirilir. İşletme içinde hem elektrikli motorlarla çalışan hem de insan gücüyle çalışan dok çekiciler bulunmaktadır. Şekil 2.2’de kumaş yüklü dok arabası, Şekil 2.3’de elektrikli dok çekici ve Şekil 2.4’de insan gücüyle çalışan dok çekici örneği gösterilmiştir.



Şekil 2.2 : Kumaş yüklenmiş dok arabası.

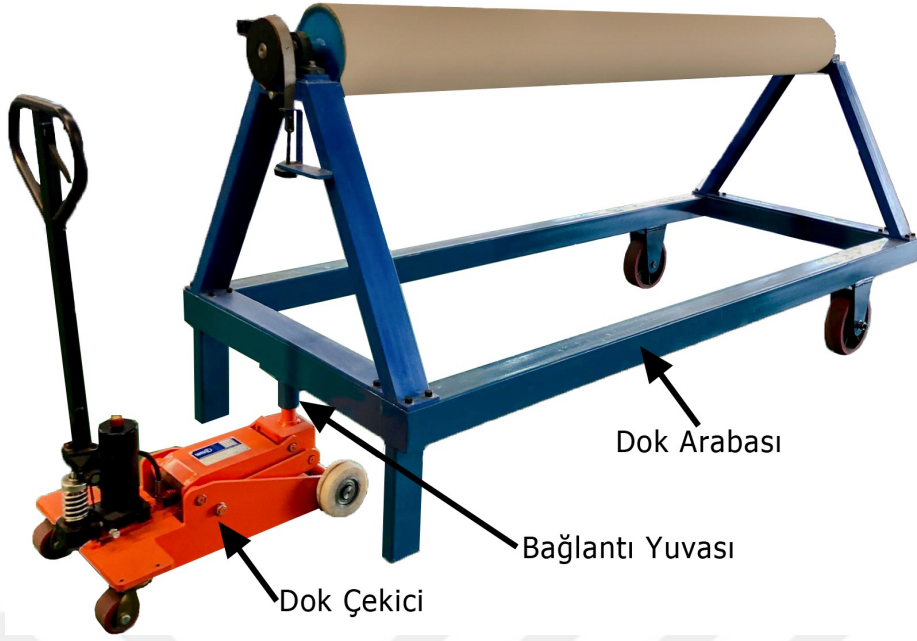


**Şekil 2.3 :** Elektrikli dok çekici.



**Şekil 2.4 :** İnsan gücüyle çalışan dok çekici.

Terbiye işletmesinde her kumaş yarı mamulden tam ürün haline gelene kadar farklı işlemlere tabi tutulur. Bu işlemler ön terbiye, renklendirme, boyama, baskı ve sanforlama gibi bazı süreçleri kapsamaktadır. Her denim kumaşı aynı işlemlere tabi tutulmayabilir. Kumaşın türüne göre kullanılan terbiye makineleri ve sıralamaları değişebilmektedir. Her kumaş için hangi sıralamayla hangi terbiye makinelerinde işlem göreceği bilgisi bir refakat kartı ile oluşturulur ve kumaşı taşıyan uzman operatöre verilir. Operatör kumaşın sarılı olduğu dok arabasını kullandığı dok çekiciye bağlayarak hareket etmesini sağlar. Şekil 2.5'te, bu bağlantının nasıl yapıldığına ilişkin örnek bir görsel yer almaktadır.



Şekil 2.5 : Dok çekicinin dok arabasına bağlanması.

Dok arabaları düz zemine sahip işletme koridorlarında hareket ettirilir ve terbiye makinelerinin giriş/çıkış portuna veya teslimat noktasına yönlendirilir. Örnek bir seyir koridoru Şekil 2.6'da ve terbiye makinesi Şekil 2.7'de gösterilmektedir. Tipik bir kumaş terbiye işletmesi, çoklu koridorlara ve bu koridorlarda çoklu terbiye makinelerine sahip olabilir.

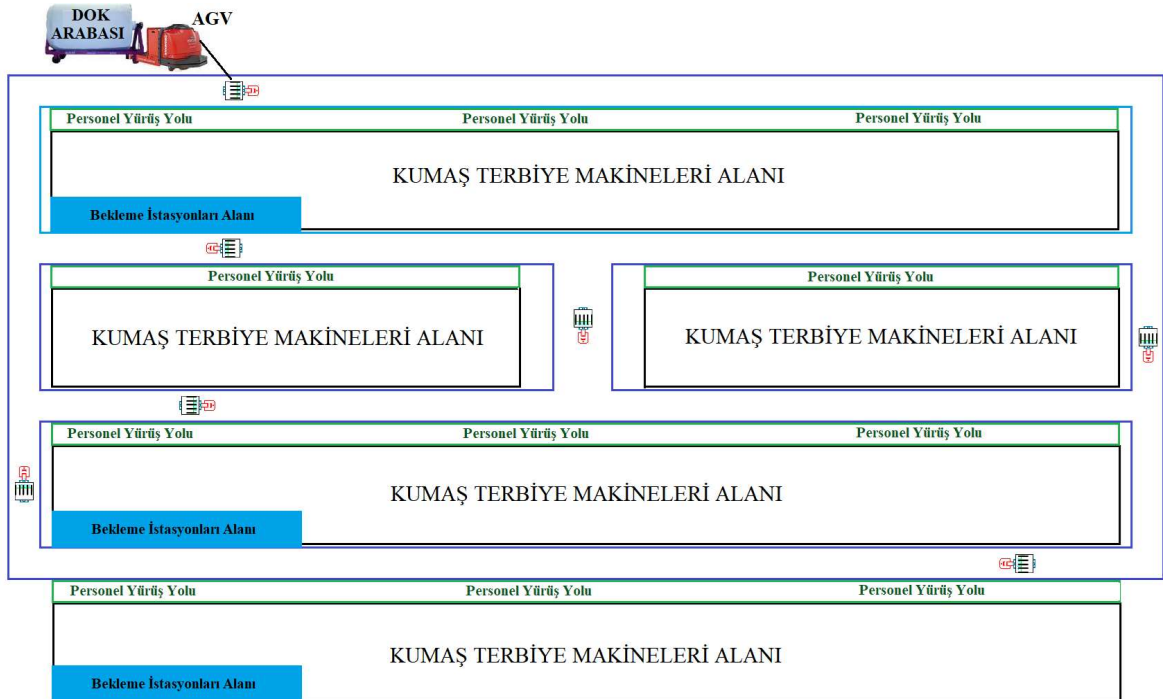


Şekil 2.6 : Dok çekici seyir koridoru.



Şekil 2.7 : Terbiye makinesi örneği.

Birçok terbiye işletmesinde, dok arabaları genellikle insan operatörler tarafından kullanılmaktadır. Bu işletmelerde aynı anda çoklu üretim gerçekleştirilir ve çoklu dok arabası hareket ettirilir. Tez çalışmasının geliştirilmesinde işbirliği yapılan Çalık Denim Ar-Ge Merkezinin terbiye işletmesine ait bir kroki örneği Şekil 2.8’de gösterilmiştir.



Şekil 2.8 : Örnek kumaş terbiye işletmesi krokisi.

## 2.5 Vaka Örneğindeki Bina İçi Lojistik Problemleri

Vaka örneğinde detaylıca anlatılan terbiye işletmelerindeki bina içi lojistik faaliyetlerinde karşılaşılan problemler, Çalık Denim A.Ş. Kumaş Terbiye İşletmesinde gerçekleştirilen analizler doğrultusunda, aşağıdaki gibidir:

- Koordinasyon: Kumaş taşımacılığının koordinasyonu hali hazırda birim şefleri tarafından gerçekleştirilmektedir. Sahadaki operatörlerin, kumaşların veya dok arabalarının durumları hakkında anlık bilgiye sahip olunmadığı için tam bir koordinasyon gerçekleşmemektedir.
- Uzman operatör ihtiyacı: Taşımacılık işleri için kullanılan dok çekicilerin yapacak personellerin bu alanda tecrübeye sahip olması gerektiğinden kalifiye çalışan istihdamı gerekliliği oluşmaktadır. Bu durumda personel giderlerinde hatırı sayılır miktarda maliyet artışı meydana gelmektedir. Ayrıca operatörler uzman olsalar dahi bazı durumlarda kumaşlar yanlış işlemlere tabi tutulabilmektedir.
- İş sağlığı ve güvenliği: Uzman personeller tarafından çekilen dok arabaları 3 tona kadar kumaşlar alabilmektedir. Taşınan bu yüklerin ağırlıkları ciddi ölçüde büyük olabildiği için bu alanda yaşanacak kazalar da ciddi yaralanmalara veya ölümlere yol açmaktadır.
- İş takibi problemi: Lojistik faaliyetlerin izlenebilir olmaması yapılan işlerin anlık olarak takip edilmesini de ortadan kaldırmaktadır. Bu durumda anlık karar mekanizmaları da oluşturulamamaktadır.
- Günlük ürün kapasitesi: Koordinasyonsuzluk ve insan odaklı çalışma prensipleri üretim hızını yavaşlatabilmektedir. Bu durumda işletmeler günlük üretim kapasitesinden dolayı müşteri kaybı yaşayabilmekte ve dolayısıyla maddi olarak zarar etmektedirler.
- Raporlama: Anlık olarak işlemde olan veya daha önce işlemde geçmiş ve müşteriye teslim edilmiş olan bir ürünün daha önce hangi lojistik aşamalarından geçtiği ile ilgili bir bilgi bulunmadığı ve bu bilgi rapor edilemediği için lojistik esnasında aksayan bölgeler tespit edilememektedir.

Lojistik faaliyetlerin otonom robotlar kullanılarak gerçekleştirilmesi, yukarıda belirtilen problemlerin direkt veya dolaylı olarak çözümlenmesini sağlayabilmektedir.

### 3. ENDÜSTRİYEL LOJİSTİK AMAÇLI OTONOM MOBİL ROBOTLARDA TEMEL PROBLEMLER ve ÇÖZÜM YAKLAŞIMLARI

Lojistik sistemlerde mobil robotların bir başlangıç noktasından hedef noktasına otonom olarak hareket ettirilmesi temel amaçtır. Eş zamanlı olarak çok sayıda üretim ve taşımacılık yapan endüstriyel işletmelerde çok robotlu taşımacılık gerektiği için robotların işbirliği içinde çalışma prensibi de göz önüne alınmalıdır. Çoklu robotların aynı ortamda eş zamanlı olarak ve işbirliği içinde hareket etmesini sağlamak için aşağıdaki temel problemlerin çözülmesi gerekmektedir:

- Eş zamanlı haritalama ve konumlandırma,
- Yol planlama,
- Çoklu robotlara çoklu görev dağılımı,
- Robot hareket modelleme
- Otonom ürün/malzeme yükleme bırakma

Yukarıda belirtilen temel problemlerin çözüm yöntemleri araştırmacılar için hala tam anlamıyla çözülememiş bir araştırma konusudur. Bu bölümde yukarıdaki problemler için literatürde geliştirilen çözüm yöntemleri ele alınmıştır.

#### 3.1 Eş Zamanlı Haritalama ve Konumlama

Mobil robotlarda haritalama, robotun bulunduğu ortamın uzamsal modelinin elde edilmesi demektir. Haritalama problemi, genel olarak, çalışma anında tasarlama ve ön tanımlı tasarım yaklaşımlarıyla ele alınmıştır. Robot veya robotu yöneten merkezi sistem tarafından bilinmeyen bir ortamda, robotun hedefine doğru ilerlemesi hesaplama maliyeti ve zaman açısından oldukça karmaşık bir durumdur. Böyle bir ortamda robot öncelikle nerede olduğu sorusunun cevabını bulmalı ve etrafındaki nesnelere anlamlandırmaya çalışmalıdır. Ön tanımlı bir harita ortamında robotun gezinmesi ise, sabit engellerin konumlarının biliniyor olmasından dolayı daha kolaydır. Endüstriyel işletmelerde hızlı, güvenilir ve kararlı sistemlerin varlığı üretim için önemli olduğundan tez çalışması, ortamın tam olarak bilinen bir haritasına dayanan konumlama stratejileriyle donatılmıştır. Haritaların bulut sunucu üzerinde oluşturulması da veri saklama ve hesaplama maliyetlerinin düşürülmesini sağlamıştır.

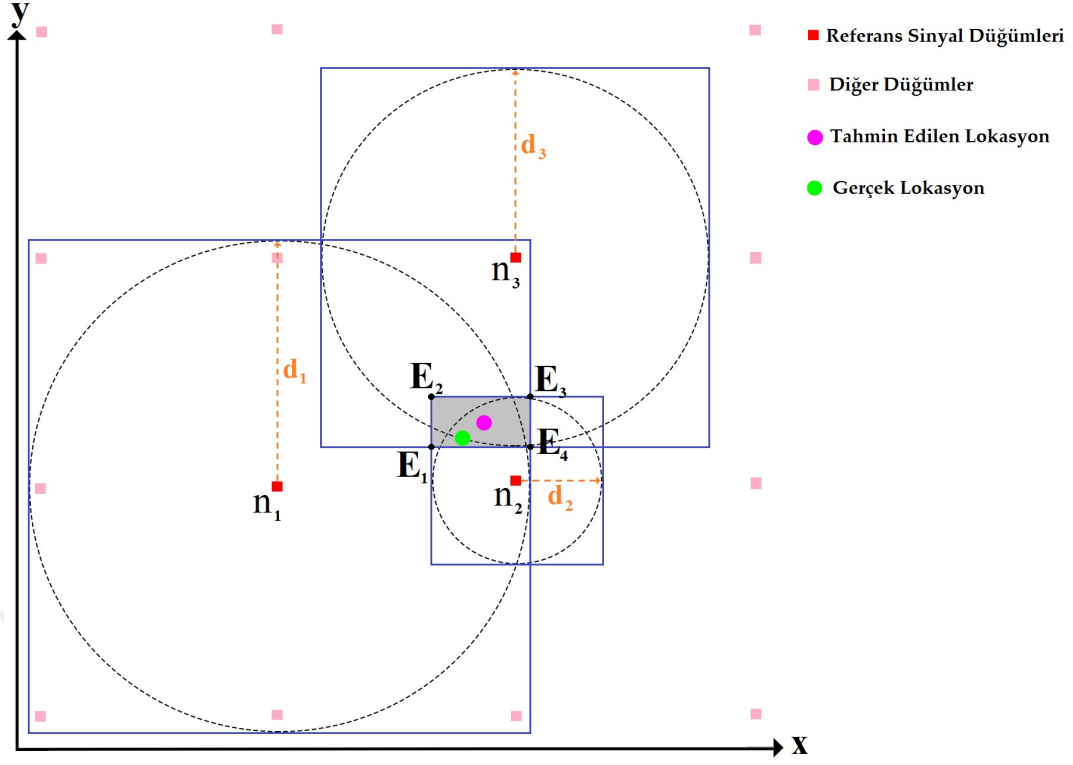
Ön tanımlı bir haritada robotun hareket etmesi için robotun kendisini bu harita üzerinde konumlandırması gerekmektedir. Bu durumda, robotun sensörleri aracılığıyla elde ettiği verilerin harita verilerine dönüştürülmesi ihtiyacı doğmaktadır. Eğer eş zamanlı bir konumlama yapılıyorsa bu durumda robotun konumunun bilinen harita üzerinde anlık olarak tespit edilmesi gerekmektedir.

Robotların konum tespitinde genellikle GPS, RFID, Bluetooth, Wi-Fi, Zeg-Bee ve Lidar gibi sensörler kullanılmaktadır. Bunun yanı sıra robotun teker enkoderlerinden gelen hareket verilerini kullanarak odometri tabanlı konumlama yapan sistemler de bulunmaktadır. GPS, RFID, Wi-fi ve Bluetooth gibi kablosuz erişim cihazlarında sinyal gücü ölçümüne dayalı konumlama yapılmaktadır. Bu cihazlarda, vericilerden alınan sinyal gücü göstergesi (RSSI) ile okunan vericinin mesafesini tahmin edilir.

RSSI bilgisi ile mesafe tahmini sinyal bozulmalarından ve ortamda sinyalleri etkileyen nesnelere bulunmasından dolayı her zaman doğru sonuçları vermeyebilir. Tek başına RSSI ile mesafe ölçümünde yanılma payı oldukça yüksek olabilir. Ancak endüstriyel işletmelerde tolere edilebilir hata payı oldukça düşüktür ve bu konuda iyileştirme yapılmalıdır. RSSI ölçümü ile mesafe tahmininde literatürde önerilen birçok yöntem vardır. Bu yöntemler arasında doğruluğunu ve güvenilirliğini kanıtlamış olan Min-Max, trilaterasyon, maksimum benzerlik, kalman filtresi ve parmak izi yaklaşımları ön plana çıkmaktadır.

### **3.1.1 Min-Max ile konum tahmini**

Min-Max konum tahmin algoritması, yüksek doğruluk, kolay hesaplama ve iyi ölçeklenebilirliğe sahip ve konumlama için en yaygın kullanılan tahmin algoritmalarından biridir. Bu yöntemde üç düğümden ( $n_1, n_2, n_3$ ) gelen RSSI değerleri referans olarak alınır. RFID antenin bu etiketlere olan uzaklığı olan  $d_1, d_2$  ve  $d_3$ , gelen RSSI verilerine göre hesaplanmaktadır. Sinyal gücünün ortamdaki çeşitli etkenlerden dolayı göreceli bir değer taşımasından dolayı bu uzaklık kesin olarak bilinemeyebilir. Ancak uzaklığın hangi aralıkta olduğu tahmin edilebilir. Min-Max metodunu uygulamadan önce referans olarak alınacak düğümlerin belirlenmesi ve bu düğümlerden gelen RSSI verilerinin uzaklık bilgisine dönüştürülmesi gerekmektedir. Robot aynı anda okuma yaptığı çok sayıda etiket okunabildiği için referans olarak alınacak düğümler RSSI değeri en yüksek olan ilk üç etiket seçilmektedir. Bu üç etiketten gelen sinyal verilerinin uzaklık bilgisine çevrilmesi ile tahmin işlemi gerçekleştirilebilir. Şekil 3.1'de Min-Max metodu ile lokasyon tahmininin nasıl gerçekleştirildiği gösterilmiştir.



**Şekil 3.1 :** Min-Max metodu ile lokasyon tahmini.

Şekil 3.1’de  $n_1, n_2$  ve  $n_3$  robota en yakın olan ve referans olarak alınan etiketlerin x-y düzlemindeki noktalarıdır.  $d_1, d_2$  ve  $d_3$  ise bu noktalardaki etiketlerden gelen RSSI bilgisinin mesafe bilgisine dönüştürülmüş değeridir. Robotun tahmini konumlama bölgesi olan ve  $(E_1, E_2, E_3, E_4)$  noktaları ile sınırlandırılan alanın hesaplanması şu şekilde yapılmaktadır:

$$E_1 = (x_{min}, y_{min}) \quad (3.1)$$

$$E_2 = (x_{min}, y_{max}) \quad (3.2)$$

$$E_3 = (x_{max}, y_{max}) \quad (3.3)$$

$$E_4 = (x_{max}, y_{min}) \quad (3.4)$$

$$n_1 = (x_1, y_1), n_2 = (x_2, y_2), n_3 = (x_3, y_3) \quad (3.5)$$

$$x_{min} = \max(x_1 - d_1, x_2 - d_2, x_3 - d_3) \quad (3.6)$$

$$x_{max} = \min(x_1 + d_1, x_2 + d_2, x_3 + d_3) \quad (3.7)$$

$$y_{min} = \max(y_1 - d_1, y_2 - d_2, y_3 - d_3) \quad (3.8)$$

$$y_{max} = \min(y_1 + d_1, y_2 + d_2, y_3 + d_3) \quad (3.9)$$

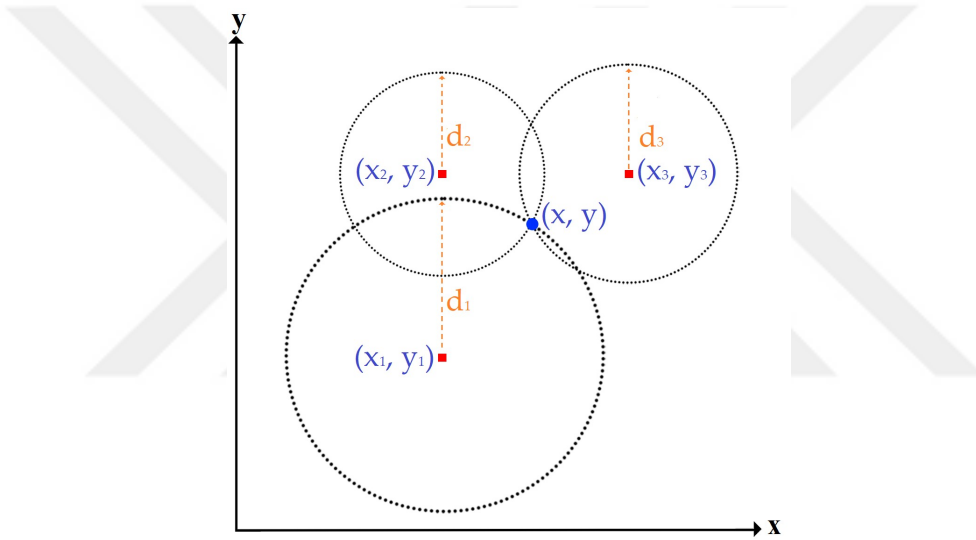
Tahmin edilen nokta  $(x_t, y_t)$  tahmin bölgesinin orta noktası olmak üzere;

$$x_t = \frac{x_{min} + x_{max}}{2} \quad (3.10)$$

$$y_t = \frac{y_{min} + y_{max}}{2} \quad (3.11)$$

### 3.1.2 Trilaterasyon (üçleme) ile konum tahmini

Trilaterasyon, bir daire, küre veya üçgenin geometrisini kullanarak mesafe ölçümü ile mutlak konumu veya göreceli noktaları belirleme işlemidir (Palaskar, Palkar, & Tawari, 2014). Trilateral, kör düğüm olduğunu düşündüğümüz düğümün konumunu tahmin etmek için esasen üç düğüm kullanır. Örnek bir konum tahmini hesaplama işlemi Şekil 3.2'de gösterilmiştir.



Şekil 3.2 : Trilaterasyon metodu ile lokasyon tahmini.

Şekil 3.2'de görüldüğü gibi RSSI sinyalinin gelen ölçümlerle elde edilen mesafe bilgisi yarıçap olacak şekilde daireler çizilir. Dairelerin kesişim noktası  $(x, y)$  aşağıdaki gibi hesaplanır:

$$d_1^2 = (x - x_1)^2 + (y - y_1)^2 \quad (3.12)$$

$$d_2^2 = (x - x_2)^2 + (y - y_2)^2 \quad (3.13)$$

$$d_3^2 = (x - x_3)^2 + (y - y_3)^2 \quad (3.14)$$

Yukarıdaki denklemler, matrisler kullanılarak kolayca çözülebilen doğrusal bir eş zamanlı denklemler kümesine indirgenir. Sistem, robotun bina içindeki konumu için benzersiz bir çözüm  $(x, y)$  verir.

Trilaterasyon (üçleme) tekniđi, mesafe ölçümlerinin doğru olduğunu varsayar. Ancak gerçekte, RFID sinyalinin sinyal gücünü etkileyen çok yönlü yayılma ve çevrede aynı frekans spektrumunu paylaşan diđer radyo sinyali kaynaklarının karışması gibi faktörlerin sayısı nedeniyle hatalara eğilimlidir. Bu nedenle, üçleme tekniđi en iyi konum doğruluđunu sunmayabilir çünkü RSSI ölçümlerinden doğru mesafelerin elde edilmesi görüldüğünden daha karmaşıktır.

### 3.1.3 Maksimum benzerlik ile konum tahmini

Maksimum benzerlik tahmini (MLE) aynı zamanda çok taraflılaştırma (multilateration) olarak da adlandırılır ve trilaterasyonun bir uzantısıdır. Üçten fazla referans düđüm ve bir hedef düđüm arasındaki mesafeler elde edildiğinde, maksimum benzerlik tahmini, hedef düđümün konumunu hesaplamak için en küçük kare yöntemini kullanır. MLE yöntemleri, referans düđümü  $P_i$  ile bilinmeyen düđüm  $P$  arasında ölçülen  $R_i$  mesafesinin, Gauss normal dağılımına  $N(\mu_i, \sigma^2)$  tâbi rastgele bir deđişken olduğunu ileri sürer; burada  $\mu_i$ , referans düđümü  $P_i$  ile bilinmeyen  $P$  düđümü arasındaki gerçek mesafedir ve  $\sigma^2$  bir sabittir. MLE ayrıca, farklı referans düđümleri ve bilinmeyen düđümler tarafından farklı ölçüm mesafelerinin birbirinden bağımsız olduğunu varsayar (J. Xu, He, Zhang, Xu, & Cai, 2016).  $R_i$  mesafesinin ölçülen örnek deđeri  $r_i$ ,  $i = 1, 2, \dots, N$  olsun;

$$d_1^2 = (x - x_1)^2 + (y - y_1)^2 \quad (3.15)$$

$$f_i(t) = (1/\sqrt{2\pi\sigma})e^{-(t-\mu_i)^2/2\sigma^2} \quad (3.16)$$

Ardından, MLE ilk olarak olabilirlik işlevini ařađıdaki gibi kurar:

$$\begin{aligned} L(x, y) &= \prod_{i=1}^n f_i(r_i) = \prod_{i=1}^n \left( \frac{1}{\sqrt{2\pi\sigma}} \right) e^{\frac{-(t-\mu_i)^2}{2\sigma^2}} \\ &= (1/\sqrt{2\pi\sigma})^n e^{-(1/2\sigma^2)\sum_{i=1}^n (r_i-\mu_i)^2} \end{aligned} \quad (3.17)$$

Burada;

$$\mu_i = \sqrt{(x_i - x)^2 + (y_i - y)^2} \quad (3.18)$$

Böylece tahmini noktalar řu řekilde hesaplanır:

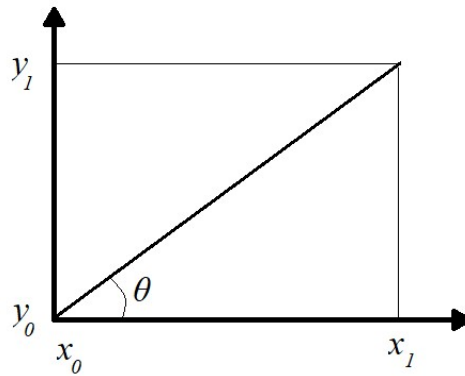
$$\sum_{i=1}^n (x_i - x) \frac{(\mu_i - r_i)}{\mu_i} = 0 \quad (3.19)$$

$$\sum_{i=1}^n (y_i - y) \frac{(\mu_i - r_i)}{\mu_i} = 0 \quad (3.20)$$

Yukarıdaki denklemleri çözerek, bilinmeyen düğüm için maksimum olasılık tahmin değerini elde edebiliriz. Bu yöntem, aralık temelli çok taraflı ölçümle konumlama yöntemlerine kıyasla konumlama doğruluğunu artırır.

### 3.1.4 Odometri ile konum tahmini

Kilometre sayacı olarak da bilinen odometri kelimesi, yol ve ölçü anlamına gelen Yunanca kelimelerden gelir ve bir robotun konumunu bulması için en çok kullanılan tekniğin adıdır. Odometri, zaman içinde artan bilgileri birleştirerek çalışır. Robot, her bir tekerleğin devir sayısını hesaplamak için tekerlek kodlayıcıları kullanarak kat ettiği mesafeyi ve gidiş yönünü ölçer. Odometri, kısa vadeli doğruluk sağladığından ve ucuz olduğundan yaygın olarak kullanılmaktadır (Ben-Ari & Mondada, 2018). Ancak, sürüklenme ve kayma nedeniyle, tekerlek devirlerinin entegrasyonu hem gidilen mesafede hem de yönelimde hatalara yol açar ve bu hatalar zamanla birikir (Negenborn, 2003). Özellikle yönelimdeki hatalar büyük konumlandırma hatalarına neden olur. Odometrinin bir başka dezavantajı, araziye duyarlılığıdır. Robotun üzerinde çalıştığı yüzey düzgün değilse, odometri sistemi alandaki düzensizlikleri algılayamayacağı için önemli konum hatalarına neden olabilir. Ayrıca, tekerlek çapındaki farklılıklar da odometri ölçümleri tarafından algılanmayan konum hatalarına neden olabilir. Odometri, konum tahmininde artan hataya neden olsa da, konum bilgisinin erişilmesi en kolay tekniktir ve bu nedenle konumlama için önemli bir bilgi kaynağıdır. Bu yöntemle göre belirli sayıda teker dönüşlerinden yola çıkarak ne kadar mesafe kat edildiği bilgisine ulaşılabilir. Odometri iki şekilde ele alınmalıdır. Lineer odometri ve dönüş odometrisi. Şekil 3.3’de lineer odometri gösterilmiştir.



Şekil 3.3 : Lineer Odometri.

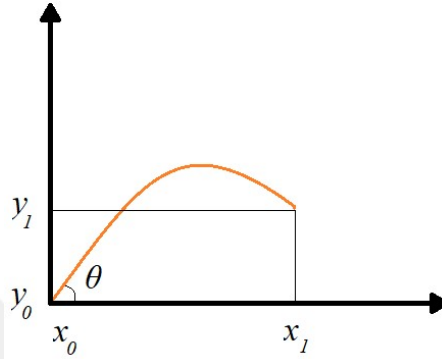
Şekil 3.3’te  $x_0$  ve  $y_0$  robotun başlangıç konumunu temsil etmektedir.  $x_1$  ve  $y_1$  ise başlangıç konumuna göre  $\theta$  açısı ile hareket eden robotun  $\Delta t$  zamanda  $v$  hızıyla aldığı  $d$  mesafesinin

konumunu temsil etmektedir. Şekil 3.3'te  $(x_1, y_1)$  lokasyonu Denklem 3.21 ve Denklem 3.22'deki eşitlikler kullanılarak hesaplanır.

$$d = \Delta t * v \quad (3.21)$$

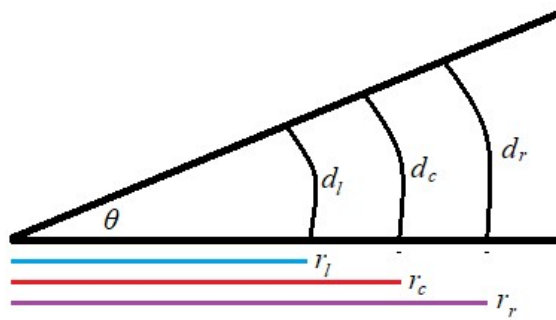
$$x_1 = d * \cos(\theta), \quad y_1 = d * \sin(\theta) \quad (3.22)$$

Robotun dönüş odometrisi ise lineer odometriye göre farklılık göstermektedir. Dönüş yapıldığında alınan mesafeyle konum doğru orantılı artmayabilir. Örnek bir dönüş odometrisi Şekil 3.4'te gösterilmiştir.



Şekil 3.4 : Dönüş Odometrisi.

