**KOCAELİ ÜNİVERSİTESİ**
**FEN BİLİMLERİ ENSTİTÜSÜ**

**MEKATRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

**DOKTORA TEZİ**

# İÇ MEKANLARDA OTONOM BİR ŞEKİLDE HAREKET EDEBİLEN BİR MOBİL ROBOTUN TASARIM, İMALAT VE KONTROLÜ

**SUAT KARAKAYA**

**KOCAELİ 2019**

**KOCAELİ ÜNİVERSİTESİ**

**FEN BİLİMLERİ ENSTİTÜSÜ**

**MEKATRONİK MÜHENDİSLİĞİ**

**ANABİLİM DALI**

**DOKTORA TEZİ**

# İÇ MEKANLARDA OTONOM BİR ŞEKİLDE HAREKET EDEBİLEN BİR MOBİL ROBOTUN TASARIM, İMALAT VE KONTROLÜ

**SUAT KARAKAYA**

| | | |
|---|---|---|
| **Prof. Dr. Hasan OCAK** | | |
| **Danışman,** | **Kocaeli Üniversitesi** | ..... |
| **Prof. Dr. Zafer BİNGÜL** | | |
| **Jüri Üyesi,** | **Kocaeli Üniversitesi** | ..... |
| **Prof. Dr. Serdar KÜÇÜK** | | |
| **Jüri Üyesi,** | **Kocaeli Üniversitesi** | ..... |
| **Doç. Dr. Şeref Naci ENGİN** | | |
| **Jüri Üyesi,** | **Yıldız Teknik Üniversitesi** | ..... |
| **Dr. Öğr. Üyesi Barış BORU** | | |
| **Jüri Üyesi,** | **Sakarya Üniversitesi** | ..... |

**Tezin Savunulduğu Tarih: 09.01.2019**

## ÖNSÖZ VE TEŞEKKÜR

Bu çalışmanın tüm aşamalarında yardımını esirgemeyen değerli danışman hocam Prof. Dr. Hasan OCAK'a ve değerli çalışma arkadaşım Dr. Gürkan KÜÇÜKYILDIZ'a desteklerinden dolayı çok teşekkür ederim. Tezin geliştirilmesinde maddi destek sağlayan TÜBİTAK'a teşekkür ederim.

Bu süreçte her zaman yanımda olan eşim Esra Gizem KARAKAYA'ya, beni bugünlere getiren anneme ve babama, her zaman destek veren ablama sonsuz teşekkürlerimi sunarım.

Ocak – 2019 Suat KARAKAYA

## İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR ..... i
İÇİNDEKİLER ..... ii
ŞEKİLLER DİZİNİ ..... iv
TABLOLAR DİZİNİ ..... viii
SİMGELER VE KISALTMALAR DİZİNİ ..... ix
ÖZET ..... xi
ABSTRACT ..... xii
GİRİŞ ..... 1
1. GENEL BİLGİLER ..... 3
1.1. Mobil Robotlarla İlgili Temel Problemler ..... 11
1.1.1. Yol planlama problemleri ..... 12
1.1.2. Konumlandırma problemleri ..... 15
1.1.3. Haritalandırma problemleri ..... 18
1.1.4. Benzetim ortamı ile ilgili problemler ..... 19
1.2. Literatür Taraması ..... 20
1.2.1. Statik yol planlama yöntemleri ..... 20
1.2.2. Dinamik yol planlama yöntemleri ..... 24
1.2.3. İç mekân konumlandırma sistemleri ..... 33
1.2.4. Mobil robot benzetim arayüzleri ..... 36
1.3. Tezin Literatüre Katkısı ..... 38
2. YOL PLANLAMA ..... 42
2.1. Statik Yol Planlayıcı ..... 45
2.2. Dinamik Yol Planlayıcı ..... 58
2.2.1. Açıklık tespiti algoritması ..... 61
2.2.2. Lokal hedef tespiti metodu ..... 67
2.2.3. Çarpışma engelleme metodu ..... 69
2.2.4. Hareketli engellerin tespiti ..... 72
2.3. DWA Lokal Yol Planlayıcısı ..... 75
3. KONUMLANDIRMA SİSTEMİ ..... 88
3.1. Pasif Etiket Tasarımı ..... 89
3.2. Kamera ve IR Projeksiyon Grubu ..... 90
3.3. Etiketlerin Tavana Yerleşimi ..... 91
3.4. Kamera Kalibrasyonu ..... 92
3.5. EKF ve İmge İşleme Tabanlı Konumlandırma Sistemi ..... 94
4. MOBİL ROBOT PLATFORMU ..... 103
4.1. Sistemin Mekanik Yapısı ve İmalatı ..... 103
4.2. Sistemin Kontrol Birimi ve Gömülü Yazılımı ..... 105
4.3. Mobil Robot Sistem Mimarisi ..... 107
5. MOBİL ROBOT BENZETİM ARAYÜZÜ ..... 109
5.1. Benzetim Arayüzünün Yapısı ..... 109
5.2. Nokta Stabilizasyonu Problemi ..... 116
5.2.1. Kullanılan sistemin kinematik modeli ..... 116
5.2.2. MPC yöntemi ..... 118

5.2.3. EEDT yöntemi ..... 119
5.3. Harita Çıkarma Eklentisi ..... 121
5.4. AMCL Eklentisi ..... 125
6. SONUÇLAR VE ÖNERİLER ..... 134
6.1. Yol Planlama ..... 134
6.1.1. Gerçek dünya deneyi ..... 134
6.1.2. Açıklık tespiti ve optimal lokal hedef tespiti ..... 139
6.1.3. DWA ve önerilen dinamik metodun karşılaştırılması ..... 140
6.1.4. Hareketli engellerin tespiti ..... 149
6.2. İç Mekân Konumlandırma Sistemi ..... 151
6.3. Benzetim Arayüzü Çıktıları ..... 154
6.4. Öneriler ..... 160
KAYNAKLAR ..... 161
EKLER ..... 171
KİŞİSEL YAYIN VE ESERLER ..... 173
ÖZGEÇMİŞ ..... 176

## ŞEKİLLER DİZİNİ

Şekil 1.1. Servis robotlarının öngörülen pazar hacmi ..... 4
Şekil 1.2. Ev temizliği için tasarlanmış bir robot ..... 4
Şekil 1.3. Otonom çim biçme makinesi: BigMow ..... 5
Şekil 1.4. Endüstriyel bir otonom mobil robot ..... 5
Şekil 1.5. Endüstriyel bir tarım robotu ..... 6
Şekil 1.6. Manyetik şerit takibi uygulaması ..... 9
Şekil 1.7. Optik şerit takibi uygulaması ..... 9
Şekil 1.8. Mıknatıs deseni uygulaması için sembolik gösterim ..... 10
Şekil 1.9. Lazer tabanlı navigasyon modelleri ..... 10
Şekil 1.10. Otonom mobil robot sistemlerine ait genel şema ..... 12
Şekil 1.11. Tangent-bug lokal yol planlayıcısı ..... 24
Şekil 1.12. Genetik algoritma gen gösterimi ..... 26
Şekil 1.13. Genetik algoritma operasyonları ..... 26
Şekil 2.1. Hibrit yol planlama metodunun genel akış şeması ..... 43
Şekil 2.2. LIDAR ölçüm gösterimi ..... 44
Şekil 2.3. Jarvis mesafe dönüşüm algoritması pseudo kodu ..... 48
Şekil 2.4. Jarvis yaklaşımı ile planlanan örnek statik yol ..... 50
Şekil 2.5. Dijkstra algoritması pseudo kodu ..... 51
Şekil 2.6. Dijkstra yaklaşımı için tanımlanan bir problem ..... 52
Şekil 2.7. Örnek statik engel haritası ..... 54
Şekil 2.8. Üretilen maliyet haritası ..... 54
Şekil 2.9. α katsayısının çok büyük seçilmesi durumu ..... 55
Şekil 2.10. α katsayısının çok küçük seçilmesi durumu ..... 55
Şekil 2.11. Gradient descent algoritması ile planlanan statik yol ..... 56
Şekil 2.12. Greedy algoritma ile planlanan statik yol ..... 57
Şekil 2.13. Görünürlük tanımı ..... 59
Şekil 2.14. Statik yol üzerinde lokal hedef seçimi ..... 59
Şekil 2.15. Dinamik engel tarafından bloke edilen birincil nokta ..... 61
Şekil 2.16. Ön işlem sonucu tespit edilen açıklıklar ..... 62
Şekil 2.17. Açıklık revizyonu pseudo kodu ..... 63
Şekil 2.18. Açıklıkların başlangıç düzeni ..... 64
Şekil 2.19. CCW revizyonu ile köşe güncelleme ..... 64
Şekil 2.20. Açıklıkların son düzeni ..... 65
Şekil 2.21. Duvar tipi sonlanan açıklık revizyonu ..... 65
Şekil 2.22. Örtüşen açıklık tespiti ..... 66
Şekil 2.23. Açıklık tespiti yöntemi ile ilgili örnek senaryo ..... 66
Şekil 2.24. Açıklık tespiti ve lokal hedef belirleme işlemi akış şeması ..... 67
Şekil 2.25. Yerel minimum durumu ..... 68
Şekil 2.26. Optimal lokal hedef tespiti ..... 69
Şekil 2.27. Çarpışma kontrolü ..... 70
Şekil 2.28. Çarpışma kontrolü örnek senaryo engel profili ..... 70
Şekil 2.29. Çarpışma kontrolü örnek senaryo LIDAR görüntüsü ..... 71
Şekil 2.30. Engellerin LIDAR eksenlerinde modellenmesi ..... 72

Şekil 2.31. Hareket kestirimi şeması ..... 73
Şekil 2.32. Ortak engellerin eşleştirilmesi ..... 74
Şekil 2.33. DWA lokal planlayıcı pseudo kodu ..... 76
Şekil 2.34. DWA yol planlayıcı şeması ..... 77
Şekil 2.35. DWA ile planlanmış örnek bir yol ve anlık VR görüntüleri ..... 77
Şekil 2.36. Lokal bölgedeki LIDAR görüntüsü ve lokal yol ..... 78
Şekil 2.37. Lokal bölgedeki maliyet haritası ..... 79
Şekil 2.38. Lokal bölgedeki engel haritası ..... 79
Şekil 2.39. DWA açısal kontrol: başlangıç konumu ..... 81
Şekil 2.40. DWA açısal kontrol: anlık konum ..... 81
Şekil 2.41. DWA açısal kontrol: hedef konumu ..... 82
Şekil 2.42. DWA açısal kontrol: $|g_h l_h|$-$k_t$ grafiği ..... 82
Şekil 2.43. DWA açısal kontrol: $|g_h l_h|$-$p_g$ grafiği ..... 83
Şekil 2.44. DWA açısal kontrol benzetim uygulaması ..... 83
Şekil 2.45. DWA benzetimi: açısal kontrol aktif ..... 84
Şekil 2.46. Açısal kontrol aktif iken statik harita ..... 85
Şekil 2.47. DWA benzetimi: açısal kontrol pasif ..... 85
Şekil 2.48. Açısal kontrol pasif iken oluşan statik harita ..... 86
Şekil 2.49. Açısal kontrol senaryosu başlangıç durumu ..... 86
Şekil 2.50. Açısal kontrol senaryosu sonuç durumu ..... 87
Şekil 2.51. Açısal kontrol uygulaması: statik harita ..... 87
Şekil 3.1. İç-mekân konumlandırma sistemi geliştirme adımları ..... 88
Şekil 3.2. Etiket tasarımı ..... 89
Şekil 3.3. Kızılötesi led projeksiyonu ve kamera ..... 90
Şekil 3.4. Yansıtıcı etiketlerin tavandaki yerleşim düzeni ..... 91
Şekil 3.5. Kalibrasyon işlemi için kullanılan Öklid uzaklıkları ..... 93
Şekil 3.6. İç mekân konumlandırma sistemi şeması ..... 94
Şekil 3.7. Etiketler üzeride doğrusallık kontrolü ..... 95
Şekil 3.8. Konumlandırma sisteminde tanımlı eksenler ..... 96
Şekil 3.9. Konumlandırma sistemi iş akışı ..... 97
Şekil 3.10. Diferansiyel sürüş modelinin sembolik gösterimi ..... 98
Şekil 4.1. Mobil robot platformu katı modeli ..... 103
Şekil 4.2. Tekerlek mekanizması ve bağlantı elemanlarının katı modeli ..... 104
Şekil 4.3. Mekanik imalat ve montaj sonrası görünüm ..... 104
Şekil 4.4. Elektronik ana kartın montaj sonrası görünümü ..... 106
Şekil 4.5. Sistemin genel mimarisi ..... 107
Şekil 4.6. Merkez PC algoritma mimarisi ..... 107
Şekil 5.1. Engel listesi arayüzü ..... 110
Şekil 5.2. VR arayüzü ..... 111
Şekil 5.3. Örnek VR arayüzü ..... 111
Şekil 5.4. Örnek LA penceresi ..... 112
Şekil 5.5. Mobil robotun Kartezyen koordinat sistemi ..... 117
Şekil 5.6. Stabilize yörünge ve elde edilen girişler ..... 119
Şekil 5.7. EEDT ve Jarvis karşılaştırması ..... 120
Şekil 5.8. EEDT yöntemi ile planlanan örnek yol ..... 121
Şekil 5.9. Engellenmiş bir harita üzerinde stabilize edilmiş yörünge ..... 121
Şekil 5.10. Harita çıkarma işlemi başlangıç durumu ..... 122
Şekil 5.11. Harita çıkarma işlemi ara adım-1 ..... 122
Şekil 5.12. Harita çıkarma işlemi ara adım-2 ..... 123

Şekil 5.13. Harita çıkarma işlemi son durum VR arayüzü ..... 123
Şekil 5.14. Çıkarılan harita ..... 124
Şekil 5.15. Çıkarılan harita ve 3 boyutlu ortam ölçülerinin karşılaştırması ..... 124
Şekil 5.16. AMCL konum kestirimi işlem adımları ..... 126
Şekil 5.17. AMCL nesnesi başlangıç değerleri için örnek tanımlama ..... 127
Şekil 5.18. Başlangıç konumu ve parçacıkların dağılımı ..... 128
Şekil 5.19. 29. güncelleme adımı ve parçacıklar ..... 128
Şekil 5.20. 44. güncelleme adımı ve parçacıklar ..... 128
Şekil 5.21. 64. güncelleme adımı ve parçacıklar ..... 129
Şekil 5.22. 71. güncelleme adımı ve parçacıklar ..... 129
Şekil 5.23. Sonuç adımı ve parçacıkların dağılımı ..... 129
Şekil 5.24. Konumlandırma penceresi ..... 130
Şekil 5.25. Hesaplanan konum verilerinin grafiği ..... 130
Şekil 5.26. Global konumlandırma başlangıç parçacık dağılımı ..... 132
Şekil 5.27. Global konumlandırma 5. güncelleme parçacık dağılımı ..... 132
Şekil 5.28. Global konumlandırma 13. güncelleme parçacık dağılımı ..... 133
Şekil 5.29. Global konumlandırma 67. güncelleme parçacık dağılımı ..... 133
Şekil 5.30. Global konumlandırma 122. güncelleme parçacık dağılımı ..... 133
Şekil 6.1. Uzak bilgisayar arayüzü ..... 135
Şekil 6.2. Gerçek zamanlı deney engel yerleşimi ..... 136
Şekil 6.3. Gerçek zamanlı deney ve benzetim arayüzü karşılaştırması ..... 137
Şekil 6.4. Benzetim arayüzü x ekseni konum hatası ..... 138
Şekil 6.5. Benzetim arayüzü y ekseni konum hatası ..... 138
Şekil 6.6. Açıklık tespiti çıktıları VR arayüzü ..... 139
Şekil 6.7. Açıklık tespiti çıktıları LA görüntüsü ..... 140
Şekil 6.8. Dışbükey haritada DWA uygulaması ..... 142
Şekil 6.9. Dışbükey haritada önerilen metodun uygulaması ..... 142
Şekil 6.10. Dışbükey haritada DWA ve önerilen metodun oluşturduğu yol ..... 143
Şekil 6.11. Kompleks haritada DWA uygulaması ..... 144
Şekil 6.12. Kompleks haritada önerilen metodun uygulaması ..... 145
Şekil 6.13. Kompleks haritada DWA uygulaması için oluşan yol ..... 146
Şekil 6.14. Kompleks haritada önerilen metot uygulaması için oluşan yol ..... 146
Şekil 6.15. İçbükey haritada önerilen metodun uygulaması ..... 147
Şekil 6.16. İçbükey koşullarda DWA uygulaması ..... 147
Şekil 6.17. İçbükey haritada önerilen metodun oluşturduğu yol ..... 148
Şekil 6.18. İçbükey haritada DWA uygulaması için oluşan yol ..... 148
Şekil 6.19. Hareketli engel algılama pasif durumu ..... 150
Şekil 6.20. Hareketli engel algılama aktif durumu ..... 150
Şekil 6.21. Metot uygulanmadığında oluşan dinamik yörünge ..... 150
Şekil 6.22. Metot uygulandığında oluşan dinamik yörünge ..... 151
Şekil 6.23. Dinamik yörüngelerin karşılaştırılması ..... 151
Şekil 6.24. EKF destekli konumlandırma sisteminin deneysel sonucu ..... 153
Şekil 6.25. Benzetim arayüzü sonuçları-1 ..... 154
Şekil 6.26. Benzetim arayüzü sonuçları-2 ..... 155
Şekil 6.27. Benzetim arayüzü sonuçları-3 ..... 156
Şekil 6.28. Benzetim arayüzü sonuçları-4 ..... 156
Şekil 6.29. Benzetim arayüzü sonuçları-5 ..... 157
Şekil 6.30. Benzetim arayüzü sonuçları-6 ..... 158
Şekil 6.31. Benzetim arayüzü sonuçları-7 ..... 159

Şekil 6.32. Benzetim arayüzü sonuçları-8..... 160

## TABLOLAR DİZİNİ

Tablo 1.1. İç mekân konumlandırma sistemlerinin karşılaştırılması ..... 35
Tablo 2.1. Jarvis mesafe dönüşüm algoritması tarama pencereleri ..... 46
Tablo 2.2. Jarvis yaklaşımı ile planlanan yolun mesafe dönüşüm matrisi ..... 49
Tablo 2.3. Greedy algoritmanın işlem adımları ..... 49
Tablo 2.4. Dijkstra algoritması işlem adımları ..... 53
Tablo 2.5. Dijkstra algoritması kapalı ve açık listelerin değişimi ..... 53
Tablo 2.6. Düğümlerin nihai maliyetleri ..... 53
Tablo 2.7. GD ve greedy algoritmalarının yol planlama çıktıları ..... 57
Tablo 2.8. Dinamik yol planlama notasyonu ..... 60
Tablo 2.9. DWA açısal kontrol yol takip parametresi ..... 80
Tablo 3.1. Etiket tasarımında kullanılan notasyon ..... 89
Tablo 3.2. Etiket veri tabanı formatı ..... 91
Tablo 3.3. Eksenler ve konumlar için kullanılan notasyon ..... 96
Tablo 3.4. Kinematik model için kullanılan notasyon ..... 99
Tablo 3.5. EKF denklem setleri ..... 101
Tablo 5.1. VR ve LA sembollerinin açıklamaları ..... 112
Tablo 5.2. VR arayüz mesajları ve açıklamaları ..... 113
Tablo 5.3. Fonksiyonların ortak giriş-çıkış parametreleri ..... 113
Tablo 6.1. 2 boyutlu benzetim arayüzü notasyonu ..... 137
Tablo 6.2. Dışbükey labirent testi parametreleri ..... 141
Tablo 6.3. Kompleks harita testi parametreleri ..... 145
Tablo 6.4. İçbükey labirent testi parametreleri ..... 147
Tablo 6.5. Dinamik yörüngelerin uzunlukları ..... 151
Tablo 6.6. İç-mekân konumlandırma sistemi ölçüm ve kestirim hataları ..... 153

## SİMGELER VE KISALTMALAR DİZİNİ

$c_x$ : Piksel genişliği
$c_y$ : Piksel yüksekliği
d : Mesafe dönüşüm fonksiyonu
$d_{min}$ : En yakın engelin kontrol çizgisine uzaklığı
$d_{tol}$ : İzin verilen en düşük uzaklık
f : Optimizasyon fonksiyonu
g : Açıklık
$h_g$ : Global hedef konum vektörü
$h_l$ : Lokal hedef konum vektörü
Ic : Ani dönme merkezi
$k_t$ : Yörünge takip parametresi
l : Açıklık sol köşe noktası
L : LIDAR ölçüm kümesi
m : Düğüm maliyeti fonksiyonu
$p_c$ : Takip modunda en yakın statik yol noktası
$p_L$ : Sensör konum vektörü
$P^*$ : Hareketli engellerin konum matrisi
Q : Süreç hatası kovaryans matrisi
r : Açıklık sağ köşe noktası
R : Ölçüm hatası kovaryans matrisi
$r_L$ : LIDAR maksimum ölçüm limiti
t : Güvenli lokal hedef
$u^*$ : Optimal giriş vektörü
vc : Ortalama doğrusal hız
$v_l$ : Sol teker hızı
$v_{max}$ : Maksimum hız
$v_{min}$ : Minimum hız
vr : Sağ teker hızı
w : Açısal hız
$x_p$ : Kutupsal durum vektörü
$\alpha$ : Gradient descent çözüm katsayısı
$\Delta_\theta$ : Açısal değişim
$\Phi$ : MPC optimizasyon fonksiyonu
$\Omega$ : Çalışma uzayı

### Kısaltmalar

AL : Açık liste
AMCL : Adaptive Monte Carlo Localization (Monte Carlo Konumlama)
BLE : Bluetooth Low Energy (Bluetooth Düşük Enerjisi)
CCW : Counter Clock-Wise (Saat Tersi Yönü)
CW : Clock-Wise (Saat Yönü)

DC : Direct Current (Doğru Akım)
DWA : Dynamic Window Approach (Dinamik Pencere Yaklaşımı)
EEDT : Exact Euclidian Distance Transform (Tam Öklid Mesafe Dönüşümü)
EKF : Extended Kalman Filter (Genişletilmiş Kalman Filtresi)
FSPF : Fast Sampling Plane Filtering (Hızlı Örnekleme Düzlemi Filtrelemesi)
GA : Genetik Algoritma
GD : Gradient Descent (Azalan Gradyan)
GNSS : Global Navigation Satellite System (Küresel Uydu Navigasyon Sistemi)
GPS : Global Positioning System (Küresel Konumlama Sistemi)
ID : Identification Number (Kimlik Numarası)
IMU : Inertial Measurement Unit (Ataletsel Ölçüm Birimi)
IPS : Indoor Positioning System (İç-Mekân Konumlama Sistemi)
IR : Infrared (Kızılötesi)
KL : Kapalı Liste
IBC : İleri Yönlü Bakış Çemberi (Look Ahead Circle)
LCD : Liquid Crystal Display (Sıvı Kristal Ekran)
LDR : Light Dependent Resistor (Işık Tabanlı Direnç)
LIDAR : Laser Detection and Ranging (Lazer Tabanlı Alan Tarayıcı)
LA : LIDAR Arayüzü
LMS : Least Mean Squares (En Küçük Kareler)
MinBBox : Minimum Bounding Box (En Küçük Kapsayıcı)
MPC : Model Predictive Control (Model Kestirimli Kontrol)
PC : Programmable Computer (Programlanabilir Bilgisayar)
PID : Proportional–Integral–Derivative (Oran-İntegral-Türev)
PRM : Probabilistic Road Map (Olasılıksal Yol Haritası)
PWM : Pulse Width Modulation (Darbe Genişlik Modülasyonu)
RFID : Radio Frequency Identification (Radyo Frekansı Kimliği)
RGB : Red-Green-Blue (Kırmızı-Yeşil-Mavi)
RHC : Reverse Hill Climbing (Tepe İnme Yöntemi)
ROS : Robot Operating System (Robot İşletim Sistemi)
RPM : Revolutions Per Minute (Dakikadaki Dönüş Sayısı)
RRT : Rapidly-Exploring Random Tree (Hızlı Keşfeden Rasgele Ağaç)
RSS : Received Signal Strength (Alınan Sinyal Gücü)
SVM : Support Vector Machine (Destek Vektör Makineleri)
TDoA : Time Difference of Arrival (Geliş Zamanı Farkı)
ToF : Time of Flight (Uçuş Zamanı)
VR : Virtual Reality (Sanal Gerçeklik)
WI-FI : Wireless Fidelity (Kablosuz Alan)

## İÇ MEKÂNLARDA OTONOM BİR ŞEKİLDE HAREKET EDEBİLEN BİR MOBİL ROBOTUN TASARIM, İMALAT VE KONTROLÜ

## ÖZET

İnsan gücünü daha ekonomik kullanma eğilimi askeri, endüstriyel ve kişisel kullanım alanlarında mobil robotların yaygınlaşmasını sağlamıştır. Birbirinden bağımsız pek çok sektörde uygulama alanı bulan mobil robotlar yük taşıma, temizlik, reklam-tanıtım, arazi tarama ve arama-kurtarma gibi çeşitli amaçlarla kullanılmaktadır. Herhangi bir operatöre ihtiyaç duymayan tam-otonom mobil robotlar ve insan müdahalesine açık yarı-otonom sistemler mevcuttur. Karar verme aşamasında öncelikli olarak uzaktan kontrol akışını kabul eden ancak otonom moda alınarak görevi sürdürebilen sistemler yarı otonom sınıfında yer almaktadır. Çalışma ortamına bağlı olarak sürekli değişken engel yapısına sahip, yüksek riskli alanlar içeren ve robot görüş alanı dışında öngörülemez durumlar oluşabilecek ortamlarda sistemin acil müdahaleye izin vermesi istenebilir. Bunun yanı sıra robotun çalışma ortamının statik ve önceden bilinmesi ya da dinamik ve bilinmeyen durumda olması robotun yörünge planlama karakteristiğinde karar verici parametrelerdir. Belirli bir çalışma alanında robota tahsis edilen sabit bir yörünge olması ve bu yörüngenin herhangi bir engelle kesilmemesi durumlarında statik yörünge planlayıcılar; harita bilgisinin olmaması ve/veya sürekli değişen engel yapısına sahip ortamlar için dinamik yörünge planlayıcılar önerilmektedir. Planlanan yörüngenin takibi ve robotun kontrolü için temel bileşenlerden bir diğeri de konumlandırma sistemidir. Dış mekânlarda konumlandırıcı olarak genellikle Global Positioning System (GPS) tabanlı sistemler kullanılmakta olup iç mekânlar için bu yaklaşım uygulanabilir değildir. İç-mekânlarda artan konumlandırma hassasiyetine bağlı olarak yöntemlerin karmaşıklığı ve geliştirme - kurulum maliyetleri artma eğilimindedir. Bu gereksinimler gözetilerek, tez kapsamında, görüntü işleme tabanlı bir iç-mekân konumlandırma sistemi ve dinamik bir yörünge planlayıcı önerilmiştir. Deney platformu olarak üç tekerlekli diferansiyel sürüş mekanizmalı tekerlekli bir mobil robot platformu geliştirilmiştir. MATLAB ortamında çalışan üç boyutlu bir benzetim arayüzü de ortaya koyulmuştur. Robot üzerinde yapılacak testler öncesinde bu arayüz üzerinde sanal testler yapılarak olası durumlar önceden kestirilebilmektedir. Kullanıcı tanımlı robot kinematiği ve yörünge planlayıcıları da destekleyen benzetim aracı, gerçek zamanlı deneylerle oldukça yakın sonuçlar vermekte, bu da gerçek zamanlı çalışmalar öncesinde enerji sarfiyatı ve iş yükünden tasarruf sağlamaktadır. Tüm bunlara ek olarak mobil robot platformu uzak bir kontrol birimi tarafından kontrol edilebilmekte, görev süreçleri ve anlık konumu uzak kontrol arayüzünden gerçek zamanlı olarak izlenebilmektedir. Gerektiğinde otonom görev kesilerek manuel moda alınabilmekte veya bağımsız uzaktan kumanda ile de kumanda edilebilmektedir.

**Anahtar Kelimeler:** Güvenli Sürüş, İç-Mekân Konumlandırma, Mobil Robot, Yol Planlama.

**DESIGN AND CONTROL OF AN INDOOR AUTONOMOUS MOBILE ROBOT**

## ABSTRACT

The tendency to use human power more economically has made mobile robots widespread in military, industrial and personal areas. Mobile robots, which find applications in many independent sectors, are used for various purposes. Mobile robots are used in key areas such as cargo handling, cleaning, advertisement-promotion, field survey and search-and-rescue. In addition to mobile robots controlled by remote control devices, full-autonomous mobile robots that do not require any operator, and semi-autonomous systems with autonomous mobility that are open to human intervention are available. In the decision-making phase, the systems, which accept the remote control flow but continue to work in the autonomous form, are in the semi-autonomous class. Depending on the working environment, it may be considered within this scope that the system permits immediate intervention in environments where sudden risks may occur outside the field of view of the robot. Similarly, the static and predictive behavior of the working environment or its dynamic and unknown state affects the trajectory planning characteristic. Static trajectory planners, where there is a constant trajectory to which a robot is assigned in a particular trajectory and where this trajectory is not blocked; Dynamic trajectory planners are proposed for the absence of map information and/or environments with ever-changing obstacles. One of the basic components for tracking the planned trajectory and controlling the robot is the localization system. Global Positioning System (GPS)-based systems are generally used as localization system in the outdoors, and this approach is low performance and even intermittent for indoor use. As positioning precision increases, costs also increase in interior spaces. By combining all these components, an image processing based indoor localization system and a dynamic trajectory planner have been proposed. As a test platform, a three-wheel mobile robot platform with differential drive mechanism was developed. Within the scope of the thesis, a three-dimensional simulator working in MATLAB environment has been presented. Possible situations can be predicted by performing virtual tests on this interface before the tests to be performed on the robot. Supporting user defined robot kinematics and trajectory planners, the simulator gives very similar results to real-time experiments, which saves energy and workload before real-time work. In addition to this, the mobile robot platform can be controlled by a remote control unit, processes and instant position can be monitored in real time from the remote control interface. If necessary, the autonomous task can be switched to manual mode and the robot can also be controlled by remote control.

**Keywords:** Safe Navigation, Indoor Localization, Mobile Robot, Path Planning.

# GİRİŞ

Robotlar, günümüzde iş hayatının bir parçası olarak görülmekte ve iş yükünü azaltmaktadırlar. Robotik sistemlerin bir kolu olan mobil robotlar gezgin hareket kabiliyetleri sayesine hava, kara ve su ortamlarında geniş yelpazeli bir kullanım alanı bulmuştur. Kara şartlarında çalıştırılan mobil robotlar iç ve dış mekânlarda savunma, arama-kurtarma, reklam-tanıtım, yük taşıma ve temizlik gibi pek çok alanda kullanılmaktadır. Çalışma çevrelerinin çeşitliliği ve sürekli değişken olması; karmaşık kontrol yapıları ve sensör ağları gerektirmektedir. Verilen bir başlangıç noktasından bir hedef konumuna ulaşması için mobil robotların temel yazılım bileşenleri ile desteklenmesi gerekir. Güvenli navigasyonun sağlanması ve olası beklenmedik koşullarda alternatif çözümler üretilmesi için birbiri ile eş zamanlı çalışan yol planlama, hareket kontrolü ve konumlandırma metotları tanımlanmalıdır. Bu tez kapsamında bilinmeyen ortam koşullarına duyarlı bir yol planlama algoritması geliştirilmiştir. Bununla birlikte görüntü işleme tabanlı bir iç-mekân konumlandırma sistemi hazırlanarak üç tekerlekli diferansiyel sürüşlü bir mobil robot platformu üzerinde gerçek dünya deneyleri yapılmıştır. Robotun tam otonom moduna ek olarak uzaktan kumanda ile kontrolü de mümkündür. Bunun yanı sıra ana kontrol bilgisayarı üzerinden robot gerçek zamanlı olarak takip edilebilmekte, gerektiğinde manuel moda geçirilerek uzak bir operatör tarafından kumanda edilebilmektedir. Engel profilindeki lokal değişimlere karşı anlık alternatif güvenli yörünge planlayan modül, hareket kontrol modülü ve konumlandırma modülü eş zamanlı olarak çalıştırılmaktadır. Gerçek dünya deneylerine geçmeden önce zaman ve enerji tasarrufu sağlamak için özgün bir benzetim aracı geliştirilmiştir. Geliştirilen tüm algoritmalar ve mobil robot modelleri üç boyutlu olan benzetim arayüzüne yüklenerek ön testler uygulanmış, istenen kriterler sağlandığında fiziksel robota aynı yazılımlar yüklenmiştir. Gerçek dünya testleri ve benzetim sonuçları arasındaki benzerlik oldukça tatmin edici olup, arayüz üzerinde kullanıcı tanımlı farklı yörünge planlama algoritmaları ve diğer eklentiler geliştirilebilmektedir. Tezin kazanımlarından bir tanesi de eğitim-öğretim ve diğer araştırmalarda kullanılabilecek olan bu benzetim arayüzünün ortaya çıkarılmasıdır. Tez çalışmasının birinci

bölümünde tez kapsamında yapılan çalışma konusu ile ilgili temel problemler, literatür taraması ve bu tezin literatüre olan katkısı ele alınmıştır.

İkinci bölümde yol planlama metotları aktarılmıştır.

Üçüncü bölümde tez kapsamında geliştirilen iç mekân konumlandırma sistemi detaylı olarak açıklanmıştır.

Dördüncü bölümde mobil robot platformunun elektromekanik altyapısı ve kontrol başlıkları aktarılmıştır.

Beşinci bölümde ise tez kapsamında geliştirilen mobil robot benzetim arayüzü hakkında kapsamlı bilgiler aktarılmıştır.

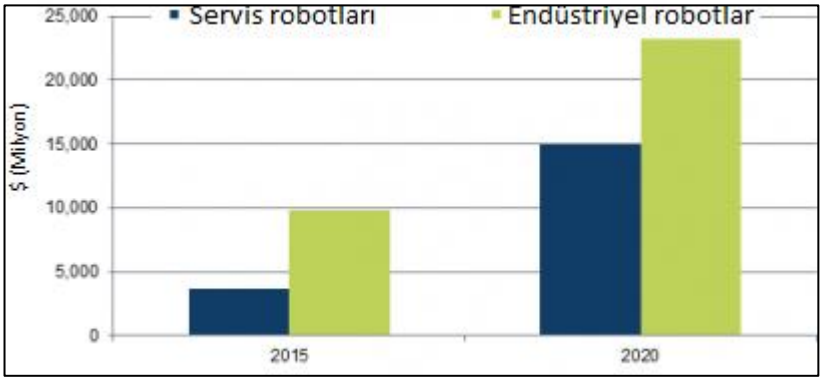
Altıncı ve son bölümde elde edilen sonuçlar ve çalışmanın geliştirilmesi ile ilgili öneriler sunulmuştur.

# 1. GENEL BİLGİLER

Mobil robotlar serbest hareket kabiliyetine sahip ve fiziksel bir sabitleme noktası olmayan robotlardır. Çalışma alanları önceden tanımlanarak su, hava ve kara koşullarına uygun mekanik ve elektronik bileşenlerle donatılmaktadırlar. Çalıştırılacakları fiziksel çevre koşullarına bağlı olarak sensör grupları ve kontrol yapılarına ihtiyaç duyarlar. Tekerlek, palet, ayak, kanat, pervane, yüzgeç ve diğer birçok hareket elemanı ile birlikte fiziksel çevreye adapte edilirler. Su altı çalışmalarından havacılık-uzay sektörüne, askeri araçlardan kişisel kullanıma kadar yayılan kullanım alanları bulunmaktadır. Birbirinden farklı sistemler olsalar da ortak noktaları insan hayatını kolaylaştırmak adına çevreye duyarlı kararlar vermek ve tanımlı görevleri otonom veya manuel olarak gerçekleştirmektir. Yaygınlaşan kullanım ile birlikte otomatik kontrol ve yapay zekâ teknikleri mobil robot sistemleri üzerinde uygulanmaya başlanmış, bu durum mobil robotlarda otonomi kavramını ortaya çıkarmıştır. Karar verme aşamasında tele-operatör kumandası, fiziksel ya da sanal rehberliğe gereksinim duymadan tanımlı görevini başarabilen, hata düzeltme, öğrenme ve problem çözümleme kabiliyetleri olan sistemler otonom olarak tanımlanmaktadır. Mobil robotlar özelinde ise otonomi, çevre algılama sistemleri olan, mevcut pozisyonu ile çevrenin anlık koşulları arasında bir sonraki aksiyonu için karar mekanizmaları bulunan, dinamik ve statik engellerle kısıtlanmış bir çalışma uzayı da tanımlı hedefe ulaşmak için yörünge planlayan sistemler olarak tanımlanmaktadır [1]. Otonominin temel özelliklerinden gelen hata düzeltme ve öğrenme kabiliyetleri otonom mobil robotlar için de tanımlanmalıdır. Ancak güncel literatür ve çalışan sistemler üzerinde yapılan araştırmalara göre bu özellikleri taşımayan ve [1] tarafından ortaya koyulan ölçekleri sağlayan mobil robotlar da tam otonom olarak değerlendirilebilmektedir. Navigasyon kavramı ise hareketli bir objenin konum ve kat ettiği yörüngenin ölçülerek bir hedefe güvenli bir şekilde ilerlemesi olarak tanımlanabilir.

IHS uluslararası piyasa araştırma kuruluşu 2017 yılı birinci çeyreğinde mobil robot ve drone teknolojilerinin robot endüstrisindeki konumu üzerine bir rapor yayınlamıştır [2]. Bu rapor 2015 yılında 3,7 milyar USD hacminde olan mobil robot pazarının 2020 yılında 15 milyar USD düzeyine çıkacağı tahmininde bulunmuştur. Endüstriyel robotlarla karşılaştırma da içeren rapora ait grafik Şekil 1.1'de verilmiştir. Küresel alanda robot endüstrisinde yer alan endüstriyel ve servis amaçlı kullanılan mobil robotların yakın geçmiş ve yakın geleceğe dair istatistik ve kestirim verilerine göre mobil robotların kullanım alanlarının genişlemesi ile birlikte pazar hacminin de yükseldiği görülmektedir.

Yapılan bu araştırmaların sonucunda askeri amaçla kullanılan mobil robotların çoğunlukla Amerika Birleşik Devletleri tarafından geliştirilip pazarlandığı; eğlence, kişisel kullanım ve hizmet sektöründe faaliyet gösteren askeri olmayan mobil robotların ise büyük çoğunlukla Japonya tarafından dünya pazarına sürüldüğü rapor edilmiştir. Avrupa merkezli mobil robot ar-ge çalışmaları ise henüz bu iki ana merkezin gerisindedir. Ülkemizde de durum Avrupa ülkelerinden farklı olmayıp mobil robot alanında ar-ge ve imalat ihtiyacı devam etmektedir.




Şekil 1.1. Servis robotlarının öngörülen pazar hacmi [1]



Şekil 1.2. Ev temizliği için tasarlanmış bir robot [3]

Otonom mobil robotlar reklam, rehberlik, tanıtım, temizlik, eğlence ve eğitim amaçlı alanlarda kullanılmaktadır. Yük taşıyıcı, çekici, forklift gibi görevlerde kullanılmak üzere tasarlanan endüstriyel mobil robotlar da bulunmaktadır. Temizlik ve çim biçme amaçlı tasarlanan otonom mobil robotlar hâlihazırda mevcut olan genel kullanıma uygun örneklerdir. Şekil 1.2'de LG firmasına ait bir otonom temizlik robotu görülmektedir [3]. Ev ortamı için tasarlanmış olan bu tip robotlar, günlük hayatta otonom karakteristikte çalışan ürünlerin ilk örneklerindendir. Şarj ünitesini bulabilen temizlik robotu, iç mekânda mobil olarak hareket ederek temizlik işlevcisini zeminde hareket ettirmektedir.

Çim biçme amaçlı geliştirilen otonom bir robot olan "BigMow", günlük ihtiyaçlara otonom çözüm üretebilen bir diğer otonom mobil robot örneğidir. Şekil 1.3'te bu robotun görünümü verilmiştir [4].



Şekil 1.3. Otonom çim biçme makinesi: BigMow [4]



Şekil 1.4. Endüstriyel bir otonom mobil robot [5]

Endüstriyel ortamlar için geliştirilen mobil robotlar da yaygınlaşmaya başlamıştır. Çoğunlukla yük taşıma amaçlı olarak kullanılan bu robotlar sensör destekli akıllı algoritmalarla donatılarak zamandan ve iş gücünden tasarruf sağlamaktadır. Şekil

1.4’te Fetch Robotics [5] tarafından geliştirilen bir yük taşıma robotu gösterilmiştir. Üç boyutlu kamera ve lazer tarayıcı sensörler ile donatılan bu robot 1500 kilogram taşıma kapasitesine sahiptir. Engel algılama ve yörünge takibi yapabilen robot güvenli bir statik yörünge takibi algoritması koşturmaktadır.

İç mekân koşullarında çalışan mobil robotlar için en temel konulardan bir tanesi konumlandırma işlemidir. Dış ortamlarda çalışan robotlarda GPS sistemleri sıklıkla kullanılmaktadır. Konum ve açısal pozisyon bilgilerini içeren GPS konumlandırıcı sistemleri ile açık alanlarda konumlandırma işlemi göreceli olarak daha kolay gerçekleştirilebilir. Bu teknolojinin gelişimine paralel olarak otonom dış mekân robotları da daha sofistike bir yapı kazanmaya başlamıştır. Örnek olarak tarımda kullanılan ürün hasat robotları insansız olarak işlem yapabilmektedir. Konum bilgisini GPS modülünden alan bu robotlar gövdelerine montajlanmış özel amaçlı uç işlevciler ile belli bir parkuru izleyerek ürünleri koparabilmektedir. Görsel geri besleme amacıyla kamera ile desteklenen bu robotlar görüntü işleme ve navigasyon algoritmalarının eş zamanlı olarak koşturulması ile otonom kimlik kazanmışlardır. Şekil 1.5’te endüstriyel bir tarım robotu görseli yer almaktadır [6].



Şekil 1.5. Endüstriyel bir tarım robotu [6]

Otonom mobil robotların kullanım alanlarına ek olarak gözetleme, arama kurtarma işlevleri sayılabilir. Bu amaçla geliştirilmiş iç mekân otonom mobil robotlara örnek olarak CCS Robotics tarafından geliştirilen Robosentry gösterilebilir [7]. Bu robot sürekli takip edilen bir tesis içerisinde devriye gezme işlemi yaparak, anlık

görüntüleri rapor olarak aktarabilmektedir. İstenirse bir operatör tarafından takip edilebilen robot, manuel olarak da görüntü almaya uygun bir kullanıma sahiptir. Ayrıca kullanım alanı olarak engelli veya yaşlı insanlara rehberlik amaçlı olarak geliştirilen mobil robotlar da mevcuttur [8]. Bu robotlardan Robchair ses tanıma başta olmak üzere diğer birçok insan makine arayüzüne sahip olarak tasarlanmış ve görme engelli bireylerin iç mekân koşullarında yön bulması için geliştirilmiştir. ArtosBot, evde bakım gerektiren özellikle yaşlı bireyleri izlemek ve tanımlı bazı durumları bakım merkezine aktarmak için geliştirilmiştir. Görüntü ve ses aktarabilen ArtosBot, otonom biçimde evde dolaşabilmektedir. Yaşlı bireyi evde takip ederek hangi odada ne kadar durduğunu ve günlük aktivitelerini aktarabilmektedir. Bakım için eğitilen özel tür köpeklerin yerini mobil robotların almasının çok uzak bir gelecek olmadığı bu örnekler ile söylenebilir. Arama kurtarma faaliyetleri gibi riskli alanlar içeren görevlerde arama alanına mobil robotlar yerleştirilerek kurtarma işlemi daha güvenli ve hızlı gerçekleştirilebilir. Robotların görüntü ve ses aktarma özellikleri, yüksek hassasiyet ve doğrulukta konumlandırılabilmeleri sayesinde tehlikeli veya geniş tarama bölgelerinde insan yerine kullanılabilmektedir. Özellikle geniş ve açık arazide yapılacak arama faaliyetlerinde faydalı olabilmektedirler [9]. Dış ortamlarda yapılan uygulamalarda zorlu arazi ve hava koşulları gibi dış etkilere uyumlu donanım ve yöntemler uygulanırken iç mekânlarda zorlayıcı başka faktörler bulunmaktadır. Özellikle konumlandırma sistemleri maliyet ve genel uygulanabilirlik açısından iç mekân haritalarında GPS tabanlı sistemlere göre daha karmaşık olabilmektedir. İnsan, eşya ve diğer hareketli veya sabit nesnelerden kaynaklı dinamik koşulların sık olarak karşılaşıldığı iç mekân koşulları için, uygulamaya özel güvenlik fonksiyonları gerekebilmektedir. Gerçek zamanlı konumlandırma sistemi eklendiğinde işlem hızı oldukça önemli bir kıstas olmaktadır.

Mobil robot sistemlerinin navigasyonu gerçekleştirilirken robotun hangi konumda olduğu ve bulunulan konuma özel görevler varsa bunların gerçekleştirilmesi ile ilgili sürecin başlatılması gibi denetimler robotun üzerine inşa edildiği navigasyon türüne bağlı olarak planlanmaktadır. Otonom mobil robotlar navigasyon süreci devam ederken mutlak konum okuma, açısal duruş hesaplama ve odometre gibi destekleyici konum geri beslemelerine gereksinim duyulmaktadır. Bu bilgilerin elde ediliş biçimine göre navigasyon modelleri iki temel sınıfta incelenebilir: sabit navigasyon

ve serbest navigasyon. Sabit navigasyon modelleri temel olarak mobil robota kısıtlanmış bir referans hat verilmesi ve çalışma bölgesinin bu hat ile sınırlandırılması olarak tanımlanabilir. Sensörler ve bunlara uygun pasif elemanlar kullanılarak nispeten kolay bir şekilde uygulanabilen bu yaklaşımlar her yeni görev tanımı için sistemin kapatılmasını ve yeniden bir hat inşa edilmesini gerektirmektedir. Robotun hareketi hattın kesilmesi durumunda duracağı ve alternatif yörüngelere izin verilmeyeceği için insan faktörü devreye girmek durumunda kalabilmektedir. Bu yaklaşımlara örnek olarak endüktif raylar ve optik şerit uygulamaları verilebilir. Her iki yaklaşıma örnek uygulamalar sırasıyla Şekil 1.6 ve Şekil 1.7'de gösterilmiştir. Endüktif ray uygulamaları zeminden yaklaşık 5 cm derinlikte döşenen aktif sargılardan oluşmaktadır. Mobil robot üzerinde monte edilmiş iletken donanım ile aktif sargılar arasında manyetik alan değişimi kontrol edilerek robotun hatta kalması sağlanmaktadır. Tam tersi olarak robot üzerine bobinler monte edilip zemine iletken bir hat döşenen uygulamalar da bulunmaktadır. Bu yaklaşım kurulum aşamasında avantaj sağlasa da aktif sistemlerin daha güvenilir olduğu bilinmektedir [10]. Optik şerit uygulamalarında zemine yerleştirilen kontrast renkli şerit, sensörler tarafından okunarak şerit takip uygulaması gerçekleştirilmektedir. Zeminde bulunabilecek metal alanlardan da etkilenmemesi bakımından avantajlı olarak değerlendirilebilir ancak çalışma bölgesinde şeritlerin yıpranmaları periyodik olarak kontrol edilmelidir. Her iki sabit navigasyon yaklaşımı için odometre desteği sağlanması olası hat kopukluğu veya optik sorunlar sırasında kısa süreli navigasyon çözümü sunması bakımından önemlidir.

Serbest navigasyon yaklaşımları fiziksel bir referans yörüngeye bağlı olmayan sistemlerden oluşmaktadır. Sistem belli bir yörüngeye tabi olmadığı için çalışma bölgesinin değiştirilmesi ekstra iş gücü gerektirmemekte, yazılımsal olarak sınırlar belirlenebilmektedir. Bu durum esnekliği arttırsa da ciddi teorik arka plan gerektiren uygulamalar içermektedir. Belli ölçülerde kısıtlanmış serbest navigasyon yöntemleri bu teorik gereksinimleri azaltarak sabit navigasyon sistemlerine göre daha esnek ve karmaşıklığı hemen hemen aynı seviyede olan bir karaktere sahip olabilmektedir. Bu kısıtlar fiziksel olarak mıknatıs deseni oluşturulan zeminler ve radyo frekansı tabanlı donanımlarla (RFID) gerçekleştirilebilmektedir. Mıknatıs deseni sistemi zemine pasif mıknatıslar önceden bilinen konumlara yerleştirilerek kurulur. Bu desenler

kullanılarak robotun duruş açısı hesaplanabilmektedir. Şekil 1.8'de sembolik olarak bu sistem verilmiştir. Odometre desteği ile konum hesaplaması da yapılabilen yöntemde odometrenin oldukça güvenilir olması beklenmektedir. Yavaş sistemlerde ve doğrusala yakın hareketli mekanizmalarda uygulanan bir yaklaşım olarak değerlendirilmektedir [10].



Şekil 1.6. Manyetik şerit takibi uygulaması [11]

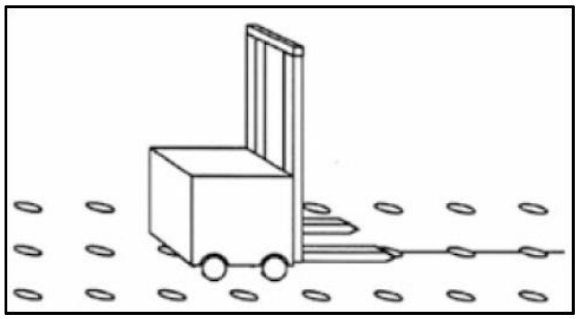


Şekil 1.7. Optik şerit takibi uygulaması [12]

RFID tabanlı sistemler de mıknatıs desenine benzer biçimde yerleştirilir. Alıcı donanımı da robot üzerinde konumlandırılarak robotun her etiketten okuduğu sinyal gücü kullanılarak konum hesaplaması yapılabilmektedir. Düşük maliyeti ile ön plana çıkan bu metot, elektromanyetik ve fiziksel etkiler karşısında etkilenmeye müsait bir yaklaşım olarak bilinmektedir.

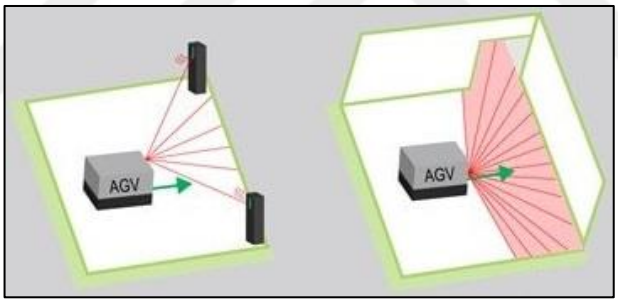
Serbest navigasyon yaklaşımlarından bir diğeri lazer tabanlı sistemlerdir. Bu sistemler pasif etiketler kullanılarak veya sadece ortamdan alınan ölçümlere dayalı yönlendirilen yapılar şeklinde sınıflandırılmaktadır. Etiketlerin kullanıldığı yapıda

lazer ışınlarının her biri benzersiz konumlandırılan etiketlerden yansımaları okunmaktadır.



Şekil 1.8. Mıknatıs deseni uygulaması için sembolik gösterim [10]

En az üç etiket okunabilirse oldukça yüksek hassasiyet ve doğrulukta konumlandırma gerçekleştirilebilmektedir. Bu yaklaşımda alıcı ile etiket arasındaki görüşün daimi olması önemlidir. Kurulum gerektirmesi ve maliyetinin görece çok yüksek olması ise dezavantaj olarak değerlendirilebilir [10].




Şekil 1.9. Lazer tabanlı navigasyon modelleri [14]

Ortamdan alınan lazer mesafe ölçümlerine dayalı yöntemler doğal navigasyon sistemleri olarak bilinmektedir. Bu yaklaşımlar herhangi bir etiket veya kurulum gerektirmediği için çalışma sınırları ölçüm limitleri ile belirlenmektedir. Ortamda bulunan köşe, duvar gibi doğal oluşumların referans alındığı bu sistemler beklendiği gibi etiket tabanlı sistemlere göre daha az kesinlik ile çalışmaktadır. Bu tür navigasyon sitemleri ile kullanılacak mobil robot platformlarında çalışma haritası önceden bilinmelidir. Harita navigasyon öncesinde bir insan ya da insan kontrollü bir robot ile çıkarılabilir veya navigasyon ile eş zamanlı olarak oluşturulabilir. Simultaneous Localization and Mapping (SLAM) [13] yöntemi bu tip navigasyon sistemlerine uygulanarak hem harita çıkarma hem de robot navigasyonu eş zamanlı

olarak gerçekleştirilebilmektedir. Temel olarak odometre, lazer ölçümü ve Kalman filtresini kullanan bu yöntem doğal navigasyon yöntemi olarak değerlendirilmektedir. Ortamda bulunabilecek saydam yüzeyler ve kullanılan sensörün yüksek maliyeti bu yaklaşımın dezavantajlarıdır. Şekil 1.9 üzerinde sembolik olarak lazer tabanlı platformların ölçüm karakteristiği gösterilmiştir. Etiket tabanlı model solda, doğal navigasyon modeli ise sağda verilmiştir.

Lazer sensörlerini temel alan navigasyon sistemleri için alternatif olarak görsel geri beslemeli (kamera kontrollü) ve sonar tabanlı yöntemler önerilmesine karşın özellikle çok geniş endüstriyel ortamlarda veya ışık etkisinin kontrol edilemediği alanlarda bu yaklaşımların lazer tabanlı sistemlerin gerisinde kaldığını gösteren çalışmalar bulunmaktadır [15].

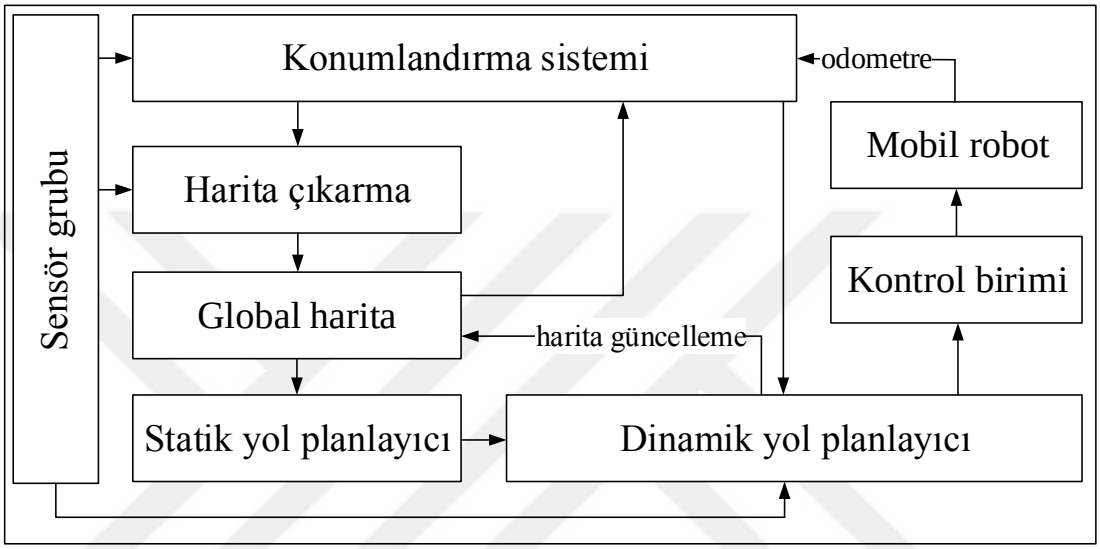
Dış ortamlarda çalışacak olan mobil robot platformları için Global Positioning System (GPS) tabanlı navigasyon yaklaşımı yaygın olarak kullanılmaktadır. Endüstriyel ortamlar ve diğer kapalı alanlarda uydu iletişimi kesileceği için iç mekân navigasyon sistemleri için verimli olmadığı literatürde yapılan çalışmalar ile gösterilmiştir [16]. GPS tabanlı kontrol edilen mobil robot platformları genellikle drone ve su üzerinde çalışan mobil robotlarda tercih edilmektedir.

Tez kapsamında geliştirilen mobil robot platformuna ait navigasyon altyapısının pozisyon ve aksiyon bileşenleri farklı iki kaynaktan elde edilmektedir. Pozisyon bilgisi kamera sistemi ile tespit edilirken anlık durumlarda yapılacak aksiyon lazer sensörü verileri işlenerek karara bağlanmaktadır. Robotun iç mekân koşullarında çalışacağı kabulü ile kullanılan teknik görsel geri besleme ve lazer tabanlı navigasyon sistemlerinin hibrit bir versiyonu olarak değerlendirilmektedir.

## 1.1. Mobil Robotlarla İlgili Temel Problemler

Mobil robot uygulamaları çok disiplinli mühendislik problemleridir. Literatürde problemin modellenmesi ile ilgili genel geçer bir notasyon olmamakla birlikte, otonom mobil robot sistemlerinin kapalı çevrim problemi Şekil 1.10'da verilen şema ile gösterilebilir. Sensör grubu LIDAR, kamera ve benzeri çevre algılama birimlerinden meydana gelmektedir. Tez kapsamında LIDAR ve IR projeksiyon destekli kamera sistemi sensör grubunu oluşturmaktadır. Harita çıkarma bloğu ise

navigasyon gerçekleştirilecek haritanın iki ya da üç boyutlu modellenmesi olarak tanımlanabilir. Bu blok tez kapsamında gerçek zamanlı olarak gerçeklenmemiştir, ancak tez kapsamında geliştirilen mobil robot benzetim arayüzünde harita çıkarma ile ilgili temel bir eklenti sunulmaktadır. Diğer bloklar kendi içinde bir çalışma alanı oluşturmaktadır. İlgili başlıklar altında bu çalışma alanları ile ilgili teorik arka plan sunulacaktır.




Şekil 1.10. Otonom mobil robot sistemlerine ait genel şema

### 1.1.1. Yol planlama problemleri

Mobil robot uygulamalarında temel konulardan birisi robotun izleyeceği yolun planlanmasıdır. Yol planlayıcılar harita modeline, tarama tekniğine ve analiz türüne göre çeşitli sınıflara ayrılmıştır. Her sınıfın uygulamaya özgü kullanım alanları bulunmaktadır. Harita modeline göre statik ve dinamik yol planlama, analiz türüne göre sezgisel ve deterministik yol planlama, tarama tekniğine göre lokal ve global yol planlama yöntemleri olarak alt sınıflara ayrılmaktadır. Statik planlayıcılar sabit engellerin bulunduğu ve zamanla bu engellerin konum ve şeklinin değişmeyeceği varsayımı bulunan ortamlar için uygulanan planlayıcılardır. Bu yaklaşımlar değişen engel profilleri karşısında işlevsel değildir. Dinamik planlayıcılar ise anlık olarak değişebilen engel karakteristiğine sahip ortamlarda yol planlanması için uygulanan tekniklerdir. Bu işlem günlük yaşantıda sık karşılaşılan ortamlar için statik planlayıcılara göre daha uygulanabilir yöntemleri içerir. Anlık engel profili değişimlerine karşı duyarlı olmaları beklenen bu metotlar statik yöntemlere göre

çoğu durumda daha karmaşıktır. Gerçek zamanlı olarak uygulanan dinamik planlayıcılar için işlem süreleri kritik bir önem taşımaktadır. Mobil robotun hareket planında başlangıçta bir kez planlanması yeterli olmayacağı için bu algoritmalar işlem yükü olarak çalışma zamanına işlem yükü olarak etki eder. Sezgisel yaklaşımlar karmaşık problemleri daha basit formatlara indirgeyerek işlem yükünü ve süresini azaltmayı amaçlayan metotlardır. Temelinde gerçek dünya, doğa ve insan mantığının ortaya koyduğu bir takım felsefeler olan bu yaklaşımlarda hepsi ile ortak olmamakla birlikte genellikle bir tarama uzayı, maliyet fonksiyonu ve bir başarım kriteri tanımlanır. Bir çözüm uzayındaki en iyi çözümü garanti etmemekle birlikte, belirlenen bir minimum başarı kriteri sağlandığı takdirde en iyiye yakın çözümler bulmayı garanti eder. Aynı problem için farklı zamanlarda farklı çözümler bulabilir ve bu çözümlerden bazıları deterministik çözümlerle bulunabilecek en iyi çözüm de olabilir. Genetik algoritma, karınca kolonisi ve tabu arama gibi metotlar sezgisel sınıfına girmektedir. Nümerik analiz metotları ile çözüm uzayında kesin çözümleme teknikleri aynı problem için farklı zamanlarda tek çözüm vermektedir. Rassal bir durum söz konusu değildir. Ancak bu yöntemlerin işlem yükleri gerçek zamanlı uygulamalar ve artan çözüm uzayı boyutları için nispeten daha yüksektir. Global planlama yaklaşımları çalışılan haritanın tümü hakkında bilgi sahibi olunmasını gerektiren yöntemlerdir. Lokal planlama yöntemleri ise sensörlerden okunan anlık mesafe ölçümlerinin işlenmesi ve sensör görüş alanı ile kısıtlı bir planlama yapılması temeline dayalı yöntemlerdir. Dijkstra, A* ve Roadmaps gibi yöntemler çeşitli formatlarda gösterilmiş haritalar üzerinde planlanan yolun tamamını hesaplamak için yaygın kullanılan metotlardır. Lokal planlayıcılar ise çoğu durumda çalışma uzayına ait bilginin olmadığı varsayımı ile geliştirilen metotlardır ve hesaplanan anlık hedef koordinatları robotun izlediği yolu oluşturur. Bu yaklaşımlarda global harita hakkında ön bilgi olmaması zorlayıcı bir etkendir ve yol planlayıcı yerel minimum çözümlere zorlanabilir. Bug 0, bug 1, bug 2 ve tangent bug gibi algoritmalar lokal yol planlayıcılara örnek olarak verilebilir. Başlangıç seviyesinden gelişmişe doğru verilen bu sıralamada tangent bug algoritmasında alan tarayıcı sensör kullanımı varsayımı vardır. Temel prensibi tespit edilen objelerin köşe noktalarına yönelmek olan bu metotta hafıza birimi olmaması ve sadece anlık engel profili üzerinde işlem yapıldığı için yerel minimum çözümlere takılma ve sonsuz bir döngüye girme gibi davranışlar gözlenebilir. Diğer bug tabanlı algoritmalar ise daha başlangıç

seviyesinde bir yaklaşım sergileyerek duvar takibi ve buna benzer ilkel bazı davranışlar sergileyerek nihai hedefe ulaşmayı amaçlar. Bug 0, bug 1 ve bug 2 için tanımlanmış çözülemez harita modelleri vardır ve bu harita tipleri ile karşılaşılması durumunda sonsuz döngüye düşmektedirler. Dynamic window approach (DWA) yaklaşımı ise global planlayıcıları lokal bir pencere içerisinde koşturarak her işlem adımında robota tam bir yörünge planlayan bir metottur. Robot ana haritada ilerlerken boyutları tanımlı sanal bir pencere içerisinde lokal düzeyde Dijkstra, A* gibi planlayıcılar koşturulur. Burada tüm harita üzerinde işlem yapılmaz ancak dinamik lokal pencere kendi içerisinde bir alt çalışma uzayı gibi varsayılarak global yörünge planlayıcı çalıştırılır. Her işlem adımında değişmesi muhtemel engel profili gerekli transformasyonlar ile bir alt haritaya dönüştürülür ve tüm haritanın işlenmesini gerektirmez. Pencere boyutu robot için tanımlanan dinamik özelliklerle kısıtlıdır ve bir alan tarayıcı donanımı ya da engel profili ile ilgili görsel bir geri besleme zorunluluğu vardır. Bu metot global planlayıcılara oranla işlem yükünü düşürmektedir. Ancak lokal pencere içerisindeki planlanan yolun başlangıç ve bitiş noktaları arasındaki her hücre işlem yüküne katkı yapmaktadır. Pencere boyutu ve lokal penceredeki engel profili işlem yükü ile doğrudan ilgilidir. Tez kapsamında önerilen dinamik yol planlayıcı lokal bir planlayıcı olup, ana harita bilgisini de kullanabilmektedir. Metot içerisinde bir çarpışma engelleme prosedürü de tanımlıdır. Tanımlı dairesel bir lokal bölgede robotun belli bir güvenlik katsayısı oranında geçebileceği açıklıklar tespit edilerek bu açıklılar çeşitli kriterlere göre yeniden organize edilir. Bu kriterler örtüşen açıklık olması, duvar-tipi engelle sonlanan açıklık tespit edilmesi, öncelikli geçilecek açıklık tespiti gibi tanımlı özel durumlar için belirlenmiştir. Özel durumlar altında organize edilen optimal açıklıklar tekrar bir filtreden geçirilerek güvenli boyutlarda olmayanlar elenir. Kalan açıklıklar üzerindeki noktalara belli bir maliyet fonksiyonu tarafından skorlar tayin edilir. Bu skorlara göre belirlenen optimal nokta lokal hedef olarak belirlenir ve bu aşamadan sonra kontrol birimi robotu ilgili noktaya yönlendirmek için yönelim açısı ve tekerlek hızları arasında bir kontrol algoritması koşturur. Çarpışma engelleme prosedürü her işlem adımında tanımlanan kontrolleri yaparak lokal hedefe güvenli bir şekilde ilerleme işlevini yürütür. Anlık hedef ile robotun anlık konumu arasında kalan serbest çalışma bölgesi herhangi bir planlayıcı (A*, D*, Dijkstra v.b.) ile işlenmez. Lokal hedef anlık olarak değişebileceği ve çarpışma engelleme algoritması

sınırlandırılmış bir bölgede engel tarama işlemini gerçek zamanlı olarak gerçekleştirdiği için böyle bir planlamaya ihtiyaç duyulmamaktadır. Önerilen metot statik planlayıcıların değişen engel profili karşısında işlevsizliği, dinamik planlayıcıların ise kısmen kullanılan global planlayıcılardan kaynaklı işlem yükünü ve kararsızlık problemlerini gidermektedir. Robot, lokal hedefe yönelim hareketi yaparken doğal bir yörünge izlemeye zorlandığı için duraklama, kendi etrafında dönme gibi kinematik yapısına uygun olmayan yörünge hareketleri yapmamaktadır. Güvenlik amaçlı tanımlanmış kurtarma modlarında ise duraklama, geri manevra yapma, kendi ekseni etrafında dönme hareketlerini yapabilmektedir. Yerel minimum problemini çözmek için global harita bilgisini işleyebilen metot, harita bilgisi olmaması durumunda kısa ölçekli hafıza özelliği ile bu problemi pek çok durum için yaşamamaktadır. Harita bilgisinin işlenmesi ise opsiyonel olup haritanın maliyet matrisi elde edilerek lokal hedef tespiti işlemi sırasında maliyet fonksiyonuna bir parametre olarak eklenmektedir. Statik planlama yapılması gereken durumlarda ise ilk olarak Dijkstra metodu ile haritanın maliyet haritası hesaplanmaktadır. Dijkstra, literatürdeki benzer algoritmalara kıyasla gerçek zamanlı çalışmalara uygulanabilir işlem süreleri sunmaktadır. Bu yaklaşım ile bir başlangıç noktasından diğer tüm erişilebilir noktalara olan bir Öklid maliyeti hesaplanır ve bu bilgi statik ve dinamik yörünge planlama işlemlerinde planlama performansını arttırmak için kullanılabilir. Statik yörünge planlama aşamasında Gradient Descent algoritması, Dijkstra ile elde edilen maliyet haritası üzerinde koşturularak en kısa statik yol hesaplanmaktadır. Bu yaklaşım statik ve dinamik yöntemlerin hibrit bir formatı olarak değerlendirilmektedir. Dinamik yol planlama algoritması hareketli engeller ile ilgili temel bir fonksiyon da içermektedir. Ortamda önceden konumu bilinmeyen ve zamanla konumu değişen engellerin bulunması, hesaplanan lokal hedefe yönelim esnasında robotun engel tarafından sürüklenmesi gibi durumlara yol açabilir. Bu durum gözetilerek hareketli objelerin konumlarının kestirilmesine yönelik temel bir metot, planlama algoritmasına dâhil edilmiştir.