Şekil 3.4'te görüleceği üzere dönüş odometrisinde doğrusal bir yol takibine göre alınan mesafe daha kısadır. Dönüş odometrisini hesaplamak için;  $d_r$  sağ tekerin aldığı mesafe,  $d_l$  sol tekerin aldığı mesafe,  $d_c$  merkez noktanın (sol ve sağ teker arasındaki bölmenin orta noktası) tekerin aldığı mesafe ve dönüş çemberlerinin yarıçaplarının ( $r_r$ ,  $r_l$ ,  $r_c$ ) ele alınması gerekir. Şekil 3.5'te örnek bir sola dönüş teker kinematiği simüle edilmiştir.



Şekil 3.5 : Sola Dönüş Teker Kinematiği.

Dönüş kinematiği için aşağıdaki denklemler oluşturulur;

$$d_l = 2\pi r_l * (\theta/2\pi) = \theta r_l \quad (3.23)$$

$$d_r = \theta r_r \quad (3.24)$$

$$d_c = (d_l + d_r)/2 \quad (3.25)$$

Sonuç olarak tahmini konum şu şekilde elde edilir;

$$x_1 = d_c * \cos(\theta), \quad y_1 = d_c * \sin(\theta) \quad (3.26)$$

### 3.1.5 Parmak izi tabanlı konum tahmin teknikleri

Parmak izi tekniğinde, kablosuz erişim noktalarından gelen RSSI değerleri, mevcut konumdaki ölçümleri bilinen referans noktası bilgisiyle depolanan RSSI verileriyle karşılaştırılarak konumlandırma için kullanılır (Golenbiewski & Tewolde, 2020). Robotun konumu hakkında iyi bir tahmin elde etmek için istatistiksel ilişkiler kullanılır. Parmak izi tabanlı konumlandırma tekniği eğitim ve konumlandırma olmak üzere iki aşamadan oluşur. Eğitim aşamasının amacı, harita ortamı için bir parmak izi veritabanı oluşturmaktır. Veritabanı oluşturulurken, ortamın tamamını tanımlayacak referans noktaları seti dikkatlice tanımlanmalıdır. Bir mobil robot daha sonra bu seçilen referans noktası konumlarına birer birer gider ve her RFID okuyucudan RSSI değerlerini toplar. Bu ölçüm, parmak izi veri setini sağlamak için veritabanında saklanır. Bu ölçümler daha sonra uygun bir arama veya istatistiksel eşleştirme algoritması kullanılarak veritabanında önceden toplanan parmak izi verileriyle karşılaştırılır. Oluşturulan sonuç daha sonra kullanıcı için tahmin edilen konumu vermektedir.

Bu yöntemde, doğru bir RF sinyal yayılma modelinin geliştirilmesine ihtiyaç bulunmamaktadır. RSSI değerlerini mesafelere dönüştüren trilaterasyon tekniğinin aksine, doğrudan RSSI değerlerini kullanarak bunu önler. Parmak izi tabanlı konumlandırma tekniğinin dezavantajı, çevrimdışı veri toplama aşamasını gerçekleştirilmenin önemli miktarda zaman almasıdır. Dahası, dinamik olarak değişen kapalı ortamlarda büyük nesnelere hareketi, konumlandırma doğruluğunu olumsuz etkileyebilir. Bu tür etkileri en aza indirmek için, parmak izi verilerinin, iç ortam değiştikçe sistemin maliyetini artıracak şekilde güncellenmesi gerekebilir.

### 3.1.6 Kalman filtresi tabanlı konum tahmini

Kalman filtresi kullanan konum tahmin yöntemlerinde ortamı tepeden görece bir kamera ile görüntü işlemeye dayalı bir yaklaşım vardır. Robotun konumunu hesaplamak için görüntü verileri ölçülerek trigonometri kullanıldığında, büyük bir hataya sahip olabilir ve o andaki ölçüme bağlı olarak çerçeveden çerçeveye önemli ölçüde değişebilir. Kalman filtresi, ölçümlerin gürültülü olduğunu ve bazen ihmal edilmeleri veya durum tahmini üzerinde yalnızca küçük bir etkiye sahip olmaları gerektiğini kabul ederek ölçüm verilerini tahmin eder (Negenborn, 2003). Güvenilir verilerden, güvenilirmez verilerden daha fazla bilgi

ekleyerek tahmin edilen durum değişkenindeki gürültünün etkilerini düzeltir. Kalman filtresine sistemde ne kadar gürültü olduğu parametresi girilerek ve gürültüyü hesaba katarak konumun tahmin edilir.

### 3.2 Yol Planlama

Bir mobil robotun yol planlaması, mesafeyi optimize ederek başlangıç noktasından hedef noktasına çarpışmasız yolu belirlemektir. Çevre hakkındaki bilgilerin mevcudiyetine bağlı olarak, çevrimdışı ve çevrimiçi olmak üzere iki tip yol planlama algoritması kategorisi vardır (Raja & Pugazhenti, 2012). Sabit engeller ve hareketli engellerin yörüngesi hakkında tam bilginin önceden bilindiği ortamlarda robotların çevrimdışı yol planlaması, küresel yol planlaması olarak da bilinir. Çevre hakkında eksiksiz bilgi önceden mevcut olmadığında, mobil robot, çevrede hareket ederken sensörler aracılığıyla bilgi alır. Bu, çevrimiçi veya yerel yol planlaması olarak bilinir. Esasen, çevrimiçi yol planlaması ilk yoluna çevrimdışı olarak başlar, ancak engel senaryosunda yeni değişiklikler keşfettiğinde çevrimiçi moda geçer.

Mobil robotun küresel yol planlama sürecinde öncelikli olarak çevresel modelleme yapılmalıdır. Bu aşama haritalama olarak da bilinir ve ortamdaki nesnelerin harita verisine dönüştürme işlemidir. Haritalamadan sonra yol planı için optimizasyon parametrelerinin belirlenmesi ve yol arama algoritmasına karar verilir. Yol arama algoritması, yol uzunluğu, pürüzsüzlük, güvenlik derecesi vb. gibi bir dizi optimizasyon kriterini karşılaması gereken durum uzayında başlangıç noktası ile hedef nokta arasında çarpışmasız bir yol bulmak için arama yapar. Küresel yol planlama alanındaki arama algoritmalarını graf tabanlı yaklaşımlar ve meta-sezgisel yaklaşımlar başlıkları altında değerlendirebiliriz. Graf tabanlı yaklaşımlarda güncel olarak kullanılan ve öne çıkan arama yöntemlerine A\*, D\*, Dijkstra algoritmaları örnek olarak gösterilebilir. Meta-sezgisel yol planlama yöntemlerine ise YSA, GA, PSO, ACO algoritmaları örnek olarak gösterilebilir.

Çevrimiçi yol planlama stratejisi yerel yol planlama ismiyle de anılır. Yerel yol planlaması robotun çalışma anında ve bilinmeyen bir ortamda yol planını oluşturmasına dayanır. Bu alandaki arama algoritmalarına örnek olarak APF, vaka temelli öğrenme ve Rolling Windows gösterilebilir.

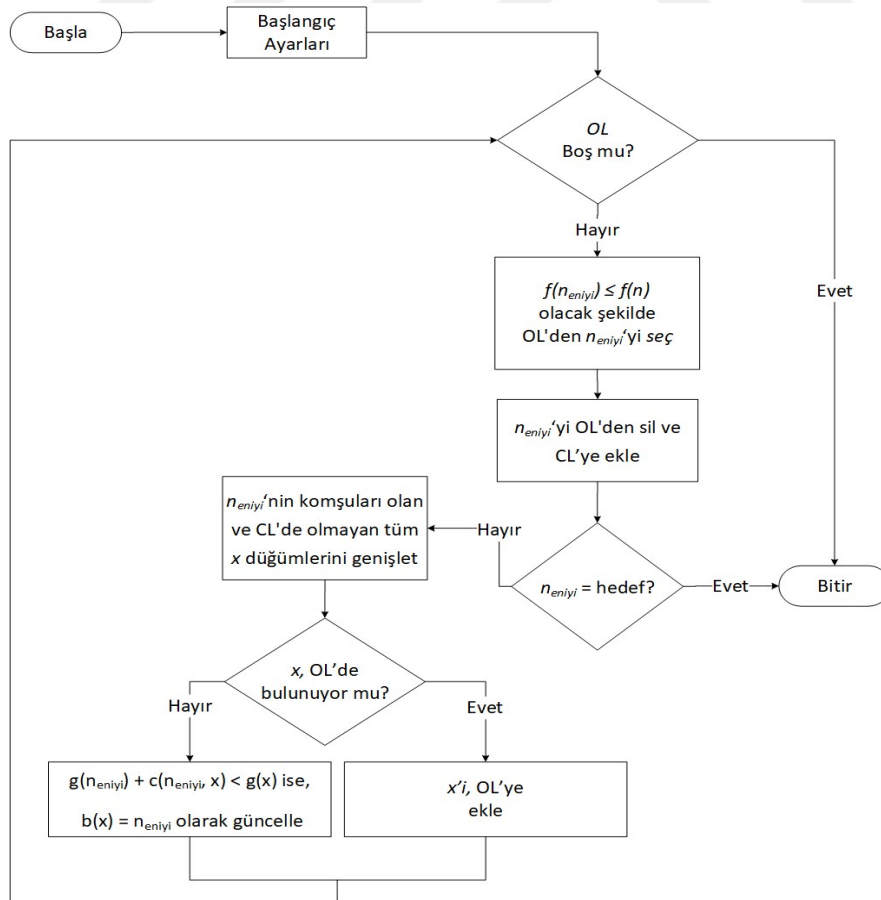
Tez çalışmasında bulut sistem temelli küresel yol planlama yöntemi kullanılmıştır. Böylece, bu bölümde küresel yol planlama algoritmalarında sıklıkla kullanılan yöntemler detaylı olarak verilmiştir.

### 3.2.1 A\* algoritması

A\* başlangıç düğümünden hedef düğüme en kısa hangi düğümler üzerinden gidileceğini bulmaya yarayan bir en iyi yerleştirme algoritmasıdır. Yapı olarak yarı sezgisel bir algoritmadır (Okumuş & Kocamaz, 2019). A\* mesafe hesaplamada aşağıdaki fonksiyonu kullanır:

$$f(n) = g(n) + c(n) \quad (3.27)$$

Denklem 3.27'de  $f(n)$  hesaplama yapan sezgisel fonksiyon,  $g(n)$  başlangıç düğümünden mevcut düğüme kadar gelmenin maliyeti ve  $c(n)$  mevcut düğümden hedef düğüme varmak için tahmin edilen mesafeyi göstermektedir. Arama, düğümler hakkında bilgi depolamak için iki liste gerektirir: 1) Açık liste (OL), genişletmeler için düğümleri depolar 2) Kapalı liste (CL) araştırdığımız düğümleri depolar. A\* algoritmasının akış diyagramı Şekil 3.6'da verilmiştir.



Şekil 3.6 : A\* algoritması akış diyagramı.

### 3.2.2 D\* algoritması

D\* algoritması, değişen ortamlarda, hızlı ve yeniden planlama yapabilen bir graf tabanlı arama algoritmasıdır (Stentz, n.d.). Dijkstra algoritmasının dinamik bir versiyonu veya sezgisel fonksiyonu olmayan A\* algoritmasının dinamik versiyonu olarak da bilinir(Dakulovi & Petrovi, 2011). Algoritma, zaman içinde ağırlıkların değiştiği graflarda en uygun yolu bulmaya çalışır.

Algoritma A\* gibi geçiş noktalarını bir düğüm olarak ele alır. Düğümler "OPEN liste" olarak bilinen dizide tutulur. Her bir düğüm birkaç durumdan ile işaretlenir. Bu durumlar: NEW (listeye ilk defa giren düğüm), OPEN (Hali hazırda listede olan düğüm), CLOSED (Artık listede olmayan düğüm), RAISE (son zamana göre maliyetinin arttığını gösteren durum) ve LOWER (maliyetin düştüğünü gösteren durum). Algoritma, OPEN listesinden bir düğüm seçerek başlar. Her adımda düğümün değişiklikleri tüm komşu düğümlere iletilir ve OPEN listesine yerleştirir. Bu iletim işlemi "genişleme" olarak adlandırılır. D\* algoritmasında en kısa yol, hedef düğümünden geriye doğru arama yaparak başlar. Her bir genişletilmiş düğüm, hedefe giden bir sonraki düğüme atıfta bulunan bir geri-ayırıcıya sahiptir ve hedefe olan tam maliyeti bilmektedir. Başlangıç düğümü, genişletilecek bir sonraki düğüm olduğunda, algoritma tamamlanır ve hedefe giden yol, yalnızca arka taraftaki nesnelere takip ederek bulunabilir.

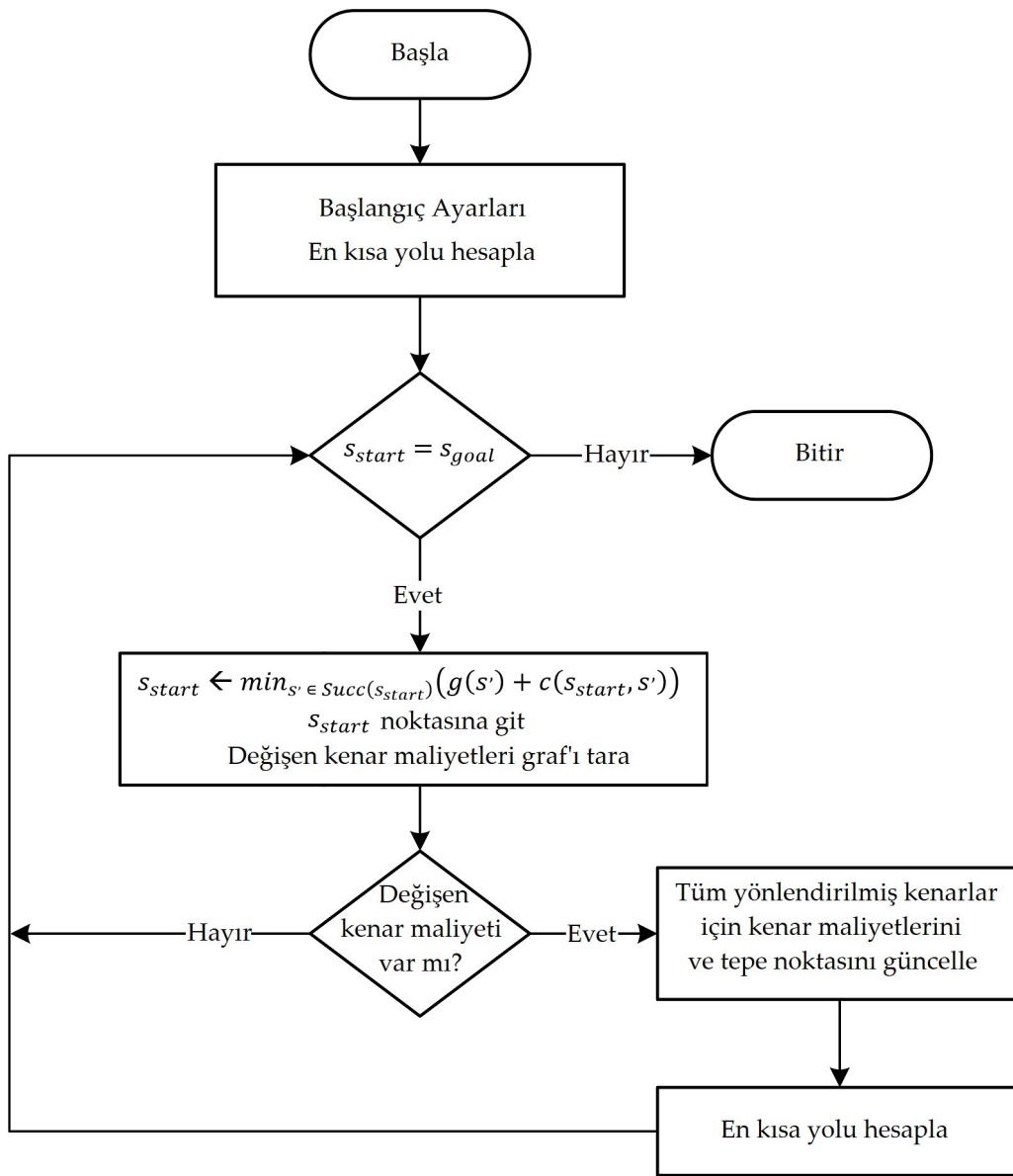
### 3.2.3 D\* Lite algoritması

D\* algoritmasının bir türevidir olan D\* Lite, Sven Koenig ve Maxim Likhachev tarafından geliştirilen, A\* ve Dynamic SWSF-FP fikirlerini birleştirerek ve LPA'yı (Uzun Süreli A\* Planlama) geliştiren, artan sezgisel bir arama algoritmasıdır (Koenig & Likhachev, 2002). D\* Lite, hedef düğümünden ( $s_{goal}$ ) başlayarak, başlangıç düğüme ( $s_{start}$ ) kadar çarpışmasız yolu arar. Algoritma, her  $g(s)$  tepe noktası ve  $rhs(s)$  hedefine her bir köşe arasındaki uzaklık arasındaki tahmini mesafeyi depolar.  $rhs$  değerleri aşağıdaki bağıntı ile tahmin edilir:

$$rhs(s) = \begin{cases} 0 & s = s_{goal} \\ \min_{s' \in Succ(s)} (g(s') + c(s, s')) & \text{diğer seçenekler} \end{cases} \quad (3.28)$$

Burada  $s'$ ,  $s$ 'nin komşusudur ve  $Succ(s)$  kümesi  $s$ 'den genişleyerek oluşan tepe noktalarıdır.  $c(s, s')$  ise,  $s$ 'den  $s'$ 'ye transferin maliyetidir. D\* Lite tüm tutarsız köşeleri depolamak için bir öncelikli kuyruk ( $U$ ) yapısı kullanır. Başlangıçta  $g$  ve

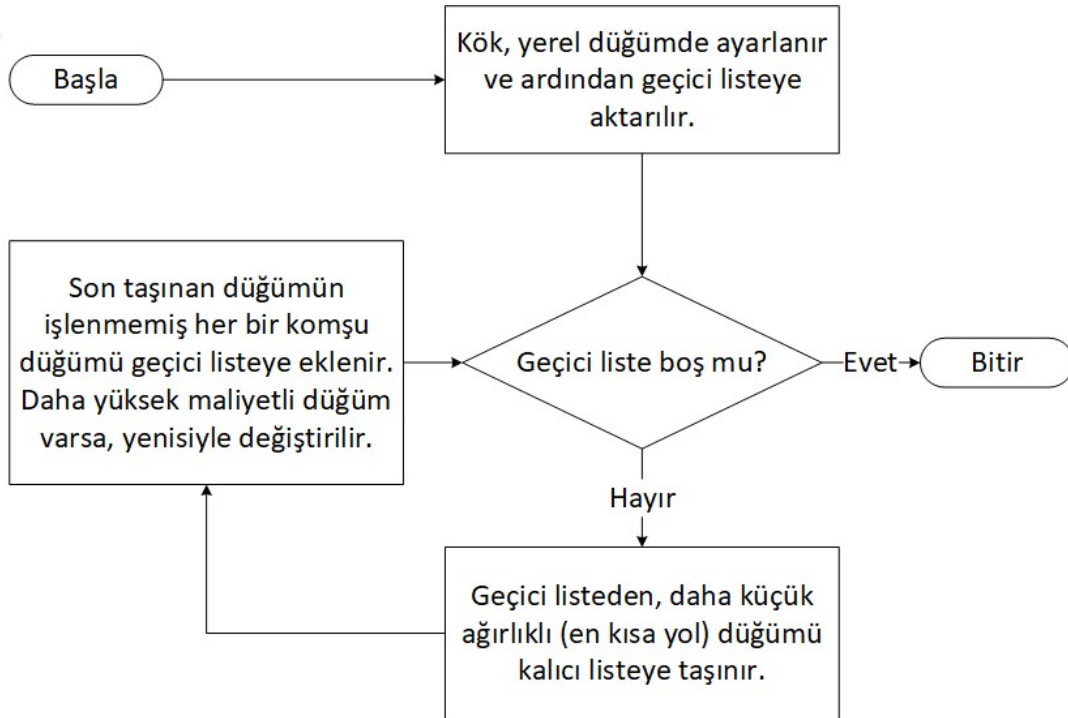
$rhs$  parametrelerinin tümü sonsuzdur ve  $rhs$  değeri 0 olan  $s_{goal}$ ,  $U$ 'ya eklenir. İlk önce,  $U$ 'dan en az öncelikli  $k$  değerine sahip  $u$  tepe noktası seçilir. Eğer  $g(u) > rhs(u)$  ise,  $u$ 'nun  $rhs$ -değeri,  $g$ -değeri olarak ayarlanır ve  $u$ ,  $U$ 'dan çıkarılır. Ardından  $u$ 'nun öncüllerinin tüm  $rhs$  değerleri değiştirilir ve eğer yerel olarak tutarsızsa,  $U$ 'ya eklenirler. Eğer  $g(u) \leq rhs(u)$  ise,  $u$ 'nun değeri sonsuz olarak ayarlanır ve yerel olarak tutarsızlarsa  $U$ 'ya eklenirler. Bu işlemler  $s_{start}$  noktası lokal olarak tutarlı olana kadar ve  $U$ 'da kalan tepe noktalarının tümünün  $k$  değeri  $s_{start}$  değerinden daha büyük olana devam eder. Bu işlemlerin sonucunda en uygun yol planı elde edilir. D\* Lite algoritmasının akış diyagramı Şekil 3.7'de gösterilmiştir.



Şekil 3.7 : D\* Lite algoritması akış diyagramı.

### 3.2.4 Dijkstra algoritması

Dijkstra Algoritması temelde seçilen kaynak düğümden (başlangıç noktası) başlar ve bu düğüm ile graftaki diğer tüm düğümler arasındaki en kısa yolu bulmak için grafi analiz eder. Her düğümden kaynak düğüme güncel olarak bilinen en kısa mesafeyi izler ve daha kısa bir yol bulursa bu değerleri günceller. Kaynak düğüm ile başka bir düğüm arasındaki en kısa yolu bulduğunda, bu düğüm "ziyaret edildi" olarak işaretlenir ve yola eklenir. İşlem, grafikteki tüm düğümler yola eklenene kadar devam eder. Bu şekilde, her bir düğüme ulaşmak için mümkün olan en kısa yolu izleyerek kaynak düğümü diğer tüm düğümlere bağlayan bir yola sahip olunur. Arama esnasında hedef düğüm bulunduğu en kısa yol da bulunmuş olur. Dijkstra algoritmasının akış diyagramı Şekil 3.8’de gösterilmiştir.



Şekil 3.8 : Dijkstra algoritması akış diyagramı.

### 3.2.5 PSO algoritması

Parçacık sürü optimizasyonu (PSO) algoritması meta-sezgisel algoritmalar içerisinde yol planlaması için en çok kullanılan yöntemlerden biridir. Kennedy ve Eberhart tarafından geliştirilen PSO temel olarak sürü istihbaratı konseptinden faydalanır (Eberhart & Kennedy, 1995). Yerel olmayan, çevreleriyle etkileşim kuran, karmaşık özelliklere sahip ajanların ortak davranışlarının tutarlı, küresel ve işlevsel kalıplar oluşturduğu bir sistem oluşturur. Geleneksel PSO'da, aday çözümlere parçacık denir. Bu parçacıklar, arama alanını dolaşarak ve komşu parçacıklarla etkileşerek en iyi çözümü elde eder. PSO algoritmasının her

iterasyonunda, tüm sürü içinde elde edilen en iyi global çözüm güncellenir. Parçacıkların tümü bu bilgiyi kullanarak parçacığın yerini fark eder ve bu parçacığa yakınsamaya çalışır. Parçacığı modellemek için t. iterasyonda her n parçacığı, konumsal  $x_n^s$  ve hız  $v_n^s$  değerlerine göre çok boyutlu bir alanda hareket eder. Bu değerler, yerel en iyi olarak bilinen  $x_{1_n}^s$  ve global en iyi olarak bilinen  $x_{2_n}^s$ 'ye bağlı olarak değişkenlik gösterir. Bu bağımlılık şu şekilde ifade edilir:

$$v_n^s[t + 1] = wv_n^s[t] + \sum_{i=0}^2 p_i r_i (x_{i_n}^s[t] - x_n^s[t]) \quad (3.29)$$

$$x_n[t + 1] = x_n[t] + v_n[t + 1] \quad (3.30)$$

Denklem 3.29'da yeni hız değerini elde etmek için belirtilen  $w$  atalet etkisi ağırlığını,  $p$  yerel en iyi katsayısını,  $r$  ise küresel en iyi katsayısını temsil eder.

### 3.3 Çoklu Robot Görev Dağılımı

Çoklu robot görev dağılımı (MRTA) problemi optimal atama problemi (OAP) olarak da bilinir ve sistemin hedeflerine ulaşmak için robota en uygun görev atamalarını bulma sorusunu ele alır. MRTA, çoklu robotların aynı ortamda eş zamanlı olarak ve işbirliği içinde görevlerini verimli ve güvenilir bir şekilde gerçekleştirmek için nasıl koordine edildikleri ile ilgilenir. Görev dağılımı problemi, çevresel değişikliklerden dolayı zaman içinde değişen dinamik bir karar problemi olduğundan, problem zaman içinde yinelemeli olarak çözülmelidir. Ancak bu durumda, optimum görev dağılımı probleminin üstesinden gelmek daha karmaşık hale gelir. Söz konusu sistemin gereksinimleri, çok robotlu görev tahsis problemlerinin özelliklerini ve karmaşıklığını etkiler (Khamis vd., 2015). Lojistik görevlerinin optimal bir şekilde dağıtıldığı bir sistem için düşünülecek olursa üretim sahası dinamik bir yapıya sahip olacaktır. Görevler eş zamanlı olarak ifa ediliyor olsa da başlangıç zamanları eş zamanlı olmayacaktır. Bu durumda sürekli değişen bir görev listesi ve görev ataması durumu ortaya çıkmaktadır.

Çoklu görev dağılımının yapılabilmesi için öncelikle problemin formüle edilmesi gerekmektedir. Bir sistemin optimal olarak çalışması formüle edildiği zaman mümkün olabilir. Sisteme etki eden faktörlerin ve unsurların formüldeki yeri belirlenir ve verilen değerlerine göre sonuca etkisi gözlemlenir. Lojistik sistemlerde görevler taşımacılık işlemleridir ve taşımacılıkta optimal maliyeti elde etmek için iki unsur minimize edilmelidir. Birinci unsur taşımacılık yapılırken robotlar tarafından takip edilen yolların toplamının

minimum olmasıdır. İkinci unsur ise gün sonunda elde edilecek olan ürün sayısının artırılması bakımından taşımacılık faaliyetlerinin toplam süresinin minimum olmasıdır. Bu durumda lojistik için MRTA çok amaçlı bir optimizasyon problemi olarak karşımıza çıkmaktadır. Henüz ifa edilmemiş çoklu taşımacılık görevlerinin en uygun biçimde robotlara atanması tam anlamıyla bir işbirliği sistemini gerektirir. Çoklu robotların işbirliği içinde çalışması için iki tip yaklaşım vardır; merkezi kontrol ve merkezi olmayan kontrol sistemleri.

Merkezi kontrol yaklaşımını benimseyen sistemlerde, her robot, görevleri tüm robotlara tahsis eden bir merkezi birimle bağlantı kurar. Böylelikle, robotlar sahip oldukları tüm bilgileri bu merkezi birime gönderir ve merkezi birim bu bilgileri işleyerek atanan görevleri yürütmek için robotlara uygun komutları gönderir. Merkezi sistemin çökmesi ile tüm faaliyetlerin duruyor olması bu yaklaşımın en büyük dezavantajıdır. Ancak bu tez çalışmasında da uygulanan bulut sistem üzerinde bir merkezi birim oluşturma fikri bu dezavantajın önüne geçebilir.

Merkezi olmayan kontrol sistemi yaklaşımında görevlerin robotlar arasında dağıtılması süreci yer almaktadır. Bu tip konfigürasyonda, görevleri diğer robotlara tahsis eden merkezi bir birim yoktur. Her robot, bilgilerini diğer robotlara iletir. Her robot, diğer robotları dikkate almadan kendi başına çalışabilir. Ayrıca, bazen merkezi olmayan sistemin bir robotunun, görevini diğer robotlarla verimli ve uyumlu bir şekilde yerine getirmek için diğer robotlarla bilgi alışverişi yapması gerekir.

Bu tez çalışmasında merkezi yaklaşımlı işbirliği sistemi benimsenmiş ve uygulanmıştır. Merkezi kontrol sisteminin merkezi olmayan yaklaşıma göre avantajları şu şekildedir:

- Merkezi olmayan yaklaşımda karar verici sistem merkezi bir mekanizma olmadığı için robotların hesaplama niteliklerinin üstün olması beklenir. Ancak merkezi kontrol sistemi yaklaşımında hesaplama birimi bir robot olmadığı için robotların yüksek işlem kapasitesi gerektirecek donanıma ihtiyacı olmayacaktır.
- Merkezi olmayan yaklaşımda robotlar arasındaki haberleşme sadece kullanılan iletişim teknolojisinin erişim sınırları arasında olmaktadır. Bu durumda sistem belirli bir bölge içinde çalışacaktır. Ancak merkezi kontrol yaklaşımında mekândan bağımsız iletişim gerçekleştiği için çok farklı lokasyonlardaki robotlar bile etkileşime girebilir.
- Merkezi olmayan yaklaşımda karar verme süreci karmaşık hale gelebilmektedir. Bu durumda çoğu zaman merkezi yaklaşıma göre optimal çözümler elde edilmeyebilir.

Çoklu robot görev dağılımı, uygulanan sistemlerin çok değişkenli ve ortam değişkenlerinin çok farklılık göstermesinden dolayı özellikli çözümlere sahip sabit algoritmalar bulunmamaktadır. Bununla birlikte problemin çözümü için geliştirilen algoritmik yaklaşımlar belirgindir. Bu yaklaşımları piyasa-tabanlı algoritmalar ve optimizasyon tabanlı algoritmalar olarak değerlendirebiliriz (Badreldin, Hussein, & Khamis, 2013).

Piyasa temelli çoklu robot koordinasyon yaklaşımları, son yıllarda robotik araştırma camiasında kayda değer bir ilgi gördü. Bu yaklaşım araçların alım satım için belirli görevlere olan ilgilerini ölçme becerisini temsil edebilen fayda fonksiyonları kavramına odaklanır. Çok robotlu koordinasyon için birçok piyasa temelli yaklaşım geliştirilmiştir (Gerkey & Mataric, 2002; Ljesnjanin & Velagic, 2009).

Optimizasyon, bir dizi mevcut çözümden bu problem için en uygun çözümü bulmaya odaklanan uygulamalı matematiğin dalıdır. Diğer bir deyişle, belirli bir problemin kârını artırmak (maksimizasyon problemi) veya zararını azaltmak (minimizasyon problemi) için optimizasyon teknikleri uygulanır. Çoklu robot görev dağılımı problemi için literatürdeki mevcut çözümler, bir dizi kısıtlama ile sınırlandırılmıştır ve optimum çözüm, belirli kriterlere göre bu kısıtlı çözümler içinde seçilir. Bu kriterler, problemin amaç fonksiyonunu tanımlar; burada amaç fonksiyonu, sistemin amacını tanımlamak için bazı değişkenleri birleştiren matematiksel bir ifadedir. Çok çeşitli optimizasyon yaklaşımları mevcuttur ve bu yaklaşımların kullanımı, optimize edilecek sorunun doğasına ve karmaşıklık derecesine bağlıdır.

### **3.4 Robot Hareket Modelleme ve Robot İşletim Sistemi (ROS)**

Robotların konum tespiti, yol planlaması ve çoklu görev dağılımı aşamasından sonraki temel problemlerinden biri de hareket koordinasyonunun sağlanmasıdır. Herhangi bir robota tahsis edilen görev neticesinde belirlenen yol planı üzerinde hareket etmesini sağlayacak bir hareket modelinin üretilmesi gerekmektedir. Bu hareket modeli robotun ileri-geri hareket etmesini ve belirli açılarla sağa sola dönmesini sağlayacak teker hareketlerini kapsamaktadır.

Birçok lojistik robotu üzerinde bulunan tekerler sayesinde hareketlilik kazansa da nadiren de olsa farklı mekanizmalarla hareket eden robotlar da bulunmaktadır. Robot üreticileri farklı sistemlerle entegre olabilmek için kendilerine has protokollerle erişim sağlamaktadırlar. Bu durumda lojistik yönetim sistemine entegre edilecek her farklı türde robot için erişim ayarlarının ayrı ayrı konfigüre edilmesi ve hatta farklı kodlamalarla

kontrollerinin sağlanması gerekliliği ortaya çıkmaktadır. Bu durumda tüm robotların aynı çerçevede yönetilebildiği ROS gibi bir yapıya ihtiyaç vardır.

ROS, geniş kütüphaneleriyle robotların tüm fonksiyonlarını çalıştırabilen ve kolay bir arayüz sağlayan bir ara katman yazılımıdır (Estefo, Simmonds, Robbes, & Fabry, 2019; Q. Gao & Cheng, 2018; Giubilato, Chiodini, Pertile, & Debei, 2019). İçeriğinde bulunan çok sayıda sensör sürücüsü sayesinde tüm robotlarla etkileşebilir. C++, Lisp, Python, Lua, Java gibi birçok yazılım dili için desteği bulunmaktadır. Rviz ve Gazebo gibi harita ve robot simülasyon ortamları ile tam uyumlu çalışarak gerçek zamanlı çalışma öncesi test olanağı sunmaktadır.

ROS'un, içerisinde bulundurduğu kütüphanelerin ve desteklediği robotların yanı sıra geliştiriciler tarafından yazılan kütüphaneler ve sürücülerle, desteklediği robot sayısı günden güne artmaktadır ve artık bir standart haline gelmektedir.

Genellikle Linux işletim sistemi içinde kullanılan ROS'un robotu yönetebilmesi için bu işletim sistemi ile çalıştırılması gerekmektedir. Ancak robot üreticileri çoğunlukla donanımsal özelliklerine göre çalışan minimal düzeyde gömülü işletim sistemlerini tercih etmektedirler. Bu durumda bir bilgisayar kontrolü ile robotların yönetilmesi gerekmektedir. Fakat bir bilgisayara bağlı şekilde robotların bağımsız bir şekilde hareket etmesi söz konusu değildir. Ancak tek-kartlı bilgisayarlar (single-board computers) kullanarak bu durum çözülebilir. Günümüzde en yaygın tek-kartlı bilgisayar modellerinden olan Raspberry Pi veya Nvidia Jetson bu amaç için ideal kullanıma sahiptir. Bu cihazlar Linux da dahil olmak üzere birçok işletim sistemini desteklemektedir ve bir bilgisayarın sahip olduğu bütün özelliklere sahip olabilmektedirler. Aynı zamanda Wi-Fi ve Bluetooth gibi erişim cihazlarına da sahip oldukları için internet veya network bağlantısı üzerinden bağlantı kurulabilir ve robotu uzaktan yönetme imkânını sunabilmektedirler.

### **3.5 Otonom ürün/malzeme yükleme bırakma**

Lojistik süreçlerinde önemli bir aşama olan ürün/malzeme yükleme ve bırakma işlemleri insan odaklı otonom olmayan sistemlerde insanlar tarafından yürütülmektedir. Otonom lojistiğin amaçlarından biri iş sağlığı ve güvenliğini sağlamak olduğundan ürün yükleme ve bırakma işlemlerinin de otonom yapılması gerekmektedir. Ürün yükleme ve bırakma mekanizması lojistik sistemlerde farklılık gösterebilmektedir. Vaka örneğine bakıldığında robotların dok arabalarına kenetlenmeden taşıma işlemini gerçekleştiremeyeceği

görülmektedir. Verilen örnekte otonom robotlarla taşımacılık yapılacaksa robotun dok arabalarında bulunan bağlantı yuvasına kenetlenmesini sağlayacak bir mekanizmaya ve bu mekanizmayı yönetecek bir yazılımsal sisteme ihtiyaç vardır. Bu bağlantı yuvası ve kenetlenme mekanizması Şekil 2.5'te gösterilmiştir. Başlangıç noktalarında, makinelerin çıkış portunda veya bekleme istasyonlarında bekleyen dok arabalarının alınması işlemlerinde otonom yükleme sistemi çalıştırılabilir.

Robotun taşımakta olduğu bir dok arabasını teslim edilecek noktaya uygun bir şekilde bırakması için de otonom bir park etme sistemine ihtiyacı olduğu görülmektedir. Otonom park sistemi vaka örneğindeki dok arabalarını bekleme istasyonlarına, makinelerin giriş portlarına veya teslimat noktalarına bırakmasını kapsamaktadır.

Lojistik süreçleri vaka örneğine benzer yapıda olan diğer sistemlerde de ürün yükleme ve bırakma aşamaları yukarıda belirtildiği gibi çalışır. Ancak farklı yükleme ve bırakma mekanizmalarına sahip sistemlerde bu aşama farklı şekilde ele alınmalıdır. Her sistem kendine has lojistik unsurlara sahip olabileceği için bu durumda probleme özgü çözüm yaklaşımları geliştirilmelidir.

#### 4. MATERYAL VE METOT

Tez çalışması bir Üniversite-Sanayi İşbirliği Projesi (ÜSİP) ile desteklenmiştir. Projede işbirliği yapılan Çalık Denim A.Ş.'nin bünyesinde bulunan kumaş terbiye işletmesi bina içi lojistik faaliyetlerinin otonom olarak yürütülmesi hedeflenmiştir. İşletme, bölüm 2.3'de verilen vaka örneğinde belirtilen ortam sabitlerine ve değişkenlerine sahiptir. Geliştirilen bulut tabanlı otonom çoklu robot yönetim sistemi anlatılırken, bu işletmenin lojistik faaliyetlerindeki etken unsurlar baz alınmıştır. Her ne kadar kumaş terbiye işletmesinde uygulanmış olsa da, geliştirilen sistem benzer bina içi lojistik faaliyetlere sahip işletmelerde, sektör ve çalışma ortamına uyarlanmak kaydıyla, uygulanma potansiyeline sahiptir.

Bu bölümde öncelikle sistem materyalleri belirlenmiş ve otonom lojistik sisteminin tasarımı yapılarak bir çalışma modeli oluşturulmuştur. Daha sonra bu materyallerin, oluşturulan ortam modeline göre yönetilmesini sağlayan iki aşamalı yönetim sistemi sunulmuştur. Yönetim sisteminin ilk aşaması, robotların koordinasyonunu ve yönetsel işlemleri sağlayan bulut sistemdir. İkinci aşama ise bulut sistemden gelen emirlerin robotlar tarafından uygulanmasını sağlayan robot kontrol sistemidir. İki sistem arasındaki ilişki Şekil 4.3'te gösterilmiştir.

##### 4.1 Sistem Materyalleri ve Çalışma Modeli

Bölüm 2.3'de vaka örneği olarak detaylıca anlatılan, denim üretim endüstrisindeki kumaş terbiye işletmeleri bina içi lojistik faaliyetlerinin otonom olarak yürütülmesi için öncelikle tesis içerisinde lojistik faaliyetlerine doğrudan etki eden varlıklar belirlenmiştir. Bu bölümde bu etken unsurların otonom olarak yönetilmesi için bir çalışma modeli oluşturulmuştur. Otonom sistemin karar mekanizması bu modele göre belirlenmektedir.

Vaka örneğinde belirtilen ve Çalık denim bünyesinde yer alan terbiye işletmesinin 2 boyutlu gerçek krokisi firmadan alınmış ve ve tez kapsamında geliştirilen otonom lojistik uygulama yazılımı için modellenmiştir. Mevcut çalışma modeline AGV'lerin dâhil edilmesiyle birlikte yeni varlıklar da sisteme eklenmiştir. Bu yeni varlıklar AGV'nin enerji sorunu yaşamadan çalışmasını sağlayan şarj istasyonları ve sistemin kararlı bir şekilde çalışmasını sağlayan başlangıç ve teslimat portlarıdır. Tüm bu unsurlar haritalama modülünde sisteme bir girdi olarak kaydedilmektedir. Girilen verilere göre harita ortamı şekillenmektedir. Her varlığın haritada kapladığı gerçek alan da sistemde tanımlanabilmektedir. Çalışma modelinde belirlenen otonom lojistik varlıkları şunlardır:

- $K_i$ : Sipariş üzerine işlem gören ürünler (yarı mamul kumaşlar),
- $M_i$ : Ürünlerin belirli işlemlere tabi tutulduğu terbiye makineleri,
- $T_i$ : Ürünleri taşıyan yük araçları (dok arabaları),
- $W_i$ : Dok arabalarının bekleme istasyonu,
- $AGV_i$ : Dok arabalarını çeken otomatik güdümlü araçlar,
- $C_i$ : AGV şarj istasyonları,
- $S_i$ : Ürünlerin haritaya (tesise) giriş yaptığı başlangıç portları,
- $D_i$ : Ürünlerin teslim edildiği teslimat portları.

Terbiye makineleri, yarı mamul kumaşın satışa hazır hale gelene kadar gördüğü kimyasal işlemlerin yapıldığı cihazlardır. Sipariş türüne göre her ürünün terbiye makinelerinde gördüğü birim işlem süresi (1 metre kumaşın makinede işleme süresi) farklılık göstermektedir. Terbiye makinesinde geçirilen sürenin miktarında, ürünün uzunluğu ve birim işlem süresi etkilidir.

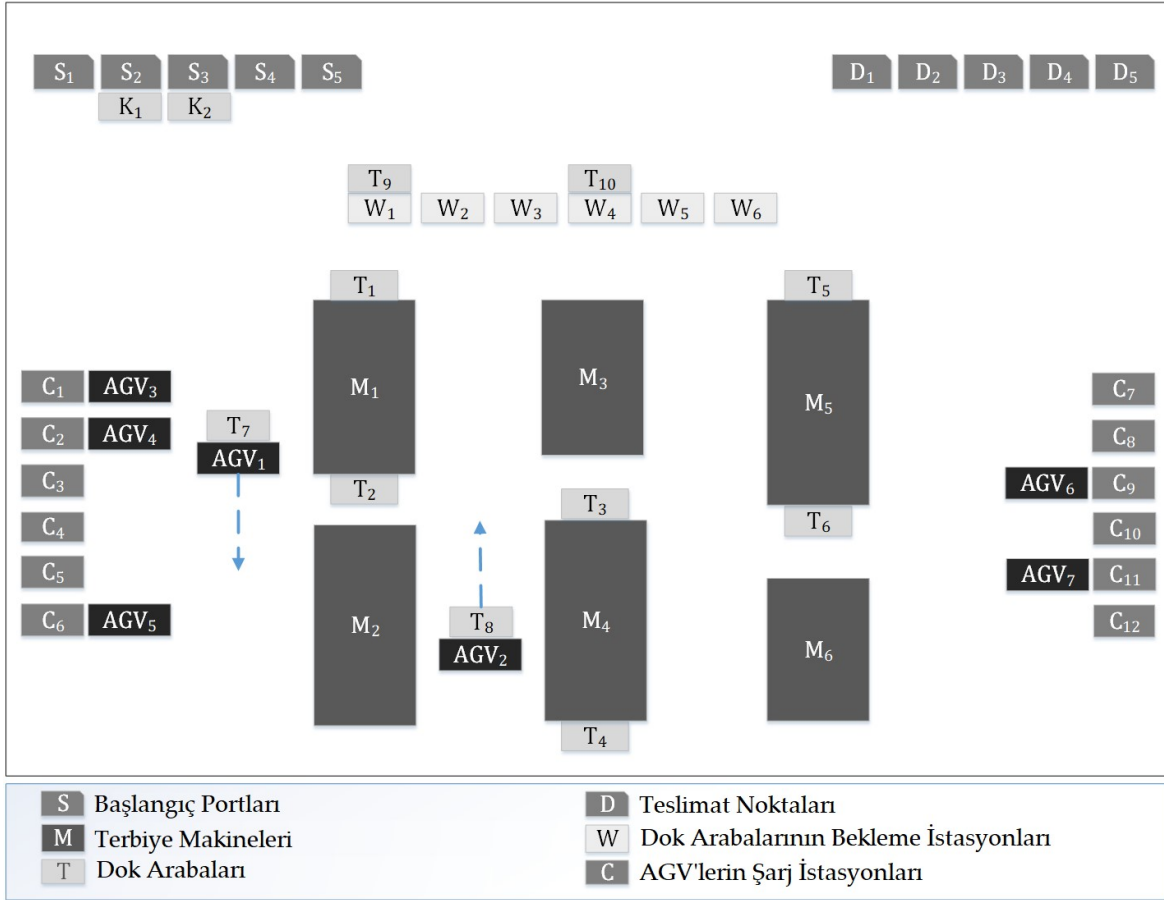
Bölüm 2.3'de belirtilen, kumaşların yüklendiği ve taşıma römorku olarak görev alan dok arabaları, kumaşları bulunduğu noktadan bir sonraki hedefine taşımak için yük arabası olarak kullanılır. Eğer bir kumaşın işlem göreceği makine meşgulse, kumaşı taşıyan dok arabası bekleme istasyonuna götürülür. Burada sipariş önceliğine göre kuyrukta bekleyen kumaşları taşıyan dok arabaları, makine müsait duruma geldiğinde, işleme alınmak üzere tekrar hareket ettirilir.

Tez çalışmasında dok çekerler yerine endüstriyel robotlar olan AGV'lerin kullanımı üzerine modelleme yapılmıştır. Kumaşın boyutuna göre dok arabalarındaki yük miktarı değişken olmakla birlikte maksimum bir sınırı vardır. Dok arabasını kaldıracak AGV'nin bu ağırlığı taşıyacak dayanıklılıkta olması gerekmektedir.

AGV'lerin enerji ihtiyacını karşılaması için sahada şarj üniteleri kurgulanmıştır. Şarj ünitelerinin konumu sabitleştirilmiş ve böylece AGV'lerin şarj olması için doğru noktaya yönlendirilmesi sağlanmıştır.

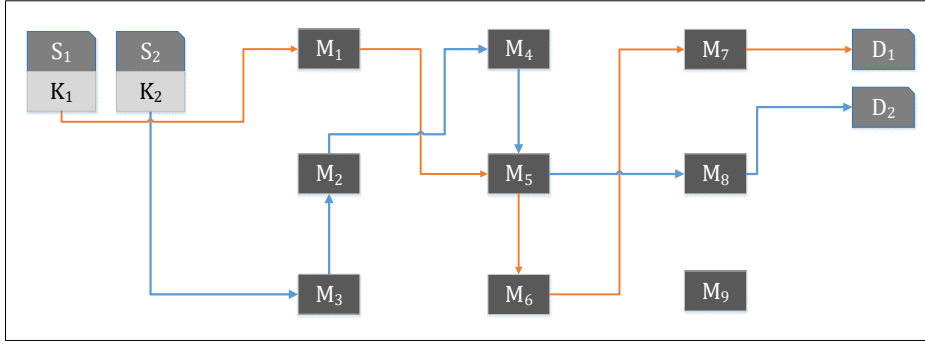
İşlenmek üzere terbiye işletmesine alınan kumaşların sisteme dâhil edilmesi için çalışma alanında başlangıç portları tasarlanmıştır. Aynı şekilde teslimat noktaları da sabit konumlarda yerleştirilmiştir. Başlangıç ve teslimat portlarının sabit bir konumda olması, kumaşın yereldeki konumunun bilinmesi ve bu sayede sistemin istikrarlı çalışması açısından

önemlidir. Şekil 4.1’de çalışma modeli için tasarlanmış olan örnek bir haritadan alınmış anlık görüntü simüle edilmiştir.



**Şekil 4.1 :** Otonom Lojistik için modellenmiş kumaş terbiye işletmesi.

Şekil 4.1’de K<sub>1</sub> ve K<sub>2</sub>, başlangıç portunda işlenmek üzere bekleyen yeni kumaşlardır. T<sub>1</sub>, T<sub>3</sub> ve T<sub>5</sub>, terbiye makinelerinin giriş portunda kumaşı boşaltan dok arabalardır. T<sub>2</sub>, T<sub>4</sub> ve T<sub>5</sub>, kumaşı makinelerin çıkış portundan yükleyen dok arabalarıdır. T<sub>7</sub> ve T<sub>8</sub>, AGV’ler tarafından varış yerine götürülen dok arabalarıdır. T<sub>9</sub> ve T<sub>10</sub>, bekleme istasyonundaki kumaş yüklü veya boş dok arabalarıdır. AGV<sub>1</sub> ve AGV<sub>2</sub>, dok arabalarını hedefe götürürken, diğer AGV’ler şarj olmaktadır. Bu AGV’ler aynı zamanda şarj durumuna göre göreve hazır olarak bekletilmektedir. Görevi biten bir robot başka bir görevi yoksa en yakın şarj ünitesine yerleştirilmektedir. Bir kumaş sisteme girdiğinde teslimat noktasına ulaşana kadar hangi terbiye makinelerinden hangi sırayla geçeceği tanımlanmaktadır. Geliştirilen otonom sistem bu işlemlerin gerçekleştirilmesi için optimum görev ataması yaparak sırasıyla yerine getirmektedir. K<sub>1</sub> ve K<sub>2</sub> kumaşlarının teslimat noktasına ulaşana dek işlemde geçeceği terbiye makineleri ve sıralamasını gösteren örnek senaryosu Şekil 4.2’de simüle edilmiştir.



**Şekil 4.2 :** İki farklı kumaşın 9 terbiye makinesinin bulunduğu ortamda taşınma senaryosu.

Şekil 4.2’de gösterilen durum örnek olarak ele alındığında, 9 farklı terbiye makinesinin olduğu bir ortamda iki farklı kumaşın teslimat noktasına iletimine kadar geçirdiği işlem trafiği görülmektedir. Bu şekle göre S<sub>1</sub> noktasında işlem sürecine başlayan birinci kumaşın sırasıyla M<sub>1</sub>, M<sub>5</sub>, M<sub>6</sub> ve M<sub>7</sub> numaralı iş makinelerinde işlemden geçmesi ve D<sub>1</sub> teslimat noktasına ulaştırılması gerekmektedir. İkinci kumaşın da sırasıyla M<sub>3</sub>, M<sub>2</sub>, M<sub>4</sub>, M<sub>5</sub> ve M<sub>8</sub> terbiye makinelerinde işlemden geçirilerek D<sub>2</sub> teslimat noktasına iletilmesi gerekiyor. Kumaşların işleme sokulması senkron veya asenkron olarak gerçekleşebilmektedir. Sistem ürünleri transfer etmek için emir verdiğinde uygun durumda olan AGV yoksa işlem bekletilmektedir. İşlemler devam ederken sahadaki AGV’lerin şarj durumlarının da kontrol edilmesi sağlanmaktadır. Şarj durumu kritik seviyede olan AGV’ler şarj istasyonlarına yönlendirilmektedir. Bu durumdaki AGV’lere görev ataması yapılmamaktadır. Eğer yapması gereken görevi tamamlayamadan şarjı kritik seviyeye inen bir AGV olursa, bu durumda o görev için başka bir AGV görevlendirilmektedir.

#### 4.2 Geliştirilen Bulut Sistem Mimarisi

Tez çalışması cihazdan bağımsız olarak yürütülen ancak etkileşim içinde çalışan iki farklı yazılım mimarisine sahiptir:

- 1) Bulut sunucu yazılımı
- 2) Robot kontrol yazılımı.

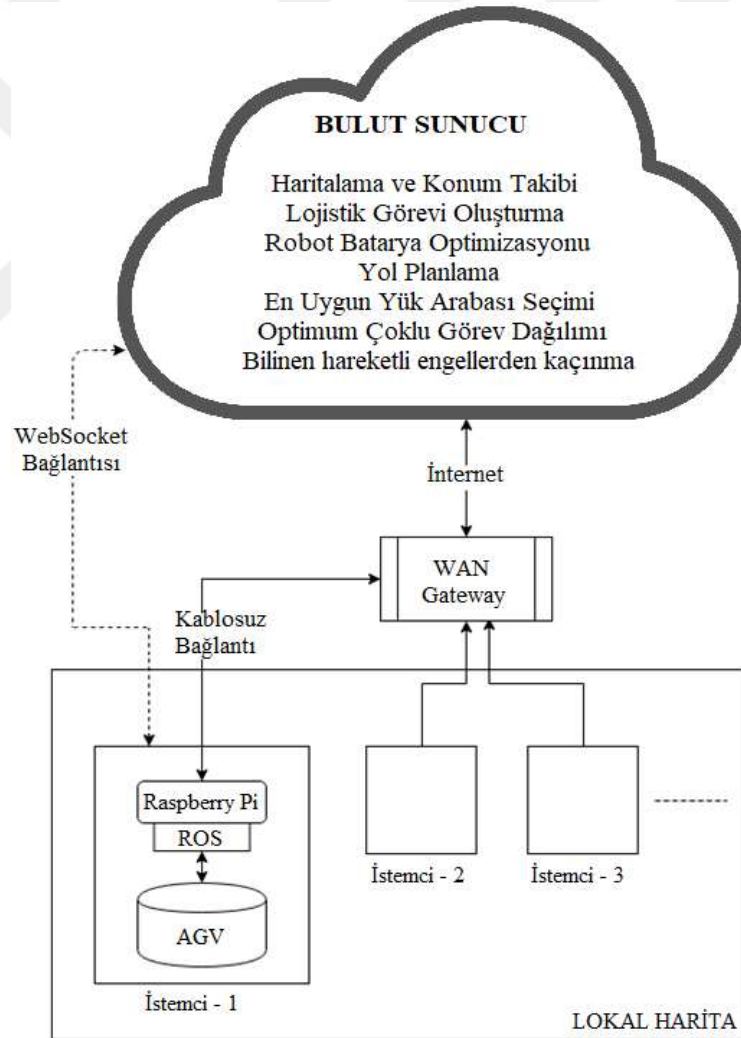
Bu bölümde işletme içi lojistik yönetimine etki eden varlıkların etkileşimi için merkezi kontrol yaklaşımını esas alarak geliştirilen bulut sistem yazılımı detaylıca anlatılmıştır.

Bulut sistem, Python dilinde Django altyapısı kullanılarak geliştirilmiştir. Sistemde oluşan veriler ilişkili tablolarda PostgreSQL veritabanı yönetim sistemi kullanılarak kaydedilmiştir.

PostgreSQL ücretsiz ve açık kaynak kodlu olması ve güçlü bir altyapıya sahip olması sebebiyle tercih edilmiştir. Bulut sistem aşağıdaki modüllerinden oluşmaktadır:

- Haritalama ve konum takibi,
- Lojistik görevi oluşturma,
- Batarya optimizasyonu,
- Yol planlama,
- Optimum çoklu görev dağılımı
- Hareketli engellerden kaçınma

Sistemin mimarisi Şekil 4.3'te gösterilmiştir.



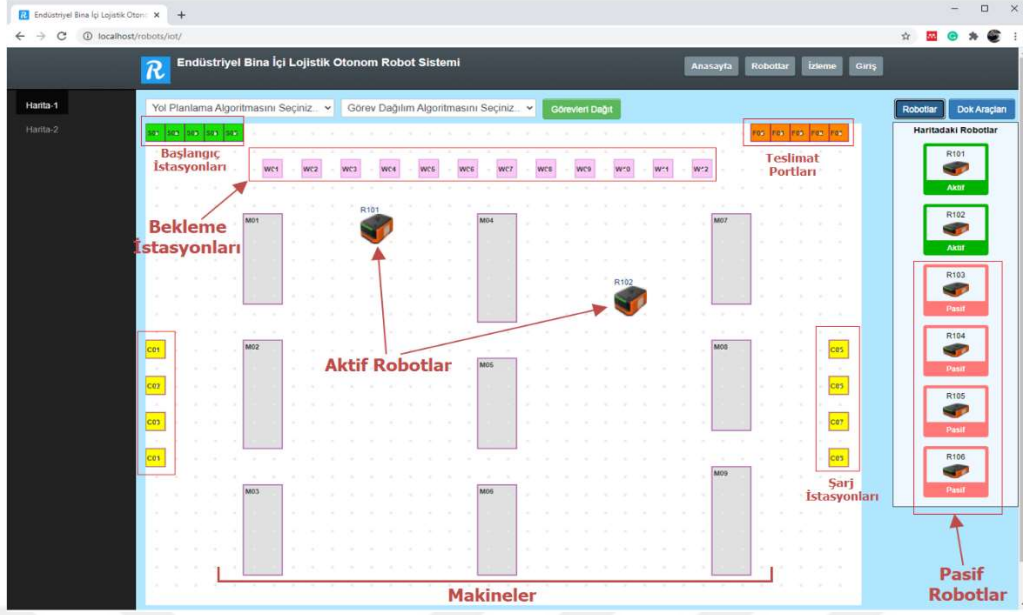
Şekil 4.3 : Bulut tabanlı otonom mobil robot yönetim sistemi mimarisi.

#### 4.2.1 Haritalama ve konum takibi

Haritalama aşamasında, robotun bulunduğu ortamın iki boyutlu düzlemde modellenmesi sağlanmıştır. Hareket halinde olan robotun modellenmiş harita üzerinde takip edilmesi için de robottan elde edilen konum bilgilerinin harita verilerine dönüştürülmesi sağlanmıştır. Ortam haritası ve harita içinde bulunan nesnelere hem bulut sistem için geliştirilen web arayüz ile hem de RESTful web servisleri ile farklı yazılımlar ile iletişim kurularak oluşturulabilmektedir. Konum takibi ise robotlar ile sunucu arasında kurulan WebSocket'ler aracılığıyla yapılmaktadır. Bulut sunucu üzerinde oluşturulan harita ortamında modellenen nesnelere şu tablolarda kaydedilmektedir:

- “*Robots*”: Ortamdaki robotların kaydedildiği veritabanı tablosudur. Her bir robota benzersiz bir kod verilir. Bu kod aynı zamanda robot üzerinde çalışan robot kontrol sistemine parametre olarak girilir.
- “*Machines*”: Ürünlerin işleneceği makinelerin yer aldığı veritabanı tablosudur. Her terbiye makinesi haritadaki konumu ve iki boyutlu düzlemdeki ebatları ile kaydedilir.
- “*WaitingStations*”: Ürünlerin (yük arabası ile üzerinde) bekletildiği istasyonların kaydedildiği veritabanı tablosudur. Her bir bekleme istasyonunun benzersiz bir kodu bulunmaktadır ve bir konumu vardır. İki boyutlu düzlemdeki ebatları da dinamik olarak girilebilmektedir.
- “*ChargingStation*”: Robotlar için şarj ünitelerinin kaydedildiği veritabanı tablosudur. Tabloda benzersiz bir kod ve harita üzerindeki konumları yer almaktadır.
- “*StartingPorts*”: Ürünlerin sisteme ilk girişinde konumlandırıldıkları başlangıç noktalarının kaydedildiği veritabanı tablosudur. Bu tablodaki kayıtlar benzersiz bir kod numarasına ve haritada yerleştirildikleri konum bilgisine sahiptir.
- “*DeliveryPorts*”: Ürünlerin teslim edildiği noktaların kaydedildiği veritabanı tablosudur. Benzersiz bir kod numarasına ve haritadaki konum bilgisine sahiptir.

Bulut yazılım sistemi üzerinde oluşturulmuş örnek bir ortam haritası görüntüsü Şekil 4.4'te gösterilmiştir. Bu örnek haritaya kayıtlı olan 6 adet AGV'den 2 tanesi aktiftir ve konumları takip edilmektedir. Haritada aynı zamanda 5 adet başlangıç portu, 5 adet teslimat portu, 9 adet terbiye makinesi, 12 adet bekleme istasyonu ve 8 adet şarj istasyonu olduğu görülmektedir.



Şekil 4.4 : Modellenmiş bir ortam haritası örneği.

#### 4.2.1.1 RESTful Web Servisler

Önerilen bulut tabanlı çoklu endüstriyel robot yönetim sistemi kendi içinde bilgilerin kaydedilmesini sağlayacak ara yüzlere sahip olmasının yanı sıra işletme içinde çalışan diğer yazılımsal uygulamalara da entegre edilebilir şekilde çalışmaktadır. Farklı platformlardan, taşınacak ürünlerin ve uygulanacak işlemlerin sırasıyla birlikte sisteme kaydının otomatik olarak yapılmasını sağlayacak web servisler bu aşamada gerçekleştirilmektedir. Ayrıca işletme içindeki sabit nesnelere de konum ve ebatlarının sisteme kaydedilmesini, güncellenmesini sağlayacak web servis yöntemleri de geliştirilmiştir. Web servisler, farklı sunucularda veya platformlarda çalışan birbirinden bağımsız yazılım uygulamalarının birbirleri arasında iletişim kurmasını sağlamaktadır. İki ana web servis protokolü vardır: 1) Basit Nesne Erişim Protokolü (SOAP) 2) Temsili Durum Transferi (REST). SOAP, HTTP ve SMTP gibi belirli bir protokol üzerinden XML tabanlı bir standart iletişim belirtimidir. REST ise XML ve JSON gibi esnek mesaj formatlarına sahip web tabanlı bir mimari stildir. REST, belirli istemci-sunucu iletişimini, tek tip arabirime ve kaynak önbelleğe aldığı için oldukça hızlıdır. RESTful çerçevesi, HTTP protokolüne dayalı REST mimarisinin bir uygulamasıdır. Daha fazla ölçeklenebilirlik, uyumluluk, performans ve basitlik, REST standardını seçmenin yaygın nedenlerindedir (Fard & Sarjoughian, 2021). Sistemde oluşturulan RESTful web servis metodları şöyledir:

“GetMapList”: Veritabanına çoklu harita kayıt edilebilmektedir. Bu metod harita bilgilerinin (adı, kodu, boyutu vb.) liste halinde gönderilmesini sağlamaktadır.

“GetMap”: Harita tablosunda birincil anahtar olan “Id” alanının parametre olarak almakta ve ilgili haritaya ait aşağıdaki bilgiler gönderilmektedir:

- *size*: Haritanın boyut bilgileri (haritanın kuşbakışı görünüşünün enlemsel ve boylamsal uzunluğu)
- *machines*: Ürünlerin işlendiği makineler ve her bir makinenin türü, adı, haritadaki konum bilgisi, harita düzlemindeki ebatı, birim zamanda işledikleri kumaş uzunluğu bilgisi,
- *waitingstations*: Bekleme istasyonları ve bu bekleme istasyonlarının harita düzlemindeki konumu, ebatı, istasyon kodu,
- *chargingstations*: Şarj istasyonları ve bu şarj istasyonlarının harita düzlemindeki konumu, istasyon kodu,
- *startingports*: Başlangıç portları ve bu başlangıç portlarının harita düzlemindeki konumu, kodu,
- *deliveryports*: Teslimat portları ve bu teslimat portlarının harita düzlemindeki konumu, kodu.