### 1.1.2. Konumlandırma problemleri

Otonom mobil robot platformları üzerinde hareket planlaması yapılırken gerçek zamanlı konum bilgisi kabul edilebilir bir doğrulukta ve gecikme bandında okunabilmelidir. Ayrıca mobil robotun çalışma ortamına uygun bir donanım

olmalıdır. Çalışma ortamının kapalı ya da açık alan olması belirleyici olmakla birlikte, iç mekânlarda bulunan duvar yapısı, metal yoğunluğu, aydınlatma koşulları, tavan şekli gibi fiziksel koşullar da konumlandırma sistemi geliştirilirken veya seçilirken belirleyici faktörlerdir. Dış ortamlar için yaygın olarak tercih edilen çözüm Global Positioning System (GPS) tabanlı sistemlerden oluşmaktadır. İç mekânlarda ise uydu erişimi kısıtlanacağı için alternatif çözümler üzerinde çalışmalar yapılmıştır. İç mekânlar için önerilen konumlandırma metotları ise yaygın olarak WI-FI, Bluetooth ve radyo sinyalleri ile haberleşme tabanlı uygulamalardır. Bu yöntemlerin belli üstünlükleri olmakla birlikte geçirgenliği düşük malzemelerin sıklığı, duvarlar, elektromanyetik gürültüler ve bağlantı kopmalarına sebep olabilecek donanımsal ya da yazılımsal faktörlerden etkilenme eğilimleri de söz konusudur. Geniş bant haberleşme tekniğiyle konumlama hassasiyeti ve geçirgenlik problemi iyileştirilse de kullanılan etiketlerin sayısı ve ortamın büyüklüğüne bağlı olan gecikmeler bu teknikte de yaşanabilmektedir. Endüstriyel ortamlar için geliştirilmiş lazer tabanlı mutlak konumlandırma cihazları da yüksek doğrulukta konum bilgisi sağlayabilmektedir. Ancak bu ürünler nispeten çok yüksek maliyetlere sahiptir. Pasif etiketlerinin iç mekânlara kurulumları da belli bir seviyede uzmanlık gerektirir. İki ve üç boyutlu lazer alan tarayıcı sensör (LIDAR) verilerinin işlenmesi esasına dayanan doğal navigasyon (natural navigation) yaklaşımları, diğer konumlandırma metotlarının aksine ortamda bulunan engellerin geometrisini kullanarak harici bir alıcı verici gerektirmeden konumlama yapabilmektedir. Bazı durumlarda önceden konumları bilinen ve geometrisi tanımlı engellerin gerçek zamanlı ölçümlerle eşleştirilmesi (landmark scan matching) yaklaşımları da uygulanmaktadır. Bu işlemin konumlama hassasiyeti ve ölçüm periyodu, kullanılan sensörün hassasiyeti ve çalışma frekansı ile doğrudan ilişkilidir. Kablosuz haberleşme içeren konumlandırma sistemlerinden çoğu durumda daha yüksek doğrulukta konumlandırma yapılabilmesine karşın dezavantajlı olduğu bazı durumlar vardır. Lazer ışınlarının ölçüm alınan yüzeyde kırılması ve doğrusallığını kaybetmesi durumlarında ölçüm hatası öngörülemeyecek düzeylere ulaşabilmektedir. Bu sebeple ortamın önceden hazır hale getirilmesi gerekebilir. Cam yüzeyler, şeffaf sıvı içeren objeler ve ışını kırabilecek malzemeler giderilerek önlemlerin alınması gerekebilir. Bunun yanı sıra birbirine paralel uzanan iki duvar arasında alınan ölçüm veya birden çok simetrik engel profilinin olduğu ortamlarda alınan ölçümler, tarama eşleştirme aşamasında

birden çok noktada hatalı konumlama sonucu üretebilmektedir. Bu tarz yaklaşımlarda birbirini tekrar eden özellikler (aynı geometrideki köşeler, çıkıntılar, silindirik gövdeler ve eş aralıklı sıralanmış özdeş nesneler) zorlayıcı durumlardır ve yüksek seviyelerde konumlama hatası ile sonuçlanmaktadır. Konumlandırma sistemlerine genel olarak destekleyici bir unsur olarak eklenen odometre hesabı tek başına konumlandırıcı olarak uygulandığında zamanla artan bir konumlama hatası ile sonuçlanmaktadır. Kusursuz düzlükte ve pürüzsüzlükte olmayan yüzeyler, sürtünmeye bağlı tekerleklerde oluşan tutunamama problemleri, patinaj ve enkoderin adım kaçırması gibi fiziksel ve donanımsal etkenler odometre hesabındaki bozucu etkenlerdir. Bu yaklaşım görece kısa mesafelerde konum ile ilgili yaklaşık bir bilgi verebilir ancak hesaplanamayan fiziksel kayıplardan kaynaklanan enkoder sayım hatasının, konum bilgisinde ne kadar bir kalıcı durum hatasına yol açtığı tam olarak öngörülemez. Odometre yaklaşımı, rasgele hata karakteristiğinde olan diğer metotlar ile entegre edilebilir. Bu durumda iki yaklaşımın da konumlama doğruluğunun iyileştirilebildiği literatürdeki uygulamalarda görülmüştür. Görüntü işleme tabanlı konumlandırma sistemleri de iç mekânlar için önerilen çözümlerden biridir. Tanımlı nesne (landmark) tespiti, şerit veya çizgi takibi, barkod okuma veya yükseğe konumlandırılmış bir kamera ile mobil robotun kuş bakışı takip edilmesi şeklinde değişik pek çok teknik üzerinde çalışılmıştır. Bu yöntemler ortam ışıklandırmasına ve doğrudan görüş alınabilmesine oldukça bağlıdır. Gürültüler ve yetersiz veya aşırı aydınlatmadan etkilenebilen bu yaklaşımlar daha çok kontrollü ortamlar için uygulanabilir durumdadır. Görüntü işleme tabanlı konumlandırıcıların ışık faktöründen en az etkilenmesi için uygulanabilecek yöntemlerden biri görünür ışık bölgesinin dışında çalışmaktır. Kızılötesi bantta yapılacak bir çalışma görünür aydınlatmadan bağımsız olarak karanlık ortamlarda dahi çalışabilmektedir. Aktif bir kızılötesi projeksiyon grubu ve pasif reflektörler sistematik bir şekilde iç mekanlara kurularak görüntü işleme tabanlı bir konumlandırma sistemi oluşturulabilir. Tez kapsamında da konumlandırma sistemi olarak kızıl ötesi aktif projeksiyon grubu ile aydınlatılan, daha önceden kodlanmış reflektör etiketlerini yansıma görüntüleri kullanılmıştır. Bu görüntüler işlenerek bir konum tespiti algoritması geliştirilmiştir. Kameraya kızılötesi (IR) lens entegre edilerek dar bir bant geçirgenliği sağlanmış ve diğer görüntü ve ışık kaynakları doğal olarak filtrelenmiştir. İkili (binary) kodlama ile kodlanan pasif etiketler tavan bölümüne kamera tarafından en az 1 etiket

görünecek bir şekilde yerleştirilmiştir. Konumlandırıcının hata karakteristiği rasgele hatalar olup zamana bağlı birikimli bir hata yığılması oluşmamaktadır. Bu sisteme enkoder tabanlı odometre bileşeni de entegre edilerek konumlandırma hatası düşürülmüş ve anlık görüş kayıpları sırasında konumlandırıcının odometre destekli veri akışına devam etmesi sağlanmıştır.

### 1.1.3. Haritalandırma problemleri

Mobil robot çalışmalarının önemli bir bileşeni olan harita çıkarma işlemini gerçekleştirmek için literatürde farklı perspektiflerde çalışmalar yapılmıştır. Bu işlemin doğruluğu genel yol planlama ve takip performansını doğrudan etkilemektedir. Haritada olası hatalı engel yerleşimi veya serbest uzaya dâhil edilen engellenmiş bölgeler statik yol planlama aşamasında öngörülemez problemlere yol açabilmektedir. Bu problem için güçlü bir teorik altyapıya sahip olan SLAM ön plana çıkmaktadır. Çalışılan harita üzerinde imge işleme veya lazer tabanlı engel algılama yöntemleri içeren SLAM, genel kabul gören iki temel haritalama yaklaşımı içermektedir. Gmapping (Grid Mapping) ve Cartographer yaklaşımları lazer tabanlı haritalama sistemlerinde oldukça yaygın kullanıma sahip yöntemlerdir. Gmapping metodu parçacık filtresi tabanlı olasılıksal bir yaklaşımdır. Cartographer yaklaşımında filtreleme yapılmaksızın açısal duruşun, okunan sensör deseni ve enkoder verileri kullanılarak optimize edilmesi temel alınmaktadır [17]. Olasılıksal yaklaşımlar çok sayıda parametre bulundurması sebebiyle ortam ve engel türüne bağlı olarak değişken performans göstermektedir. Doğrusal olmayan optimizasyon içeren yaklaşımlar için gerçek zamanlı süreç planlaması kritik bir konudur. Tüm SLAM türlerinde doğruluk seviyesi olarak kabul gören bu yaklaşım işlem yükü bakımından dezavantajlıdır. Lazer tabanlı haritalandırmanın dışında görsel geri besleme ile gerçekleştirilen RGB-D SLAM ve RTAB-Map benzeri metotlar da uygulanmaktadır. Bu yöntemlerde temel olarak Microsoft Kinect ve Intel Real Sense R200 gibi derinlik kameraları kullanılarak üç boyutlu imge üzerinde öznitelik eşleştirme veya kamera aracılığıyla alınan imgeler üzerinde RANSAC, Hough dönüşümü gibi metotlar kullanılarak doğru tespiti, köşe noktalarının tespiti gibi iki boyutlu imge işleme teknikleri uygulanmaktadır. Haritalama konusunda MATLAB ortamında sunulan “Mapping with known poses” metodunu temel alan bir eklenti geliştirilen benzetim ortamında sunulmuştur.

### 1.1.4. Benzetim ortamı ile ilgili problemler

Geliştirilen metotların gerçek dünya koşullarında uygulanmasından önce bir benzetim ortamında testlerinin yapılması, metotların ve fiziksel mobil robot platformunun iyileştirilmesi açısından etkin bir yoldur. Bu amaçla mobil robot araştırmacıları kendi çalışmalarına en uygun benzetim aracını kullanmak isterler. Literatürde bu amaca yönelik geliştirilen çok sayıda benzetim aracı vardır. MATLAB yazılımı sağladığı güçlü programlama altyapısı ve grafik arayüzü ile çeşitli geliştiriciler tarafından mobil robot benzetim aracı platformu olarak tercih edilmiştir [18-19]. Bu arayüzler benzetim ortamı için tatminkâr sonuçlar verse de kinematik kısıtlar gözetilmediği için fiziki uygulamalara tam bir uyumu söz konusu değildir. Kinematik ve dinamik model gerektiren ve hareket kontrolü için gerekli kontrol çıkışlarını sayısal değerler olarak çıktı veren bir diğer benzetim ortamı olan [20] gelişmiş görsel bir arayüze sahip olmadığı için tam bir benzetim ortamı değildir. Model tabanlı bir benzetim ortamı olan EasyLab [21], blok şemalar olarak mobil robot yol planlama ve diğer bileşenleri modellemektedir. Göreceli olarak kolay kullanım sağlayan bu yaklaşım kullanıcıyı bloklarla sınırlandırmaktadır. LinkBot ve MoBot robotları için özel olarak geliştirilen RoboSim yazılımı, sanal 3 boyutlu bir ortam sunmaktadır. Robota has olması bakımından kısıtlı olsa da 3 boyutlu arayüz kullanılması gerçeklik algısını yükseltmiştir [22].

Corke [23] tarafından geliştirilen MATLAB tabanlı "MATLAB Robotic Toolbox", kullanıcıya kolaylık sunan basit arayüzü ile öne çıkmaktadır. Başlangıç versiyonlarında mobil robotların görselleştirilmesi iki boyutlu bir arayüz üzerinde üçgen geometriler olarak verilmiş olup engeller ise basit dairesel nesneler olarak görselleştirilmiştir. Son versiyonunda bu arayüz üç boyutlu basit bir grafik arayüzü ile değiştirilmiştir. Gelişimi devam eden MATLAB Robotic Toolbox, çevresel faktörlerin modellenmesi konusunda temel düzeyde seçenekler sunabilmektedir. Benzetim ortamında tasarlanan bir senaryonun gerçek dünya koşullarında deneysel olarak gerçekleştirilmesine elverişli olan ve yaygın kullanım alanı bulan Gazebo [24], üç boyutlu bir benzetim ortamıdır. Gazebo hazır obje kütüphanesi ve mobil robot geometrileri ile esnek bir kullanıma sahiptir. Otonom mobil robot uygulamaları için C++ tabanlı özgün metotlar geliştirilmesi mümkün olan Gazebo ortamı Robot Operating System (ROS) işletim sistemi üzerine kurulabilmektedir. ROS ise Linux

tabanlı işletim sistemleri üzerine kurulabilmektedir. Ana işletim sistemi üzerine ROS işletim sistemi kurularak Gazebo kurulumu da ROS üzerine yapılmaktadır. Bu yazılımlar arasındaki iletişimi kurgulamak için diğer arayüzlere kıyasla önemli bir yazılımsal altyapı gerekmektedir. Windows tabanlı işletim sistemlerine tam uyumlu bir arayüz olmaması ve gerek kurulum, gerek uygulama aşamasında duyulan altyapı gereksiniminin yüksek olması, Gazebo'nun zorlayıcı yönlerindendir. Kullanılan sensörlerin ROS işletim sistemi ile fiziksel haberleşmesi için tanımlı sürücülerin Windows tabanlı işletim sistemleri için geliştirilen sürücüler kadar yaygın olmaması da donanımsal zorluklardan biridir. Tez kapsamında bu ortamların güçlü ve zayıf yönleri araştırılarak yeni bir benzetim aracı geliştirilmiştir. Üç boyutlu bir arayüz olması ve daha önceden tanımlanmış nesnelerin bulunduğu bir obje kütüphanesinin bulunması gerçeklik algısını yükseltmiştir. MATLAB tabanlı Virtual Reality (VR) platformu kullanılması da Windows işletim sistemi kullanıcıları için ROS platformuna göre daha hızlı bir adaptasyon süreci sağlayacaktır. Tez kapsamında geliştirilen mobil robot platformunun modeline uygun bir sanal mobil robot, varsayılan olarak benzetim arayüzü içerisinde tanımlıdır. Geliştirilen yol planlama ve kontrol algoritmaları da benzetim arayüzü içerisinde varsayılan metotlar olarak bulunmaktadır. Özgün metotlar geliştirmek için harici bir fonksiyon tanımlanmış ve bu fonksiyon altında kullanıcı tanımlı yol planlama, kontrol ve diğer navigasyon görevleri tanımlanabilmektedir. Kinematik olarak tanımlanabilen özgün mobil robotlar hem üç boyutlu görsel olarak hem de hareket karakteristiği olarak ayrıca tanımlanabilmektedir. Bu anlamda literatürdeki emsallerine kıyasla görsellik, işlevsellik, esneklik ve kullanım kolaylığı kriterleri tatmin edici düzeyde bir benzetim ortamı olması hedeflenmiştir.

## 1.2. Literatür Taraması

Yol planlama, konumlandırma ve mobil robot benzetim arayüzü yazılımları hakkında yapılan çalışmalar mobil robot çalışmalarının önemli bir bölümünü oluşturmaktadır. Bu alt bileşenler ile ilgili yapılan çok sayıda çalışmaya rastlamak mümkündür.

### 1.2.1. Statik yol planlama yöntemleri

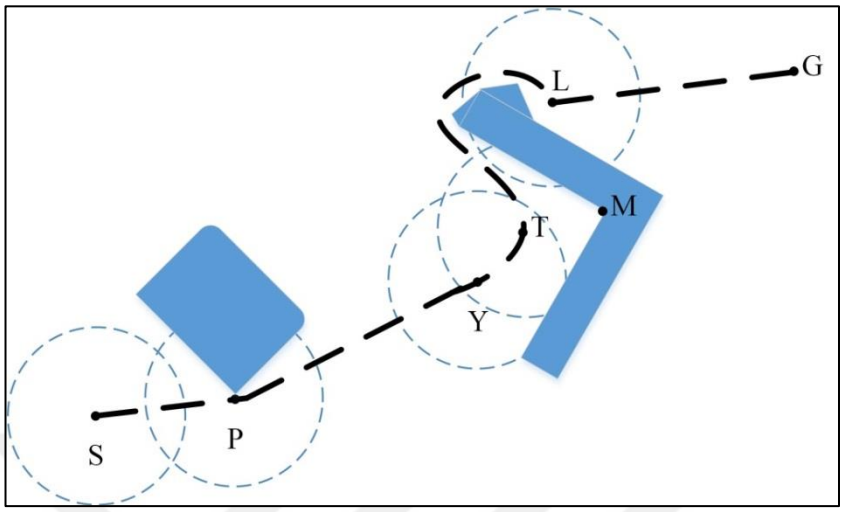
Statik ortamlar ve bilinen haritalar için uygulanabilecek metotlar, bilinmeyen haritalar ve dinamik ortamlar için yapılan çalışmalara göre sayıca daha fazladır ve

daha uzun bir süredir çalışılmaktadır. Bu tarz haritalar için daha çok yöntem geliştirilmiş olması, engellerin sabit olması ve konumlarının tam olarak önceden bilinmesi kısıtları bulunduğu için doğal bir sonuç olarak değerlendirilebilir. Bu konuda ilk olarak karşılaşılan ve kabul görmüş metotlar görünürlük grafikleri [25], hücre metodu [26], potansiyel alan metodu [27], hücre ayıklama (cell decomposition) [28], Dijkstra [29] ve Voronoi diyagramları [30] olarak verilebilir. Bu metotların yanı sıra sezgisel metotlar da statik haritalarda optimal yol planlama amacıyla uygulanabilmektedir. Genetik algoritmalar [31], karınca kolonisi ve benzetilmiş tavlama yöntemleri bu metotlara bir örnek olarak verilebilir. Görünürlük grafikleri ile elde edilen kenarların kromozom olarak kullanıldığı karma yöntemler de geliştirilmiştir [32]. Genetik algoritmanın seleksiyon prosedüründe yeni bir yaklaşım öneren bu metot, performans olarak geleneksel görünürlük uygulamasının performansını iyileştirmiştir. Benzer bir amaçla Cai ve Peng yol planlama için uygulanan genetik yaklaşımda kullanılan sabit uzunluklu ikili kodlama mekanizmalarını iyileştirmek için sabit ondalık şifreleme mekanizması geliştirmiştir [33]. Warren çalışmasında geleneksel potansiyel alan yaklaşımı kullanarak global yol planlama üzerinde çalışmıştır [34]. Zeng ve ekibi tarafından yapılan çalışmada Genetik algoritma ve A* algoritması karşılaştırmalı olarak statik yol planlama probleminin çözümünde kullanılmıştır [35]. Carlos ve ekibi yaptıkları çalışmada "exact distance transform" isimli özgün bir global yörünge planlayıcı geliştirmişlerdir. Bu yöntem ile görünürlük temelli özgün bir yol planlama metodu ortaya koymuşlardır [36]. Canny mobil robot yol planlayıcısı olarak Voronoi diyagramlarını kullanmıştır [37]. Asano ve ekibi görünürlük haritalarını kullanma fikrini mobil robot yol planlama problemine ilk uygulayan çalışmalardan birini yapmıştır [38]. Santos ve ekibi mobil robotlar için lokal yol planlama probleminde yapay sinir ağları yaklaşımını kullanmışlardır [39]. Saska ve ekibi yaptıkları çalışmada parçacık sürü optimizasyonu kullanarak mobil robot statik yörünge planlama problemine çözüm önermişlerdir [40]. Canny bir başka yol planlama yaklaşımı olan "silhouette" yaklaşımını ortaya koymuştur [41]. Elizondo-Leal ve Ramirez-Torres çalışmalarında satır bazlı özgün bir tarama yaparak global haritalar için farklı bir görünürlük kavramı tanımlamışlardır [42]. Harita üzerinde rasgele noktalar oluşturup, bu noktalar üzerinde tarama yöntemlerinden Dijkstra algoritmasının koşturulması temeline dayanan olasılıksal yol haritaları (PRM)

metodu statik planlayıcı olarak [43] uygulanan bir diğer yöntemdir. Olasılıksal diğer bir yol planlama metodu ise rapidly-exploring random tree (RRT) yöntemidir [44]. Statik ortamlarda haritanın bilinmesi durumunda geleneksel yaklaşımlar ve sezgisel metotlar bağımsız veya birlikte çözüm sunabilmektedirler. Engellerin hareketsiz olması ancak harita hakkında bilginin olmaması ve robotun yalnızca lokal sensör verileri ile optimal yol planlama yapmak zorunda olması problemi biraz daha zorlayıcı yapmaktadır. Bu tip haritalarla çalışırken yerel minimuma düşme, kilitlenme ve kararsızlığa düşme gibi durumlar, karşılaşılması muhtemel problemlerdir. Bu ve bunlara benzer durumlar için önerilen metotlar literatürde çalışılmıştır. Potansiyel alan metodunda yerel minimuma düşmeyi engellemek için asimtotik kararlı bir nokta-kütle sistemi öneren bir çalışma yapılmıştır [45]. Asimtotik kararlı olan sistemde Lyapunov fonksiyonu sadece global hedef ve engeller üzerinde bir potansiyel alan üretmektedir. Haritadaki tek yerel minimum noktası global hedef olarak belirlenerek yerel minimum problemine çözüm önerilmiştir. Potansiyel alan metoduna sanal bir su akış modeli entegre edilerek yerel minimum problemine bir çözüm önerisi yapılmıştır [46]. Potansiyel alan yönteminde yerel minimumlara karşı yapılan bir diğer çalışmada [47] lokal bölgede sanal yerel eğim oluşturarak robotun global hedefe yönelmesi amaçlanmıştır. Yerel minimum tespit edildiğinde global hedefin potansiyeli yükseltilerek yerel noktadan kaçmak hedeflenmiştir. Lokal bölgede çok kısıtlı bilgiler olduğu için global optimum çözüm bu yaklaşımda her zaman sağlanamamaktadır. Voronoi diyagramları ve potansiyel alan modelinin kombine edilmesi ile potansiyel alan haritasında alt lokal Voronoi hedefleri belirlenerek yerel minimumdan kaçınma çalışması da önerilmektedir [48]. Robotun hareketi yönünde 2 boyutlu sanal ve lokal bir pencere tanımlayarak bu bölgede planlanan yolu kesen en uzak noktanın hedef olarak belirlenmesi esasına dayalı bir metot önerilmiştir [49]. İşlem süresinin tatmin edici düzeylerde olduğu ortaya koyulan metot bilinmeyen ortamlarda statik engellerin olması varsayımı altında çalışan lokal bir yol planlama metodudur. Hücre tipinde tanımlanmış bilinmeyen haritalar için önerilen bir diğer metot, potansiyel alan ve yapay bağışıklık metotlarını birlikte kullanarak karmaşık haritalar için çarpışmasız bir statik yol planlamama çözümü önermiştir [50]. Bu yaklaşım ile genetik algoritma tabanlı bir diğer metot [51] aynı koşullarda karşılaştırılmış ve yol planlama performansının iyileştirildiği sonucuna varılmıştır. A* tabanlı lokal yol planlama metodu ve

görünürlük noktaları dinamik olarak değiştirilerek lokal statik yol planlama çözümleri de önerilmiştir [52-53]. Lokal planlayıcı olan Tangent-bug metodu bilinmeyen ortamlarda yol planlayıcı olarak ortaya koyulan bir diğer yöntemdir [54]. Bu yöntem bug algoritmalarının en gelişmiş formlarından birisi olarak kabul edilir. Alan tarayıcı sensör verileri kullanarak sensör görüş alanına giren engeller ve ana hedefin konumlarına göre hedefe yönelme ve engel çevresini dolaşma hareketleri tanımlanmıştır. Harita bilgisi olmaması ve sadece lokal sensör verilerinin kullanılması bu yöntemin temel varsayımları olarak bilinmektedir. Bu yöntem üzerinden yola çıkarak çeşitli lokal planlama yöntemleri geliştirilmiştir [55-56]. Tangent-bug yaklaşımı temel bir metot olması itibarı ile örnek bir senaryo üzerinde Şekil 1.11'de basit grafiklerle gösterilmiştir. Mavi renkli kapalı geometriler engelleri, siyah kesik çizgiler izlenen yolu, mavi kesik çemberler ise anlık sensör tarama limitlerini göstermektedir. Tangent-bug altında tanımlı temel görevler olan hedefe yönelme ve engel çevresini dolaşma hareketleri bu grafikte görülebilmektedir. Noktasal varsayılan hareketli obje S noktasından G noktasına ulaştırılmak istenmektedir. Engellerin konumu önceden bilinmediği için sensör görüş alanında kalan engellerin köşe noktaları baz alınarak işlem yapılır. Bu noktaların robota ve G noktasına olan Öklid uzaklığı toplamı her adımda hesaplanır. Bu toplam mesafeyi minimum yapan köşe anlık hedef olarak seçilir. Tüm aşamalar için seçilen köşenin G noktasına olan Öklid uzaklığı kaydedilir. Bu işlem hedefe yönelme hareketi olarak adlandırılmıştır. Bu aşamada, her adımda robot G noktasına daha da yakınlaşır. S ile P arası, P ile Y arası, Y ile T arası ve L ile G arasında robot hedefe yönelme prosedürünü işletir. T noktasına gelindiğinde ise robot o ana kadar G noktasına en yakın konumundadır ve bu noktadan sonra seçilen lokal hedef, robotu G noktasından uzaklaştırma eğilimindedir. T konumunda iken tespit edilen engel profilindeki, G noktasına en yakın nokta hafızada tutulur ve bu nokta M noktasıdır. |MG| uzunluğu kritik eşik olarak belirlenir ve bu eşiği düşürebilecek ilk engel noktası aranmaya başlanır. Bu durum tespit edildiği anda, algoritma engelin çevresini takip etme prosedürünü işletir. Anlık konum ile G hedefi arasındaki Öklid uzaklığı sürekli kaydedilerek |MG| uzunluğu ile karşılaştırılır. Engel çevresinde iken |MG| kritik eşiğinden daha düşük bir uzaklığı sağlayan ilk nokta L noktasıdır ve bu noktada engel duvarından ayrılarak engelin çevresini takip etme prosedürü sonlandırılır. L ile

G arasında hedefe yönelme işlemi yürütülerek hedefe ulaşılmış olur. Tüm bu aşamalarda sensör görüş alanı sembolik olarak gösterilmiştir.




Şekil 1.11. Tangent-bug lokal yol planlayıcısı

### 1.2.2. Dinamik yol planlama yöntemleri

Statik haritalar üzerinde çok sayıda çalışma yapılmıştır ve bu çalışmaların bütünü, probleme güçlü çözümler üretilmesini sağlamıştır. Haritanın tamamen önceden bilinmesi ve engellerin sabit olması varsayımları yol planlama problemini dinamik koşullara göre nispeten daha kolay bir şekilde çözmeye olanak tanımaktadır. Ancak uygulamada karşılaşılan problemlerden biri de hareket eden objelerin varlığıdır. Robotun çalışma bölgesinde insan veya diğer canlı-cansız hareketli nesnelerin şekil veya konumlarının sürekli değişmesi, yol planlama işlemini karmaşıklaştıran bir durumdur. Statik yol planlayıcılar kadar eski ve köklü bir literatür ortaya konulmamış olsa da yükselen bir trend olan dinamik yol planlama konusunda da tatmin edici çalışmalar yapılmıştır. Dinamik yol planlama probleminde belirsizlik ve karmaşıklık hareketli engel sayısı ile artmaktadır. Voronoi diyagramları, görünürlük grafikleri ve benzer metotların dinamik ortamlarda optimal yol planlama performansı statik haritalar için sunulan çözümler gibi efektif olmamaktadır. Bilinmeyen bir haritada çarpışma engelleme ve diğer güvenlik kriterleri daha büyük bir önem arz etmektedir.

Dinamik ortamlarda yol planlama işlemi için modifiye edilmiş bir A* algoritması önerilmiştir [57]. İnsansı robotlar için önerilen bu yöntem gerçek robotlardan daha

çok benzetim arayüzlerinde uygulanarak deneysel amaçlarla kullanılmıştır. Izgara tipi (grid cell) haritalar üzerinde geliştirilen bu yöntem renklendirilmiş bir gösterim sunmaktadır. Engeller ve serbest alanlar bitmap resmi olarak farklı renklerde işlenmiştir. Geliştirilen bu yöntem robotun anlık konumundan bir sonraki işlem zamanında atması gereken adımları bir konum dizisi şeklinde vermektedir. Çalışılan haritada sabit ve önceden konumları bilinen engellerin ve öngörülebilir hareketler yapan hareketli objelerin olduğu varsayımı ile bir dinamik yol planlama çözümü sunmaktadır. Çözüm adımları oluşturulurken üç parçalı bir maliyet fonksiyonu kullanılmıştır. Toplam maliyeti oluşturan alt fonksiyonlar verilen maddelerden oluşmaktadır:

- Konum maliyeti: Olası çözüm hücresinin statik engeller veya hareketli engellerin öngörülen konumları ile kesişme durumları bu fonksiyon altında maliyetlendirilir.

- Adım maliyeti: Çözüm hücrelerine mevcut robot konumundan hareket edilmesinin maliyeti bu alt fonksiyonda hesaplanır.

- Global hedef maliyeti: Global hedeften çözüm hücresine giden, statik engellerin konumu ve dinamik engellerin kestirilen konumları gözetilerek planlanmış yolun maliyeti bu fonksiyon altında hesaplanır. Bu fonksiyon yerel minimumlara düşmeyi engellemek için eklenmiştir.

Dinamik ortamlar için yol planlama problemine genetik algoritma tabanlı çözümler de önerilmiştir [58]. Önerilen bu metot optimal veya optimal altı yol planlama çözümlerini dinamik ortamlar için hızlı bir şekilde hesaplayabilmektedir. Yöntemin çözüm uzayı engellerin köşe noktaları olarak kabul edilmiştir. İlk aşamada robotun anlık doğrusal hızı gen havuzuna eklenmektedir. Optimizasyon kriteri işlem süresi veya kat edilen mesafe olarak belirlenebilen yöntemde, robot harekete başlamadan önce sensör görüş alanındaki hareketli engellerin konum ve hızları hakkında tüm bilgiye sahiptir. Algoritmanın varsayımları ise engellerin poligon şeklinde olması, hareketli engellerin hızlarının sabit olması ve boyutlarının ihmal edilerek noktasal robot kabul edilmesidir. Genler başlangıç, hedef ve varsa engellerin köşe noktalarından oluşmaktadır. Her bir gen yatay konum, düşey konum, anlık robot hızı ve erişilebilirlik bileşenlerinden oluşmaktadır. Erişilebilirlik terimi, bir konumdan başka bir konuma herhangi bir engel tarafından kesilmeyen en az bir segment

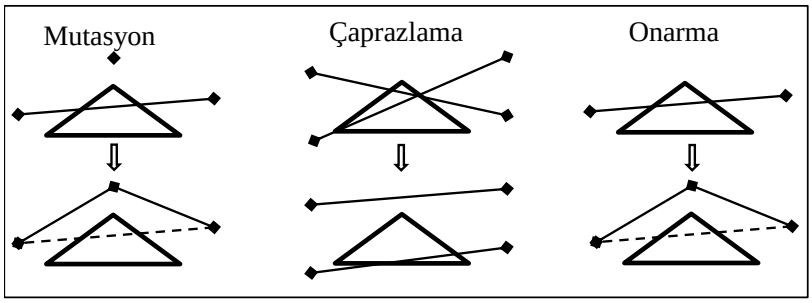
tanımlanabilir ise 0, tanımlanamaz ise 1 değerini alır. Hedef geni için bu parametre 0 değerini almak zorundadır. Toplam gen sayısı, planlanan yol üzerindeki köşe sayısına ek olarak başlangıç ve hedef noktalarıdır. Toplam n adet köşeden geçilecekse n+2 genden oluşan bir kromozom çözüm olarak ortaya çıkmaktadır. Şekil 1.12'de genetik algoritma tabanlı dinamik yol planlama yaklaşımında gen notasyonu ve bir kromozomun yapısı verilmiştir. Başlangıç noktası b, hedef noktası h, gen sayısı n, yatay konum x, düşey konum y, hız v, erişilebilirlik e kısalmaları ile verilmiştir.

| $gen_b$ | $gen_1$ | | | $gen_{n-2}$ | $gen_h$ |
|---|---|---|---|---|---|
| $x_b$ | $x_1$ | | | $x_{n-2}$ | $x_h$ |
| $y_b$ | $y_1$ | ... | ... | $y_{n-2}$ | $y_h$ |
| $v_b$ | $v_1$ | | | $v_{n-2}$ | $v_h$ |
| $e_b$ | $e_1$ | | | $e_{n-2}$ | $e_h$ |

Şekil 1.12. Genetik algoritma gen gösterimi

Şekil 1.12'de görüldüğü gibi çözüm bir dizi genden oluşan bir kromozom olarak tanımlanmıştır. Bir çözüm kromozomunda en az iki gen tanımlanabilmektedir. Başlangıç ve hedef arasında hiçbir engel olmaması durumunda çözüm bu iki gen arasındaki doğru segmenti olarak tanımlanır. Genetik algoritma tabanlı yol planlama yönteminde üç temel operasyon tanımlanmıştır: mutasyon, çaprazlama ve onarma.

Tanımlanan bu operasyonların görsel olarak gösterimi Şekil 1.13'te verilmiştir.




Şekil 1.13. Genetik algoritma operasyonları [58]

Önerilen bu yöntem rasgele seçim işlemi ile başlatılmış ve bir sıralama algoritması ile hedefe giden genler anlamlı bir sırada düzenlenmiştir. Oluşturulan belirli sayıdaki kromozomlar arasında çaprazlama yapılacak olanlar rulet tekniği ile belirlenmiştir. Oluşturulan yeni kromozomlar en kötü aktivasyon fonksiyonu çıktısını veren

kromozomlarla değiştirilmiştir. Seleksiyon olarak tanımlanan bu işlem her nesil için belli bir performans kriteri ya da nesil sayısı eşiği sağlanana kadar devam ettirilmiştir. En iyi kromozom ise çözüm olarak verilmiştir. Bu yöntemin uygulanmasında robot hızının yol planlama performansına oldukça fazla etki ettiği gözlemlenmiştir. Kritik bir hız seviyesinin üzerinde hareketli engellerin aşılmasının güçleştiği ortaya koyulmuştur.

Dinamik ortamlarda dinamik yol planlama çözümü için önerilen bir başka yöntem de global ve lokal planlayıcıların hibrit bir formda uygulanmasıdır. Global harita bilgisi ve lokal sensör ölçümlerinin eş zamanlı olarak işlenmesi temeline dayalı olan bu yöntem iki modülden oluşmaktadır: global planlama modülü ve lokal planlama modülü. Bu kapsamda yapılan [59] çalışmasında global planlama modülü global harita bilgisini kullanarak A* algoritması ile statik bir yol planlamış ve bu yol üzerinde kritik olarak nitelendirilmiş ara hedef noktalarını hesaplamıştır. Lokal planlama modülü ise lokal sensör verilerini işleyerek robotun navigasyonunu bulanık-sinirsel bir kontrol altyapısı ile yürüten ana fonksiyon olarak işletmiştir. İlk olarak global planlama modülü optimal global yörünge üzerinde konumlanmış ara geçiş noktalarını tespit etmiştir. Bu ara noktalar lokal planlayıcı tarafından çarpışma durumu kontrol edilerek önceden tanımlanmış bir potansiyel alan fonksiyonuna verilmiştir. Sonuç olarak hareketli herhangi bir engel tarafından engellenmemiş bir ara hedef noktası var ise doğal olarak o noktanın fonksiyona vereceği cevap optimal sonuç olacaktır. Hareketli engel tarafından kesilen ara nokta tespit edildiğinde potansiyel fonksiyonun lokal bölgede optimal çıktı verdiği nokta ara hedef olarak seçilmiştir. Lokal bölgede hiçbir açık bölge bulunamaması durumunda global modül tekrar çağırılarak navigasyon döngüsü tekrarlanmaktadır.

Dinamik ortamlarda global harita bilgisinin tam olarak ya da kısmen bilinmesi yerel minimumlardan sakınmak için önemli bir kaynaktır. Sadece anlık sensör verilerini işlemek yerel minimumlara takılmaya veya optimal altı çözümlerle karşılaşmaya neden olabilir. Harita oluşturma tabanlı çözümlere başvurmak ise yüksek hafıza gerektirebileceği gibi değişken engel profiline karşı yeterli olmayabilir. Olasılıksal metotlar ile hareketli engelin haritadan tekrar silinmesi veya hareketli engeli haritaya eklemek de işlem yüküne sahip yaklaşımlardır. Harita hakkında ön bilgi olmadan yapılan gerçek zamanlı dinamik yol planlama çalışmalarında yüzde yüz başarı elde

etmek pek çok harita için zorlayıcı bir görevdir. Bu problemin çözümü için literatürde yapılan çalışmalar mevcuttur.

Çoklu mobil robot uygulamaları için yapılan yapay potansiyel alan tabanlı bir çalışmada beş farklı mobil robot bir kavşak noktasından öncelik sıralarına göre çarpışmadan geçirilmiştir. Dinamik ve bilinmeyen ortamlar için geliştirilen çalışmada hareketli engellerin eşit boyutlarda ve hızlarda olduğu varsayımı yapılmıştır. Bu varsayımlar altında robotlar statik ve dinamik engeller arasından güvenli bir şekilde geçerek hedefe ulaşabilmiştir. Potansiyel alan metodu çoklu robot navigasyonu uygulamalarında kullanılması ile özgün bir çalışma olarak değerlendirilmektedir [60].

Dinamik ortamlar için geliştirilen bir diğer potansiyel alan tabanlı planlama yöntemine karınca kolonisi optimizasyonu entegre edilmiştir [61]. Hareketli engellerin hareketleri lokal haritadaki potansiyel alan değişimi ile ifade edilmiştir. İlk olarak lokal haritada düşük potansiyel alanlı bölgeler hesaplanmış, son aşamada ise karınca kolonisi metodu ile bu bölgelerden geçen optimal yol planlanmıştır.

Engellerin ve hedefin hareketli olduğu bir senaryo için geliştirilen yapay potansiyel alan metodu futbol oynayan turnuva robotları üzerinde uygulanmıştır [62]. Yeni bir kuvvet fonksiyonu ve çarpışma katsayısı tanımlanmıştır. Bu katsayı ve fonksiyonu temel alan özgün bir potansiyel alan yöntemi ile yol planlama operasyonu gerçekleştirilmiştir. Benzetim ortamında testleri yapılan metot, futbol turnuvasında yarışan robotlar üzerinde uygulanmıştır.

Hedef ve engellerin hareketli olması durumunda hız kontrolü ve dinamik yol planlama işlemlerinin birlikte yapılmasını amaçlayan potansiyel alan tabanlı başka bir yöntem de önerilmiştir [63]. Robotun bağıl hızı hedefin ve hareketli engellerin hız vektörlerine göre hesaplanmaktadır. Benzetim ortamında yapılan deneyler robotun hareketli engellerden sakınarak hareketli hedefi başarılı bir şekilde takip ettiğini göstermiştir.

Genetik algoritma ve lambda-interchange yerel arama metodu entegre edilerek geliştirilen bir başka dinamik yol planlama yöntemi ise A* yaklaşımının gelişmiş bir versiyonu olarak sunulmuştur [64]. Geliştirilen yöntem yol planlama performansı

olarak üzerine inşa edildiği standart yöntemlere göre optimal çözüme daha yakın bir sonuç vermiştir.

Engebeli ve pürüzlü yüzeyler için geliştirilen özel bir metot ise yol planlama problemini çevresel modelleme olarak ele almaktadır [65]. Bulanık mantık tabanlı bir yüzey modeli çıkaran metot iki alt fonksiyondan oluşmaktadır. Global fonksiyon haritanın engebe ve kıvrımlarını modellerken lokal fonksiyon daha önce hafızada olmayan bir engelle karşılaşılması durumunda global fonksiyona bu engeli işlemesi için bilgi servis etmektedir. Yeniden planlanan global yol takip edilerek robotun navigasyonunun sağlanması amaçlanmıştır.

Dinamik yol planlama problemi için önerilen güçlü bir metot olan D* gerçek zamanlı uygulamalarıyla ön plana çıkmaktadır. Dinamik A* olarak da bilinen bu yöntem, lokal bölgede karşılaşılan engel profilini robotun bilinen konum ve oryantasyon bilgisine dayanarak ilgili transformasyonlar ile global eksenlerde ifade eder. A* metodu ile planlanmış optimal yol dinamik koşullar altında bloke edilmiş ise, beklenenin aksine, toplam yol tekrar hesaplanmaz. Bunun yerine kesintiye uğrayan bölge ile bir önceki işlem zamanında hesaplanmış olan global yörünge arasında bir alt yörünge tanımlanır. Bu işlem yapılırken bloke edilen yol segmenti için onarım şeklinde bir lokal planlama yapılmaktadır. Standart A* ve türevi yaklaşımlara nazaran işlem süresi önemli ölçüde iyileştirilmiştir [66].

Çoklu robot sistemleri için özel bir böcek türünün avlanma yöntemlerinden esinlenerek Ladybug isimli dinamik yol planlama metodu geliştirilmiştir [67]. Eş zamanlı olarak sensör verileri işlenerek sadece lokal düzeyde bir planlama yapılan çalışma benzetim ortamında çalıştırılmıştır. Adaptif olan ve olmayan türleri ortaya koyulan metot yol planlama yönteminden daha çok hareket kontrol metodu olarak değerlendirilmektedir.

Bug algoritmaları üzerine inşa edilen bir diğer yöntem, yol planlamanın yanı sıra bir açıklık tespiti ve bir engelden sakınma algoritması ortaya koymuştur. Lazer sensör verileri lokal bir pencerede işlenerek geçilebilir aralıklar belirlenmiş, bu aralıklar üzerindeki her harita hücresi yarı sezgisel bir maliyet ile ifade edilmiştir. En düşük sezgisel maliyete sahip hücre lokal hedef olarak belirlenmiştir. Çarpışma tespiti ise kritik bir yakınlıkta bulunan engel tespit edilmesi temeline dayanmaktadır. Tespit

edilen kritik engel eğer robotun anlık hedefinde ise tanımlı bir eşik açı değeri kadar robot hedeften saptırılmaktadır. Güncellenen doğrultuda çarpışma kontrolü yapılır ve çarpışmasız bir yönelim bulunana kadar robota ekseni etrafında dönme hareketi yaptırılır. Bu sayede robotun yönelimi her işlem adımında engelden sakınacak biçimde değiştirilmektedir [68].

Hua-Qing ve ekibi dinamik ortamlar için çok amaçlı parçacık sürü optimizasyonu uygulayarak engelden sakınma yöntemi önermişlerdir. Bu yöntem ikinci dereceden bir hareket modeli ortaya koymuştur. Hareketli engellerin konum tahmini ve çarpışma engelleme modüllerini içermektedir. Güvenli navigasyon metodu olarak geliştirilen bu yöntem standart yapay potansiyel alan tabanlı dinamik yol planlama metotları ile benzetim ortamında karşılaştırılmış, performansının daha ileri seviyede olduğu ortaya koyulmuştur [69].

Lu ve Gong, dinamik ve bilinmeyen ortamlar için parçacık sürü optimizasyonu tabanlı bir yol planlama metodu sunmuşlardır. Yol planlama problemi, hedef ve engellerin konumu arasında bir minimum problemine dönüştürülmüştür. Engellere en uzak ve hedefe en yakın noktalar arasında optimal parçacık olarak hesaplanan konum anlık hedef olarak belirlenmiştir. Değişen ortam profiline göre algoritma parçacık optimizasyonunu tekrar yapmaktadır. Benzetim ortamında yapılan deneylerde başarım elde edildiği görülmüştür [70].

Hareketli engellerin konumlarının kestirilmesi temeline dayanarak yapılan bir diğer çalışmada ise kayan pencere metodu ile lokal bir bölgede hareketli objelerin yer değiştirme vektörleri hesaplanmıştır. Bu bilgi kullanılarak global yörünge kısmi olarak güncellenmiştir. Statik ve dinamik engellerin konumlarının bilinmediği varsayımı ile yapılan çalışma sonucunda optimal çarpışmasız bir yol planlama yapıldığı gözlemlenmiştir [71].

Koryakovskiy ve ekibi tarafından yapılan çalışma yeni bir genetik çaprazlama yöntemi önererek bunu dinamik yol planlama probleminin çözümünde uygulamışlardır. Dinamik engellerden sakınırken planlanan yolu kısaltan modifiye edilmiş bir genetik yol planlayıcı olarak özgündür [72].

Kesper ve ekibi tarafından yapılan çalışmada iki boyutlu LIDAR ile donatılmış yürüyen bir robotun navigasyonu ve kontrolü sağlanmıştır. Engel ve açıklık tespitinin yanı sıra çalışılan ortamın zeminindeki yükseltiler ve çöküntülerin tespiti için LIDAR verileri işlenmiştir. Engelden sakınma, basamaklardan çıkma, alçak zemine inme ve standart adım hareketlerinin karar aşamasında ortam sınıflandırması ve engel profilinin durumuna bağlı bir yöntem kullanılmıştır [73].

Mobil robot navigasyon ve kontrolü için sık kullanılan bir sensör olan LIDAR tasarımı üzerine yapılan bir çalışmada, RGB kamera ve servo motor mekanizması kullanılarak düşük maliyetli görüntü işleme tabanlı bir LIDAR tasarımı yapılmıştır [74]. Noktasal lazer ışık kaynağının temel görüntü işleme teknikleri ile kamera ekseninden kaçıklığı hesaplanmış, bu kaçıklık ile lazer noktasının kamera merkezine olan mesafesi arasında oluşturulan çok dereceli bir fonksiyon üretilmiştir. Frekansı düşük (1 Hz) uygulamalar için uygulanabilir sonuçlar elde edilmiştir.

Engel algılama ve güvenli navigasyon çalışmaları çoğunlukla çarpışma engelleme amacı ile yapılan çalışmaları içermesine karşın, robotların negatif engel olarak da bilinen düşük zeminlerden sakınması için yapılan çalışmalar da vardır. [75] çalışmasında üç boyutlu mesafe verileri işlenerek düzgün olmayan zeminlerde çalışan arama kurtarma robotları için negatif engellerden sakınma yöntemi geliştirilmiştir.

Negatif engeller ile ilgili yapılan bir diğer çalışmada, yüksek hızlarda hareket eden off-road tipi mobil robotların kontrolü üzerinde çalışılmıştır. Üç boyutlu LIDAR verileri işlenerek arazinin modeli çıkartılmış, destek vektör makineleri (SVM) tabanlı bir negatif engel dedektörü oluşturulmuştur [76]. Yapılan deneylerde negatif engellerden sakınma başarımı tatmin edici olarak değerlendirilmiştir.

Yol planlama algoritmalarında alt bir işlev olarak değerlendirilen açıklık tespiti üzerine yapılan bir çalışmada, LIDAR ile donatılmış mobil robotların sensör görüş alanındaki geçilebilir ve optimize edilmiş açıklıkların bulunması amaçlanmıştır [77]. Gerçek zamanlı uygulamalarda karşılaşılabilecek pek çok özel durum ele alınarak bu durumlara karşı alınan önlemler ve nihai durumda elde edilen organize edilmiş açıklık profilinin elde edilmesi üzerine yapılan bir çalışmadır.

Surmann ve ekibi tarafından yapılan çalışmada üç boyutlu LIDAR verileri işlenerek çok büyük iç mekân haritalarında SLAM uygulamaları yapılmıştır. Üç temel modülden oluşan metot tarama ve konumlandırma, optimal yol planlama ve motor kontrolü modüllerini eş zamanlı olarak koşturmaktadır. İç mekân haritası sayısal imge olarak görselleştirilmiştir [78].

Tzafestas ve ekibi tarafından yapılan çalışma holonomik olmayan robotlar için statik yörünge takibi amacıyla geliştirilen bir metot ortaya koymuştur. Mikroişlemci üzerinde gömülü olarak sayısal bir bulanık mantık kontrolörü tasarlanmıştır. Pioneer 3-DX8 robotu üzerinde ek bir FPGA hız kontrolörü ile desteklenen platform diferansiyel sürüşlü mobil robotlar için uyumludur [79].

Dinamik yol planlama yöntemi olarak tabu arama metodu da bir diğer çözüm olarak önerilmiştir [80]. Yerel minimumlara karşı güçlü önlemleri olan metot, harita üzerinde belirli bölgelerden tekrar geçme sayısı, engellenme durumu ve yolun genişliği gibi sözel ifadelerle tanımlanabilmektedir. Tanımlı robot hareketleri belirli konumlar için skorlanarak bir tabu tablosunda hafızaya kaydedilir ve global optimum çözüme ulaşmak için bu hafızadan faydalanılır [81].

Dinamik yol planlama konusunda artan bir uygulama eğilimi olan yöntemlerden birisi dinamik pencere yaklaşımı (dynamic window approach-DWA)'dır. DWA metodunda robotun merkezinde bulunduğu sanal bir pencere tanımlanmakta ve robot çalışma uzayında tanımlı bir maliyet fonksiyonuna bağlı olarak çarpışmasız hareket sağlanmaktadır. Kinematik modele uygun olarak belli bir zaman periyodunda robotun yapabileceği alt ve üst limitlerde açısal ve doğrusal hız bileşenlerinden, maliyet fonksiyonunu optimize eden çift kontrol girişi olarak belirlenir. Çeşitli varyasyonları olan yöntem üzerinde gelişmiş maliyet fonksiyonu tanımlamalarına yönelik çalışmalar yapılmıştır [82]. Yaygın kullanılan robot benzetim araçlarından Gazebo üzerinde lokal planlayıcı olarak kullanılan en yaygın metotlardan biri olmuştur ve endüstriyel ortamlarda gerçek zamanlı olarak kullanılan otonom hareket planlayıcılar arasında popüler bir konumdadır. Robot Operating System (ROS) tabanlı mobil robot yazılımı geliştirme ortamları olmak üzere yaygınlaşan kullanıma sahip bir diğer dinamik yol planlama yaklaşımı Time Elastic Band (TEB) yörünge planlayıcıdır. Robotun kine-dinamik limitleri dahilinde belirlenen bir optimizasyon

işlemi olarak ortaya çıkmıştır [83]. Orijinal versiyonu üzerinde geliştirmeler yapılarak çeşitli versiyonlar kazandırılan metot temelde planlanan bir statik yörünge üzerinde geçiş noktaları tanımlayarak bu noktalar arasında model tabanlı optimizasyon probleminin çözümünü hesaplamaktadır. En yakın engele uzaklık, statik yörüngeye uzaklık, yörünge uzunluğu ve model kısıtlarından oluşan karesel bir maliyet fonksiyonu tanımlıdır. Bir işlem adımında alternatif yörüngeler planlayan metot, özellikle hareketli engeller karşısında oldukça etkili sonuçlar verdiği için tercih edilmektedir. İşlem yükünün görece yüksek olması ve global hedef noktasında kalıcı durum hataları oluşabilmesi gibi eksi yönleri üzerinde literatürde halihazırda çalışmalar yapılmaya devam etmektedir [84].

### 1.2.3. İç mekân konumlandırma sistemleri

Açık alanlarda konumlama problemi çoğu uygulama için küresel uydu navigasyon sistemleri (GNSS) tabanlı sistemler kullanılarak çözülebilmektedir. Kapalı mekânlarda ise genel kabul gören benzer bir uygulama hâlihazırda yoktur. İç mekân konumlandırma sistemleri (IPS) üzerinde halen devam eden pek çok çalışma vardır ve birçoğu uygulamaya özel kısıtlar ile sınırlandırılmış çalışmalardır. İç mekân konumlandırma sistemlerinin temel amacı bina, fabrika ve benzeri bir kapalı ortamdaki insan veya başka bir hareketli objenin konum bilgisinin sağlanmasıdır. IPS ile ilgili literatür tarandığında [85], [86], [87], iç mekân konumlandırma sistemleri ataletsel, haberleşme tabanlı, görüntü işleme tabanlı ve benzeri yaklaşımlarla geliştirilebilmektedir. Ataletsel sistemler, konum bilgisi istenen objenin bir referans noktasına göre bağıl konumunun ivmeölçer ve gyroscope gibi ataletsel sensör verilerin işlenmesi yolu ile hesaplanması olarak tanımlanabilir. Haberleşme tabanlı sistemler kablosuz haberleşme protokolleri kullanılarak konum tespit edilmesini içermektedir. Bunlara ek olarak haberleşme ve atalet tabanlı sistemlerin birlikte kullanıldığı veya bunların dışında kalan imge işleme gibi çok farklı yaklaşımları içeren yöntem bulunmaktadır [88].

Mautz, çalışmasında iç mekân konumlandırma sistemleri hakkında genel bir karşılaştırma yapmış ve radyo frekansı tabanlı bir iç mekân konumlandırma sistemi önermiştir [89]. Youngsu ve ekibi yaptıkları çalışmada çoklu bir kablosuz sensör ağı üzerinden bir iç mekân konumlandırma sistemi geliştirmiştir [90]. Hazas ve Hooper

dar bant genişliğine sahip standart ultrasonik konumlandırıcıların aksine geniş bantlı ses işaretlerinin işlenmesi temeline dayanan hassas iç mekân konumlandırma sistemi üzerinde çalışmıştır. İç mekânlar için insan veya mobil robotların gerçek zamanlı konum takibi için hassas bir konumlandırma yöntemi ortaya koymuşlardır [91]. Harter ve ekibi radyo sinyalleri üzerinden haberleşen bir alıcı grubu ve verici ile iç mekân konumlandırma sistemi geliştirmişlerdir. Tavan bölümüne belli bir düzende yerleştirilen alıcılar ile konumlandırılacak nesne üzerine yerleştirilen verici arasındaki kablosuz haberleşmede, TDoA prensibi ile konum tespiti yapılmıştır. Active Bat ismi verilen bu çalışmada 9 cm doğruluk elde edilmiştir. Geniş alanlarda çok sayıda alıcı gerektirdiği için maliyet ve kurulum işçiliği yüksek bir yöntemdir [92]. Bir başka kablosuz haberleşme tabanlı konumlandırma sistemi olan Cricket sisteminde ise alıcı donanım konumlandırılacak olan objenin üzerine yerleştirilmiştir. Verici grubu ise iç mekânda bilinen konumlara yerleştirilerek senkronizasyonu sağlamıştır [93]. İç mekân konumlandırma sistemlerinde yükselen eğilimlerden biri de geniş bant teknolojisidir [94]. Geniş bant altyapısı zaman çözünürlüğünü yükselttiği için ToF kestirimleri daha doğru yapılabilmektedir. Bu da konumlama hassasiyetine olumlu etki etmektedir. Kablosuz alıcı tarafından alınan sinyalin şiddetini (RSS) referans alarak konumlandırma yapılan çalışmalar da mevcuttur [95]. Yang ve Chen yaptıkları çalışmada bir bina içerisindeki bilinen seçili konumlar için RSS değerlerinin korelasyonları üzerinde çalışarak binanın bir RSS modelini çıkarmışlardır. Bilinmeyen herhangi bir konum için bu model referans alınarak konum tespiti yapmaya çalışmışlardır [96]. Lin ve ekibi çalışmalarında Bluetooth Low Energy (BLE) etiketlerini kullanarak hastane ortamında hasta takibi yapmışlardır [97]. Munoz Diaz, insan takibi yapabilmek için mobil cihazlar üzerindeki gyroscope verilerini kullanmıştır [98]. Foxlin, ayaklara giyilen IMU sensörleri ile Kalman filtresi uygulaması yaparak insan hareketlerini iç mekânlarda takip etmeyi başarmıştır [99]. RFID haberleşmesinde RSS değerlerinin ölçümü ve ivmeölçer verisinin işlenmesi kombinasyonu ile yapılan çalışma ise kablosuz haberleşme ve ataletsel yaklaşımların hibrit bir uygulaması olarak değerlendirilebilir [100]. İç mekânlarda harita oluştururken eş zamanlı olarak konumlandırma yapmak amacıyla geliştirilen SLAM, araştırmacılar tarafından yaygın kullanılan bir konumlandırma yöntemidir [101]. İnsan takibine özel olarak geliştirilen IMU destekli footSLAM yöntemi, insanların belli bir konumdan bir diğerine gidiş-

dönüşlerinde hemen hemen aynı yolu takip etmesi gibi insana özgü varsayımlar üzerine kurgulanan olasılıksal bir konum belirleme yöntemidir [102-104].

Biswas ve Veloso, üç boyutlu derinlik imgelerini işleyerek bir mobil robot konumlandırma sistemi geliştirmişlerdir. Derinlik imgelerini Kinect sensörü kullanarak elde etmişlerdir. Çok boyutlu ölçüm verisinin işlenmesindeki hacmi azaltmaya ve gerçek zamanda istenilen gecikme düzeylerine erişmeye yönelik geliştirdikleri "fast sampling plane filtering" (FSPF) metodunu önermişlerdir [105]. Rongbing ve ekibi geliştirdikleri mobil robot iç mekân konumlandırma sisteminde LIDAR kullanmışlardır [13]. Ganganath ve Leung Kinect sensör ve odometre kullanarak iç mekân konumlandırma sistemi geliştirmişlerdir. Çalışmada konum etiketleri ve enkoder verileri kullanarak düşük maliyetli bir konumlandırıcı geliştirmişlerdir [106]. Choi ve Lee mobil robotlar için hareketli engellerin bulunduğu ortamlarda konum kestirimi yapmak için ultrasonik sensörler ve Kalman filtresi kombinasyonu kullanmıştır [107]. Parçacık filtresi, iç mekân haritası, odometre bilgisi ve iki boyutlu LIDAR verileri kullanılarak gerçekleştirilen Monte Carlo Localization (MCL) yöntemi, sensör modeli ve robot kinematiğini temel alan olasılıksal bir konumlandırma metodudur [108].

Tablo 1.1. İç mekân konumlandırma sistemlerinin karşılaştırılması [109]

| | | | | Parametreler | | |
|---|---|---|---|---|---|---|
| Teknik | Global konumlama | Doğruluk [m] | Kapsama [m] | Maliyet | Karmaşıklık | Kaçırılma problemine çözüm |
| İmge işleme | +/- | $10^{-3}$-$10^{-1}$ | 1-10 | Yüksek | Yüksek | + |
| IR | + | $10^{-2}$-1 | 1-5 | Orta/Yüksek | Düşük | + |
| MCL | + | $10^{-2}$-$10^{-1}$ | - | Orta | Yüksek | Kısmen |
| Wi-Fi | + | 1-10 | 20-50 | Orta/Düşük | Düşük | + |
| RFID | + | $10^{-1}$-1 | 1-10 | Düşük | Düşük | + |
| Bluetooth | + | 1-10 | 1-30 | Düşük | Düşük | + |

Literatürde birçok çalışmada konumlandırma sistemleri çeşitli parametreler bazında karşılaştırılmıştır. Tam bir değerlendirme yapmak mümkün olmasa da bunlardan doğruluk, kapsama alanı, maliyet, karmaşıklık ve kullanım yeri parametrelerine göre yapılan karşılaştırmalar ortak bir zeminde kesişmektedir [109]. Kamera, kızılötesi, ultrasonik, Wi-Fi, RFID ve bluetooth tabanlı konumlandırma sistemlerinin uygulamaya özel kısıtlar da göz önünde bulundurularak yapılan karşılaştırma sonuçları Tablo 1.1'de verilmiştir.

Konumlandırma sisteminin doğruluğu etkileyen çok sayıda faktör olması ve aynı sistemin bu etkiler altında farklı seviyelerde performans göstermesinden dolayı doğruluk parametresi bir aralık olarak verilmiştir.

### 1.2.4. Mobil robot benzetim arayüzleri

Mobil robot yazılımlarının fiziksel uygulamasına geçmeden önce çeşitli benzetim arayüzlerinde testler yapılması kendi içinde bir çalışma alanı olarak değerlendirilebilir. Bu alanda yapılan çalışmalar ve uygulamada kullanılan çeşitli kapsamdaki yazılımlar literatürde yer almaktadır. Marilou isimli yazılım 2010 yılında ticari amaçla sunulmuştur ve fiziksel ortam geliştirme, yerçekimi, moment gibi faktörleri değerlendirebilme gibi özellikler taşımaktadır. Omni-drive tipi robot analizleri, dağıtık fonksiyon analizleri, çoklu sistemlerin tasarımı ve haritalama gibi gelişmiş özellikleri barındıran bir yazılım olarak ortaya çıkmıştır [110]. 2005 yılından bu güne faaliyet gösteren LEGO firmasını geliştirdiği roboSim yazılımı LabView kütüphanelerine verdiği destek ile yaygın bir kullanım bulmuştur. Çok yüksek mekanik bilgiye ihtiyaç duymaksızın projeler geliştirilmesine imkân veren bir arayüzü bulunmaktadır. Üç boyutlu bir ortam geliştirme imkânı da bulunan yazılım fizik kurallarına uygun ortamlar geliştirilmesine izin vermektedir. Microsoft tarafından 2006 yılında geliştirilmeye başlanan Windows tabanlı bir mobil robot benzetim programı olan Robotics Developer Studio 4, önceden tanımlanmış blokların bir araya getirilmesi prensibi ile çalışan bir arayüzdür. Sensör kütüphanesinde mesafe sensörü ve kamera gibi temel elemanlar mevcuttur. Önceden tanımlanmış nesneler kullanılarak üç boyutlu çevre modellemesi yapılabilmektedir. Pons ve ekibi çalışmalarında Robotics Developer Studio 4 benzetim ortamını kullanarak model tabanlı mobil robot sistem tasarımı üzerine çalışma yapmıştır [22]. 2010 yılında ilk versiyonu geliştirilen çıkan V-REP, basit üç boyutlu tasarım yapma imkânı sağlayan geniş yelpazeli bir çevresel araç kütüphanesi barındıran bir arayüzdür. Üç boyutlu ortam geliştirme seçeneği da sunan V-REP içerisinde fiziksel robot modelleri de vardır. Hexapod, Nao ve E-Puck tipi robotlar kullanıcıya hazır olarak sunulmaktadır. Mobil robot benzetiminin yanı sıra endüstriyel seri robot kolu tasarımı da yapılabilmektedir. Kuka ve ABB robot kinematik çözümleri hazır olarak yazılım içinde sunulmuştur. V-REP ortamını kullanan Anh ve ekibi çalışmalarında sinirsel PID kontrol tabanlı özgün bir robot yürüme kontrolü gerçekleştirmişlerdir

[111]. Cyberbotics firması tarafından geliştirilen Webots üç boyutlu tasarım arayüzü ve zengin fiziksel araç kütüphanesi sunmaktadır. Nao ve Khepera gibi robotları kullanıcıya hazır olarak sunan yazılım, C/C++, Java, Phyton ve MATLAB gibi yaygın dilleri destekleyen bir kod editörü sağlamaktadır. Araştırmacılar bu dillerden biri ile kendi programlarını geliştirebilmektedirler. Grafik dosyası olarak çıktı vermesi, çeşitli fiziksel çevre elemanlarının hazır sunulması, dâhili olarak tanımlı gerçek robotlara program yükleyebilme arabirimi gibi özelliklerinin bulunması sayesinde yaygın kullanım bulmuştur. Sick LIDAR sensör modelini de içeren kütüphanesi otonom mobil robot benzetimlerinde kullanım kolaylığı sağlamaktadır. MATLAB arayüzü ile desteklenen program geliştirilen algoritmaların bu ortamda çalışmasını sağlamaktadır. Mohammad ve ekibi Webots arayüzü kullanarak mobil robotlar için bulanık mantık tabanlı engelden sakınma konulu bir çalışma yapmışlardır [112]. 2018 yılında versiyon 2.0 olarak güncellenen MATLAB Robotics Toolbox, Peter Corke tarafından geliştirilmiş ve bug, D* gibi algoritmalar içerisinde hazır olarak sunulmuştur. Gelişmiş bir üç boyutlu tasarım seçeneği sunmayan yazılım üzerinde seri robot çalışmaları da yapılabilmektedir. Farklı kontrolör ve kinematik formların eklenebilmesiyle, robot yazılımlarının test edilmesi için rahat bir uygulama alanı olarak değerlendirilebilir. Bu ortamın kullanılması için MATLAB programının yüklü olması gerekir. Xu ve ekibi MATLAB Robotics Toolbox kullanarak mobil robotlar için bir fizibilite çalışması gerçekleştirmişlerdir [113]. Linux tabanlı iki boyutlu bir benzetim ortamı olan STDR, çoklu robot sistemlerine uyumlu bir araçtır. Fiziksel çevre tasarımı seçeneği olamayan yazılımda duvar takibi ve labirent problemlerinin çözümlerine dönük uygulamalar yapılabilmektedir. ROS ortamında tanımlı bir paketi bulunan yazılım, kullanıcılar için başlangıç seviyesinin üzerinde bir programlama bilgisi gerektirmektedir. Çoklu robot haberleşmesi ve senkron çalışma gibi işlemlere dönük bir yazılımdır [114]. Java tabanlı bir mobil robot benzetimi olan Simbad, üç boyutlu bir arayüze sahip olmasına karşın gerçek objeler sağlamamaktadır. Otonom mobil robot programlama amacı ile geliştirilen Simbad, kütüphanesinde örnek kodlar ve sensör modelleri bulundurarak kullanım kolaylığı sağlamaktadır, buna karşın özgün bir yazılım geliştirmek için ileri seviyede bir programlama bilgisi gerektirmektedir [115]. Arduino platformunun grafik arayüzü olarak geliştirilen miniBloq tam anlamıyla bir benzetim ortamı olamamakla birlikte Pi-bot, Red-bot gibi mobil robotların kontrol

uygulamalarının geliştirme ortamı olarak değerlendirilebilir. Yeni başlayanlar için eğitici bir ortam sunmaktadır [116]. Gazebo 2002 yılında geliştirilmeye başlanmıştır. Yalnızca Linux ortamında çalışan yazılım C++ dili üzerine inşa edilmiştir. Linux tabanlıdır ve ROS işletim sistemi üzerine kurulur. Düşük seviyeli bir tasarım dili vardır. XML formatında tanımlanan robot kinematiği ve çeşitli katmanlardan oluşan düğüm ve haberleşme yapısı ile belli bir seviyede yazılım bilgisi gerektirmektedir. Üç boyutlu gerçekçi ortamlar oluşturulabilen bir ortam olan Gazebo, içerisinde bulunan iRobot, Kinect, Pioneer gibi hazır donanımlar ve zengin fiziksel araç kütüphanesi sayesinde karmaşık mobil robot benzetimlerinin geliştirilmesine olanak tanımaktadır. Bu yazılımların dışında Yoshida, yaptığı çalışmalarda MATLAB yazılımını kullanarak mobil robotlar için benzetim arayüzleri geliştirmiştir [18]. Bu yazılımlar görsel öğeler ile destekli bir arayüze sahip olmalarına karşın kullanıcı tanımlı kontroller ekleme noktalarında eksiklikler taşımaktadır [19]. Dean ve ekibi bir kod üretici arayüz geliştirerek tasarlanan robot konfigürasyonunu kodlara dönüştürmeyi hedeflemiştir [20].

### 1.3. Tezin Literatüre Katkısı

Tez çalışması kapsamında geliştirilen yol planlama metodu altında özgün açıklık tespiti, çarpışma engelleme, engel algılama ve optimal lokal hedef tespiti algoritmaları geliştirilmiştir. Konumlandırma sistemi ile yol planlama şeması entegre biçimde çalıştırılarak mobil robot platformuna otonom hareket kabiliyeti kazandırılmıştır. Önerilen yöntem üzerinde yapılan deneyler sonucunda ortalama 10 ms işlem süresi ile çalışıldığı gözlemlenmiştir. MATLAB 2017 altyapısında geliştirilen ve çok sayıda görsel arayüz ile desteklenen şema, her işlem adımında yeniden global yol planlama yaklaşımlarını koşturmak yerine planlanan hedefe robotun kontrollü bir biçimde yönlendirilmesi prensibine dayanmaktadır. Kontrollü yönlendirme hareketi, robotun lokal çevresinde güvenlik limitlerinin aşılmasının kontrol edilmesi olarak ifade edilmektedir. Gerçek zamanda her işlem adımında lokal dinamik bir yol planlama gerçekleştiren Dynamic Window Approach (Dinamik pencere yaklaşımı-DWA) ve Time Elastic Band (Zaman elastik bant-TEB) gibi yöntemlerde lokal pencere boyutu ve statik planlayıcının karmaşıklığına bağlı olarak işlem süreleri nispeten yüksektir. Model tabanlı kontrol kriterleri taşıyan yol planlayıcılar ise (lattice planner, Model Predictive Control-MPC) kalıcı durum

hataları ve yüksek işlem hacmi gibi dezavantajlara sahiptir. Yalnızca lokal düzeyde işlem yapan ve ilkel güvenlik kriterleri ortaya koyan metotlar (bug algoritmaları) ise yerel minimum problemleri ve metoda özgü çözülemez haritaların varlığı nedeniyle kısıtlı kullanım alanına sahip yöntemlerdir. Tez kapsamında geliştirilen yöntem planlama aşamasının ardından yerel minimum tespit edilene kadar tekrar yol planlama yapmamaktadır. Model tabanlı yaklaşımlarda karşılaşılan kalıcı durum hatalarına çözüm olarak kinematik modele özgü tanımlanan hareketler ile (sıfır-yarıçap dönüşü, geri manevra tanımları) sunulmuştur. Modele uygun sürüşü sağlamak için ise optimal lokal hedef belirleme algoritması altında tanımlanan maliyet fonksiyonunun bileşenlerinden açısal değişim parametresi ile çözüm uzayı robotun doğal navigasyon yapabileceği sınırlarla kısıtlanmaktadır. Çarpışma engelleme yöntemi olarak geliştirilen algoritma kullanılarak optimal hedef, lokal gerekli durumlarda güvenli olan eşdeğeri ile güncellenmektedir. Anlık robot konumu ile güvenli hedef arasında yol planlama yapmak yerine oransal kontrol tabanlı bir yaklaşım ile robotun tekerlek hızları kontrol edilmektedir. Konumlandırma sisteminden alınan geri besleme ile navigasyon işleminin doğruluğu kapalı çevrim olarak kontrol edilmektedir. Ayrıca DWA metodu için, hedef konumunda açısal kontrol yapılmasına olanak tanıyan bir yaklaşım önerilmiştir. Hedef konumunda istenen açıya ve robot parametrelerine bağlı olarak sanal bir garaj tanımlanarak robotun düşük işlem maliyeti ile hedef konumuna istenilen açısal pozisyonda konumlanması sağlanmıştır.