“AddObject”: Haritaya işlenmek veya teslim edilmek üzere yeni bir ürün/malzeme eklemek için bu metot kullanılmaktadır. Ürün veya malzeme eklemek için aşağıdaki veri seti gönderilmektedir:

- *code*: Ürün/malzeme için belirlenen ve sistemde tanınmasını sağlayan benzersiz kod.
- *startingport*: ürünün sisteme giriş yaptığı port.
- *machines*: ürünün işlemde geçeceği makineler. Bu bilgi liste halinde gönderilir ve gönderim sırası aynı zamanda ürünün işlem sırasıdır.

“UpdateMap”: Haritada herhangi bir veri değiştiğinde, örneğin yeni bir sabit nesne eklendiğinde veya var olan sabit nesneye ait bilgiler değiştiğinde bu metot çalıştırılır. Değişiklikler toplu halde gönderilebilmektedir.

#### 4.2.1.2 WebSocket Metotları

Robot kontrol ve bulut yazılımı farklı cihazlarda çalışmaları için aralarında bilgi aktarımı yapılması ve bu bilgi aktarımının anlık olarak yapılması gerekmektedir. Ayrıca aktarımın otomatik olarak cihazlar tarafından fark edilir olması gerekmektedir. Web servis metotları,

bilgi aktarımı olup olmadığının kontrol edilmesi için sürekli sorgulama yapılması gerektiğinden bu işlem için yetersizdir. Bu işlem için en uygun teknoloji bilgi aktarımında olay tetikleyicisi mekanizması olan ve internet/network üzerinden çalıştırılabilen WebSocket teknolojisidir. WebSocket'ler, istemciler ve sunucular arasında internet veya yerel ağ üzerinden iki yönlü iletişim sağlamaktadır. Bu teknoloji ile bir TCP socketi üzerinde iki yönlü iletişim kanalı kurulur ve bu kanal iki uç arasında kesintisiz bilgi alışverişi için açık kalır. Aynı kanala birden fazla istemci bağlanabilmektedir. Kanala bir mesaj gönderildiğinde bu mesaj socketin tüm uçlarına gönderilir ve böylece bilgi tüm cihazlarla anlık olarak paylaşılır. WebSocket'te anlık mesajlaşmayı sağlayan dört ana olay tetikleyicisi şöyledir:

“*OnError*”:

İletişim kurulmaya çalışıldığında veya kurulduktan sonra herhangi bir hata alındığında bu metod otomatik olarak çalışır. Sistemde bu hata alındığı anda bağlantı kapatılıp yeniden kurulmaya çalışılır. Eğer hata alınmaya devam ederse sistem yöneticisine otomatik bildirim yapılmaktadır.

“*OnClose*”:

WebSocket kanalı kapatıldığında bu metod otomatik olarak çalışır. Robotun gücü kesildiği zaman (enerjisi tükenip kapandığında veya bilinçli olarak kapatıldığında) tetiklenir.

“*OnMessage*”:

Bağlantı kurulduktan sonra bir veri iletimi yapıldığında bu metod çalışır. Metod kanala bağlı tüm uçlarda otomatik olarak tetiklenir ve gönderilen mesaj parametre olarak alınır. Burada dikkat edilmesi gereken nokta veriyi gönderen kanalda da bu metodun çalışmasıdır. Robotlar konum bilgileri değiştiği zaman yeni konumlarını mesaj olarak sunucuya gönderir. Ayrıca robotların izleyecekleri yol planları da sunucular tarafından mesaj olarak gönderilir. Aynı kanalda iki bilginin de gönderiliyor olmasından doğacak bilgi karmaşıklığını önlemek için mesajın içeriğine gönderilen bilginin türü de eklenmektedir. Örneğin gönderilen mesaj konum bilgisi ise aşağıdaki gibi bir veri gönderilir:

```
1 {
2   "messageType": "Konum",
3   "message": "Son Konum || x:100 || y:100"
4 }
```

Eğer gönderilen bilgi yol planı ise mesaj tipi “Yol Planı” olarak gönderilir ve mesaj içeriğinde izlenecek konum bilgileri sıralı olarak (x,y) şeklinde gönderilir:

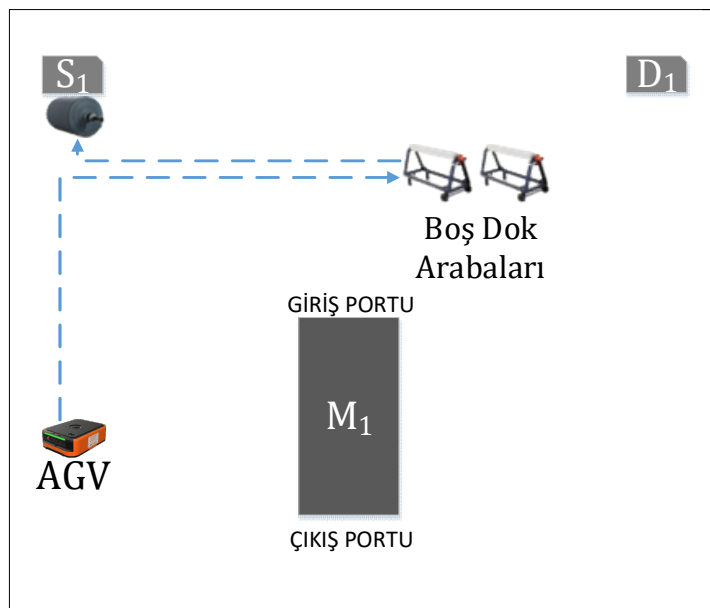
```
1 {
2   "messageType": "Yol Planı",
3   "message": "[ (100,101), (100,102), (101,102), ... ]"
4 }
```

“OnOpen”: WebSocket kanalı açıldığında bu metot otomatik olarak çalışır. Bir robot aktifleştirildiğinde WebSocket bağlantısı açılır ve bağlantı başarılıysa göreve hazır bir şekilde çalışmaya devam eder. Bağlantı gerçekleşmediğinde uyarı sinyali verir.

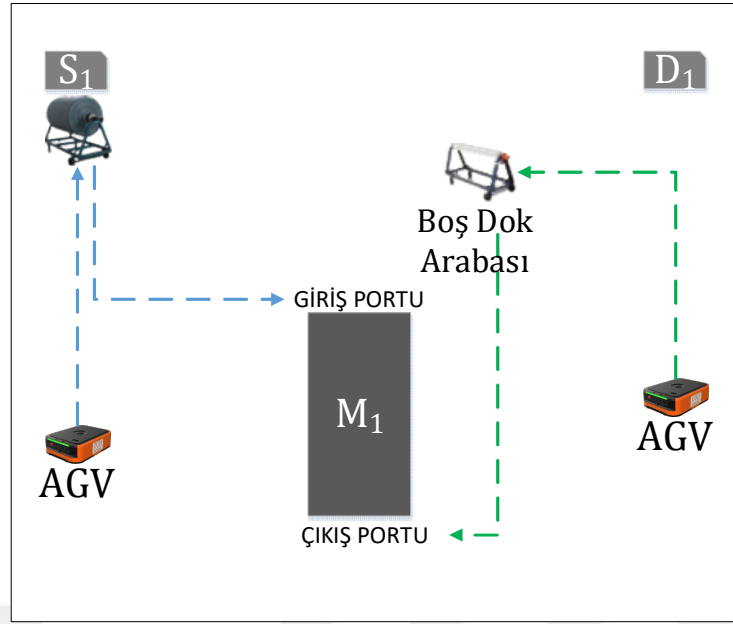
#### 4.2.2 Lojistik görevi oluşturma

Görev dağılımını doğru bir şekilde yapabilmek için görev türlerinin doğru bir şekilde tanımlanması gerekmektedir. Verimli bir çalışma ortamı için kumaşların makinelerde kaldığı süreçte AGV’lerin farklı görevleri ifa edebilmeleri sağlanmıştır. Dolayısıyla kumaşı teslimat noktasına ulaştıracak görev parçalarına bölünmüş ve her bir kumaş için birden fazla görev tanımı oluşmuştur. Bu görev tanımları aşağıdaki gibidir:

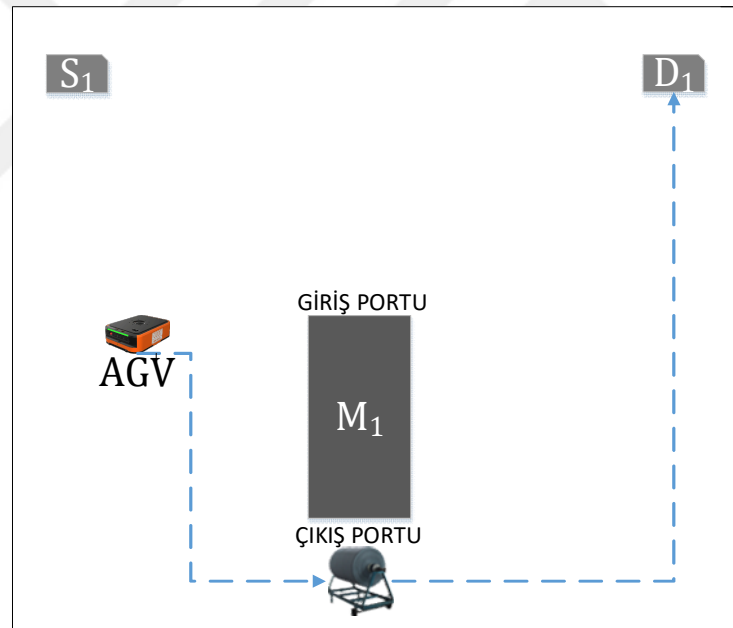
- Kumaş işlenmek için işletmeye girdiğinde, boş bir dok arabasının bu kumaşa iletilmesi: “*başlangıç görevi*”. Şekil 4.5’te örnek gösterim verilmiştir.
- Kumaşın işlem göreceği herhangi bir makinenin giriş portuna iletilmesi gerektiğinde oluşturulan yeni görev: “*giriş portu görevi*”. Şekil 4.6’da örnek gösterim verilmiştir.
- Kumaş terbiye makinesine girip işlemden geçtikten sonra makinenin çıkışında başka bir dok arabasına sarılmalıdır. Kumaş makinenin giriş portuna iletilirken aynı zamanda boş bir dok arabası da makinenin çıkış portuna iletilmesi görevi: “*çıkış portu görevi*”. Şekil 4.6’da örnek gösterim verilmiştir.
- Son makinede işlem bittikten sonra ürünün teslimat noktasına ulaştırılması: “*teslimat görevi*”. Şekil 4.7’de örnek gösterim verilmiştir.



Şekil 4.5 : Başlangıç görevi örneği.



Şekil 4.6 : Kumaşın  $M_1$  makinesinde işlenmesi için oluşturulan giriş portu görevi ve çıkış portu görevi örnekleri.



Şekil 4.7 : Teslimat görevi örneği.

#### 4.2.3 Robot batarya optimizasyonu

Bulut sisteme, AGV'lerin konumlarının yanı sıra, batarya doluluk oranları da robot kontrol sistemi tarafından gönderilmektedir. AGV'ler üzerlerine atanmış bir görevi ifa ederken bataryasındaki enerji miktarı kritik seviyenin ( $l_c$ ) altına düşerse görevi başka bir AGV'ye teslim ederek en yakın şarj istasyonuna yönlendirilmektedir. Çoklu robot görev dağılım algoritmasında önemli bir role sahip olan  $l_c$  parametresi, bir AGV'nin harita üzerinde

birbirine en uzak iki mesafe arasındaki yol planı ile oluşan mesafe boyunca üzerinde taşıyabileceği maksimum yük miktarı ile hesaplanmaktadır. Harita güncellendiğinde yeni engeller bu yolun uzunluğunu değiştirirse  $l_c$  seviyesi de değişir. Kritik batarya seviyesi Denklem 4.1'deki eşitlik kullanılarak hesaplanır.

$$l_c = \delta * P_{max} + \mu * P_{max} * W_{max} \quad (4.1)$$

Denklem 4.1'de,  $\delta$  bir AGV'nin 1 metrelik mesafede harcadığı enerji (joule) miktarıdır.  $\mu$  ise ağırlığın enerjiye etkisini göstermektedir ve bir AGV'nin 1 metre mesafede 1 kg ağırlığı taşımak için harcanması gereken enerji miktarını göstermektedir.  $W_{max}$ , bir dok arabasının taşıyabileceği maksimum ağırlıktır.  $\mu$  ve  $W_{max}$  değerleri sisteme parametrik olarak tanımlanmaktadır.  $P_{max}$ , harita üzerindeki maksimum uzunluklu yolu temsil eder ve Denklem 4.2 kullanılarak noktalar arası Öklid uzaklıklarının toplamıyla elde edilir.

$$P_{max} = \sum_{i=1}^{k-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \quad (4.2)$$

Denklem 4.2'de  $k$ , yol planlama algoritması ile elde edilen optimum yolun harita üzerindeki sıralı geçiş noktaları olan  $[(x, y)]$  dizisinin eleman sayısıdır.  $[(x, y)]$  dizisi, (1,1) noktasından haritanın maksimum genişlik ve maksimum uzunluk noktasına olan en uzun yolun planı ile oluşturulmaktadır. Yol planı oluşturulurken sahada bulunan iş makineleri, bekleme istasyonları, şarj istasyonları başlama ve teslimat portları birer engel olarak tanımlanmaktadır.

#### 4.2.4 Yol Planlama

Bulut sistem birden fazla yol planlama algoritmasını destekleyecek şekilde tasarlanmıştır ve parametrik olarak seçilebilir hale getirilmiştir. Sistem, A\*, D\*, Dijkstra, APF, PSO ve D\* Lite algoritmalarıyla gerçekleştirilen yol planlama yöntemlerini desteklemektedir. Ancak yapılan deneysel çalışmalar, D\* Lite algoritmasının karmaşık problemleri çözme konusunda diğerlerinden daha iyi çalıştığını göstermiştir. Yol planlama ile ilgili deneysel çalışma sonuçları ve karşılaştırmalar beşinci bölümde verilmiştir. D\* Lite algoritmasının sözde kodu Algoritma 4.1'de verilmiştir.

---

**Algoritma 4.1 : D\* Lite Algoritması**

---

```
function CalculateKey(s)
  return [ $\min(g(s), rhs(s)) + h(s_{start}, s) + k_m$ ;  $\min(g(s), rhs(s))$ ]

function UpdateVertex(u)
  if  $u \neq s_{goal}$  then  $rhs(u) \leftarrow \min_{s' \in Succ(u)} (g(s') + c(u, s'))$ 
  if  $u \in U$  then  $U.Remove(u)$ ;
  if  $g(s) = rhs(s)$  then  $U.Insert(u, CalculateKey(u))$ 

function ComputeShortestPath ()
  while  $U.TopKey() < CalculateKey(s_{start})$  or  $g(s_{start}) = rhs(s_{start})$ 
     $k_{old} \leftarrow U.TopKey()$ 
     $u \leftarrow U.Pop()$ 
    if ( $k_{old} < CalculateKey(u)$ ) then
       $U.Insert(u, CalculateKey(u))$ 
    else if ( $g(s) > rhs(s)$ ) then
       $g(s) \leftarrow rhs(s)$ 
      for all  $s \in Pred(u)$  do UpdateVertex(s)
    else
       $g(s) \leftarrow \infty$ 
      for all  $s \in Pred(u) \cup \{u\}$  do UpdateVertex(s)

function GetOptimumPath ()
   $s_{last} \leftarrow s_{start}$ ,  $U = \emptyset$ ,  $k_m = 0$ 

  for all  $s \in S$  do  $rhs(s) \leftarrow \infty$ ,  $g(s) \leftarrow \infty$ 
   $rhs(s_{goal}) \leftarrow 0$ 
   $U.Insert(s_{goal}, CalculateKey(s_{goal}))$ 
  path = ComputeShortestPath ()
  while ( $s_{start} = s_{goal}$ )
     $s_{start} \leftarrow \min_{s' \in Succ(s_{start})} (g(s') + c(s_{start}, s'))$ 
    Move to  $s_{start}$ 
    Scan graph for changed edge costs
    if any edge costs changed then
       $k_m \leftarrow k_m + h(s_{last}, s_{start})$ 
       $s_{last} \leftarrow s_{start}$ 
      for all directed edges  $(u, v)$  with changed edge costs do
        Update the edge cost  $c(u, v)$ 
        UpdateVertex(u)
      path = ComputeShortestPath ()
  return path
```

---

#### 4.2.5 Taşıma görevleri için en uygun yük arabasının seçimi

Vaka çalışmasında, kumaşların lojistik faaliyetlerinde dok arabalarının yük taşımada kullanıldığı belirtilmişti. Elbette sistem, sektör bağımsız çalıştığı için yük arabası kullanmayan sistemlere de uyarlanabilir. Bazı sistemlerde AGV'ler yük arabası

kullanmadan direkt olarak taşıyıcı olabilmektedir. Bu tip sistemlerde bu aşama yok sayılmaktadır. Vaka örneğinde ve benzer sistemlerde ise bu aşama dikkate alınmadan optimum görev dağılımı sağlıklı olarak gerçekleştirilemez.

Vaka örneğinden yola çıkarak; Bir göreve atanacak bir dok arabası seçilirken görevin başlayacağı konuma en yakın olan tercih edilmelidir. Ancak bu dok arabasının meşguliyet durumuna da bakılmalıdır. Seçilen dok arabasının meşgul olduğu durumda başka birini seçmenin zaman yönünden tasarruf sağlama ihtimali bulunmaktadır. Bu durumda meşguliyet süresinin hesaplanması gerekmektedir. Meşguliyet süresi  $t_m$ , Denklem 4.3'teki eşitlikle ve saniye cinsinden hesaplanmıştır.

$$t_m = w/h_m - (t_c - t_s) * h_m \quad (4.3)$$

Denklem 4.3'de  $w$  meşgul olan makinedeki işlemde geçen kumaşın uzunluğunu temsil etmektedir.  $h_m$ , ilgili terbiye makinesinde 1 saniyede işlemde geçen kumaş uzunluğudur.  $t_c$ , içinde bulunulan güncel zamanı,  $t_s$  ise makinenin meşgul olmaya başladığı zamanı temsil etmektedir. Meşguliyet süresi bir kumaşın işlem göreceği makineye götürülmesi durumunda da kullanılmıştır. Eğer makine meşguliyet süresi, görevin bitiş süresinden fazlaysa kumaş bekleme istasyonuna götürülmektedir. Dok arabasının göreve atanmasında yol yerine zaman optimizasyonu yapılmıştır.  $i$ . dok arabasının görev noktasına ulaşacağı süre olan  $T_i$ , Denklem 4.4'deki gibi hesaplanır.

$$T_i = Mesafe(s_l, s_e) * \frac{1}{v} + t_m + Mesafe(s_b, s_g) * \frac{1}{v} \quad (4.4)$$

Denklem 4.4'de  $Mesafe(s_l, s_e)$ , dok arabası bir görevi ifa ediyorsa bulunduğu noktadan görevin bitiş noktasına olan ve yol planlama algoritmasıyla hesaplanan optimum uzaklığını temsil etmektedir.  $v$ , AGV'nin hızını,  $t_m$  ise dok arabasının meşgul olduğu görevin terbiye makinesinde geçireceği süredir. Denklem 4.4'te  $s_b$ , dok arabasının başlangıç pozisyonudur ve dok arabası meşgulse  $s_b = s_e$  olur. Eğer dok arabası meşgul değilse  $Mesafe(s_l, s_e) = t_m = 0$  olacaktır.

$s_l$ , dok arabasının bulunduğu güncel konumdur.  $s_e$ , dok arabasının mevcut görev atanmadan önce ifa ettiği eski görevin bitiş noktasıdır. En uygun boş yük (dok) arabasının seçilim algoritmasının sözde kodu Algoritma 4.2'de verilmiştir.

---

**Algoritma 4.2** : En uygun yük arabasının bulunması

---

**function** FindMostSuitableVehicle () $V \leftarrow$  list of all vehicles $B_v \leftarrow \emptyset$  $B_p \leftarrow P_{max} * v$ **for all**  $v \in V$  **do** $time = 0$ **if**  $v$  is busy **then** $s_l \leftarrow position(v)$ // Calculate path from current position to the end point of task ( $s_{task}$ ) $time \leftarrow$  GetOptimumPath ( $s_l, s_e$ ) \* speed**if**  $v$  have a work station task **then**  $time \leftarrow time + t_m$  $s_b \leftarrow s_e$ **else** $s_b \leftarrow position(v)$  $time \leftarrow time +$  GetOptimumPath ( $s_b, s_g$ ) \* speed**if**  $B_p > time$  **then**  $B_v \leftarrow v, B_p \leftarrow time$ **return**  $B_v$ 

---

**4.2.6 Önerilen optimum çoklu robot görev dağılımı yöntemi**

AGV'lerin çoklu görev dağılımlarını en uygun (optimum) şekilde yapılabilmesi için, tüm görev atamalarının toplam maliyeti minimize edilmesi gerekmektedir. Bu maliyet, toplam yolu ve zamanı en aza indiren çoklu amaç fonksiyonu ile hesaplanır. Genellikle, yolu minimize etmek, çok görevli robot tahsisinde zaman maliyetini de minimize eder. Ancak bazı durumlarda yolu küçültmek süreyi uzatabilir. Örneğin, bir görev atanırken, en uygun pozisyondaki AGV o anda başka bir görevi yerine getiriyor olabilir. Bu nedenle bu çalışmada, hem toplam yolun hem de toplam sürenin minimize edildiği çoklu amaç fonksiyonu kullanılmıştır. Bu çoklu amaç fonksiyonu Denklem 4.5'de gösterilmiştir. Fonksiyon, yol ve zaman için dinamik olarak belirlenebilen önem katsayılarına ( $\lambda_{1,2}$ ) sahiptir.

$$f(R, G) = \lambda_1 * \sum_{i=1}^{a-1} Mesafe(s_{r_i}, s_{g_i}) + \lambda_2 * \sum_{i=1}^{a-1} T_i \quad (4.5)$$

Denklem 4.5'de,  $R = \{r_1, r_2, r_3, \dots, r_n\}$ , AGV dizisidir.  $G = \{g_1, g_2, g_3, \dots, g_m\}$  ise görevlerin başlangıç konumları dizisidir. Önem katsayılarının toplamı 1 olmalıdır ve sisteme parametrik olarak tanımlanabilmektedir.  $a$ , atanan görev sayısını,  $Mesafe(s_{r_i}, s_{g_i})$ , robot ve görevler arasında yol planlaması algoritmasıyla hesaplanan optimum yolun uzunluklarını,

$T_i$  ise Denklem 4.4'deki eşitlikle hesaplanan, AGV'nin hedefe varma süresini temsil etmektedir.

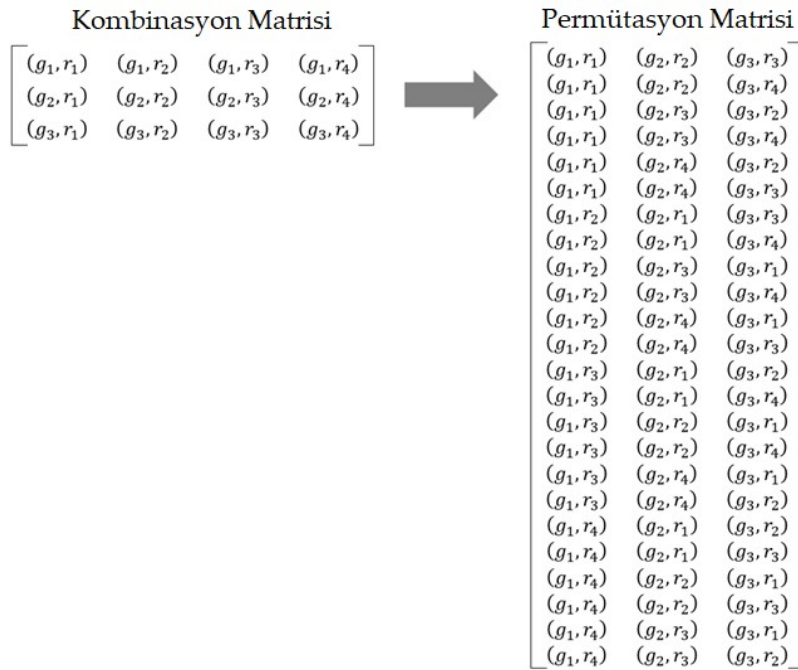
Çoklu AGV'lerin aynı anda birden çok göreve en uygun şekilde tahsis edilmesi için  $f(R, G)$  değeri minimum olmalıdır. Amaç fonksiyonu olan  $f(R, G)$ 'yi minimize etmek için AGV'lerin ve hedeflerin tüm olası eşleşmelerini içeren bir atama matrisi oluşturulmuştur. Bu matrisin amacı, uygun eşleşme satırını minimum maliyetle tespit etmektir.  $R$  ve  $G$  arasında eşleşmeler yapılmadan önce, bu iki küme aşağıdaki gibi  $C_{sol}$  ve  $C_{sağ}$  ara kümelerinde tutulmuştur:

$$R = \{r_1, r_2, r_3, \dots, r_n\}, \quad G = \{g_1, g_2, g_3, \dots, g_m\}$$

$$\text{Eğer } n \geq m \text{ ise; } C_{left} = R, \text{ ve } C_{right} = G$$

$$\text{Değilse; } C_{left} = G, \text{ and } C_{right} = R$$

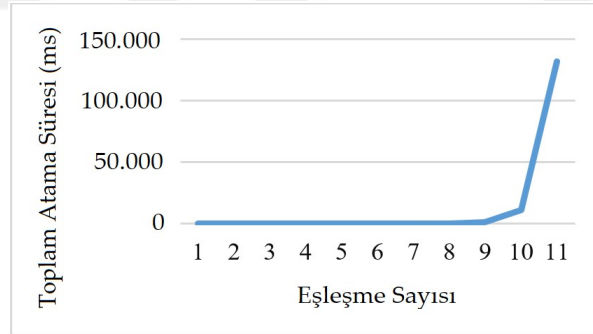
$C_{left}$  aynı anda yapılabilecek atamaların sayısını belirlediğinden, yukarıdaki işlemler gerçekleştirilmiştir. Bunun anlamı: AGV sayısı hedef sayısından düşükse, eşleştirme sayısı AGV sayısı kadar olabilir, aksi takdirde hedef sayısı kadar olabilir. Böylece, eşleşmenin solunda  $C_{left}$  ve sağında  $C_{right}$  olacak şekilde bir kombinasyon matrisi oluşturulmuştur. Ardından, her satırda benzersiz atamalara sahip tüm eşleşmeler kullanılarak bir permütasyon matrisi oluşturulmuştur. Şekil 4.8'de, örnek olarak verilen  $R = \{r_1, r_2, r_3, r_4\}$  ve  $G = \{g_1, g_2, g_3\}$  için permütasyon matrisinin oluşturulması gösterilmiştir.



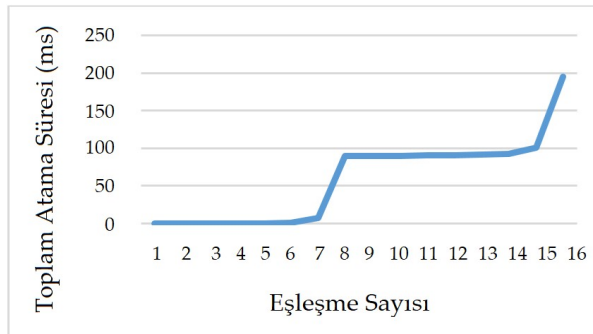
**Şekil 4.8 :**  $R$  ve  $G$  kümeleri için permütasyon matrisinin oluşturulması.

Tüm satırları benzersiz eşleşme kombinasyonlarına sahip olan permütasyon matrisi ile birlikte  $R$  ve  $G$  için tüm eşleşme olasılıkları da belirlenir. Böylece, her eşleşen yol uzunluklarının minimum toplamına sahip satır seçilebilir.

Yukarıdaki yöntemle her zaman en uygun maliyetli eşleşme satırının bulunması mümkün olsa da, AGV'lerin ve hedeflerin sayısı arttıkça, permütasyon matrisinin hesaplama karmaşıklığının artması problemi oluşabilmektedir. Bu problemi gidermek için robot ve hedef listeleri ön tanımlı bir  $\varphi$  değeri ile k-NN yöntemiyle sınıflandırıldı.  $\varphi$  değeri çok yüksek tutulduğunda, bilgisayarın hesaplama yeteneğine bağlı olarak atama süresi daha uzun olabilir. Bu değer çok düşük olduğunda optimum çözümden uzaklaşılabilir. Bu nedenle bu değer optimum düzeyde tutulması gerekmektedir.  $\varphi$  Farklı donanıma sahip bilgisayarlarda hesaplama süreleri değişken olduğundan bulut sisteminde dinamik olarak değer değiştirilebilmektedir. Şekil 4.9'da ve Şekil 4.10'daki grafiklerde görülebileceği üzere, atama süresi, AGV'ler ve hedefler sınıflandırılmadığında katlanarak artar. Ancak k-NN ile sınıflandırıldıklarında atama süresi çok daha kısadır. Şekil 4.9'da sınıflandırılmamış ve Şekil 4.10'da sınıflandırılmış permütasyon hesaplarının, eşleştirme sayısı gösterilmiştir. Burada, eşleşme sayısı  $n$  ise, atama süresi, ortamda  $n$  adet AGV ve  $n$  adet hedef ile hesaplanmış demektir.



Şekil 4.9 : Sınıflandırılmamış permütasyon hesabı ile atama süreleri.



Şekil 4.10 :  $\varphi = 8$  olduğu durumda k-NN ile sınıflandırılmış permütasyon hesabı ile elde edilen atama süreleri.

Atama matrisinde minimum toplam yola sahip satırı seçmek, optimum görev tahsisi için tek başına yeterli değildir. AGV'lerin eşleştikleri hedefe giden yolda harcayacakları enerjinin yeterli olup olmadığı da kontrol edilir. Atama matrisindeki herhangi bir satır, pil seviyesi eşleştiği mesafe için yeterli olmayan bir AGV'ye sahipse, bu satır atama matrisinden silinir. Böylelikle kalan satırlar arasından en az birinde arama yapılarak görev dağılımı gerçekleştirilir. Optimum çoklu görev dağılımının sözde kodu, Algoritma 4.3'te sunulmuştur.

---

**Algoritma 4.3 :** Optimum çoklu robot görev dağılımı algoritması

---

**function** getOptimumAllocation ()

$C \leftarrow \emptyset$ ,  $P \leftarrow \emptyset$ ,  $C_{left} \leftarrow \emptyset$ ,  $C_{right} \leftarrow \emptyset$

**if**  $length(R) > length(G)$  **then**  $C_{left} \leftarrow G$ ,  $C_{right} \leftarrow R$

**else**  $C_{left} \leftarrow R$ ,  $C_{right} \leftarrow G$

$D \leftarrow$  Create sub-matrices with length  $\varphi$  according to nearest neighborhoods

**for**  $d$  **in**  $D$  **do**

$C \leftarrow$  Create combination matrix from  $d_{left}$  and  $d_{right}$

$P_d \leftarrow$  Create permutation matrix from  $C$

$P.Append(P_d)$

Remove rows from  $P$  if energy level does not meet the cost

Calculate the total cost of each row of  $P$  by summing costs of each assignment in row.

Return first row with the lowest total cost in matrix  $P$

---

Algoritma 4.3'teki optimum çoklu robot görev dağılımı algoritması eş zamanlı olarak bekleyen görevlerin dağılımında kullanılmıştır. Görev tanımlarının yapılması ve bu görevlerin optimum görev dağılımı algoritması kullanılarak ifa edilmesi için bir işbirliği algoritması geliştirilmiştir. İş birliği algoritması herhangi bir aktivitenin gerçekleştirilmesi için görev durumlarına göre karar verir. Çalışma modelinde belirtilen ve kumaşların tüm süreçlerinin yönetilmesi için oluşturulan alt görevlerin ve bu görevler için atanan AGV'lerin görev durumlarının belirlenmesi işbirliğinin sağlanması için önemlidir. Bu görev durumları aşağıdaki gibidir:

- $S_1$ : Görev oluşturuldu
- $S_2$ : Başlangıç görevi
- $S_3$ : Görev çalıştırılmak için bekleniyor,
- $S_4$ : Görev çalıştırıldı,

- $S_5$ : Robot yük arabasına doğru ilerliyor,
- $S_6$ : Robot yük arabasına bağlanıyor,
- $S_7$ : Robot yükü bırakılacağı hedefe doğru hareket ediyor,
- $S_8$ : Robot hedefe vardı ve yükü bırakıyor,
- $S_9$ : Görev tamamlandı,

Bulut sisteme bir kumaş kaydedildiğinde ve çalışma modeline göre bir hareket planı oluşturulduğunda aşağıdaki adımlar çalıştırılır:

- Hareketin her bölümü için bir alt görev tanımlanır ve durumu  $S_1$  olarak ayarlanır. Her alt görev için bir sıra numarası verilir ve bu sıra ile çalıştırılması sağlanır.
- İlk görevin durumu  $S_2$  olarak ayarlanır. Bu görevin durumu  $S_9$  olduğunda, sonraki görevin durumu  $S_2$  olarak ayarlanır. Bu süreç tüm görevler tamamlanana kadar devam eder.
- Durumu  $S_2$  olan görevler için "en uygun yük arabasını seçme" algoritması çalıştırılır. Taşınan malzemeye en yakın yük arabası atandığında, görevin durumu  $S_3$  olarak değiştirilir.
- $S_3$  durumundaki kumaşları hedefe iletmek için en uygun AGV seçilir. Aynı anda birden fazla  $S_3$  görevi varsa, optimum çoklu görev dağılım algoritması, tüm görevler için en uygun AGV'lerin seçilmesini sağlar. Görev için en uygun AGV seçildiğinde görevin durumu  $S_4$  olarak güncellenir.
- Görev için oluşturulan optimum yol planları, WebSocket aracılığıyla seçilen AGV'lere gönderilir.
- AGV herhangi bir hedef noktasına bir yük arabası iletiyorsa ve bu makine meşgulse, en yakın boş bekleme istasyonuna iletilmelidir. Bu durumda, mevcut görevin durumu  $S_1$  olarak güncellenir ve AGV'yi en yakın bekleme istasyonuna götürmek için yeni bir görev oluşturulur ve bu görevin durumu  $S_4$  olarak ayarlanır.
- Yük arabası AGV tarafından bekleme istasyonuna ulaştığında, önceki görevin durumu  $S_3$  olarak güncellenir.

Bu işbirliğini sağlayan modülün sözde kodu Algoritma 4.4'te verilmiştir.

---

**Algoritma 4.4 : Çoklu AGV'ler için İşbirliği Algoritması**

---

```
function main ()
   $F \leftarrow$  task list of status is  $S_2$ 
  for all  $t \in F$  do
     $v \leftarrow$  FindMostSuitableVehicle ( $position(t)$ )
    Create new task for taking transport vehicle to task point and set status  $S_3$ 
    Update  $v$  as busy
    Update status of  $t$  to  $S_1$ 

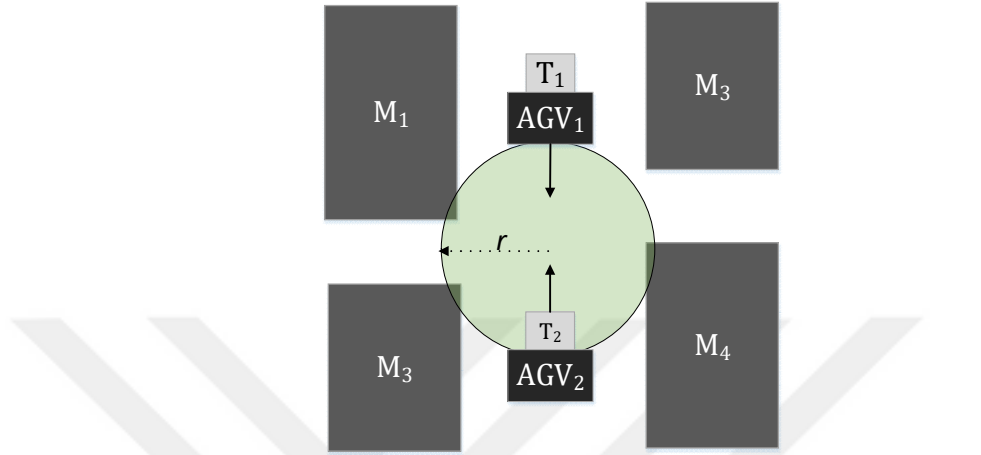
   $W \leftarrow$  Task list that status field is  $S_3$ 
   $R \leftarrow$  AGV list which is not busy and the battery level is above the critical value
   $L \leftarrow \emptyset$  //Allocation list
  for all  $t \in W$  do
    for all  $r \in R$  do
      if the type of task is "transfer" then
         $v \leftarrow$  FindMostSuitableVehicle ( $position(output\ port\ of\ workstation)$ )
        assign  $v$  to  $t$ 
        update  $v$  as busy
         $path \leftarrow$  GetOptimumPath ( $position(r), position(v)$ )
         $L.Insert(r, t, path)$ 
   $A \leftarrow$  getOptimumAllocation( $L$ )
  for all  $a \in A$  do
    if goal of  $a$  is a busy work station then
       $p \leftarrow$  find nearest waiting station to position of vehicle
      Update goal position as  $p$ 
       $a.path \leftarrow$  GetOptimumPath ( $position(vehicle), p$ )
      Update status of task as  $S_3$ 
      Update start position of task
      Increase the priority level of the task
```

---

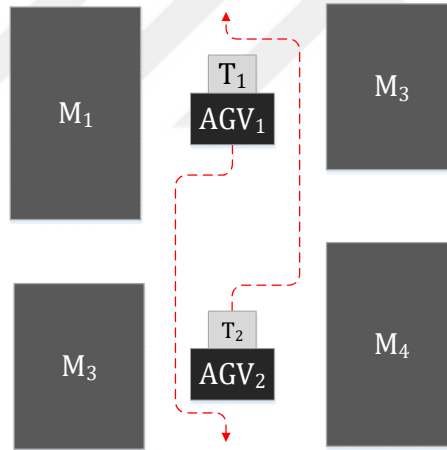
**4.2.7 Dinamik engellerden kaçınma**

Mobil robot uygulamalarındaki en büyük zorluklardan biri de, gezinme esnasında engellere çarpmadan yönlendirilmesini sağlamaktır. Ortamdaki statik engeller, haritalama esnasında konumları ve boyutları ile bulut sistemine kaydedildiği daha önce belirtilmiştir. Yol planlama algoritmasında robotun bu statik engellerden kaçınarak ilerlemesini sağlayacak şekilde planlama yapılmaktadır. Ancak sistemde dinamik engellerin de tespit edilmesi ve çarpmadan ilerlemenin sağlanması gerekmektedir. Dinamik engeller, bulut sistemi için tanımlanmış/tanımlanmamış hareketli nesnelerdir. Dinamik engel bir işletme çalışanı, bir araç veya başka bir AGV olabilir. Görev yürütme sırasında beklenmedik şekilde AGV'nin belirlenen yolu üzerinde hareket eden engeller olabilir. Bu hareketli engel bir AGV ise, çarpışma otomatik olarak önlenir. Böylece, aynı anda birden fazla AGV karşı karşıya kaldığında, çarpışma önleme modülü etkinleştirilir. Dinamik olarak tanımlanabilen  $r$

yarıçaplı güvenli bir dairesel alana girdiklerinde, AGV'ler için yeni bir yerel yol oluşturulur. Hareketli engel bir AGV değilse, varsayılan bir süre içinde engelin ortadan kalkmasını bekler. Bu sürenin ardından engel ortadan kalkmadığında WebSockets üzerinden bulut sisteme alarm mesajı gönderilir. Şekil 4.11 ve Şekil 4.12'de, çarpışmadan kaçınma sisteminin aşamaları örnek olarak verilmiştir.



**Şekil 4.11** : Aynı yarıçap içine giren robotların tespit edilmesi.

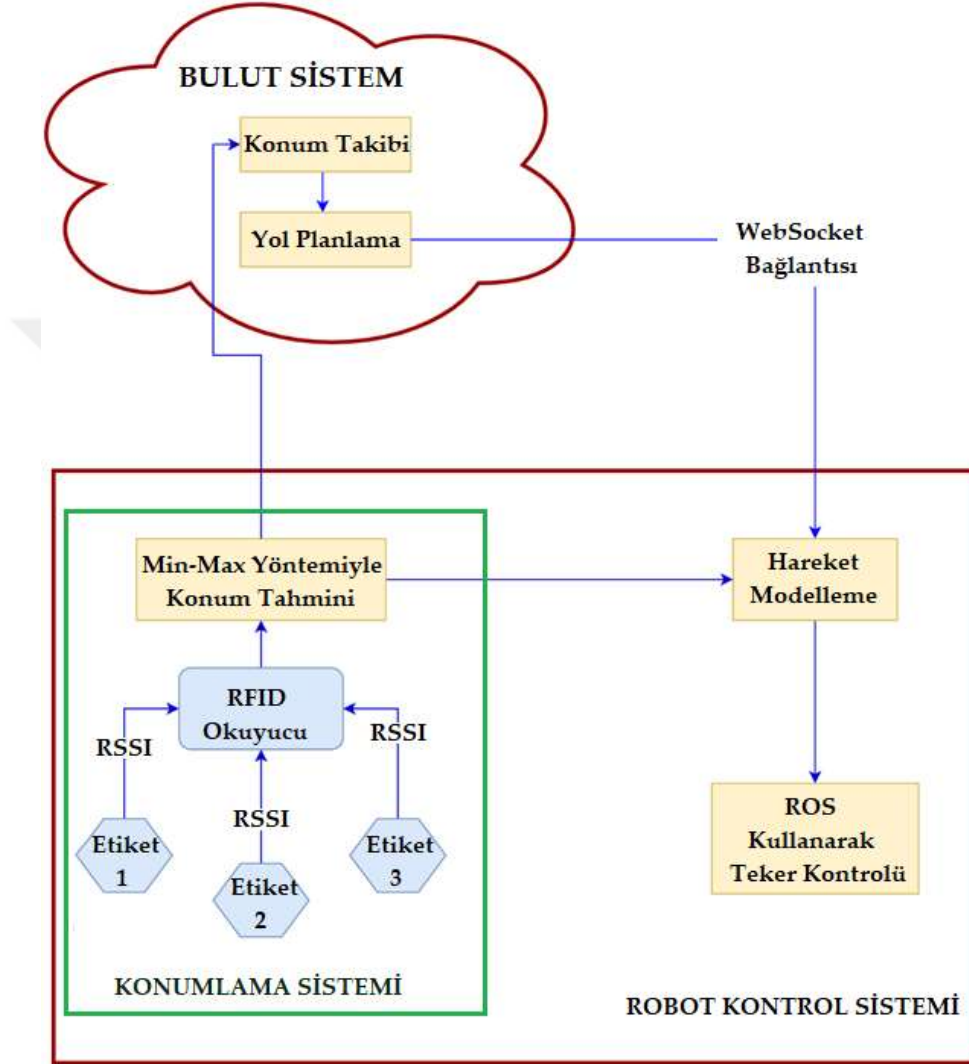


**Şekil 4.12** : Güvenli alana giren robotlar için yerel rota oluşturulması ve alandan çıkıldığında tekrar eski rotaya girilmesi.

### 4.3 Robot Kontrol Mimarisi

Ortamdaki robotların konumlarının tespit edilmesi, tespit edilen konumların bulut sisteme gönderilmesi, bulut sistemden yol planı emrinin alınması, bu yol planına göre robotun hareket modellemesinin yapılması gibi işlemlerin yönetimi için bir robot kontrol sistemi geliştirilmiştir. Python programlama dili kullanılarak geliştirilen bu sistem, Linux işletim sistemine sahip tek kartlı (single board) bilgisayarlar üzerinde çalıştırılabilmektedir. Robotun hareket modellemesi için ROS kullanılmıştır. Konumlama için RFID

teknolojisinden faydalanılmış ve sinyal gücünden hesaplanan uzaklık bilgilerini kullanan Min-Max yöntemi ile konum tahmini yapılmıştır. Konumlama hassasiyetinin artırılması için özgün bir ortam modeli önerilmiştir. Konum bilgisinin bulut sisteme gönderilmesi ve yol planı bilgisinin alınması için WebSocket kullanılmıştır. Bulut sistem ve robot kontrol sisteminin bir arada çalışmasını sağlayan mimari Şekil 4.13’de gösterilmiştir.



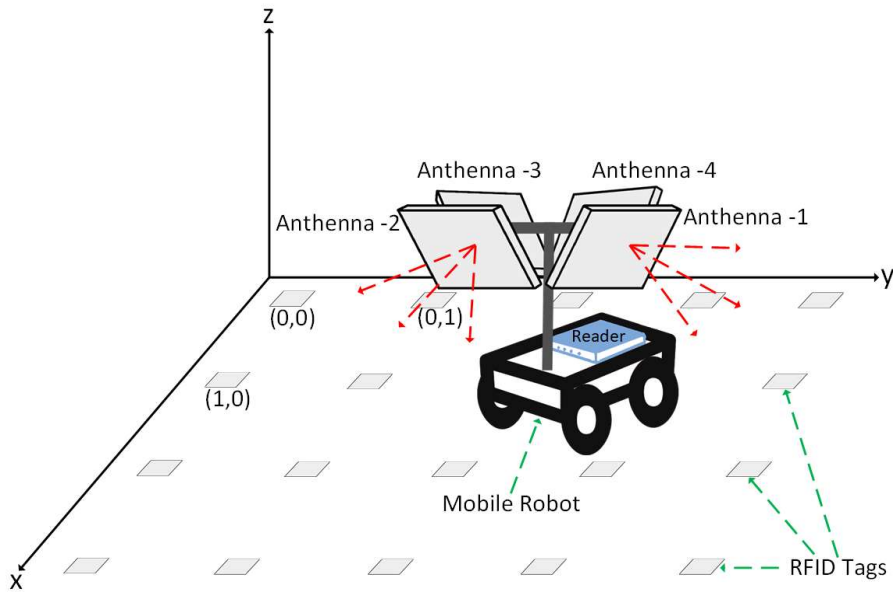
Şekil 4.13 :Bulut tabanlı robot kontrol mimarisi.

#### 4.3.1 Konumlama Sistemi

Endüstriyel işletmelerde mobil robotların kullanımı artarken kablosuz erişim ve yönetim fikri daha belirgin hale geldi. Mobil robotların kablosuz erişim ile yönetilmesi beraberinde bulut üzerinden geliştirilen uygulamalarla entegrasyonunu da getirdi. Bu çalışmada geliştirilen konumlama sistemi, kablosuz erişim ile robotların yönlendirilmesinin yanı sıra internet üzerinden, hatta mobil cihazlarla bile, robotların takip edilip yönetilebilmesini sağlamayı amaçlamaktadır. Geliştirilen sistemde hassas konum ölçümü ve hatasız

navigasyon için bir ortam modeli önerilmektedir. Konum tahmini için de RFID okuyucu ve etiketler vasıtasıyla elde edilen RSSI değerlerini temel alan Min-Max yöntemi kullanılmaktadır. Navigasyon için bulut uygulama üzerinden gelen yol planına göre otomatik tekerlek sürüşü gerçekleştirilmektedir. Otomatik sürüş esnasında tekerlek enkoderlerinden alınan bilgilerle odometrik konumlama hesaplanmaktadır. Min-Max ile tahmin edilen konum vasıtasıyla odometrik konum hesaplamalarında oluşan hatalar düzeltilmektedir.

RFID sinyalinin bozulması konumun tespitini etkileyerek konumlama sisteminin doğru çalışmasını etkileyebilmektedir. Ortamdaki metal nesnelere, Wi-Fi erişim noktaları, radyo frekansı yayan cihazlar, sinyal emen sıvılar, gezinti halindeki insanlar vs. RFID sinyalini doğru bir şekilde okumayı engellemektedir. Ayrıca zayıf anten gücü ve yanlış etiket seçimi de okuma hatasını artırmaktadır. Tüm bu etmenler, konum tahmini algoritması ne kadar güçlü olursa olsun okuma hassasiyetini ciddi oranda olumsuz etkilemektedir. Bu sebeple, çalışmada öncelikle hassasiyetin artırılması için çevresel etkenleri azaltan bir okuma modeli önerilmiştir. Önerilen bu modele göre RFID etiketleri zemine sabit aralıklarla ve komşuluk ilkesiyle ızgara şeklinde yerleştirilmiştir. Mobil robotun üzerine bir RFID okuyucu ve bu okuyucuya bağlı 4 adet yönlü RFID anten monte edilmiştir. Antenler robotun etrafındaki tüm etiketleri görebilecek sayıda ve şekilde yerleştirilmiştir. Robot, zemindeki etiketlerden gelen RSSI verilerine göre konumunu tespit etmektedir. Etiket okuma modeli Şekil 4.14'te gösterilmiştir.



**Şekil 4.14** :Etiket okuma modeli.

Her etiket arasındaki mesafe 1.2 m'lik sabit bir uzaklıkla belirlenmiştir. Bu mesafe okuyucunun maksimum okuma limitine kadar artırılarak maliyet düşürülebilir. Ancak, mesafenin artırılması maliyeti düşürebileceği gibi sinyal bozulmalarına daha çok maruz kalmakta ve böylece doğru okuma hassasiyetini olumsuz etkilemektedir. Etiketler arasındaki mesafe sabit bir değerle belirlendiği için komşuluk ilişkilerinden faydalanılarak mesafe daha hassas ölçülebilmektedir.

RFID okuyucu aynı anda çok sayıda etiketten sinyal alabilmektedir. Geliştirilen konumlama sisteminde RFID okuyucu ile okunan etiketler içinde RSSI değeri en fazla olan ilk üç etiketten gelen sinyaller kullanılarak Min-Max metodu uygulanmıştır. Etiketlerin seçimindeki önemli kriterlerden biri de üç etiketin birbirine komşu olması zorunluluğudur.

Önerilen ortam modelinde etiketler zemine yerleştirilerek antenlerin yönlü okuması sağlanmış ve ortamdaki yabancı maddelerin ve yabancı RF sinyallerinin etkisi azaltılmaya çalışılmıştır. Konum tespiti gerçek zamanlı yapılacağı için robotun, aynı anda 4 yönü de gözlemleyebilmesi gerekmektedir. Literatürde aynı anda tüm yönlerdeki etiketleri okuyabilmek için step motor ile antenleri yönlendiren çalışmalar da mevcuttur. Ancak dönme hızı motorun dönme şiddetine bağlı olarak okuma hızını doğrudan etkilemektedir. Ayrıca dönme hızı arttırıldığında sinyalin yayılımını da olumsuz etkilediği görülmüştür. Bu nedenle bu tez çalışmasında bir antenin motor vasıtasıyla döndürülmesi yerine, robotun etrafındaki tüm etiketleri okuyabilmesi için dört yönü gören 4 anten kullanılması tercih edilmiştir.

RSSI değerinin daha net elde edilebilmesi için etiket türü de önemlidir. Uygulamada etiketler zemine yerleştirileceği için beton veya fayans yüzeylerde okunabilen etiketler tercih edilmiştir. Antenlerin etiketleri daha iyi okuyabilmesi için yere doğru eğimli olması sağlanmıştır. Eğim açısı, robotun aynı anda görebileceği alana göre belirlenmektedir.

#### **4.3.2 Otonom hareket modelleme ve navigasyon**

Navigasyon kontrol modülü bulut sistem ile iletişimini WebSocket ile sağlamaktadır. Bulut sistem üzerinden gelen bir yol planlama emri ile birlikte aracın bu yol üzerinde ilerlemesini sağlayacak teker kontrolü bu aşamada gerçekleştirilmektedir. RFID tabanlı konumlama, robotun konumunu takip ederek doğru yol üzerinde ilerlemesini sağlamak için bir kontrol aracı olarak kullanılmıştır. Bulut sistemde oluşturulan yol planı robota iletildiğinde bu yol planı için gerekli olan tüm tekerlek hareketleri hesaplanır. Robota gönderilen yol planı üzerindeki noktalar, zeminde bulunan etiketlerin iki boyutlu düzlemdeki yerlerine göre

oluşturulmaktadır. Robot yol planını aldığıında bir noktadan diğerine gitmek için gereken hareket kinematiği oluşturularak bu kinematiğe göre tekerler hareket ettirilir. Asenkron olarak çalıştırılan RFID tabanlı konum tespit sistemi ile robotun yol planından çıkıp çıkmadığı tespit edilir. Eğer belirlenen yoldan sapma gerçekleşmişse hata düşürülmeye çalışılır. Önerilen navigasyon yönteminin sözde kodu, Algoritma 4.5'te verilmiştir.

---

**Algoritma 4.5** : Navigasyon Kontrol Algoritması

---

$E(t, a) \leftarrow \emptyset$   
 $L \leftarrow \emptyset$

**function** Move ( $t, a$ )

Turn the wheels by the angle  $a$  and move forward by  $t$  time  
 $L \leftarrow$  Calculate odometric location

**function** GetRSSILocation()

$N \leftarrow$  Top three nodes having maximum RSSIs  
 $(x, y) \leftarrow$  Calculate estimated location with Min-Max using  $N$   
Send  $(x, y)$  to Cloud Server via WebSocket  
**return**  $(x, y)$

**function** main ()

$G \leftarrow$  location of goal  
 $S \leftarrow$  starting location  
 $P \leftarrow$  ComputeShortestPath( $S, G$ )  
 $L \leftarrow S$   
**for all**  $p \in P$  **do**  
     $W(t, a) \leftarrow$  Calculate wheel movement between  $p_i$  and  $p_{i+1}$   
    Move ( $W, a$ )  
     $R \leftarrow$  GetRSSILocation()  
    **if**  $L \neq R$  **then**  
         $E(t, a) \leftarrow$  Calculate error between  $L$  and  $R$   
        Move ( $E(t, a)$ )

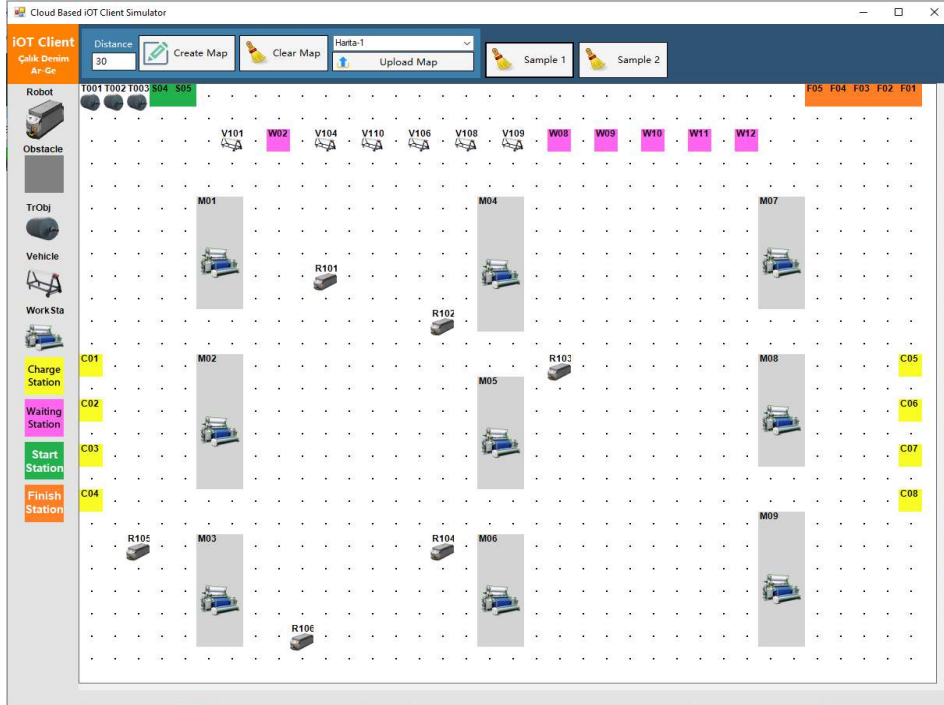
---

## 5. DENEYSEL SONUÇLAR

Materyal ve Metot bölümünde önerilen yöntemin doğruluğunun tespit edilmesi için ilk aşamada simülasyon yazılımları geliştirilmiştir. Simülasyon yazılımları ile yapılan deneysel çalışmalarla doğrulanmış yöntem, laboratuvar deneyleri ile gerçek ortamda da uygulanarak başarılı bir şekilde çalıştırılmıştır. Bu bölümde öncelikle simülasyon deneylerinden elde edilen sonuçlar ve ardından laboratuvar deneylerinin sonuçları sunulmuştur.

### 5.1 Simülasyon Sonuçları

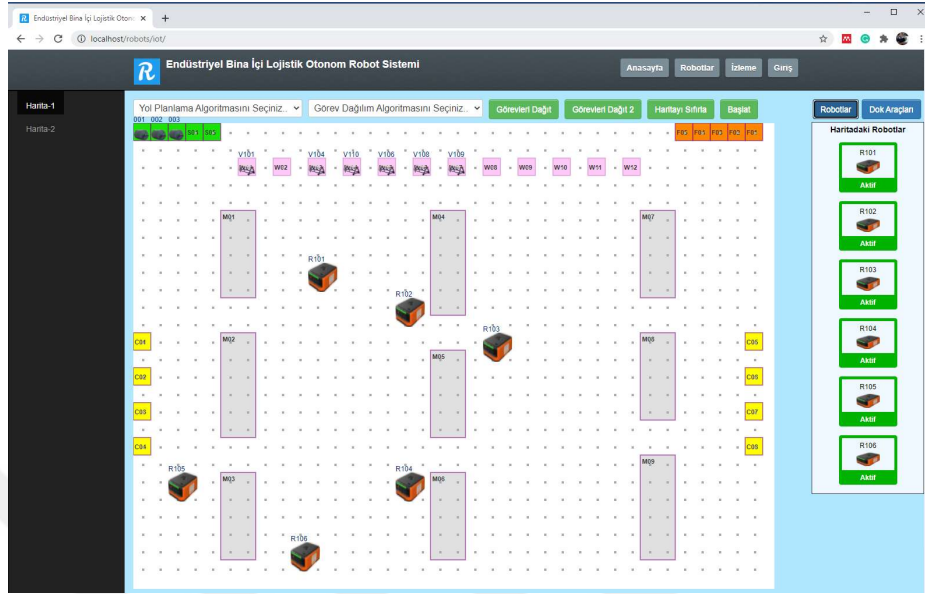
Simülasyon ortamı Visual Studio platformunda C# dili kullanılarak gerçekleştirildi. Oluşturulan simülasyon yazılımında harita ve bu haritaya bağlı iş makineleri, bekleme istasyonları, şarj istasyonları, başlangıç ve teslimat portları oluşturma özellikleri eklendi. Eklenen bu sabit varlıklar poligonal boyutları ve haritadaki konumları ile birlikte RESTful API'leri kullanılarak bulut sisteme yüklenmesi sağlandı. Simülasyon ortamında birden fazla harita eklenebilmektedir. Vaka çalışmasındaki haritaya benzer bir harita ortamı oluşturuldu ve bu harita üzerinde yapılan deneysel sonuçlar gözlemlendi. Şekil 5.1'de örnek olarak simülasyon yazılımı ekranında sanal bir ortam oluşturulması gösterilmiştir.



Şekil 5.1 : Simülasyon yazılımında oluşturulmuş sanal bir ortam haritası görüntüsü.

Simülasyon ekranında haritanın boyutları belirlenebilmekte, haritadaki nesnelere sürükleyip bırakarak ortamına eklenebilmekte ve her nesnenin harita üzerindeki boyutları

tanımlanabilmektedir. Harita üzerinde gerçekleştirilen her güncelleme RESTful web servisleri kullanılarak bulut sisteme aktarılmaktadır. Şekil 5.1’de verilen sanal ortamın bulut sistemdeki izleme ekranında oluşturduğu görüntü Şekil 5.2’de gösterilmiştir.



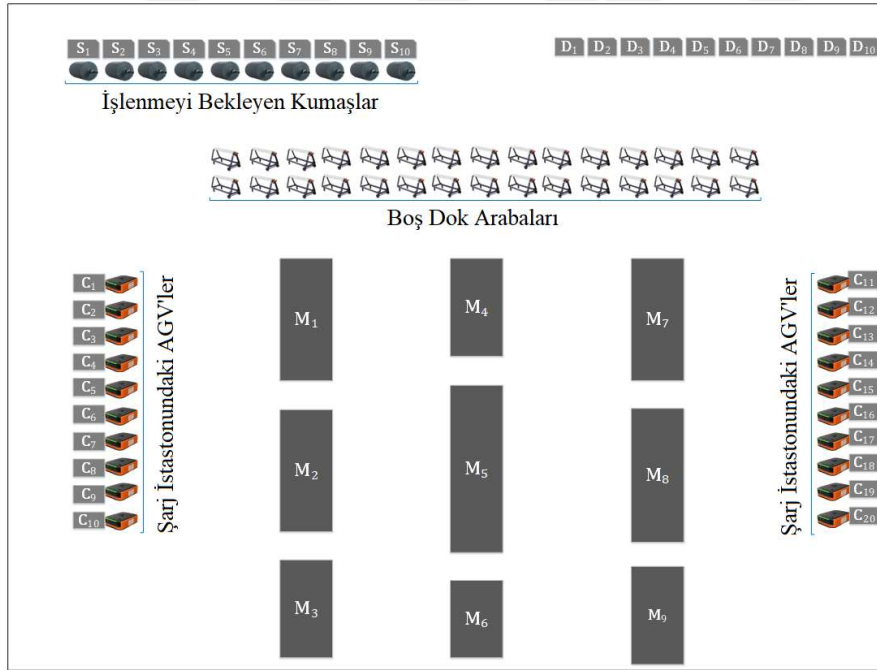
**Şekil 5.2 :** Sanal ortamın bulut sistemdeki harita izleme ekranı ile takip edilmesi.