Tez çalışması kapsamında düşük maliyetli, görüntü işleme tabanlı bir iç mekân konumlandırma sistemi de tasarlanmış, mobil robot platformu üzerinde uygulanmıştır. Literatürde yaygın kullanıma sahip olan konumlandırma yaklaşımlarından AMCL yöntemine göre harita gereksinimi olmaması ve birikimli hatalara yol açmaması en belirgin artılarıdır. Her ölçüm bağımsız bir konum bilgisi sağlamakta ve önceki ölçüm verilerinden olasılıksal veya kestirimsel bir katkı almamaktadır. Ön bilgi olarak global harita gerektiren yöntemlerin aksine başlangıç konumu gerekmemesi de önemli bir artıdır. Alınan ilk ölçüm başlangıç pozisyonu olarak kabul edilmektedir. Konum bilgisinin yanında açısal pozisyon da elde edilebilmektedir. Konumlandırma sisteminin ölçüm hatası en çok 7 cm düzeyinde, açısal hata ise $1^{o}$'nin altında ölçülmüştür. EKF tabanlı veya MCL türevi

konumlandırma sistemlerinde güncel konum tespiti işleminde geçmiş konum verilerinden yararlanılmaktadır. Bu durum "kaçırılmış robot problemi" gibi zorlayıcı problemler karşısında ciddi dezavantaja neden olmaktadır. Otonom mod dışında operatör tarafından farklı bir konuma taşınan robotun tekrar mutlak konumunu bulması belli bir süre almaktadır. Önerilen konumlandırma sistemi ise mutlak konumlandırıcılara bir örnek teşkil etmektedir. Her işlem adımında bağımsız konum ölçümü yapıldığı için robotun hangi koşullarda nereye götürüldüğü önemli değildir. Tez çalışması kapsamında MATLAB tabanlı 3 boyutlu bir mobil robot benzetim aracı de geliştirilmiştir. Benzetim arayüzü olarak kullanılabilen bu yazılım kullanıcı tanımlı mobil robot modellerine ve navigasyon algoritmalarına uyumlu olacak bir yapıda tasarlanmıştır. Fiziksel mobil robot platformu üzerinde uygulanan tüm senaryolar öncelikle benzetim arayüzü üzerinde test edilmiş, gerçek dünya deneyleri ile karşılaştırıldığında oldukça tatmin edici sonuçlar elde edilmiştir. Gerçek dünya koşulları ile ortak haritada gerçekleştirilen benzetim ortamı deneylerinde yatay ve düşey eksenlerde oluşan konum hataları sonuç bölümünde verilmiştir. Benzetim arayüzü üzerinde bir harita çıkarma (mapping) eklentisi de bulunmaktadır. LIDAR verilerini temel alan mapping eklentisi, bilinen engel konumları ve konumlama hataları altında sensör görüş alanına dâhil edilen ortamın statik haritasını oluşturmaktadır. Bir diğer eklenti ise bilinen harita koşullarında konumlandırma sistemi olarak kullanılan AMCL metodudur. Harita çıkarma fonksiyonu ile çıkartılan harita kullanılarak AMCL eklentisi benzetim arayüzünde koşturulabilmektedir. Başlangıç konumu verilen robot benzetim arayüzünde AMCL ile konumlandırılabilmektedir. Windows işletim sistemi üzerinde ve MATLAB ortamında desteklenen benzetim arayüzü, eşdeğeri olan Linux tabanlı gelişmiş arayüzlere (Gazebo, Rviz) göre kurulum gerektirmemesi ve yaygın kullanımı olan Windows ortamında çalışması bakımından avantajlıdır. Benzetim arayüzü üzerinde iki boyutlu LIDAR arayüzü, iki ve üç boyutlu mobil robot modeli ve iç mekân haritası tanımlanmıştır. Robot platformunun uzak bir bilgisayardan kumanda edilebilmesine imkân tanıyan bir arayüz de geliştirilmiştir. Otonom modda çalışan robot bu arayüz üzerinden öncelikli olarak durdurulabilmekte, yeni görev tanımlanabilmekte veya manuel olarak kontrol edilebilmektedir. Bunlara ek olarak robot RC uzaktan kumanda üzerinden manuel olarak kontrol edilebilmektedir. Bu

özellikler bakımından geliştirilen mobil robot platformu ve navigasyon şeması otonom bir sistem olarak değerlendirilmektedir.

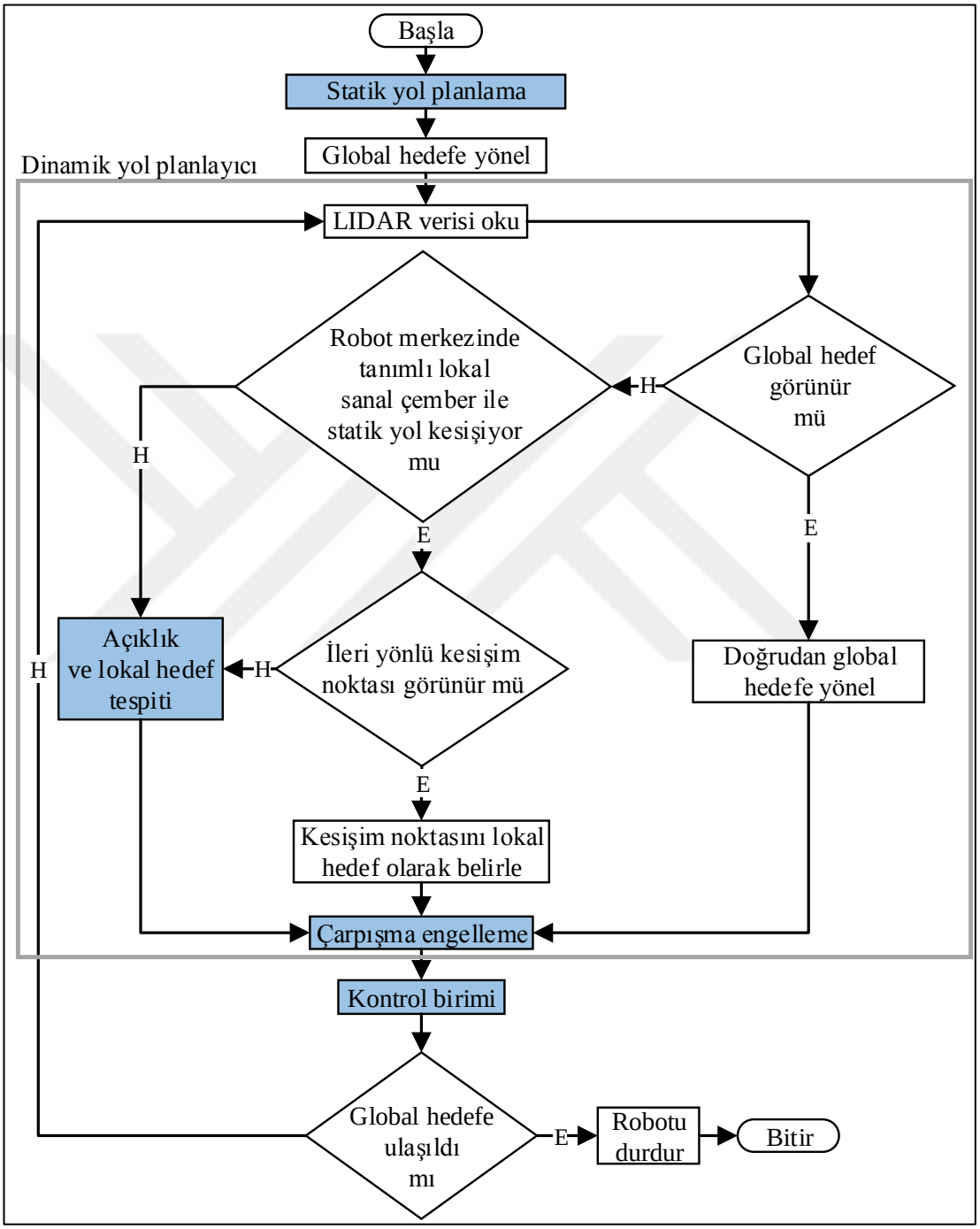
# 2. YOL PLANLAMA

Geliştirilen metot, alan tarama tabanlı hibrit bir yol planlama yöntemidir. Bu metodun uygulanmasında alan tarama işlemi için gerekli donanım(lar)ın, uygulama yapılacak mobil robot platformuna, zemine paralel ve ileri yönlü olarak monte edilmesi gerekmektedir. Geliştirilen yol planlama metodu, iç mekân harita bilgisini ve anlık LIDAR ölçümlerini giriş parametresi olarak kabul etmektedir. Harita bilgisinin önceden bilinmesi zorunlu değildir ancak, çarpışma engelleme ve dinamik yol planlama gibi gerçek zamanlı olarak takip edilmesi gereken işlemler için lokal sensör bilgisi zorunlu bir parametredir. Statik haritanın bilinmesi, başlangıçta oluşturulan statik yörüngenin güvenilirliğini etkilemekle birlikte yerel minimum oluşturabilecek engel profillerine karşı algoritmanın genel performansını iyileştirmektedir. Harita bilgisinin olmaması durumunda kurtarma modları ile çözülebilecek bu problemlerle karşılaşma sıklığı, statik haritanın bilinmesi durumunda karşılaşılma sıklığından daha düşük olmaktadır. Metodun genel işleyişi, engellerin hareketli, sabit, öncede konumu bilinen ve bilinmeyen olması durumları altında standart bir prosedür ortaya koymaktadır. Hareketli engellerin sayısı, hızları ve hareket karmaşıklığı zorlayıcı etkiler olarak kabul edilmektedir. Geliştirilen yol planlama metodu, işlenebilen haritaların türü ve yol planlanacak ortamın engel özellikleri göz önünde tutulduğunda dinamik ve lokal bir planlama yaklaşımı olarak değerlendirilmektedir.

Önerilen yol planlama metodu statik yol planlama, dinamik yol planlama ve kontrol fazlarından oluşmaktadır. Her faz içinde alt modüller tanımlanmıştır. Metodun genel akış şeması Şekil 2.1'de verilmiştir.

Genel akış şemasında koyu tonla gösterilen bloklar kendi içerisinde ayrıca detaylandırılacaktır. Statik planlama fazında iç mekân haritası önceden biliniyorsa ilgili statik planlama algoritması koşturularak maliyet haritası ve statik yol koordinatları algoritma hafızasına kaydedilir. Önceden bilinen bir harita yoksa hedef ile başlangıç konumu arasında düz bir hat, statik yolun başlangıç formu olarak

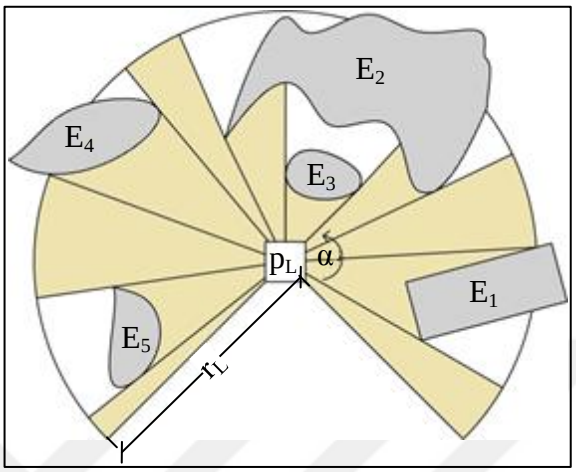
kabul edilir. Bu durumda yöntem açıklık tespiti temelli bir navigasyon izleyecektir. Robot düşey eksende kendi etrafında sıfır yarıçap ile dönme hareketi yaparak verilen hedefe yönelir ve ilk LIDAR ölçümünü alır.




Şekil 2.1. Hibrit yol planlama metodunun genel akış şeması

LIDAR ölçümleri bir mesafe ölçüm vektörüdür. Vektörün elemanları sensör odak merkezi ile lazer kaynağının çarptığı yüzey arasındaki milimetre biriminde

mesafelerdir. Mesafe ölçümlerinin sayısı, sensörün açısal ölçme çözünürlüğü ile ilgilidir. Örnek olarak $180^o$ görüş açısına sahip bir sensörün $1^o$ çözünürlükle ölçüm alması durumunda $0^o$ ile $180^o$ açısal konumda bulunan her engel için sıralı olarak 181 elemanlı bir vektör elde edilecektir.




Şekil 2.2. LIDAR ölçüm gösterimi

Şekil 2.2'de iki boyutlu uzayda, LIDAR tarama menzili içerisinde rasgele konumlara konumlanan engeller ve LIDAR tarama kesiti sembolik olarak verilmiştir. Koyu tonla vurgulanan alanlar LIDAR ışınlarının ölçüm alabileceği alanları göstermektedir. E sembolü ile etiketlenen kapalı geometriler engelleri temsil etmektedir. Sensörün iki boyutlu uzaydaki konumu $p_L$, ölçümlerin açı bilgisi $\alpha$ ve sensörün maksimum ölçüm limiti $r_L$ ile gösterilmiştir. Bu değer marka ve modele bağlı olarak değişmekle birlikte 20 metre menzile sahip iç mekân uyumlu LIDAR uygulamada kullanılmıştır. LIDAR sensörlerinin maksimum ölçüm limitine ek olarak minimum ölçüm limiti de tanımlıdır. Firmaların ürün özelliklerine göre değişmekle birlikte, minimum ölçüm uzaklığı 10 cm değerine sahip bir LIDAR uygulamalarda kullanılmıştır. LIDAR ölçüm modeli Denklem (2.1)'de verilmiştir. LIDAR modelinde maksimum ölçüm limitinde bir engel bulunmayan açısal ölçüm noktaları için, ilgili noktaya $r_L$ parametresinin değeri atanır. Bu saturasyon işlemi Denklem (2.2)'de verilmiştir.

$$m(p_L, \alpha)= \min_{\gamma\in[0,r_L]}\left( p_L, p_L+\gamma\begin{bmatrix}\cos(\alpha)\\ \sin(\alpha)\end{bmatrix}\right) \Leftrightarrow \left( p_L, p_L+\gamma\begin{bmatrix}\cos(\alpha)\\ \sin(\alpha)\end{bmatrix}\right) \in \bigcup_E \qquad (2.1)$$

$$m(p_L, \alpha) = \begin{cases} m(p_L, \alpha), & m(p_L, \alpha) < r_L \\ r_L, & m(p_L, \alpha) \geq r_L \end{cases} \quad (2.2)$$

Denklem (2.1) ve Denklem (2.2)'de m ile gösterilen fonksiyon LIDAR mesafe ölçüm fonksiyonudur. Verilen modeldeki engel konumları, iç mekân haritasına ait koordinat sistemindeki konumlara göre tanımlanmış mutlak konumlardır.

## 2.1. Statik Yol Planlayıcı

Yol planlama metodunun başlangıç fazını statik yol planlama işlemi oluşturmaktadır. Harita bilgisinin olmaması durumunda, verilen hedefe doğrudan bir yol üretilir, statik yol planlama algoritması koşturulmaz. Bilinen bir iç mekân için hücre tipi (grid cell) bir harita modeli oluşturulmalıdır. Hücre tipi harita modeli binary imge olarak tanımlanır, lojik 1 ve 0 değerleri engel veya serbest uzay olarak belirlenir. Aynı boyutlarda bir maliyet haritası oluşturularak harita üzerinde bir başlangıç noktasından herhangi bir noktaya gitme maliyeti değerlendirilir. Maliyet haritası oluşturmak için literatürde zaman maliyeti en düşük yöntemlerden olan Dijkstra algoritması kullanılmıştır. Satır tarama tabanlı metotlar (raster scan methods) statik harita üzerinde satır bazında ileri ve geri yönlü olarak birden fazla turda sonuca ulaşmaktadır. Haritanın konkav tipinde çok sayıda engel içermesi, tarama sayısını arttırırken işlem süresini dramatik olarak yükseltmektedir. Dijkstra algoritması, maliyet haritasını satır tarama şeklinde gerçekleştirmez. Brushfire veya wavefront algoritmalarına [36] benzer bir karakteristikte, tarama bölgesinde merkezden dışa doğru yayılmacı bir yol izler. Komşu düğümlere yayılarak parent-child [29] ve açık-kapalı liste gibi yönteme özgü tanımlamalar ile etkili bir tarama yöntemi sunar.

Satır bazlı tarama yaparak mesafe dönüşümü haritasının çıkarılması ve bu harita kullanılarak statik yol planlama önerisi ilk olarak Jarvis tarafından distance transform [42] ismiyle ortaya çıkarılmıştır. Başlangıçta imge işleme tekniği olarak kullanılan bu yöntemi yol planlama metoduna dönüştürmek için, belirlenen hedef noktasından (imge uzamsal bölgesinde bir nokta) başlayarak tanımlı komşuluk pencereleri kullanılmaktadır. Erişilebilir komşu pikseller üzerinde, merkezinde güncel piksel olan bir pencereleme yapılır. Pencere elemanları ile kesişen hücreler üzerinde engellenmiş noktalar ihmal edilerek, serbest uzaydaki her noktaya mesafe dönüşümü uygulanır. Güncel pikselin mesafe dönüşüm değeri ile pencere elemanını toplamı

hesaplanır. Pencere elemanlarına karşılık gelen komşu pikselin mesafe dönüşüm değeri hesaplanan toplam değeri olarak kaydedilir. Pencere ileri yönlü (sol üst köşeden sağ alt köşeye) ve geri yönlü (sağ alt köşeden sol üst köşeye) olarak kaydırılarak herhangi bir pikselin mesafe dönüşüm değeri değişmeyene kadar bu işlem tekrarlanır. Bu yöntemde haritanın karmaşıklığı (engel sayısı, konkav, konveks objelerin sayısı, labirent yapısı vb.) arttıkça optimal sonuç almak için yapılması gereken ileri ve geri yönlü tarama sayısı da artmaktadır. Mesafe dönüşümü yapılan haritanın çalışma uzayı iki boyutludur. Genel çalışma uzayı üzerinde tanımlı, engellenmiş ve serbest bölgeleri içeren harita, binary olarak tanımlıdır. Genel çalışma uzayı ve harita tanımı Denklem (2.3)'te gösterilen formatta tanımlanmıştır.

$$\Omega=\{0,1,2\ldots\ldots,n\text{-}1\}\,x\,\{0,1,2\ldots\ldots,m\text{-}1\},\quad I{:}\Omega\rightarrow\{0,1\} \tag{2.3}$$

Harita üzerinde lojik 1 ile tanımlı hücreler (I(x,y) = 1) engellenmiş, lojik 0 ile gösterilen hücreler (I(x,y)=0) ise serbest alanlardır. Genel çalışma uzayı üzerinde tanımlanmış bir diğer matris ise mesafe dönüşüm matrisidir. Bu matris, bilinen bir başlangıç konumu için harita (I) üzerindeki herhangi bir başlangıç noktasına ulaşma maliyetini saklayan bir matristir. Elemanlarının başlangıç değerleri, hedef konumuna karşılık gelen eleman haricinde teorik sonsuz değeridir. Hedef konumu için başlangıç mesafe dönüşüm değeri ise 0'dır.

$$d(x,y)=\begin{cases}0, & h(x,y)\\ \infty, & h(x,y)^{c}\end{cases},\quad d\in D,\ D{:}\Omega\rightarrow R^{+} \tag{2.4}$$

Mesafe dönüşüm matrisinin tanımı ve terim bazında başlangıç değer ataması Denklem (2.4)'te verilmiştir.

Tablo 2.1. Jarvis mesafe dönüşüm algoritması tarama pencereleri

| Pencere tipi | İleri yönlü tarama (→↓) | Geri yönlü tarama (←↑) |
|---|---|---|
| Euclidian penceresi | $\begin{bmatrix}\sqrt{2} & 1 & \sqrt{2}\\ 1 & C & -\\ - & - & -\end{bmatrix}$ | $\begin{bmatrix}- & - & -\\ - & C & 1\\ \sqrt{2} & 1 & \sqrt{2}\end{bmatrix}$ |
| Chessboard penceresi | $\begin{bmatrix}1 & 1 & 1\\ 1 & C & -\\ - & - & -\end{bmatrix}$ | $\begin{bmatrix}- & - & -\\ - & C & 1\\ 1 & 1 & 1\end{bmatrix}$ |
| City block penceresi | $\begin{bmatrix}- & 1 & -\\ 1 & C & -\\ - & - & -\end{bmatrix}$ | $\begin{bmatrix}- & - & -\\ - & C & 1\\ - & 1 & -\end{bmatrix}$ |

Denklem (2.4)'te h(x,y) noktası I haritası üzerindeki keyfi bir noktadır. Başlangıç durumunda hedef olarak bilinen nokta dışındaki tüm hücrelere çok büyük sayısal değerler atanır. Bu aşamada ileri ve geri yönlü pencereleme işlemi başlatılır. İşlem yapılan hücre, pencere matrisinin merkezinde olacak şekilde lokal komşular üzerinde Tablo 2.1'de verilen pencere modellerinden biri kullanılarak Denklem (2.5)'te tanımlanan işlemler sırasıyla yapılır. İlgili pencere elemanı ile, işlem yapılan hücrenin mesafe dönüşüm değerinin toplamlarının en küçük değeri, merkez hücrenin mesafe dönüşüm değeri olarak belirlenir.

Yapılan pencerelemenin amacı işlem yapılan hücrenin (C) mesafe dönüşüm değerini güncel durumda komşularının değerini baz alarak minimize etmektir.

$$d(x,y)=\min\begin{cases} d(x-1,y-1)+p(-1,-1), \\ d(x,y-1)+p(0,-1), \\ d(x+1,y-1)+p(1,-1), \\ d(x-1,y)+p(-1,0), \\ d(x,y) \end{cases} \rightarrow d(x,y)=\min\begin{cases} d(x+1,y+1)+p(1,1), \\ d(x,y+1)+p(0,1), \\ d(x-1,y+1)+p(-1,1), \\ d(x+1,y)+p(1,0), \\ d(x,y) \end{cases} \quad (2.5)$$

İleri ve geri yönlü taramalar hücrelerin mesafe dönüşüm değerleri (d(x,y)) sabit kalana kadar sürdürülür. Algoritmanın pseudo kodu Şekil 2.3'te verilmiştir.

Algoritmanın çıktısı olarak elde edilen mesafe dönüşüm matrisi üzerinde, bir hedef noktasından haritanın diğer tüm noktalarına azalarak devam eden mesafe dönüşüm değerlerine sahip bitişik hücreler yer almaktadır. Herhangi bir başlangıç noktası için, her seferinde en düşük mesafe değerine sahip 8 komşudan en düşük maliyete sahip olan seçilir. Bu işlem sonucunda iki nokta arasında statik bir yol hesaplanmış olmaktadır. 8 komşulu pencereleme yapılarak hesaplanan mesafe dönüşüm matrisinde, bir hücreden komşulara yapılabilecek hareketin açısı $45^o$ ve katları ile kısıtlıdır. Oluşturulan yolun bu anlamda optimal olması söz konusu değildir. Jarvis satır tarama metodu ile optimal altı bir çözüm üretilebilir.

Şekil 2.4'te örnek bir hücresel harita (grid map) üzerinde mesafe dönüşümü ve yol planlama işlemleri gösterilmiştir. 10 x 10 boyutunda bir harita üzerinde H karakteri ile etiketlenmiş hedef noktası için haritanın mesafe dönüşüm matrisi Tablo 2.2'de

gösterilmiştir. Koyu renkli hücreler engellerdir. Tabloda i ve j olarak gösterilen parametreler sırasıyla satır ve sütun indeksleridir. Pencere türü olarak Euclidian penceresi uygulanmış, engellenmiş bölgelerin mesafe dönüşüm elemanlarına -1 değeri atanmıştır. Harita üzerinde serbest bölgede keyfi bir nokta olan, B harfi ile etiketlenmiş başlangıç hücresinden hedefe giden statik yolun hesaplanmasında greedy algoritma kullanılmıştır.

```
Import: Binary harita(I)
Run:Denklem (2.4)-Başlangıç değer tayin et
Do
  For each line=1 to #satır, Do
    For each grid=1 to #sütun, Do
      Run:Tablo 2.1-İleri yönlü pencere uygula
      Run:Denklem (2.5)-D matrisini güncelle
    End
  End
  For each line=#satır to 1, Do
    For each grid=#sütun to 1, Do
      Run:Tablo 2.1-Geri yönlü pencere
      Run:Denklem (2.5)-D matrisini güncelle
    End
  End
While (∇(∀d)=0), d ∈D
```

Şekil 2.3. Jarvis mesafe dönüşüm algoritması pseudo kodu

Bu algoritmada başlangıç olarak verilen hücrenin erişilebilir tüm komşuları kontrol edilerek, minimum değerdeki komşu hücre bir sonraki adımda seçilen hücre olarak kabul edilir. Yeni hücre için de hedefe ulaşma durumu kontrol edilerek 8 adet lokal komşunun kontrolü yapılır. Hedef hücreye ulaşıldığında algoritma durdurulur. Şekil 2.4'te verilen örnek uygulama için algoritmanın izlediği yol indeksler ve greedy algoritma tarafından seçilen hücre maliyetleri Tablo 2.3'te verilmiştir. Her işlem adımında 8 komşu arasından seçilen en düşük maliyetli komşunun indeksi ve mesafe dönüşüm maliyetleri sıralı olarak verilmiştir.

Jarvis tarafından önerilen satır bazlı mesafe dönüşümü yöntemi, karmaşık haritalar üzerinde çok sayıda ileri ve geri yönde pencereleme yapmak zorunda kalmaktadır. Mesafe dönüşümünü yerine maliyet haritası hesaplamak ve bu dönüşümü kullanarak statik yol planlamak için alternatif bir yöntem olan Dijkstra algoritması bahsi geçen problemlere karşı etkin bir çözüm getirmektedir. Bir harita üzerinde bir başlangıç noktasının diğer tüm noktalar için maliyet değerleri merkezden yayılma şeklinde bir karakteristikte hesaplanır. Harita üzerinde düğüm, kenar, düğüm maliyeti ve kenar maliyeti parametreleri tanımlanmıştır. Her düğüm serbest ya da engellenmiş olarak tanımlanır. Harita modeli Denklem (2.3)'te verilen formatta tanımlanmaktadır. Haritanın başlangıçtaki düğüm maliyetleri de Denklem (2.4)'te verilen şekilde tanımlanmaktadır.

Tablo 2.2. Jarvis yaklaşımı ile planlanan yolun mesafe dönüşüm matrisi

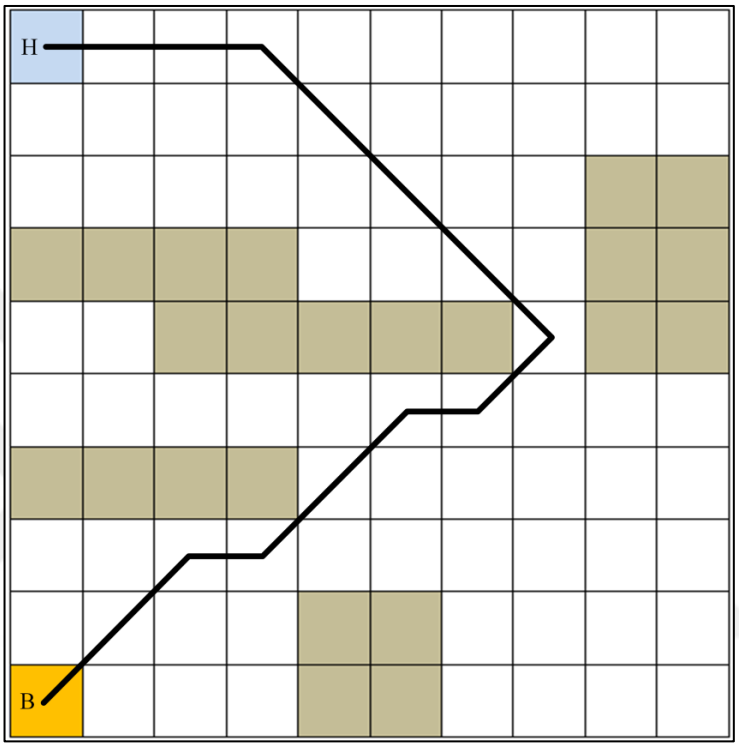
| **j** / **i** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 0,00 | 1,00 | 2,00 | 3,00 | 4,00 | 5,00 | 6,00 | 7,00 | 8,00 | 9,00 |
| **2** | 1,00 | 1,41 | 2,41 | 3,41 | 4,41 | 5,41 | 6,41 | 7,41 | 8,41 | 9,41 |
| **3** | 2,00 | 2,41 | 2,83 | 3,83 | 4,83 | 5,83 | 6,83 | 7,83 | -1 | -1 |
| **4** | -1 | -1 | -1 | -1 | 5,24 | 6,24 | 7,24 | 8,24 | -1 | -1 |
| **5** | 16,49 | 15,49 | -1 | -1 | -1 | -1 | -1 | 8,66 | -1 | -1 |
| **6** | 16,07 | 15,07 | 14,07 | 13,07 | 12,07 | 11,07 | 10,07 | 9,66 | 10,07 | 11,07 |
| **7** | -1 | -1 | -1 | -1 | 12,49 | 11,49 | 11,07 | 10,66 | 11,07 | 11,49 |
| **8** | 16,90 | 15,90 | 14,90 | 13,90 | 12,90 | 12,49 | 12,07 | 11,66 | 12,07 | 12,49 |
| **9** | 17,31 | 16,31 | 15,31 | 14,31 | -1 | -1 | 13,07 | 12,66 | 13,07 | 13,49 |
| **10** | 17,73 | 16,73 | 15,73 | 15,31 | -1 | -1 | 14,07 | 13,66 | 14,07 | 14,49 |

Tablo 2.3. Greedy algoritmanın işlem adımları

| **İşlem adımı** | **Seçilen indeksler (i,j)** | **Mesafe dönüşüm maliyeti** | **İşlem adımı** | **Seçilen indeksler (i,j)** | **Mesafe dönüşüm maliyeti** |
|---|---|---|---|---|---|
| **1** | (9,2) | 16,31 | **8** | (4,7) | 7,24 |
| **2** | (8,3) | 14,90 | **9** | (3,6) | 5,83 |
| **3** | (8,4) | 13,90 | **10** | (2,5) | 4,41 |
| **4** | (7,5) | 12,49 | **11** | (1,4) | 3,00 |
| **5** | (6,6) | 11,07 | **12** | (1,3) | 2,00 |
| **6** | (6,7) | 10,07 | **13** | (1,2) | 1,00 |
| **7** | (5,8) | 8,66 | **14** | (1,1) | 0,00 |

İşlem yapılan düğüm güncel düğüm, güncel düğüme bitişik düğümlere komşu düğümler olarak tanımlanmıştır. Düğümlerin işlem yapılma durumuna göre açık liste ve kapalı listeler tanımlanmış, başlangıçta tüm düğümler kapalı listeye dâhil

edilmiştir. Güncel düğüm, komşu düğümlerden erişilebilir olanlara düğüm maliyetleri tayin eder. Bir düğümün kapalı listeden açık listeye geçmesi için, kapalı listede bulunan tüm komşularına düğüm maliyeti tayin etmesi gerekmektedir. Bir düğüm için farklı komşuları tarafından farklı düğüm maliyetleri hesaplanması durumunda her zaman en küçük maliyet ile güncel maliyet güncellenir.




Şekil 2.4. Jarvis yaklaşımı ile planlanan örnek statik yol

Güncel düğüm tüm komşularına birer maliyet hesaplayıp, kapalı listeden açık listeye aktarıldıktan sonra, yeni güncel düğüm belirlenir. Yeni güncel düğüm, kapalı listede bulunan en düşük düğüm maliyetine sahip düğüm olarak belirlenir. Algoritma hem maliyet haritası oluşturmak hem de yol planlamak için uygulanabilir. Algoritmayı durdurma kriterlerinde kullanma amacı belirleyicidir. Belli iki nokta arasında yol planlama amacıyla uygulanan Dijkstra algoritması, hedef nokta açık listeye dâhil edildiğinde durdurulur. Bu durumda hedef haricindeki kapalı listede bulunan düğümlerin maliyet değerleri başlangıç değerlerinde kalır ve belirsizdir. Algoritma kapalı listede eleman kalmaması durumunda sonlandırılır. Bu durum tüm erişilebilir düğümlere düğüm maliyeti atanmış ve bu düğümlerin komşu düğümlerine maliyet verilmiş olduğunu gösterir. Uygulama amacından bağımsız olarak, kapalı listedeki en düşük maliyetin başlangıç değeri olarak atanan görece çok yüksek (sonsuz) değere

eşit olması durumunda sonlandırılır. Bu tip bir durum, maliyet haritası hesaplanan referans düğümü ile harita üzerinde belli düğüm ya da düğümlerin fiziksel olarak erişilemez olması anlamına gelmektedir. Algoritmanın pseudo kodu Şekil 2.5'te verilmiştir.

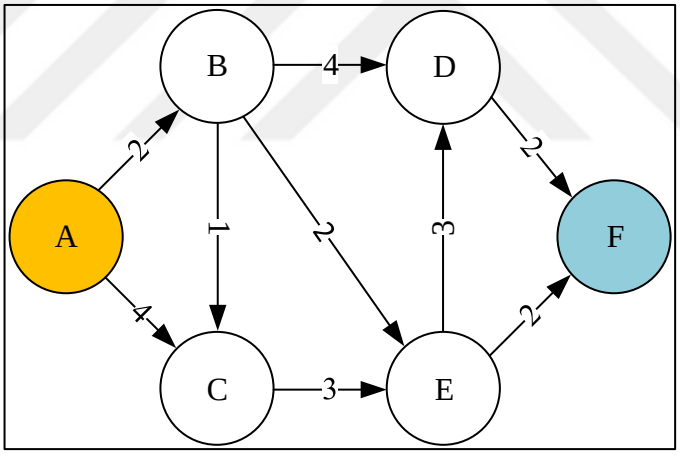
Dijkstra algoritmasının genel çalışma prensibine örnek olarak verilen problemin şeması Şekil 2.6'da verilmiştir.

**Import:** Binary harita(I)
**Run**: Denklem (2.4)-Başlangıç değer tayin et
Kapalı liste oluştur (Tüm düğümleri kapalı listeye ata): {KL}
Açık liste oluştur (Başlangıçta boş küme): {AL}
t: Hedef düğümü
Güncel düğüm belirle(Başlangıç düğümü): g
m[]: Her düğümün maliyeti
dist(a,b): a düğümü ile b düğümü arasındaki uzaklık
d[]: Güncel düğümün erişilebilir komşuları
**Do**
  **For each** i in d[]
    n=dist(g,i)+g(i)
    m(i)=min(n,m(i))
  **End**
g →{AL}
{KL}={KL}\g
$g=x_{min(x \in \{KL\}|m(x))}$
**While** ({KL}≠Ø) V (t∈{AL}) V (min(m(x)|x ∈{KL})=∞)

Şekil 2.5. Dijkstra algoritması pseudo kodu

Şekil 2.6'da A ile F arasında etiketlenen düğümler harita üzerinde serbest bölgeye ait düğümlerdir. Okların yönü ilerleme kısıtını göstermektedir. Okların üzerindeki büyüklükler düğümler arasında tanımlı hareket maliyetidir. Bir düğümden, hareket kısıtı dâhilinde komşu düğüme ilerlemenin maliyeti bu parametre ile tanımlanmıştır. Düğümlerin kendi üzerinde tanımlı düğüm maliyetleri ayrıca tanımlanacaktır. Hedef olarak F düğümü belirlenirken A düğümü başlangıç düğümü olarak seçilmiştir.

Başlangıç düğümünden başlanarak hareket yönünde bulunan her komşu düğüm için düğüm maliyeti hesaplanır. Güncel düğümün herhangi bir komşu düğüm için hesapladığı maliyet, kendi maliyeti ile ilgili komşuyla tanımlı hareket maliyetinin toplamıdır. Şekil 2.6'da verilen düğüm yapısı için; A düğümünün C düğümü için hesapladığı düğüm maliyeti kendi maliyeti (0) ve A düğümünden C düğümüne olan hareket maliyetinin (4) toplamıdır. Ancak C düğümüne komşu olan bir diğer düğüm olan B tarafından da C için tanımlanmış bir düğüm maliyeti vardır. B düğümünün kendi maliyeti A tarafından belirlenen 2 değeridir. C'nin B tarafından oluşturulan maliyeti B'den C'ye hareket maliyeti (1) ve B'nin maliyetinin (2) toplamıdır. Sonuç olarak C düğümü için iki farklı komşusundan iki farklı düğüm maliyeti oluşturulur. Bunlar arasından en düşük değerli olan tercih edileceği için C'nin düğüm maliyeti 3 olarak belirlenir. En düşük maliyetin alındığı komşu "parent", maliyetin atandığı düğüm ise "child" olarak değerlendirilir. Tablo 2.4'te her düğüm için her adımda tanımlanan düğüm maliyetleri verilmiştir.




Şekil 2.6. Dijkstra yaklaşımı için tanımlanan bir problem

Adımlar 1-7 arasında numaralandırılmış olup düğümlerin son durumdaki maliyetleri Tablo 2.6'da verilmiştir. Maliyet büyüklükleri üzerinde gösterilen üst indisler bu maliyetlerin hangi komşu tarafından oluşturulduğunu göstermektedir. İlgili düğümün parent düğümü bu indislerle verilmiştir. Örneğin E düğümü için tanımlı maliyet $4^{B}$ olarak verilmiştir. Bu durumda B düğümü, E düğümü için parent olarak belirlenir. Tablo 2.4'te g sembolü ile gösterilen sütun güncel düğümü temsil etmektedir. Güncel düğüm, hareket kısıtları nedeniyle erişemediği komşular veya komşu olmayan uzak düğümler için maliyet hesaplayamaz. Bu düğümlere teorik sonsuz ifadesi atanır.

Tablo 2.4'te verilen işlem adımlarında kısaltılarak verilen açık liste ve kapalı listelerin işlem sırasına göre elemanları Tablo 2.5'te verilmiştir. Güncel düğüm erişim yetkisi olan tüm düğümlere maliyet önerdikten sonra kapalı listeden çıkarılarak açık listeye aktarılır. Son adımda ise tüm düğümlere ulaşıldığı için kapalı liste boş küme olmaktadır.

Tablo 2.4. Dijkstra algoritması işlem adımları

| {KL} | {AL} | g | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|
| $K_1$ | $A_1$ | - | $\infty$ | $\infty$ | $\infty$ | $\infty$ | $\infty$ | $\infty$ |
| $K_2$ | $A_2$ | A | $0^A$ | $2^A$ | $4^A$ | $\infty^A$ | $\infty^A$ | $\infty^A$ |
| $K_3$ | $A_3$ | B | $0^A$ | $2^A$ | $3^B$ | $6^B$ | $4^B$ | $\infty^A$ |
| $K_4$ | $A_4$ | C | $0^A$ | $2^A$ | $3^B$ | $6^B$ | $4^B$ | $\infty^A$ |
| $K_5$ | $A_5$ | D | $0^A$ | $2^A$ | $3^B$ | $6^B$ | $4^B$ | $6^E$ |
| $K_6$ | $A_6$ | E | $0^A$ | $2^A$ | $3^B$ | $6^B$ | $4^B$ | $6^E$ |
| $K_7$ | $A_7$ | F | $0^A$ | $2^A$ | $3^B$ | $6^B$ | $4^B$ | $6^E$ |

Tablo 2.5. Dijkstra algoritması kapalı ve açık listelerin değişimi

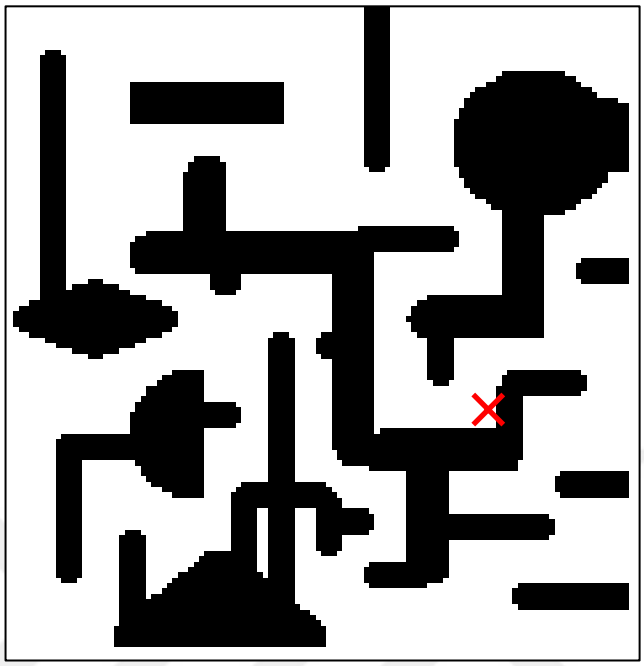
| Adım | Kapalı liste (K) | Açık liste (A) |
|---|---|---|
| 1 | $K_1$={A,B,C,D,E,F} | $A_1$= Ø |
| 2 | $K_2$={B,C,D,E,F} | $A_2$={A} |
| 3 | $K_3$={C,D,E,F} | $A_3$={ A,B} |
| 4 | $K_4$={D,E,F} | $A_4$={ A,B,C} |
| 5 | $K_5$={E,F} | $A_5$={ A,B,C,D} |
| 6 | $K_6$={F} | $A_6$={ A,B,C,D,E} |
| 7 | $K_7$= Ø | $A_7$={A,B,C,D,E,F} |

Tablo 2.6. Düğümlerin nihai maliyetleri

| A | B | C | D | E | F |
|---|---|---|---|---|---|
| 0 | 2 | 3 | 6 | 4 | 6 |

İmgesel bir harita üzerinde Dijkstra algoritması uygulanırken teoride rasgele düğümler belirlenerek bu düğümler üzerinde algoritma koşturulabilir. Ancak uygulamada robotun çalışma uzayı sürekli bir aralıkta tanımlıdır. Bu aralık örneklenirken yüksek çözünürlükte bir harita tanımı yapılması istenilen bir

durumdur. İmgesel bir haritanın erişilebilir her pikseli bu anlamda bir düğüm olarak ele alınmaktadır.



Şekil 2.7. Örnek statik engel haritası



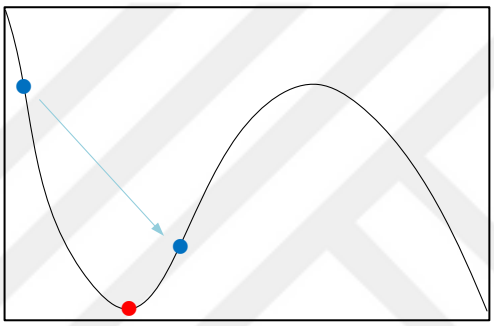
Şekil 2.8. Üretilen maliyet haritası

İmge uzamsal bölgesinde uygulamaya göre değişiklik gösterebilmekle birlikte, yaygın kullanımda siyah (lojik 0) pikseller (düğümler) engellenmiş, beyaz (lojik 1) pikseller serbest uzayı temsil etmektedir. Matematiksel modeli Denklem (2.3)'te verilen örnek bir engel haritası Şekil 2.7'de gösterilmiştir. Oluşturulan maliyet

haritası ise Şekil 2.8'de gösterilmiştir. Harita üzerinde × sembolü ile gösterilen konum Dijkstra algoritması için verilen başlangıç noktasıdır.

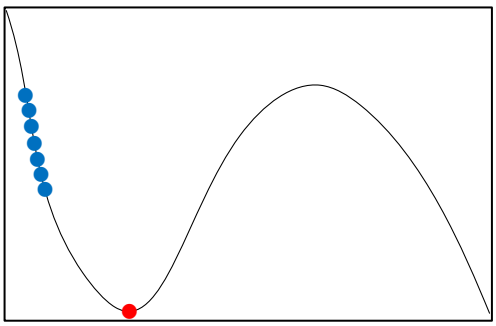
Yol planlama aşamasında kullanılan gradient descent yönteminin matematiksel ifadesi Denklem (2.6)'da verilmiştir. Denklemde f fonksiyonunun türevi bir çözüm katsayısı (α) ile ağırlıklandırılarak anlık giriş parametresinden iteratif bir şekilde çıkarılır. Türev ifadesinin 0'a belli bir tolerans ile yakınsaması durumunda algoritma sonlandırılır. Bu koşulu sağlayan x parametresi çözüm olarak kabul edilir.

$$x_i = x_{i-1} - \alpha \frac{\partial}{\partial_x} f(x_{i-1}) \tag{2.6}$$



Şekil 2.9. α katsayısının çok büyük seçilmesi durumu

Şekil 2.9'da α katsayısının çok büyük seçilmesi durumunda çözüm adımlarının lokal minimum noktasını atlayarak kararsızlığa sebep olduğu görülmektedir.

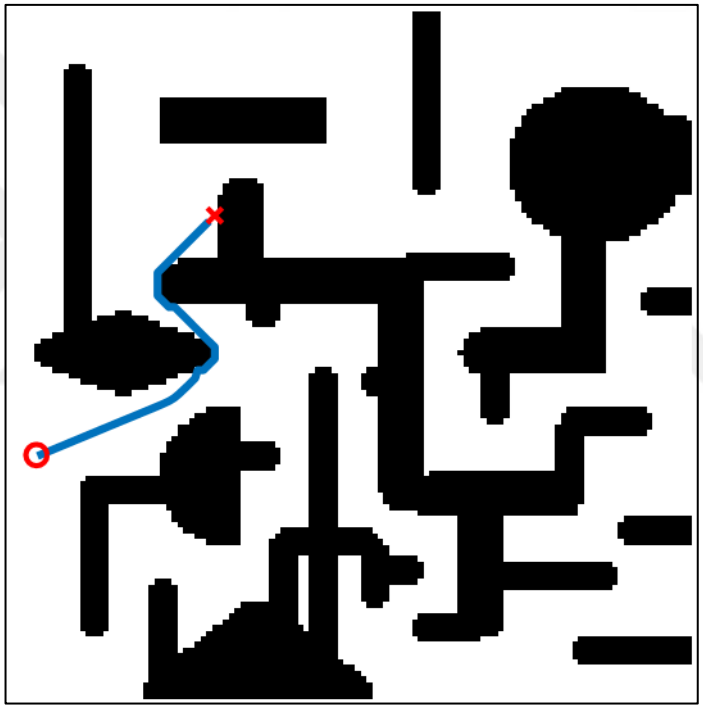


Şekil 2.10. α katsayısının çok küçük seçilmesi durumu

Şekil 2.10'da verilen durum gibi, katsayının çok küçük değerlerde tutulması durumunda ise yüksek hassasiyetle bir tarama yapılır ancak işlem süresi aşırı yükselebilir. Bu katsayı çözüm uzayının limitleri gözetilerek deneme-yanılma

yöntemi ile belirlenebilir. Dijkstra algoritmasının Şekil 2.7'de verilen statik harita üzerinde verilen konum için ürettiği maliyet haritası Şekil 2.8'de verilmiştir. Haritanın uzamsal boyutu 120x120 pikseldir. Maliyet değerleri 0-255 arasında işaretsiz tamsayı değerleri ile normalize edilerek gri seviye imge olarak verilmiştir. Başlangıç düğümünün, düğüm maliyeti 0 olmak üzere, normalize edilmiş maliyetler haritanın ilgili düğümlerine tayin edilmiştir.

Elde edilen dönüşüm haritası, statik yol planlama aşamasında kullanılmaktadır. Yol planlama aşamasında Hill climbing, Gradient descent ve literatürde bulunan diğer greedy algoritmalar uygulanabilmektedir.



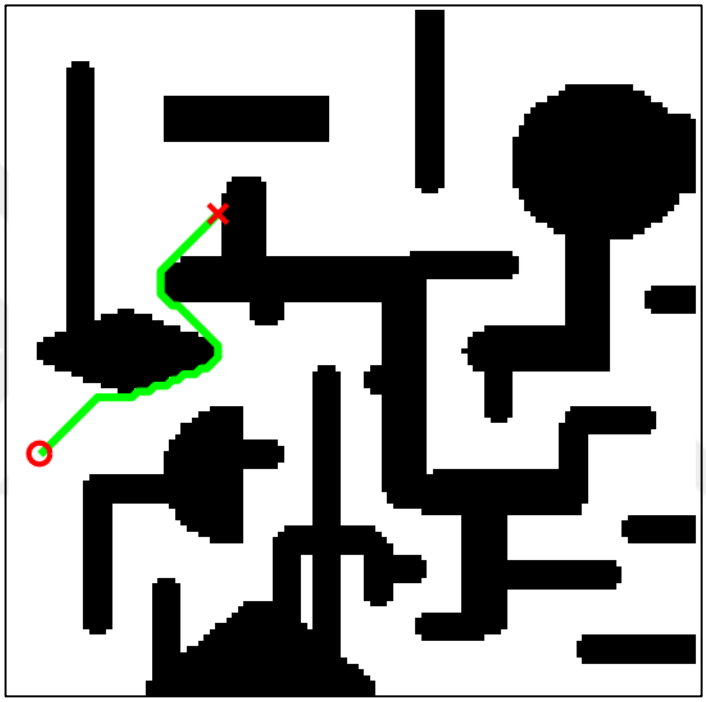
Şekil 2.11. Gradient descent algoritması ile planlanan statik yol

İşlem süresi ve planlanan yolun manevra açısındaki limitler, yol uzunluğu gibi fiziksel özellikler bakımından gradient descent algoritması, greedy algoritmaya göre daha etkin bir yol planlama karakteristiği sergilemektedir.

Şekil 2.7'de verilen statik harita ve Şekil 2.8'de verilen Dijkstra algoritması ile elde edilen maliyet haritası üzerinde iki farklı algoritma kullanılarak statik yol planlama işlemi yapılmıştır. Gradient descent ve greedy algoritma için harita üzerinde seçilmiş

keyfi iki nokta arasında planlanan statik yol profilleri, sırasıyla Şekil 2.11 ve Şekil 2.12'da verilmiştir.

Planlanan yol uzunlukları ve yol planlama işlem süreleri bakımından iki yöntemin karşılaştırılması Tablo 2.7'de verilmiştir. Statik yol uzunluğu, harita üzerinde piksel bazında kat edilen uzunluktur. İşlem süreleri bazında karşılaştırma yapılırken, aynı harita konfigürasyonu için rasgele zamanlarda yapılan 5 planlama işleminin ortalama işlem süreleri verilmiştir.



Şekil 2.12. Greedy algoritma ile planlanan statik yol

Yol uzunluğu ve ortalama işlem süreleri bazında gradient descent algoritmasının Dijkstra algoritması ile entegrasyonu daha yüksek statik yol planlama performansı sağlamıştır.

Tablo 2.7. GD ve greedy algoritmalarının yol planlama çıktıları

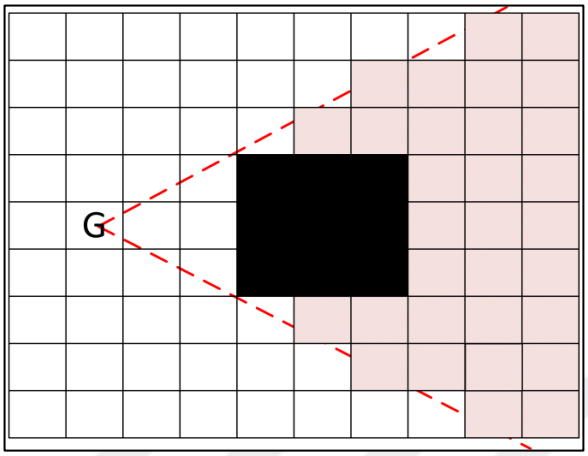
| | Yol uzunluğu (Piksel) | İşlem süresi (Milisaniye) |
|---|---|---|
| Gradient descent | 85.9312 | 15.185 |
| Greedy arama | 97.6518 | 25.478 |

## 2.2. Dinamik Yol Planlayıcı

Bilinen harita olması durumunda yol planlama modülünün statik yol planlama fazı tarafından global hedef ile verilen başlangıç konumu arasında statik yol planlanır. Bu yol doğrudan takip edilirse engel profilindeki değişikliklerden kaynaklı istenmeyen durumlar ortaya çıkabilir. Başlangıçta serbest uzayda tanımlı bir bölgenin sonraki bir işlem zamanında engellenmesi, engelin kaldırılması veya kinematik kısıtlardan kaynaklı olarak yapılamayan keskin dönüşler, gerçek statik yol takibi sırasında karşılaşılabilecek olumsuz durumlara örneklerdir. Bu ve benzeri durumları gidermek için genel yol planlama şeması altında dinamik yol planlama fazı tanımlanmıştır. Statik yol takibi ile dinamik yol planlama arasında geçişi kontrol etmek amacıyla, robot dönme merkezi ile eş merkezli bir sanal ileri bakış çemberi (IBC) tanımlanmıştır. Planlanan statik yol ile bu çemberin kesişim noktaları gerçek zamanlı olarak hesaplanarak lokal hedef tespiti gerçekleştirilmiştir. IBC görüş alanı içerisinde statik haritada oluşan değişimler de taranmıştır. Başlangıçta planlanan statik yol daha sonraki bir işlem adımında bir engel tarafından bloke edilirse bu durumun tespiti de IBC sınırlarında işlenen LIDAR verileri ile gerçekleştirilmiş, statik haritanın anlık değişimi ile IBC sınırlarında kalan alt uzayda statik yol takip edilemez durumda ise dinamik planlama aşamasına geçiş yapılmıştır. Lokal hedef tespiti için statik yol ile IBC kesişimleri yerine açıklık tespiti tabanlı yaklaşım uygulanmıştır. Her iki durumda belirlenen lokal hedef noktası çarpışma engelleme algoritması tarafından önceden tanımlı çarpışma koşulu kontrol edilerek revize edilmiştir. Son durumda belirlenen nihai lokal hedef hız kontrolörüne iletilerek robotun ilgili noktaya navigasyonu gerçekleştirilmiştir. İşlem adımlarının tamamında global hedefe belirli bir tolerans bandında yaklaşımı kontrol edilerek bu koşul sağlanmışsa algoritma sonlandırılmıştır. IBC sınırları dışında, sensörün görüş limitlerinde global hedef ile robot arasında serbest uzayda doğrudan bir koridor tanımlı ise lokal hedef tespiti adımı atlanarak robotun doğrudan hedefe hareketi sağlanmıştır. Bu durumda global hedef sanki lokal bir hedefmiş gibi varsayılarak işlenmiş ve çarpışma engelleme algoritması tarafından kontrol edilerek motor hız kontrolörüne, uygun kontrol işareti gönderilmiştir.

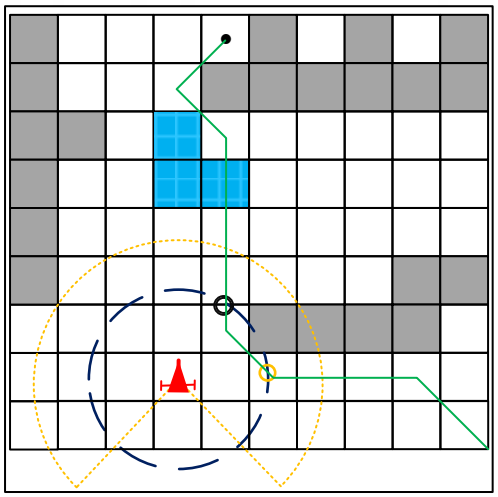
Dinamik yol planlama algoritmasının işlem adımları açıklanırken görünürlük kavramından söz edilmelidir. Örneğin global hedefin LIDAR tarafından görünür

olmasının tanımı, LIDAR konumu ile hedef konumuna karşılık gelen harita üzerindeki hücreler arasında Bresenham doğru algoritması ile oluşturulan hat üzerinde hiçbir engellenmiş hücrenin olmamasıdır.




Şekil 2.13. Görünürlük tanımı

Robot geometrisi söz konusu olduğunda görünürlük tanımı, robot diyagonal sınırları arasında kalan bölgenin aynı yöntemle oluşturulan hat boyunca engellenmemiş olması anlamı taşımaktadır.



Şekil 2.14. Statik yol üzerinde lokal hedef seçimi

Şekil 2.13'te verilen grafikte bir G noktasında bulunan LIDAR için görüş limitlerinde bulunan lokal bir bölgenin görünürlük haritası gösterilmiştir. Siyah renkli dikdörtgen geometri engel olmak üzere diğer hücreler serbest uzaya dâhildir.

Kesik çizgiler görüş çizgileri olup, serbest bölgede koyu tonda verilen hücreler görünmez, diğer hücreler görünür olarak tanımlanmaktadır.

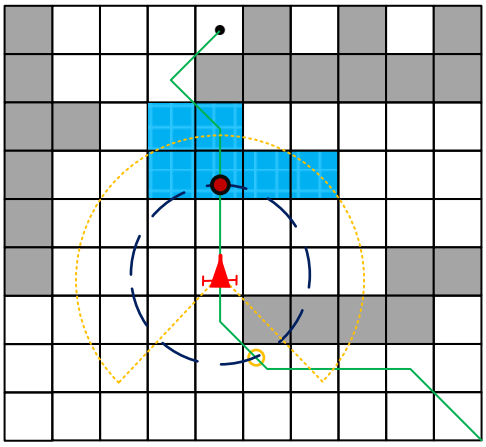
Etiketleme yapılırken hücrenin geometrik merkezi baz alınmaktadır. Robot global hedefe yöneldikten sonra IBC ile statik yolun kesişim noktaları taranır. Kesişim noktalarının birden çok olması durumunda Dijkstra metodu ile elde edilen maliyet haritasında en düşük değere sahip olan nokta lokal hedef olarak tercih edilir. Bu nokta birincil nokta, diğer noktalar varsa ikincil, üçüncül vb. olarak adlandırılmıştır. Statik yol üzerinde lokal hedef seçimi Şekil 2.14'te verilen şemada gösterilmiştir.

Şekil 2.14'te verilen sembolik durum için kullanılan semboller ve notasyon Tablo 2.8'de verilmiştir. Statik haritanın bilindiği varsayımı ile planlanan yol, IBC sınırları dışında dinamik bir engel tarafından kesilebilir. Bu durum lokal yol planlayıcı tarafından IBC sınırlarına girene kadar ihmal edilir. Statik yol ile iki farklı noktada kesişim tespit edilmesi durumunda birincil nokta olarak etiketlenen konuma karşılık gelen hücre lokal hedef olarak tercih edilir. Görünürlük testinden geçen birincil nokta çarpışma kontrolü kapsamında lokal hedef olarak belirlenir. Bazı durumlarda IBC ile statik yolun birincil kesişim noktası, dinamik bir engel tarafından bloke edilebilir. Bu duruma örnek bir şema Şekil 2.15'te verilmiştir. Birincil kesişim noktası bilinmeyen bir engel tarafından bloke edilmiş, nokta görünürlük testinden geçememiştir.

Tablo 2.8. Dinamik yol planlama notasyonu

| Simge | Tanımı |
|---|---|
| | IBC sınırları |
| | Statik yol |
| | Görünmez lokal hedef |
| | Görünür lokal hedef |
| | Global hedef |
| | Mobil Robot |
| | İkincil kesişim noktası |
| | Bilinmeyen engel |
| | Bilinen engel |
| | LIDAR tarama limiti |

Bu tip bir durumda geliştirilen açıklık tespiti algoritması ile lokal bölgede geçilebilecek açılıklar organize bir formatta tespit edilir. Bu açıklık(lar) üzerinde, önceden tanımlı maliyet fonksiyonu uygulanarak aday lokal hedef noktaları belirlenir.



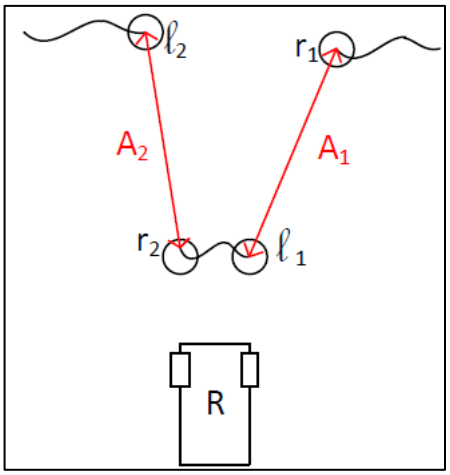
Şekil 2.15. Dinamik engel tarafından bloke edilen birincil nokta

### 2.2.1. Açıklık tespiti algoritması

Açıklık tespiti algoritması, LIDAR tabanlı lokal yol planlama yöntemlerinde uygulanabilecek bir yaklaşımla tasarlanmıştır. Temel donanım, lazer mesafe tarayıcı veya ortamın engel profilini veren herhangi bir mesafe sensörüdür. Geliştirilen yöntemin temel amacı iki boyutlu engel profiline ait mesafe verilerinin işlenerek anlamlı geçilebilir açıklıkların elde edilmesidir. Açıklık tespiti yapılırken LIDAR datası üzerinde bir ön işlem yapılarak organize olmayan açıklıklar ortaya çıkarılır. LIDAR datası mesafeler türünden Denklem (2.7)'de verilen formatta bir dizi şeklinde tanımlıdır. Dizi elemanları milimetre türünden mesafe, alt indisleri ise ilgili açı bilgisini taşımaktadır. Robot üzerinde sabitlenen LIDAR'ın optik merkezinden, belli açılarla gönderilen lazer ışınları ilgili açı doğrultusundaki ilk çarptığı yüzeyden yansıma yapar. Engellerin mesafe bilgisi sıralı olarak sensörden okunabilmektedir.

$$L = \{d_0,\ d_1,\ d_2,\ d_3,\ \ldots.\ d_{90},\ \ldots.\ d_{178},\ d_{179},\ d_{180}\} \tag{2.7}$$

Okunan mesafe ölçümünün birinci dereceden türevi alınarak elde edilen türev dizisindeki marjinal değişim gösteren indeksler taranır. Ardışık ölçümlerde tespit edilen indeksler açıklıkların köşe noktaları olarak kaydedilir. Bu aşamada hesaplanan açıklıklar, belirlenen koşullar sağlandığı takdirde yeniden organize edilmektedir. Şekil 2.16'da gösterilen şemada R etiketi robotu (sensörü), $r_1$ ve $l_1$ $A_1$ ile gösterilen açıklığın köşe noktalarını, $r_2$ ve $l_2$ de $A_2$ ile gösterilen açıklığın köşe noktalarını temsil etmektedir.




Şekil 2.16. Ön işlem sonucu tespit edilen açıklıklar

Serbest geometride verilen eğri, engel profili olarak varsayılmıştır. Engel profilinde $r_1$ noktasından $l_1$ noktasına, $r_2$ noktasından $l_2$ noktasına marjinal geçişler olması bu noktalar arasında açıklık olduğu şeklinde yorumlanmaktadır. Tespit edilen açıklıklar tanımlı bazı durumlarda revize edilmektedir. Revizyon gerektirmeyen durumlara örnek bir açıklık yapısı Şekil 2.16'da verilen düzendir.

Açıklık revizyonu, türev sonucu elde edilen köşe noktalarının ve köşe çiftlerinin sırasını ve yerini değiştirmek veya istenmeyen açıklığı ortadan kaldırmak şeklinde işlemler içermektedir. Köşelerin sırasını değiştirme işlemi iki aşamada gerçekleştirilir. Saat yönünde (CW) ve aksi yönde (CCW) yapılan tarama ile bir köşenin eşleştiği diğer köşenin, başka bir köşe ile eşleşmesi sağlanır. CCW taraması yapılırken, her açıklığın sağ köşesi ile, sol köşesinin solunda kalan diğer tüm köşelerin Öklid uzaklığı hesaplanır. Hesaplanan uzaklıklardan en düşük olanı, açıklığın başlangıç genişliğinden düşükse ilgili köşe ile başlangıçtaki sol köşe değiştirilir. Benzer işlem CW taramasında sol köşelerden başlanarak tekrarlanır. Bu işlemin temel amacı, robot bir açıklığa yönelirken öncelikle geçmesi gereken gizli açıklıkların var olmasıdır. Bunların tespiti ve önceliklendirilmesi bu işlemlerle gerçekleştirilir. Bu işlem yapılırken önemli bir güncelleme kriteri de güncellenecek olan köşenin robota (sensöre) olan Öklid uzaklığıdır. Güncelleme kriterini sağlayan yeni köşenin robota olan uzaklığı, başlangıçta hesaplanan köşe noktasının robota uzaklığından daha fazla ise güncelleme işlemi yapılmaz. Bu istisna için örnek durum Şekil 2.16'da görülmektedir. $|r_1l_2|$ uzunluğu $|r_1l_1|$ uzunluğundan daha küçüktür ancak

$|l_1R|$ uzunluğu $|l_2R|$ uzunluğundan daha küçük olduğu için $l_1$ köşesi $l_2$ ile değiştirilmez.

CCW revizyonunun matematiksel ifadesi Denklem (2.8) ve Denklem (2.9)'da verilmiştir.

$$j^* = \min_j \left( \left| r_i r_j \right| \right), \left| Rr_j \right| < \left| Rl_i \right|, \; j > i \tag{2.8}$$

$$k^* = \min_k \left( \left| r_i l_k \right| \right), \left| Rl_k \right| < \left| Rr_i \right|, \; k > i \tag{2.9}$$

CW revizyonunun matematiksel ifadesi Denklem (2.10) ve Denklem (2.11)'de verilmiştir.

$$j^* = \min_j \left( \left| l_i r_j \right| \right), \left| Rr_j \right| < \left| Rl_i \right|, \; j < i \tag{2.10}$$

$$k^* = \min_k \left( \left| l_i l_k \right| \right), \left| Rl_k \right| < \left| Rr_i \right|, \; k < i \tag{2.11}$$

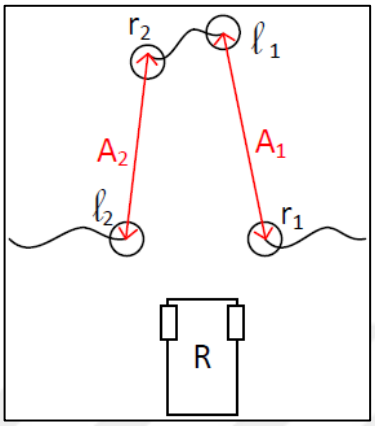
Açıklık revizyon işleminin CCW ve CW taramalarına ait Pseudo kodları Şekil 2.17'de verilmiştir. Şekil 2.17'de l, r ve R sırasıyla açıklığın sol köşesini, sağ köşesini ve robotun (sensörün) konumunu göstermektedir. İki turda açıklık konfigürasyonunun değişimini ifade eden işlemlerde $l_{i_updated}$ ve $r_{i_updated}$ terimleri sırasıyla, güncellenen sol ve sağ açıklık köşelerini ifade etmektedir.

| CCW açıklık revizyonu: | CW açıklık revizyonu: |
|---|---|
| **If** $j^* \neq$NULL $\wedge$ $k^* \neq$NULL | **If** $j^* \neq$NULL $\wedge$ $k^* \neq$NULL |
| **If** $\|r_i\ r_{j^*}\| < \|r_i\ l_{k^*}\|$ | **If** $\|l_i\ r_{j^*}\| < \|l_i\ l_{k^*}\|$ |
| $l_{i_updated} = r_{j^*}$ | $r_{i_updated} = r_{j^*}$ |
| **Else** | **Else** |
| $l_{i_updated} = l_{k^*}$ | $r_{i_updated} = l_{k^*}$ |
| **End If** | **End If** |
| **End If** | **End If** |

Şekil 2.17. Açıklık revizyonu pseudo kodu

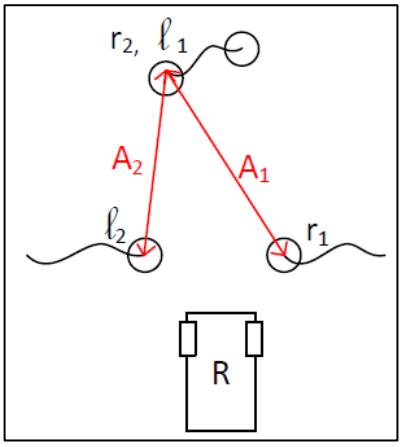
Revize edilen açıklıklar mobil robotun fiziksel boyutuna göre tekrar filtrelenerek, kritik bir genişlikten daha dar olanlar elenir. Açıklık revizyonunu adım adım gösteren örnek figürler Şekil 2.18, Şekil 2.19 ve Şekil 2.20'de gösterilmiştir. A1 ve A2 açıklıkları $r_1$, $l_1$, $r_2$, $l_2$ köşeleri ile tanımlanmaktadır. Robotun A1 ya da A2 açıklığından geçebilmesi için öncelikle $r_1$, $l_2$ arasında bulunan fiziksel açıklıktan

geçmesi gerekmektedir. Ancak türev işlemi ile bu hat doğrudan elde edilemez. Bu açıklığın köşe noktaları cinsinden ifade edilmesi için açıklıkların sağ ve sol köşeleri Denklem (2.8), Denklem (2.9), Denklem (2.10) ve Denklem (2.11)'de gösterilen işlemler Şekil 2.17'de verilen algoritmalar ile birlikte uygulanmıştır.




Şekil 2.18. Açıklıkların başlangıç düzeni

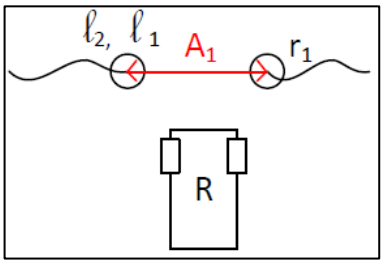
Bu işlemler sonucunda Şekil 2.20'da verilen (öncelikli olarak geçilecek) açıklık tespit edilmiştir. Bu revizyonlara ek olarak duvar tipi engellerle sonlanan açıklıklara özel bir revizyon da tanımlanmıştır.




Şekil 2.19. CCW revizyonu ile köşe güncelleme

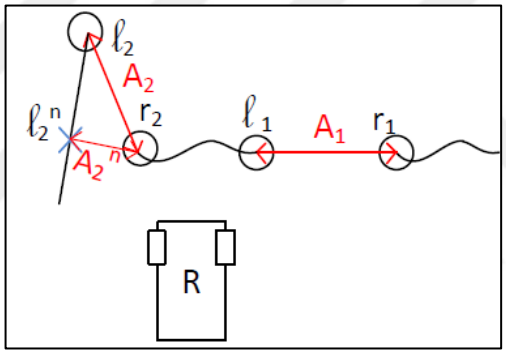
Köşelerinden bir tanesi duvar tipi engeller üzerinde bulunan açıklıklar, öncelikli olarak geçilmesi gereken sanal bir açıklığın ihmal edilmesine neden olmaktadır. Öncelikli açıklık fiziksel olarak robotun geçişine yeterli olmasa da, hesaplanan açıklığın orijinal köşe yerleşimi yeterli genişliğe sahip olabilir. Bu durum önceden

tespit edilerek ilgili açıklığın elenmesi önemli bir güvenlik uygulamasıdır. Bu duruma örnek bir şema Şekil 2.21'de gösterilmiştir.




Şekil 2.20. Açıklıkların son düzeni

Türev işlemi ve gerekli şartlar sağlanmış ise köşe noktalarının revizyonu adımlarından sonra $A_1$ ve $A_2$ açıklıkları elde edilmektedir.