Ortamdaki sabit nesnelerin harita verileri RESTful web servisleri kullanılarak bulut sisteme iletilirken, hareketli olan nesneler (AGV’ler, kumaşlar ve dok arabaları) haritaya eklendiğinde, bu değişiklikler bulut sisteme WebSocket’ler kullanılarak anlık olarak iletilmektedir. Simülasyon yazılımı ile gerçekleştirilen tüm deneylerde oluşturulan görevlerin başarıyla yerine getirildiği görülmüştür. Örnek bir deneysel çalışma ile ilgili olarak alınan sonuçların ve işleyişin detaylıca açıklaması aşağıda verilmiştir. Bu deneysel çalışmada sistemin işleyişi için gerekli olan dinamik parametreler şu şekilde tanımlandı:

- Tüm AGV’lerin aynı model ve aynı özelliklere sahip olduğu varsayıldı.
- Bir AGV’nin 1 metrelik mesafede harcadığı enerji değeri olan  $\delta$  parametresi 1 joule olarak ayarlandı.
- AGV’nin 1 metre mesafede taşıdığı ağırlığın enerji tüketimine etkisini belirleyen  $\mu$  parametresi 0.001 olarak ayarlandı.
- Kritik batarya seviyesinin hesaplanması için  $W_{max}$  1000 kg olarak belirlendi.
- AGV dönüş yaptığında harcanan enerji,  $c * 0.01$  olarak belirlendi.  $c$ , AGV’nin dönüş açısı miktarıdır.

- AGV'nin dok arabasına bağlanması ve kaldırması 2 joule enerji tüketecek şekilde ayarlandı.
- Boş bir arabasına ürünün yerleştirme süresi 60 saniye olarak belirlendi.
- Her makine 1 metre uzunluğundaki ürünü 1 saniyede işleyecek şekilde ayarlandı.
- 1 metre uzunluktaki ürünün ağırlığı 1 kg olarak varsayıldı.
- 1 AGV'nin bataryasının dolu haldeyken 250 joule enerji barındırdığı varsayıldı.
- Çarpışmayı engelleyen güvenlik alanının yarıçapı 5 metre olarak belirlendi.
- Atama matrisini parçalayan  $\phi$  parametresi 8 olarak ayarlandı.

Simülasyon testi için örnek bir çalışma Şekil 5.3'te gösterilmiştir. Bu örnekte 9 adet terbiye makinesinin, 20 adet AGV'nin ve 30 adet dok arabasının olduğu bir harita simüle edildi ve gelen 10 sipariş için işlenmek üzere 10 adet kumaş tanımlandı. AGV'ler rastgele olarak şarj istasyonlarında konumlandırıldı. Boş dok arabaları da bekleme istasyonlarına rastgele yerleştirildi.



**Şekil 5.3 :** 10-adet kumaşın, 9 adet terbiye makinesinin, 20 adet AGV'nin ve 30 adet dok arabasının bulunduğu örnek bir simülasyon haritası.

Şekil 5.3'teki örnek için her bir kumaşın hangi terbiye makinelerinde hangi sırayla işleme gireceği ile ilgili bilgiler Çizelge 5.1'deki gibi tanımlandı. Örnek olarak K<sub>1</sub> kumaşının S<sub>1</sub> başlangıç noktasından D<sub>1</sub> teslimat noktasına ulaşana kadar sırasıyla M<sub>1</sub>, M<sub>4</sub>, M<sub>5</sub> ve M<sub>8</sub> terbiye makinelerinde işlenmesi gerektiği görülmektedir.

**Çizelge 5.1 : Tanımlanan ürünler ve görevleri.**

Kumaş	Görev Atama ve Sıraları
K <sub>1</sub>	S <sub>1</sub> → M <sub>1</sub> → M <sub>4</sub> → M <sub>5</sub> → M <sub>8</sub> → D <sub>1</sub>
K <sub>2</sub>	S <sub>2</sub> → M <sub>2</sub> → M <sub>3</sub> → M <sub>7</sub> → M <sub>6</sub> → M <sub>9</sub> → D <sub>2</sub>
K <sub>3</sub>	S <sub>3</sub> → M <sub>3</sub> → M <sub>6</sub> → M <sub>4</sub> → M <sub>5</sub> → M <sub>8</sub> → D <sub>3</sub>
K <sub>4</sub>	S <sub>4</sub> → M <sub>1</sub> → M <sub>7</sub> → M <sub>8</sub> → D <sub>4</sub>
K <sub>5</sub>	S <sub>5</sub> → M <sub>3</sub> → M <sub>2</sub> → M <sub>7</sub> → M <sub>9</sub> → M <sub>6</sub> → D <sub>5</sub>
K <sub>6</sub>	S <sub>6</sub> → M <sub>4</sub> → M <sub>5</sub> → M <sub>2</sub> → M <sub>3</sub> → M <sub>7</sub> → D <sub>6</sub>
K <sub>7</sub>	S <sub>7</sub> → M <sub>2</sub> → M <sub>7</sub> → M <sub>3</sub> → M <sub>5</sub> → D <sub>7</sub>
K <sub>8</sub>	S <sub>8</sub> → M <sub>4</sub> → M <sub>1</sub> → M <sub>3</sub> → M <sub>6</sub> → M <sub>9</sub> → M <sub>8</sub> → D <sub>8</sub>
K <sub>9</sub>	S <sub>9</sub> → M <sub>5</sub> → M <sub>6</sub> → M <sub>4</sub> → M <sub>8</sub> → D <sub>9</sub>
K <sub>10</sub>	S <sub>10</sub> → M <sub>5</sub> → M <sub>6</sub> → M <sub>4</sub> → M <sub>8</sub> → D <sub>10</sub>

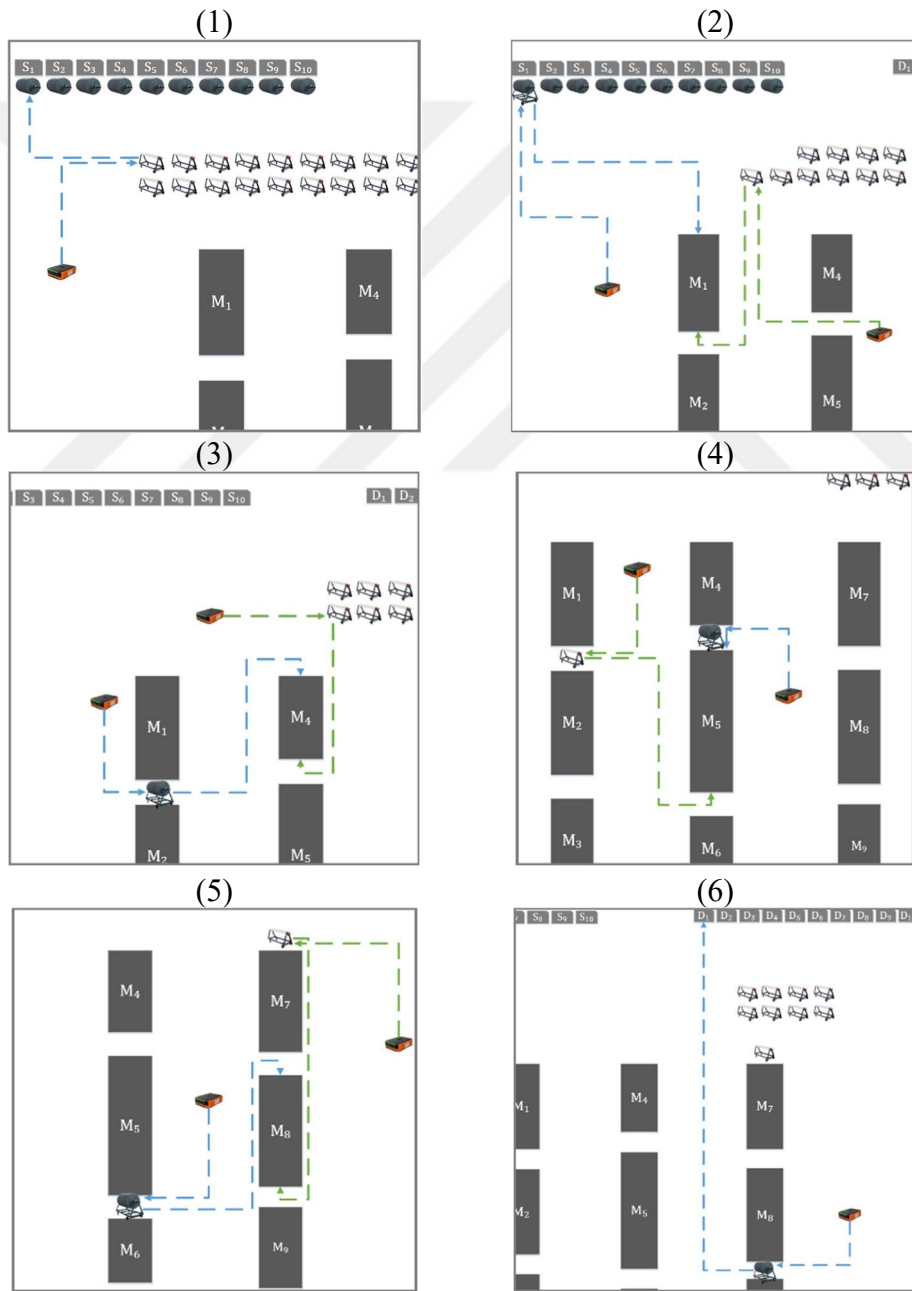
Bölüm 4.1’de belirtilen çalışma modeline göre, bir terbiye makinesinin giriş portuna bir kumaş ileildiğinde, boş bir dok arabası da aynı anda makinenin çıkış portuna iletilmelidir. Bu kurala göre, her makine giriş görevi için bir çıkış görevi oluşturulmuştur. Bu nedenle bu kumaş için belirlenen görev sayısının makine sayısı kadar arttığı gözlenmiştir. Bir robotun üzerine atanan görevde dok arabasının iletileceği makine meşgulse, çalışma modelindeki kural gereği, robotun dok arabasını uygun durumdaki bekleme istasyonuna iletmesi gerekmektedir. Bu durumdaki dok arabalarını bekleme istasyonuna iletme için bekleme görevleri oluşturulmuştur. Bu durum, kumaş için oluşturulan görev sayısının tekrar artması anlamına gelmektedir. Örnek deney için gerçekleşen sonuçlar Çizelge 5.2’de görülmektedir.

**Çizelge 5.2 : Örnek deney için gerçekleşen lojistik yönetiminin sonuçları.**

Kumaş	Uzunluk (m)	Makine Sayısı	Görev Sayısı	Bekleme Sayısı	Çarpışmadan Kaçınma	Toplam Yol (m)	Zaman (san.)	Enerji (joule)
K <sub>1</sub>	200	4	10	0	2	152	242	346
K <sub>2</sub>	250	5	13	1	5	296	368	548,4
K <sub>3</sub>	375	5	12	0	3	228	357	623,65
K <sub>4</sub>	300	3	9	1	0	131	196	309
K <sub>5</sub>	280	5	14	2	2	317	423	639,8
K <sub>6</sub>	325	5	12	0	4	222	349	541,32
K <sub>7</sub>	400	4	14	2	1	224	343	545,55
K <sub>8</sub>	415	6	16	2	6	368	516	881,4
K <sub>9</sub>	340	4	12	2	1	226	341	586,85
K <sub>10</sub>	950	4	15	3	4	303	372	636,9

Çizelge 5.2’de her bir kumaşın uzunluğu “Uzunluk” sütununda belirtilmiştir. "Makine Sayısı" sütununda, kumaşın işlendiği makinelerin sayısı hakkında bilgi bulunmaktadır. "Görevler" sütununda kumaşı teslimat noktasına iletme için oluşturulmuş toplam görev sayısı vardır. Bu görev numaraları, dok arabaları için oluşturulan başlangıç görevleri, giriş portu görevleri, çıkış portu görevleri, bekleme görevleri ve teslimat görevlerinin toplamını

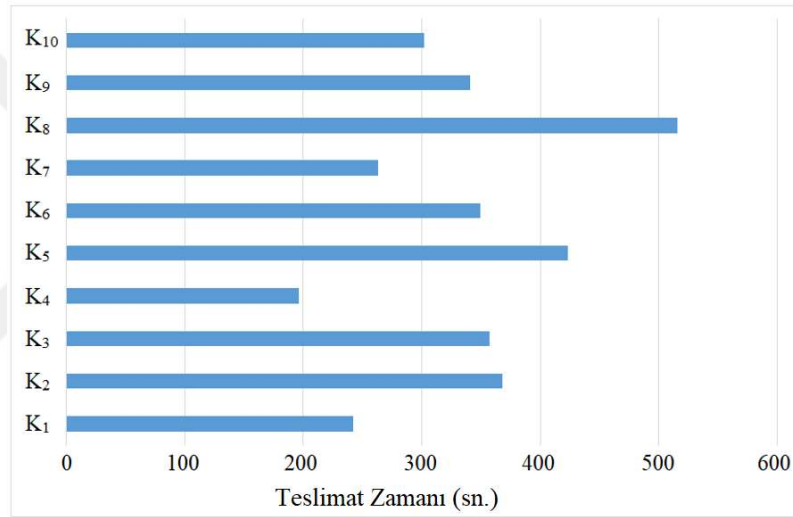
ifade eder. Örneğin  $K_1$  ürünü için Şekil 5.4'teki gibi 10 adet görev oluşturulduğu görülmektedir. "Bekleme Sayısı", kumaşın iletildiği makine meşgulken bekleme istasyonuna aktarılan parti arabası sayısıdır. "Çarpışmadan Kaçınma" sütunu, toplam süre boyunca yoldaki diğer AGV'lerle karşılaşma ve çarpışmadan kaçınma sayısıdır. "Toplam Yol" sütununda kumaşın tüm makinelerde işlenip teslimat noktasına teslim edilmesi için kat edilen toplam mesafe metre cinsinden belirtilmiştir. "Zaman" sütununda kumaş için tüm işlemlerin toplam süresi saniye cinsinden gösterilmiştir. "Enerji" sütununda harcanan toplam enerji joule cinsinden verilmiştir. Örnek deneyde tüm görevlerin başarıyla yerine getirildiği ve ürünlerin işlemlerden geçerek teslimat noktasına ulaştırıldığı görülmüştür.



Şekil 5.4 :  $K_1$  kumaşını teslimat noktasına ulaştıracak 10 adet görev.

Şekil 5.4'te (1) ile gösterilen imge başlangıç görevidir; boş bir dok arabası kumaşın bulunduğu  $S_1$  portuna götürülecektir. (2), kumaşı  $M_1$  makinesinin giriş portuna ulaştıracak ve aynı zamanda boş bir dok arabasını makinenin çıkış portuna ulaştıracak iki ayrı görevi göstermektedir. (3)  $M_4$  için, (4)  $M_5$  için, (5) ise  $M_8$  için giriş ve çıkış portu görevlerini ihtiva etmektedir. (6) işlem süresi biten ürün için teslimat görevini göstermektedir.

Simülasyon örneğinde tüm kumaşların lojistik görevleri aynı anda başlatılmıştır. Çizelge 5.2, en uzun teslim süresine sahip kumaşın  $K_8$  olduğunu göstermektedir. Bu durumda teslimat noktasına ulaşan son kumaş  $K_8$ 'dir. Simülasyonun çalışma süresi de  $K_8$ 'in teslim süresine eşdeğerdir. Şekil 5.5'te, yukarıdaki simülasyon örneğinde verilen 10 kumaşın teslim edilme süresinin zaman çizelgeleri gösterilmektedir. Şekilden de görülebileceği üzere, tüm görevler 516 sn. sonunda tamamlanmaktadır.



Şekil 5.5 : Ürünlerin teslimat-zaman çizelgesi.

Simülasyon örneğinde, kullanılan AGV'lerin 12'sine ait pillerin kritik seviyenin altına düştüğü görülmüştür. Bu AGV'ler için şarj görevleri oluşturuldu ve en yakın şarj istasyonuna iletildi. Tüm kumaşları teslim noktasına götürmek için harcanan toplam enerji 5.658,87 joule olmuştur. Simülasyonda gerçekleştirilen yukarıdaki örnek de dâhil olmak üzere tüm deneyler, bulut sisteminde oluşturulan tüm görevlerin başarıyla yerine getirildiğini ve kumaşların tüm terbiye işlemlerinden sonra teslimat noktasına iletildiğini göstermiştir. Geliştirilen optimum çoklu robot görev dağılımı algoritması rastgele görev atama yöntemiyle karşılaştırıldığında çok daha iyi sonuçlar elde edilmiştir. Aynı çalışma ortamında ve aynı parametrelerle rastgele görev atama yöntemi uygulandığında (meşgul olmayan dok arabaları ve AGV'ler kullanılarak) bu 10 adet kumaş 622 saniyede ve toplam 7.296,3 joule enerji harcanarak teslimat noktasına ulaştırılmıştır.

Yukarıdaki senaryoya benzer şekilde farklı sayıda kumaşların lojistik görevlerinin dağılımlarının aynı ortam değişkenlerinin olduğu haritada rastgele atama yöntemiyle karşılaştırılması Çizelge 5.3'te verilmiştir.

**Çizelge 5.3 :** Önerilen görev atama yönteminin rastgele atama yöntemiyle karşılaştırılması.

Kumaş Sayısı	Önerilen Yöntemle Harcanan Enerji (joule)	Önerilen Yöntemle İşlem Zamanı (sn.)	Rastgele Atamayla Harcanan Enerji (joule)	Rastgele Atamayla İşlem Zamanı (sn.)
10	5.658,87	516	7.296,3	622
15	8.779,84	488	12.055,42	841
20	14.429,65	472	24.683,12	994
25	20.739,82	506	43.558,08	1.107

Çizelge 5.3'te görüldüğü üzere önerilen yöntemde, rastgele atama yapıldığında alınan sonuca göre, hem enerji yönünden hem de işlem zamanı yönünden daha iyi sonuçlar alınmıştır. Ayrıca kumaş sayısı arttığında rastgele atamada karmaşıklık arttığı için verimlilik de azalmaktadır. Önerilen algoritmada ise kumaş sayısı (görev sayısı) ile harcanan enerjinin doğru orantılı bir şekilde arttığı gözlemlenmiştir. Önerilen yöntemde kumaş sayısı ne kadar olursa olsun işlem zamanı en uzun görev kapasitesine sahip kumaşın toplam işlem süresine bağlı kalmıştır. Ancak rastgele atama yönteminde herhangi bir planlama yapılmadığı için işlem süresi kontrolsüz bir şekilde artmıştır.

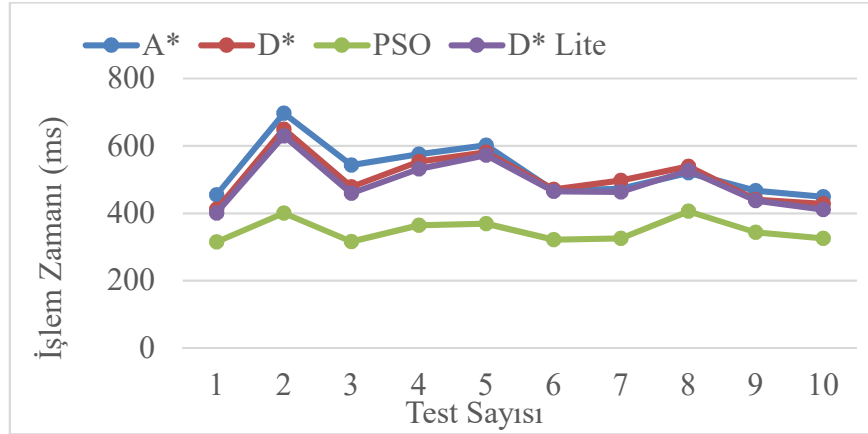
### 5.1.1 Yol planlama algoritmalarının karşılaştırılması

Yol planlama mobil robotlarda başlıca problemlerden biridir. Bu konuda literatürde yapılan birçok yaklaşım olduğu ve bu yaklaşımların örnekleri Bölüm 3.2'de belirtilmiştir. Bölüm 4.2.4'de ise yol planlama yöntemi olarak D\* Lite algoritmasının tercih edildiği belirtilmiştir.

Yapılan çalışmalarda yol planlama için en fazla tercih edilen yöntemler graf tabanlı ve meta-sezgisel algoritmalarlardır. Bu bölümde literatürde sıklıkla kullanılan A\*, D\*, D\* Lite ve PSO algoritmalarının karşılaştırılması için yapılan simülasyon çalışmasının sonuçlarına yer verilmiştir. Yol planlama simülasyon yazılımında robotların gezindiği alan 20x20 düğümünden oluşan bir grid olarak tasarlanmıştır. Grid üzerindeki düğümler birden fazla engel, robot ve hedef objeleri olarak tanımlanmıştır. Her robot, düğümler üzerinde oluşturulan yol planı çerçevesinde ilerleyecek şekilde tasarlanmıştır.

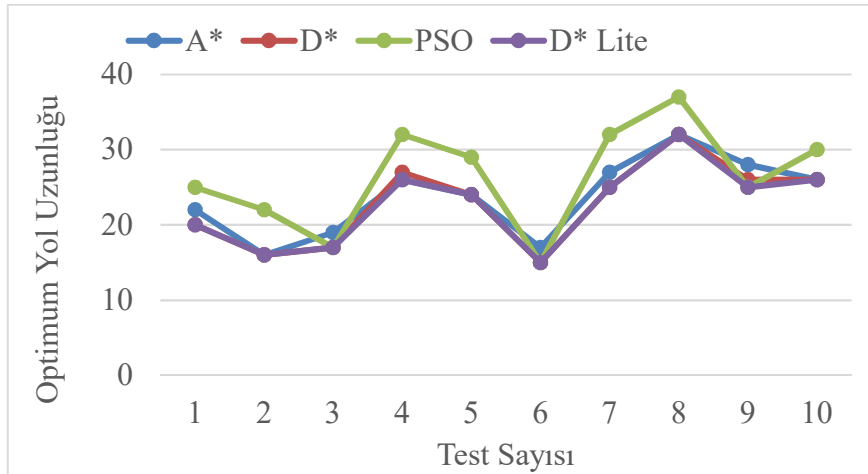
Algoritmaların karşılaştırılması, işlem zamanı ve düğüm sayısı kriterleri gözetilerek yapılmıştır. Şekil 5.6'da 10 farklı test ortamında üç algoritma için de işlem zamanlarının grafiği gösterilmiştir. Grafikte de görüleceği üzere D\* Lite, D\* ve A\* algoritmalarının

çalışma süreleri PSO'ya göre daha fazladır. 10 adet test ortamı için ortalama işlem zamanı A\* ile 525 ms, D\* ile 505 ms, D\* Lite ile 490 ms ve PSO ile 323 ms olmuştur.



Şekil 5.6 : İşlem sürelerinin karşılaştırılması.

Aynı şekilde optimal yol uzunluğu için de 10 farklı test simülasyonunda algoritmaların performansları ölçülmüş ve sonuçlar Şekil 5.7'de gösterilmiştir. Yol uzunluğu hesaplanırken robotların başlangıç noktasından hedefe varana kadar uğradığı düğümlerin sayısına bakılmıştır. Grafikte de görüleceği üzere D\* Lite algoritmasıyla diğer yöntemlere kıyasla optimal çözüme daha yakın sonuçlar elde edilmiştir. Ortalama yol uzunlukları A\* için 23,7; D\* için 22,8; D\* Lite için 22,6 ve PSO için 26,4 olmuştur.

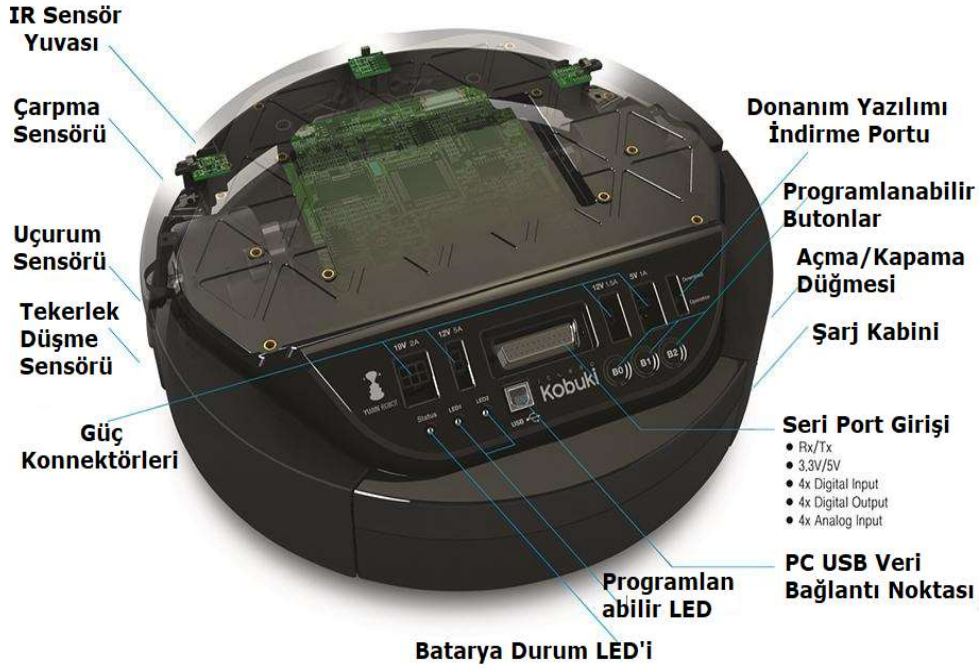


Şekil 5.7 : Optimal yol uzunluklarının karşılaştırılması

Elde edilen sonuçlara göre graf tabanlı algoritmalar meta-sezgisel algoritmalara göre daha yavaş çalışırken optimal yol hesaplamasında daha iyi sonuçlar vermektedir. Meta-sezgisel algoritmalarda yerel minimum çözümünden kaçamama problemi, optimum yol planı hesaplamayı oldukça etkilemektedir.

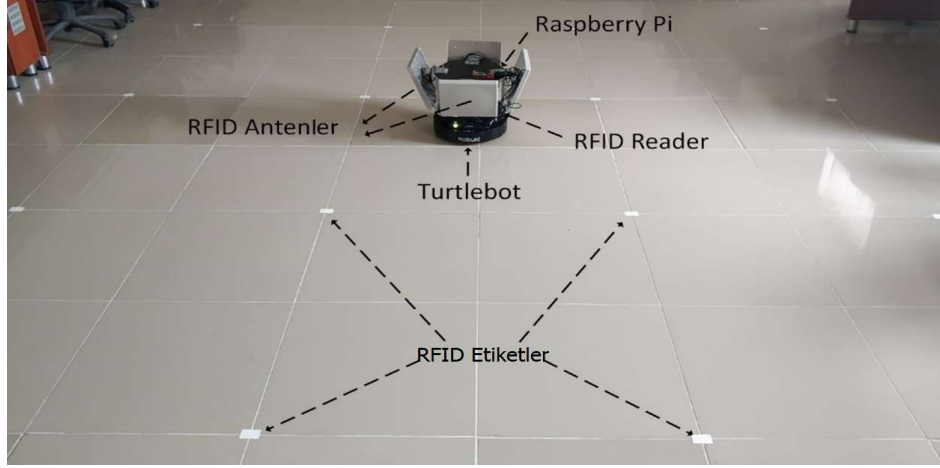
## 5.2 Laboratuvar (Gerçek Ortam) Sonuçları

Laboratuvar deneylerinde test robotu olarak TurtleBot 2 (Kobuki) modeli kullanılmıştır. TurtleBot, açık kaynaklı yazılıma sahip, düşük maliyetli bir robot kitidir ve ROS uyumlu çalışabilmektedir. Şekil 5.8’de özellikleri ile birlikte gösterilmiştir.



Şekil 5.8 : TurtleBot 2

Önerilen RFID okuma modelinin laboratuvar ortamında uygulanması için zemine 120 cm aralıklarla RFGate UPM Web 208-3 model RFID etiketler yerleştirilmiştir. Bu etiketler beton zeminde veya beton içinde sinyal yansıtma özelliğine sahip olduğu için tercih edilmiştir. RFID etiketlerin okunması için robot üzerine RSSI desteği olan RFGate RGR400 model sabit UHF RFID okuyucu monte edilmiştir. Robotun etrafındaki tüm etiketleri okuyabilmesi için 4 adet ANT-263 12 DBI UHF Circular model RFID anten, okuyucunun portlarına bağlanarak 30 derecelik açıyla yere bakacak şekilde robota yerleştirilmiştir. Robotu kontrol etmek ve bulut sunucuya bağlamak için Raspberry Pi 3 kullanılmıştır. Bu sayede robotun esnek hareket edebilmesi ve mobilitesi de sağlanmıştır. TurtleBot üzerindeki 5V’lik güç konnektörü kullanılarak Raspberry Pi için enerji sağlanmıştır. Raspberry Pi’ye Ubuntu Mate işletim sistemi ve ROS kurulumu yapılmıştır. Otonom tekerlek yönetimi ROS kullanılarak gerçekleştirilmiştir. Konumlama ve Navigasyon kontrol modülleri Raspberry Pi üzerinde çalıştırılmıştır. Laboratuvar ortamından örnek bir görüntü Şekil 5.9’da gösterilmiştir.



**Şekil 5.9 :** Önerilen modelin laboratuvar ortamına uyarlaması.

### 5.2.1 Konumlama sistemi sonuçları

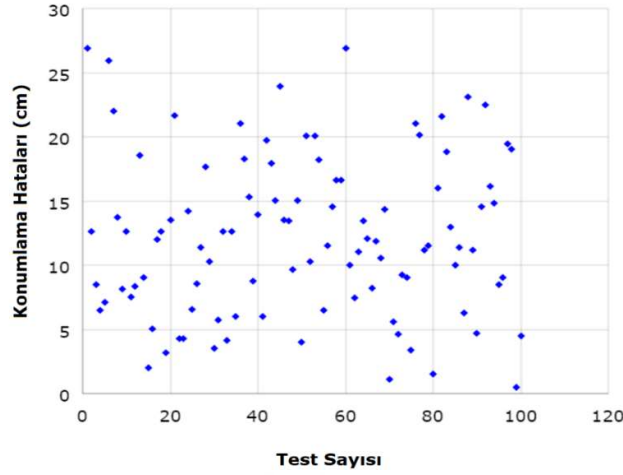
Konum tespiti algoritmasının işleyişini test etmek için Şekil 5.9'daki çalışma ortamında robot,  $(x, y)$  düzlemindeki koordinatı bilinen noktalarda konumlandırılmış ve aynı zamanda önerilen konumlama sistemi ile konum tahmini yapılmıştır. Tahmini konum ve gerçek konum arasındaki hata Öklid uzaklığı ile ölçülmüştür. Bu şekilde  $360 \times 360$  (cm<sup>2</sup>)'lik alanda, aynı şartlar altında 100 farklı konumda ölçümler yapılarak Çizelge 5.4'teki sonuçlar elde edilmiştir.

**Çizelge 5.4 :** Gerçek ortam konumlama hassasiyeti.

Test No	d <sub>1</sub> (cm)	d <sub>2</sub> (cm)	d <sub>3</sub> (cm)	E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>	E <sub>4</sub>	Tahmini Nokta	Gerçek Konum	Hata (cm)
1	69,11	126,71	51,15	(62,62)	(117, 62)	(62,67)	(117,67)	(89,64)	(115,70)	26,68
2	84,17	85,58	82,73	(153,33)	(224,33)	(153,11)	(224,11)	(189,71)	(180,60)	12,72
3	54,70	103,31	78,18	(116,16)	(188,16)	(116,18)	(188,18)	(152,17)	(150,17)	4,47
4	99,72	24,18	110,02	(237,20)	(272,20)	(237,24)	(272,24)	(254,22)	(250,22)	6,40
5	89,81	45,45	97,49	(197,24)	(215,24)	(197,28)	(197,28)	(206,26)	(205,27)	8,06
6	50,44	118,59	72,00	(240,28)	(311,28)	(240,31)	(311,31)	(275,30)	(255,29)	20,09
...	...	...	...	...	...	...	...	...	...	...
100	65,45	114,73	61,68	(50,35)	(115,35)	(50,70)	(115,70)	(82,52)	(100,60)	19,69
<b>Ort.</b>	<b>95,50</b>	<b>91,77</b>	<b>89,62</b>	-	-	-	-	-	-	<b>12,26</b>

Çizelge 5.4'te görüldüğü üzere 100 test sonucunda ortalama hata miktarı 12,26 cm olarak ölçülmüştür. En yüksek hata miktarı 27,56 cm olarak ölçülürken en az hata 0,5 cm ölçülmüştür. 100 adet testin hata dağılım grafiği Şekil 5.10'da gösterilmiştir. Şekil 5.10'da görüleceği üzere hata miktarlarında homojen bir dağılım gerçekleşmiştir. Ortaya çıkan hata

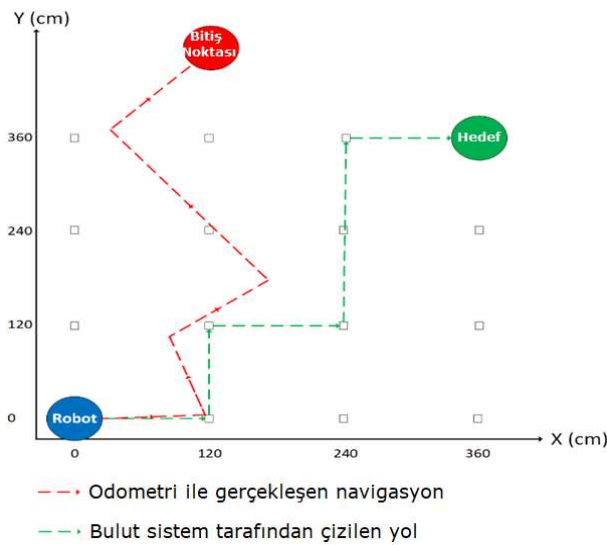
ortalaması, benzer çalışmalara nazaran oldukça başarılı sonuçlar elde edildiğini göstermektedir.



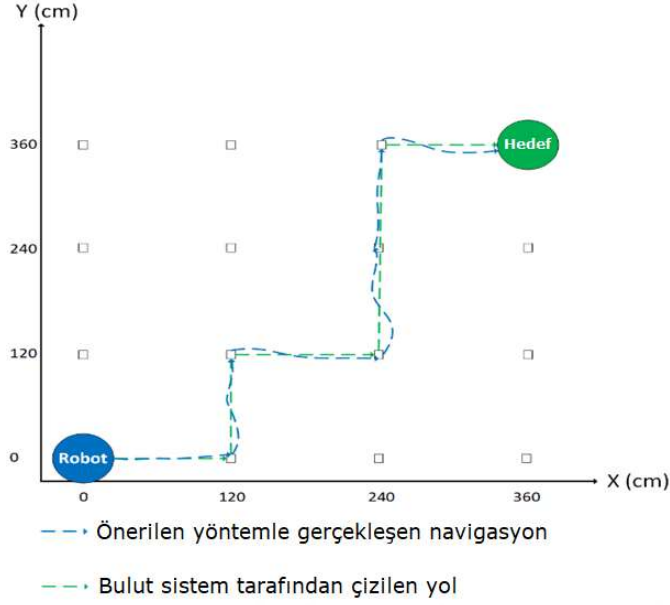
Şekil 5.10 : Test bazında hata dağılımı.

### 5.2.2 Bulut tabanlı navigasyon sonuçları

Navigasyon kontrolünün testi için birden fazla hareket senaryosu oluşturulmuş ve başarıyla uygulanmıştır. Test sonuçları göstermiştir ki sadece teker kinematiği ile doğru bir navigasyon sağlanamamaktadır. Yapılan testlerde sadece teker kinematiği kullanıldığında, sürtünme, köşelerden dönme ve yol üzerindeki irili ufaklı çukurlar nedeniyle hedef yoldan sapmalar gerçekleştiği tespit edilmiştir. Bu sapmaların yol boyunca kümülatif olarak arttığı ve toplam sapmayla birlikte robotun hedef noktaya oldukça uzak bir konumda durduğu tespit edilmiştir. Bu farklı sonucun elde edilmesinde robotun dönme hareketlerinde oluşan aç kayıpları önemli bir rol oynamaktadır. Aynı ortam ve şartlarda gerçekleştirilen örnek bir navigasyon karşılaştırması Şekil 5.11 ve Şekil 5.12’de gösterilmiştir.



Şekil 5.11 : Odometrik konum tahmini ile navigasyon.



**Şekil 5.12 :**Önerilen yöntem ile navigasyon.

Önerilen navigasyon yöntemiyle yapılan uygulamalarda, odometri ile navigasyon yöntemine kıyasla hedef noktaya oldukça yakın sonuçlar elde edilmiştir. Şekil 5.12’de görüldüğü üzere, özellikle dönüş esnasında gerçekleşen hatalar, sonucu en çok etkileyen unsurlardır.

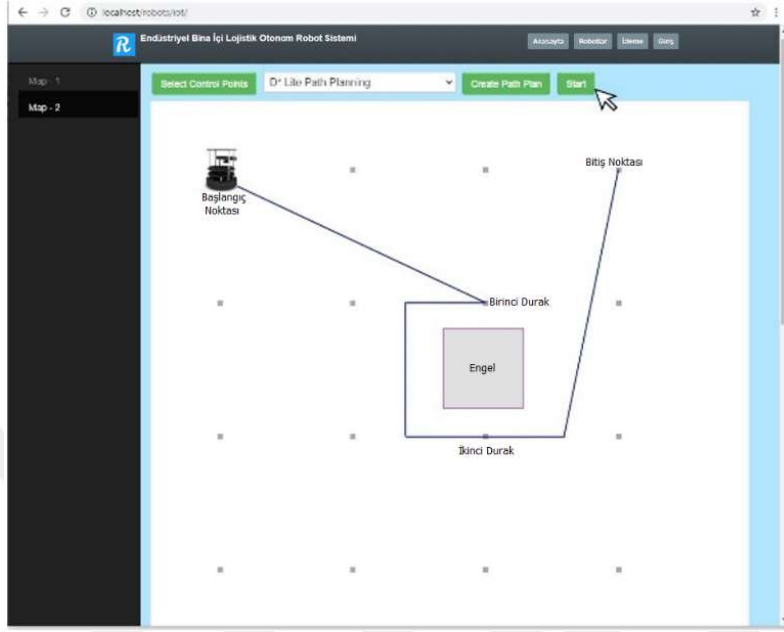
Bulut sistemde geliştirilen web uygulama ile laboratuvar ortamının haritası sisteme tanıtılmış, robotun uğrayacağı istasyonlar ve bitiş noktası tanımlanmış ve robotun en kısa yoldan bu istasyonlara uğrayarak bitiş noktasına iletilmesi için komut gönderilmesi sağlanmıştır. Aynı ortamda ve aynı yol üzerinde hem önerilen metot ile hem de odometri ile yapılan 5 adet testin sonucu Çizelge 5.5’te gösterilmiştir.

**Çizelge 5.5 :** Teker kinematiği ve önerilen sistem ile oluşan toplam sapmaların karşılaştırılması.

Deney No	Toplam Yol (cm)	Dönüş sayısı	Odometri ile Hedeften Sapma (cm)	Önerilen sistem ile hedeften sapma (cm)	Başarım Yüzdesi
Test 1	680	4	218	12	94,5
Test 2	640	3	115	17	85,2
Test 3	1080	3	382	9	97,6
Test 4	720	2	155	13	91,6
Test 5	480	1	56	11	80,3

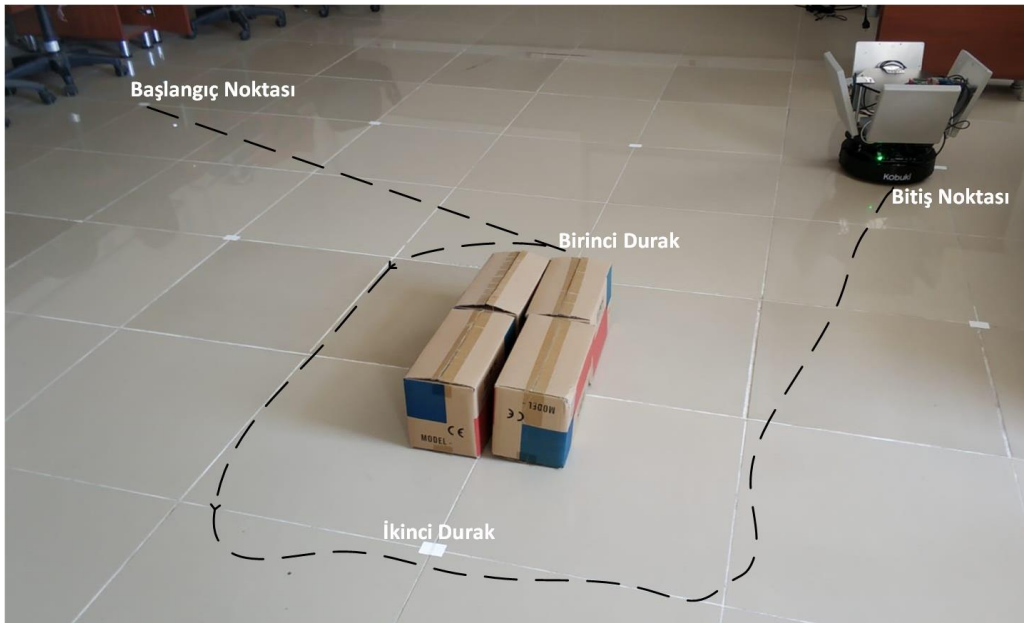
Çizelge 5.5’te görüldüğü üzere, önerilen sistem ile elde edilen hedeften sapma miktarı oldukça düşüktür. Belirtilen 5 adet deneyde oluşan ortalama sapma değeri 12,4 cm’dir. Yapılan deneylerde robotun maruz kaldığı dönüş işlemlerinin sapma miktarını daha çok etkilediği görülmüştür. Bu durum Çizelge 5.5’teki teker kinematiği ile elde edilen sonuçlara da yansımıştır. Önerilen yöntemde ise yüksek miktarda sapma durumlarıyla

karşılaşmamıştır. Dönüşlerde veya direkt yol üzerinde ilerlerken yaşanan sapma miktarlarının konum tahmin algoritmasıyla minimize edilmesi sağlanmıştır. Önerilen sistem bu 5 deney için ortalama %90,46 daha başarılı çalışmıştır. Örnek bir navigasyon kontrolü Şekil 5.13'te ve Şekil 5.14'te gösterilmiştir.



Şekil 5.13 :Bulut yazılım ile yol planı oluşturma.

Şekil 5.13'te gösterilen navigasyon kontrolü laboratuvar çalışmasının web uygulamadan izlenme ve yönlendirme aşamasıdır. Bulut sistemin gerçekleştirdiği yönlendirme sonucunda Şekil 5.14'teki navigasyon işlemi gerçekleşmiştir. Hareket esnasında bulut sistem, robotun konumundaki anlık değişimleri harita üzerinden göstermiştir.



Şekil 5.14 :Bulut yazılımdan gelen yol planı emirin uygulanması.

Şekil 5.14'te verilen örnek görüntüde öncelikli olarak ortam haritası bulut sisteme tanıtılmıştır. Robot aktif edilip başlangıç pozisyonu bulut yazılıma gönderildiğinde harita üzerinde belirgin hale gelmiştir. Ortama bir engel eklenerek pozisyon bilgileri buluta girilmiştir. Robot için 2 adet durak ve bir son durak bilgisi tanımlandıktan sonra navigasyon işlemine başlama komutu gönderilmiştir. Otomatik yol planı oluşturularak robota hareketlilik kazandırılmış ve D\* Lite ile oluşturulan engelsiz yolda belirlenen lokasyonlara uğrayarak son durağa iletilmesi sağlanmıştır. Yukarıda verilen örnekte olduğu gibi gerçekleştirilen tüm deneylerde navigasyon işlemi başarıyla tamamlanmıştır.



## 6. SONUÇLAR VE ÖNERİLER

Tez çalışmasında, endüstriyel otonom bina içi lojistik için bulut tabanlı çoklu mobil robot yönetim sistemi geliştirilmiştir. Tezin ana motivasyonu, Çalık Denim A.Ş.'nin kumaş terbiye işletmesindeki bina içi kumaş lojistiği süreçlerinin otonomlaştırılmasıdır. Bu bağlamda ilgili kuruluş Ar-Ge merkezinde yarı zamanlı araştırmacı olarak görev alınmıştır. Sonrasında İnönü Üniversitesi BAP birimi üzerinden tez ile ilgili iki ÜSİP projesi yazılmış ve kabul almıştır. Bu vesile ile tez kapsamında örnek bir Üniversite-Sanayi İşbirliği Projesi yürütülmüştür. Geliştirilen sistem kumaş terbiye işletmeleri özelinde uygulanmış olsa da, karmaşık veya basit lojistik süreçlerine sahip birçok endüstri için de çözümler sunmaktadır. Sistemdeki modüler özellikler oldukça umut vericidir. Bu modüler özellikler, son yıllarda öne çıkan ilgili çalışmalarla karşılaştırılmış ve karşılaştırma sonuçları Çizelge 1.1'de verilmiştir.

Çalışmada öncelikle terbiye işletmesindeki lojistiğe etki eden tüm nesnelere tanımlanmış ve kumaş taşımacılığını otonom hale getirecek bir çalışma modeli oluşturulmuştur. Oluşturulan bu modelde her bir nesnenin ortamdaki rolü tanımlanmıştır. Optimum çoklu görev dağılımı ve yol planlama algoritmalarının çalışma stratejisinde bu model önemli bir rol oynamıştır.

Lojistikte kullanılan çoklu robotların (AGV) işbirliği içinde yönetilmesini sağlamak için merkezi kontrol yaklaşımını benimseyen bir bulut sistemi geliştirildi. Bulut sistemde, WebSocket ve RESTful teknolojileri kullanılarak robotlar uyum içinde ve en düşük maliyetli çalışma ilkesiyle çalışması sağlandı. Bu en düşük maliyetli ilke, toplam yolu ve zamanı en aza indiren çoklu amaç fonksiyonuna dayanmaktadır. Bu amaç fonksiyonuna uygun bir işbirliği algoritması geliştirildi. Algoritma, görev tanımı, pil yönetimi-optimizasyonu, en uygun yük arabasını (dok arabası) seçme, optimum çoklu görev tahsisi ve çarpışmadan kaçınma aşamalarından oluşmaktadır.

Görev tanımlama aşamasında kumaş üzerinde yapılacak tüm işlemlerin tanımlanması sağlandı. Bu işlemler, kumaşları buldukları yerden alıp sıralı olarak işlenecek makinelere ileten ve son olarak teslimat noktasına ileten alt görevlerdir.

Batarya yönetimi ve optimizasyon aşamasında, her bir robot için kritik batarya seviyesinin hesaplanması ve takip edilmesi sağlandı. Bataryaların doluluk oranı bu seviyenin altına düştüğünde robota atanan görevin iptal edilmesi ve başka bir robota atanması sağlandı.

En uygun yük arabasını seçme aşamasında, yük arabalarının doluluk durumu dikkate alınarak en yakındakini bulan bir algoritma geliştirildi. Bu algorithmada, engellerden kaçınarak kumaşlar ve yük arabaları arasındaki en kısa yolu hesaplamak için D\* Lite kullanıldı.

Optimum çoklu görev dağılımı aşamasında, toplam enerji maliyetini düşürmeyi ve günlük üretim sayısını artırmak için minimum yolu temel alan çoklu amaç fonksiyonu doğrultusunda bir algoritma geliştirildi. Algorithmada öncelikle robotlar ile görev pozisyonları arasında bir kombinasyon matrisi oluşturuldu ve aralarındaki optimum yollar D\* Lite ile hesaplandı. Minimum yolları hesapladıktan sonra, her robot ve görev için birebir atama olasılıklarını kapsayan benzersiz satırlara sahip bir permütasyon atama matrisi oluşturuldu. Matristeki her satır için toplam yollar hesaplanarak, en düşük maliyet satırı seçildi. Atama matrisi permütasyonel olduğundan, robotlar ve görevler arttıkça hesaplama karmaşıklığının arttığı gözlenmiştir. Bu problemi çözmek için, en yakın komşuluğuna göre k-NN'ye göre  $\varphi$  sayıda kombinasyonlar sınıflandırıldı. Permütasyon hesaplaması bu sınıflandırılmış verilerle ayrı ayrı hesaplandı ve nihai bir matris halinde birleştirildi. Nihai atama matrisinde en uygun satırı seçerken, sadece yol uzunlukları değil, aynı zamanda robotlara atanan görevi yerine getirmek için enerji seviyesi de dikkate alındı. Görev için yeterli enerjiye sahip olmayan bir robot içeren satırlar atama matrisinden çıkarıldı. Böylece görev dağılımı, atama matrisinde kalan satırlar arasından en düşük toplam yol seçilerek gerçekleştirildi. Seçilen satırdaki görev dağılımına uygun olarak robotlara izlemesi gereken yol bilgisi gönderildi. Yapılan deneylerde geliştirilen optimum çoklu robot görev dağılımı algoritmasının her zaman optimum dağılımı gerçekleştirdiği görüldü.

Robotların gelen emirleri algılayıp gerçekleştirmesi için bir robot kontrol sistemi geliştirildi. Robot kontrol sistemi, robotun haritadaki yerini belirlemesini sağlayan konumlama aşaması ve yol planı üzerinde hareket etmesini sağlayan otonom hareket modelleme/navigasyon aşamalarından oluşmaktadır.

Geliştirilen konum tahmini sisteminde, RFID teknolojisinden faydalanıldı ve sinyal gücü göstergesi (RSSI) değerleri kullanılarak Min-Max yöntemiyle en yakın noktanın tahmin edilmesi sağlandı. RSSI ölçümünün çevresel faktörlerden etkilenerek yanlış okumalara yol açması problemine odaklanılarak bu etkenleri olabildiğince azaltmak için bir ortam modeli önerildi. Önerilen ortam modeli, zemine belirli aralıklarla ızgara şeklinde yerleştirilen RFID etiketlerinin robot üzerinde bulunan ve yere doğru eğimli bir şekilde monte edilen RFID antenler sayesinde okunmasına dayanmaktadır. Konum tahmininde, yapılan laboratuvar

deneyleri sonucunda ortalama  $\pm 12,26$  cm'lik bir konumlama hassasiyeti elde edildiği görüldü.

Otonom hareket modelleme ve navigasyon kontrol modülünde, bulut yazılım ile WebSocket kullanarak entegre olarak çalışan ve merkezden aldığı yol bilgisini teker kinematiğine çeviren bir mekanizma tasarlandı. Teker kinematiğinin oluşturulmasında ve hareket işlemlerinde ROS kullanıldı. Konum tahmin sisteminden gelen lokasyon bilgisiyle yoldan sapma miktarı hesaplandı ve bu hesaba göre robotun teker kinematiğini değiştiren asenkron bir kontrol yazılımı gerçekleştirildi. Navigasyon kontrolü için yapılan laboratuvar deneylerinde ortalama  $\pm 12,4$  cm'lik hassasiyetle hedefe varıldığı görüldü.

## 6.1 Sonuçların Değerlendirilmesi

Geliştirilen bulut tabanlı otonom robot yönetim sistemi genel olarak değerlendirildiğinde çok sayıda modülden oluştuğu görülmektedir. literatürde verilen örneklere kıyasla modüler yapısının oldukça kapsamlı olduğu Çizelge 1.1'de gösterilmiştir. Çizelgeden de görüleceği üzere tez çalışması birçok alanda diğer çalışmalardan fazla içeriğe sahiptir. Geliştirilen sistemde literatürdeki birçok yöntemden faydalanılsa da, çoklu robot görev dağılımı algoritması, konumlama sistemi ve navigasyon yöntemi özgün çalışmalarla oluşturulmuştur. Bu özgün çalışmalar için gerçekleştirilen deneysel çalışmaların sonuçları aşağıda değerlendirilmiştir:

- Geliştirilen çoklu robot görev dağılım algoritması birebir görev atama ilkesiyle çalışmaktadır. Çizelge 5.2'de 10 adet kumaşın eş zamanlı olarak işlenmesi için oluşturulan görevleri en uygun şekilde dağıtıldığında elde edilen sonuçlar gösterilmiştir. Elde edilen sonuçlara göre 10 adet kumaşın lojistik işlemleri 516 saniyede toplam 5.658,87 joule enerji harcanarak gerçekleştirilmiştir. Görevler rastgele atandığında bu 10 adet kumaş 622 saniyede ve toplam 7.296,3 joule enerji harcanarak teslimat noktasına ulaştırılmıştır. Çizelge 5.3'te ise dört adet senaryonun hem önerilen sistemle hem de rastgele atamayla oluşan maliyetlerin karşılaştırılması verilmiştir. Bu çizelgede görüldüğü üzere, oluşturulan tüm senaryolarda, önerilen görev dağılım yöntemi rastgele görev atama yöntemine göre hem harcanan enerji hem de işlem zamanı açısından daha iyi sonuçlar üretmiştir.
- Önerilen konumlama sisteminde robotun harita üzerine bulunduğu yer doğru bir şekilde tespit edilmeye çalışılmıştır. Konumlama sisteminin hassasiyetini tespit etmek

için 360x360 cm<sup>2</sup>'lik bir alanda 100 farklı ve uzamsal noktası bilinen konumda ölçümler yapılmıştır. Yapılan ölçümler sonunda konumlama hassasiyeti  $\pm 12,56$  cm olarak ölçülmüştür. Ma vd. (L. Ma vd., 2019)'de yaptığı çalışmada konumlama hassasiyeti 24 cm'dir. Zeng vd. (Zeng, Chen, Li, & Tan, 2019)'de yaptığı çalışmada ise konumlama hassasiyeti yaklaşık olarak 22 cm olarak ölçülmüştür. Bu çalışmalarla kıyaslandığında önerilen konum tespit sistemi oldukça başarılıdır.

- Gerçekleştirilen otonom hareket modelleme ve navigasyon sistemi temelde odometriyi kullanarak konum tespit sisteminden gelen veriye göre bulunduğu konumu güncelleyen bir yaklaşıma sahiptir. Navigasyon sistemi için yapılan 5 adet deneysel çalışma sonunda elde edilen sonuçlar Çizelge 5.5'te sunulmuştur. Çizelge 5.5'te görüleceği üzere yalnızca odometri kullanıldığında fiziksel etmenlerin teker enkoderiyle yapılan ölçümleri olumsuz etkilemesi sonucunda yoldan sapma miktarı oldukça fazladır. Ancak önerilen yöntemde bu durumla karşılaşılsa dahi konumlama sisteminden gelen bilgiyle konum güncellemesi yapıldığı için tekrar belirlenen yol planına doğru girilmesi sağlanmıştır. Önerilen sistem odometrik ölçümle karşılaştırıldığında hareket sonunda hedefe varma başarımı %90,46 daha iyidir. Bu şu anlama gelmektedir: Eğer odometrik hata oranını %100 olarak değerlendirirsek, önerilen yöntemde hata oranı %9,55'tir.

## 6.2 Öneriler

Tez çalışmasında çoklu robotların otonom bina içi lojistikte uyum içinde çalışması geliştirilen sistem kendi içinde birçok çalışma alanı barındırmaktadır. Bu çalışma alanları ayrı ayrı değerlendirildiğinde gelecekte şu çalışmalar yapılabilir:

- Önerilen konumlama sistemi her ne kadar yüksek hassasiyete sahip olsa da endüstriyel işletmelerde üretimin aksamaması için daha yüksek hassasiyetle konumlama ihtiyacı doğabilir. Gelecekte daha hassas bir konumlama yapabilmek için lidar sensörler, görüntü işleme ve QR-kod tabanlı hibrit bir konumlama sistemi üzerinde çalışılması planlanmaktadır.
- Yol planlama üzerine graf tabanlı algoritmalar yerine meta-sezgisel algoritmaların kullanılması düşünülmektedir. Ancak yerel minimumda sıkışma probleminden dolayı optimum yol planının elde edilemeyebilmektedir. Gelecekteki çalışmalarda yerel

minimumda sıkışma problemi üzerine yapılacak çalışmalar meta-sezgisel algoritmaları ön plana çıkaracaktır.

- Tez kapsamında robotların lojistik işlemlerinin koordinasyonu üzerine çalışılmıştır. Ancak lojistik süreçleriyle birebir ilişkili olan, robotların ürünleri yükleme ve bırakma işlemleri üzerine çalışılmamıştır. Gelecekteki çalışmalarda robotların otonom olarak yük bırakma veya yük alma problemleri de ele anılacaktır.
- Robotların bilinmeyen hareketli engellerden kaçınma problemi de başlı başına bir çalışma alanıdır. Robotların endüstriyel ortamda gezinirken sistem tarafından bilinmeyen hareketli veya hareketsiz engelleri (insan, araç, malzeme) otomatik olarak tanıyabilmesini sağlayacak bir çalışma yapılması planlanmaktadır. Çalışmada görüntü işleme yapılarak derin öğrenmeyle nesnenin tanınmasını ve lidar sensörlerle üç boyutlu algılama yapılmasını sağlayacak yöntemler üzerinde çalışılacaktır.

## KAYNAKLAR

- Afrin, M., Jin, J., Rahman, A., Tian, Y. C., & Kulkarni, A. (2019). Multi-objective resource allocation for Edge Cloud based robotic workflow in smart factory. *Future Generation Computer Systems*, 97, 119–130. <https://doi.org/10.1016/j.future.2019.02.062>
- Ajeil, F. H., Ibraheem, I. K., Azar, A. T., & Humaidi, A. J. (2020). Grid-based mobile robot path planning using aging-based ant colony optimization algorithm in static and dynamic environments. *Sensors (Switzerland)*, 20(7). <https://doi.org/10.3390/s20071880>
- Akbar, A., Lewis, P. R., & Wanner, E. (2020). A Self-Aware and Scalable Solution for Efficient Mobile-Cloud Hybrid Robotics. *Frontiers in Robotics and AI*, 7. <https://doi.org/10.3389/frobt.2020.00102>
- Al-Hourani, A., & Ristic, B. (2020). MapperBot/iSCAN: open-source integrated robotic platform and algorithm for 2D mapping. *International Journal of Intelligent Robotics and Applications*, 4(1), 44–56. <https://doi.org/10.1007/s41315-020-00118-3>
- Alves, R., Silva De Morais, J., & Yamanaka, K. (2020). Cost-effective indoor localization for autonomous robots using kinect and wifi sensors. *Inteligencia Artificial*, 23(65), 33–55. <https://doi.org/10.4114/intartif.vol23iss65pp33-55>
- Autonomous Mobile Robots Market Research Report: By Offering, End User - Global Industry Size, Share and Trends Analysis, Forecast to 2030.* (2020). Retrieved from <https://www.researchandmarkets.com/reports/5010701/autonomous-mobile-robots-market-research-report>
- Badreldin, M., Hussein, A., & Khamis, A. (2013). A Comparative Study between Optimization and Market-Based Approaches to Multi-Robot Task Allocation. *Advances in Artificial Intelligence, 2013*, 1–11. <https://doi.org/10.1155/2013/256524>
- Bae, J., & Chung, W. (2018). A Heuristic for Path Planning of Multiple Heterogeneous Automated Guided Vehicles. *International Journal of Precision Engineering and Manufacturing*, 19(12), 1765–1771. <https://doi.org/10.1007/s12541-018-0205-x>
- Bae, Y. (2019). Robust Localization for Robot and IoT Using RSSI. *Energies*, 12(11), 2212. <https://doi.org/10.3390/en12112212>
- Baek, D., Hwang, M., Kim, H., & Kwon, D. S. (2018). Path Planning for Automation of Surgery Robot based on Probabilistic Roadmap and Reinforcement Learning. *2018 15th International Conference on Ubiquitous Robots, UR 2018, (C)*, 342–347. <https://doi.org/10.1109/URAI.2018.8441801>
- Bayat, F., Najafinia, S., & Aliyari, M. (2018). Mobile robots path planning: Electrostatic potential field approach. *Expert Systems with Applications*, 100, 68–78. <https://doi.org/10.1016/j.eswa.2018.01.050>
- Ben-Ari, M., & Mondada, F. (2018). Robotic Motion and Odometry. In *Elements of Robotics* (pp. 63–93). Cham: Springer International Publishing.