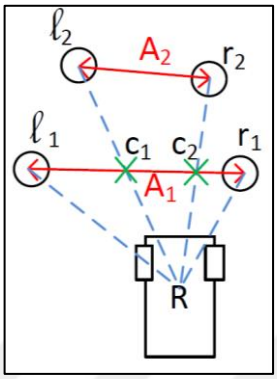

Şekil 2.21. Duvar tipi sonlanan açıklık revizyonu

$A_1$ genişlik kriterine göre geçerli ise filtrelenmez ancak $A_2$ açıklığına doğrudan bir hareket yapmak genişliği yeterli olsa da uygulanabilir bir hareket değildir.

Bu durum gözetilerek $l_2$ köşe noktası $l_2^n$ noktasıyla güncellenmiştir. Yapılan bu işlem sonrasında $A_2$ açıklığının revize edilmiş şekli $A_2^n$ olarak verilmiştir. Açıklığın sol köşesi kontrol edilirken, sol köşeden başlayarak saat yönü tersi yönde bir sonraki köşe noktasına kadar olan nokta bulutu içerisinde sağ köşeye daha yakın bir nokta varsa, sol köşe bu nokta ile güncellenir.

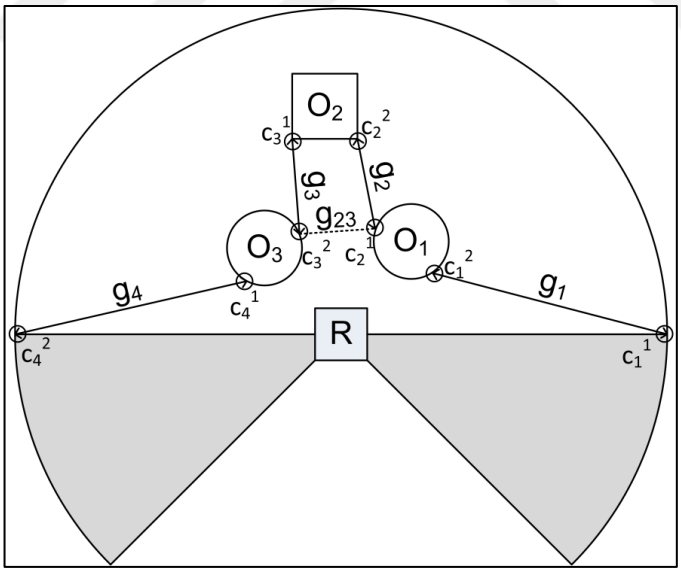
Sağ köşe kontrol edilirken, ilgili köşenin sağ düzleminde kalan, bir sonraki köşe noktasına kadar bulunan nokta bulutu için aynı işlem tekrarlanır.

Revizyon işlemlerinden sonuncusu örtüşen açıklıklar için yapılmıştır. Silinen ve güncellenen köşe noktalarının örtüşmesi problemi ile karşılaşılabilir. Bu durum, birden fazla açıklık için aynı doğrultuda hareket sağlayan işlemlere sebep olmaktadır. Bunu önlemek için yakın olan açıklık tercih edilerek örtüşen diğer açıklıklar elenir.




Şekil 2.22. Örtüşen açıklık tespiti

Sensör merkezinden her açıklığın köşelerine çizilen doğrular açıklıklardan bir tanesini iki farklı noktada kesiyorsa bu açıklıklar örtüşmektedir sonucuna varılır.



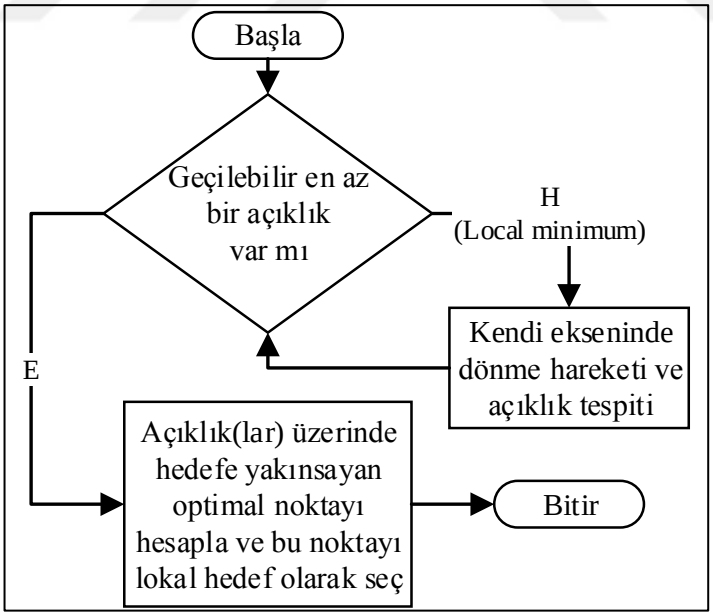

Şekil 2.23. Açıklık tespiti yöntemi ile ilgili örnek senaryo

Örnek bir şema Şekil 2.22'de verilmiştir. Burada $|l_2R|$ ve $|r_2R|$ doğruları $A_1$ açıklığını, $c_1$ ve $c_2$ noktalarında kesmiştir. Kesilen açıklık öncelikli olarak belirlenir ve diğer açıklık elenir.

Açıklık tespiti yönteminin çalışma prensibi ile ilgili örnek bir senaryo Şekil 2.23'te verilmiştir. Şekil üzerinde $g_i$, $c_i^j$, $O_k$ ile gösterilen parametreler sırasıyla i indeksli açıklık, i indeksli açıklığa ait j indeksli köşe ve k indeksli engeldir. Açıklık revizyonu yöntemi uygulanarak başlangıç durumunda tespit edilen $g_2$ ve $g_3$ açıklıkları filtrelenerek robotun öncelikli olarak geçeceği $g_{23}$ açıklığı ortaya çıkarılmıştır. Herhangi bir açıklığa indirgenemeyen $g_1$ ve $g_4$ açıklıkları ise değişikliğe uğramamıştır. Robotun marjinal dönüşler yapmaması için $0^o$ ve $180^o$ konumlarında engel olmaması durumunda ilgili noktalara sanal köşeler yerleştirilmiştir. $g_1$ ve $g_4$ açıklıklarının sırasıyla $c_1^1$ ve $c_4^2$ indeksli köşeleri bu yöntemle oluşturulmuştur. Sensörün maksimum tarama açısı ile açıklık tespiti için değerlendirmeye dâhil edilen açısal görüş limitleri farklı tutulabilmektedir. $270^o$ tarama açısına sahip LIDAR için açıklık tespiti için değerlendirilen açısal bölge $180^o$ ile kısıtlanmıştır.

### 2.2.2. Lokal hedef tespiti metodu

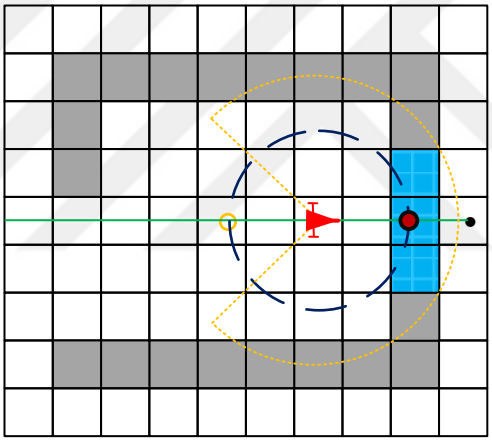
Yol planlama metodunun genel akış şemasının gösterildiği Şekil 2.1'de açıklık tespiti ve lokal hedef tespiti bloğu Şekil 2.24'te genişletilmiş olarak verilmiştir.




Şekil 2.24. Açıklık tespiti ve lokal hedef belirleme işlemi akış şeması

İlk aşamada açıklık tespiti algoritması uygulanır. Açıklık tespiti algoritması sonucunda tamamen bloke edilmiş bir lokal bölge için herhangi bir açıklık tespit edilemeyecektir. Statik yol ile IBC kesişiminin görünürlük testinden geçemediği

varsayımı altında açıklık tespit edilememesi, robotun yerel minimuma düştüğü anlamına gelmektedir. Bu durumda robot sıfır-yarıçap dönüşü (zero-radius turn) hareketi yaparak açıklık tespiti algoritması üzerinden bir lokal hedef aramaya başlar. Bu işlem sonrasında bir açıklık bulunana kadar robot belirli periyotlarla bu hareketi tekrarlar. Böyle bir durum robotun çevresinin tamamen kapatılmış bir şekilde bulunması şeklinde yorumlanır. Bu durumu görselleştiren bir figür Şekil 2.25'te gösterilmiştir. Robot başlangıçta planlanan statik yol ve IBC kesişimlerini izleyerek belirli bir konuma geldiğinde ve dinamik bir engel tarafından statik yol bloke edildiğinde açıklık tespiti modülü uygulanır ancak mevcut lokal haritada açıklık oluşturabilecek serbest alan bulunmamaktadır. Kurtarma modu olarak tanımlanan dönüş ve tarama hareketi sonrasında tespit edilen açıklık üzerinde belirlenen lokal hedefler doğrultusunda yerel minimumdan çıkılır.



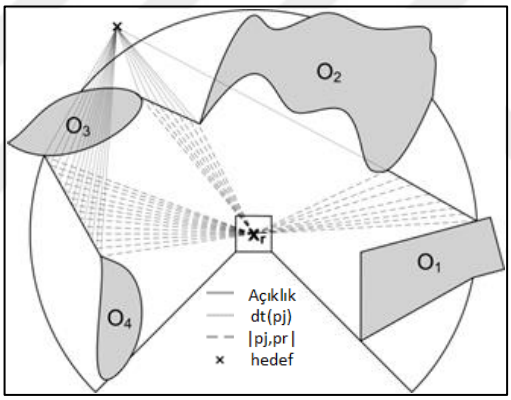
Şekil 2.25. Yerel minimum durumu

Açıklık tespiti sonrası bu açıklıklar üzerinde bulunan hücrelerin sezgisel maliyeti hesaplanarak bir çözüm uzayı oluşturulur. Bu çözüm uzayında minimum sezgisel maliyete sahip nokta lokal hedef olarak belirlenir. Sezgisel maliyet fonksiyonu Denklem (2.12)'de verilmiştir.

$$f(p_j)=\left\|p_j, p_r\right\|+D(p_j)+k\Delta\theta_j \tag{2.12}$$

$p_j$, $p_r$, $D(p_j)$, k, $\Delta\theta_j$ parametreleri sırasıyla açıklık boyunca çizilen doğru üzerinde bulunan j indeksli noktanın konum vektörü, mobil robot konum vektörü, ilgili noktanın maliyet değeri, açısal değişim katsayısı ve mobil robotu j indeksli noktaya yönlendirme açısı olarak tanımlamıştır. Maliyet fonksiyonu, yüksek hücre maliyeti

değerini, mobil robot ile ilgili nokta arasındaki Öklid uzaklığını, hareket edilecek noktalar için kat edilecek rotasyonu negatif olarak değerlendirmektedir. Şekil 2.26'da lokal hedef tespitinde izlenen adımlar görselleştirilmiştir. "O" ile gösterilen engeller arasında tespit edilen açıklıklar ve bu açıklıklar üzerinde tanımlı noktalar olası lokal hedef çözümleridir. Bu çözüm uzayı üzerinde verilen maliyet fonksiyonu uygulanarak lokal hedef tespiti gerçekleştirilir. Tangent-bug ve benzer diğer lokal yaklaşımlarda mobil robot doğrudan engellerin köşe noktalarına veya açıklıkların geometrik orta noktalarına yönlendirilir. Tez kapsamında önerilen metot ise harita hakkında ön bilgi varsa bu bilgiyi maliyet fonksiyonuna dâhil eder. Açısal manevra parametresi de benzer metotların kriter olarak değerlendirmediği bir uygulamadır. Lokal hedef tespitinde doğrudan Öklid mesafesi temel alınmadığı için planlanan yolun geometrisi mobil robot için daha doğal bir navigasyon hareketi sağlamaktadır.

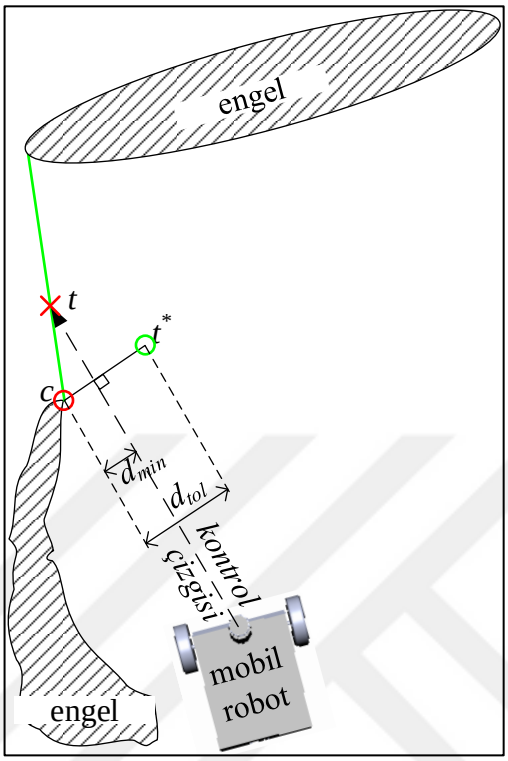



Şekil 2.26. Optimal lokal hedef tespiti

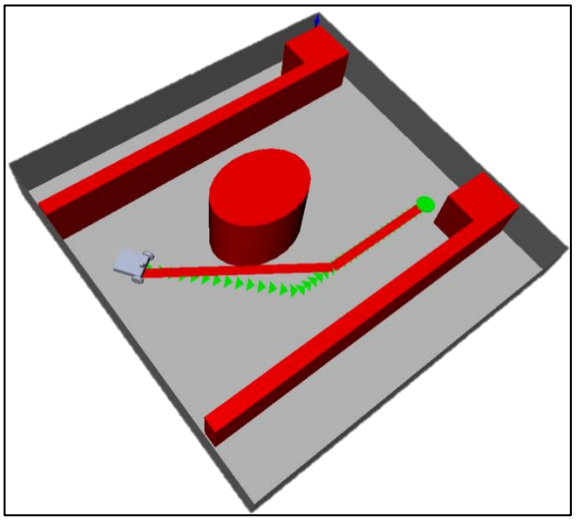
### 2.2.3. Çarpışma engelleme metodu

Lokal hedef tespiti sonucu elde edilen noktaya doğrudan erişmek her durumda mümkün olmamaktadır. Hesaplanan noktaya yönelim sırasında robotun fiziksel dönüş limitleri ve boyutlarına bağlı olarak istenmeyen çarpışmalar ortaya çıkabilmektedir. Bu tip durumları engellemek için lokal hedef tespiti algoritması tarafından hesaplanan nokta, kontrol birimine iletilmeden önce çarpışma kontrolünden geçirilmektedir. Çarpışma kontrolü sonrasında, belirlenen koşullar oluşursa lokal hedef güncellenerek kontrol modülüne gönderilir. Çarpışma kontrolü ve lokal hedef güncelleme işlemleri Şekil 2.27'de şema olarak gösterilmiştir. Engeller arasında tespit edilen açıklık ve açıklık üzerinde hesaplanan optimal lokal hedef sırasıyla yeşil renkle kodlanmış doğru ve "t" sembolü ile gösterilmiştir.

Çarpışma kontrolü için robot merkezinden lokal hedefe bir kontrol çizgisi tanımlanmaktadır.



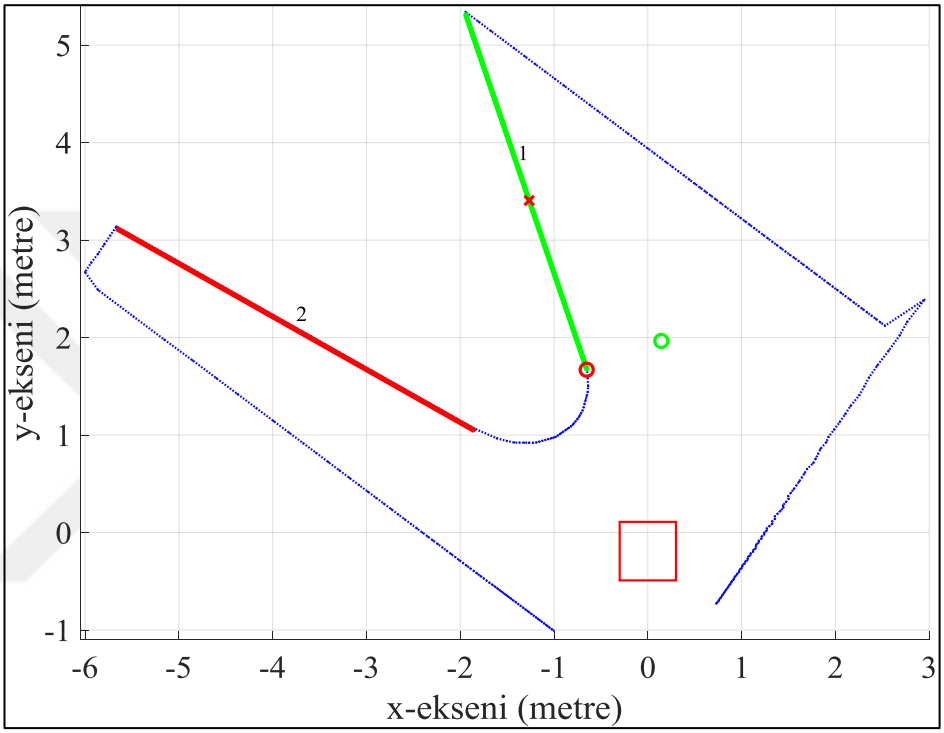

Şekil 2.27. Çarpışma kontrolü



Şekil 2.28. Çarpışma kontrolü örnek senaryo engel profili

LIDAR tarafından engel olarak tanımlanan noktalardan lokal hedefe, mobil robottan daha yakın olan noktalar içerisinden, kontrol çizgisine en yakın nokta tespit edilir. Bu nokta "c" ile gösterilmiştir ve kontrol çizgisine olan dik uzaklığı ($d_{min}$) belli bir

kritik eşik uzunluktan ($d_{tol}$) küçükse bu nokta çarpışma noktası olarak etiketlenmiştir. Lokal hedef olarak belirlenmiş t noktası bu aşamada güncellenmektedir. c noktasından, $d_{tol}$ kadar uzakta ve kontrol çizgisine dik olan bir konumda yeni bir nokta tanımlanır ($t^*$). Mobil robot t noktası yerine t* noktasına yönlendirilerek olası bir çarpışma engellenir. Bu yöntem robotun olası çarpışma noktası etrafında kavisli bir yol izleyerek lokal hedefe ilerlemesini sağlamaktadır.




Şekil 2.29. Çarpışma kontrolü örnek senaryo LIDAR görüntüsü

Çarpışma engelleme metodunun yol planlama algoritmasının akışı içerisinde gerçekleştirdiği çarpışma noktası tespiti ve lokal hedef güncelleme işlemleri örnek bir senaryo altında gösterilmiştir. Tez kapsamında gerçekleştirilen 3 boyutlu mobil robot benzetim arayüzü üzerinde kurgulanan iç mekân koşulları Şekil 2.28'de verilmiştir. Benzetim arayüzü ile ilgili detaylı bilgi ilgili başlık altında verilmiştir. Verilen anlık mobil robot konumu için LIDAR uzayında belirlenen lokal hedef ve güvenli lokal hedef noktaları Şekil 2.29'da verilmiştir. Verilen durum için LIDAR arayüzünde mobil robot kırmızı renkli dikdörtgen olarak gösterilmiştir. Tespit edilen iki açıklık üzerinde tespit edilen optimal lokal hedef kırmızı renkli "x" sembolü ile gösterilmiş, 2 numaralı açıklık elenerek robot 1 numaralı açıklık üzerinden geçmek için kontrol edilmiştir. Çarpışma engelleme metodu tarafından kırmızı renkli "o"

sembolü ile gösterilen çarpışma noktası tespit edilmiş, bu sebeple lokal hedef güncellenerek, robot yeşil renkli "o" sembolü ile gösterilen noktaya yönlendirilmiştir.

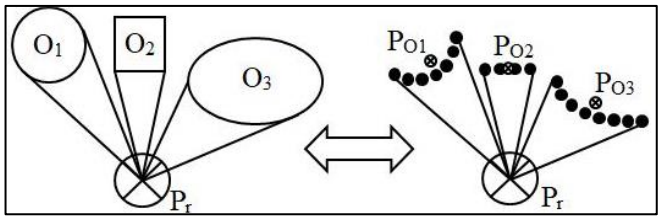
Bu senaryo için açıklık tespiti aşamasında herhangi bir revizyona ihtiyaç duyulmamış, açıklıklar yeniden düzenlenmemiştir.

### 2.2.4. Hareketli engellerin tespiti

Mobil robot uygulamalarında hareketli engellerin bulunduğu ortamlarda çalışmak, statik engellerle çalışmaktan daha karmaşık bir durumdur. Engellerin sabit olması durumunda yalnızca robotun bir sonraki hareketi planlanırken engellerin hareketli olması öngörülebilirliği olumsuz yönde etkilemektedir. Bu tip ortamlarda engelin boyutu, hareket yönü ve hızı da hesaba katılarak robotun kontrol edilmesi istenmektedir. Tez kapsamında geliştirilen dinamik yol planlama yöntemi üzerine geliştirilen temel hareketli obje tespit metodu, Denklem (2.1) ve Denklem (2.2)'de tanımlanan LIDAR modeline uygun verilerin gerçek zamanlı işlenmesi temeline dayanır. Bu model üzerinde tarama açısı $180^o$, tarama çözünürlüğü $1^o$ ve görüş limitleri 3 metre olarak belirlenmiştir.

$$D_L = \left\{ d@1^o,\ d@2^o,\ d@3^o,\ ..\ d@90^o,\ \ldots,\ d@180^o \right\} \tag{2.13}$$

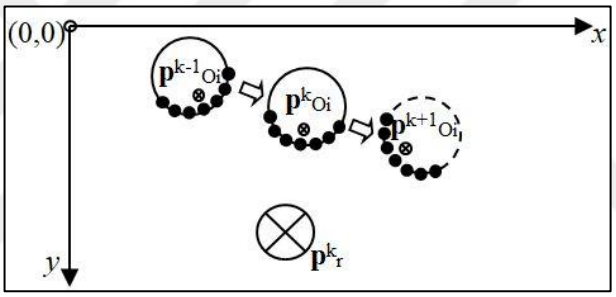
Denklem (2.13)'te verilen ölçüm modelinde, $d@x^o$ ifadesi "x derece açısında d milimetre uzakta bulunan engel noktası" olarak tanımlanmıştır.




Şekil 2.30. Engellerin LIDAR eksenlerinde modellenmesi

Mesafe dizisindeki uzaklık ölçümleri sensör merkezini referans almaktadır. Ardışık ölçümlerden belirli bir eşik değerinden fazla değişim gösteren indis iki engel ya da bir engel ve bir açıklık arasındaki sınır olarak değerlendirilmiştir. Bu varsayımla belli

bir eşik altında değişim gösteren uzaklık bilgisine sahip ölçüm alt dizileri engel olarak değerlendirilmiştir. Engel algılama ve açıklık tespiti ile ilgili teorik altyapı 2.2.1 numaralı başlık altında verilmiştir. Anlık engel profilinin LIDAR eksenlerinde modellenmesi Şekil 2.30'da verilmiştir. $O_{1,2,3}$ ile gösterilen engeller $P_r$ konumunda bulunan LIDAR tarafından belli bir çözünürlükle tarandığında $P_{O1,2,3}$ ile gösterilen ağırlık merkezlerine sahip nokta kümeleri olarak elde edilmektedir. Bu nokta kümelerinin sayısı $D_L$ dizisindeki kritik değişimler işlenerek hesaplanır. Her bir engelin başlangıç ve bitiş ölçümünde bir kritik değişim oluştuğu kabulü ile engel sayısı bu sayının yarısı olarak kabul edilir. Sonraki aşamada bu nokta bulutu 2 boyutlu sınıflandırma yapılarak kümelenir. Her küme kendi ağırlık merkezi ile ifade edilmektedir. Verilen model üzerinde örneklenirse, $P_{O1}$ noktası $O_1$ engelinin konumunu göstermektedir.




Şekil 2.31. Hareket kestirimi şeması

Hareketli engeller sensörün görüş alanına girdiğinde robota arkadan ya da yandan çarpmayacağı varsayılmaktadır. Homojen geometrilerde oldukları ve hızlarının da öngörülebilir limitlerde olacağı varsayımı vardır. Görüş alanına giren engeller için anlık tespit edilmiş olan ağırlık merkezi ve küme indeksleri kaydedilerek bir sonraki ağırlık merkezleri ile karşılaştırılır ve en yakın çiftler eşleştirilir. Bu ikililer herhangi bir engelin bir önceki ve şu andaki konumlarına karşılık gelmektedir. Tespit edilen engellerin bir sonraki periyottaki konumları, engellerin 2 boyutlu uzaydaki gradyanından yola çıkılarak hesaplanır.

Şekil 2.31'de verilen şemada $p^k_r$ sensör merkezinin k anındaki konum vektörünü, $p^k_{Oi}$ indeksi i olan engelin k anındaki konum vektörünü, $p^{k-1}_{Oi}$ indeksi i olan engelin k-1 anındaki konum vektörünü, $p^{k+1}_{Oi}$ ise indeksi i olan engelin k+1 anındaki konum tahminini temsil etmektedir. Lidar kritik görüş menziline ($d_c$) giren engellerin ifadesi Denklem (2.14) ve Denklem (2.15)'de verilmiştir.

$$\left|p_r - p_{Oj}\right| - d_c \geq 0 = q(p_r),\ j=1,2,3,\ldots,\ \#engel \tag{2.14}$$

$$P_o^* = \left\{\forall p_O | q(p_r) \geq 0\right\} = \left\{p_{O1}, p_{O2}, p_{O3}, \ldots\right\} \tag{2.15}$$

$P_o^*$ ile ifade edilen küme kritik görüş alanına giren engellerin konum vektörlerini kapsamaktadır.

**For** a=1 to n
 **For** b=1 to m
  $\{n^*, m^*\} = \{(n,m) | \min(|p_a^{k-1} - p_b^k|)\}$
 **End**
**End**

Şekil 2.32. Ortak engellerin eşleştirilmesi

$$P_o^{(k-1)^*} = \left\{p_i^{k-1}, p_{i+1}^{k-1}, p_{i+2}^{k-1}, \ldots, p_m^{k-1}\right\} \tag{2.16}$$

$$P_o^{k^*} = \left\{p_j^k, p_{j+1}^k, p_{j+2}^k, \ldots, p_n^k\right\} \tag{2.17}$$

$$\vec{\nabla}_{p_{oi}} = \frac{\partial\left(p_{oi_x}^k - p_{oi_x}^{k-1}\right)}{\partial_x} d_x + \frac{\partial\left(p_{oi_y}^k - p_{oi_y}^{k-1}\right)}{\partial_y} d_y, \quad \forall p_{oi} \in P_o^* \tag{2.18}$$

$$\hat{p}_{oi}^{k+1} = p_{oi}^k + \vec{\nabla}_{p_{oi}} \tag{2.19}$$

Ardışık zamanlarda hesaplanan konum vektörlerinin aynı indeksli engele ait olup olmadığının kontrolü ve aynı olanların eşleştirilmesi Denklem (2.16) ve Denklem (2.17)'de verilmiştir. Aynı olan engellerin eşleştirilme prosedürü pseudo kod olarak Şekil 3.32'de gösterilmiştir.

Engellerin hareket yönüne bağlı gradyanları ve konum tahmini ifadeleri Denklem (2.18) ve Denklem (2.19)'de verilmiştir.

$p_{oi}$ ifadesi indeksi i olan engelin sensör ölçüm kümesinin ağırlık merkezi vektörüdür. Objelerin 2 boyutlu koordinat sisteminde hareket eğilimleri tespit edildikten sonra engel algılama ve lokal hedef belirleme algoritmasına giriş parametresi olarak

engellerin anlık konum vektörleri değil, kestirilen konum vektörleri verilir. Herhangi bir hareketli engelin sensörün görüş alanı içerisinde robotun hareket doğrultusu ile kesişeceği süre hesaplanır ve bu süre bir zaman periyodu veya daha kısa bir süreye karşılık geliyorsa önerilen hareketli engel tespit yöntemi navigasyon algoritmasına müdahale eder. Önerilen yöntemle ilgili bulgular Sonuçlar ve Öneriler başlığı altında verilmiştir.

## 2.3. DWA Lokal Yol Planlayıcısı

Dinamik pencere yaklaşımı (Dynamic Window Approach-DWA) mobil robotun lokal hedefe ulaşabilmesi için optimal açısal ve doğrusal hız bileşenlerinin hesaplanmasını temel alan bir lokal yol planlayıcıdır. Hız bileşenleri güvenli olarak belirlenen bir tarama uzayından seçilir. Kartezyen uzayda bilinen hedefi hız uzayına dönüştüren bir yaklaşımdır. Temel olarak iki aşamada gerçekleştirilen metot, ilk olarak hız tarama uzayının hesaplanması ile başlatılır. Robotun çarpmadan durabileceği tüm doğrusal ve açısal hız değerleri belli bir çözünürlükte hesaplanarak tarama uzayına dâhil edilir. İkinci aşamada her bir hız bileşeni için maliyet fonksiyonunun değeri hesaplanarak optimal açısal ve doğrusal hız bileşenleri belirlenir. Maliyet fonksiyonu robotun hızını, en yakın engelle olan uzaklığını ve hedef oryantasyonuna olan benzerliğini maksimum yapan durumu optimal olarak belirlemek için kullanılmaktadır. Mobil robot ile eş merkezli olarak tanımlanan lokal pencere, global harita üzerinde robotun sonraki adımlarda bulunduğu konuma bağlı olarak yer değiştirmektedir. Yöntemle ilgili akışı gösteren Pseudo kod Şekil 2.33'te verilmiştir. Verilen Pseudo kod üzerinde yapılan işlemlerde öncelikle robotun anlık konumu ve hedef konumu arasında birim zaman periyodunda sahip olunması istenilen hız büyüklüğü hesaplanmaktadır. Bu hız değeri, robotun maksimum ivmesi ve zaman değişkenine bağlı olarak erişilmez bir büyüklük olabilir ancak maliyet fonksiyonunda fiziksel limitler dâhilinde olan hız bileşenlerinin hedeflenen hıza olan yakınlığı parametre olarak bulunduğu için kabul edilebilir en yüksek hız hedeflenmektedir. Geçerli hız bileşenlerinin her biri için en yakın engel konumları hesaplanmaktadır. Robot ilgili hız komutu ile kontrol edildiğinde olası bir çarpışmanın kontrolü sağlanmış olur. Bu mesafe robot için tanımlı durma mesafesinden yüksekse bu hız bileşeni ihmal edilerek çözüm uzayına dâhil edilmez. Bu kriteri sağlayan hız bileşenleri için Denklem (2.20)'de verilen formatta bir

maliyet fonksiyonu uygulanarak geçerli işlem adımı için optimal kontrol girişi elde edilmektedir.

```
Import: robotKonum, robotModel, robotHedef
hedefHız = hızHesapla(robotKonum, robotHedef)
lazerData = lidarOku()
lineerHızAralığı = pencereOluştur(robotLineerHız, robotModel)
açısalHızAralığı = pencereOluştur(robotAçısalHız, robotModel)
For each v in lineerHızAralığı
  For each w in açısalHızAralığı
    d = mesafeBul(v, w, lazerData, robotModel)
    dFren = frenlemeMesafeBul(v)
    If d>dFren
      açısalPoz = açısalPozHesapla(robotKonum, robotHedef, v, w)
      dKritikNorm = (d-dFren)/(dmaks-dFren)
      maliyet = maliyetFonksiyonu(açısalPoz, dKritikNorm, |hedefHız – v|)
      If maliyet > optimal
        vOptimal = v
        wOptimal = w
        optimal = maliyet
robotKontrol(vOptimal, wOptimal)
```

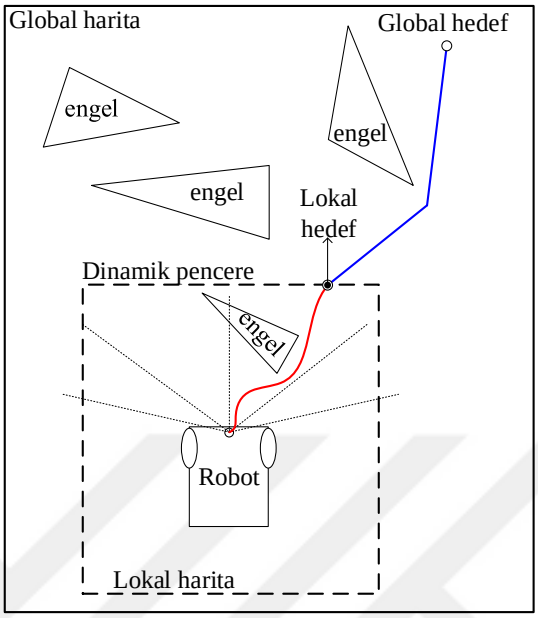
Şekil 2.33. DWA lokal planlayıcı pseudo kodu

Verilen maliyet fonksiyonunda α hedeflenen duruş açısı ile güncel hız bileşenlerinin sağlayacağı açısal duruş arasındaki benzerlik terimidir. $d_{norm}$ parametresi lokal pencerenin köşegen uzunluğuna göre normalize edilmiş kritik engel uzaklığıdır.

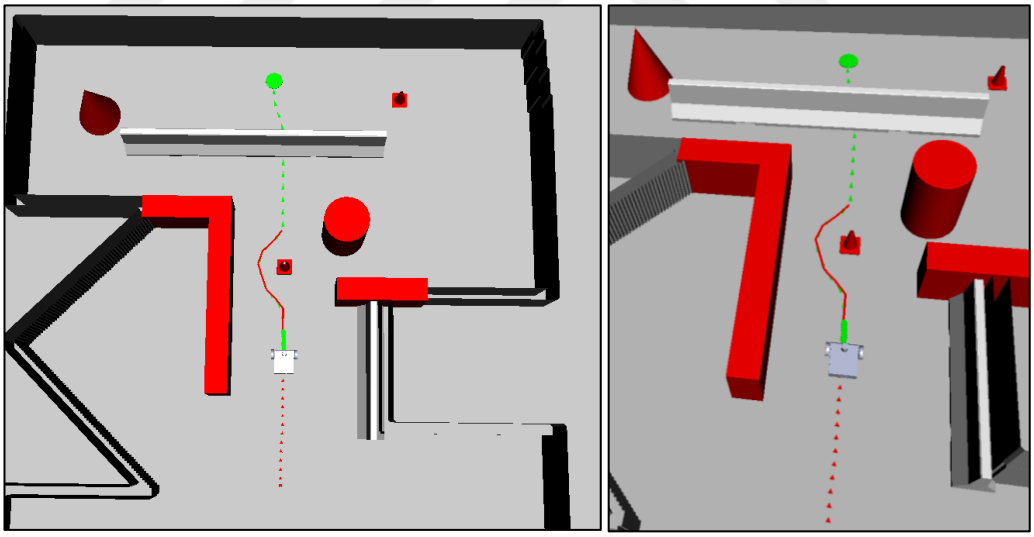
$$f_{cost} = k \times \alpha + n \times d_{norm} + m \times \Delta_v \tag{2.20}$$

$\Delta_v$ ise hedeflenen hız ile güncel durumda çözümlenen hız arasındaki mutlak farktır. Bu parametreleri ağırlıklandırmak için sırasıyla k, n, m katsayıları verilmiştir. Bu fonksiyonu maksimum yapan açısal ve doğrusal hız büyüklükleri optimal olarak kabul edilmektedir. Dinamik pencere yaklaşımı statik bir yörünge verilmesi durumunda lokal ve global karakteristiğe sahip olabilmektedir [117]. Lokal

bölgedeki engelleri tarayabilen sensörlerle donatılan robot, yerel pencere içerisinde planlanan optimal yörünge üzerinde hareket ettirilir.




Şekil 2.34. DWA yol planlayıcı şeması

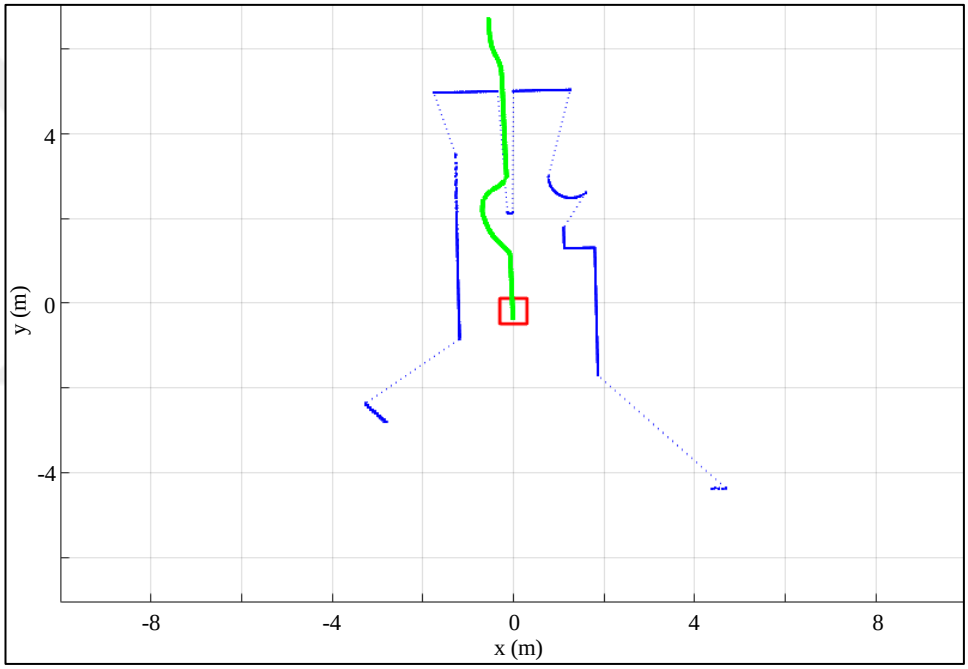


Şekil 2.35. DWA ile planlanmış örnek bir yol ve anlık VR görüntüleri

Lokal yol planlanırken lokal maliyet haritası üzerinde global yol planlama prosedürü koşturulabilir. Lokal yol üzerinde hareket eden mobil robot, dinamik bir engelle karşılaşması veya mevcut engelin ortadan kalkması durumunda güncellenen lokal yolu izlemeye devam etmektedir. Lokal pencere sınırlarında belirlenen lokal hedef ile robotun anlık konumu arasında planlanan yol ve lokal hedef ile global hedef arasında planlanan yol toplam yörüngeyi oluşturmaktadır. Bu varsayımlar altında

tasarlanan DWA yol planlayıcısına ait şematik bir tanım Şekil 2.34'te gösterilmektedir.

Lokal harita üzerinde yol planlama işlemi literatürde yer alan farklı yaklaşımlarla gerçekleştirilebilmektedir. İşlem süreleri gözetildiğinde hücre ayıklama tabanlı yaklaşımlar ve bu dönüşümlere uygulanan lokal tarama çözümleri yaygın olarak kullanılmaktadır. Tez kapsamında önerilen açısal kontrol metodu ise Dijkstra algoritması ile maliyetlendirilmiş ve gradient descent metodu uygulanarak lokal yol planlanmış DWA uygulaması için geliştirilmiştir. Dijkstra ve gradient descent algoritmaları hakkında detaylı bilgi statik yol planlama başlığında verilmiştir.

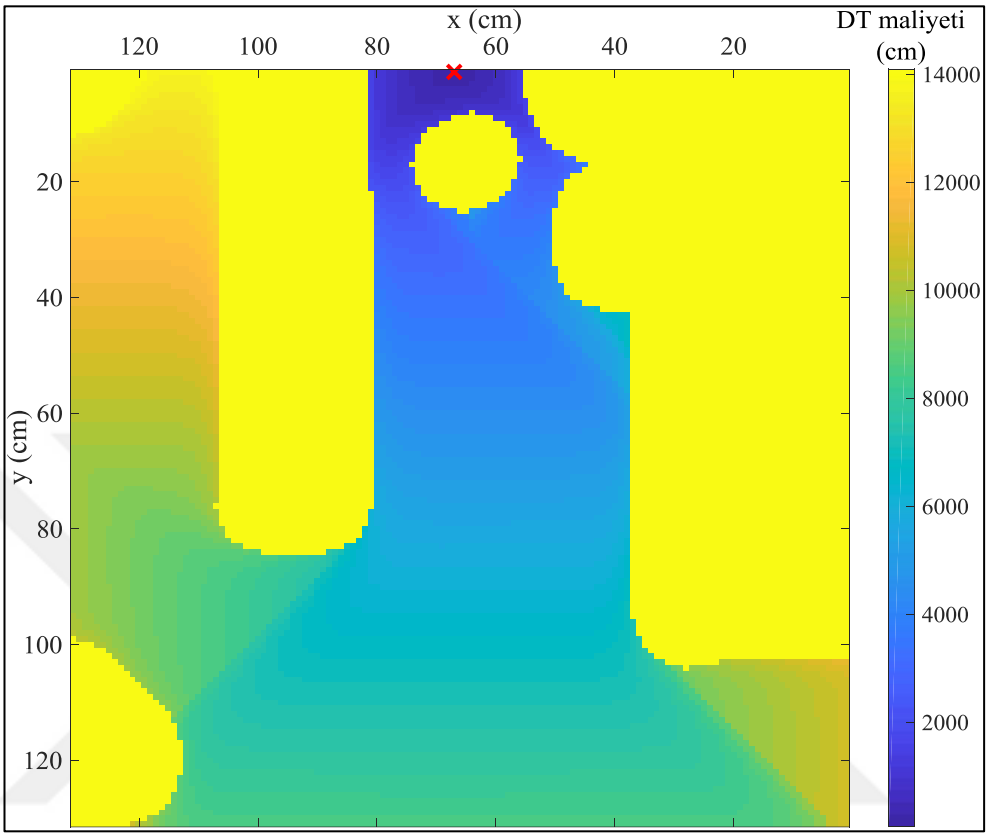



Şekil 2.36. Lokal bölgedeki LIDAR görüntüsü ve lokal yol

Tez kapsamında geliştirilen benzetim arayüzünde DWA algoritmasının uygulamasına ait ekran alıntıları Şekil 2.35'te verilmiştir. Benzetim arayüzü ile ilgili detaylı bilgi ve kullanılan notasyonlar ilgili başlık altında verilmiştir. Üç boyutlu şekiller aynı durumun farklı açılardan görüntülerini içermektedir.

Tarama açısı $270^{o}$ olan anlık LIDAR ölçümünün görselleştirildiği grafik Şekil 2.36'da verilmiştir. Bu grafik üzerinde engel ölçümleri mavi noktalarla gösterilirken, planlanan lokal yol yeşil renkli yörünge biçiminde görselleştirilmiştir. Robotu

merkezine alan LIDAR arayüzünde mobil robot kırmızı dikdörtgen olarak gösterilmiştir.




Şekil 2.37. Lokal bölgedeki maliyet haritası



Şekil 2.38. Lokal bölgedeki engel haritası

Robot, benzetim arayüzünde yeşil daire ile gösterilen hedefe ilerlerken anlık olarak lokal pencerede oluşan maliyet haritası Şekil 2.37'de verilmiştir. Maliyet haritası, renk skalası üzerinde Dijkstra algoritması ile elde edilen Öklid maliyetlerinin değerlerine karşılık gelecek biçimde gösterilmiştir. Optimal lokal hedef noktası da bu durumda en düşük maliyete sahip nokta olarak belirlenmiştir. Harita üzerinde bu

nokta “x” sembolü ile gösterilmiştir. Şekil 2.35’te verilen anlık durum için engeller, belli bir güvenlik katsayısı ölçüsünde genişletilerek lokal maliyet haritasına dahil edilmiştir. Lokal bölgedeki engel haritası ise ikili sistemde (binary) tanımlamıştır. Engellerin LIDAR koordinatlarındaki konumları baz alınarak engellenen bölgeler lojik 1, serbest bölgeler lojik 0 olarak tanımlanmıştır. Verilen anlık durum için lokal engel haritası Şekil 2.38’de verilmiştir. Koyu pikseller serbest, beyaz pikseller engellenmiş bölgeleri göstermektedir.

Standart DWA yöntemi mobil robotun hedef konumu için tam olarak açısal bir kontrol uygulamamaktadır. Lokal pencere içerisinde global hedef tespit edilip bu noktaya belli bir tolerans bandında yaklaşılması durumunda hedefe ulaşıldığı kabul edilmektedir. Tez çalışması kapsamında sanal garaj (virtual garage) isimli bir yöntem ortaya koyularak DWA metodunda bir iyileştirme önerilmiştir. Önerilen yöntemin çalışma prensibi hedef konumunun çevresine robotun boyutlarına bağlı olarak boyutlandırılmış dikdörtgen geometrili, üç kenarı kapalı bir engel yerleştirilmesi temeline dayanır. Bu engel gerçekte olmayan sanal bir engeldir ve duruş açısı, hedef konumunda robotun sahip olması istenilen açıya bağlı olarak belirlenmektedir.

Navigasyon sırasında lokal hedef, global hedefe belli bir eşik değerinden daha yakın bir duruma geldiğinde robot yörünge takip moduna alınır ve takip edilen yol engellense dahi alternatif yol planlama işlemi yapılmayarak engelin yoldan çekilmesi beklenir. Bu aşamada lokal hedef ($h_l$) ile global hedef ($h_g$) arasındaki Öklid uzaklığı ile orantılı olarak bir yörünge takip değeri ($k_t$) tanımlanmıştır. Bu değer robotun planlanan yol üzerinde hangi uzaklıktaki noktayı lokal hedef olarak belirleyeceğine etki eder. Değer sıfıra yakınsadıkça robot planlanan yolu daha agresif bir şekilde takip etme eğilimi göstermektedir. Aksine, bu değer arttıkça robot yolun daha uzak indekslerindeki noktaları hedefleme eğilimi göstermektedir. Uygulamaların yapıldığı mobil robot platformu için tanımlı mesafeler ve tanımlı katsayılar Tablo 2.9’da verilmiştir.

Tablo 2.9. DWA açısal kontrol yol takip parametresi

| $\|g_h l_h\|$ (m) | $k_t$ (m) |
|---|---|
| $\|g_h l_h\| > 4$ | 2 |
| $2 < \|g_h l_h\| \leq 4$ | 1 |
| $1 < \|g_h l_h\| \leq 2$ | 0,75 |
| $\|g_h l_h\| \leq 1$ | 0,3 |

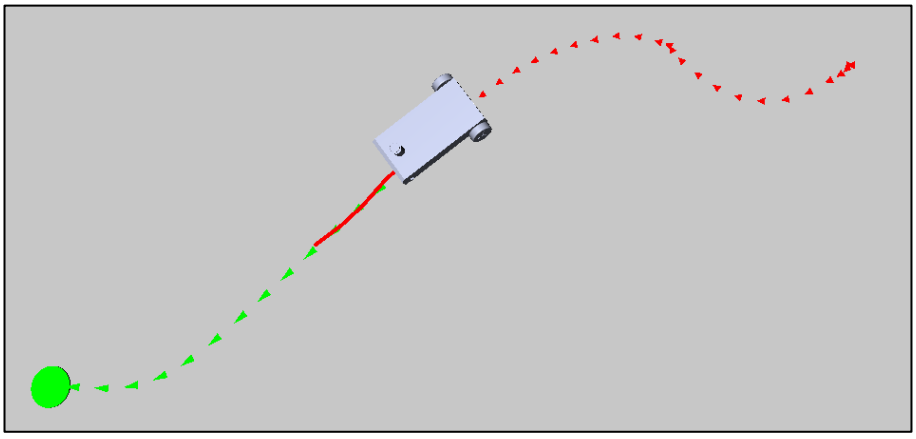
Planlanan ve takip modunda izlenen yolun mobil robot konumuna en yakın olan noktası $p_c$ ile gösterilmek üzere, mobil robotun takip ettiği yol üzerinde anlık hedef olarak belirlenen nokta $p_g$ Denklem (2.21)'de verilen işlem ile hesaplanmaktadır.

$$p_g = p_c + k_t \tag{2.21}$$

Global hedef ile lokal hedef arasındaki uzaklığa bağlı olarak robotun yol takip modunda izlediği adımları gösteren benzetim uygulamasının başlangıç durumu Şekil 2.39'da verilmiştir.



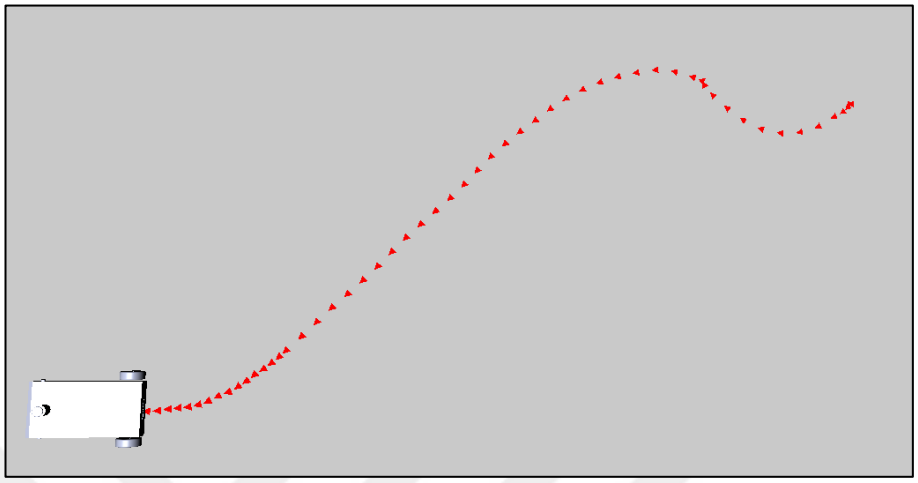
Şekil 2.39. DWA açısal kontrol: başlangıç konumu



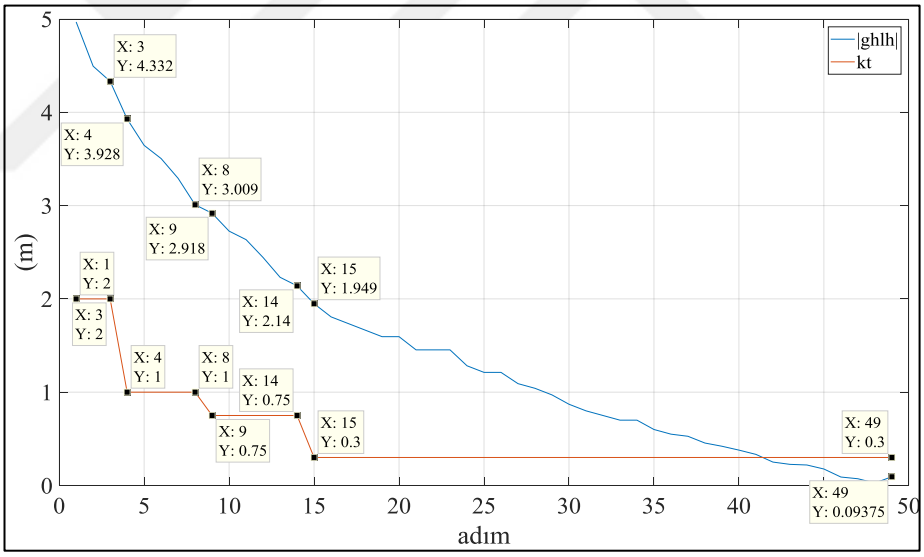
Şekil 2.40. DWA açısal kontrol: anlık konum

Robotun başlangıç oryantasyonu $270^o$, hedef oryantasyonu ise $180^o$ olarak belirlenmiştir. Hareket başladıktan sonra rasgele bir anlık durum için robotun konumu, planlanan ve izlenen yol durumları Şekil 2.40'ta gösterilmiştir. Yörüngeyi tamamlayan robotun duruş oryantasyonu Şekil 2.41'de verilmiştir. Açısal duruş

hatası 1,32° olarak belirlenmiştir. Robot takip moduna girdikten sonra $|g_h l_h|$ uzaklığının değişimine bağlı olarak oluşturulan $k_t$ parametrelerinin değerleri Şekil 2.42'de görülmektedir. Bu değerler Tablo 2.9'da verilen kriterleri sağlamaktadır.



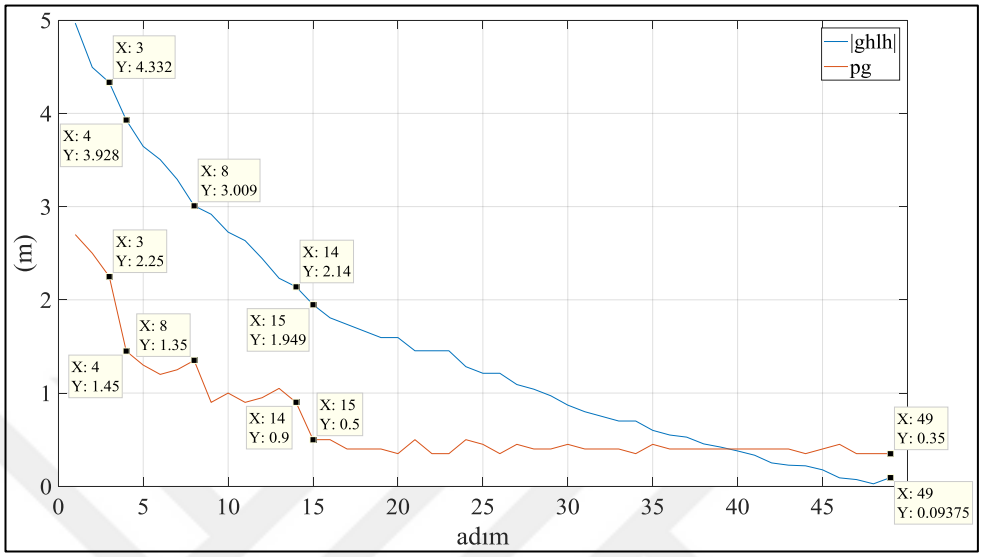
Şekil 2.41. DWA açısal kontrol: hedef konumu



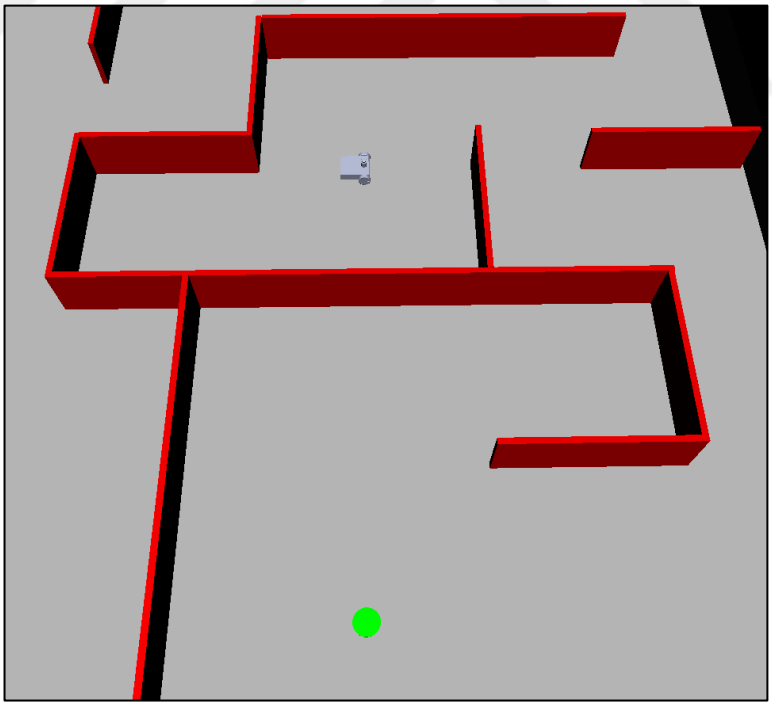

Şekil 2.42. DWA açısal kontrol: $|g_h l_h|$-$k_t$ grafiği

Her işlem adımında lokal hedefin global hedefe yakınsadığı ve buna bağlı olarak robotun takip ettiği yol üzerinde, robot konum merkezine daha yakın bir noktayı lokal hedef olarak belirlediği görülmektedir. Lokal hedefin metrik uzaklığının $|g_h l_h|$ uzaklığına bağlı olarak $p_g$ parametresini değişimi Şekil 2.43'te verilmiştir. Azalan $k_t$ değerine bağlı olarak takip dilen yörünge üzerinde bulunana lokal hedef ile mobil robot arası mesafe de azalarak 35 cm seviyesine kadar düşmüştür. Bu mesafe yol parçasının uzunluğu olup robot ile ilgili nokta arasındaki Öklid uzaklığı değildir.

Robot global hedefe belli bir eşik değerinden daha yakın olduğu anda ise hedefe ulaşıldığı kabul edilmiştir. Robot boyutlarına bağlı olan bu değer, uygulama yapılan robotlar için 5 cm ve 10 cm arasında değişmektedir.



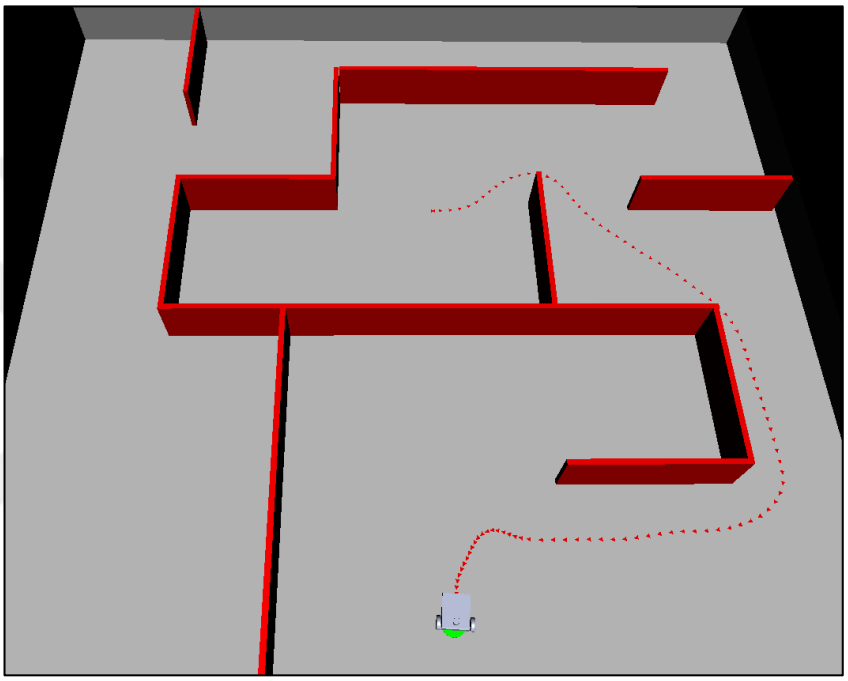

Şekil 2.43. DWA açısal kontrol: $|g_h l_h|$-$p_g$ grafiği



Şekil 2.44. DWA açısal kontrol benzetim uygulaması

Açısal kontrol işleminin temel bileşeni olan sanal garaj eklentisi mobil robotun hedef oryantasyonuna istenilen açıda konumlandırılabilmesi için global haritaya statik bir engel olarak eklenmiştir. Sanal garaj kafes biçiminde ve robotun belli bir toleransta

sığabileceği boyutlarda tanımlanmıştır. Robot yörünge takip moduna alındığında engelden sakınma prosedürü sanal garajdan kaynaklı istenmeyen müdahalelerde bulunabilmektedir. Global haritada tanımlı sanal garaj görüntüsünden kaynaklı güvenlik hareketlerini engellemek için, yörünge takip modunda sanal garaj global haritadan kaldırılır. Bu aşamada robotun alternatif yol planlama opsiyonları engellenir ve kontrollü bir yörünge üzerinde hedef konumuna istenilen açıda yerleşmesi sağlanmaktadır. Örnek bir benzetim uygulaması Şekil 2.44'te verilmiştir.

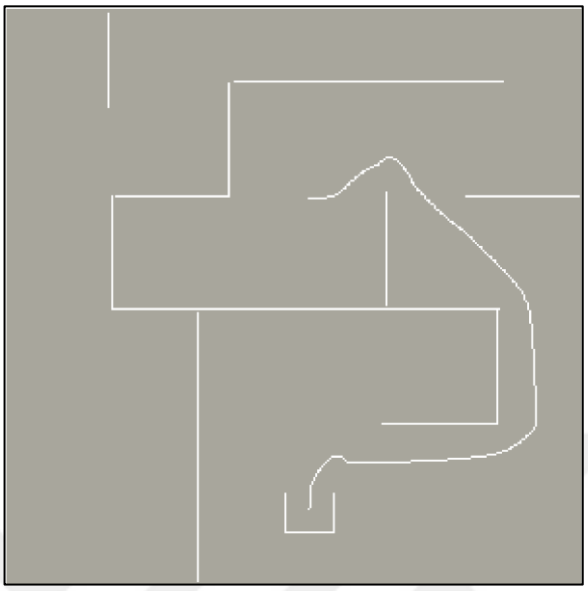


Şekil 2.45. DWA benzetimi: açısal kontrol aktif

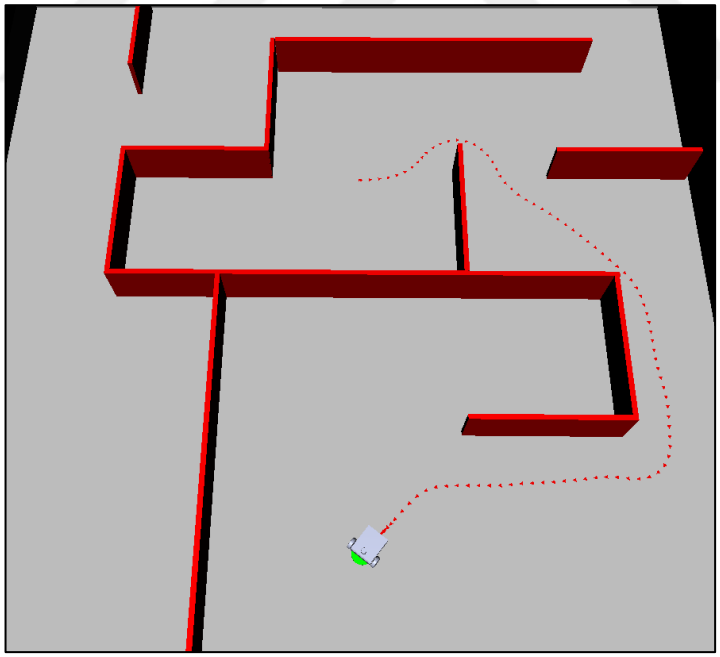
Başlangıç konumunda gösterilen robot için verilen hedefin konumu daire olarak gösterilmiştir. Duvar biçimindeki engeller 3 boyutlu olarak gösterilmiştir. Robotun başlangıç oryantasyon açısı $0^o$, hedef konumundaki istenilen açısı ise $270^o$ olarak tanımlanmıştır. Açısal kontrol uygulanması durumunda elde edilen yörünge ve mobil robot hedef oryantasyonu Şekil 2.45'te verilmiştir. Hedef konumundaki açısal sapma $2.4^o$ olarak belirlenmiştir. Sanal garajın global engel haritasında görüntüsü Şekil 2.46'da verilmiştir.

Siyah (lojik 0) pikseller serbest alanları, beyaz (lojik 1) pikseller engellenmiş bölgeleri göstermektedir. Açısal kontrol uygulamasında sanal garaj eklentisinin etkisi 2 boyutlu statik harita üzerinde görülebilmektedir. Benzetim ortamında yapılan

uygulamada açısal kontrol uygulanmaması durumunda robot verilen hedefe belli bir tolerans ölçüsünde yaklaştığı durumda hedefe ulaşıldığı kabul edilmiştir.



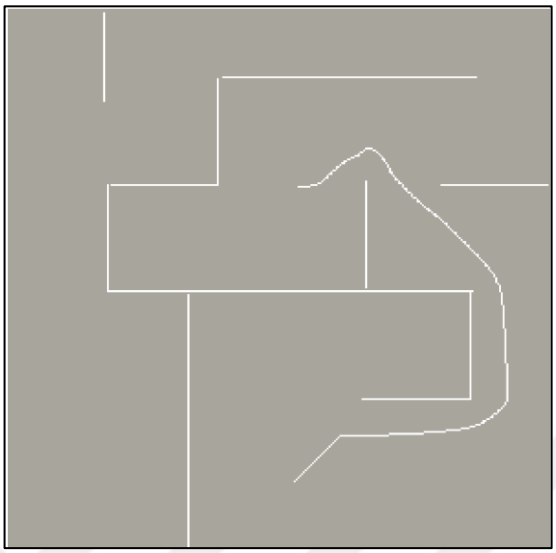
Şekil 2.46. Açısal kontrol aktif iken statik harita



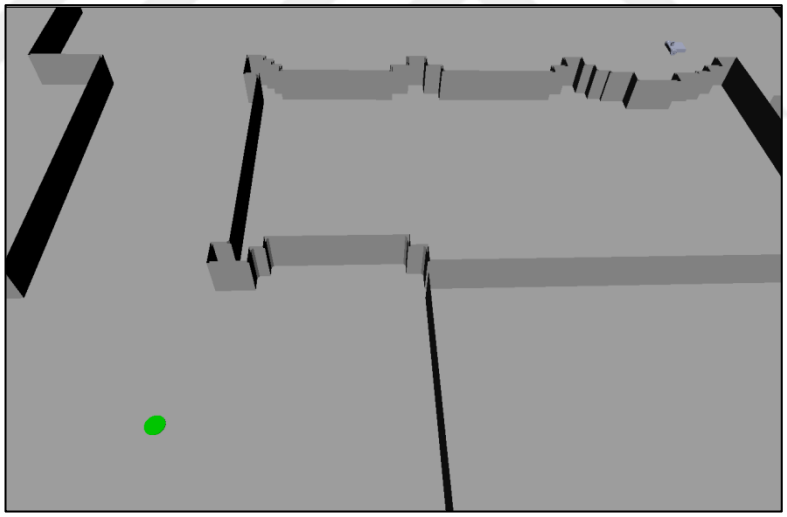
Şekil 2.47. DWA benzetimi: açısal kontrol pasif

Hedefe yaklaşım açısı robotun anlık duruş açısına bağlı olarak rasgele bir değer olmaktadır. Bu durumun 3 boyutlu arayüz üzerinde gösterimi Şekil 2.47'de verilmiştir. Statik planlayıcı tarafından planlanan yol ve 2 boyutlu global harita Şekil 2.48'de verilmiştir. Sanal garaj uygulanmayan bu durum sadece global hedefe olan

Öklid uzaklığını baz alan standart DWA yönteminin yol planlama çıktısını göstermektedir.



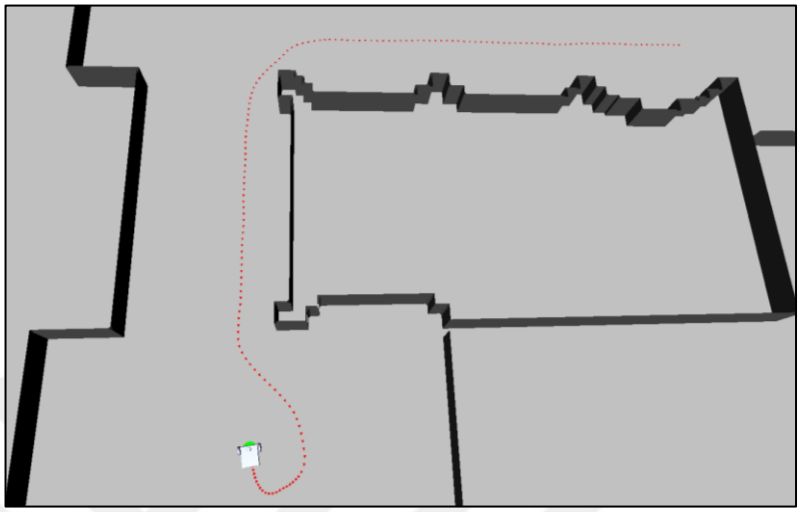
Şekil 2.48. Açısal kontrol pasif iken oluşan statik harita



Şekil 2.49. Açısal kontrol senaryosu başlangıç durumu

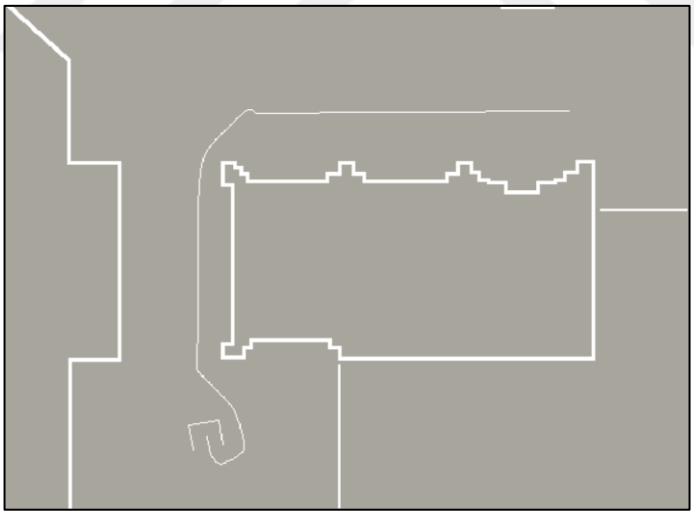
Açısal kontrol yöntemi ile ilgili bir başka senaryo Şekil 2.49'da verilen benzetim arayüzünde gösterilmiştir. Başlangıç duruş açısı $180^{o}$ olan robot için navigasyon süreci başlatılmıştır. Hedef konumunda sahip olması istenen açı $100^{o}$ olarak belirlenmiştir. Statik yol planlama algoritması sanal garaj yapısındaki engeli hedef koordinatlarına verilen açıda yerleştirerek statik yol planlayıcı hedef açıyı sağlamaya zorlanmıştır. Navigasyon işleminin bitişi ile elde edilen 3 boyutlu görünüm ve 2 boyutlu global engel haritası sırasıyla Şekil 2.50 ve Şekil 2.51'de verilmiştir. Hedef

konumundaki açısal hata 1,53$^{o}$ olarak belirlenmiştir. Benzetim arayüzünde 3 boyutlu olarak gösterilen haritada hedefe yaklaşan mobil robotun sanal garaja girerek istenilen açıda belli bir hata ölçüsü ile hedefe konumlandığı görülmektedir.



Şekil 2.50. Açısal kontrol senaryosu sonuç durumu

Sanal garajın görüntüsü global statik harita üzerinde görülebilmektedir.



Şekil 2.51. Açısal kontrol uygulaması: statik harita

Verilen uygulamalarda oluşan statik haritalar incelendiğinde, yörünge üzerinde süreksizlik arz eden geçişler gözlemlenebilmektedir. Bu durum imge olarak gösterilen haritada cm/piksel çözünürlüğünün düşük tutulmasından kaynaklanmaktadır. Daha geniş fiziksel alanları görselleştirmek için ideal bir ölçeklendirme olarak 5 cm ≡ 1 piksel kabulü yapılmıştır. Bu çözünürlük haritaları görselleştirmek için en yüksek kabul edilebilir oran olarak değerlendirilmiştir.

## 3. KONUMLANDIRMA SİSTEMİ

Geliştirilen konumlandırma sistemi için tanımlanan iç mekân, Kocaeli Üniversitesi Mühendislik Fakültesi Mekatronik Mühendisliği Bölümü 3. kat koridorudur. Gerçek dünya koordinat merkezi ise koridoru oluşturan yan duvarlardan belirlenen ikisinin izdüşüm kesişimi olarak kabul edilmiştir.

Tez kapsamında geliştirilen gerçek zamanlı iç mekân konumlandırma sistemi bir kamera, kızılötesi projektör grubu ve pasif etiketlerden oluşmaktadır. Konumlandırma sisteminin geliştirme adımları Şekil 3.1'de verilen sırada gerçekleştirilmiştir.



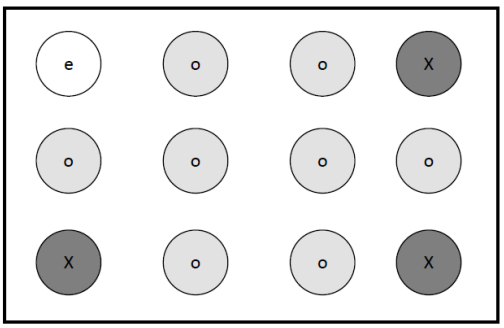

Şekil 3.1. İç-mekân konumlandırma sistemi geliştirme adımları

İlk aşamada kızılötesi (IR) ışık altında yansıtma özelliği olan malzemeden oluşturulan, binary kodlanmış bir pasif etiket tasarımı yapılmıştır. Bu etiketlerin aydınlatılmasını sağlayan IR led projektör devresi ve kamera birbirine entegre edilerek robot üzerine tanımlı bir oryantasyonda monte edilmiştir. Kameranın görüş açısına bağlı olarak, tasarlanan etiketler en az bir etiketin görülebileceği biçimde birbirlerine paralel düzende tavana yerleştirilmiştir. Her biri benzersiz kodlanmış etiketlerin mutlak konumları ve tanımlayıcı numaraları (ID) bir veri tabanına işlenmiştir. Bu aşamada artık iç mekân konumlandırma sisteminin fiziksel altyapısı hazır durumdadır. Robot üzerinde sabitlenen kameranın merkez koordinat sistemi ile gerçek dünya koordinat sistemi arasında bir dönüşüm yapılabilmesi için mobil robot uzaktan kumanda ile kontrol moduna (manuel mod) alınarak kameranın kalibrasyonu yapılmış ve elde edilen parametreler gerçek zamanlı konumlandırma algoritmasında

kullanılmak üzere kaydedilmiştir. Gerçek zamanlı konumlandırma işlemi ise görüntü işleme teknikleri kullanılarak etiketlerin tespiti ve iki boyutlu koordinat dönüşümlerini içermektedir. Son aşamada enkoder tabanlı konumlandırma yaklaşımı olan odometre ile gerçek zamanlı olarak elde edilen konumlandırma ölçümleri, Kalman filtresi ve mobil robot platformunun matematiksel modeli kullanılarak entegre edilmiştir.

### 3.1. Pasif Etiket Tasarımı

Her bir pasif etiket ikili sistemde 3x4 matris formatında tasarlanmıştır. Rotasyon açısının doğru tespit edilebilmesi için köşe elemanlarına her etiket için aynı olan değerler atanmıştır. Etiketin diğer elemanlarının oluşturduğu kombinasyonlar benzersiz ikili kodlar üretmektedir. Bir etiketin sembolik olarak gösterimi Şekil 3.2'de görülmektedir.




Şekil 3.2. Etiket tasarımı

Etiket tasarımında kullanılan semboller ve açıklamaları Tablo 3.1'de verilmiştir.

Tablo 3.1. Etiket tasarımında kullanılan notasyon

| Sembol | Tanımı |
|---|---|
| e | Daima boş (0) |
| x | Daima dolu (1) |
| o | Boş veya dolu (0/1) |

Dikdörtgen plaka IR reflektif olmayan herhangi bir sert malzemeden olabilmekle birlikte, uygulamada düşük maliyeti ve hafifliği sebebiyle mukavva plakalar kullanılmıştır. Plakalar üzerine binary sistemde 1'e karşılık gelmesi istenen alana 25 mm çapında yansıtıcı malzemeden kesilmiş daireler yapıştırılmıştır. Yansıtıcı

bulunan bölgeler projektörden yayılan IR ışığı yansıtacakları için kamera tarafından okunan imgede bu bölgeler parlak, diğer alanlar ise karanlık görünmektedir.

### 3.2. Kamera ve IR Projeksiyon Grubu

Kızılötesi dalga boyunda ışık kaynağı olarak kullanılan led projeksiyon grubu ve bu ışık altında yansıma yapan etiketleri sayısal imgeye dönüştüren kamera Şekil 3.3'te gösterilmiştir.




Şekil 3.3. Kızılötesi led projeksiyonu ve kamera

Kamerada kullanılan lense bant-geçiren filtre yerleştirilerek kameranın sadece kızılötesi ışığı algılaması sağlanmıştır. Kamera dışına bir kabuk geçirilmiş, led grubu da bu dış kabuğun üzerine kamera lensine paralel olacak biçimde monte edilmiştir. Led grubunun beslemesi için, bu tip uygulamalarda kullanılan standart bir güç devresi kullanılmıştır. Üzerinde dâhili bir LDR sensörü de bulunduran bu devre ortam ışık seviyesine göre aydınlatma düzeyini kontrol eden çalışmalarda kullanılan bir devre olup, tez kapsamında geliştirilen sistemde LDR'nin üzeri kapatılarak devre dışı bırakılmıştır. Ortamın görülebilir ışık altındaki aydınlanma seviyesinden bağımsız olarak, sürekli çalışma geriliminde beslenen kızılötesi ledler daimi bir aydınlatma sağlamaktadır. Kamera ve led grubu gerekli ölçüm aletleri kullanılarak hassas bir şekilde zemine paralel biçimde robotun üst plakasına yerleştirilmiştir. Tavan ile zeminin paralel olduğu varsayımı ile kamera ve led ünitesi zemine paralel olarak yerleştirilmiştir. Zemin ile tavan arasındaki öngörülemeyen açısal kaçıklıklar kalibrasyon aşamasında yazılımsal olarak minimize edilmektedir.

### 3.3. Etiketlerin Tavana Yerleşimi

Tavan ile kamera düzlemi arasındaki dik uzaklık ve kamera görüş açısı dikkate alınarak, kameranın herhangi bir konumda iken en az bir etiketi tam olarak görebileceği, belli bazı konumlarda ise birden fazla etiketi aynı karede tam olarak görebileceği bir etiket yerleşim düzeni yapılmıştır. Her biri ikili sistemle benzersiz kodlanmış pasif etiketler konumlandırma yapılacak iç mekânın tavanına belirlenen konumlara monte edilmiştir. Pasif etiketlerin tavana yerleşimi Şekil 3.4'te gösterilmiştir. Şekil üzerinde belirtilen alanlarda etiketler görülebilmektedir. Etiketler belirlenen aralıklarda, birbirine paralel biçimde ve ana koordinat sistemine göre $0^o$ oryantasyon açılarıyla tavana monte edilmiştir



Şekil 3.4. Yansıtıcı etiketlerin tavandaki yerleşim düzeni

Tablo 3.2. Etiket veri tabanı formatı

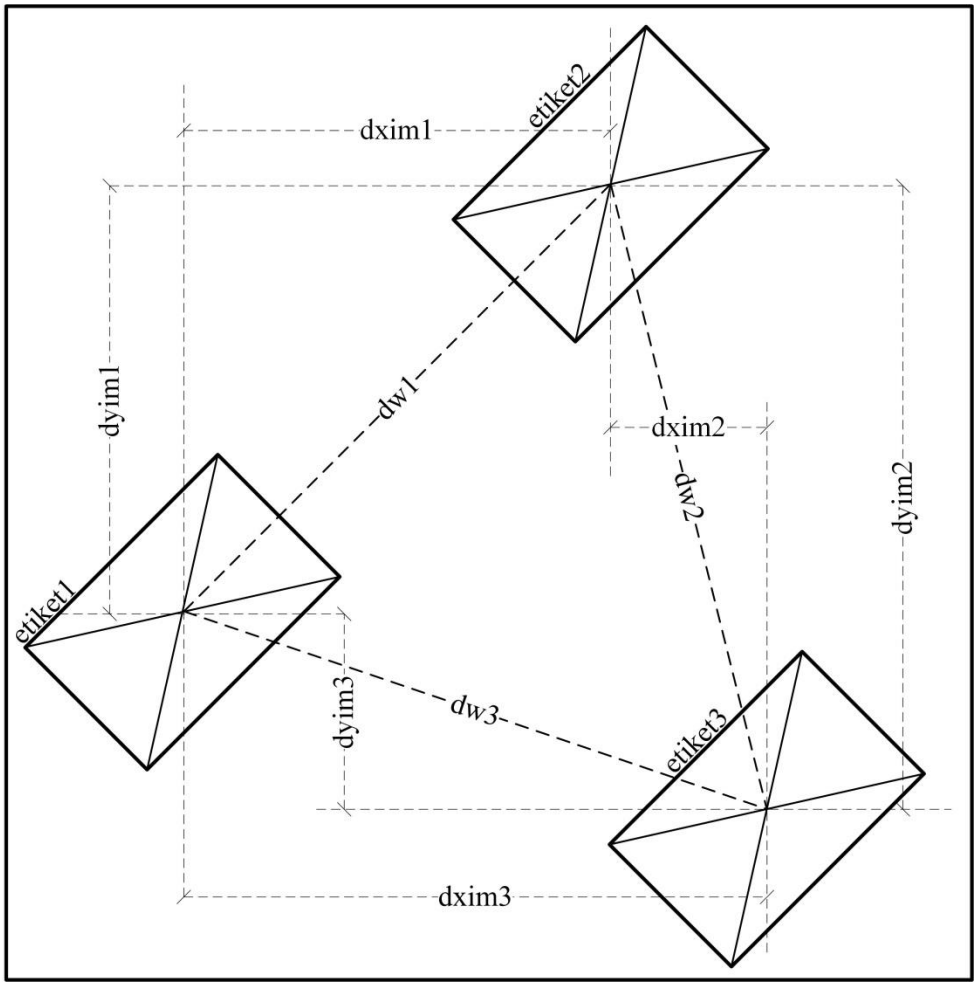
| ID | x-ekseni konumu | y-ekseni konumu |
|---|---|---|
| 12 | 2755 | 13456 |
| 13 | 2790 | 13098 |
| … | … | … |
| … | … | … |
| 250 | 235 | 1356 |

Etiketlerin yerleşimi sırasında yüksek hassasiyetlerde (milimetre seviyesinde) ölçüm yapan lazer mesafe ölçüm cihazı kullanılarak etiketlerin geometrik merkezlerinin mutlak konumları ölçülmüştür. Ölçülen bu konumlar ve benzersiz etiket ID numaraları Tablo 3.2'de verilen formatta bir veri tabanına kaydedilmiştir. ID numaraları 8 bit binary sayıların onluk sistem karşılıkları ile ifade edilirken konum

bilgileri gerçek dünya koordinat sistemine göre milimetre birimindeki konumları ifade etmektedir.

### 3.4. Kamera Kalibrasyonu

Gerçek dünya koordinat sistemi ile kamera koordinat sistemi arasında imgelerin iki boyutlu koordinat dönüşümünü yapabilmek için bir kamera kalibrasyon prosedürü tanımlanmıştır. Kalibrasyon işlemi için mobil robot uzaktan kontrol edilerek etiketlerin yerleştirildiği iç mekânda gezdirilir. Arka planda etiket tanıma algoritması koşturularak en az iki etiketin aynı imge üzerinde tespit edildiği görüntüler işlenir. Kalibrasyon işlemi sonucunda elde edilmek istenen parametreler, iki boyutlu imge uzayında dikdörtgen geometrideki bir pikselin her bir dik kenarının, gerçek dünya koordinatlarında x-ekseni ve y-eksenindeki milimetre karşılığını hesaplamada kullanılacak dönüşüm katsayılarıdır. İmge üzerinde tespit edilen etiketlerin merkezleri arasındaki Öklid uzaklıkları ile bu etiketlerin veri tabanından okunan gerçek merkezleri arasındaki Öklid uzaklıkları arasında her iki eksen bazında optimal çözüm veren bir eşitlik tanımlanır. Bu eşitlik en küçük kareler (LMS) yöntemi kullanılarak çözülür ve vektörel olarak bulunan çözüm kalibrasyon katsayıları olarak değerlendirilir. Aynı imge üzerinde tespit edilen etiketler arasında Öklid uzaklıklarından oluşan ve Denklem (3.1)'de matrissel formda gösterilen ifadeler kullanılarak kalibrasyon işlemi gerçekleştirilmiştir. Kamera kalibrasyonu temel olarak Denklem (3.2)'de verilen problemin çözümü olarak değerlendirilebilir. Burada A matrisi nx2, b vektörü nx1, x vektörü ise 2x1 boyutlarındadır. Aranan çözüm x vektörüdür. Aynı imge üzerinde tam olarak tespit edilen etiket sayısı m ile gösterilmiştir. n sayısı ise m adet etiket arasındaki tek yönlü etkileşim sayısı olarak tanımlanmış ve hesaplanması Denklem (3.1)'de verilmiştir. $d_{xim}$ ve $d_{yim}$ ile gösterilen değişkenler imge uzamsal bölgesinde (piksel biriminde) etiketlerin merkezleri arasında x-ekseni ve y-ekseni için ayrı olarak hesaplanan uzaklıklardır. $d_w$ ile gösterilen değişkenler ise etiketlerin veri tabanında kayıtlı milimetre birimindeki merkezler arası uzaklıklarıdır. Tespit edilen etiket sayısının 3 olması durumu (m=3) için, kalibrasyon işleminde kullanılan etiketler arasındaki Öklid uzaklıkları sembolik olarak Şekil 3.5'te verilmiştir. Burada verilen şema zeminden tavana kamera eksenleri ile bakılması durumunda görülen etiket pozisyonlarıdır. x-ekseni ve y-eksenleri de imge koordinat sisteminin eksenleridir.




Şekil 3.5. Kalibrasyon işlemi için kullanılan Öklid uzaklıkları

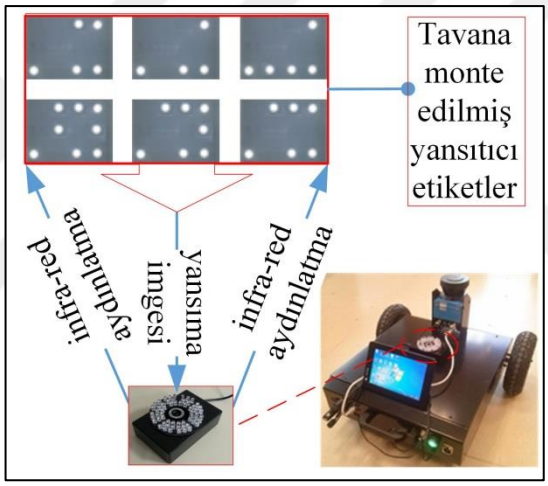
$$A=\begin{bmatrix} d^2_{xim_1} & d^2_{yim_1} \\ \vdots & \vdots \\ d^2_{xim_n} & d^2_{yim_n} \end{bmatrix}, x=\begin{bmatrix} c^2_x \\ c^2_y \end{bmatrix}, b=\begin{bmatrix} d^2_{w_1} \\ \vdots \\ d^2_{w_n} \end{bmatrix}, n=\left\{ \sum_{i=1}^{m-1} i \mid m \in Z^+, m>1 \right\} \quad (3.1)$$

$$Ax=b \quad (3.2)$$

Denklem (3.2)'de verilen sistem en küçük kareler metodu ile çözümlendiğinde x vektörünün elemanları terim bazında karekökleri alınarak $c_x$ ve $c_y$ kalibrasyon katsayıları elde edilmiştir. Bu katsayılar, gerçek zamanlı konumlandırma işleminde ilgili transformasyonlar için parametre olarak kullanılmıştır.

### 3.5. EKF ve İmge İşleme Tabanlı Konumlandırma Sistemi

Etiketlerin, kamera ve IR led grubunun tasarlanıp montajlarının yapılmasının ardından kalibrasyon yapılmış ve gerçek zamanlı konumlandırma için robot platformu hazır duruma getirilmiştir. Bu işlemin temel şeması Şekil 3.6'da verilmiştir. Gerçek zamanlı konumlandırma işlemi, kendi içerisinde etiket tespiti, konum ölçümünün hesaplanması ve odometre bilgisinin Kalman filtresi ile ölçümlere entegre edilmesi adımlarını içermektedir. Etiket tespiti işlemi, kamera kalibrasyonu aşamasında da uygulanan bir prosedürdür. Kalibrasyon aşamasında en az iki etiketin tespit edilmesi gerekirken gerçek zamanlı konumlandırma aşamasında en az bir etiketin tam olarak görülmesi yeterli olmaktadır.

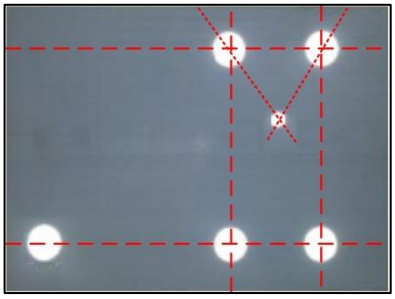



Şekil 3.6. İç mekân konumlandırma sistemi şeması

Etiket tespiti işlemi kameradan RGB görüntü okunması ile başlar. Gri seviye ve siyah-beyaz imge dönüşümlerinin ardından elde edilen imge üzerinde kızıl ötesi ışık altında parlak bir görüntü yansıtan dairesel bölgelerin lojik 1, kalan piksellerin lojik 0 olması beklenmektedir. Uygulamada ise güneş ışığının kısmi etkisi ve titreşimlerden kaynaklı bozunumlar gibi beklenmeyen gürültüler oluşabilmektedir. Öngörülemeyen bu etkileri imge işleme teknikleri ile giderebilecek bir filtreleme, kümeleme ve etiketleme algoritması geliştirilmiştir. Tavanda yer alan pasif etiketlerin siyah-beyaz (binary) imgeleri gerçek zamanlı video olarak işlenmiştir.

Binary imge üzerinde "blob analysis" metodu kullanılarak lojik 1 değere sahip kümeler ağırlık merkezi ve piksel boyutları olarak tespit edilir. Alınan görüntüde

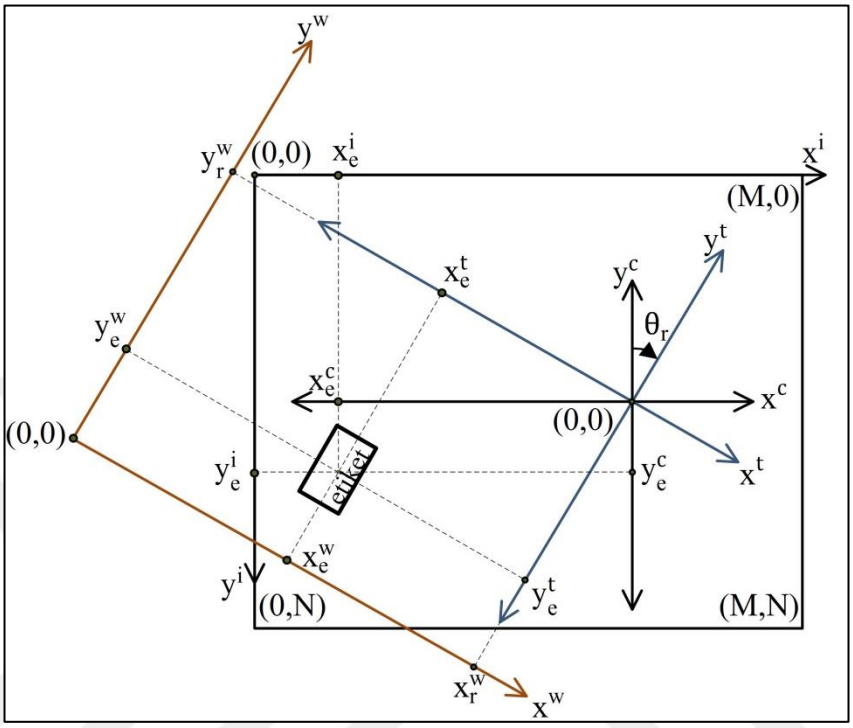
herhangi bir lojik 1 değeri olmaması durumunda yeni bir görüntü alınır. Etiketlerin varsayılan özelliklerinden olan belirli üç köşenin dolu olma durumu kontrol edilerek bu sayının altında blob (lojik 1 olan bağımsız kümeler) olması durumunda mevcut imge filtrelenir ve yeni imge okunur. Etiketler üzerindeki dairelerin komşu diğer bir etikete olan en kısa uzaklığı her durumda kendi etiketinin geometrik merkezine olan uzaklığından daha fazladır. Bu bilgi dâhilinde tespit edilen blobların birbirinden belli bir eşik mesafeden daha uzak olması durumunda farklı etiketlere ait olacağı sonucuna varılır ve algoritma tarafından tespit edilen etiket sayısı bir arttırılır. Her etiketin üzerindeki bloblar ilgili etiketin indeksi ile indekslenir. Yeni bir blobun hangi etikete ait olduğuna karar vermek için aynı indeksli tüm bloblar ile olan uzaklığı hesaplanır, her bir blob için tanımlı eşik değerin altında ise ilgili etikete dâhil edilir. Eşik değerin altında bir yakınlıkta olan ancak gürültü kaynağı tarafından üretilmiş bir blobun filtrelenmesi için ise çeşitli ekstra kriterler belirlenmiştir. Tespit edilen bir blob bir etikete dâhil edilmeden önce etiket üzerindeki diğer bloblar ile doğrusallık kontrolü yapılır. Şekil 3.7'de bu durumu görselleştiren bir resim verilmiştir. Reflektörler için tasarım aşamasında ön görülen doğrusallık desenine uygun olmayan eleman reflektör olarak değerlendirilmez ve etikete dâhil edilmeden filtrelenir.



Şekil 3.7. Etiketler üzerinde doğrusallık kontrolü

Tespit edilen reflektörler için minimum bounding box (MinBBox) tanımlanarak en-boy oranı kontrolü yapılır. Belirli bir tolerans dâhilinde en-boy oranı tutarsız olan etiketler bu aşamada elenir. MinBBox ile imge kenarlarının kesişmesi durumunda da ilgili etiket tüm filtrelerden geçse dahi elenir. Etiket köşelerinin durumu son aşamada kontrol edilerek kriterleri sağlayan bloblardan en az üç tanesinin etiketin köşelerinde yer alıp almadığı kontrol edilir. Son olarak etiket ID numarası hesaplanır, veri tabanında kayıtlı olup olmadığı kontrol edilir ve eğer kayıt dışı bir ID numarasına

sahip ise ilgili etiket elenir. Nihai durumda kalan etiket(ler)in veri tabanında kayıtlı gerçek dünya koordinatlarındaki konumları okunarak, konum ölçümünün hesaplanması aşamasında kullanılacak alt fonksiyona gönderilir.

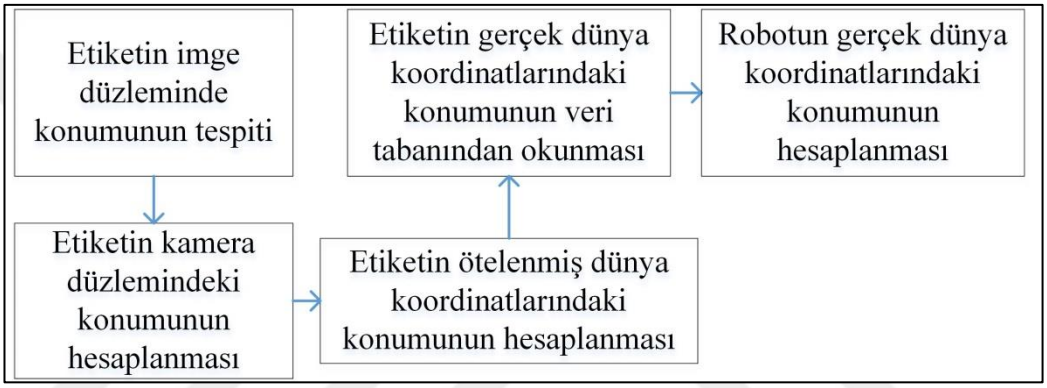



Şekil 3.8. Konumlandırma sisteminde tanımlı eksenler

Tablo 3.3. Eksenler ve konumlar için kullanılan notasyon

| Sembol | Tanımı |
|---|---|
| $x^w$, $y^w$ | Gerçek dünya koordinat sisteminin sırasıyla x-ekseni ve y-ekseni |
| $x^i$, $y^i$ | İmge koordinat sisteminin sırasıyla x-ekseni ve y-ekseni |
| $x^c$, $y^c$ | Kamera koordinat sisteminin sırasıyla x-ekseni ve y-ekseni |
| $x^t$, $y^t$ | Ötelenmiş dünya koordinat sisteminin sırasıyla x-ekseni ve y-ekseni |
| N, M | İmgenin uzamsal yükseklik ve genişliği |
| $\theta_r$ | Robotun oryantasyon açısı |
| $x^w_e$, $y^w_e$ | Etiketin gerçek dünya koordinatlarında sırasıyla x-ekseni ve y-eksenindeki konumu |
| $x^t_e$, $y^t_e$ | Etiketin ötelenmiş dünya koordinatlarında sırasıyla x-ekseni ve y-eksenindeki konumu |
| $x^i_e$, $y^i_e$ | Etiketin imge koordinatlarında sırasıyla x-ekseni ve y-eksenindeki konumu |
| $x^c_e$, $y^c_e$ | Etiketin kamera koordinatlarında sırasıyla x-ekseni ve y-eksenindeki konumu |
| $x^w_r$, $y^w_r$ | Robotun gerçek dünya koordinatlarında sırasıyla x-ekseni ve y-eksenindeki konumu |

Şekil 3.8'de verilen koordinat sistemleri ve kullanılan notasyon Tablo 3.3'te açıklanmıştır. Tespit edilen her bir etiketin imge uzayındaki, kamera merkez koordinatlarındaki ve ötelenmiş dünya koordinatlarındaki konumları sırasıyla hesaplandıktan sonra ötelenmiş koordinatlardaki konum bilgisinin etiketin gerçek dünya koordinatlarındaki konum bilgisine göre konumu hesaplanarak robotun bir etiket bazındaki gerçek konumu hesaplanmaktadır. Birden fazla etiket için hesaplanan x-ekseni ve y-ekseni konumlarının aritmetik ortalaması robotun gerçek konumu olarak kabul edilmektedir. İmge işleme tabanlı konum ölçüm sistemi işlem adımları Şekil 3.9'da sırasıyla verilmiştir.




Şekil 3.9. Konumlandırma sistemi iş akışı

$$\begin{bmatrix} x_e^c \\ y_e^c \end{bmatrix} = \begin{bmatrix} -(0.5\times N - x_e^i) \\ 0.5\times M - y_e^i \end{bmatrix} \tag{3.3}$$

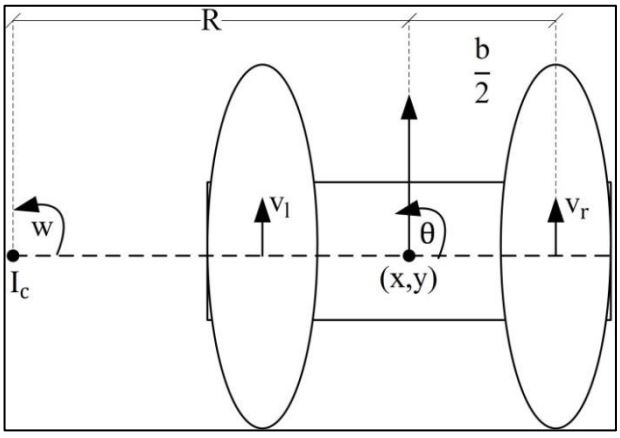
$$\begin{bmatrix} x_e^t \\ y_e^t \end{bmatrix} = R(\theta_r) \begin{bmatrix} c_x x_e^c \\ c_y y_e^c \end{bmatrix} \tag{3.4}$$

$$\begin{bmatrix} x_r^w \\ y_r^w \end{bmatrix} = \begin{bmatrix} x_e^w \\ y_e^w \end{bmatrix} - \begin{bmatrix} x_e^t \\ y_e^t \end{bmatrix} \tag{3.5}$$

$$x_r^w = \sum_{i=1}^{n_e} \frac{x_{r_i}^w}{n_e}, y_r^w = \sum_{i=1}^{n_e} \frac{y_{r_i}^w}{n_e}, n_e \in Z^+ \tag{3.6}$$

Bir etiketin imge boyutunda tespit edilmesi ile ilgili prosedür tamamlandıktan sonra etiketin geometrik merkezinin kamera koordinat sistemine dönüştürülmesi Denklem

(3.3)'te verilmiştir. Burada ilgili etiket için orijin sol üst köşe noktasından imge merkezine ötelenmiş, y-ekseni ise terslenmiştir. Bu işlemin amacı, kamera odak noktasının robotun konum merkezi olarak kabul edilmesidir. Nihai durumda hesaplanacak olan konum kameranın odak merkezinin gerçek dünya koordinatlarındaki konumu olmaktadır. Etiketin ötelenmiş dünya koordinatlarındaki konumu Denklem (3.4)'te verilen işlem ile hesaplanmaktadır. Denklem (3.4)'te verilen $R(\theta_r)$ ifadesi, z-ekseninde $\theta_r$ açısı ile ifade edilen rotasyon matrisidir. Aynı denklemde $c_x$ ve $c_y$ ile verilen ifadeler, kamera dahili parametreleri olup, sırasıyla x-ekseni ve y-ekseni için hesaplanan kalibrasyon katsayılarıdır. Tespit edilen etiketin gerçek dünya koordinatlarındaki konumu, etiket tasarımı aşamasında tanımlanan veri tabanından, benzersiz ID numarası ile karşılıklı kontroller yapılarak okunur. Bilinen bu değer ve Denklem (3.4)'te hesaplanan konum vektörü Denklem (3.5)'te verilen işleme tabi tutularak kamera odak noktasının yani mobil robot platformunun gerçek dünya koordinatlarındaki konumu hesaplanır. Tek bir etikete ait imge işleme prosedürü ve verilen dönüşümler ile konum hesaplanabileceği gibi, birden çok etikete ait çıktılar kullanılarak da konumlandırma yapılabilmektedir. Bu durumda yapılan işlem temel olarak farklı etiketlerden alınan konumlandırma sonuçlarının aritmetik ortalamalarının hesaplanmasıdır. Denklem (3.6)'da tespit edilen etiket sayısının $n_e$ pozitif tamsayısı olması kabulü altında yapılan konum hesaplanma işlemi gösterilmiştir.




Şekil 3.10. Diferansiyel sürüş modelinin sembolik gösterimi

Konum ölçüm sisteminden alınan konum bilgisi, model tabanlı konum tahminleri EKF metodu uygulanarak iyileştirilmiştir. Model tahmini için varsayılan kinematik

model diferansiyel sürüş olarak belirlenmiştir. Diferansiyel sürüş sistemine sahip mobil robotlar ortak dönme eksenli iki aktif tekerlek ve sarhoş tekerlerden oluşan hareket birimlerinden oluşmaktadır. Robotun açısal yönelimi, bağımsız tekerleklerin değişen hızları ile sağlanmaktadır. Robot ortak ekseni üzerinde bulunan bir nokta etrafında dönme eğilimi gösterir. Kinematik modelin sembolik gösterimi Şekil 3.10'da gösterilmiştir.

Şekil 3.10'da verilen modelde kullanılan notasyon Tablo 3.4'te açıklanmıştır.

Tablo 3.4. Kinematik model için kullanılan notasyon

| Sembol | Tanımı |
|---|---|
| $v_r$, $v_l$ | Sırasıyla sağ ve sol teker hızları |
| $I_c$ | Dönme merkezi |
| $\theta$ | Oryantasyon açısı |
| b | Dingil genişliği |
| R | Dönme yarıçapı |
| w | Açısal hız |
| x, y | 2 boyutlu uzayda robotun sırasıyla x-ekseni ve y-eksenindeki konumları |

Şekil 3.10'da verilen kinematik model baz alınarak, robotun şekilde verilen yön ve doğrultuda sağ ve sol tekerleklerin kaymadan yuvarlanma hareketi yaptığı varsayılmıştır. Ön ve arkada simetrik olarak serbest küresel tekerlekler ile dengeli ve zemine paralel bir durma pozisyonu sağlanabilir.

$$R=\frac{b}{2}\frac{v_r+v_l}{v_r-v_l} \tag{3.7}$$

$$w=\frac{v_r-v_l}{b} \tag{3.8}$$

$$\begin{bmatrix} I_{c_x} \\ I_{c_y} \end{bmatrix}=I_c=\begin{bmatrix} x-R\sin\theta \\ y+R\cos\theta \end{bmatrix} \tag{3.9}$$

Uygulamada serbest tekerleklerin konumu ve sayısı değişebilmektedir. Tahrik edilen sağ ve sol tekerlekler, dönme ekseninde yanal hareket yapamazlar. Holonomik olmayan kısıtlar bu modelde geçerlidir. Bu model için robotun dingil genişliği ve

tekerleklerin doğrusal hızlarına bağlı olan ani dönme yarıçapı Denklem (3.7)'de verilmiştir. Ani dönme merkezi etrafında oluşan açısal hız Denklem (3.8)'de verilmiştir. İki boyutlu uzayda robotun dönme merkezi de iki bileşenden oluşmaktadır ve bileşenlerinin hesaplanması Denklem (3.9)'da verilmiştir.

Gerçek zamanlı çalışma periyodu kadar geçen bir sürede ($\delta_t$) sağ ve sol tekerleklerin aldıkları mesafeler ($s_r$ ve $s_l$) ve robotun oryantasyon açısındaki değişim ($\delta_\theta$) Denklem (3.10)'da verilmiştir.

$$s_l=V_l\delta_t,\ \ s_r=V_r\delta_t,\ \ \delta_\theta=w\delta_t \tag{3.10}$$

Diferansiyel sürüş modeline sahip mobil robotun durum-uzayı modeli Denklem (3.11)'de verilmiştir.

$$\begin{bmatrix} x_{t+\delta_t} \\ y_{t+\delta_t} \\ \theta_{t+\delta_t} \end{bmatrix} = \begin{bmatrix} \cos(w\delta_t) & -\sin(w\delta_t) & 0 \\ \sin(w\delta_t) & \cos(w\delta_t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x-I_{c_x} \\ y-I_{c_y} \\ \theta \end{bmatrix} + \begin{bmatrix} I_{c_x} \\ I_{c_y} \\ w\delta_t \end{bmatrix} \tag{3.11}$$

Denklem (3.7) ile Denklem (3.11) arasında tanımlanan ifadeler kullanılarak yapılan ara işlemler ile diferansiyel sürüş modeline ait sistem vektörü hesaplanmıştır. Denklem (3.12)'de bu işlemlerin sonucu olarak elde edilen model gösterimi verilmiştir.

$$\begin{bmatrix} x_{t+\delta_t} \\ y_{t+\delta_t} \\ \theta_{t+\delta_t} \end{bmatrix} = f(x_t,y_t,\theta_t,s_r,s_l) = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + \begin{bmatrix} \frac{b}{2}\frac{s_r+s_l}{s_r-s_l}(\sin(\theta_t+\delta_\theta)-\sin\theta_t) \\ \frac{b}{2}\frac{s_r+s_l}{s_r-s_l}(-\cos(\theta_t+\delta_\theta)+\cos\theta_t) \\ \delta_\theta \end{bmatrix} \tag{3.12}$$

Diferansiyel sürüş modeline ait durum geçiş ve giriş matrisleri (A ve B matrisleri), doğrusal modeli için sistem vektörü (f vektörü) üzerinde Jacobian işlemi yapılarak elde edilebilir. Bu işlemleri sırasıyla Denklem (3.13) ve Denklem (3.14)'te verilmiştir. Giriş matrisi Kalman filtresi uygulamasında kullanılmayacağı için sadece hesaplama yöntemi verilmiştir. Çıkış matrisi olarak gösterilen C matrisi, tüm durumların ölçülebildiği varsayımı altında 3x3 boyutunda birim matris olarak

değerlendirilmiştir. Denklem (3.14)'te çıkış matrisi verilmiştir.

$$A=\begin{bmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} & \frac{\partial f_x}{\partial \theta} \\ \frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y} & \frac{\partial f_y}{\partial \theta} \\ \frac{\partial f_\theta}{\partial x} & \frac{\partial f_\theta}{\partial y} & \frac{\partial f_\theta}{\partial \theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \frac{s_r+s_l}{s_r-s_l}(\cos\theta-\cos(\theta+\delta_\theta)) \\ 0 & 1 & \frac{s_r+s_l}{s_r-s_l}(\sin\theta-\sin(\theta+\delta_\theta)) \\ 0 & 0 & 1 \end{bmatrix} \quad (3.13)$$

$$B=\begin{bmatrix} \frac{\partial f_x}{\partial s_r} & \frac{\partial f_x}{\partial s_l} \\ \frac{\partial f_y}{\partial s_r} & \frac{\partial f_y}{\partial s_l} \\ \frac{\partial f_\theta}{\partial s_r} & \frac{\partial f_\theta}{\partial s_l} \end{bmatrix}, C=\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.14)$$

Elde edilen A ve C matrisleri genişletilmiş Kalman Filtresi (EKF) uygulamasında parametre olarak kullanılmıştır. EKF denklem setleri Tablo 3.5'te verilmiştir.

Tablo 3.5. EKF denklem setleri

| Model tahmini | Ölçüm güncellemesi |
|---|---|
| $x_k^- = f(x_{k-1}, s_r, s_l)$ | $K_k = P_k^- C^T (CP_k^- C^T + R)^{-1}$ |
| | $\hat{x}_k = x_k^- + K_k(y_k - Cx_k^-)$ |
| $P_k^- = AP_{k-1}A^T + Q$ | $P_k = (I - K_k C)P_k^-$ |

Tablo 3.5'te verilen setlerde Q ve R ile gösterilen değişkenler sırasıyla süreç ve ölçüm hata kovaryans matrisleridir. Model altında tanımlı x,y ve θ durum hatalarının kendi aralarında bir korelasyon olmadığı varsayımı ile Q ve R matrisleri köşegen matris olarak tanımlanmıştır.

$$Q=\begin{bmatrix} q_x^2 & 0 & 0 \\ 0 & q_y^2 & 0 \\ 0 & 0 & q_\theta^2 \end{bmatrix}, \; R=\begin{bmatrix} r_x^2 & 0 & 0 \\ 0 & r_y^2 & 0 \\ 0 & 0 & r_\theta^2 \end{bmatrix} \quad (3.15)$$

Denklem (3.15)'te bu matrisler verilmiştir. Burada $q_x$, $q_y$, $q_\theta$ süreç hata varyansları, $r_x$, $r_y$, $r_\theta$ ise ölçüm hata varyanslarıdır. K matrisi Kalman kazanç matrisidir. Bu

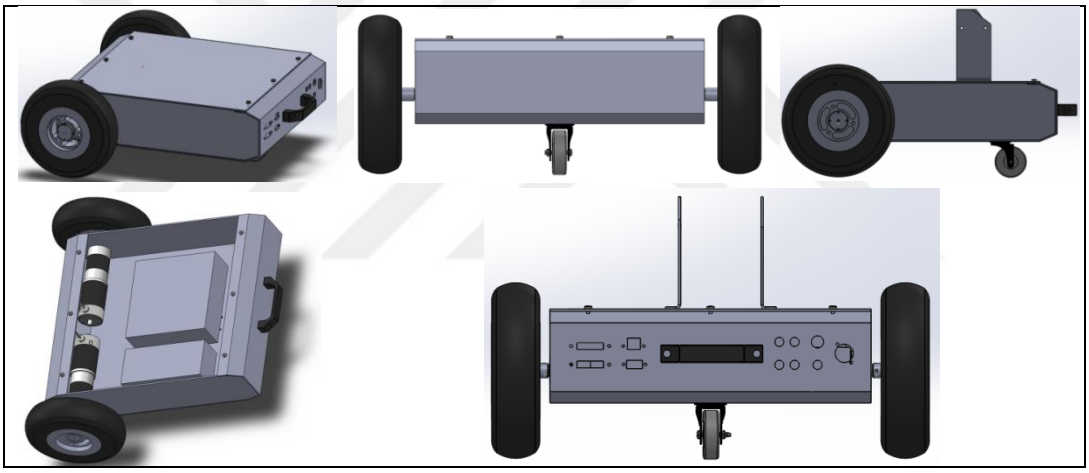
matrisin değeri rekürsif olarak sürekli güncellenir. Ölçüm güncellemesi adımında y ile gösterilen ölçümün çıkışa etki derecesini belirler. P matrisi Kalman filtresi hata matrisidir. Başlangıç durumunda Q matrisine eşit olarak başlatılır ve Kalman kazancına bağlı olarak her adımda güncellenir.

Süreç hata varyansları odometre hesabında; ölçüm hata varyansları ise iç mekân konumlandırma sisteminde birim zamanda x-ekseni, y-ekseni ve oryantasyon açısı üzerinde oluşan hataların varyanslarıdır.

# 4. MOBİL ROBOT PLATFORMU

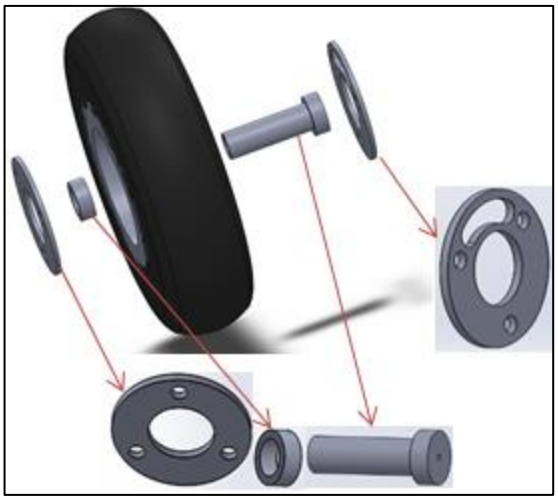
## 4.1. Sistemin Mekanik Yapısı ve İmalatı

Robotun şasesi 3 mm kalınlığında alüminyum malzemeden üretilmiştir. Ön ve arka plakaları alt plakayla bükülmüş; yan plakalar ise kaynaklanmıştır. Üst kapak ise vidalanarak sabitlenmiştir. Önde iki adet 22 cm çapında ve 6 cm et kalınlığında pnömatik tekerlek, arkada ise bir adet serbest tekerlek bulunmaktadır. Robotun taşımasını kolaylaştırmak amacıyla arka plakaya bir adet plastik tutamak monte edilmiştir. LIDAR üst kapağa iki adet metal ayak ile sabitlenmiştir.



Şekil 4.1. Mobil robot platformu katı modeli

Mobil robot platformunun katı modeli Şekil 4.1'de gösterilmiştir. Mobil robot platformunun arka bölümünde donanımsal girişler için açılmış soket girişleri bulunmaktadır. Aynı bölgeye monte edilen tutamak taşıma ve tampon işlevi görmektedir. İç bölmede DC motorlar tekerlek dönme ekseni boyunca yerleştirilmiştir. Üst kapak iki yönlü olarak monte edilerek robotun varsayılan yönü değiştirilebilmektedir. Tahrikli tekerleklerin şaseye bağlantısının yapılabilmesi için gerekli bağlantı elemanlarının katı modeli Şekil 4.2'de gösterilmiştir. Motor bağlantısı gerçekleştirilecek olan tekerleklerin robot şasesine olan bağlantısı yapılırken havalı tekerlekler karşılıklı iki disk arasına üç noktadan vidalanarak sabitlenmiştir.



Şekil 4.2. Tekerlek mekanizması ve bağlantı elemanlarının katı modeli



Şekil 4.3. Mekanik imalat ve montaj sonrası görünüm

Redüktör miline doğrudan sabitlenen diskli tekerleklerin göbeğine burç geçirilerek olası aşınmaların önlenmesi amaçlanmıştır. Üç tekerlekli tasarımın imalat sonucundaki genel görünümü Şekil 4.3'te verilmiştir. LIDAR, robotun ön-üst bölmesine cihaza özgü tasarlanan montaj ayakları yardımı ile monte edilmiştir. Robotun dış gövdesinde iç kısımdaki elektronik donanımlara erişimi sağlayan soketler bulunmaktadır. Üst kapak üzerinde LCD monitör, konumlandırma sistemine ait kamera - led grubu ve LIDAR bulunmaktadır. Robotun arka tarafında plastik bir tutamak, şarj konnektörü, robotu açıp kapatabilmek için ON/OFF anahtarı, robotun içerisinde bulunan endüstriyel bilgisayar için ON/OFF butonu, 4 adet USB, 1 adet Ethernet ve 1 adet HDMI konektörü, 2 adet 24V ve 2 adet 12V güç konnektörleri bulunmaktadır. Dâhili bilgisayara bağlı LCD monitör, $45^{o}$ eğimli bir bağlantı aparatı ile kapak birimine monte edilmiştir.

## 4.2. Sistemin Kontrol Birimi ve Gömülü Yazılımı

Otonom hareket sırasında açıklık tespiti, lokal hedef belirleme ve çarpışma engelleme işlemlerinin ardından güvenli lokal hedef elde edilmektedir. Mobil robotun bu noktaya yönlendirilebilmesi için bir kontrol birimi tanımlanmıştır. Oluşturulan hız büyüklükleri, motor sürücü kartına iletilerek robotun fiziksel olarak ilgili hedefe yönlendirilmesi sağlanmaktadır. Hedefin konumu ve anlık robot oryantasyonuna göre bir sonraki işlem adımında istenen hız büyüklükleri oransal bir kontrol yaklaşımı ile hesaplanmaktadır. Sağ ve sol tekerlek hızları cm/s birimi ile hesaplanarak sonraki adımda, kullanılan sürücü kartında tanımlı haberleşme protokolüne uygun formata dönüştürülmektedir.

Mobil robotun yatay ve düşey eksenlerdeki konumu $pr_x$ ve $pr_y$, hedefin yatay ve düşey eksenlerdeki konumu $pt_x$ ve $pt_y$, robotun anlık oryantasyonu $\theta_r$ olmak üzere, mobil robotun global harita üzerinde verilen bir hedefe yönelmesi için yapması gereken açısal değişim ($\Delta_\theta$) Denklem (4.1) ve Denklem (4.2)'te verilmiştir.

$$\Delta_\theta^* = \text{atan2}(pt_y - pr_y,\ pr_x - pr_x),\ \Delta_\theta^* = \begin{cases} \Delta_\theta^*, & \Delta_\theta^* \geq 0 \\ \Delta_\theta^* + 360, & \Delta_\theta^* < 0 \end{cases} \tag{4.1}$$

$$\Delta_\theta = \Delta_\theta^* - \theta_r,\ \Delta_\theta = \begin{cases} \Delta_\theta - 360, & \Delta_\theta > 180 \\ \Delta_\theta + 360, & \Delta_\theta < -180 \end{cases} \tag{4.2}$$

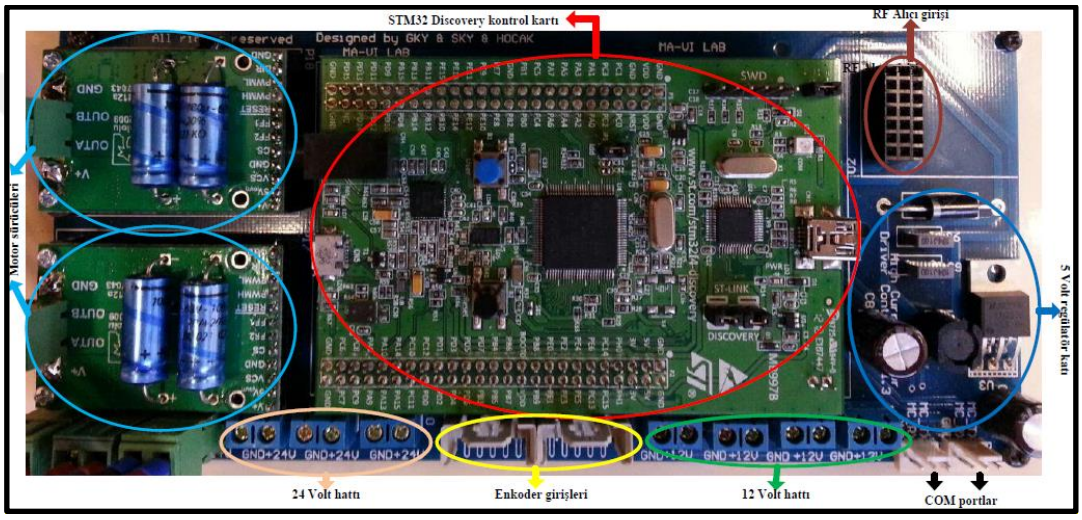
Navigasyon hızı ($v_c$), maksimum hız ($v_{max}$), minimum hız ($v_{min}$) ve açısal hata ($\Delta_\theta$) verilerine bağlı olan sağ ve sol tekerlek hızlarının kontrolü sırasıyla Denklem (4.3)'te verilmiştir. Oransal kontrol katsayısı k ise 0 ile 1 arasında keyfi bir katsayıdır.

$$v_r = \begin{cases} v_c + \dfrac{k\theta_r (v_{max} - v_c)}{90} & \theta_r > 0 \\ v_c + \dfrac{k\theta_r (v_c - v_{min})}{90} & \theta_r \leq 0 \end{cases},\quad v_l = \begin{cases} v_c - \dfrac{k\theta_r (v_c - v_{min})}{90} & \theta_r > 0 \\ v_c - \dfrac{k\theta_r (v_{max} - v_c)}{90} & \theta_r \leq 0 \end{cases} \tag{4.3}$$

Denklem (4.3)'te verilen sağ ve sol teker hızları cm/s birimi ile ifade edilmektedir. Kullanılan sürücü devresinin teknik yapısına uygun olan formata dönüştürmek için bir alt program bulunmaktadır. Bu program aracılığı ile temel olarak hız bilgisi cm/s

biriminden devir/dakika (RPM) birimine dönüştürülerek seri haberleşme protokolü üzerinden kontrol kartına gönderilmektedir.

Geliştirilen elektronik ana kartın donanımsal bileşenleri STM32F4 Discovery kontrol kartı, RF uzaktan kumanda alıcısı, 5 Volt regülatör katı, DC motor sürücüleri, 12 Volt hattı, 24 Volt hattı, seri port (COM port) soketleri ve enkoder soketlerinden oluşmaktadır. Motor sürücüleri Pololu firmasından temin edilen sürücü kartlarıdır. Ana kontrol kartının elektronik şeması da Ek-A' da verilmiştir. Geliştirilen kartın görünümü Şekil 4.4'te verilmiştir.

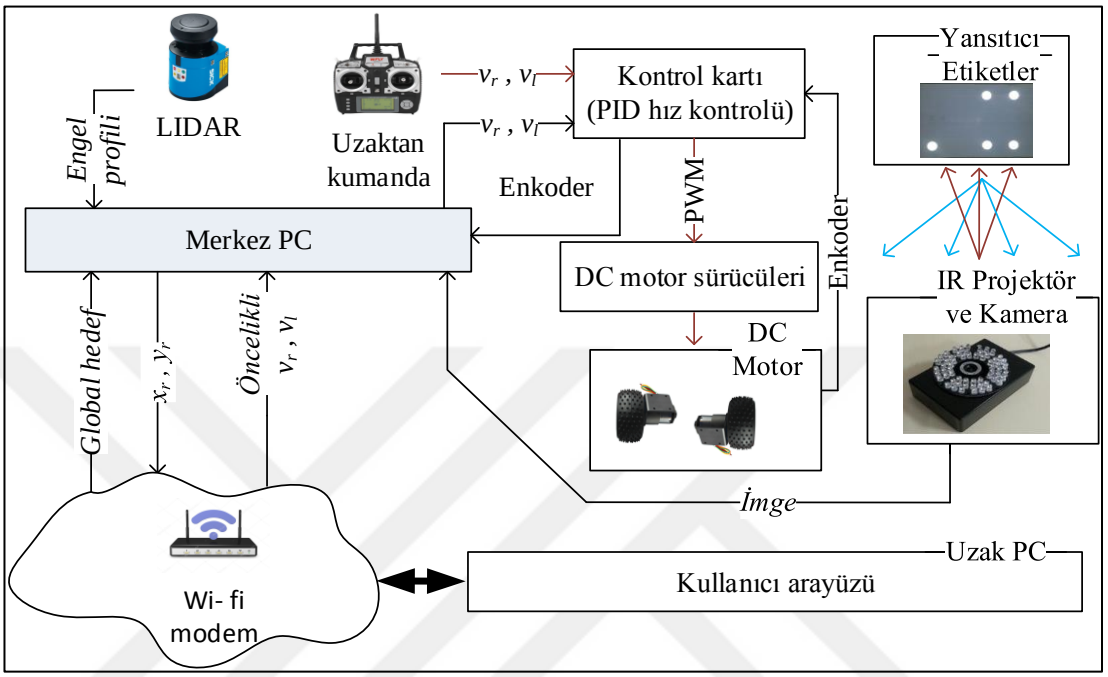



Şekil 4.4. Elektronik ana kartın montaj sonrası görünümü

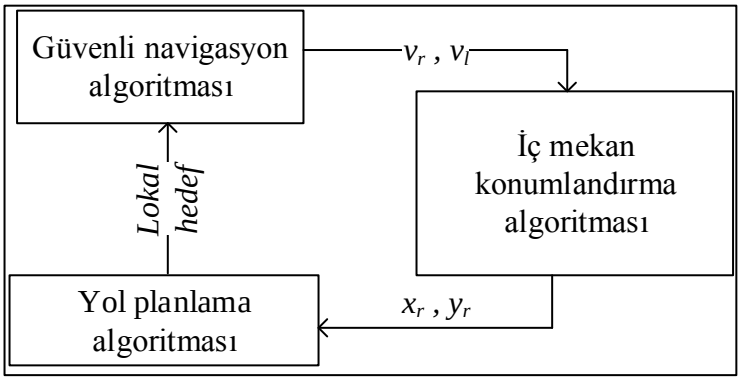
STM32F4 Discovery kontrol kartı üzerinde bir kontrol algoritması geliştirilmiştir. Kontrol kartı üzerinden okunan enkoder verileri işlenerek tekerleklerin hızları bağımsız olarak tespit edilebilmektedir. COM port üzerinden herhangi bir cihazdan referans hız ve yön bilgisi veri paketi olarak alınabilmektedir. Bilinen hız değerleri kullanılarak bir PID hız kontrolü uygulanmaktadır. Ayrıca, enkoder konum bilgileri COM port üzerinden veri paketi olarak harici bir terminale gönderilebilmektedir. Bunların dışında, uzaktan kumanda verici modülünden okunan sağ-sol, ileri-geri ve hız komutları üç kanaldan okunarak robotun hızı kontrol edilebilmektedir. Bu sayede mobil robot hem uzaktan kumanda ile hem de başka bir cihazdan gönderilecek veri paketleri ile kontrol edilebilmektedir. Uzaktan kumanda verisi öncelikli olarak işleme alınarak otonom moddan çıkarılıp, manuel moda geçilebilmektedir.

## 4.3. Mobil Robot Sistem Mimarisi

Geliştirilen mobil robot sisteminin genel sistem mimarisi Şekil 4.5 ve Şekil 4.6'da verilmiştir.




Şekil 4.5. Sistemin genel mimarisi




Şekil 4.6. Merkez PC algoritma mimarisi

Sistemin temel bileşenleri aşağıdaki gibi sıralanabilir:

- Intel Core i7 3770 3,4 GHz işlemci, 128 GB SSD, 8 GB ddr3 RAM taşıyan IPC Mini-ITX NF9E-Q77-LF endüstriyel PC (robotun içerisinde konumlandırılan merkez PC)
- 7 inç dokunmatik LCD monitör

- Midwest Motion Products firmasından temin edilen 158 rpm, 3,5 Nm tork ve 24 Volt plaka değerlerine sahip redüktörlü DC motorlar ve bunlara entegre olan 500 pulse/tur enkoderler
- DC motor sürücü devreleri (Pololu)
- Güvenli navigasyon, statik ve dinamik yol planlama algoritmaları
- Elektronik kontrol kartı
- SICK LMS100 LIDAR
- 2,4 GHz radyo frekans bandında çalışan uzaktan kumanda alıcı-vericisi
- Kablosuz modem
- İç-mekân konumlandırma sistemi
- Uzaktan kontrol ve takip işlemlerinin gerçekleştirilebileceği uzak PC

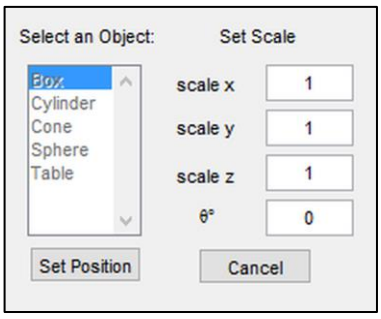
Uzak PC veya dahili panelden verilen global hedef ve varsa statik harita imgesi yol planlama modülü tarafından işlenerek gerçek zamanlı dinamik navigasyon süreci başlatılmaktadır. Güvenli navigasyon algoritması tarafından üretilen hız bilgisi kontrol kartına gönderilerek, PWM formatına dönüştürülür ve DC motor sürücüleri beslenir. Motorlardan okunan enkoder verileri kontrol kartına geri beslenerek hız bilgisi hesaplanmaktadır ve geri beslemeli olarak hız kontrolü gerçekleştirilmektedir. Kontrol kartı için gerekli olan referans hız girişi eş zamanlı olarak uzak PC'den, navigasyon algoritmasından veya uzaktan kumandadan sağlanabilmektedir. Ancak, uzaktan kumanda veya uzak PC'den gönderilen referans hız ve yön değerleri güvenlik amacıyla öncelikli olarak işleme alınmaktadır. Robot hareket halindeyken merkez PC sürekli olarak kablosuz modem üzerinden uzak PC' ye robotun anlık konum ve oryantasyon bilgilerini göndermektedir. Uzak PC için geliştirilen takip ve kontrol arayüzü üzerinden robotun hızına ve yönüne müdahale edilebilmektedir. Ayrıca, robot üzerine monte edilen dokunmatik ekran üzerinden de robotun hareketleri izlenebilir ve merkez PC ile ilgili tüm işlemler (güvenli navigasyon, iç-mekân konumlandırma ve yol planlama algoritmaları) gerçekleştirilebilmektedir. Uzak PC arayüzü ve robot üzerindeki monitörden geçek zamanlı olarak robotun hangi konum ve hızda olduğu izlenebilmektedir.

# 5. MOBİL ROBOT BENZETİM ARAYÜZÜ

## 5.1. Benzetim Arayüzünün Yapısı

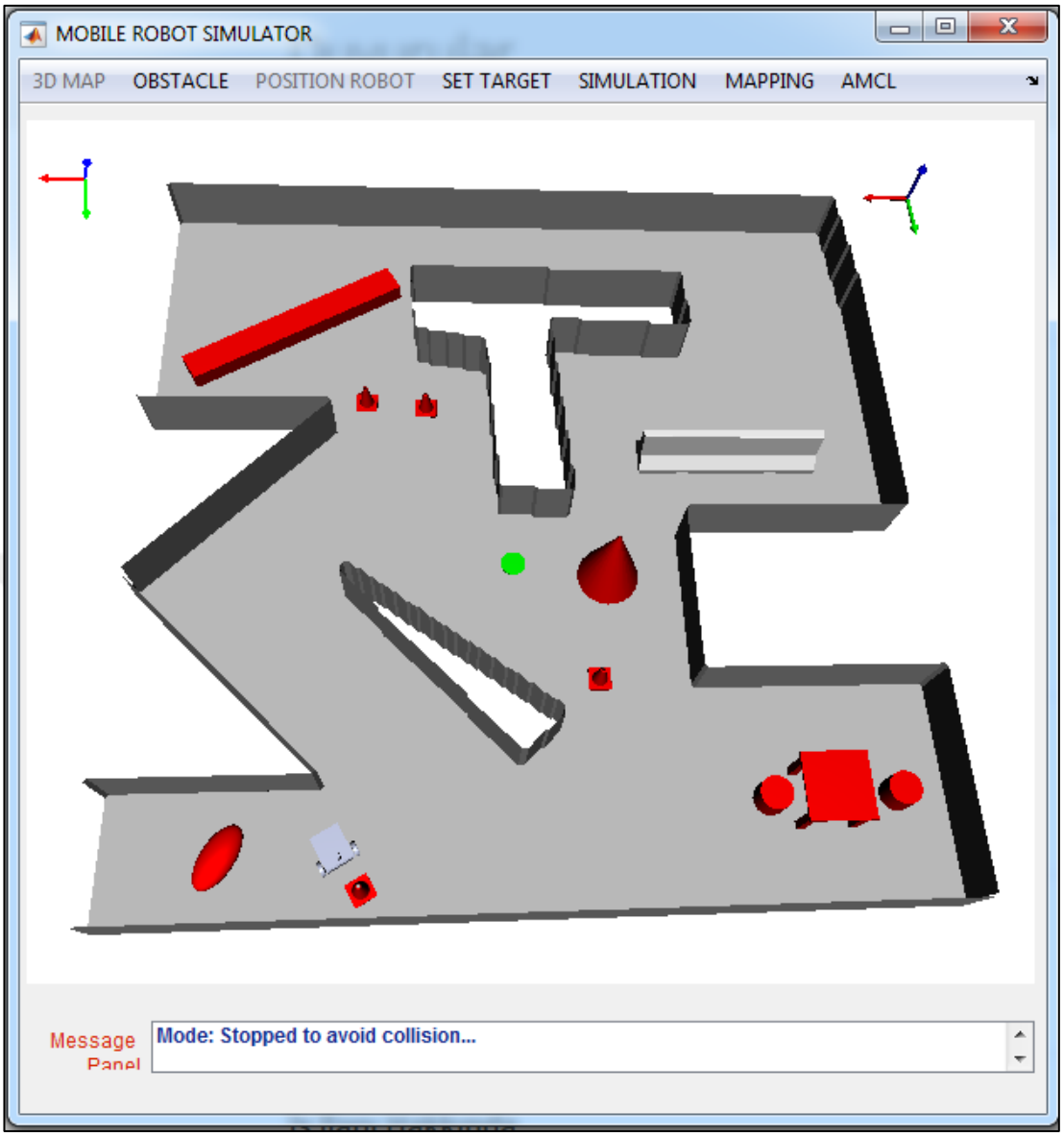
Uygulamada karşılaşılan çok sayıda problemin gerçek zaman koşullarında giderilmesi zaman ve enerji maliyeti yüksek bir iştir. Gerçek dünya koşullarına yakın seçenekler sunan benzetim arayüzleri üzerinde yapılan testler ve bu testlerde elde edilen bulgular gerçek dünya deneylerine ön hazırlık olarak destek olmaktadır. Benzetim arayüzleri üzerinde fiziksel dış etkenlerden daha çok metotlar üzerinde iyileştirmeler veya köklü değişikliler yapmak mümkündür. Tez kapsamında geliştirilen mobil robot platformunun kinematik yapısı baz alınarak geliştirilen benzetim arayüzü üzerinde, yol planlama, güvenli navigasyon ve Kalman filtresi gibi metotlar test edilmiş, gerçek dünya deneylerine tatmin edici düzeylerde eşdeğer sonuçlar alınmıştır. Modellenebilen diğer kinematik robot türlerine de uyarlanabilen arayüzde 2 veya 3 boyutlu statik ve dinamik haritalar kurgulanabilmektedir. Tez kapsamında çalışılan iç mekânın yüksek hassasiyetli lazer ölçüm cihazları ile ölçülerek çıkarılan 2 ve 3 boyutlu haritaları varsayılan olarak eklenmiştir. Bu haritanın dışında çok sayıda deneysel harita, labirent ve zorlayıcı parkurlar tanımlanmış, hareketli hedefler ve engeller altında geliştirilen metotların hangi çıktıları verdiği gözlemlenmiştir. Dinamik yol planlama, çarpışma engelleme ve LIDAR verilerini işleme gibi kritik metotlar geliştirilen benzetim arayüzü üzerinde test edilerek fiziksel platforma entegre edilmiştir. Mobil robot platformunun çalışması sırasında uzak PC üzerinden takip edilmesi ve LIDAR verilerinin takip edilmesi de geliştirilen arayüz ile sağlanmaktadır. Fiziksel robot ve çalıştığı harita eş zamanlı olarak arayüz üzerinde gerçek zamanlı olarak koşturulabilmekte veya sanal mobil robot sanal harita üzerinde çevrim dışı olarak çalıştırılabilmektedir. Benzetim arayüzü üzerinde LMS100 LIDAR sensörü modellenerek sanal bir LIDAR arayüzü de oluşturulmuştur. Bu model üzerinde sensöre ait tarama açısı, çözünürlüğü ve maksimum ölçüm limiti belirlenebilmektedir. Bu sayede engel algılama, açıklık tespiti, hareket kestirimi, çarpışma engelleme ve diğer LIDAR tabanlı navigasyon metotları çevrimdışı olarak geliştirilebilmektedir. Tanımlanan engel kütüphanesinde

sık kullanılan objeler 3 boyutlu olarak tanımlanmıştır. Bunların dışında koni, dikdörtgen prizma, küre, yamuk gibi temel geometrik cisimler ölçeklendirilebilir biçimde haritalara eklenip silinebilmektedir. Engeller ve haritayı oluşturan duvar gibi yapılar bilinen statik veya bilinmeyen dinamik engeller olarak tanımlanabilmektedir. Mobil robot modellenirken tekerlek ve LIDAR yerleşimi belirlenebilmektedir. Geliştirilen benzetim arayüzü bir sanal gerçeklik arayüzü ve LIDAR arayüzünden (LA) oluşmaktadır. VR arayüzü üzerinde harita, mobil robot, engeller, planlanan yol ve izlenen yol bilgileri global eksende yer almaktadır. LA üzerinde ise mobil robot, anlık engel profili, açıklık düzeni, lokal hedef, çarpışma noktası ve güvenli lokal hedef konumları lokal koordinat sistemine göre izlenebilmektedir. VR arayüzü üzerinde kullanıcı, tanımlanmış tüm kontrolleri yapabilmektedir. Harita yükleme, engel kütüphanesine erişilerek engel ekleyip silme, robot başlangıç konumu belirleme, global hedef belirleme, mesaj panelini görüntüleme, benzetimi başlatma, duraklatma, durdurma ve duraklatıp bilinmeyen engel ekleme, harita çıkarma, AMCL (Adaptive Monte Carlo Localization) eklentisini çalıştırma gibi işlemlerin tamamı bu arayüzde tanımlanmıştır. Kullanıcının bu pencereden erişebildiği engel kütüphanesi alt menüsünde küp, silindir, koni, küre ve masa şekilleri tanımlıdır. Tüm şekiller üç eksende bağımsız olarak ölçeklendirilebilmekte ve istenilen açıda haritaya konumlandırılabilmektedir. Benzetim sürerken duraklatıp engel ekleme ve kaldırma işlemi ile bilinmeyen bir engel profili çizilebilmektedir. Bu işlem fare ile sürükleme yolu ile gerçekleştirilebileceği gibi konsol üzerinden de ilgili komut girilerek gerçekleştirilebilmektedir. VR arayüzü üzerinde oluşturulan bir harita ve engel senaryosu kaydedilip daha sonra kullanılmak üzere saklanabilmektedir.

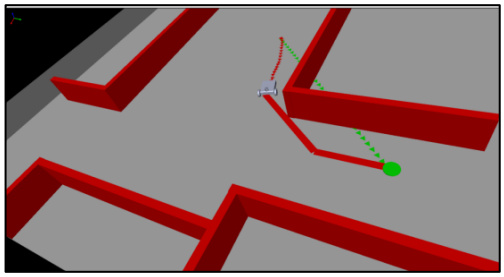



Şekil 5.1. Engel listesi arayüzü

Engel kütüphanesi ve VR arayüzü sırasıyla Şekil 5.1 ve Şekil 5.2'de gösterilmiştir.