[https://doi.org/10.1007/978-3-319-62533-1\\_5](https://doi.org/10.1007/978-3-319-62533-1_5)

- Beokhaimook, C., Ohno, K., Westfechtel, T., Nishinoma, H., Tamura, R., & Tadokoro, S. (2020). Cyber-enhanced canine suit with wide-view angle for three-dimensional LiDAR SLAM for indoor environments. *Advanced Robotics*, *34*(11), 715–729. <https://doi.org/10.1080/01691864.2020.1772104>
- Booranawong, A., Sengchuai, K., & Jindapetch, N. (2019). Implementation and test of an RSSI-based indoor target localization system: Human movement effects on the accuracy. *Measurement: Journal of the International Measurement Confederation*, *133*, 370–382. <https://doi.org/10.1016/j.measurement.2018.10.031>
- Chen, J., Yang, Y., & Wu, Y. (2011). Multi-robot task allocation based on the modified particle swarm optimization algorithm. In *Proceedings - 2011 7th International Conference on Natural Computation, ICNC 2011* (Vol. 3, pp. 1744–1749). <https://doi.org/10.1109/ICNC.2011.6022303>
- Chopra, S., Notarstefano, G., Rice, M., & Egerstedt, M. (2017). A Distributed Version of the Hungarian Method for Multirobot Assignment. *IEEE Transactions on Robotics*, *33*(4), 932–947. <https://doi.org/10.1109/TRO.2017.2693377>
- Choudhury, B. B., & Biswal, B. B. (2011). A PSO based multi-robot task allocation. *International Journal of Computational Vision and Robotics*, *2*(1), 49–61. <https://doi.org/10.1504/IJCVR.2011.039356>
- Chowdhury, M., & Maier, M. (2017). Collaborative Computing for Advanced Tactile Internet Human-to-Robot (H2R) Communications in Integrated FiWi Multirobot Infrastructures. *IEEE Internet of Things Journal*, *4*(6), 2142–2158. <https://doi.org/10.1109/JIOT.2017.2761599>
- Clark, C. M. (2005). Probabilistic Road Map sampling strategies for multi-robot motion planning. *Robotics and Autonomous Systems*, *53*(3–4), 244–264. <https://doi.org/10.1016/j.robot.2005.09.002>
- Costley, A., & Christensen, R. (2020). Landmark Aided GPS-Denied Navigation for Orchards and Vineyards. In *2020 IEEE/ION Position, Location and Navigation Symposium, PLANS 2020* (pp. 987–995). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/PLANS46316.2020.9110130>
- Dai, X., Long, S., Zhang, Z., & Gong, D. (2019). Mobile robot path planning based on ant colony algorithm with a\* heuristic method. *Frontiers in Neurorobotics*, *13*. <https://doi.org/10.3389/fnbot.2019.00015>
- Dakulovi, M., & Petrovi, I. (2011). Two-way D\* algorithm for path planning and replanning. *Robotics and Autonomous Systems*, *59*(5), 329–342. <https://doi.org/10.1016/j.robot.2011.02.007>
- Dewangan, R. K., Shukla, A., & Godfrey, W. W. (2017). Survey on prioritized multi robot path planning. *2017 IEEE International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials, ICSTM 2017 - Proceedings*, (August), 423–428. <https://doi.org/10.1109/ICSTM.2017.8089197>

- Digani, V., Hsieh, M. A., Sabbatini, L., & Secchi, C. (2018). Coordination of multiple AGVs: a quadratic optimization method. *Autonomous Robots*, (February 2017), 1–17. <https://doi.org/10.1007/s10514-018-9730-9>
- Digani, V., Hsieh, M. A., Sabbatini, L., & Secchi, C. (2019). Coordination of multiple AGVs: a quadratic optimization method. *Autonomous Robots*, 43(3), 539–555. <https://doi.org/10.1007/s10514-018-9730-9>
- Digiampaolo, E., & Martinelli, F. (2018). A Robotic System for Localization of Passive UHF-RFID Tagged Objects on Shelves. *IEEE Sensors Journal*, 18(20), 8558–8568. <https://doi.org/10.1109/JSEN.2018.2865339>
- Dönmez, E., & Kocamaz, A. F. (2019). Design of Mobile Robot Control Infrastructure Based on Decision Trees and Adaptive Potential Area Methods. *Iranian Journal of Science and Technology - Transactions of Electrical Engineering*. <https://doi.org/10.1007/s40998-019-00228-0>
- DÖNMEZ, E., & KOCAMAZ, A. F. (2019). Çoklu Hedeflerin Çoklu Robotlara Paylaştırılması İçin Bir Yük Dengeleme Sistemi. *Bitlis Eren Üniversitesi Fen Bilimleri Dergisi*. <https://doi.org/10.17798/bitlisfen.467757>
- Draganjac, I., Miklic, D., Kovacic, Z., Vasiljevic, G., & Bogdan, S. (2016). Decentralized Control of Multi-AGV Systems in Autonomous Warehousing Applications. *IEEE Transactions on Automation Science and Engineering*, 13(4), 1433–1447. <https://doi.org/10.1109/TASE.2016.2603781>
- Du, L. zhen, Ke, S., Wang, Z., Tao, J., Yu, L., & Li, H. (2019). Research on multi-load AGV path planning of weaving workshop based on time priority. *Mathematical Biosciences and Engineering*, 16(4), 2277–2292. <https://doi.org/10.3934/mbe.2019113>
- Eberhart, R., & Kennedy, J. (1995). New optimizer using particle swarm theory. In *Proceedings of the International Symposium on Micro Machine and Human Science* (pp. 39–43). IEEE. <https://doi.org/10.1109/mhs.1995.494215>
- Estefo, P., Simmonds, J., Robbes, R., & Fabry, J. (2019). The Robot Operating System: Package reuse and community dynamics. *Journal of Systems and Software*, 151, 226–242. <https://doi.org/10.1016/j.jss.2019.02.024>
- Fard, M. D., & Sarjoughian, H. S. (2021). A RESTful framework design for componentizing the water evaluation and planning (WEAP) system. *Simulation Modelling Practice and Theory*, 106, 102199. <https://doi.org/10.1016/j.simpat.2020.102199>
- Fellan, A., Schellenberger, C., Zimmermann, M., & Schotten, H. D. (2018). Enabling Communication Technologies for Automated Unmanned Vehicles in Industry 4.0. *9th International Conference on Information and Communication Technology Convergence: ICT Convergence Powered by Smart Intelligence, ICTC 2018*, 171–176. <https://doi.org/10.1109/ICTC.2018.8539695>
- Fu, B., Chen, L., Zhou, Y., Zheng, D., Wei, Z., Dai, J., & Pan, H. (2018). An improved A\* algorithm for the industrial robot path planning with high success rate and short

- length. *Robotics and Autonomous Systems*, 106, 26–37.  
<https://doi.org/10.1016/j.robot.2018.04.007>
- Fu, M., Zhu, M., Yang, Y., Song, W., & Wang, M. (2020). LiDAR-based vehicle localization on the satellite image via a neural network. *Robotics and Autonomous Systems*, 129. <https://doi.org/10.1016/j.robot.2020.103519>
- Fu, W., Liu, R., Wang, H., Ali, R., He, Y., Cao, Z., & Qin, Z. (2020). A Method of Multiple Dynamic Objects Identification and Localization Based on Laser and RFID. *Sensors*, 20(14), 3948. <https://doi.org/10.3390/s20143948>
- Gao, M., Yu, M., Guo, H., & Xu, Y. (2019). Mobile robot indoor positioning based on a combination of visual and inertial sensors. *Sensors (Switzerland)*, 19(8). <https://doi.org/10.3390/s19081773>
- Gao, Q., & Cheng, H. (2018). Design of a CloudROS enabled Mobile Robot. In *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (pp. 2487–2492). IEEE. <https://doi.org/10.1109/ROBIO.2018.8665101>
- Gerkey, B. P., & Mataric, M. J. (2002). Sold!: Auction methods for multirobot coordination. *IEEE Transactions on Robotics and Automation*, 18(5), 758–768. <https://doi.org/10.1109/TRA.2002.803462>
- Ghazal, M., Basmaji, T., Yaghi, M., Alkhedher, M., Mahmoud, M., & El-Baz, A. S. (2020). Cloud-Based Monitoring of Thermal Anomalies in Industrial Environments Using AI and the Internet of Robotic Things. *Sensors (Basel, Switzerland)*, 20(21). <https://doi.org/10.3390/s20216348>
- Giubilato, R., Chiodini, S., Pertile, M., & Debei, S. (2019). An evaluation of ROS-compatible stereo visual SLAM methods on a nVidia Jetson TX2. *Measurement: Journal of the International Measurement Confederation*, 140, 161–170. <https://doi.org/10.1016/j.measurement.2019.03.038>
- Giuliano, R., Cardarilli, G. C., Cesarini, C., Di Nunzio, L., Fallucchi, F., Fazzolari, R., ... Vizzari, A. (2020). Indoor Localization System Based on Bluetooth Low Energy for Museum Applications. *Electronics*, 9(6), 1055. <https://doi.org/10.3390/electronics9061055>
- Golenbiewski, J., & Tewolde, G. (2020). Wi-Fi Based Indoor Positioning and Navigation System (IPS/INS) (pp. 1–7). Institute of Electrical and Electronics Engineers (IEEE). <https://doi.org/10.1109/iemtronics51293.2020.9216376>
- Guerrero, J., Valero, Ó., & Oliver, G. (2017). Toward a Possibilistic Swarm Multi-robot Task Allocation: Theoretical and Experimental Results. *Neural Processing Letters*, 46(3), 881–897. <https://doi.org/10.1007/s11063-017-9647-x>
- Guo, S., Fang, T. T., Song, T., Xi, F. F., & Wei, B. G. (2018). Tracking and localization for omni-directional mobile industrial robot using reflectors. *Advances in Manufacturing*, 6(1), 118–125. <https://doi.org/10.1007/s40436-018-0216-y>
- Hsu, C. C., Yeh, S. S., & Hsu, P. Lo. (2016). Particle filter design for mobile robot localization based on received signal strength indicator. *Transactions of the Institute*

*of Measurement and Control*, 38(11), 1311–1319.  
<https://doi.org/10.1177/0142331215586615>

- Ismail, A. H., & Terashima, K. (2018). Prediction of WiFi signal using kalman filter for fingerprinting-based mobile robot wireless positioning system. *Journal of Telecommunication, Electronic and Computer Engineering*, 10(1–15), 17–21.
- Jin, L., Li, S., La, H. M., Zhang, X., & Hu, B. (2019). Dynamic task allocation in multi-robot coordination for moving target tracking: A distributed approach. *Automatica*, 100, 75–81. <https://doi.org/10.1016/j.automatica.2018.11.001>
- Jondhale, S. R., Deshpande, R. S., Walke, S. M., & Jondhale, A. S. (2017). Issues and challenges in RSSI based target localization and tracking in wireless sensor networks. In *International Conference on Automatic Control and Dynamic Optimization Techniques, ICACDOT 2016* (pp. 594–598). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/ICACDOT.2016.7877655>
- Jose, K., & Pratihari, D. K. (2016). Task allocation and collision-free path planning of centralized multi-robots system for industrial plant inspection using heuristic methods. *Robotics and Autonomous Systems*, 80, 34–42. <https://doi.org/10.1016/j.robot.2016.02.003>
- Keung, K. L., Lee, C. K. M., Ji, P., & Ng, K. K. H. (2020). Cloud-Based Cyber-Physical Robotic Mobile Fulfillment Systems: A Case Study of Collision Avoidance. *IEEE Access*, 8, 89318–89336. <https://doi.org/10.1109/ACCESS.2020.2992475>
- Khamis, A., Hussein, A., & Elmogy, A. (2015). Multi-robot task allocation: A review of the state-of-the-art. *Studies in Computational Intelligence*, 604, 31–51. [https://doi.org/10.1007/978-3-319-18299-5\\_2](https://doi.org/10.1007/978-3-319-18299-5_2)
- Khattak, S., Papachristos, C., & Alexis, K. (2020). Keyframe-based thermal–inertial odometry. *Journal of Field Robotics*, 37(4), 552–579. <https://doi.org/10.1002/rob.21932>
- Koenig, S., & Likhachev, M. (2002). *D\* Lite*. Retrieved from [www.aai.org](http://www.aai.org)
- Korsah, G. A., Stentz, A., & Dias, M. B. (2013). A comprehensive taxonomy for multi-robot task allocation. *The International Journal of Robotics Research*, 32(12), 1495–1512. <https://doi.org/10.1177/0278364913496484>
- Lajoie, P. Y., Ramtoula, B., Chang, Y., Carlone, L., & Beltrame, G. (2020). DOOR-SLAM: Distributed, Online, and Outlier Resilient SLAM for Robotic Teams. *IEEE Robotics and Automation Letters*, 5(2), 1656–1663. <https://doi.org/10.1109/LRA.2020.2967681>
- Lee, D. H. (2018). Resource-based task allocation for multi-robot systems. *Robotics and Autonomous Systems*, 103, 151–161. <https://doi.org/10.1016/j.robot.2018.02.016>
- Lee, T. J., Kim, C. H., & Cho, D. I. D. (2019). A Monocular Vision Sensor-Based Efficient SLAM Method for Indoor Service Robots. *IEEE Transactions on Industrial Electronics*, 66(1), 318–328. <https://doi.org/10.1109/TIE.2018.2826471>

- Liu, H., Liu, S., & Zheng, K. (2018). A Reinforcement Learning-Based Resource Allocation Scheme for Cloud Robotics. *IEEE Access*, 6, 17215–17222. <https://doi.org/10.1109/ACCESS.2018.2814606>
- Liu, W., Caruso, D., Ilg, E., Dong, J., Mourikis, A. I., Daniilidis, K., ... Engel, J. (2020). TLIO: Tight Learned Inertial Odometry. *IEEE Robotics and Automation Letters*, 5(4), 5653–5660. <https://doi.org/10.1109/lra.2020.3007421>
- Ljesnjanin, M., & Velagic, J. (2009). A market based approach for complex task allocation for wireless network based multi-robot system. In *ICAT 2009 - 2009 22nd International Symposium on Information, Communication and Automation Technologies*. <https://doi.org/10.1109/ICAT.2009.5348428>
- Llopis-Albert, C., Rubio, F., & Valero, F. (2019). Fuzzy-set qualitative comparative analysis applied to the design of a network flow of automated guided vehicles for improving business productivity. *Journal of Business Research*, (June 2018), 0–1. <https://doi.org/10.1016/j.jbusres.2018.12.076>
- Low, E. S., Ong, P., & Cheah, K. C. (2019). Solving the optimal path planning of a mobile robot using improved Q-learning. *Robotics and Autonomous Systems*, 115, 143–161. <https://doi.org/10.1016/j.robot.2019.02.013>
- Lu, S., Xu, C., Zhong, R. Y., & Wang, L. (2018). A passive RFID tag-based locating and navigating approach for automated guided vehicle. *Computers and Industrial Engineering*, 125(xxxx), 628–636. <https://doi.org/10.1016/j.cie.2017.12.026>
- Ma, L., Liu, M., Wang, H., Yang, Y., Wang, N., & Zhang, Y. (2019). Wallsense: Device-free indoor localization using wall-mounted uhf rfid tags. *Sensors (Switzerland)*, 19(1). <https://doi.org/10.3390/s19010068>
- Ma, X., Jiao, Z., Wang, Z., & Panagou, D. (2016). Decentralized prioritized motion planning for multiple autonomous UAVs in 3D polygonal obstacle environments. *2016 International Conference on Unmanned Aircraft Systems, ICUAS 2016*, 292–300. <https://doi.org/10.1109/ICUAS.2016.7502596>
- Ma, Y., Tian, C., & Jiang, Y. (2019). A Multitag Cooperative Localization Algorithm Based on Weighted Multidimensional Scaling for Passive UHF RFID. *IEEE Internet of Things Journal*, 6(4), 6548–6555. <https://doi.org/10.1109/JIOT.2019.2907771>
- Mac, T. T., Copot, C., Tran, D. T., & De Keyser, R. (2016). Heuristic approaches in robot path planning: A survey. *Robotics and Autonomous Systems*, 86, 13–28. <https://doi.org/10.1016/j.robot.2016.08.001>
- Magnago, V., Palopoli, L., Buffi, A., Tellini, B., Motroni, A., Nepa, P., ... Fontanelli, D. (2020). Ranging-Free UHF-RFID Robot Positioning through Phase Measurements of Passive Tags. *IEEE Transactions on Instrumentation and Measurement*, 69(5), 2408–2418. <https://doi.org/10.1109/TIM.2019.2960900>
- Mellado, M., Vendrell, E., Crespo, A., López, P., Garbajosa, J., Lomba, C., ... Mayerhofer, R. (1999). Application of a real time expert system platform for flexible autonomous transport in industrial production. *Computers in Industry*, 38(2), 187–200. [https://doi.org/10.1016/S0166-3615\(98\)00118-3](https://doi.org/10.1016/S0166-3615(98)00118-3)

- Mousavi, M., Yap, H. J., Musa, S. N., Tahriri, F., & Md Dawal, S. Z. (2017). Multi-objective AGV scheduling in an FMS using a hybrid of genetic algorithm and particle swarm optimization. *PLoS ONE*, *12*(3), 1–24. <https://doi.org/10.1371/journal.pone.0169817>
- Mousavi, S., Afghah, F., Ashdown, J. D., & Turck, K. (2019). Use of a quantum genetic algorithm for coalition formation in large-scale UAV networks. *Ad Hoc Networks*, *87*, 26–36. <https://doi.org/10.1016/j.adhoc.2018.11.008>
- Narayanan, V., Phillips, M., & Likhachev, M. (n.d.). *Anytime Safe Interval Path Planning for Dynamic Environments*.
- Nedjah, N., De Mendonça, R. M., & De Macedo Mourelle, L. (2015). Pso-based distributed algorithm for dynamic task allocation in a robotic swarm. In *Procedia Computer Science* (Vol. 51, pp. 326–335). Elsevier B.V. <https://doi.org/10.1016/j.procs.2015.05.250>
- Negenborn, R. (2003). *Robot Localization and Kalman Filters On finding your position in a noisy world*.
- Nunes, E., Manner, M., Mitiche, H., & Gini, M. (2017). A taxonomy for task allocation problems with temporal and ordering constraints. *Robotics and Autonomous Systems*, *90*, 55–70. <https://doi.org/10.1016/j.robot.2016.10.008>
- Oh, G., Kim, Y., Ahn, J., & Choi, H. L. (2016). PSO-based Optimal Task Allocation for Cooperative Timing Missions. *IFAC-PapersOnLine*, *49*(17), 314–319. <https://doi.org/10.1016/j.ifacol.2016.09.054>
- Okumus, F., & Kocamaz, A. F. (2019). Cloud based indoor navigation for ros-enabled automated guided vehicles. In *2019 International Conference on Artificial Intelligence and Data Processing Symposium, IDAP 2019*. Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/IDAP.2019.8875993>
- Okumuş, F., & Kocamaz, A. F. (2019). Comparing Path Planning Algorithms for Multiple Mobile Robots. *2018 International Conference on Artificial Intelligence and Data Processing, IDAP 2018*, (2018), 1–4. <https://doi.org/10.1109/IDAP.2018.8620785>
- Orozco-Rosas, U., Montiel, O., & Sepúlveda, R. (2019). Mobile robot path planning using membrane evolutionary artificial potential field. *Applied Soft Computing*, *77*, 236–251. <https://doi.org/10.1016/j.asoc.2019.01.036>
- Palaskar, P., Palkar, R., & Tawari, M. (2014). Wi-Fi Indoor Positioning System Based on RSSI Measurements from Wi-Fi Access Points –A Tri-lateration Approach. *International Journal of Scientific & Engineering Research*, *5*(4). Retrieved from <http://www.ijser.org>
- Raja, P., & Pugazhenthii, S. (2012). Optimal path planning of mobile robots: A review. *International Journal of Physical Sciences*, *7*(9), 1314–1320. <https://doi.org/10.5897/IJPS11.1745>
- Riazuelo, L., Civera, J., & Montiel, J. M. M. (2014). C2TAM: A Cloud framework for cooperative tracking and mapping. *Robotics and Autonomous Systems*, *62*(4), 401–

413. <https://doi.org/10.1016/j.robot.2013.11.007>

- Riazuelo, Luis, Tenorth, M., Di Marco, D., Salas, M., Gálvez-López, D., Mösenlechner, L., ... Montiel, J. M. M. (2015). RoboEarth Semantic Mapping: A Cloud Enabled Knowledge-Based Approach. *IEEE Transactions on Automation Science and Engineering*, 12(2), 432–443. <https://doi.org/10.1109/TASE.2014.2377791>
- Rostami, S. M. H., Sangaiah, A. K., Wang, J., & Liu, X. (2019). Obstacle avoidance of mobile robots using modified artificial potential field algorithm. *Eurasip Journal on Wireless Communications and Networking*, 2019(1). <https://doi.org/10.1186/s13638-019-1396-2>
- Schillinger, P., Bürger, M., & Dimarogonas, D. V. (2018). Simultaneous task allocation and planning for temporal logic goals in heterogeneous multi-robot systems. *International Journal of Robotics Research*, 37(7), 818–838. <https://doi.org/10.1177/0278364918774135>
- Shamsfakhr, F., Palopoli, L., Fontanelli, D., Motroni, A., & Buffi, A. (2020). Robot Localisation using UHF-RFID Tags for Industrial IoT Applications. In *2020 IEEE International Workshop on Metrology for Industry 4.0 and IoT, MetroInd 4.0 and IoT 2020 - Proceedings* (pp. 659–664). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/MetroInd4.0IoT48571.2020.9138210>
- Shi, W., Du, J., Cao, X., Yu, Y., Cao, Y., Yan, S., & Ni, C. (2019). IKULDAS: An improved kNN-based UHF RFID indoor localization algorithm for directional radiation scenario. *Sensors (Switzerland)*, 19(4). <https://doi.org/10.3390/s19040968>
- Silveira, L., Maffei, R. Q., Botelho, S. S. C., Drews, P. L., Bicho, A. de L., & Duarte Filho, N. L. (2012). Space D\*. *Journal of the Brazilian Computer Society*, 18(4), 363–373. <https://doi.org/10.1007/s13173-012-0071-9>
- Solovey, K., & Halperin, D. (2013). K-color multi-robot motion planning. In *Springer Tracts in Advanced Robotics* (Vol. 86, pp. 191–207). Springer Verlag. [https://doi.org/10.1007/978-3-642-36279-8\\_12](https://doi.org/10.1007/978-3-642-36279-8_12)
- Stentz, A. (n.d.). Optimal and efficient path planning for partially-known environments. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation* (pp. 3310–3317). IEEE Comput. Soc. Press. <https://doi.org/10.1109/ROBOT.1994.351061>
- Suemitsu, I., Izui, K., Yamada, T., Nishiwaki, S., Noda, A., & Nagatani, T. (2016). Simultaneous optimization of layout and task schedule for robotic cellular manufacturing systems. *Computers and Industrial Engineering*, 102, 396–407. <https://doi.org/10.1016/j.cie.2016.05.036>
- Trigui, S., Cheikhrouhou, O., Koubaa, A., Zarrad, A., & Youssef, H. (2018). An analytical hierarchy process-based approach to solve the multi-objective multiple traveling salesman problem. *Intelligent Service Robotics*, 11(4), 355–369. <https://doi.org/10.1007/s11370-018-0259-8>
- Tsarouchi, P., Michalos, G., Makris, S., Athanasatos, T., Dimoulas, K., & Chryssolouris, G. (2017). On a human–robot workplace design and task allocation system.

*International Journal of Computer Integrated Manufacturing*, 30(12), 1272–1279.  
<https://doi.org/10.1080/0951192X.2017.1307524>

- Turner, J., Meng, Q., Schaefer, G., Whitbrook, A., & Soltoggio, A. (2018). Distributed Task Rescheduling with Time Constraints for the Optimization of Total Task Allocations in a Multirobot System. *IEEE Transactions on Cybernetics*, 48(9), 2583–2597. <https://doi.org/10.1109/TCYB.2017.2743164>
- Varma, A. K., Karjee, J., Rath, H. K., & Pal, A. (2020). Dynamic Path Selection for Cloud-based Multi-Hop Multi-Robot Wireless Networks. *IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India)*, 37(1), 98–107. <https://doi.org/10.1080/02564602.2019.1566031>
- Velagapudi, P., Sycara, K., & Scerri, P. (n.d.). *Decentralized prioritized planning in large multirobot teams*.
- Wan, J., Chen, B., Wang, S., Xia, M., Li, D., & Liu, C. (2018). Fog Computing for Energy-Aware Load Balancing and Scheduling in Smart Factory. *IEEE Transactions on Industrial Informatics*, 14(10), 4548–4556. <https://doi.org/10.1109/TII.2018.2818932>
- Wan, J., Yi, M., Li, D., Zhang, C., Wang, S., & Zhou, K. (2016). Mobile services for customization manufacturing systems: An example of industry 4.0. *IEEE Access*, 4, 8977–8986. <https://doi.org/10.1109/ACCESS.2016.2631152>
- Wang, L., Liu, M., & Meng, M. Q. H. (2017). A Hierarchical Auction-Based Mechanism for Real-Time Resource Allocation in Cloud Robotic Systems. *IEEE Transactions on Cybernetics*, 47(2), 473–484. <https://doi.org/10.1109/TCYB.2016.2519525>
- Wang, Z., Yang, S., Xiang, X., Vasilijević, A., Mišković, N., & Nađ, Đ. (2021). Cloud-based mission control of USV fleet: Architecture, implementation and experiments. *Control Engineering Practice*, 106. <https://doi.org/10.1016/j.conengprac.2020.104657>
- Wattananavin, T., Sengchuai, K., Jindapetch, N., & Booranawong, A. (2020). A Comparative Study of RSSI-Based Localization Methods: RSSI Variation Caused by Human Presence and Movement. *Sensing and Imaging*, 21(1), 31. <https://doi.org/10.1007/s11220-020-00296-1>
- Wei, C., Hindriks, K. V., & Jonker, C. M. (2016). Dynamic task allocation for multi-robot search and retrieval tasks. *Applied Intelligence*, 45(2), 383–401. <https://doi.org/10.1007/s10489-016-0771-5>
- Wu, Haibing, Tao, B., Gong, Z., Yin, Z., & Ding, H. (2019). A Fast UHF RFID Localization Method Using Unwrapped Phase-Position Model. *IEEE Transactions on Automation Science and Engineering*, 16(4), 1698–1707. <https://doi.org/10.1109/TASE.2019.2895104>
- Wu, Hao, Wu, X., & Tian, G. (2018). Indoor robot localization based on single RFID tag. *Artificial Life and Robotics*, 23(3), 373–379. <https://doi.org/10.1007/s10015-018-0449-7>
- Xu, F., Li, H., Pun, C. M., Hu, H., Li, Y., Song, Y., & Gao, H. (2020). A new global best

- guided artificial bee colony algorithm with application in robot path planning. *Applied Soft Computing Journal*, 88. <https://doi.org/10.1016/j.asoc.2019.106037>
- Xu, J., He, J., Zhang, Y., Xu, F., & Cai, F. (2016). A distance-based maximum likelihood estimation method for sensor localization in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2016. <https://doi.org/10.1155/2016/2080536>
- Xun, L., & Hong-xu, M. (2008). Particle swarm optimization based multi-robot task allocation using wireless sensor network. In *Proceedings of the 2008 IEEE International Conference on Information and Automation, ICIA 2008*. <https://doi.org/10.1109/ICINFA.2008.4608201>
- Yan, H., Hua, Q., Wang, Y., Wei, W., & Imran, M. (2017). Cloud robotics in Smart Manufacturing Environments: Challenges and countermeasures. *Computers and Electrical Engineering*, 63, 56–65. <https://doi.org/10.1016/j.compeleceng.2017.05.024>
- Yan, Y., & Qiang, Z. (2008). Logisitcs and supply chain integration model of port in transportation networks. In *Proceedings of the International Conference on Information Management Proceedings of the International Conference on Information Management, Innovation Management and Industrial Engineering, ICIII 2008* (Vol. 2, pp. 402–406). <https://doi.org/10.1109/ICIII.2008.144>
- Yao, W., Qi, N., Liu, Y., Xu, S., & Du, D. (2019). Homotopic Approach for Robot Allocation Optimization Coupled with Path Constraints. *IEEE Robotics and Automation Letters*. <https://doi.org/10.1109/LRA.2019.2944060>
- Yen, H. C., Wang, C. C., & Chou, C. F. (2020). Orientation constraints for Wi-Fi SLAM using signal strength gradients. *Autonomous Robots*. <https://doi.org/10.1007/s10514-020-09914-z>
- Zeng, Y., Chen, X., Li, R., & Tan, H. Z. (2019). UHF RFID Indoor Positioning System with Phase Interference Model Based on Double Tag Array. *IEEE Access*, 7, 76768–76778. <https://doi.org/10.1109/ACCESS.2019.2921560>
- Zhang, Q. bin, Wang, P., & Chen, Z. hai. (2019). An improved particle filter for mobile robot localization based on particle swarm optimization. *Expert Systems with Applications*, 135, 181–193. <https://doi.org/10.1016/j.eswa.2019.06.006>
- Zhang, J., Zhou, Y., & Zhang, Y. (2018). Multi-objective Robot Path Planning based on Bare Bones Particle Swarm Optimization with Crossover Operation. In *2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)* (pp. 330–335). IEEE. <https://doi.org/10.1109/FSKD.2018.8687230>
- Zhong, X., Tian, J., Hu, H., & Peng, X. (2020). Hybrid Path Planning Based on Safe A\* Algorithm and Adaptive Window Approach for Mobile Robot in Large-Scale Dynamic Environment. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 99(1), 65–77. <https://doi.org/10.1007/s10846-019-01112-z>
- Zhu, Z., Tang, B., & Yuan, J. (2017). Multirobot task allocation based on an improved particle swarm optimization approach. *International Journal of Advanced Robotic*

*Systems*, 14(3), 172988141771031. <https://doi.org/10.1177/1729881417710312>

Zitouni, F., Maamri, R., & Harous, S. (2019). FA–QABC–MRTA: a solution for solving the multi-robot task allocation problem. *Intelligent Service Robotics*, 12(4), 407–418. <https://doi.org/10.1007/s11370-019-00291-w>



## ÖZGEÇMİŞ

**Ad-Soyad** : Fatih Okumuş  
**Doğum Tarihi ve Yeri** : 19.02.1983 / Arapgir  
**E-posta** : fatih.okumus@inonu.edu.tr

### ÖĞRENİM DURUMU:

- **Lisans** : 2002-2006, Fırat Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği
- **Yüksek Lisans** : 2014-2016, İnönü Üniversitesi, Bilgisayar Mühendisliği A.B.D, Yazılım Bilim Dalı
- **Doktora** : 2016-2020, İnönü Üniversite, Bilgisayar Mühendisliği A.B.D, Yazılım Bilim Dalı

### MESLEKİ DENEYİM:

- 2006-2008 Obisis Yazılım Ltd.Şti.'de çalıştı.
- 2008-2010 Fonet Bilgi Teknolojileri A.Ş.'de çalıştı.
- 2011-2013 Türksat A.Ş.'de çalıştı.
- 2013-2018 İnönü Üniversitesinde çalıştı.
- 2018'den beri Malatya Turgut Özal Üniversitesinde çalışıyor.

### DOKTORA TEZİNDEN TÜRETİLEN ÇALIŞMALAR

- Okumuş, F., Dönmez, E., & Kocamaz, A. F. (2020). A Cloudware Architecture for Collaboration of Multiple AGVs in Indoor Logistics: Case Study in Fabric Manufacturing Enterprises. *Electronics*, 9(12), 2023. <https://doi.org/10.3390/electronics9122023>
- Okumus, F., & Kocamaz, A. F. (2019). Cloud based indoor navigation for ros-enabled automated guided vehicles. 2019 International Conference on Artificial Intelligence and Data Processing Symposium, IDAP 2019. Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/IDAP.2019.8875993>
- Okumuş, F., & Fatih, A. (2019). A Multiple Robot Path Planning Approach For Industrial Indoor Logistics. 2019, IRDC
- Okumuş, F., & Kocamaz, A. F. (2019). Comparing Path Planning Algorithms for Multiple Mobile Robots. 2018 International Conference on Artificial Intelligence and Data Processing, IDAP 2018, (2013), 1–4. <https://doi.org/10.1109/IDAP.2018.8620785>

- Okumuş, F., & Kocamaz, A.F. (2019). Exploring the Feasibility of a Multifunctional Software Platform for Cloud Robotics. 2018 International Conference on Artificial Intelligence and Data Processing, IDAP 2018, 1–4. <https://doi.org/10.1109/IDAP.2018.8620865>
- ÜSİP, Araştırmacı, “ Endüstriyel Mobil Robot için Karekod Tabanlı Konumlama ve Kontrol Altyapısı”, Üniversite Sanayi İşbirliği Projesi, Proje Kodu: FKP-2020-1993, Yürütülen Kuruluş: Çalık Denim A.Ş. Ar-Ge Merkezi, Destek Alınan Kuruluş: İnönü Üniversitesi BAP Birimi, 2020
- ÜSİP, Araştırmacı, “Endüstriyel Lojistik Amaçlı Bina İçi Otonom Robot Sistemi”, Üniversite Sanayi İşbirliği Projesi, Proje Kodu: FKP-2019-1598, Yürütülen Kuruluş: Çalık Denim A.Ş. Ar-Ge Merkezi, Destek Alınan Kuruluş: İnönü Üniversitesi BAP Birimi, 2019
- Tez çalışmasında geliştirilen sistemin açık kaynak kodları:  
Bulut robot yönetim sistemi: <https://github.com/fatihokumus/CloudRoboticProject>  
Simülasyon yazılımı: <https://github.com/fatihokumus/ClientSimulation>