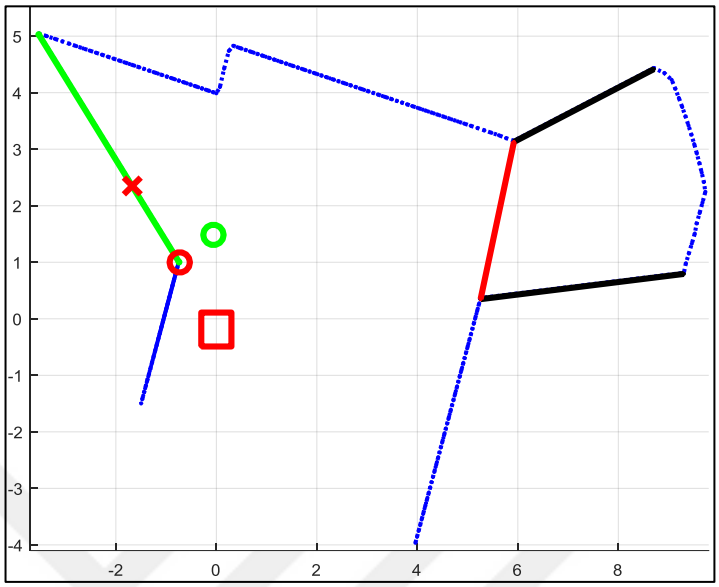

Şekil 5.2. VR arayüzü



Şekil 5.3. Örnek VR arayüzü

LA üzerinde anlık olarak LIDAR ölçümleri gözlemlenebilmektedir. Sensörün robot üzerinde konumlandırıldığı konuma bağlı olarak engellerin ölçüm noktaları bu

ekranda yatay kesit olarak konumlanmaktadır. Örnek bir VR arayüzü ve anlık LA görüntüsü sırasıyla Şekil 5.3 ve Şekil 5.4'te verilmiştir.




Şekil 5.4. Örnek LA penceresi

Tablo 5.1. VR ve LA sembollerinin açıklamaları

| Sembol | Monitör | Tanımı |
|---|---|---|
| | VR | Mobil robot |
| | VR | Bilinen engeller |
| | VR | Bilinmeyen engeller |
| | VR | Statik yol |
| | VR | İzlenen (dinamik) yol |
| | VR | Kestirilen yol |
| | VR | Global hedef |
| | LA | Engel ölçümleri |
| | LA | Çarpışma noktası |
| X | LA | Lokal hedef |
| | LA | Güvenli lokal hedef |
| | LA | Tespit edilen açıklık |
| | LA | Filtrelenmiş açıklık |
| | LA | Optimal açıklık |
| | LA | Mobil robot |

Gri renkle gösterilen zemin, duvar ve diğer nesneler bilinen haritaya ait objeler, kırmızı renkli duvarlar ise önceden bilinmeyen engelleri göstermektedir. Yeşil renkli statik yol, bilinmeyen engeller ihmal edilerek planladığı için engellerle kesişmektedir. Kırmızı renkli kesikli yol dinamik olarak izlenen ve kırmızı sürekli

yol ise güncellenen statik koşullar altında yeniden planlanan yol olarak verilmektedir. Bu anlık durum için LA görüntüsü Şekil 5.4'te gösterilmiştir.

Tablo 5.2. VR arayüz mesajları ve açıklamaları

| Mesaj | Tanımı |
|---|---|
| Navigating the robot | Navigasyon başarılı bir şekilde başladı ve robot global hedefe doğru ilerlemektedir. |
| The mobile robot successfully reached the target | Robot global hedefe belli bir tolerans bandında ulaşmıştır. |
| No obstacles on the path, heading directly towards the target | Robot global hedefe güvenli bir koridor boyunca ilerlemektedir. |
| Robot is on collision course | Çarpışma noktası tespit edilmiştir. Çarpışma engelleme fonksiyonu aktive edilmiştir. |
| Cannot assign a local target | Hiçbir açıklık bulunamamıştır. Lokal minimum durumundan çıkış moduna geçilmiştir. |
| Target is unreachable | Verilen hedefe kurtarma modlarının uygulanmasına rağmen erişilememektedir. Hedef çevresi tamamen kapalıdır. |

Tablo 5.3. Fonksiyonların ortak giriş-çıkış parametreleri

| Parametre | Tanımı |
|---|---|
| Imap_BW | Serbest ve statik engellenmiş bölgeleri içeren hücre tipi global harita |
| Ireal_BW | Dinamik ve statik engelleri ortak olarak içeren hücre tipi global harita |
| robotPosition | Mobil robotun global koordinat sistemine göre konumu |
| distMtrx | Verilen statik haritanın hücresel maliyet matrisi |
| LIDAR | LIDAR tarafından okunan uzaklık ölçüm vektörü |
| lidarRes | LIDAR açısal ölçüm çözünürlüğü (0.5o veya 1o) |
| start, target | Robotun başlangıç ve hedef konum vektörleri |
| KalmanFilter | Durum kestirimi, giriş, anlık ölçüm, proses, ölçüm ve kovaryansı parametrelerini içeren bir yapı |
| SystemModel | Durum ve giriş vektörlerini saklayan yapı |
| safeGaps | Robotun geçebileceği açıklıklar |
| OptimalGap | Üzerinde optimal lokal hedefin tanımlandığı açıklık |
| OptimalGapDist | Optimal açıklığın genişliği |
| gapDist | Tespit edilen açıklıkların genişlik dizisi |
| localTarget | Lokal hedefin global koordinat sistemindeki 2 boyutlu konum vektörü |
| localTarget_upd | Güvenli manevra noktasının konum vektörü |
| dist2Collision | Kritik bir mesafenin altındaki engellerden en yakınına olan mesafe |
| Vl | Sol tekerlek hızı (in cm/sn) |
| Vr | Sağ tekerlek hızı (in cm/sn) |

Verilen LA penceresinde sanal robot, anlık olarak tespit edilen açıklıklar, revize edilmiş açıklıklar, optimal açıklık, lokal hedef, çarpışma noktası, güvenli manevra noktası ve engeller görülmektedir. Eksenlerdeki ölçü birimleri metre türündendir. VR arayüzünde ve LA penceresinde kullanılan tüm simgeler ve açıklamaları Tablo 5.1'de verilmiştir. VR arayüzü üzerinde bulunan mesaj panelinde navigasyon sırasında oluşan tanımlı durumlar kullanıcıya bildirilmektedir. Bu mesajlar ve tanımları Tablo 5.2'de verilmiştir. Benzetim arayüzü üzerinde tanımlı çok sayıda fonksiyon ve alt fonksiyon bulunmaktadır. Bu bölümde sık kullanılan ve metotlarla doğrudan ilgili olan fonksiyonlar verilecektir. Bu fonksiyonlarda kullanılan ortak giriş çıkış parametreleri Tablo 5.3'te verilmiştir.

Sık kullanılan fonksiyonlar maddeler halinde verilmiştir:

- [LIDAR] = acquireLIDARdata (robotPosition): Verilen robot konumuna göre sanal LIDAR ölçüm çıktısı veren fonksiyon.
- [safeGaps, gapDist] = FindObstacleFreeGaps (robotPosition, LIDAR): Engeller arasından robotun geçebileceği aralıkları hesaplayan fonksiyon. Açıklıklar engel köşelerini birleştiren doğru parçaları olarak tanımlanmaktadır.
- distMtrx = fcn_dist_trans (Ireal_BW, target): Verilen statik harita ve hedef noktası için mesafe dönüşüm matrisini hesaplayan fonksiyon.
- [localTarget, OptimalGap, OptimalGapDist] = SetLocalTarget (safeGaps, gapDist, distMtrx, robotPosition, LIDAR): Mobil robot konumu, açıklık konfigürasyonu ve mesafe dönüşüm matrisi girişleri için optimal lokal hedef hesaplayan fonksiyon.
- [Vr, Vl] = SpeedController (robotPosition, localTarget, dist2Collision): Lokal hedefe ulaşmak için geren sağ ve sol tekerlek hızlarını hesaplayan fonksiyon. Hedef konumu, robot konum vektörü ve çarpışma noktasının konumu giriş olarak verilir.
- posEst = LocalizeRobot (): Mobil robotun konumunu veren fonksiyon.
- kalman_filter (KalmanFilter): Kalman filtresinin uygulayan fonksiyon.
- system_model (SystemModel): Kinematik model kullanılarak mobil robotun konumunu hesaplayan fonksiyon.
- displayPath (xp, yp, type): Planlanan (type:1), kestirilen (type:2), takip edilen (type:3) veya diğer bir yol (type:4) görselinin VR arayüzü üzerinde gösterilmesini sağlayan fonksiyon.

- deletePath (type): Belirlenen tipteki bir yolun VR arayüzü üzerinden silinmesini sağlayan fonksiyon.
- [localTarget_upd] = CollisionAvoidance (robotPosition, localTarget, OptimalGap, safeGaps, gapDist, OptimalGapDist, LIDAR): Lokal hedefe yönelim sırasında olası bir çarpışma noktası olup olmadığını kontrol eden fonksiyon. Bu tip bir engelin var olması durumunda güvenli manevra noktası tanımlanır.
- [objNo] = insertObject (pos, scale, strObj): Verilen bir konuma bir engel eklemek için kullanılan fonksiyon. 'pos' vektörü engele ait yatay, düşey konumları ve açısal oryantasyonu kapsamaktadır. strObj parametresi alanına 'Box', 'Cone', Cylinder', 'Sphere' veya 'Table' parametreleri girilerek objenin kütüphanede tanımlı türü belirlenebilir. scale parametresi 3 eksende boyut ölçeği olarak 0 ile 1 arasında tanımlanmalıdır. Her objeye objNo parametresinde saklanan benzersiz bir indeks atanır. Bu indeks kullanılarak ilgili obje daha sonra silinebilir veya konumu değiştirilebilir.
- deleteObject(objNo): 'objNo' parametresi ile belirtilen indeksli objeyi silen fonksiyon.
- moveObject(objNo, pos): Verilen indeksteki objenin konumunu belirleyen fonksiyon.
- setGlobalTarget(pos): Global hedef konumunun belirlenmesi için çağırılan fonksiyon. 'pos' vektörü dinamik olarak tanımlanarak hareketli hedef takibi uygulaması yapılabilmektedir.
- NAV_tangentBug(): Navigasyon algoritması olarak Tangent bug algoritmasının seçimini yapan fonksiyon.
- NAV_dynaBug(): Varsayılan navigasyon algoritması olarak tez kapsamında geliştirilen dinamik yol planlama ve güvenli navigasyon metodunu belirleyen fonksiyon.
- NAV_customCode(): Kullanıcılar kendi özgün yol planlama ve navigasyon metotlarını geliştirmek için gerekli tanımlama ve işlemleri bu fonksiyonun altında gerçekleştirmelidirler.

## 5.2.Nokta Stabilizasyonu Problemi

Mobil robot uygulamalarında robotun hedef konumunun yanı sıra duruş açısı da önem arz edebilmektedir. Belli bir konuma ulaşan robotun o konuma istenilen bir açı ile yaklaşması ve belli bir açı ile hedefte konumlanması istenebilmektedir. Bu durum noktasal stabilizasyon problemi olarak değerlendirilmektedir. Geliştirilen benzetim ortamında uygulanmak üzere bir noktasal stabilizasyon fonksiyonu önerilmiştir. Bu yöntem işlem yükü gözetildiğinde gerçek zamanlı uygulamalar için bu aşamada uygulanabilir olmadığı için yazılımsal olarak uygulanmıştır. Model kestirimli kontrol (Model predictive control - MPC) yaklaşımı [118] ve exact Euclidian distance transform (EEDT) [42] yönteminin hibrit bir uygulaması olarak geliştirilen yöntem diferansiyel sürüş kinematiğine uygun bir altyapıda tanımlanmıştır. Robot modeli kutupsal koordinatlarda ifade edilmiştir. EEDT algoritması ile planlanan stabilize yörüngenin kırılma noktaları (via points) hesaplanmıştır. MPC yaklaşımı uygulanırken holonomik olmayan kısıtlar altında, modelde tanımlanan giriş parametreleri optimize edilerek ardışık her kırılma noktası arasında kalan alt yörüngeler planlanmış, tüm alt yörüngelerin birleştirilmesi ile toplam bir stabilize yörünge elde edilmiştir.

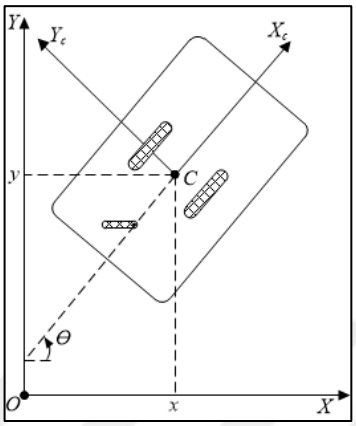
### 5.2.1. Kullanılan sistemin kinematik modeli

Kartezyen koordinatları Şekil 5.5'te verilen mobil robot şemasının matematiksel modeli Denklem (5.1)'de verilmiştir.

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= w \end{aligned} \tag{5.1}$$

Modeli verilen robotun tekerleklerinin yanal ve doğrusal kaymalar yapmadığı ve dönme hareketi yaptığı varsayılarak C noktası etrafında açısal hareket

yapabilmektedir. $X_c$ ve $Y_c$ robotun yerel eksenleri, {x,y,θ} elemanları ise ana koordinatlara göre konum ve duruş açısı olarak tanımlanmıştır. Model giriş parametreleri ise doğrusal ve açısal hızlardır (v, w). Modelin vektörel gösterimi Denklem (5.2)'de verilmiştir. Giriş vektörü u ile gösterilmiştir.




Şekil 5.5. Mobil robotun Kartezyen koordinat sistemi

$$\dot{x} = f(x,u) \tag{5.2}$$

Verilen modelin ayrık ifadesi Denklem (5.3)'te verilmiştir. Ayrık zaman terimi k ile gösterilmiştir.

$$x(k+1) = f_{dis}(x(k),u(k)) \tag{5.3}$$

Matematiksel model, kutupsal koordinatlarda {e, ψ, α} durumları ile ifade edilmiştir. Yatay ve düşey konumları ifade eden durumlar (x, y) birbirinden tam bağımsız olmadığı için Kartezyen koordinatlarda yapılan optimizasyon çözümleri optimal sonuçlar vermemektedir. Kalıcı durum hatalarına neden olan bu durum kutupsal koordinatlarda yapılan çözümleme ile giderilmiştir. Koordinat dönüşüm denklemleri Denklem (5.4)'te verilmiştir. $x_p$ ifadesi kutupsal koordinatlarda tanımlanan durum vektörüdür.

$$\begin{aligned} e &= \sqrt{x^2 + y^2} \\ \psi &= \arctan(y,x) \\ \alpha &= \theta - \psi \\ x_p &= [e \quad \psi \quad \alpha]^T \end{aligned} \tag{5.4}$$

Kutupsal koordinatlarda tanımlanan modele göre, robot orijine konumlandırılmak istendiğinde e ve ψ durumları 0'a yakınsarken x ve y durumları da eş zamanlı olarak 0'a yakınsamaktadır, ancak robotun açısının (θ) 0'a yakınsaması garanti edilemez. α durumunun tanımlanması ile açının da 0'a yakınsaması sağlanmıştır.

### 5.2.2. MPC yöntemi

Tanımlı bir kestirim derecesi altında modelin davranışı MPC yaklaşımı ile kestirilmiştir. Optimal giriş vektörü hesaplanarak bu vektörün birinci elemanı anlık optimal giriş olarak seçilmiştir. Bu yaklaşımda bir optimizasyon fonksiyonunu özyineli olarak minimize eden giriş parametreleri belirlenmektedir. Karesel yapıdaki optimizasyon fonksiyonu Denklem (5.5)'te verilmiştir.

$$\Phi_p(k) = \sum_{i=1}^{N} x_p{}^T(k+i\,|\,k)Qx_p(k+i\,|\,k) + u^T(k+i-1\,|\,k)Ru(k+1-i\,|\,k)) \tag{5.5}$$

Verilen denklemde N kestirim derecesini, Q ve R sırasıyla durum ve girişlerin ağırlık matrislerini, $x_p$ durum vektörünü göstermektedir. Öncelikle optimizasyon fonksiyonunu minimize eden girişleri ($u^*$) hesaplanmıştır. Bu girişlerin sayısı kestirim derecesi kadardır. Problemin tanımı Denklem (5.6)'da verilmiştir.

$$\bar{u}^* = \arg\min_{\bar{u}} \left\{\Phi_p(k)\right\} \tag{5.6}$$

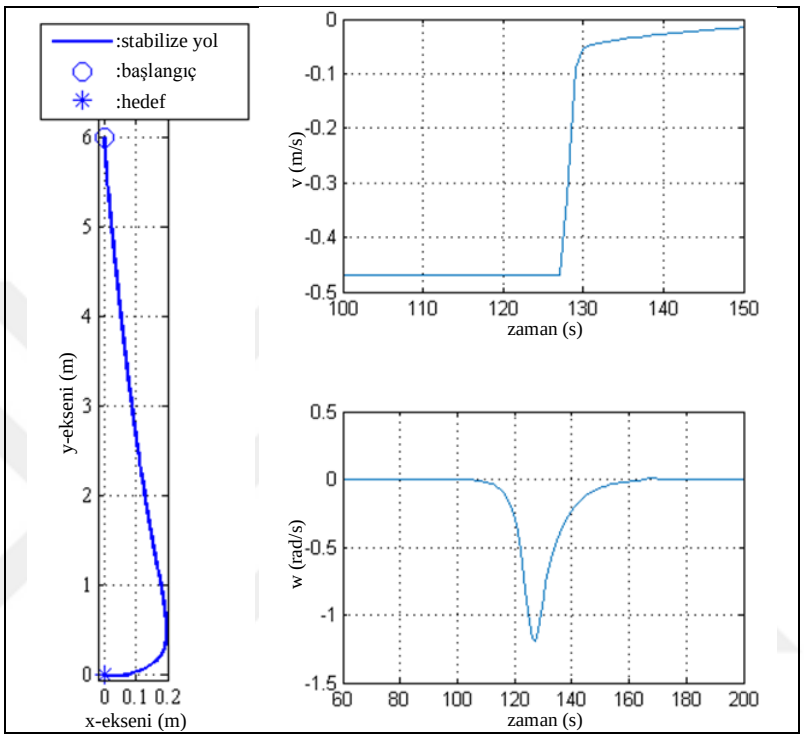
Örnek bir tanım olarak n(m|t) ifadesi n parametresinin t anında kestirilen değerinin m kestirim derecesindeki değeridir. Bu tanıma göre giriş kontrol dizisinin herhangi bir i anındaki değeri, t zamanından t+m zamanına kadar olan kestirimleri içermektedir. Denklem (5.6)'da verilen denklem her işlem adımında belli bir kestirim derecesi boyunca optimize edilerek çözüldüğünde herhangi bir i zamanı için optimal kontrol giriş dizisi Denklem (5.7)'de verilmiştir.

$$\bar{u}^* \cong \left\{u^*(i\,|\,i), u^*(i+1\,|\,i), u^*(i+2\,|\,i), \ldots, u^*(i+N-1\,|\,i)\right\} \tag{5.7}$$

Tahrik elemanlarının sınırları gözetildiğinde kontrol girişleri belli alt ve üst değerleri ile kısıtlanmıştır. Girişlerin kısıtlanması Denklem (5.8)'da gösterilmiştir. Durumlar üzerinde bir kısıtlama uygulanmamıştır. EEDT metodu tarafından belirlenen alt yörüngelerin oluşturduğu kırılma açıları ve yol uzunlukları, durumlar için dolaylı

kısıtları oluşturmaktadır. Karesel optimizasyon probleminin çözümü için MATLAB ortamında tanımlı ‘fmincon’ fonksiyonu kullanılmıştır.

$$\begin{bmatrix} I \\ -I \end{bmatrix} u(k+i\,|\,k) \le \begin{bmatrix} u_{max}(k) \\ -u_{min}(k) \end{bmatrix} \tag{5.8}$$



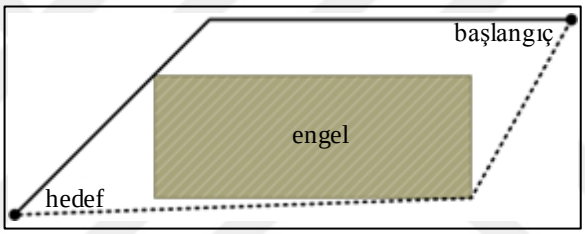

Şekil 5.6. Stabilize yörünge ve elde edilen girişler

Başlangıç ve hedef konum ve oryantasyon vektörleri sırasıyla $[0, 6, \pi/2]^T$ ve $[0, 0, 0]^T$ olarak verilen bir senaryoda giriş kısıtları $u_{min}$=[-0,5 m/s, -1,4 rad/s] ve $u_{max}$=[0,5 m/s, 1,4 rad/s] olarak belirlenmiştir. Konum bilgisi metre, oryantasyon derece birimi ile verilmiştir. Girişlerin zamana bağlı değişimi ve oluşan yörünge Şekil 5.6’da, verilmiştir. Oluşan yörünge kinematik modele ve girişlerde uygulanan kısıtlara uyumlu olarak elde edilmektedir. Mobil robotun hedef konumuna kalıcı durum hatası olmadan konumlandığı görülmektedir.

### 5.2.3. EEDT yöntemi

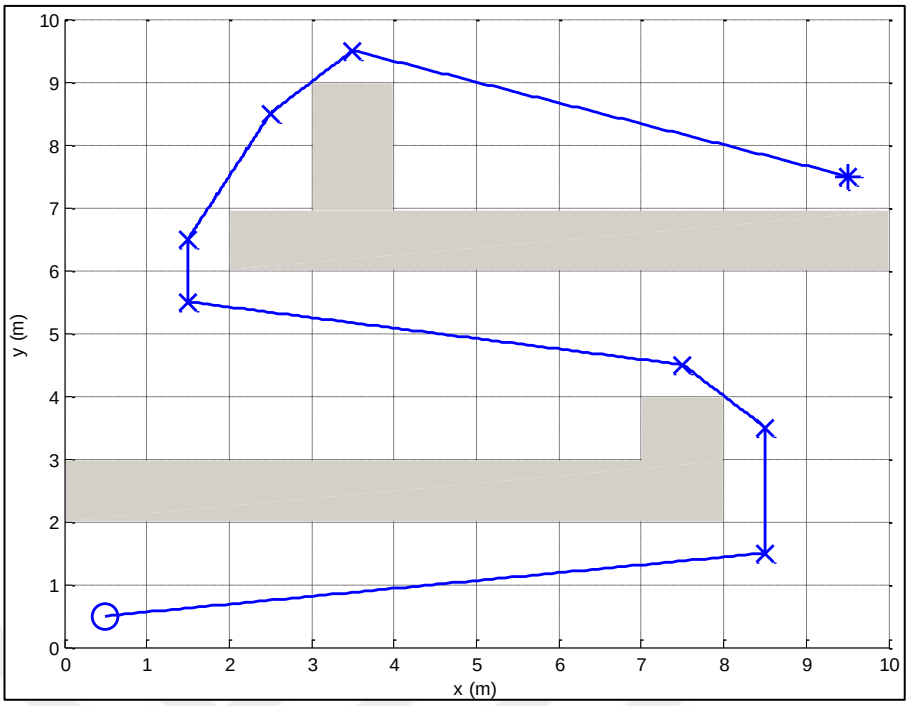
Engel bulunmayan ortamlarda MPC yöntemi uygulanarak yörünge planlama işlemi, iki nokta arasında durum kısıtlaması olmayan, girişlerin kısıtlandırıldığı bir optimizasyon olarak tanımlanmaktadır. Engellerin olduğu bir haritada başlangıç ve

hedef arasında bulunan engellerden kaynaklı durumların kısıtlanmasını gerektirecek bir optimizasyon problemi ortaya çıkmaktadır. Bu kısıtlama matematiksel olarak yapılabilmektedir. Ancak işlem yükünü artıran bu yaklaşım yerine kısıtlanmış problem, kısıtlanmamış alt optimizasyon problemlerine indirgenmiştir. Alt problemleri oluşturan her yörünge, EEDT yöntemi ile elde edilmiştir. EEDT yöntemi Jarvis yaklaşımı [36] üzerine inşa edilmiş bir statik yol planlama yaklaşımıdır [42]. Jarvis yaklaşımında bulunan hareketlerin $45^o$ ve katları ile sınırlandırılması, EEDT yönteminde kaldırılarak daha optimize bir yol uzunluğu elde edilmiştir. EEDT görünürlük tabanlı bir yöntemdir ve engelin görünür köşeleri üzerinde bir tarama gerçekleştirir. Her iki yöntemin aynı engel karşısında gösterdiği yol planlama karakteristiği görsel olarak Şekil 5.7'de verilmiştir. Noktalı çizgi ile gösterilen yol EEDT, düz çizgi ile gösterilen yol Jarvis metodu ile planlanan yoldur.

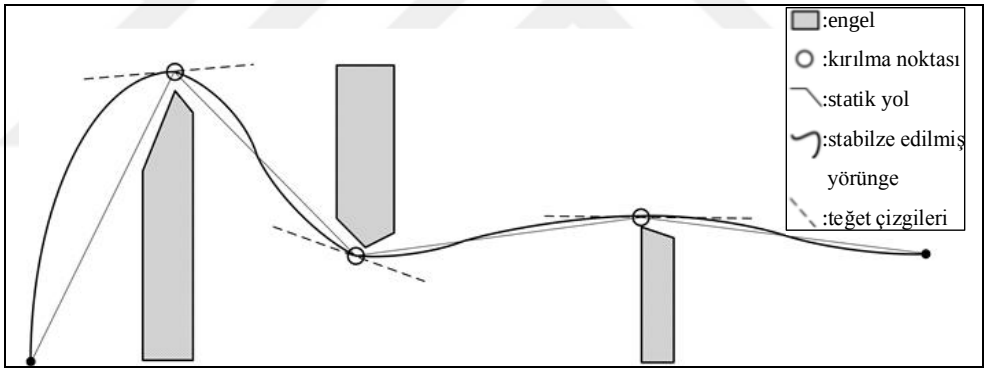



Şekil 5.7. EEDT ve Jarvis karşılaştırması

Örnek bir statik harita üzerinde EEDT yöntemi ile planlanan yol ve kırılma noktaları Şekil 5.8'de gösterilmiştir. Planlanan yol, alt segmentlere bölünerek her segmenti oluşturan uç noktaları arasında MPC yaklaşımı ile stabilize yörüngeler planlanmıştır. Planlanan yörüngeler için tanımlı başlangıç ve hedef oryantasyonları, alt segmentlerin kesişim noktalarındaki açılarının aritmetik ortalaması olarak hesaplanmıştır. Bir segmentin hedef oryantasyonu, ardışık sıradaki segmentin başlangıç oryantasyonu olarak belirlenerek sıralı her segment için bu işlem tekrarlanmıştır. Başlangıç noktasındaki $90^o$ oryantasyondan $0^o$ hedef oryantasyonuna konumlanması istenen bir statik yörüngenin alt yörüngeleri ve stabilize edilmiş toplam yörünge örneği Şekil 5.9'da verilmiştir. Teğet açıları her alt segmentin başlangıç ve hedef oryantasyonları olarak belirlenmiştir. Engeller belli bir tolerans bandında genişletilerek EEDT algoritmasına giriş olarak verilmiştir ve stabilizasyon aşamasında bu tolerans gözetilerek planlama yapılmıştır.




Şekil 5.8. EEDT yöntemi ile planlanan örnek yol

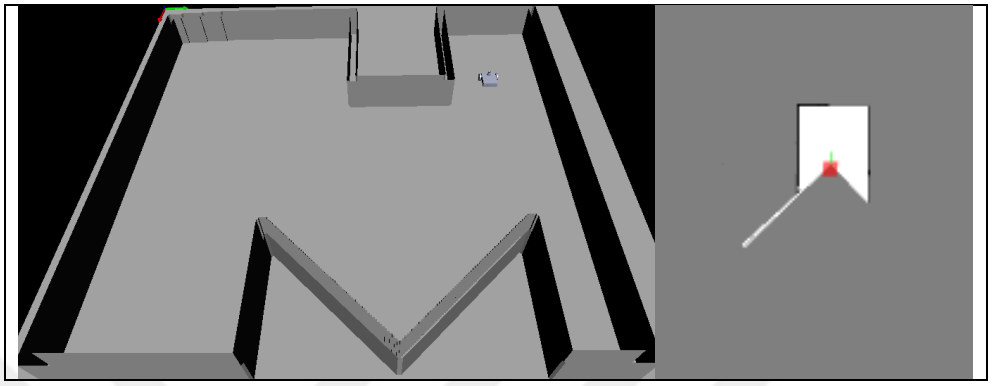



Şekil 5.9. Engellenmiş bir harita üzerinde stabilize edilmiş yörünge

## 5.3. Harita Çıkarma Eklentisi

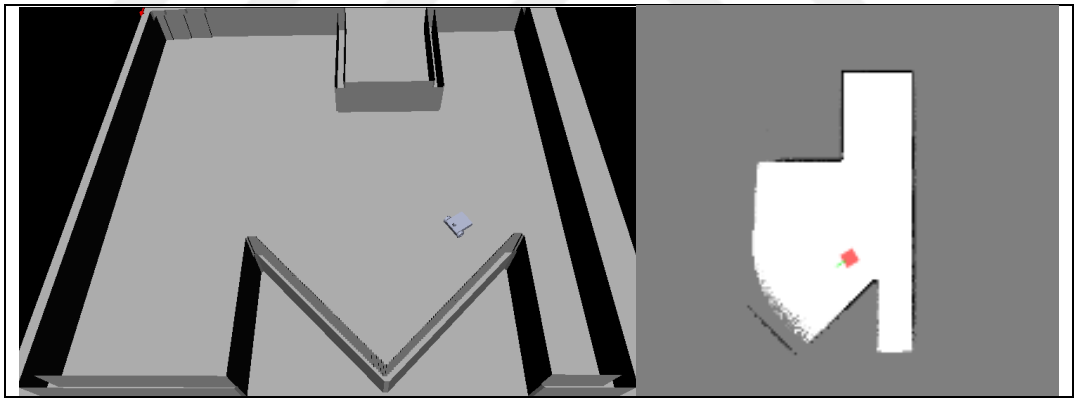
Benzetim arayüzünde ek olarak bir harita çıkarma fonksiyonu da eklenmiştir. Konumu bilinen engelleri kapsayan bir ortamın 2 boyutlu haritası bu eklenti ile elde edilebilmektedir. MATLAB programında bulunan mapping with known poses işlemine eşdeğer bir fonksiyon olarak değerlendirilebilir. Benzetim arayüzü üzerinde oluşturulan 3 boyutlu bir ortamda robot rasgele bir başlangıç konumuna yerleştirilir. LIDAR ile donatılmış olan robot klavye ile kontrol edilerek, ortamda bulunan engellere ait yüzey kesitlerinin LIDAR görüş alanına girmesi sağlanmaktadır. Başlangıçta bilinmeyen boş imge olarak tanımlı harita, LIDAR görüş alanına giren

ölçümler ve robotun anlık konum ve açı bilgileri kullanılarak 2 boyutlu harita üzerine binary değerler atanarak olarak güncellenir. Engeller lojik 0, serbest alanlar lojik 1 ile görüntülenir. Henüz keşfedilmemiş alanlar ise gri renk tonu ile harita üzerinde bilinmeyen bölgeleri temsil etmektedir.



Şekil 5.10. Harita çıkarma işlemi başlangıç durumu

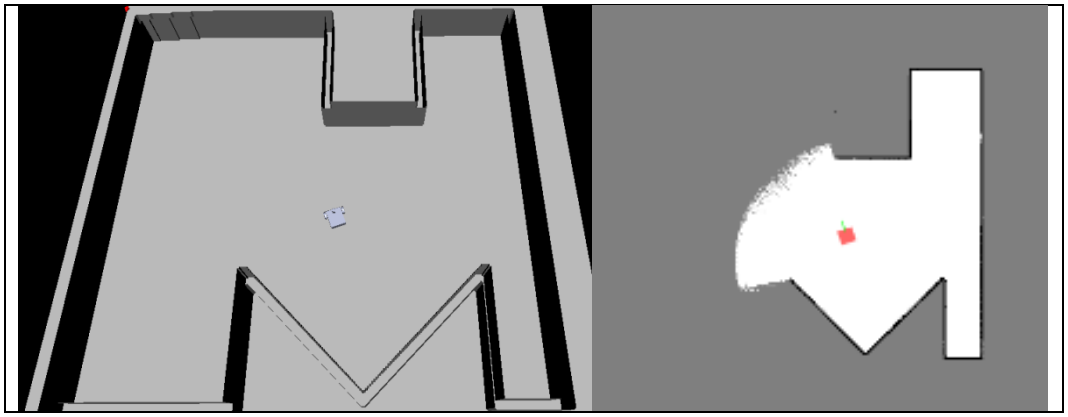
Örnek 3 boyutlu harita üzerinde çıkarılan harita ve ara adımları grafikler üzerinde Şekil 5.10 ile Şekil 5.15 arasında verilmiştir.



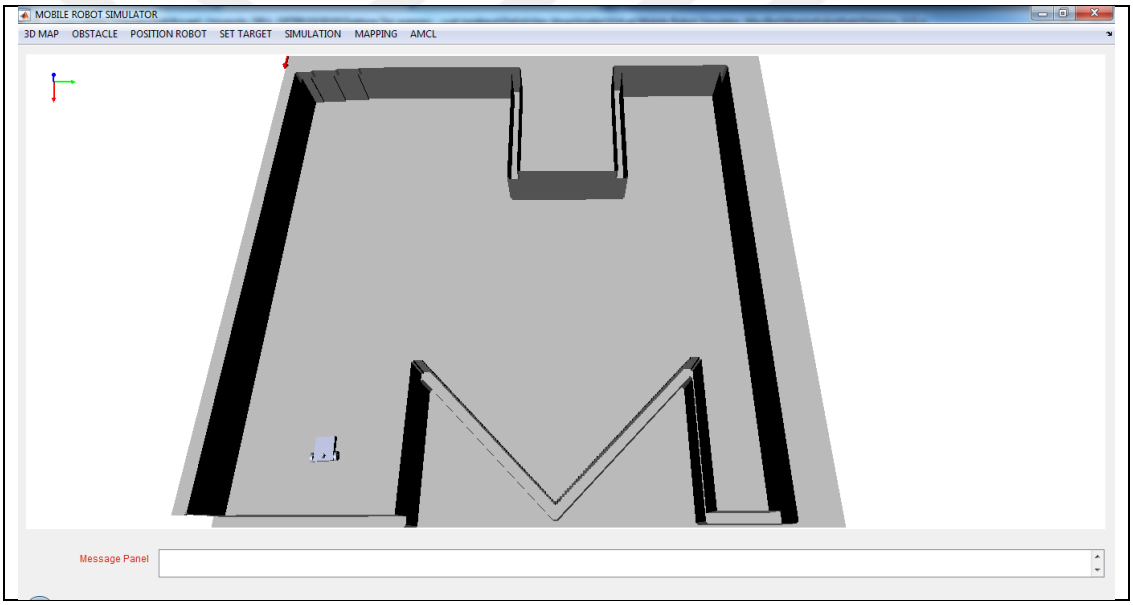
Şekil 5.11. Harita çıkarma işlemi ara adım-1

Rasgele bir başlangıç konumuna yerleştirilen robot, üzerinde bulunan LIDAR görüş limitleri dâhilinde ilk ölçümünü alarak boş haritayı doldurmaya başlamıştır. Şekil 5.10 üzerinde sol tarafta robotun başlangıç konumu ve sağ tarafta çıkarılan haritanın başlangıç durumu yer almaktadır. Rasgele yönlerde bir süre yönlendirilen robotun 3 boyutlu harita üzerindeki anlık görünümü ve çıkarılan haritanın güncel durumu Şekil 5.11 ve Şekil 5.12'de farklı zamanlar için gösterilmiştir. Her iki durum için sol tarafta 3 boyutlu harita üzerindeki pozisyon ve sağ tarafta çıkarılan 2 boyutlu haritanın anlık durumu verilmiştir. Haritası çıkarılmak istenen ortam tam olarak

tarandıktan sonra robotun bulunduğu konum ve elde edilen haritanın nihai durumu sırasıyla Şekil 5.13 ve Şekil 5.14'te verilmiştir.



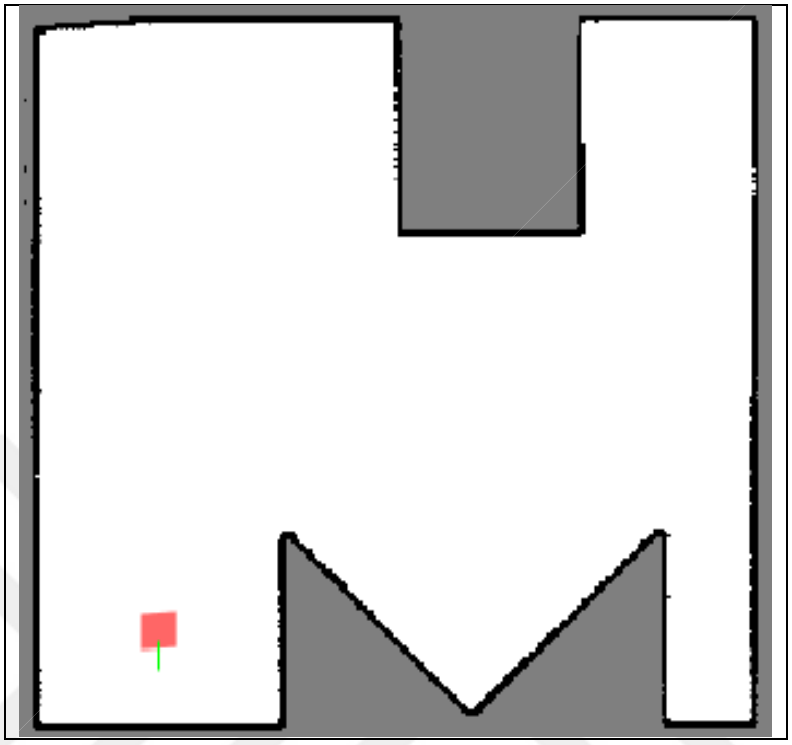
Şekil 5.12. Harita çıkarma işlemi ara adım-2



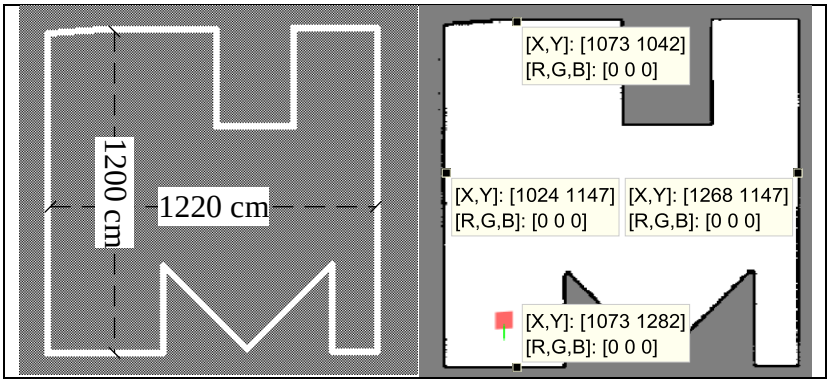

Şekil 5.13. Harita çıkarma işlemi son durum VR arayüzü

Elde edilen harita MATLAB "figure" penceresinde oluşturulmaktadır. Dolayısı ile kullanıcıya farklı kaydetme seçenekleri de sunulmaktadır. Harita üzerindeki mobil robot simgesi haritaya dâhil edilmezken, bilgi amaçlı olarak sadece haritanın oluşturulması aşamasında kullanıcıya gösterilmektedir. Bu simge üzerindeki yeşil renkli çizgi LIDAR bakış yönünü göstermektedir. Üzerinde çalışılan 3 boyutlu ortama ait izdüşüm resmi ve çıkarılan harita üzerinde 2 boyutlu bir boyut karşılaştırması yapıldığında çıkarılan haritanın oldukça başarılı sonuçlandığı görülmektedir. Şekil 5.15'te görülen izdüşüm resmi (solda) üzerinde verilen gerçek

ölçülere göre ortamın her iki eksende sahip olduğu maksimum genişlikler 1200 cm ve 1220 cm olarak verilmiştir.



Şekil 5.14. Çıkarılan harita




Şekil 5.15. Çıkarılan harita ve 3 boyutlu ortam ölçülerinin karşılaştırması

Çıkarılan harita üzerinde işaretlenen noktalar arasında aynı eksenler için piksel bazında hesaplanan genişlikler sırasıyla 240 ve 244 piksel olarak hesaplanmaktadır. Fiziksel harita ile uzamsal bölge arasında 1 piksel ≡ 5 x 5 $cm^2$ dönüşümü yapılmıştır. Bu durumda 240 piksel farkı uzunluk olarak 1200 cm, 244 piksel farkının ise 1220 cm olarak hesaplanmaktadır. Yapılan harita çıkarma işlemi temel olarak LIDAR koordinat sisteminde alınan mesafe vektörünün Kartezyen koordinatlarda, bilinen

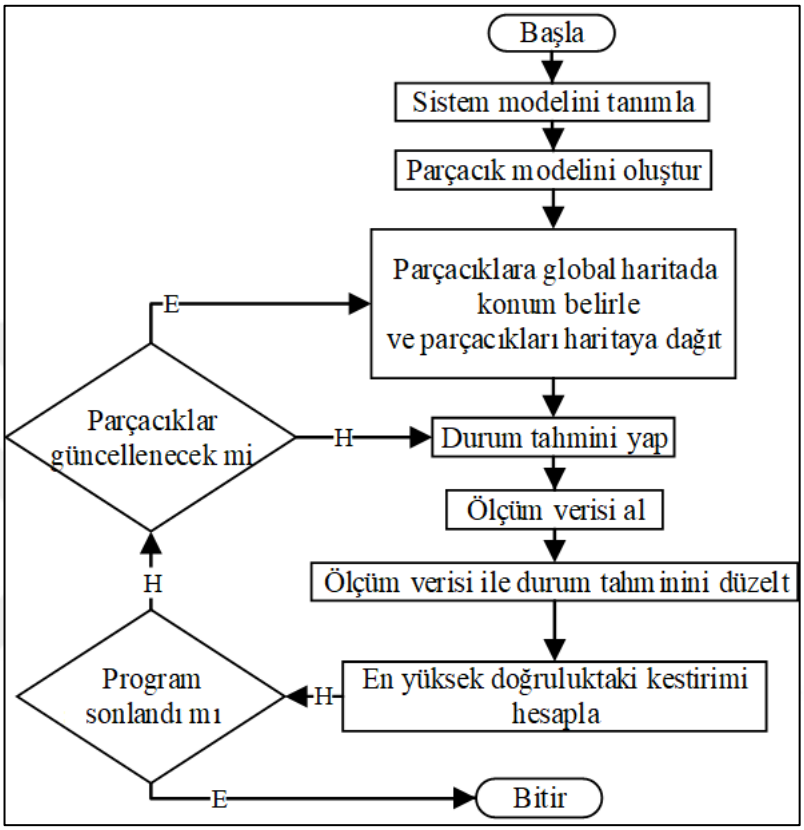
robot açısı ve konumu kullanılarak oluşturulan transformasyon matrisi ile işleme dahil edilmesidir. Maksimum ölçüm limiti altında oluşan ölçüm noktaları lojik 0, LIDAR konumu ile bu noktalar arasında kalan alanlar lojik 1, alınan ölçüm noktası veya maksimum ölçüm limitleri dışında kalan bölgeler de belirsiz etiketi ile sınıflandırılmıştır. Fiziksel boyutlar ile çıkarılan harita boyutlarının örtüşmesi nedeni ile harita çıkarma eklentisinin başarılı sonuçlar çıkardığı sonucuna varılmıştır.

### 5.4. AMCL Eklentisi

Tez kapsamında geliştirilen benzetim arayüzünde MATLAB 2016a sürümünde bulunan robotics toolbox altında tanımlı olan robotics.MonteCarloLocalization objesi, tanımlı mobil robot modeli ve LIDAR ölçüm modeline adapte edilmiştir. AMCL yöntemi temel olarak bilinen bir harita üzerinde olasılıksal bir yaklaşımla robotun konumunu hesaplamak için kullanılmaktadır. Arka planda parçacık filtresi koşturan yöntemde her bir parçacık robotun olası gerçek konum ve açısal duruşuna sahip durum vektörlerine karşılık gelmektedir. Bu yaklaşımda her bir parçacığın haritanın farklı bir bölümünü, modellenen bir sensör ile algıladığı varsayılır. Ayrıca bir odometre sensörü ile robotun hareketinin okunabildiği kabul edilir.

robotics.MonteCarloLocalization objesinden türetilen AMCL fonksiyonu giriş parametreleri olarak LIDAR tarama verisi ve odometre konum değerini almaktadır. LIDAR sensörü, SensorModel parametresi altında likelihoodFieldSensorModel objesinden türetilmektedir. LIDAR verisi sensör koordinatlarında tanımlı olacağı için global harita koordinatlarına dönüştürülmektedir. Bu dönüşüm için SensorModel.SensorPose vektörü tanımlanmalıdır. Odometre modelinden alınan konum bilgisi ile kestirilen konum arasında UpdateThresholds parametresi ile tanımlı eşikten daha yüksek bir mutlak fark oluşursa parçacıklar güncellenir. Yeni parçacıklar için durum kestirimi yapılarak algoritma devam ettirilir. Robotun hareket modeli MotionModel parametresi altında tanımlanmaktadır. Her parçacıktan okunan LIDAR verisinin gerçek okunan veri olma olasılığına bağlı olarak parçacıklara katsayılar atanır. ResamplingInterval özelliğine atanan değere bağlı olarak düşük ağırlıklı parçacıklar elenerek parçacıkların yüksek olasılıklı olanlara yoğunlaşması sağlanabilmektedir. Her adımda elde edilen çıkışlar kestirilen konum vektörü, kovaryans değerleri ve güncelleme bilgisidir. Her adım sonrasında geride kalan

parçacıklar yüksek ağırlıklardaki kümeler şeklinde ortaya çıkmaktadır. İşlemler bir döngü biçiminde tekrarlanarak parçacıkların tek bir noktaya yoğunlaşması sağlanmaktadır. AMCL konumlandırma işleminin akış şeması olarak gösterimi Şekil 5.16'da verilmiştir.



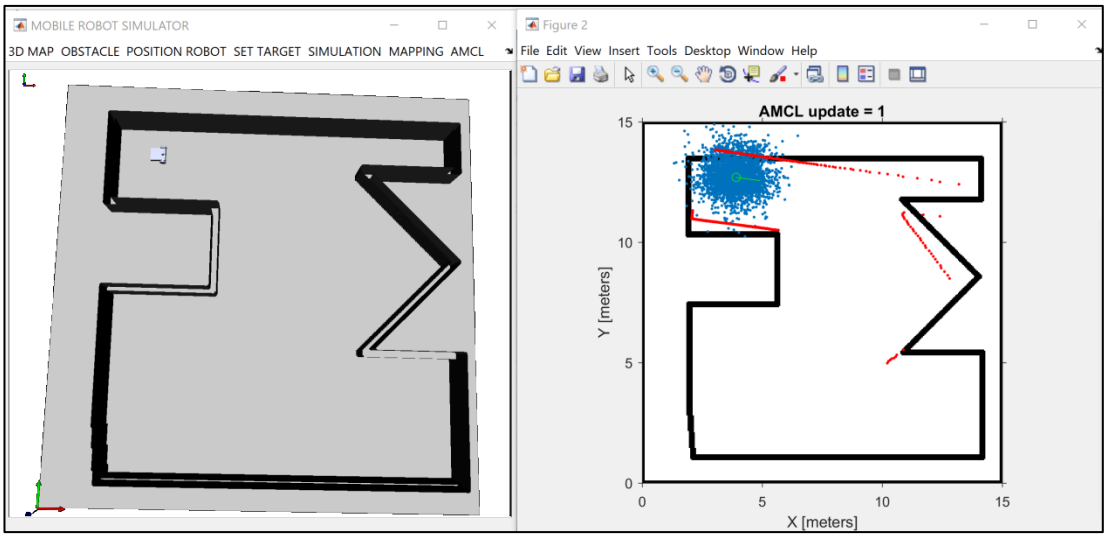

Şekil 5.16. AMCL konum kestirimi işlem adımları

AMCL tabanlı konumlandırma işlemi temel olarak tahmin ve düzeltme olarak iki aşamaya ayrılabilir. Tahmin aşamasında önceki durum vektörü ve odometre bilgisi kullanılarak sonraki durum hesaplanır. Tahmin aşamasında sistem modeli esas alınmaktadır. Düzeltme işlemi sensör ölçümü kullanılarak gerçekleştirilir. Alınan ölçümün parçacıklara ait olma koşullu olasılıkları hesaplanarak odometre verisi optimal parçacık etrafında doğrulanır. AMCL uygulamaları için MATLAB programında "robotics.MonteCarloLocalization" sınıfından türetilmiş bir MCL nesnesi tanımlanmıştır. Bu sınıfa ait temel özellikler GlobalLocalization, MotionModel, SensorModel, ResamplingInterval, UpdateThreshold olarak tanımlanmaktadır. Tanımlanmayan özelliklerin varsayılan değerleri kullanılır. MATLAB ortamında sadece tanımlamaların yer aldığı örnek kod parçası Şekil 5.17'de verilmiştir.

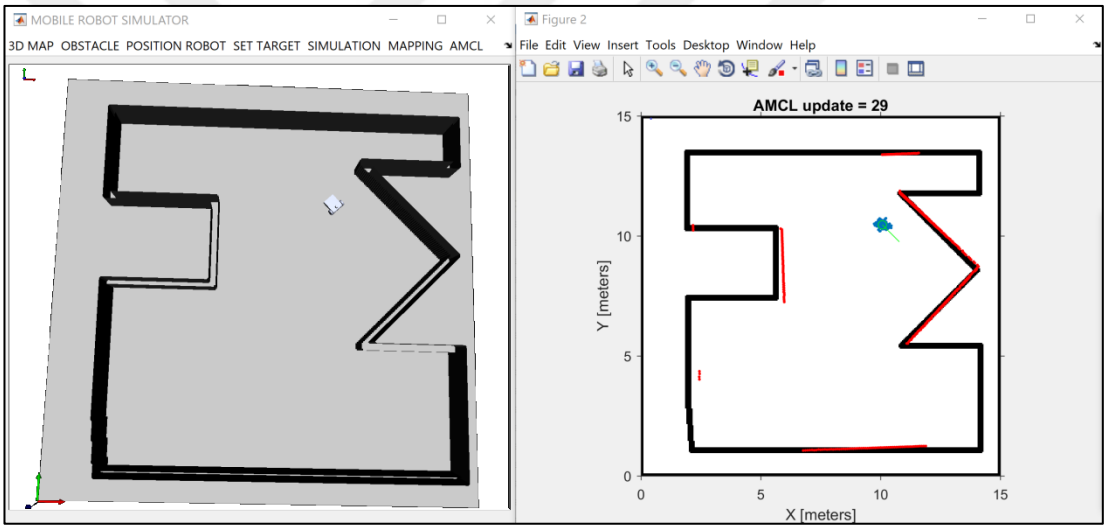
```
amcl = robotics.MonteCarloLocalization;
amcl.GlobalLocalization = false;
amcl.UpdateThresholds = [0.2 0.2 0.2];
rangeFinderModel = robotics.LikelihoodFieldSensorModel;
rangeFinderModel.SensorLimits = [0 maxLIDARrange];
rangeFinderModel.SensorPose = [0 0 0]
Map = robotics.BinaryOccupancyGrid(~,'grid');
rangeFinderModel.Map = Map;
amcl.SensorModel = rangeFinderModel;
odometryModel = robotics.OdometryMotionModel;
odometryModel.Noise = [0.2 0.2 0.2];
amcl.MotionModel = odometryModel;
amcl.ParticleLimits = [500 5000];
```

Şekil 5.17. AMCL nesnesi başlangıç değerleri için örnek tanımlama

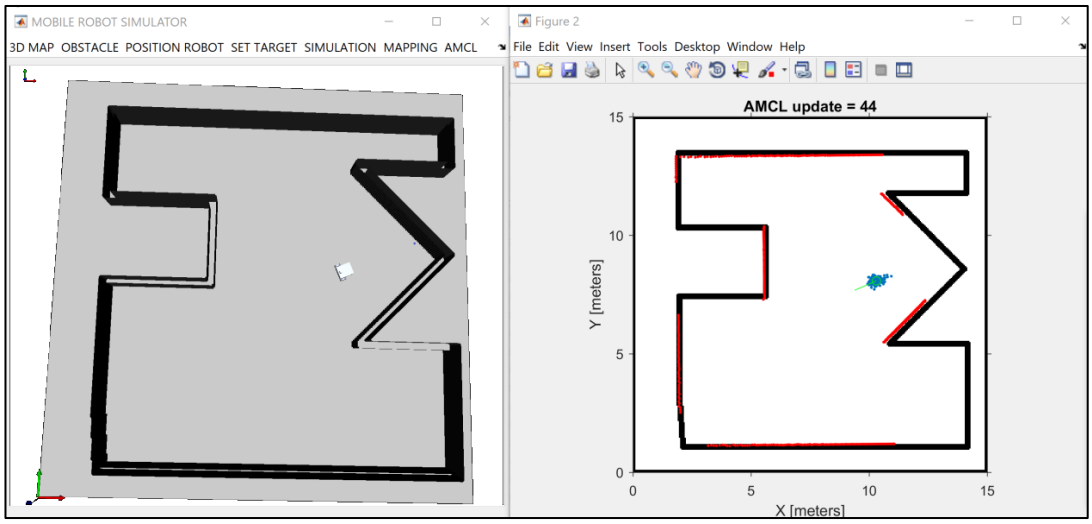
Burada "amcl" isimli nesneye ait GlobalLocalization özelliğine "false" değeri atanarak önceden bilinen yaklaşık bir başlangıç konumu olduğu varsayılmıştır. Başlangıç konumu, verilen hata kovaryans limitleri dâhilinde haritaya parçacıklar düzeyinde dağıtılmıştır. Parçacıkların mevcut dağılımının yeniden yapılması kriteri olarak x, y, θ durumlarından en az birisi 0,2 metre ve/veya radyan değişimi belirlenmiştir. Ölçüm verilerinin alınacağı sensör modeli alan tarama sensörlerini ifade eden sınıftan örneklenmiştir. Sensöre ait maksimum ve minimum ölçüm limitleri, sensörün robot koordinatlarında tanımlı lokal konumu ve statik ölçüm alacağı harita bilgisi sınıf özellikleri olarak sırasıyla tanımlanmıştır. Tanımlanan sensör paketi amcl nesnesine özellik olarak atanmıştır. Hareket tabanlı konum tahmini için odometre modeli belirlenmiş, model gürültüsü de vektörel olarak girilmiştir. AMCL yönteminde robotun başlangıç konumu hakkında belli bir sapma değeri limitlerinde ön bilgi verilmesi durumunda başlangıç dağılımı verilen konum etrafında bir dağılım sergilemektedir. Başlangıç tanımlamalarında "amcl.GlobalLocalization" özelliğine "false" değeri atanarak, benzetim ortamında gerçekleştirilen (kısmen bilinen başlangıç konumu için) konumlandırma deneyine ait çıktılar sıralı biçimde Şekil 5.18 ile Şekil 5.23 arasında verilmiştir.



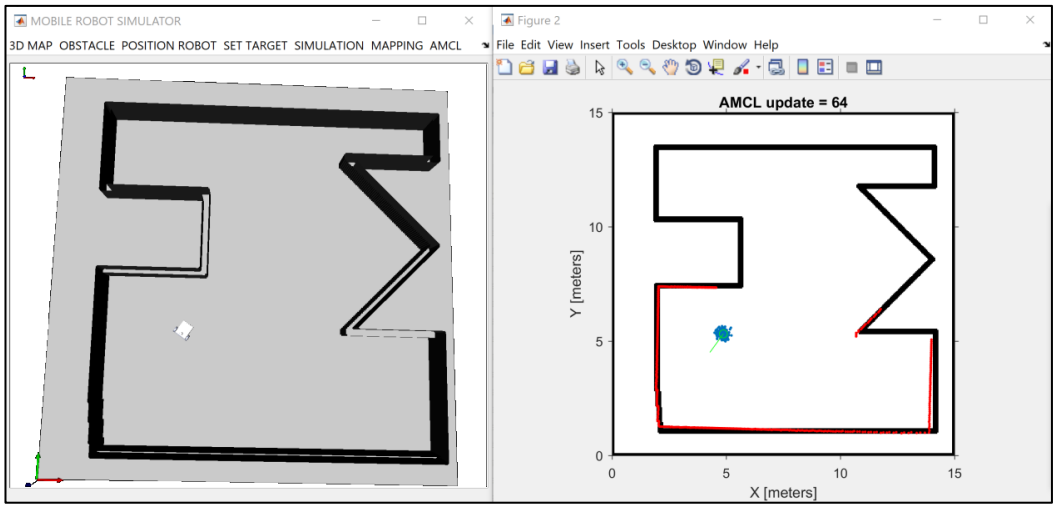

Şekil 5.18. Başlangıç konumu ve parçacıkların dağılımı



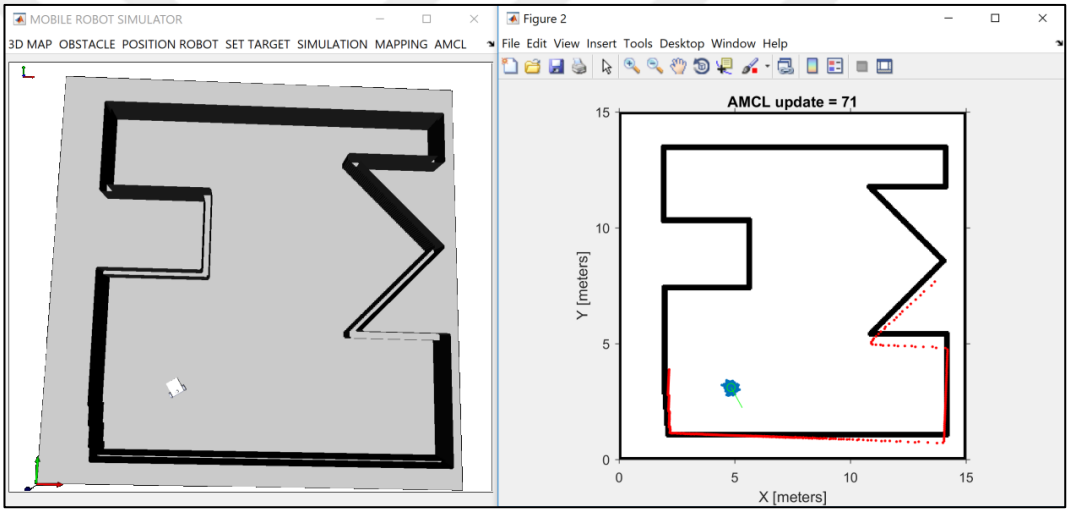

Şekil 5.19. 29. güncelleme adımı ve parçacıklar



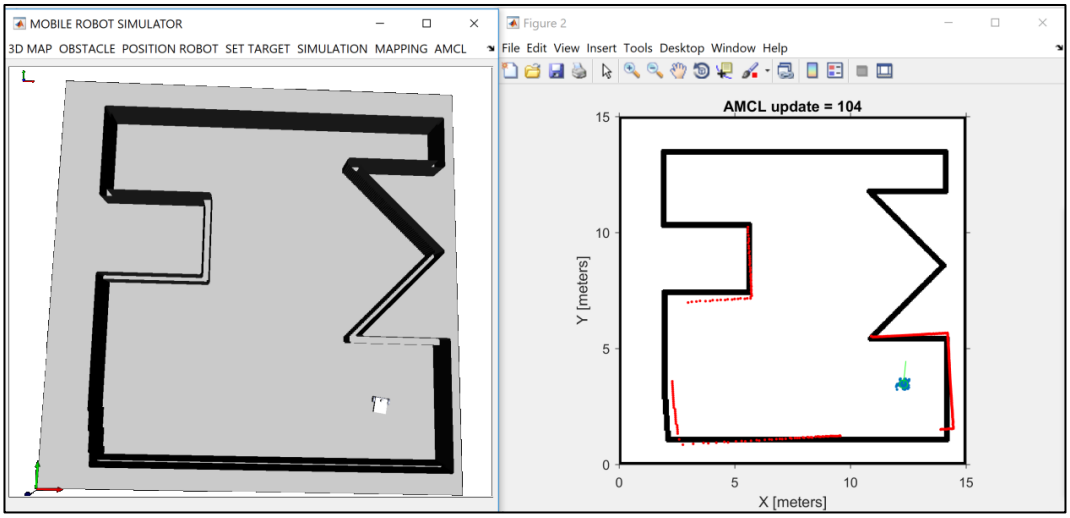

Şekil 5.20. 44. güncelleme adımı ve parçacıklar



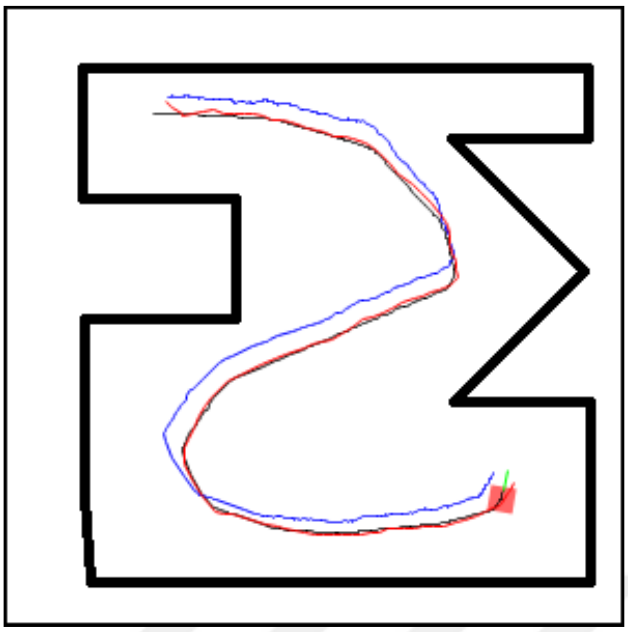

Şekil 5.21. 64. güncelleme adımı ve parçacıklar



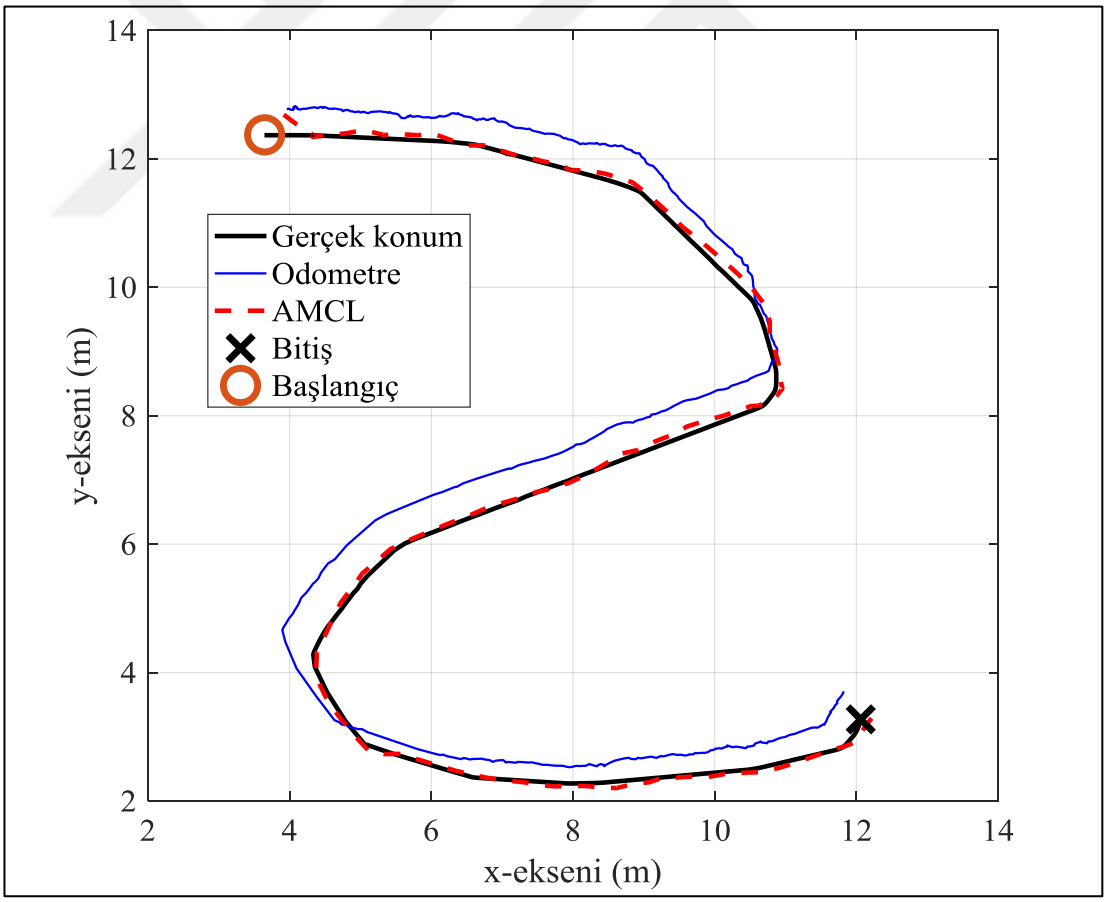

Şekil 5.22. 71. güncelleme adımı ve parçacıklar




Şekil 5.23. Sonuç adımı ve parçacıkların dağılımı



Şekil 5.24. Konumlandırma penceresi




Şekil 5.25. Hesaplanan konum verilerinin grafiği

AMCL eklentisi kapsamında oluşturulan mobil robotun anlık konum bilgisini gösteren pencere Şekil 24'te verilmiştir. Mavi renkle kodlanan yol odometre verisini,

kırmızı kodlanan yol AMCL kestirimini, siyah yol ise karşılaştırma amaçlı kullanılan gerçek konum bilgisini ifade etmektedir. Verilen uygulamada rasgele konumlara klavye kontrolü ile yönlendirilen sanal mobil robotun bilinen gerçek konumu, AMCL kestirimi ve odometre tahmini ile karşılaştırmalı olarak ortaya koyulmuştur. Konum verilerinin eksenler üzerinde görünümü Şekil 5.25'te verilmiştir.

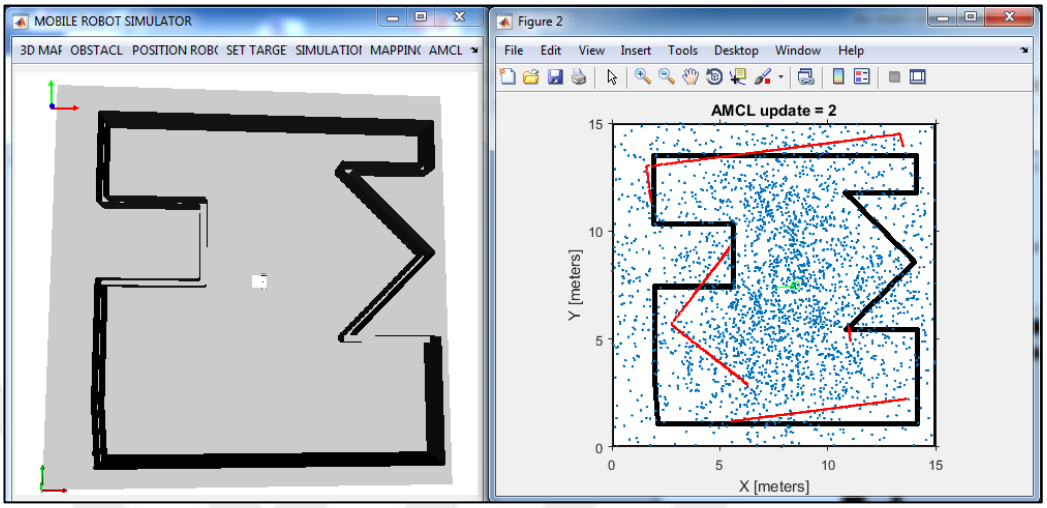
Gerçek konum bilgisi sadece önerilen metotların doğruluğunun test edilmesi aşamasında kullanılmıştır. Konumlandırma hatası Denklem (5.19)'da verilen ortalama karesel hatanın karekökü (RMSE) yaklaşımı ile hesaplanmıştır. Hata değerleri hesaplanırken kullanılan konum vektörleri aynı boyutlardadır ve denklemde kullanılan $x^o$ ve $x^g$ sırasıyla ölçüm sistemi değeri ve gerçek konum değerini göstermektedir. N parametresi ise vektörlerin eleman sayısıdır.

$$e_{rmse} = \sqrt{\frac{\sum_{n=1}^{N}\left(x_n^o - x_n^g\right)^2}{N}} \tag{5.9}$$

Gerçek konum matrisi ile AMCL konum matrisinin x-ekseni ve y-ekseni üzerinde hesaplanan RMSE değerleri sırası ile 17,3 cm ve 12,6 cm olarak hesaplanmıştır. Benzer yaklaşımla odometre konum tahmini matrisi ile gerçek konum verileri arasında x-ekseni ve y-ekseni boyutlarında hesaplanan RMSE büyüklükleri sırasıyla 25,2 cm ve 45,3 cm olarak bulunmuştur. AMCL konumlandırma yaklaşımı, uygulama sonucu elde edilen ortalama karesel hatanın karekökü değeri baz alındığında, salt model tabanlı konumlandırma yaklaşımına göre belirgin bir düzeyde yüksek doğruluğa sahiptir. Toplam çalışma alanı yaklaşık 8x12 $m^2$ olan bir yörünge üzerinde 20 cm'nin altında bir konumlandırma hatası odometre çıktısına göre oldukça tatmin edici bir sonuç olarak değerlendirilmiştir.

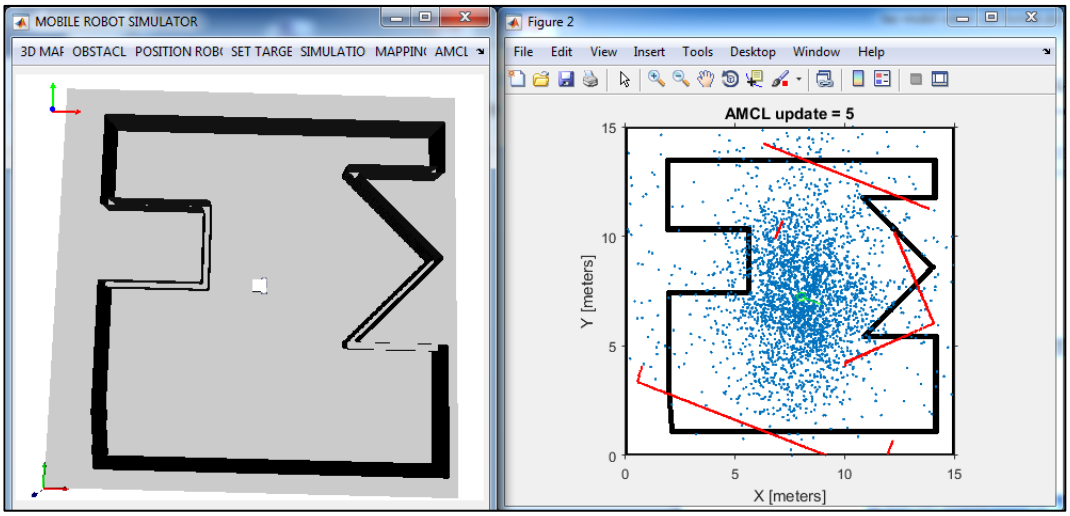
AMCL konumlandırma sistemine, başlangıç konumu hakkında bir ön bilgi verilmeden yapılan uygulamalarda "amcl.GlobalLocalization" özelliğine "true" değeri atanarak başlangıç konfigürasyonu gerçekleştirilmelidir. Bu durumda parçacıklar başlangıç durumunda tanımlı çalışma uzayı üzerinde uniform dağılım göstermektedir. Global konumlandırma yapılırken daha fazla parçacık ile robotun olası durumları modellenmektedir. Konum kestirimleri gerçek konuma yakınsadıkça parçacık sayısı ve buna bağlı olarak işlem yükü azalma eğilimi göstermektedir.

Ancak başlangıç durumunda konumlandırma doğruluğu oldukça düşük seviyelerde başladığı için genel konumlandırma performansının da düşük başladığı yorumu yapılabilir.

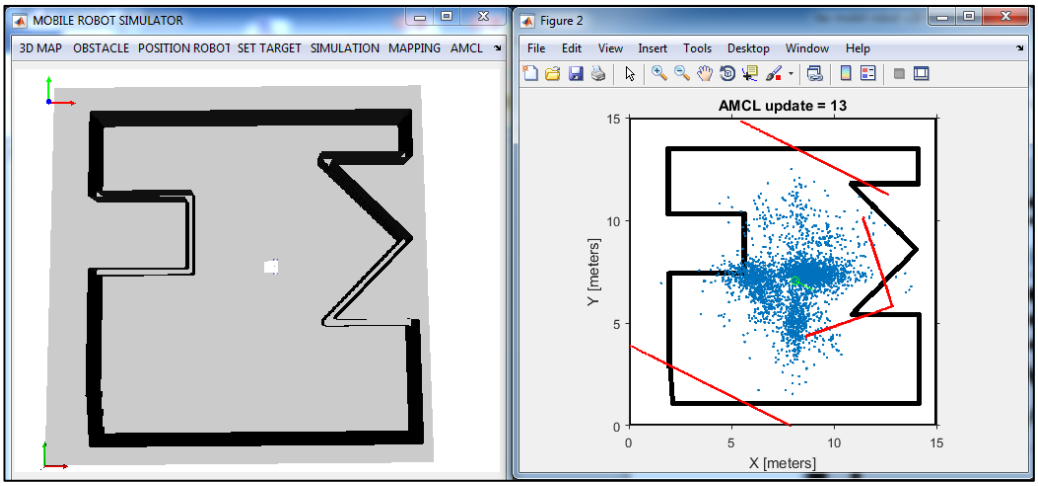



Şekil 5.26. Global konumlandırma başlangıç parçacık dağılımı

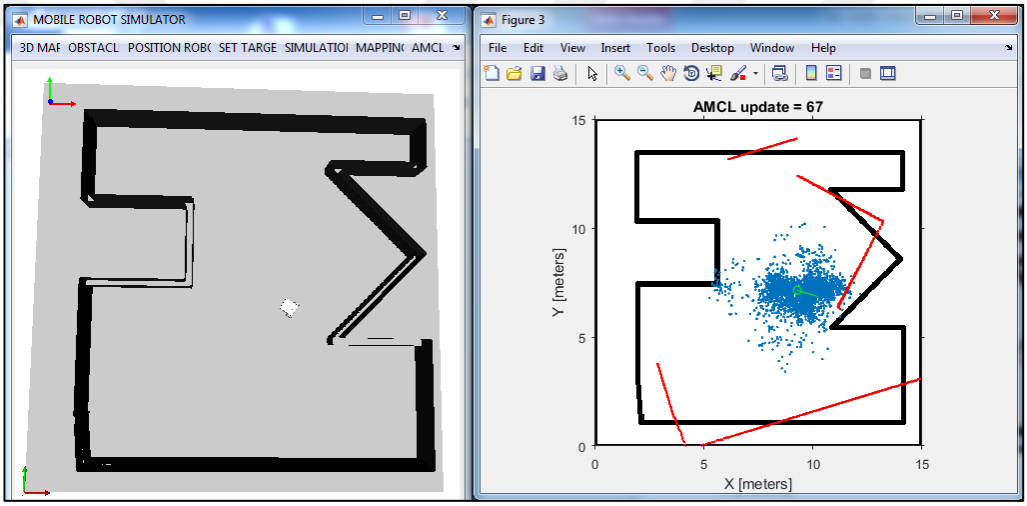
Global konumlandırma uygulaması için, başlangıç durumu parçacık dağılımı Şekil 5.26'da görülmektedir. Başlangıç dağılımından itibaren ilerleyen iterasyonlarda sensör ölçümlerinin doğruladığı parçacık konum tahminleri gerçek konum etrafında yoğunlaşmaya başlamıştır. Kestirilen konum bilgisi, parçacıkların ortalama konumu olarak kabul edilmiştir. İlerleyen adımlarda gerçekleşen iterasyonlarda parçacıkların dağılımları Şekil 5.27 ile Şekil 5.30 arasında sıralı olarak verilmiştir.



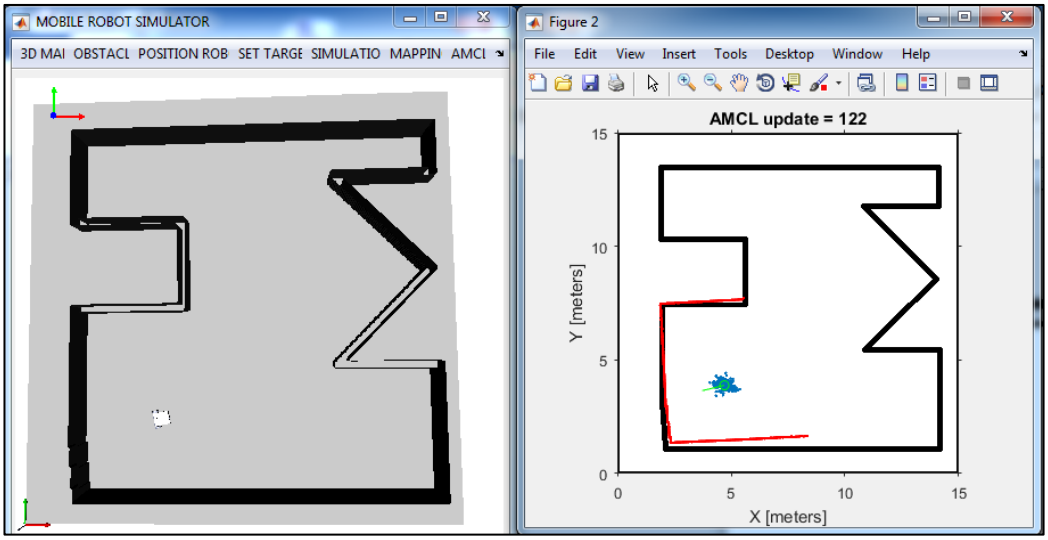

Şekil 5.27. Global konumlandırma 5. güncelleme parçacık dağılımı




Şekil 5.28. Global konumlandırma 13. güncelleme parçacık dağılımı




Şekil 5.29. Global konumlandırma 67. güncelleme parçacık dağılımı




Şekil 5.30. Global konumlandırma 122. güncelleme parçacık dağılımı

# 6. SONUÇLAR VE ÖNERİLER

## 6.1. Yol Planlama

### 6.1.1. Gerçek dünya deneyi

Deneysel sonuçlar fiziksel mobil robot platformu kullanılarak elde edilmiştir. İç mekân olarak Kocaeli Üniversitesi Mühendislik Fakültesi Mekatronik Mühendisliği Bölümü 3. Kat koridoru kullanılmıştır. Konumlandırma sistemine ait gerekli donanımlar koridorun tavan bölümüne kurulmuş, iç mekânın teknik haritası 2 boyutlu kat krokisi olarak yol planlama şemasına dâhil edilmiştir. Deneyler süresince 10 kg yük altında tüm donanımlar (PC, LIDAR, kamera-projeksiyon grubu) aktif durumdayken, otonom modda yaklaşık 2 saat kesintisiz çalışılabildiği kaydedilmiştir. Uzaktan kumanda ile kullanımda ve yüksüz durumda ise 5 saat çalışılabildiği gözlemlenmiştir.

Yapılan uygulama sırasında uzak bilgisayar için hazırlanan bir MATLAB arayüzü kullanılmıştır. Arayüzün görünümü Şekil 6.1'de verilmiştir. Uzak bilgisayar ile mobil robot dâhili bilgisayarı arasındaki UDP haberleşmesi için kullanılan IP numarası kullanılarak robot ile bağlantı kurulmuştur. Hazırlanan arayüz üzerinde IP numarasının girileceği ve anlık bağlantı durumunun gözlenebileceği alanlar bulunmaktadır. Ayrıca, çalışılan iç mekânın 2 boyutlu statik haritası, konumlandırma sisteminden alınan gerçek zamanlı konum ve açısal pozisyon verileri, sağ ve sol tekerleklerin anlık hızları okunabilmektedir. Mobil robotun ulaşması istenilen konum fare komutu ile arayüz üzerindeki harita üzerinde tanımlanabilmektedir. Hedef belirlendikten sonra ilgili buton üzerinden otonom görev başlatılmaktadır. Mobil robotun harita üzerindeki konumu hem sayısal olarak hem de statik harita üzerinde görsel olarak takip edilebilmektedir. Öngörülemeyen durumlar sonucunda otonom modda oluşabilecek kontrol dışı hareketler, donanımsal sorunlar veya hareketli engel ve insan kaynaklı acil durumlara karşın arayüz üzerinde acil durdurma butonu tanımlanmıştır. Mobil robotu manuel modda kullanmak için ekstra kontroller

eklenmiştir. Sağ ve sol tekerlek hızlarını ve dönüş yönlerini belirlemek için tanımlanmış kontroller robotu manuel olarak kullanma imkânı sağlamaktadır.




Şekil 6.1. Uzak bilgisayar arayüzü

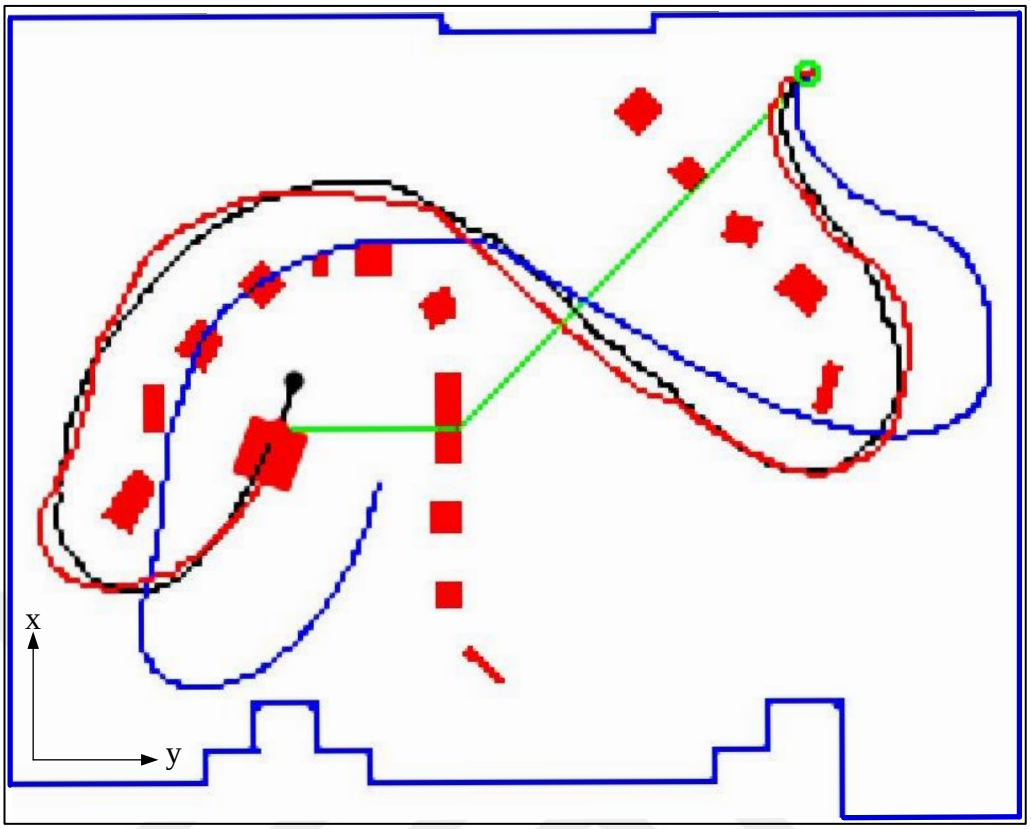
Yapılan deney gerçek zamanlı ve benzetim olmak üzere iki aşamada gerçekleştirilmiştir. Gerçek dünya koşullarında oluşturulan engel yerleşimi benzetim ortamında da oluşturularak geliştirilen yol planlama algoritmasının fiziksel robot ve sanal robot üzerinde oluşturduğu yörüngeler karşılaştırılmıştır. Bu test sonucunda her iki eksen için oluşan yörünge konum hataları benzetim arayüzünün doğruluğu için bir ölçek olarak değerlendirilmiştir. Ayrıca destekleyici bir konum belirleme yaklaşımı olan odometre yönteminin tek kaynak olarak fiziksel robot üzerinde kullanılması durumunda ortaya çıkan yörünge de tespit edilmiştir. Engelleri benzetim ortamına en az hata ile aktarabilmek için dikdörtgensel nesneler kullanılmış, köşe noktalarının gerçek dünya koordinatları yüksek hassasiyetli lazer mesafe ölçüm cihazı ile ölçülerek benzetim arayüzündeki haritaya aktarılmıştır. Benzetim ortamında imge olarak tanımlanan harita üzerinde 1 pikselin uzunluğu yatay ve düşey eksenler için 5 cm olarak belirlenmiştir. Robotun her 5 cm yer değiştirmesi benzetim ortamında 1 piksel değişime karşılık gelmektedir. Robotun başlangıç konumu ve açısal pozisyonu benzetim ortamında tanımlanarak benzetim uygulaması çalıştırılmış ve her iki deney süresince robotun anlık konumu kaydedilmiştir. EDT metodu tarafından oluşturulan statik yörünge dışında robotun izlediği dinamik yol

benzetim ve gerçek dünya deneylerinde belli farklılıklar göstermiştir. Engellerin iç mekan yerleşimi ve mobil robotun başlangıç durumundaki pozisyonu Şekil 6.2'de verilmiştir. Farklı boyut ve açılarda konumlandırılmış, önceden konumu verilmemiş olan engeller kullanılmıştır. Verilen hedef için noktasal stabilizasyon uygulanmamıştır. Deneyler sırasında acil durdurma gerekliliği oluşmamıştır.



Şekil 6.2. Gerçek zamanlı deney engel yerleşimi

Benzetim ortamında yapılan deney için elde edilen konum çıktısı ve gerçek deney sonuçları ortak şema olarak Şekil 6.3'te gösterilmiştir. Şekil üzerinde odometreden alınan konum verisi de gösterilmiştir. Şekil üzerinde kullanılan notasyon ve açıklamaları Tablo 6.1'de verilmiştir. Gerçek dünya deneyi ve benzetim çıktıları için 2 boyutlu harita üzerinde kat edilen yörüngenin x-ekseni ve y-ekseni bileşenleri sırasıyla Şekil 6.4 ve Şekil 6.5'te verilmiştir. Örnek sayıları yukarı-örnekleme (upsampling) ile eşitlenen yörünge bileşenleri arasında ortalama karesel hatanın karekökü (RMSE) değerleri Denklem (6.1)'de verilen formatta hesaplanmıştır. Denklemde kullanılan $x^b$ ve $x^g$ sırasıyla benzetim ve gerçek dünya deneylerinde alınan konum vektörleri; N ise vektörlerin eleman sayısıdır.




Şekil 6.3. Gerçek zamanlı deney ve benzetim arayüzü karşılaştırması

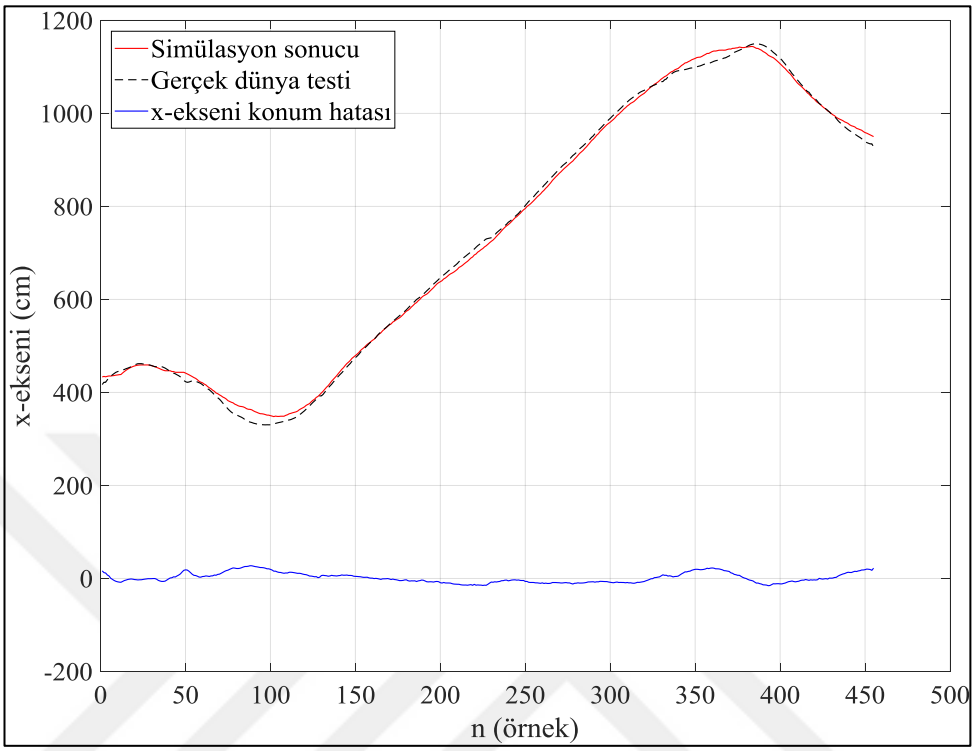
Tablo 6.1. 2 boyutlu benzetim arayüzü notasyonu

| Sembol | Tanımı |
|---|---|
| | Benzetim ortamında elde edilen yörünge |
| | Gerçek deney sonucunda elde edilen yörünge |
| | Odometre tarafından hesaplanan yörünge |
| | Statik yol |
| | Bilinen engeller |
| | Bilinmeyen engeller |
| | Mobil robot |

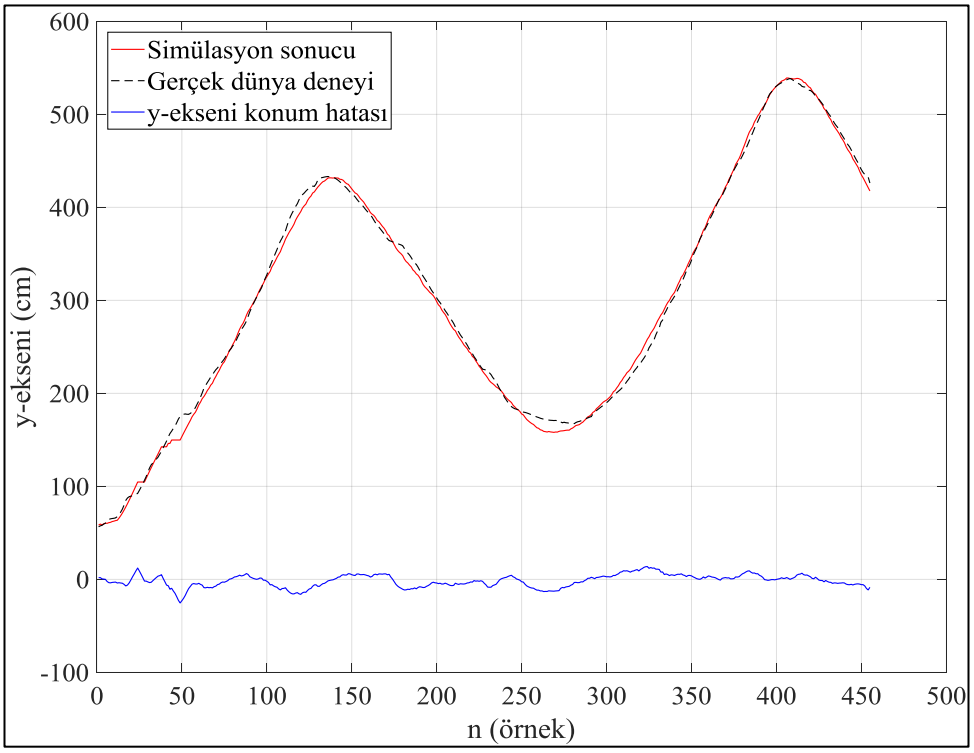
$$e_{rmse} = \sqrt{\frac{\sum_{n=1}^{N}\left(x_n^b - x_n^g\right)^2}{N}} \tag{6.1}$$

Denklem (6.1)'de verilen işlem sonucunda $e_{rmse}$ vektörü $\begin{bmatrix}10,9\\6,8\end{bmatrix}$ cm olarak elde edilmiştir. Aynı algoritmalar ve ortam koşulları için x-ekseninde 10,9 cm, y-ekseninde 6,8 cm büyüklüğünde bir konum farkı oluşmuştur. Ortalama pozisyon hatası (köşegen) 12,8 cm olup, bir pikselin 5 cm olduğu kabulüyle uzamsal bölgede yaklaşık 2 piksel hata ortaya çıkmıştır. Benzetim ile gerçek dünya deneyi arasındaki

bu fark, fiziksel kayıplar, engellerin konum ölçümü sırasında oluşan ölçüm hataları, konumlama farkları ve örnekleme kayıplarından kaynaklanmaktadır.



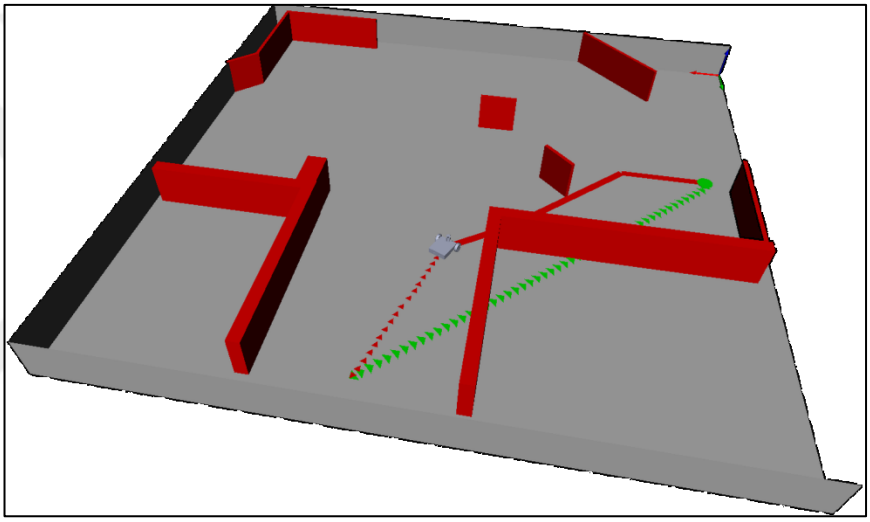

Şekil 6.4. Benzetim arayüzü x ekseni konum hatası




Şekil 6.5. Benzetim arayüzü y ekseni konum hatası

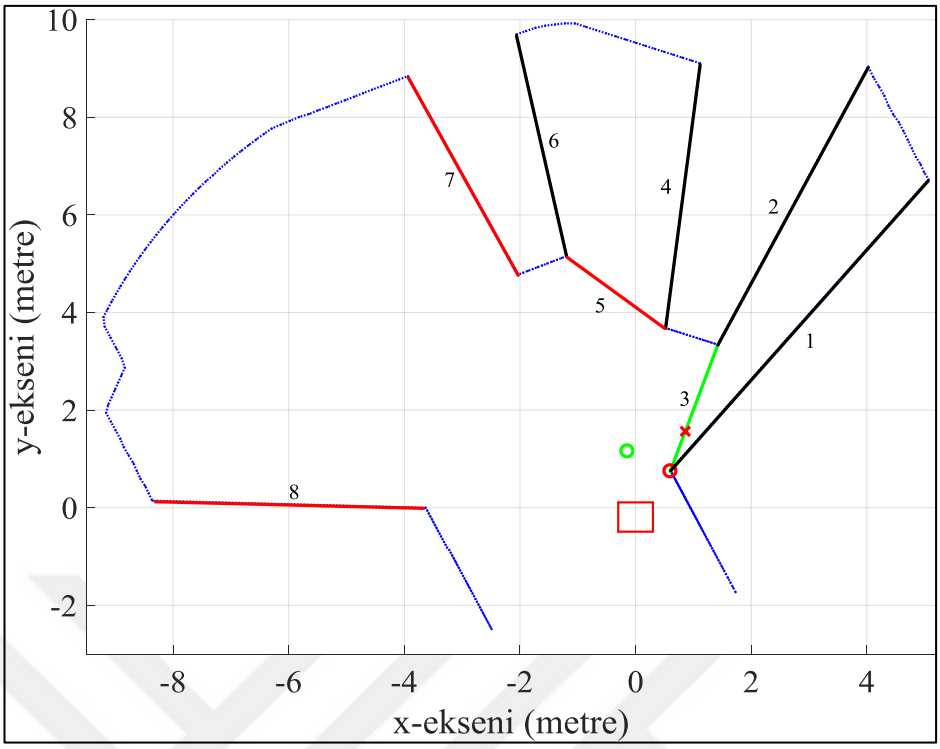
### 6.1.2. Açıklık tespiti ve optimal lokal hedef tespiti

Dinamik yol planlama metodu kapsamında önerilen açıklık tespiti, optimal lokal hedef tespiti ve çarpışma engelleme yaklaşımlarına ait çıktılar örnek bir senaryo üzerinde verilmiştir. VR arayüz görüntüsü ve anlık LA pencereleri sırasıyla Şekil 6.6 ve Şekil 6.7'de verilmiştir. Bu grafikler üzerinde yapılan indekslemeler ile bahsi geçen yöntemlerin ortaya koyduğu çıktılar değerlendirilmiştir. Benzetimi yapılan diferansiyel sürüş modelindeki mobil robot ortalama 50 cm/s hız ve 60 cm x 60 cm boyutlara sahiptir. LIDAR tarama menzili 10 m, tarama açısı ise $270^{o}$, açısal çözünürlüğü $1^{o}$ olarak belirlenmiştir.



Şekil 6.6. Açıklık tespiti çıktıları VR arayüzü

Verilen anlık duruma ait grafikler üzerinde açıklık tespiti algoritmasının ve çarpışma engelleme metodunun işleyişi gözlemlenebilmektedir. 1, 2, 4, 6, 7 ve 8 numaralı açıklıklar herhangi bir revizyona uğramayan ve ilk taramada elde edilen açıklıklardır. 1 ve 2 numaralı açıklıklar 3 numaralı açıklığa, 4 ve 6 numaralı açıklıklar 5 numaralı açıklığa indirgenerek revizyon işlemi gerçekleştirilmiştir. Nihai durumda 3, 5, 7 ve 8 numaralı açıklıklar üzerinde lokal hedef tespiti gerçekleştirilerek kırmızı renkli x simgesi ile gösterilen lokal hedef tespit edilmiştir. 5, 7 ve 8 numaralı açıklar elenerek robot 3 numaralı açıklığa yönlendirilmiştir. Çarpışma engelleme metodu tarafından kırmızı renkli "o" sembolü ile gösterilen çarpışma noktası tespit edilmiş, robot bu noktaya dik ve $d_{tol}$ mesafesi kadar uzaklıkta güvenli bir lokal hedef noktası (yeşil renkli o sembolü) ile gösterilen noktaya yönlendirilmiştir.




Şekil 6.7. Açıklık tespiti çıktıları LA görüntüsü

### 6.1.3. DWA ve önerilen dinamik metodun karşılaştırılması

DWA dinamik yol planlama uygulamaları için yaygın kullanıma sahip bir metottur. Global ve lokal düzeyde işlemler içeren metot yaygın kullanıma sahip birçok mobil robot yazılımında da kullanıcılara sunulmaktadır. Etkin lokal minimum çözümleri ve güçlü teorik Ar kaplanı olan bu yöntemin tez kapsamında geliştirilen dinamik yol planlama yaklaşımı ile ortak bir statik harita üzerinde yol planlama süreçleri karşılaştırılmıştır. Planlanan yol uzunluğu ve işlem süresi bakımından her iki yöntemin ortaya koyduğu sonuçlar incelenmiştir.

DWA algoritması lokal pencere sınırlarında tanımladığı dinamik maliyet haritalarını kullanır ve güncellenen her global harita için lokal statik yolu tekrar hesaplar. Bu durum işlem süreleri bakımından olumsuz bir etki ortaya çıkarmasına karşın, özellikle çok sayıda engelin rasgele ve parçalı olarak yerleştiği kompleks ortamlarda önerilen yönteme göre planlanan yol uzunluğu ve optimal çözüm bulma noktalarında daha etkili çözümler bulabilmektedir. Çok sayıda dağınık engellerin ortaya çıkardığı açıklık sayısı, labirent, koridor ve nispeten engeller arasında belirgin açıklıkların bulunduğu ortamlara göre daha yüksek olmaktadır. DWA metodunun global harita bilgisine sahip olması bakımından özellikle iç bükey engellerin bulunduğu

ortamlarda yerel minimumlara düşme durumu oluşmamaktadır. Buna karşın önerilen yöntemde harita gereksinimi duyulmaksızın yapılan iç mekân yol planlama işlemlerinde önceden bilinmeyen iç bükey engeller yerel minimumlara neden olmaktadır. Bu tip kapalı uçlu koridorlara girilmesi durumunda tanımlı kurtarma hareketleri uygulanarak global hedefe ulaşan alternatif yörüngeler planlanmıştır. Yerel minimum kaynağı olmayan ortamlar için her iki metot birbirine benzer özelliklere sahip yörüngeler planlamıştır.

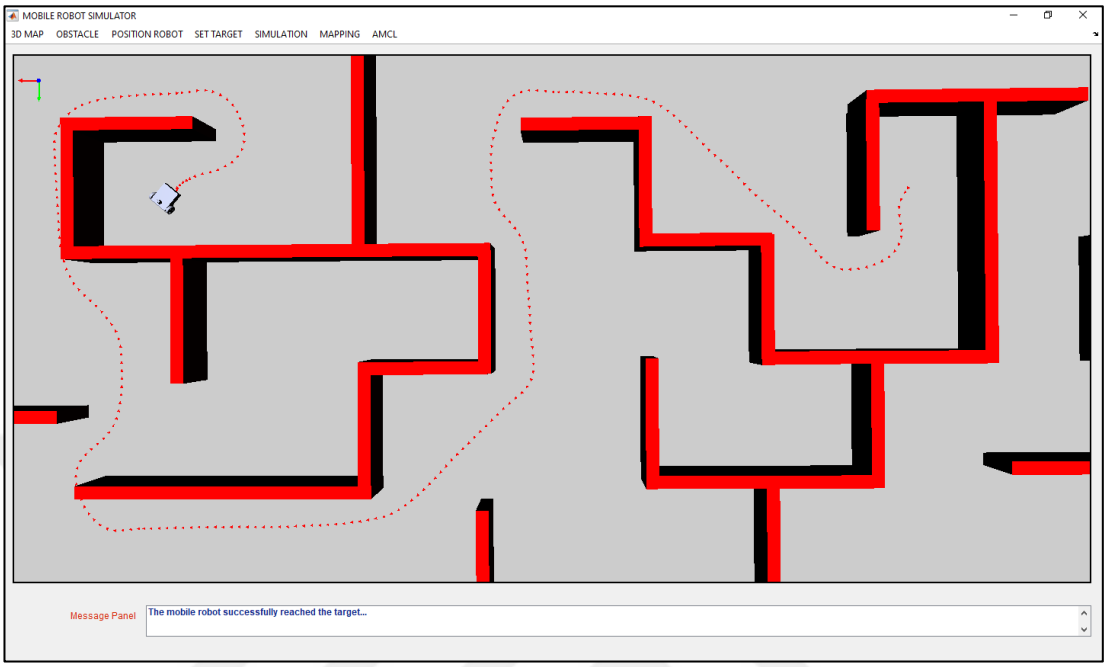
Başlangıç konumundan global hedefe yapılan navigasyon sürecinde hareket doğrultusunda iç bükey engeller ile karşılaşılmayan, labirent tipindeki bir harita üzerinde DWA ve önerilen dinamik yol planlama yöntemleri uygulanmıştır. DWA metodu global harita verisini kullanırken önerilen planlayıcı lokal düzeyde sensör verileri ile planlama yapmıştır. Hedef konumu için nokta stabilizasyonu veya DWA açı kontrolü uygulanmamıştır. Uygulamaya ait parametreler Tablo 6.2'de verilmiştir. Uzunluk birimleri santimetre, açı birimleri derece türünden ifade edilmiştir. Verilen parametreler dâhilinde yapılan benzetim uygulamasının arayüz çıktıları Şekil 6.8 ve Şekil 6.9'da verilmiştir. Oluşan yörüngelerin grafik çıktıları da her iki metot için Şekil 6.10 ve Şekil 6.11'de verilmiştir.

Tablo 6.2. Dışbükey labirent testi parametreleri

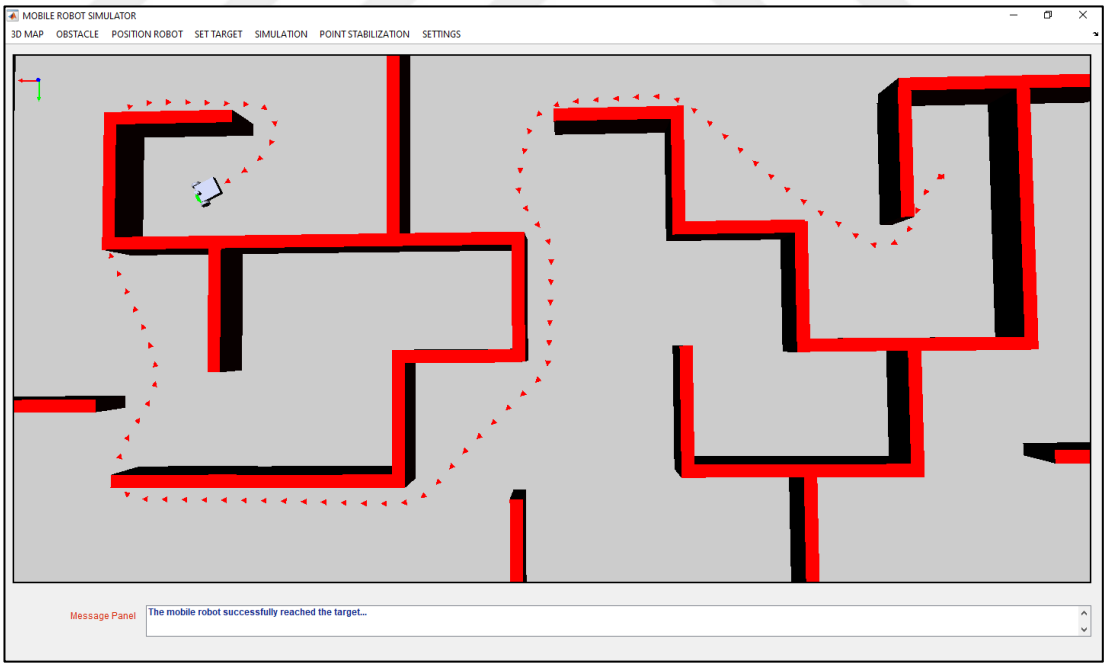
| Parametreler | Değerleri |
|---|---|
| Başlangıç konumu (x, y, θ) | (671 cm, 491 cm, 45$^o$) |
| Hedef konumu (x, y) | (2500 cm, 509 cm) |
| DWA lokal pencere boyutu (x,y) | (150, 150) |
| IBC çapı (r) | 300 |
| Mobil robot | Diferansiyel sürüş (Mavi-Bot Sim) |
| Yörünge uzunluğu - DWA | 5155,1 cm |
| Yörünge uzunluğu - önerilen metot | 4774,9 cm |
| Ort. işlem süresi (1 periyot) - DWA | 22 ms |
| Ort. işlem süresi (1 periyot) - önerilen metot | 8 ms |

Haritanın yapısı gereği robotun başlangıç konumundan itibaren lokal minimumlara neden olan içbükey engellerle karşılaşılmamaktadır. Önerilen metot DWA tarafından planlanan yörüngeye oldukça paralel bir yol planlayarak salt yol planlama

performansı olarak (kurtarma modu kullanılmaması, lokal minimum olmaması) başarılı bir sonuç ortaya koymuştur.



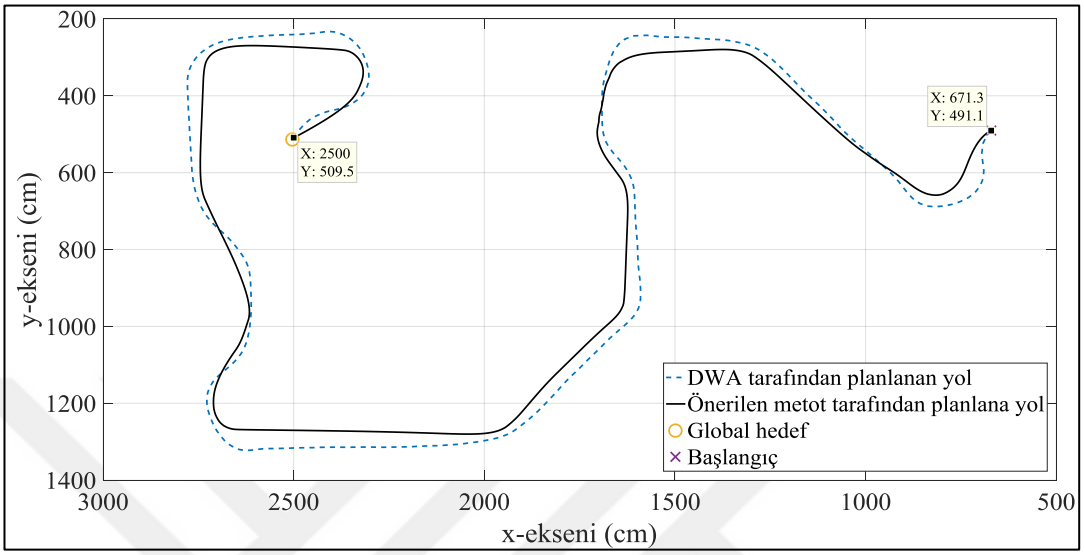

Şekil 6.8. Dışbükey haritada DWA uygulaması




Şekil 6.9. Dışbükey haritada önerilen metodun uygulaması

Şekil 6.8 ve Şekil 6.9'da DWA ve önerilen dinamik yol planlama algoritmaları tarafından oluşturulan yörüngeler verilmiştir. Robot başlangıç pozisyonu x ve y eksenlerinde sırasıyla 671 cm ve 491 cm, duruş açısı $45^{o}$ olarak tanımlanmıştır.

Verilen başlangıç konumunun x ve y bileşenleri ile grafik üzerindeki konum verileri arasındaki milimetrik farklılıklar benzetim arayüzü üzerinde tanımlı konumlandırma modelinin hata varyansları dâhilindedir.

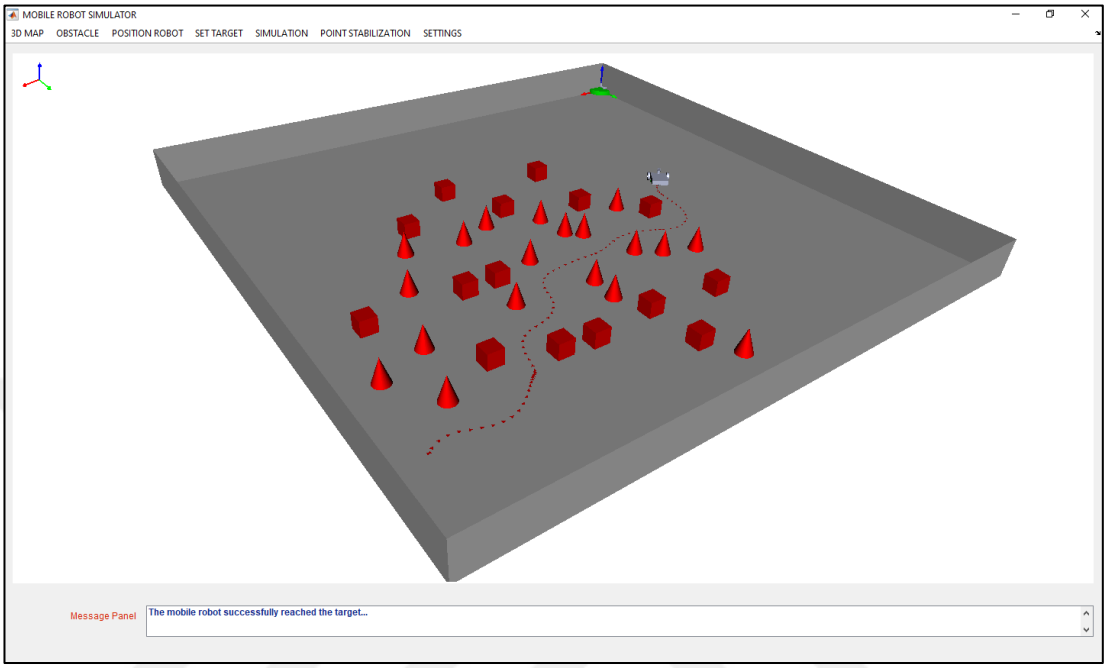



Şekil 6.10. Dışbükey haritada DWA ve önerilen metodun oluşturduğu yol

Aynı harita için yapılan uygulamalar sonucunda benzer geometride yörüngeler elde edilmiştir. Metotlar arasında yörünge uzunlukları bakımından bir değerlendirme yapıldığında önerilen yol planlama metodu tarafından oluşturulan yörüngenin daha kısa olduğu görülmektedir. Yol uzunlukları önerilen metot ve DWA için sırasıyla 4474,9 cm ve 5155,1 cm olarak hesaplanmıştır. DWA algoritmasının işlem yükü ağırlıklı olarak lokal pencerede planlanan yörüngelerden kaynaklanırken önerilen metot için optimal lokal hedef tespiti için taranan açıklık sayısının fazla olması ve bu açıklıkların indirgenmesi işlemlerinden kaynaklanmaktadır. Dolayısıyla parçalı ve çok sayıda engel profiline sahip ortamlar, engel profili süreklilik arz eden ortamlara göre önerilen metot için daha zorlayıcıdır. DWA için ise lokal maliyet haritasının çıkarılması işlemi planlanan yolun optimal çözüme yaklaşmasını doğrudan etkilemektedir.

Rasgele engellerin harita üzerinde dağıtıldığı kompleks bir harita üzerinde DWA ve önerilen yol planlama yöntemlerinin karşılaştırılmasına ait parametreler Tablo 6.3'te verilmiştir. DWA metodu global harita bilgisine önceden sahiptir. Buna karşın önerilen yöntem lokal düzeyde işlem yapmaktadır. Uzunluk birimleri santimetre, açı birimleri derece türünden ifade edilmiştir. Verilen parametreler dâhilinde yapılan

benzetim uygulamasının arayüz çıktıları Şekil 6.11 ve Şekil 6.12'te verilmiştir. Oluşan yörüngelerin grafik çıktıları da her iki metot için Şekil 6.13 ve Şekil 6.14'te verilmiştir.

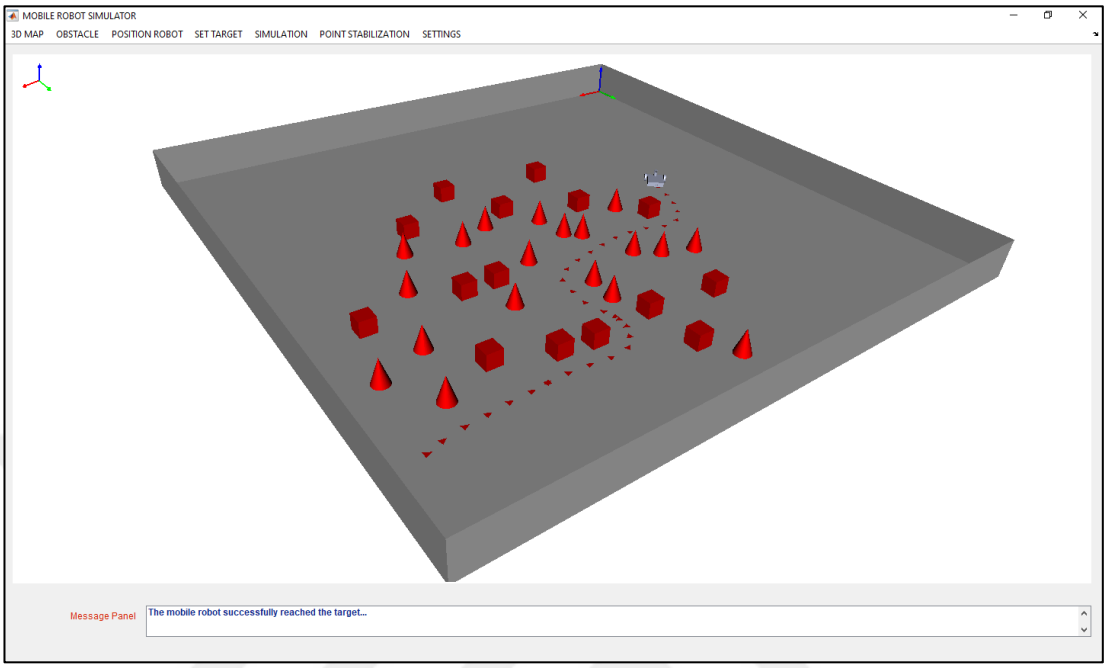



Şekil 6.11. Kompleks haritada DWA uygulaması

Çok sayıda ve rasgele dağıtılmış engellerin bulunduğu bir haritada aynı başlangıç ve hedef konumları için DWA ve önerilen yol planlama metodu tarafından izlenen yörüngelerin uzunlukları sırasıyla 1406 cm ve 1457 cm olarak hesaplanmıştır. Engellerin çok sayıda olması ve buna bağlı olarak robotun geçebileceği açıklık yerleşiminin revizyon işlemleri önerilen planlama yaklaşımının optimal altı bir çözüme gitmesinde etkili olmuştur. Bir diğer faktör ise maliyet fonksiyonunda tanımlı marjinal açısal hareketleri engelleme parametresinin robotun mevcut hareket doğrultusuna pozitif yönde ağırlık vermesidir.

Optimal lokal hedef noktası belirlenirken maliyet haritası üzerindeki ilgili değerinin yanı sıra ilgili noktaya yönelim için gereken açısal değişim de belli bir katsayı oranında kararı etkilemektedir. Bu yaklaşımın temel amacı kararsızlık durumları ve düşük maliyet farkı olan noktalar arasında robotun salınım yapmasını önlemektir. DWA ise lokal maliyet haritasını baz alarak anlık yol planlama yaptığı için açısal yer değişimi ile ilgili bir kriter uygulanmamıştır. Bilinmeyen harita koşullarında çalışması ve engellerin yapısı gereği açıklık sayısının göreceli olarak fazla olması

sebebiyle önerilen yol planlama yöntemi yörünge uzunluğu bakımından DWA metoduna göre dezavantajlı olmaktadır.



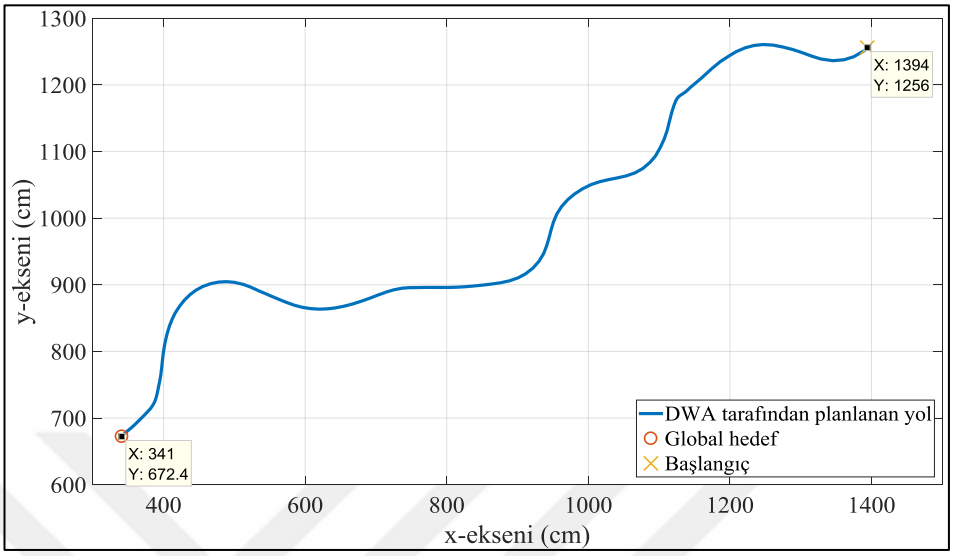

Şekil 6.12. Kompleks haritada önerilen metodun uygulaması

Tablo 6.3. Kompleks harita testi parametreleri

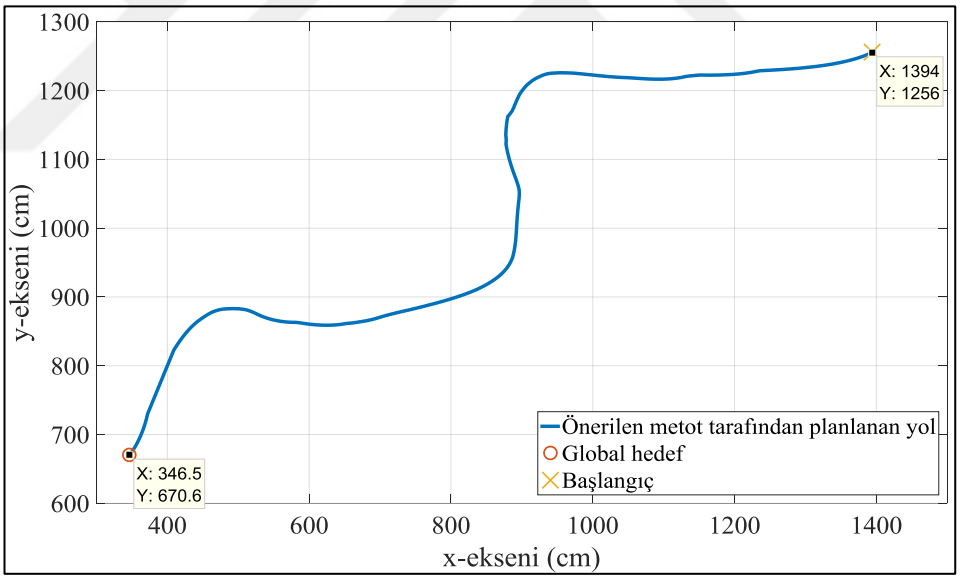
| Parametreler | Değerleri |
|---|---|
| Başlangıç konumu (x, y, θ) | (1394 cm, 1256 cm, 222$^{o}$) |
| Hedef konumu (x, y) | (345 cm, 671 cm) |
| DWA lokal pencere boyutu (x,y) | (150 cm, 150 c) |
| IBC çapı (r) | 300 cm |
| Mobil robot | Diferansiyel sürüş (Mavi-Bot Sim) |
| Yörünge uzunluğu - DWA | 1406 cm |
| Yörünge uzunluğu - önerilen metot | 1457 cm |
| Ort. işlem süresi (1 periyot) - DWA | 20 ms |
| Ort. işlem süresi (1 periyot) - önerilen metot | 9 ms |

Başlangıç konumundan global hedefe yapılan navigasyon sürecinde hareket doğrultusunda iç bükey engeller ile karşılaşılan, labirent tipindeki bir harita üzerinde de DWA ve önerilen yol planlama yöntemi uygulanmıştır. Diğer uygulamalarla benzer biçimde DWA metodu global harita verisini kullanırken önerilen yol

planlama yöntemi lokal düzeyde sensör verileri ile planlama yapmıştır. Hedef konumu için nokta stabilizasyonu veya DWA açı kontrolü uygulanmamıştır.




Şekil 6.13. Kompleks haritada DWA uygulaması için oluşan yol

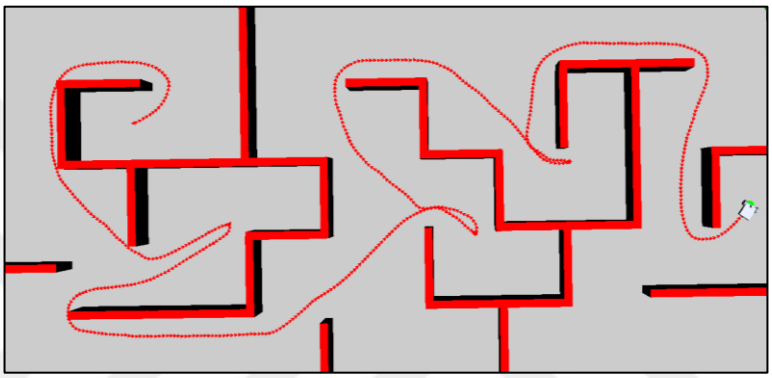



Şekil 6.14. Kompleks haritada önerilen metot uygulaması için oluşan yol

Uygulamaya ait parametreler Tablo 6.4'de verilmiştir. Uzunluk birimleri santimetre, açı birimleri derece türünden ifade edilmiştir. Verilen parametreler dâhilinde yapılan benzetim uygulamasının arayüz çıktıları Şekil 6.15 ve Şekil 6.16'de verilmiştir. Oluşan yörüngelerin grafik çıktıları da her iki metot için Şekil 6.17 ve Şekil 6.18'da verilmiştir.

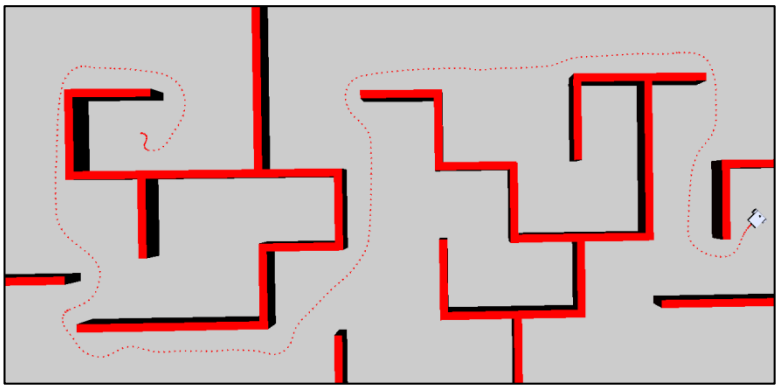
Tablo 6.4. İçbükey labirent testi parametreleri

| Parametreler | Değerleri |
|---|---|
| Başlangıç konumu (x, y, θ) | (2460 cm, 483 cm, 180$^o$) |
| Hedef konumu (x, y) | (81 cm, 837 cm) |
| DWA lokal pencere boyutu (x,y) | (150 cm, 150 cm) |
| IBC çapı (r) | 300 cm |
| Mobil robot | Diferansiyel sürüş (Mavi-Bot) |
| Yörünge uzunluğu - DWA | 6544 cm |
| Yörünge uzunluğu - önerilen metot | 9381 cm |
| Ort. işlem süresi (1 periyot) - DWA | 25 ms |
| Ort. işlem süresi (1 periyot) - önerilen metot | 9 ms |

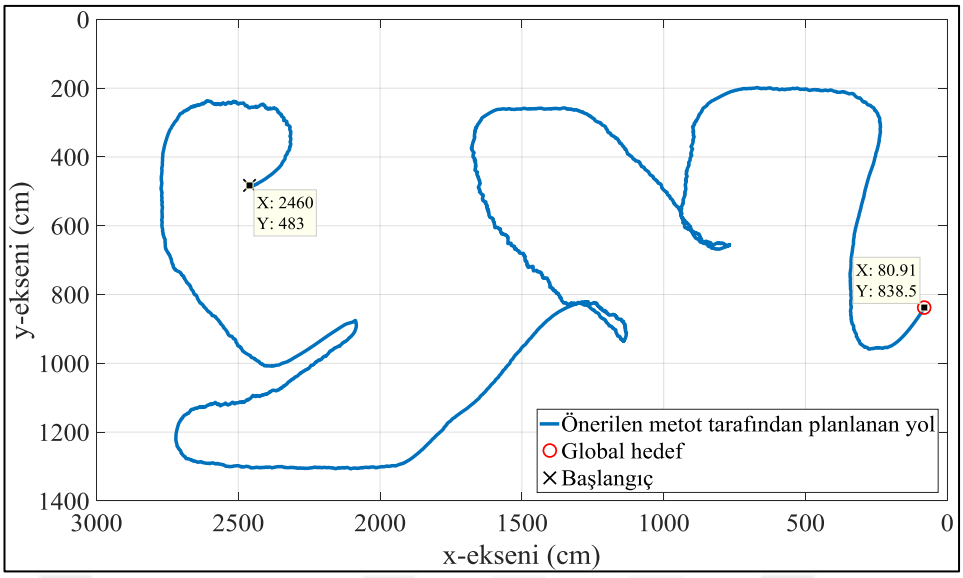


Şekil 6.15. İçbükey haritada önerilen metodun uygulaması

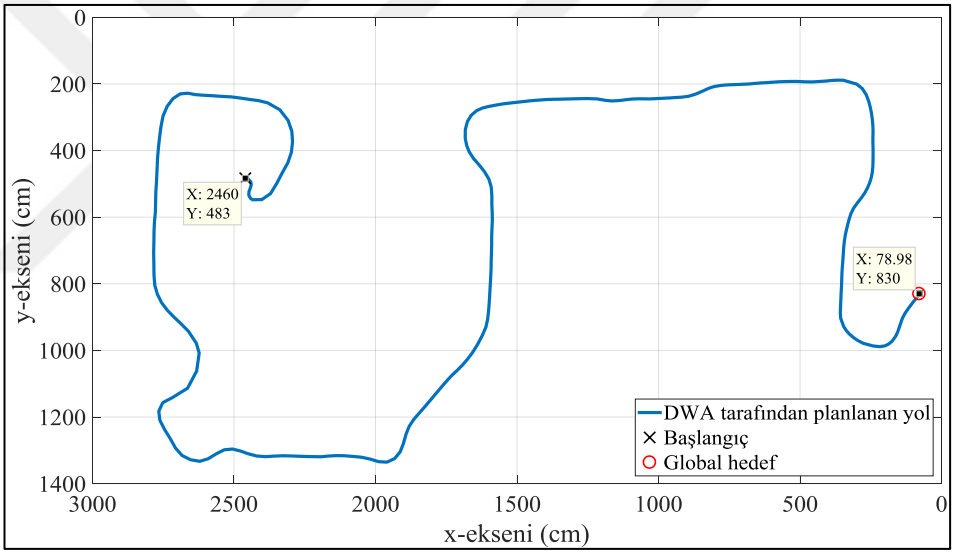
Harita bilgisinin bilinmemesi durumu kapalı uçlu koridor benzeri içbükey engeller karşısında yerel minimum bölgeleri oluşturabilmektedir. Optimal açıklık tespiti algoritması global hedefe olan uzaklığı baz alarak haritayı maliyetlendirdiği için daha sonraki aşamada tamamen kapalı bir LIDAR ölçümü ile karşılaşılabilir. Bu durumda sıfır-yarıçap dönüşü yapılarak tekrar açıklık tespiti aşamasına geçilir. Bu hareketler görsel arayüz üzerinde ve yörünge grafiği üzerinde sırasıyla Şekil 6.15 ve Şekil 6.16 üzerinde görülmektedir.



Şekil 6.16. İçbükey koşullarda DWA uygulaması




Şekil 6.17. İçbükey haritada önerilen metodun oluşturduğu yol



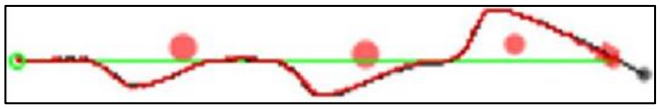

Şekil 6.18. İçbükey haritada DWA uygulaması için oluşan yol

Ortam haritasına gereksinim duyan DWA uygulamasında ise Dijkstra algoritması tarafından global harita önceden bir maliyet haritası çıkardığı için planlayıcı algoritma yerel minimumla karşılaşmamaktadır. Her iki yaklaşım ortak haritalar üzerinde uygulanarak yol planlama çıktıları verilmiştir. DWA önceden verilen statik harita bilgilerini kullanmıştır. Harita bilgisine gereksinim duymayan önerilen yol planlama metodu bu avantajı karşısında yerel minimum durumları ile karşılaşabilmektedir. Bu tip durumlarda tanımlı kurtarma modları ile etkili bir biçimde alternatif yol planlama davranışı sergilemektedir. Önceden bilinmesi zor

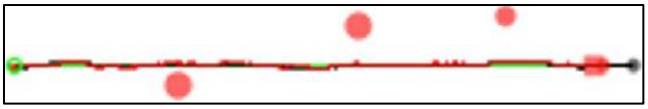
veya değişken yapıdaki haritalar gerçek zamanlı uygulamalarda sık karşılaşılan bir problemdir ve bu probleme etkin bir çözüm sunulmuştur.

### 6.1.4. Hareketli engellerin tespiti

Tez kapsamında üzerinde çalışılan hareketli engel tespitine ilişkin sonuçlar 2 boyutlu benzetim arayüzü üzerinde gösterilmiştir. Tanımlanan engellerin geometrisi dairesel kesitte verilmiştir. Boyutları farklı olan dairesel engellerin doğrusal bir yörünge izlediği ve sabit hızlarla iki nokta arasında sürekli hareket ettiği varsayılmıştır. Sensör görüş alanı dışında kalan yönlerden engellerin robota çarpmayacağı ve kör noktalarda meydana gelebilecek çarpma durumları için ekstra bir önlem alınmadığı varsayılmıştır. Planlanan statik yolu kesecek şekilde hareket eden engeller arasından geçen robot, hareket tespiti uygulanmaması durumunda, engelin hareket yönüne bağlı olarak sürüklenme eğilimi göstermektedir. Değişen engel konumuna bağlı olarak tespit edilen lokal hedef konumu hareketli hedef takibi etkisi yaparak robotun global hedeften uzaklaşmasına sebep olmaktadır. Yapılan uygulamada engellerin robot ile kesişecek biçimde hareket zamanlaması ayarlanarak 3 engel tarafından da sürüklenmesi sağlanmıştır. Şekil 6.19'da görülen durum buna örnek olarak verilmiştir. Kırmızı renkli daireler hareketli engelleri, yeşil yol statik yörüngeyi, kırmızı-siyah yol ise dinamik yörüngeyi göstermektedir. Mobil robot ise kırmızı dikdörtgen ile gösterilmiş ve siyah doğru parçası ile yönü belirtilmiştir. Robot soldan sağa hareket ederken ilk engelle karşılaştığı anda engel yukarıdan aşağı hareket yaparken tam ters yönde harekete başladığında lokal hedefler de engelin konumuna bağlı olarak global hedef doğrultusundan uzaklaşmıştır. Robot statik haritada global hedefe doğrusal bir yörüngede sabit hızla seyretmesi halinde çarpışma olmayacak olmasına karşın, açıklık tespiti algoritmasının anlık engel profiline göre karar vermesi sebebi ile istenmeyen bir yörünge ortaya çıkmıştır. Aynı senaryo hareket tespiti aktif edilerek de gerçekleştirilmiştir. Şekil 6.20'de bu uygulama sonucunda oluşan yörünge verilmiştir. Engeller bu senaryo için de robotun izlediği yörüngeyi robota aynı mesafe uzaklıklarda kesmiştir. Engellerin sonraki işlem adımlarında bulunacağı konumları baz alan lokal hedef tespiti ve açıklık bulma algoritması, robotun mevcut yörüngesini takip etmesi durumunda çarpışma yaşanmayacağı varsayımını doğrulamıştır.

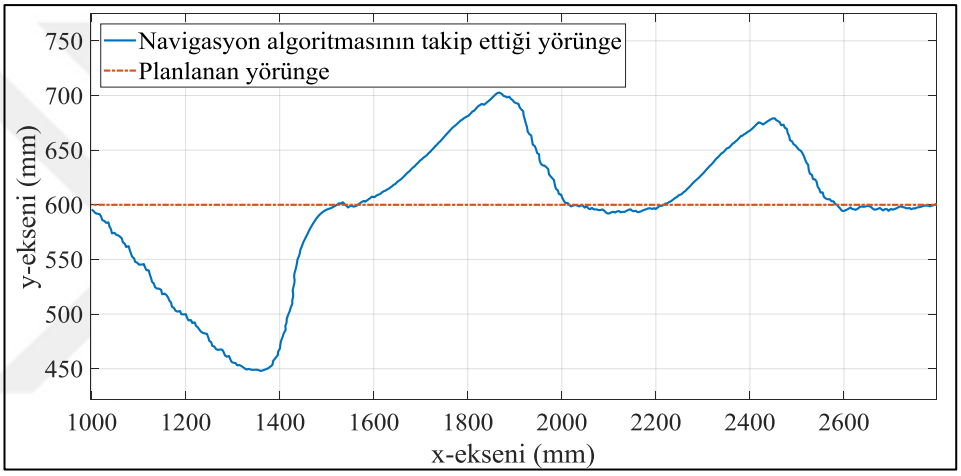


Şekil 6.19. Hareketli engel algılama pasif durumu



Şekil 6.20. Hareketli engel algılama aktif durumu

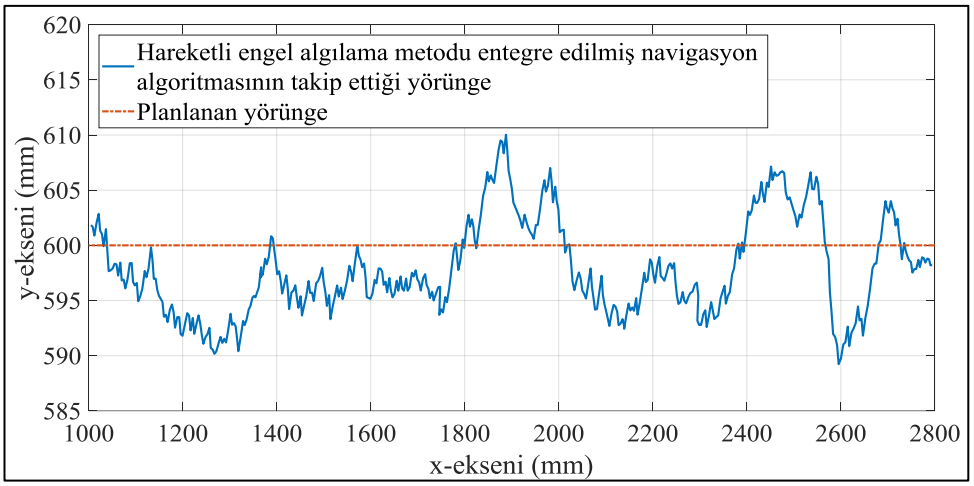
Hareketli engel tespitinin uygulanmadığı durum için planlanan yol temel koordinat sistemine göre iki boyutlu olarak milimetre biriminde Şekil 6.21'de verilmiştir.



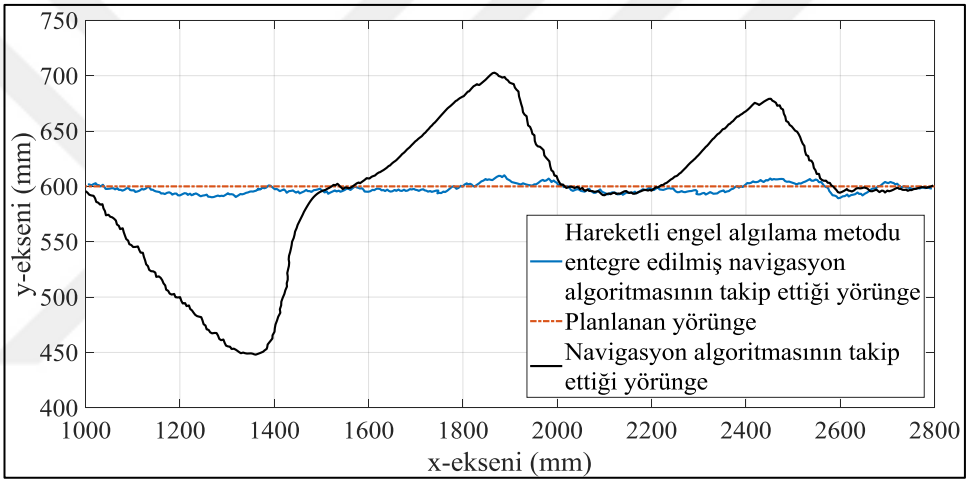

Şekil 6.21. Metot uygulanmadığında oluşan dinamik yörünge

Hareketli engel tespiti uygulanmadığı durum için planlanan yol temel koordinat sistemine göre iki boyutlu olarak milimetre biriminde Şekil 6.21'de verilmiştir. Aynı başlangıç ve hedef noktaları için hareketli engel tespiti metodunun uygulanması sonucunda oluşan yörünge milimetre biriminde Şekil 6.22'de gösterilmiştir. Her iki grafik üzerinde de dinamik ve statik yol karşılaştırmalı olarak verilmiştir.

Dinamik yörüngelerin birbiri ile karşılaştırıldığı grafik Şekil 6.23'de gösterilmiştir. Bu yörüngeler metrik uzunluklar bazında Tablo 6.5'te verilmiştir. İdeal durumda oluşan statik yol ile dinamik yörüngelerin ideal yol uzunluğundan %14,94 olan sapma değeri %3,44 seviyesine düşürülerek gözle görülür bir iyileştirme sağlanmıştır.




Şekil 6.22. Metot uygulandığında oluşan dinamik yörünge




Şekil 6.23. Dinamik yörüngelerin karşılaştırılması

Tablo 6.5. Dinamik yörüngelerin uzunlukları

| Hareketli engel takibi engel takibi | Planlanan yörünge [mm] | Takip edilen yörünge [mm] | Hata [%] |
|---|---|---|---|
| Aktif | 1800 | 1862 | 3,44 |
| Pasif | 1800 | 2069 | 14,94 |

## 6.2. İç Mekân Konumlandırma Sistemi

Görüntü işleme tabanlı iç mekân konumlandırma sistemi için yapılan deneyler Kocaeli Üniversitesi Mühendislik Fakültesi 3. kat koridorunda gerçekleştirilmiştir. Mobil robot platformu uzaktan kumanda kontrolü ile rasgele bir yörüngede ettirilmiştir. Hareket sırasında 4 farklı kaynaktan konum ölçümü kaydedilmiştir. Enkoder tabanlı odometre bilgisi, görüntü işleme tabanlı konumlandırma sisteminden gelen konum bilgisi, EKF tabanlı konum kestirimi ve çok yüksek doğrulukta ölçüm

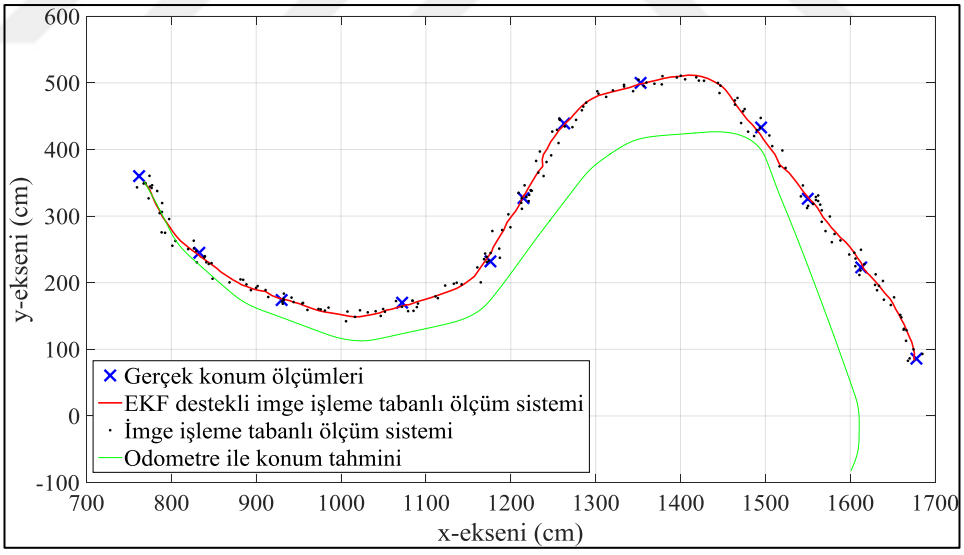
yapabilen noktasal lazer mesafe ölçüm cihazından alınan gerçek konum bilgisi robotun hareketi sırasında kaydedilmiştir. Lazer ölçüm cihazından alınan konum verisi, robot durdurularak anlık olarak ölçülmüştür. Bu ölçümler robotun durdurulduğu noktalardaki gerçek konum bilgisi olarak varsayılmış ve diğer ölçüm kaynakları için referans olarak kabul edilmiştir. Lazer ölçümü dışındaki konum hesaplamaları robot hareketi ile eşzamanlı olarak gerçekleştirilmiştir. Odometre ve görüntü işleme tabanlı konumlandırma sistemi için öngörülen hata varyans değerleri sırasıyla Denklem (6.1)'de verilmiştir. Bu değerler enkoder verisindeki fiziksel etkilerden kaynaklı kayıplardan ve ölçüm sisteminden kaynaklanan ölçüm hatalarına dayanmaktadır. Odometre ile yapılan konum ölçümü zamanla birikimli bir hata ile sonuçlanmaktadır. Yatay ve düşey eksenlerde saniyede 2 cm pozisyon hatası, $1^o$ oryantasyon hatası birikimine yol açtığı varsayılmıştır. İç mekan konumlandırma sistemi ise zamandan bağımsız olarak maksimum 10 cm pozisyon hatası ve $1^o$ oryantasyon hatası ile ölçüm yapmaktadır.

$$q_x = q_y = 2\,\frac{cm}{s},\ q_\theta = \frac{1^\circ}{s},\ r_x = r_y = 10\,cm,\ r_\theta = 1^\circ \tag{6.1}$$

Yapılan deneyde 12 farklı noktada ölçülen gerçek konum ve bu konumların ölçüldüğü noktalarda görüntü işleme tabanlı konumlandırıcıdan okunan konum, odometre kaynaklı konum ve EKF destekli konum verileri karşılaştırmalı bir biçimde Tablo 6.6'da verilmiştir. Oluşan ölçüm ve kestirim hataları da ilgili satırlarda verilmiştir. Tablo olarak verilen deneysel sonuçlar bir yörünge formatına dönüştürülerek Şekil 6.24'te gösterilmiştir. Robot harekete başlamadan önce ilk mutlak konum ölçümü yapılmıştır. Bu noktadan itibaren yeşil renkli yörünge ile gösterilen odometre kaynağı zamana bağlı olarak birikimli bir konum hatasına yol açmıştır. Bu hatanın temel sebebi lastiklerin yüzeye kusursuz bir tutunma sağlayamaması, boşa dönme (patinaj) ve keskin dönüşlerde hesaplanamayan ataletsel savrulmalardır. Ölçüm sisteminden okunan veriler incelendiğinde belli bir hata bandında rasgele ölçümler verdiği görülmektedir. Birikimli bir hata karakteristiğine sahip değildir. Gerçek konum ölçümü civarında bir hata bandında (ölçüm hata varyansı) konum bilgisi elde edilmiştir. Ölçüm değerleri şekil üzerinde siyah noktalar kümesi ile gösterilmiştir.

Tablo 6.6. İç-mekân konumlandırma sistemi ölçüm ve kestirim hataları

| Gerçek Konum (cm) | | EKF destekli konumlandırma sistemi ölçümü (cm) | | Konumlandırma sistemi ölçümü (cm) | | Kestirim hatası (cm) | | Ölçüm hatası (cm) | |
|---|---|---|---|---|---|---|---|---|---|
| x | y | x | y | x | y | x | y | x | y |
| 762 | 360 | 759 | 356 | 767 | 354 | -3 | -4 | 5 | -6 |
| 833 | 245 | 835 | 241 | 826 | 240 | 2 | -4 | -7 | -5 |
| 930 | 174 | 932 | 175 | 937 | 179 | 2 | 1 | 7 | 5 |
| 1072 | 170 | 1075 | 167 | 1079 | 165 | 3 | -3 | 7 | -5 |
| 1176 | 232 | 1173 | 235 | 1172 | 237 | -3 | 3 | -4 | 5 |
| 1215 | 327 | 1216 | 331 | 1210 | 334 | 1 | 4 | -5 | 7 |
| 1263 | 439 | 1265 | 440 | 1268 | 443 | 2 | 1 | 5 | 4 |
| 1353 | 500 | 1350 | 498 | 1357 | 496 | -3 | -2 | 4 | -4 |
| 1495 | 433 | 1492 | 430 | 1490 | 429 | -3 | -3 | -5 | -4 |
| 1550 | 326 | 1554 | 322 | 1554 | 332 | 4 | -4 | 6 | 4 |
| 1613 | 223 | 1617 | 219 | 1617 | 217 | 4 | -4 | 4 | -6 |
| 1678 | 86 | 1675 | 83 | 1685 | 91 | -3 | -3 | 7 | 6 |



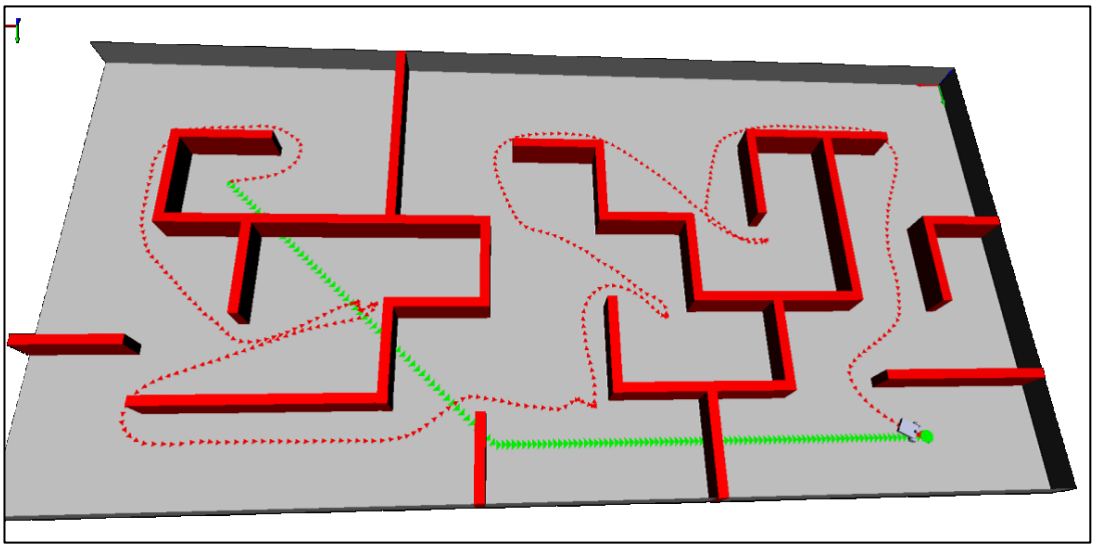

Şekil 6.24. EKF destekli konumlandırma sisteminin deneysel sonucu

Mutlak konum ölçümleri robot durdurularak yüksek hassasiyetli (milimetre düzeyinde) lazer mesafe ölçer ile ölçülmüştür. Bu noktalar şekil üzerinde mavi renkli "x" sembolleri ile verilmiştir. EKF destekli yaklaşım ile kestirilen konum bilgisi ise kırmızı renkli yörünge formatında gösterilmiştir. Gerçek konum bilgisine diğer kaynaklardan daha yakın sonuçlar elde edildiği gözle görülebileceği gibi Tablo

6.6'da verilen kestirim hatalarının ölçüm hatalarına göre daha düşük bir aralıkta seyretmesi ölçüm verilerinin iyileştirildiğini göstermektedir. Ölçüm sistemi en çok 8 cm hata ile ölçüm yapabilirken EKF destekli konumlandırma yaklaşımında bu değer 5 cm olarak elde edilmektedir.

### 6.3.Benzetim Arayüzü Çıktıları

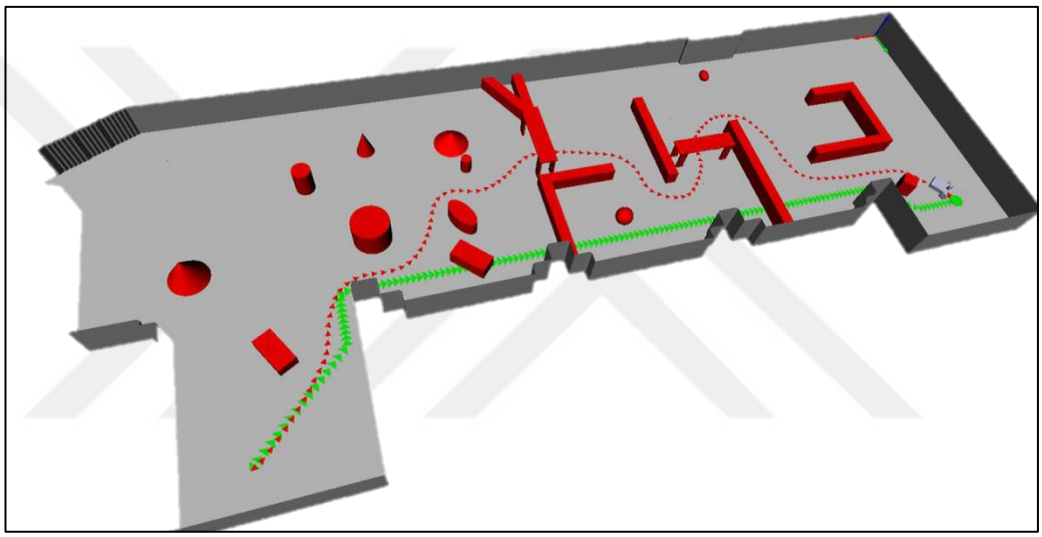
Benzetim arayüzü üzerinde oluşturulan farklı koşullardaki haritalar üzerinde dinamik yol planlayıcının oluşturduğu yörüngeler incelenmiştir. Statik yol planlayıcı EDT olarak belirlenmiş olup yeşil renk ile kodlanmıştır. MATLAB programının çalışma periyodu çizim araçları ve 3 boyutlu grafik arayüzünün çalıştırılması da dâhil olmak üzere yaklaşık 80 ms seviyelerindedir. Çalışılan makine Intel Core i3-3220 3,30 GHz işlemci, 8GB RAM kapasitesi ve Windows 10 işletim sistemine sahip bilgisayardır. İşlem süresine konumlandırma, LIDAR benzetimi, açıklık tespiti, lokal hedef tespiti ve grafiklerin görüntülenmesi işlemleri dahildir. Sanal mobil robot fiziksel robot boyutlarında tanımlı olup, tekerlek boyut, yerleşim ve kinematik model gerçek parametrelere göre tanımlanmıştır. Tekerlek dönüş hızları 0 ile 100 cm/sn arasında sınırlandırılmış, ortalama hareket hızı 50 cm/sn olarak belirlenmiştir.



Şekil 6.25. Benzetim arayüzü sonuçları-1

LIDAR menzili 10 m, tarama açısı $270^{o}$, açısal çözünürlüğü $1^{o}$ olarak tanımlanmış, robot üzerindeki yerleşimi fiziksel robot ile aynı konuma yerleştirilmiştir. Global hedefe ulaşma toleransı 30 cm olarak tanımlanmıştır. Kalman filtresi için tanımlı proses ve ölçüm gürültüleri sırasıyla 15 cm/s ve 30 cm olarak belirlenmiştir.

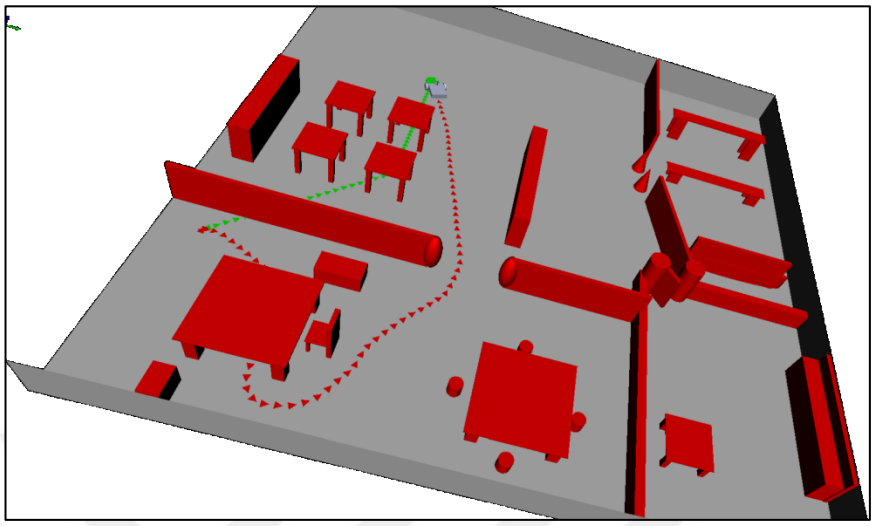
Çarpışma engelleme algoritması dahilinde planlanan yola 1 metreden yakın bir engel tespit edilirse lokal hedef güncelleme işlemi gerçekleştirilmiştir. Şekil 6.25'de önceden konumu bilinmeyen iç bükey engeller yerleştirilmiş labirent tipi bir harita üzerinde oluşturulan dinamik yörünge görülmektedir. Lokal hedefleri takip ederek ilerleyen mobil robot ilk etapta kapalı uçlu koridorlara girmektedir. LIDAR görüş menzilinde açıklık tespit edilemeyecek kadar koridorda ilerlendiğinde ise sıfır-yarıçap dönüşü ile alternatif lokal hedefler taranarak global hedefe giden alternatif yörüngeler oluşturulmuştur EDT metodu ile hesaplanmış statik yol da harita üzerinde yeşil renk tonunda gösterilmiştir.



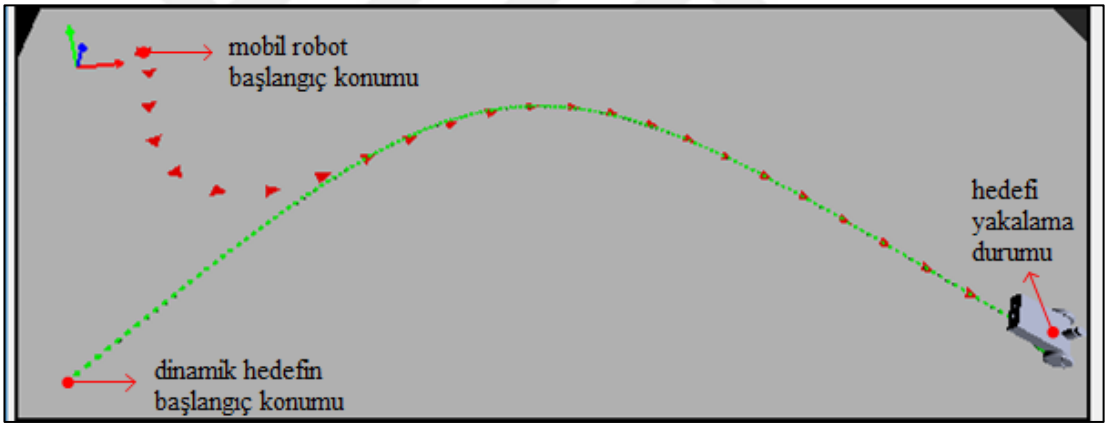
Şekil 6.26. Benzetim arayüzü sonuçları-2 [119]

Şekil 6.26'da verilen grafik üzerinde önceden konumu bilinen duvar ve kolon girinti-çıkıntıları ve de konumu bilinmeyen rasgele şekillerde engellerden oluşan bir harita verilmiştir. Harita Kocaeli Üniversitesi Mühendislik Fakültesi 3. Kat koridorunun ölçüleri benzetim arayüzüne aktarılmıştır. Fiziksel genişliği 30x30 $m^2$ olan alan 2 boyutlu harita olarak 600x600 piksel boyutunda bir imgeye indirgenmiştir. İmge boyutunda 1 pikselin ifade ettiği fiziksel alan 5x5 $cm^2$ olarak değerlendirilmiştir. Önceden konumu bilinen engeller statik planlayıcı tarafından gözetilmiştir. Önceden bilinmeyen engeller ise statik planlayıcı tarafından işleme alınmadığı için bu engeller ile kesişmektedir. Temel engel kütüphanesi kullanılarak oluşturulan bir iç mekân haritası Şekil 6.27'de verilmiştir. Masa, sandalye, duvar, silindir, koni gibi sık kullanılan iç mekân harita araçları kütüphane elemanları ile oluşturulmuştur.

Önceden bilinmeyen objeler arasından verilen hedefe doğru oluşturulan yörünge ve statik yol planlama çıktıları bu harita üzerinde gösterilmiştir.



Şekil 6.27. Benzetim arayüzü sonuçları-3 [119]

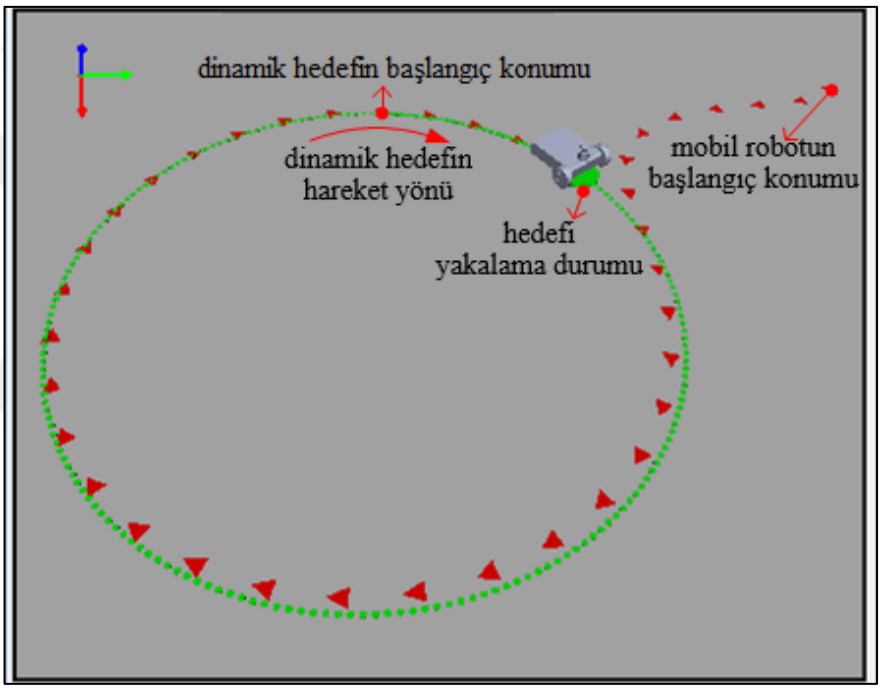



Şekil 6.28. Benzetim arayüzü sonuçları-4 [119]

Global hedef benzetim başladıktan sonra güncellenebilmektedir. Dinamik yol planlama algoritması değişen hedef konfigürasyonuna adapte olarak lokal yönelimlerini güncel hedefe göre gerçekleştirebilmektedir. Benzetim ortamında modellenen değişken global hedef uygulamalarının sonuçları Şekil 6.28, Şekil 6.29 ve Şekil 6.30 üzerinde görülebilmektedir.

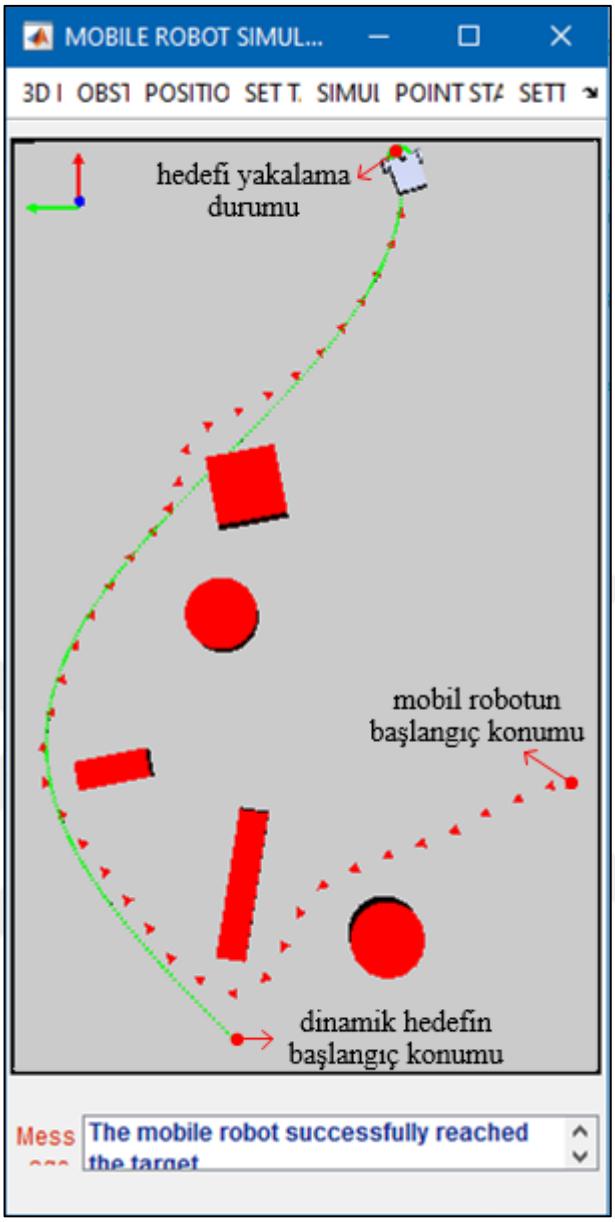
Şekil 6.28'de verilen durumda mobil robotun ortalama hızı global hedefi yakalayabileceği düzeyde görece daha yüksek tanımlanmıştır. Global hedef tanımlanan yörünge üzerinde hareket etmeye başlarken eş zamanlı olarak mobil robot başlangıç noktasından hareketine başlatılmıştır. Sinüzoidal bir yörünge izleyen

global hedefin izlediği yörünge yeşil, mobil robotun eşzamanlı olarak takip ettiği yörünge ise kırmızı renk kodu ile gösterilmiştir. Mobil robot hareketli global hedefi yakaladığında benzetim işlemi durdurulmuştur. Benzer bir senaryo Şekil 6.29'da verilen dairesel bir yörünge üzerinde gerçekleştirilmiştir. Aynı anda harekete başlayan dinamik global hedef ve mobil robot yörünge üzerinde bir tam tur hareket ettikten sonra rasgele bir noktada robot hedefi yakalamıştır. Verilen bu iki senaryoda ortamda herhangi bir engel bulunmadığı için açıklık tespiti, lokal hedef tespiti ve çarpışma engelleme algoritmaları devreye girmemiştir. Mobil robot doğrudan hedefe yönelim modunda çalışarak global hedefi yakalamıştır.




Şekil 6.29. Benzetim arayüzü sonuçları-5 [119]

Sinüzoidal bir yörüngede sabit hızda hareket eden dinamik global hedef uygulamasının bilinmeyen engeller altında işleyişini gösteren senaryo Şekil 6.30'da verilmiştir. İlk iki senaryonun aksine temel navigasyon fonksiyonları da devreye girmiştir. Robot doğrudan hedefe yönelim modu ile dinamik yol planlama işlemlerini dönüşümlü olarak gerçekleştirmiştir. Engellerin konumları önceden bilinmediği için dinamik yol planlama algoritması hedef takibi sırasında karşılaşılan her kritik engel için doğrudan yönelim modundan çıkmak zorunda kalmıştır. Çarpışma engelleme algoritması kapsamında belirlenen limitler dâhilinde kritik olarak değerlendirilmeyen engellerin takip sürecine herhangi bir etkisi olmamıştır.

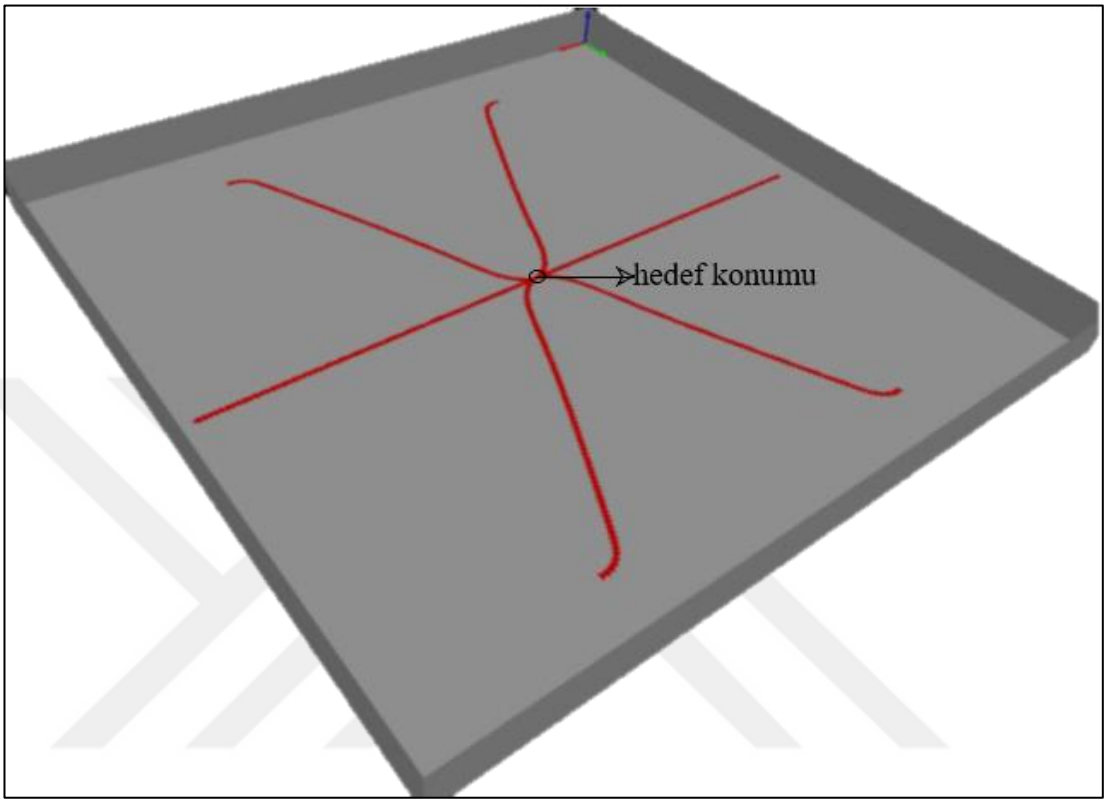



Şekil 6.30. Benzetim arayüzü sonuçları-6 [119]

Sürekli değişen global hedef konumu altında verilen yörünge planlama yanıtı beklendiği gibi dinamik yol planlama fonksiyonları ile eşzamanlı olarak koşturulabilmiştir. Navigasyonun herhangi bir zamanında sabit ve başlangıç konumundan farklı global hedef belirlenebileceği yorumu yapılabilir.

Model tabanlı kontrol ve EEDT metodunun hibrit bir yaklaşımı olarak geliştirilen noktasal stabilizasyon yöntemi benzetim arayüzü üzerinde test edilmiştir. Bir başlangıç noktasından hedefe istenilen açı ile yerleşilmesini sağlayan yöntem engellerin olmaması ve olması durumları için sırası ile Şekil 6.31 ve Şekil 6.32'de verilmiştir. 15 m x 15 m boyutlarında tanımlanan engelsiz harita üzerinde (7,5 m, 7,5

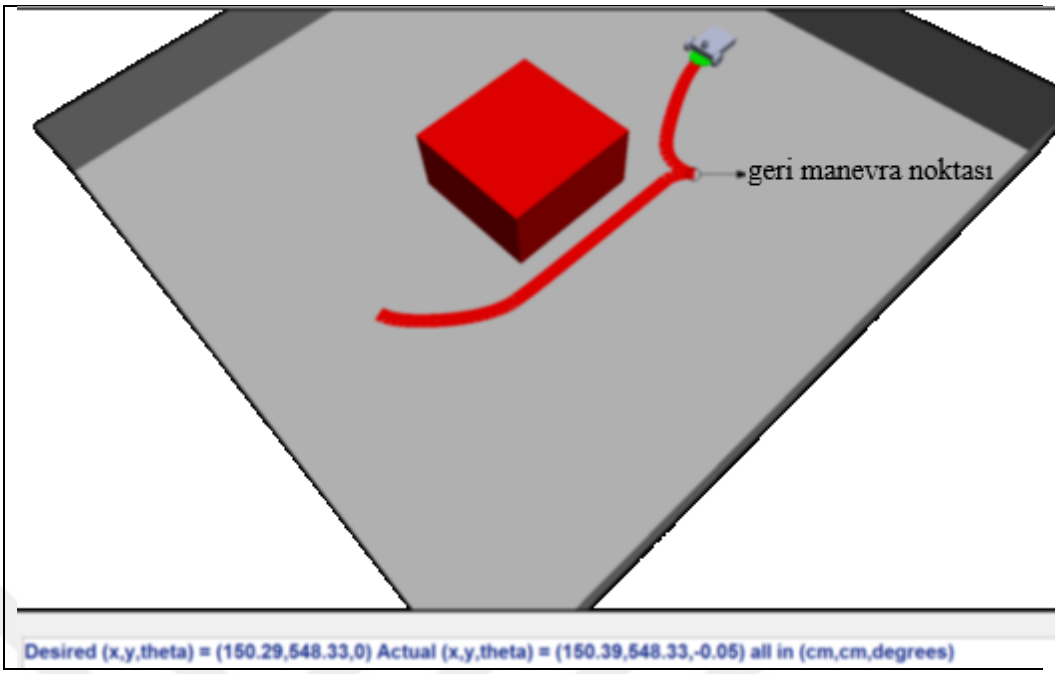
m) Kartezyen konumuna $0^o$ açısal pozisyon ile yerleşmesi istenen robot 6 farklı konumdan $0^o$ başlangıç açısı ile harekete başlamıştır. Başlangıç konumları hedef çevresinde tanımlı çember üzerine $60^o$'' lik açılar ile yerleştirilmiştir.




Şekil 6.31. Benzetim arayüzü sonuçları-7 [119]

Şekil 6.32'de verilen stabilizasyon işleminde rasgele bir konumdan $90^o$ açı ile harekete başlayan mobil robot hedef konumuna $0^o$ açı ile yerleşmiştir. Hedeflenen açısal duruş kriterini sağlamak için modele uygun bir geri manevra hareketi gerçekleştirilmiştir.

Hedef konumlarında sahip olunan gerçek ve olması istenen açısal pozisyon bilgileri arayüzün alt bölümünde bulunan mesaj panelinde kullanıcıya sunulmuştur.




Şekil 6.32. Benzetim arayüzü sonuçları-8 [119]

### 6.4. Öneriler

Dinamik yol planlama yöntemi üzerinde hareketli engellerin sistematik olmayan davranışları karşısında çözümsüz kalabileceği öngörülmektedir. Literatürde bu konuda genel kabul görmüş köklü bir yaklaşım olmaması nedeniyle bu konu üzerinde yapılacak yeni çalışmalara ihtiyaç vardır. İç mekan konumlandırma sistemi olarak yükselen bir eğilim olan ve doğal navigasyon başlığında uygulanan lazer tabanlı veya görü tabanlı konumlandırma sistemleri ileri çalışma olarak tez kapsamında geliştirilen toplam navigasyon sistemine entegre edilebilir. Gerekli donanımsal ve yazılımsal altyapısı doğal konumlandırma yöntemleri için hazır olan mobil robot platformuna AMCL gibi yöntemler hızlı bir şekilde uygulanabilir. Lazer tabanlı engel algılama ve yol planlama yaklaşımları içeren sistem ekstra güvenlik donanımları ile düşük zeminleri algılayarak negatif engel algılama bileşeni eklenebilir. Bu işlemler için sonar ve kızılötesi mesafe sensörleri robotun belirli noktalarına zemini tarayacak biçimde yerleştirilerek dâhili kontrol kartı üzerinden kontrol edilebilir.

## KAYNAKLAR


[1] Cox I. J., Wilfong G. T., *Autonomous Robot Vehicles*, 1st ed., Springer-Verlag, New York, 1990.

[2] https://technology.ihs.com (Ziyaret tarihi: 27 Kasım 2018).

[3] https://www.lg.com (Ziyaret tarihi: 30 Mart 2016).

[4] https://www.belrobotics.com/en-gb/mowers/bigmow (Ziyaret tarihi: 30 Nisan 2016).

[5] https://fetchrobotics.com (Ziyaret tarihi: 3 Haziran 2016).

[6] Ruckelshausen A., Biber P., Dorna M., Gremmes H., Klose R., Linz A., Rahe R., Resch R., Thiel M., Trautz D., Weiss U., BoniRob an Autonomous Field Robot Platform for Individual Plant Phenotyping, *Precision Agriculture*, 2009, **9**(841), 841-847.

[7] www.ccsrobotics.com/products/robosentry.html (Ziyaret tarihi: 31 Temmuz 2018).

[8] http://robchair.isr.uc.pt (Ziyaret tarihi: 31 Temmuz 2018).

[9] Vadakkepat P., Kim J. H., Jesse N., Al Mamun A., Kiong T. K., Baltes J., Anderson J., Verner I., Ahlgren D., *Trends in Intelligent Robotics*, 1st ed., Springer, Bangalore, 2010.

[10] Ullrich G., *Automated Guided Vehicle Systems*, 2nd ed., Springer, Heidelberg, 2015.

[11] https://kaesarchina.com (Ziyaret tarihi: 20 Kasım 2018).

[12] https //godrejlawkimmotors.com (Ziyaret tarihi: 30 Kasım 2018).

[13] Rongbing L., Jianye L., Zhang L., Hang Y., LIDAR/MEMS IMU Integrated Navigation (SLAM) Method for a Small UAV in Indoor Environments, *IEEE Inertial Sensors and Systems Symposium*, Karlsruhe, Germany, 16-17 September 2014.

[14] https://epsnews.com (Ziyaret tarihi: 21 Kasım 2018).

[15] Zunino G., Simultaneous Localization and Mapping for Navigation in Realistic Environments, Masters Thesis, The Royal Institute of Technology, Stockholm, 2002, 020220.

[16] Carlo H., Vis I., Roodbergen K., Transport Operations in Container Terminals: Literature Overview, Trends, Research Directions and Classification Scheme, *Europian Journal of Operational Research*, 2014, **236**(1), 1-13.

[17] Hess W., Kohler D., Rapp H., Andor D., Real-Time Loop Closure in 2D LIDAR SLAM, *IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, 16-21 May 2016.

[18] Yoshida K., The Spacedyn: A Matlab Toolbox for Space and Mobile Robots, *IEEE International Conference on Intelligent Robots and Systems*, Kyongju, South Korea, 17-21 October 1999.

[19] Mirats J. M., Pfeiffer C. F., Mobile Robot Design in Education, *IEEE Robotics and Automation Magazine*, 2006, **13**(1), 69-75.

[20] Dean E., Nair S., Knoll A., User-Friendly Matlab Toolbox for Symbolic Robot Dynamic Modeling Used for Control Design, *IEEE International Conference on Robotics and Biomimetic*, Guangzhou, China, 11-14 December 2012.

[21] Barner S., Geisinger M., Buckl C., Knoll A., EasyLab: Model-Based Development of Software for Mechatronic Systems, *IEEE International Conference on Mechatronic and Embedded Systems and Applications*, Beijing, China, 12-15 October 2008.

[22] Pons C., Perez G., Giandini R., Baum G., A Model-Driving Approach to Constructing Robotic Systems, *Journal of Computer Science & Technology*, 2014, **14**(1), 1-8.

[23] Corke P., A Robotics Toolbox for Matlab, *IEEE Robotics and Automation Magazine*, 1996, **3**, 24-32.

[24] Ivaldi S., Padois V., Nori F., Tools for Dynamics Simulation of Robots: A Survey Based on User Feedback, *Tools for Dynamics Simulation of Robots-Extended Report*, 1402.7050, 1-15, 2014.

[25] Mitchell H., An Algorithmic Approach to Some Problems in Terrain Navigation, *Artificial Intelligence*, 1988, **37**, 171-201.

[26] Payton D., Rosenblatt J., Keirsey D., Grid-Based Mapping for Autonomous Mobile Robot, *Robotics and Autonomous Systems*, 1993, **11**, 13-21.

[27] Khatib O., Real-Time Obstacle Avoidance for Manipulators and Mobile Robots, *IEEE Conference on Robotics and Automation*, St. Louis, USA, 25-28 March 1985.

[28] Brihaye T., Michaux C., Riviere C., Cell Decomposition and Dimension Function in the Theory of Closed Ordered Differential Fields, *Annals of Pure and Applied Logic*, 2009, **159**, 111-128.

[29] Wang H., Yu Y., Yuan Q., Application of Dijkstra Algorithm in Robot Path-Planning, *International Conference on Mechanic Automation and Control Engineering*, Hohhot, China, 15-17 July 2011.

[30] O'Dunlaing C., Yap C., A Retraction Method for Planning the Motion of a Disc, *Journal of Algorithms*, 1982, **6**, 104-111.

[31] Davidor Y., Genetic Algorithms and Robotics: A Heuristic Strategy for Optimization, *Singapore: World Scientific Publishing*, 1991, **1**, 220-225.

[32] Nearchou A., Path Planning of a Mobile Robot Using Genetic Heuristics, *Robotica*, 1998, **16**, 575-588.

[33] Cai J., Peng M., Ma S., RL-ART2 Neural Network Based Mobile Robot Path Planning, *IEEE Conference on Intelligent System Design and Applications*, Jinan, China, 16-18 October 2006.

[34] Warren C. W., Global Path Planning Using Artificial Potential Fields, *IEEE International Conference on Robotics and Automation*, Scottsdale, USA, 14-19 May 1989.

[35] Zeng C., Zhang Q., Wei X., Robotic Global Path Planning Based Modified Genetic Algorithm and A* Algorithm, *International Conference on Measuring Technology and Mechatronics Automation*, Shangshai, China, 6-7 February 2011.

[36] Carlos E. J., Federico P. E., Gabriel R. J., The Exact Euclidian Distance Transform: A New Algorithm for Universal Path Planning, *International Journal of Advanced Robotic Systems*, 2013, **10**(266), 1-10.

[37] Canny J. F., A Voronoi Method for the Piano-Movers Problem, *IEEE International Conference on Robotics and Automation*, St. Louis, USA, 25-28 March 1985.

[38] Asano T., Asano T., Guibas L., Hershberger J., Imai H., Visibility-Polygon Search and Euclidean Shortest Path, *Symposium on Foundation of Computer Science*, Portland, USA, 21-23 October 1985.

[39] Santos V., Goncalve J. G. M., Vaz F., Perception Maps for the Local Navigation of a Mobile Robot: A Neural Network Approach, *IEEE International Conference on Robotics and Automation*, San Diego, USA, 8-13 May 1994.

[40] Saska M., Macas M., Preucil L., Lhotska L., Robot Path Planning Using Particle Swarm Optimization of Ferguson Splines, *IEEE Conference on Emerging Technologies and Factory Automation*, Prague, Czech Republic, 20-22 September 2006.

[41] Canny J. F., A New Algebraic Method for Robot Motion Planning and Real Geometry, *IEEE Annual Symposium on Foundation of Computer Science*, Los Angeles, USA, 12-14 October 1987.

[42] Elizondo-Leal J. C., Ramirez-Torres G., An Exact Euclidean Distance Transform for Universal Path Planning, *Electronics, Robotics and Automotive Mechanics Conference*, Cuernavaca, Mexico, 28 September-1 October 2010.

[43] Kavraki L. E., Svestka P., Latombe J. C., Overmars M. H., Probabilistic Roadmaps for Path Planning in High Dimensional Configuration Spaces, *IEEE Transactions on Robotics and Automation*, 1996, **12**(4), 566-580.

[44] Islam F., Nasir J., Malik U., Ayaz Y., Hasan O., RRT Smart Rapid Convergence Implementation of RRT Towards Optimal Solution, *IEEE International Conference on Mechatronics and Automation*, Chengdu, China, 5-8 August 2012.

[45] Vanualailai J., Sharma B., An Asymptotically Stable Collision Avoidance System, *International Journal of Non-linear Mechanics*, 2008, **43**, 925-932.

[46] Xie L., Chen H., Solution to Reinforcement Learning Problems with Artificial Potential Field, *Journal of Central South University of Technology*, 2008, **15**, 552-557.

[47] Park M., Lee M., Real-Time Path Planning in Unknown Environment and a Virtual Hill Concept to Escape Local Minima, *IEEE Conference on Industrial Electronics Society*, Busan, South Korea, 2-6 November 2004.

[48] Chang Y., Yamamoto Y., Online Path Planning Strategy Integrated with Collision and Dead Lock Avoidance Schemes for Wheeled Mobile Robot in Indoor Environment, *Industrial Robot-An International Journal*, 2008, **35**, 421-434.

[49] Mansor M., Morris A., Path Planning in Unknown Environment with Obstacles using Virtual Window, *Journal of Intelligent and Robotic Systems*, 1999, **24**, 235-251.

[50] Hu X., Xu Q., Robot Path Planning based on Artificial Immune Network, *IEEE International Conference on Robotics and Biomimetics*, Sanya, China, 15-18 December 2007.

[51] Willms A., Yang S., Real Time Robot Path Planning via a Distance Propagating Dynamic System with Obstacle Clearance, *IEEE Transactions on System, Man and Cybernetics*, 2008, **38**, 884-893.

[52] Wan T., Chen H., Earnshaw R., Real Time Path Planning for Navigation in Unknown Environment, *International Conference on Theory and Practice of Computer Graphics*, Birmingham, UK, 5 June 2003.

[53] Ying N., Eicher L., Real Time 3D Path Planning for Sensor-Based Underwater Robotics Vehicles in Unknown Environment, *OCEANS 2000 MTSIIEEE Conference and Exhibition*, Providence, USA, 11-14 September 2000.

[54] Choset H. M., *Principles of Robot Motion: Theory, Algorithms and Implementation*, 1st ed., MIT press, Massachusetts, 2005.

[55] Mohamed E. F., El-Metwally K., Hanafy A. R., An Improved Tangent Bug Method Integrated with Artificial Potential Field for Multi Robot Path Planning, *International Symposium on Innovations in Intelligent Systems and Applications*, Istanbul, Turkey, 15-18 June 2011.

[56] Chander V., Asokan T., Ravindran B., A New Multi-Bug Path Planning Algorithm for Robot Navigation in Known Environments, *IEEE Region Conference*, Singapore, Singapore, 22-25 November 2016.

[57] Chestnutt J., Lau M., Footstep Planning for the Honda ASIMO Humanoid, *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 18-22 April 2005.

[58] Wang Y., Sillitoe P., Mulvaney J., Mobile Robot Path Planning in Dynamic Environments, *IEEE International Conference on Robotics and Automation*, Rome, Italy, 10-14 April 2007.

[59] Kiss L., Varkonyi-Koczy A. R., Baranyi N., Autonomous Navigation in a Known Dynamic Environment, *IEEE International Conference on Fuzzy Systems*, St. Louis, USA, 25-28 May 2003.

[60] Zheng T., Zhao X., A Novel Approach for Multiple Mobile Robot Path Planning in Dynamic Unknown Environment, *IEEE Conference on Robotics Automation Mechatronics*, Bangkok, Thailand, 1-3 June 2006.

[61] Lv N., Feng Z., Numerical Potential Field and Ant Colony Optimization Based Path Planning in Dynamic Environment, *IEEE World Congress on Intelligent Control and Automation*, Dalian, China, 21-23 June 2006.

[62] Cao Q., Huan Y., Zhou J., An Evolutionary Artificial Potential Field Algorithm for Dynamic Path Planning of Mobile Robot, *IEEE Conference on Intelligent Robots and System*, Beijing, China, 9-15 October 2006.

[63] Huang L., Velocity Planning for a Mobile Robot to Track a Moving Target: A Potential Field Approach, *Robotics and Autonomous Systems*, 2009, **57**, 55-63.

[64] Hu L., Gu Z., Research and Realization of Optimum Route Planning in Vehicle Navigation Systems based on a Hybrid Genetic Algorithm, *Institution of Mechanical Engineers part D-Journal of Automobile Engineering*, 2008, **22**, 757-763.

[65] Tarokh M., Hybird Intelligent Path Planning for Articulated Rovers in Rough Terrain, *Fuzzy Sets and Systems*, 2008, **159**, 2927-2937.

[66] Stentz A., Optimal and Efficient Path Planning for Partially-Known Environments, *IEEE International Conference on Robotics and Automation*, San Diego, USA, 8-13 May 1994.

[67] Schwager M., Bullo F., Skelly D., Rus D., Ladybug Exploration Strategy for Distributed Adaptive Coverage Control, *IEEE International Conference on Robotics and Automation*, Pasadena, USA, 19-23 May 2008.

[68] Karakaya S., Ocak H., Küçükyıldız G., A Bug-Based Local Path Planning Method for Static and Dynamic Environments, *International Symposium on Innovative Technologies in Engineering and Science*, Valencia, Spain, 3-5 June 2015.

[69] Hua-Qing M., Jin-Hui Z., Xi-Jing Z., Obstacle Avoidance with Multiobjective Optimization by PSO in Dynamic Environment, *International Conference of Machine Learning and Cybernetics*, Guangzhou, China, 18-21 August 2005.

[70] Lu L., Gong D., Robot Path Planning in Unknown Environments Using Particle Swarm Optimization, *International Conference on Natural Computation*, Jinan, China, 18-20 October 2008.

[71] Wu Z., Feng L., Obstacle Prediction-Based Dynamic Path Planning for a Mobile Robot, *International Journal of Advancements in Computing Technology*, 2012, **4**(3), 118-124.

[72] Koryakovskiy I., XuanHoai N., MuLee K., A Genetic Algorithm with Local Map for Path Planning in Dynamic Environments, *Genetic and Evolutionary Computation Conference*, Boston, USA, 8-12 July 2009.

[73] Kesper P., Grinke E., Hesse F., Wörgötter F., Manoonpong P., Obstacle-Gap Detection and Terrain Classification of Walking Robots based on a 2D Laser Range Finder, *International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, Sydney, Australia, 14-17 July 2013.

[74] Kılınç O., Küçükyıldız G., Karakaya S., Ocak H., Kamera ve Lazer Kullanılarak LIDAR Sistemi Geliştirilmesi, *Otomatik Kontrol Ulusal Toplantısı*, Malatya, Türkiye, 26-28 Eylül 2013.

[75] Shina A., Papadakis P., Mind the Gap: Detection and Traversability Analysis of Terrain Gaps Using LIDAR for Safe Robot Navigation, *Robotica*, 2013, **31**(7), 1085-1101.

[76] Larson J., Trivedi M., Lidar Based Off-Road Negative Obstacle Detection and Analyis, *IEEE International Conference on Intelligent Transportation Systems*, Washington, USA, 5-7 October 2011.

[77] Karakaya S., Küçükyıldız G., Ocak H., Bingül Z., Obstacle and Optimal Heading Direction Detection Algorithm on a Mobile Robot Platform, *IEEE International Conference on Signal Processing and Applications*, Muğla, Turkey, 18-20 April 2012.

[78] Surmann H., Nüchter A., Hertzberg J., An Autonomous Mobile Robot with a 3D Laser Range Finder for 3D Exploration and Digitalization of Indoor

Environments, *Robotics and Autonomous Systems Journal*, 2003, **45**(1), 181-198.

[79] Tzafestas S. G., Deliparaschos K. M., Moustris G. P., Fuzzy Logic Path Tracking Control for Autonomous Non-Holonomic Mobile Robots: Design of System on a Chip, *Journal of Robotics and Autonomous Systems*, 2010, **58**(1), 1017-1027.

[80] Masehian E., Naseri M. R., A Tabu Search based Approach for Online Motion Planning, *IEEE Conference on Industrial Technology*, Mumbai, India, 15-17 December 2006.

[81] Glover F., Laguna M., *Tabu Search*, 1st ed., Kluwer Academic Publishers, Dordrecht, 1998.

[82] Blasco P., Diaz F., Vicente S., Muniz D., The Shared Control Dynamic Window Approach for Non-Holonomic Semi-Autonomous Robots, *International Conference of ISR Robotik*, Munich, Germany, 2-3 June 2014.

[83] Marin P., Hussein A., Martin D., Escalera A.,Global and Local Path Planning Study in a ROS-Based Research Platform for Autonomous Vehicles, *Journal of Advanced Transportation*, 2018, **1**, 1-10.

[84] Alves D., Victorino A., An Image Based Dynamic Window Approach for Local Navigation of an Autonomous Vehicle in Urban Environments, *IEEE ICRA Workshop on Modelling Estimation Perception and Control of all Terrain Mobile Robots*, Hong Kong, China, 1 June 2014.

[85] Hightower J., Borriello G., Location Systems for Ubiquitous Computing, *Computer*, 2001, **34**, 57-66.

[86] Liu H., Darabi H., Banerjee P., Liu J., Survey of Wireless Indoor Positioning Techniques and Systems, *IEEE Transactions on Systems Man and Cybernetics Part C (Applications and Reviews)*, 2007, **37**(6), 1067-1080.

[87] Gu Y., Lo A., Niemegeers I., A Survey of Indoor Positioning Systems for Wireless Personal Networks, *IEEE Communications Surveys & Tutorials*, 2009, **11**, 13-32.

[88] Harle R., A Survey of Indoor Inertial Positioning Systems for Pedestrians. *IEEE Communications Surveys & Tutorials*, 2013, **15**, 1281-1293.

[89] Mautz R., Overview of Current Indoor Positioning Systems, *Journal of Geodesy and Cartography*, 2009, **35**(1), 18-22.

[90] Youngsu P., Jewon L., Sangwoo K., Improving Position Estimation on RFID Tag Floor Localization Using RFID Reader Transmission Power Control, *IEEE International Conference on Robotics and Biomimetic*, Bangkok, Thailand, 22-25 February 2009.

[91] Hazas M., Hooper A., A Novel Broadband Ultrasonic Location System for Improved Indoor Positioning, *IEEE Transactions on Mobile Computing*, 2006, **5**(5), 536-547.

[92] Harter A., Hopper A., Steggles P., Ward A., Webster P., The Anatomy of a Context Aware Application, *Annual ACM/IEEE International Conference on Mobile Computing and Networking*, Seattle, USA, 15-19 August 1999.

[93] Priyantha N. B., Chakraborty A., Balakrishnan H., The Cricket Location Support System, *Annual International Conference on Mobile Computing and Networking*, Boston, USA, 6-11 August 2000.

[94] Gezici S., Tian Z., Giannakis G. B., Kobayashi H., Molisch A. F., Poor H. V., Sahinoglu Z., Localization via Ultra-Wideband Radios: A Look at Positioning Aspects for Future Sensor Networks, *IEEE Signal Processing Magazine*, 2005, **22**(4), 70-84.

[95] Zanella A., Best Practice in RSS Measurements and Ranging, *IEEE Communications Surveys & Tutorials*, 2016, **18**(4), 2662-2686.

[96] Yang J., Chen Y., Indoor Localization Using Improved RSS Based Lateration Methods, *IEEE Global Telecommunications Conference on Global Telecommunications*, Honolulu, USA, 30 November-4 December 2009.

[97] Lin X. Y., Ho T. W., Fang C. C., Yen Z. S., Yang B. J., Lai F., A Mobile Indoor Positioning System Based on Ibeacon Technology, *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Milano, Italy, 25-29 August 2015.

[98] Munoz E., Inertial Pocket Navigation System: Unaided 3D Positioning, *Sensors*, 2015, **15**, 9156-9178.

[99] Foxlin E., Pedestrian Tracking with Shoe Mounted Inertial Sensors, *IEEE Computer Graphics and Applications*, 2005, **25**(6), 38-46.

[100] Jimenez A., Granja F., Honorato J., Rosas J., Accurate Pedestrian Indoor Navigation by Tightly Coupling Foot-Mounted IMU and RFID Measurements, *IEEE Transactions on Instrumentation and Measurement*, 2012, **61**(1), 178-189.

[101] Durrant-Whyte H., Bailey T., Simultaneous Localization and Mapping: Part I, *IEEE Robotics & Automation Magazine*, 2006, **13**, 99-110.

[102] Angermann M., Robertson P., FootSLAM: Pedestrian Simultaneous Localization and Mapping without Exteroceptive Sensors Hitchhiking on Human Perception and Cognition, *Proceedings of the IEEE*, 2012, **100**, 1840-1848.

[103] Kaiser S., Diaz E. M., PocketSLAM Based on the Principle of the FootSLAM Algorithm, *International Conference on Location and GNSS*, Gothenburg, Sweden, 22-24 June 2015.

[104] Garcia M., Frassl M., Robertson P., Collaborative Mapping for Pedestrian Navigation in Security Applications, *Security Research Conference on Future Security*, Berlin, Germany, 4-6 September 2012.

[105] Biswas J., Veloso M., Depth Camera Based Indoor Mobile Robot Localization and Navigation, *IEEE International Conference on Robotics and Automation*, St. Paul, USA, 14-18 May 2012.

[106] Ganganath N., Leung H., Mobile Robot Localization Using Odometry and Kinect Sensor, *IEEE International Conference on Emerging Signal Processing Applications*, Las Vegas, USA, 12-14 January 2012.

[107] Choi B. S., Lee J. J., The Position Estimation of Mobile Robot Under Dynamic Environment, *Annual Conference of the IEEE Industrial Electronics Society*, Taipei, Taiwan, 5-8 November 2007.

[108] Xu S., Chou W., An Improved Indoor Localization Method for Mobile Robot Based on WiFi Fingerprint and AMCL, *International Symposium on Computational Intelligence and Design*, Hangzhou, China, 9-10 December 2017.

[109] Mainetti L., Patrono L., Sergi I., A Survey on Indoor Positioning Systems, *IEEE International Conference on Software, Telecommunications and Computer Networks*, Split, Croatia, 17-19 September 2014.

[110] http://anykode.com (Ziyaret tarihi: 1 Mart 2016).

[111] Anh H. P., Huan H. T., Nam N. T., Novel Robust Walking for Biped Robot Using Adaptive Neural PID Controller, *International Conference on Automatic Control Theory and Application*, Melbourne, Australia, 4 September 2014.

[112] Mohammad S., Jeffril M. A., Sariff N., Mobile Robot Obstacle Avoidance by Using Fuzzy Logic, *IEEE International Conference on System Engineering and Technology*, Shah Alam, Malaysia, 19-20 August 2013.

[113] Xu W., Yuan S. C., Cao Y. H., Research on Yinma Palletizing Robot Motion Feasibility Based on MATLAB Robotics Toolbox, *Applied Mechanics and Materials*, 2014, **543**(1), 1296-1300.

[114] http://stdr-simulator-ros-pkg.github.io (Ziyaret tarihi: 11 Ağustos 2017).

[115] Hugues L., Bredeche N., Simbad: An Autonomous Robot Simulation Package for Education and Research, *Simulation of Adaptive Behavior*, 2006, **9**, 831-842.

[116] http://blog.minibloq.org (Ziyaret tarihi: 1 Mart 2016).

[117] Fox D., Burgard W., Thrun S., The Dynamic Window Approach to Collision Avoidance, *IEEE Robotics & Automation Magazine*, 1997, **4**(1), 23-33.

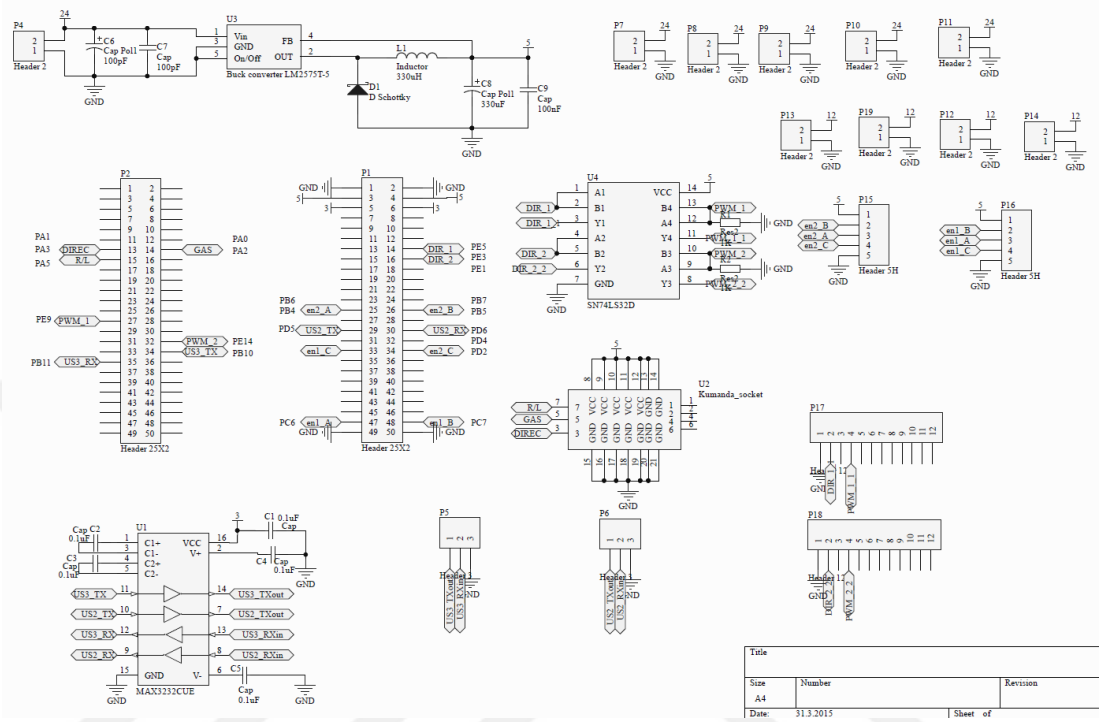
[118] Hsieh C. H., Liu J. S., Nonlinear Model Predictive Control for Wheeled Mobile Robot in Dynamic Environment, *International Conference on Advanced Intelligent Mechatronics*, Kachsiung, Taiwan, 11-14 July 2012.

[119] Karakaya S., Küçükyıldız G., Ocak H., A New Mobile Robot Toolbox for MATLAB, *The Journal of Intelligent and Robotic Systems*, 2017, **6**(1), 22-30.

# EKLER

## Ek-A

# ANA KONTROL KARTININ ELEKTRONİK ŞEMASI



## KİŞİSEL YAYIN VE ESERLER

[1] **Karakaya S**., Küçükyıldız G., Ocak H., One Dimensional Moving Obstacle Detection Method for Mobile Robotics Applications, *26th International Signal Processing and Communications Applications Conference*, İzmir, Turkey, 2-5 May 2018.

[2] Karamustafa B., Küçükyıldız G., **Karakaya S.**, Ocak H., Beacon Tabanlı Konumlandırma Sistemleri için Hızlı Kurulum Algoritması, *International Conference on Data Science and Applications*, Yalova, Turkey, 4-7 October 2018.

[3] Karamustafa B., Küçükyıldız G., **Karakaya S.**, Ocak H., Mobil Robotların Konumlandırılması için BLE ve UWB Tabanlı Teknolojilerin Karşılaştırılması, *International Conference on Data Science and Applications*, Yalova, Turkey, 4-7 October 2018.

[4] Dilaver T., Salı K., Deniz G., Durmaz M., **Karakaya S**., Ocak H., A Novel and Efficient Hexapod System: Robocek, *International Conference on Data Science and Applications*, Yalova, Turkey, 4-7 October 2018.

[5] **Karakaya S**., Küçükyıldız G., Ocak H., A New Mobile Robot Toolbox for MATLAB, *The Journal of Intelligent and Robotic Systems,* 2017, **6**(1), 22-30.

[6] **Karakaya S**., Küçükyıldız G., Ocak H., Point Stabilization Technique Using Model Predictive Control and Exact Euclidian Distance Transform Methods, *The Sigma Journal of Engineering and Natural Sciences,* 2017, **6**(1), 22-30.

[7] **Karakaya S.**, Küçükyıldız G., Ocak H., A Dynamic Path Planning Method for Wheeled Mobile Robots (Dyna-bug), *Global Journal of Computer Sciences: Theory and Research*, 2017, **7**(3), 123-128.

[8] Küçükyıldız G., **Karakaya S**., Ocak H., Classification of Motor Imaginary in EEG Using Random Forest Cassifier, *Global Journal of Computer Sciences: Theory and Research*, 2017, **7**(3), 129-135.

[9] Akkaya U., Şekerci N., Karagöz A., Özay A., **Karakaya S**., Küçükyıldız G., Ocak H., Virtual Reality Based Remote Controlled Mobile Robot Platform, *Global Journal of Computer Sciences: Theory and Research*, 2017, **7**(3), 136-144.

[10] Küçükyıldız G., Ocak H., **Karakaya S**., Sayılı O., Design and Implementation of a Multi Sensor Based Brain Computer Interface for a Robotic Wheelchair, *The Journal of Intelligent and Robotic Systems,* 2017, **5**(1), 17-25.

[11] **Karakaya S.**, Küçükyıldız G., Ocak H., A Dynamic Path Planning Method for Wheeled Mobile Robots (Dyna-bug), *4th Global Conference on Computer Engineering COMENG*, İstanbul, Turkey, 28-30 August 2017.

[12] Küçükyıldız G., **Karakaya S.**, Ocak H., Classification of Motor Imaginary in EEG Using Random Forest Cassifier, *4th Global Conference on Computer Engineering COMENG*, İstanbul, Turkey, 28-30 August 2017.

[13] Akkaya U., Şekerci N., Karagöz A., Özay A., **Karakaya S**., Küçükyıldız G., Ocak H., Virtual Reality Based Remote Controlled Mobile Robot Platform, *4th Global Conference on Computer Engineering COMENG*, İstanbul, Turkey, 28-30 August 2017.

[14] **Karakaya S**., Küçükyıldız G., Ocak H., Detection of Obstacle-Free Gaps for Mobile Robot Applications Using 2-D LIDAR Data, *International Congress on Natural and Engineering Sciences ICNES2016*, Skopje, Macedonia, 24 – 26 September 2016.

[15] **Karakaya S**., Küçükyıldız G., Ocak H., A Hybrid Posture Stabilization Method for Mobile Robots, *International Congress on Academic Researches INES2016*, Antalya, Turkey, 22–25 April 2016.

[16] **Karakaya S**., Küçükyıldız G., Ocak H., Speed Control Adaptation on Static Trajectories, *International Congress on Academic Researches INES2016*, Antalya, Turkey, 22–25 April 2016.

[17] **Karakaya S**., Küçükyıldız G., Ocak H., Detection of Obstacle-Free Gaps for Mobile Robot Applications Using 2-D LIDAR Data, *International Journal of Natural and Engineering Sciences*, 2016, **10**(3), 23-27.

[18] **Karakaya S**., Küçükyıldız G., Ocak H., A Bug-Based Local Path Planning Method for Static and Dynamic Environments, *3. International Symposium on Innovative Technologies in Engineering and Science*, Valensiya, İspanya, 22–24 Eylül 2015.

[19] **Karakaya S**., Ocak H., Küçükyıldız G., Kılınç O., A Hybrid Indoor Localization System Based on Infra-red Imaging and Odometry, *International Conference on Image Processing, Computer Vision & Pattern Recognition*, Las Vegas, USA, 25–27 August 2015.

[20] Küçükyıldız G., **Karakaya S.**, Ocak H., Real Time Control of a Wheel Chair based on EMG and Kinect for the Disabled People, *Medical Technologies Congress*, Muğla, Turkey, 22–24 July 2015.

[21] Küçükyıldız G., **Karakaya S**., Ocak H., Sayılı O., Brain-Computer Interface with Low Cost Commercial EEG Device, *3. International Symposium on Innovative Technologies in Engineering and Science*, Valencia, Spain, 22–24 September 2015.

[22] Ocak H., Ambarkütük H., Küçükyıldız G., **Karakaya S.**, Image Processing Based Package Volume Detection with Kinect Sensor, *Signal Processing and*

*Communications Applications Conference SIU 2015*, Malatya, Turkey, 19–22 May 2015.

[23] **Karakaya S**., Küçükyıldız G., Ocak H., Kinematic Model Based Path Tracking Algorithm For Differential Drive Mobile Robots, *Signal Processing and Communications Applications Conference SIU2014*, Trabzon, Turkey, 19–22 May 2014.

[24] **Karakaya S**., Küçükyıldız G., Toprak C., Ocak H., Image Processing Based Low Cost Range Finder, *IEEE Conference of Mechatronika*, Brno, Czech Repoblic, 21–24 September 2014.

[25] Kılınç O., Küçükyıldız G., **Karakaya S.**, Ocak H., Image Processing Based Indoor Localization System, *Signal Processing and Communications Applications Conference SIU 2014*, Trabzon, Turkey, 19–22 May 2014.

[26] **Karakaya S**., Küçükyıldız G., Ocak H., Bulanık Mantık Hız Kontrolü Destekli Distance Transform Yol Planlama, *Otomatik Kontrol Ulusal Toplantısı TOK2013*, Malatya, Türkiye, 1–4 Mayıs 2013.

[27] **Karakaya S.**, Küçükyıldız G., Ocak H., Image Processing Based Low Cost Range Finder, *International Conference Mechanika,* Kaunas, Lithuania, 1–4 June 2013.

[28] Kılınç O., Küçükyıldız G., **Karakaya S.**, Ocak H., Kamera ve Lazer Kullanılarak Lidar Sistemi Geliştirilmesi, *Otomatik Kontrol Ulusal Toplantısı TOK2013*, Malatya, Türkiye, 1–4 Mayıs 2013.

[29] Özkök İ., Küçükyıldız G., **Karakaya S**., Ocak H., Kinect Tabanlı Robot Kolu Kontrolü, *Otomatik Kontrol Ulusal Toplantısı TOK2013*, Malatya, Türkiye, 1–4 Mayıs 2013.

[30] **Karakaya S**., Küçükyıldız G., Ocak H., Bingül Z., Obstacle and Optimal Heading Direction Detection Algorithm on a Mobile Robot Platform, *Signal Processing and Communications Applications Conference SIU 2012*, Muğla, Turkey, 1–4 April 2012.

## ÖZGEÇMİŞ

Ağustos 1989'da Giresun'da doğmuştur. İlköğretim eğitimine İstanbul Kartal Şeyh Şamil İlköğretim Okulu'nda başlamış ve son üç yılını Sakarya Akyazı J. & J. Konuralp İlköğretim Okulu'nda tamamlamıştır. Ortaöğrenimini 2003-2007 yılları arasında Sakarya Akyazı Anadolu Lisesinde tamamlamıştır. 2007 yılında Kocaeli Üniversitesi Mühendislik Fakültesi Mekatronik Mühendisliği bölümünü kazanmıştır ve 2011 yılında 3.52/4.00 ortalama ile bölüm birincisi olarak mezun olmuştur. 2011 yılı Eylül ayında Kocaeli Üniversitesi Fen Bilimleri Enstitüsü, Mekatronik Mühendisliği Anabilim Dalı'nda Bütünleşik Doktora eğitimine başlamıştır. Aynı tarihte Kocaeli Üniversitesi Mühendislik Fakültesi Mekatronik Mühendisliği Bölümü'nde Araştırma Görevlisi olarak göreve başlamıştır ve halen görevine devam etmektedir.