



T.C.

TOKAT GAZİOSMANPAŞA ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

MEKATRONİK MÜHENDİSLİĞİ ANABİLİM DALI

MEKATRONİK MÜHENDİSLİĞİ YÜKSEK LİSANS

**SÜRÜ ROBOTİĞİNDE YOL PLANLAMA YAKLAŞIMININ
İNCELENMESİ**

YÜKSEK LİSANS TEZİ

Müsemma ALTINDAŞ

Danışman: Doç. Dr. Levent GÖKREM

TOKAT- 2023

ETİK SÖZLEŞMESİ

Tokat Gaziosmanpaşa Üniversitesi Lisansüstü Eğitim Enstitüsü tez yazım kılavuzuna göre, Doç. Dr. Levent GÖKREM danışmanlığında hazırlamış olduğum “Sürü Robotiğinde Yol Planlama Yaklaşımının İncelenmesi”adlı Yüksek Lisans tezinin bilimsel etik değerlere ve kurallara uygun, özgün bir çalışma olduğunu, aksinin tespit edilmesi halinde her türlü yasal yaptırımını kabul edeceğimi beyan ederim.

05/07/2023

MüsemmaALTINDAŞ

JÜRİ KABUL VE ONAY

MÜSEMMA ALTINDAŞ Tarafından hazırlanan “Sürü Robotiğinde Yol Planlama Yaklaşımının İncelenmesi” adlı tez çalışmasının savunma sınavı 05.07.2023 tarihinde yapılmış olup aşağıda verilen Jüri tarafından Oy Birliği ile Tokat Gaziosmanpaşa Üniversitesi Lisansüstü Eğitim Enstitüsü Mekatronik Mühendisliği Anabilim Dalı’nda Yüksek Lisans Tezi olarak kabul edilmiştir.

Jüri Üyeleri

İmzası

Üye (Başkan) : Dr. Öğr. Üyesi Mehmet Serhat CAN

Üye : Doç. Dr. Levent GÖKREM

Üye : Dr. Öğr. Üyesi Oğuz MISIR

ONAY

05/07/2023

Lisansüstü Eğitim Enstitüsü Müdürü

TEŞEKKÜR

Bu tez çalışması süresince emeği, rehberliği ve desteği ile bana her zaman öncü olan danışman hocam Doç. Dr. Levent Görkem'e teşekkürlerimi sunarım. Çalışmalarım süresinde her anımda yanımda olan desteklerini her zaman gösteren anneme, babama, ablalarım, yeğenlerime ve İlker Özmen'e teşekkürlerimi sunarım.



ÖZET

SÜRÜ ROBOTİĞİNDE YOL PLANLAMAYAKLAŞIMININ İNCELENMESİ

ALTINDAŞ, Müsemma

Yüksek Lisans, Eğitim Programları ve Öğretim Bilim Dalı

Tez Danışmanı: Doç. Dr. Levent GÖKREM

Haziran 2023

Bu tez çalışmasında sürü robotiğinde yol planlama davranışı için veüç farklı algoritmaile karşılaştırma yapılmıştır. Sürü robotlarının belirlenen arenalarda başlangıç konumdan hedef konuma ulaşabilmeleri için yol planlama yaklaşımıönerilmiştir. Ayrıca yol planlama sürecinde sürü robotlarının engelden kaçınmasını sağlayacak algoritmalar geliştirilmiştir.Geliştirilen yol planlama yaklaşımını ölçeklenebilirlik ve esneklik açısından değerlendirmek ve performanslarını ölçmek için,VFH, RRT ve PRM algoritmaları ile farklı robot sayılarıylave farklı ileriye bakma mesafelerine göre üç farklı engelli arenalardasistematiik simülasyonlar uygulanmıştır. Geliştirilen algoritmalar ile her bir sürü robotu, üç farklı arenalarda hedef konuma ulaşmak için takip edilecek yola olan ileriye bakma mesafesine bağlı olarak organize bir şekilde sürü davranışı sergilenmektedir. Geliştirilen VFH-PurePursuit tabanlı iş birliği algoritması, sürü robotları engellerden ve robotlardançarpmışmadan kaçınmak için engel veya robotların olmadığı yöne VFH yöntemi ile yönlendirilir. Engel ve robotlara rastlanmadığı zaman önceden belirlenmiş yolu PurePursuitile takip eder. RRT-PurePursuit tabanlı iş birliğine dayalı algortimada ve PRM-PurePursuittabanlı iş birliğine dayalı algortimalarda arenalarda önceden belirlenmiş başlangıç ve hedef konum arasında engelsiz bir yol oluşturulur.Oluşturulan yolun,PurePusuitalgoritması ile takip edilmesi amaçlanmıştır.Yol planlama ve engellerden kaçınma yaklaşımının performansını ölçmek ve karşılaştırmak için literatürde kullanılan dört farklı ölçüt kullanılmıştır.Karşılaşmalar sonucunda, sürü robotları için PRM-PurePursuit tabanlı iş birliğine dayalı engelden kaçınma ve yol planlama yaklaşımı diğer algortimalara göre daha başarılı olduğu görölmüştür.

Anahtar Kelimeler: Yol Planlama, Engelden Kaçma, Sürü robotiği, Mobil Robotlar, Yol takip etme

ABSTRACT

INVESTIGATION OF ROAD PLANNING APPROACH IN SWARM ROBOTICS

ALTINDAŞ, Müsemma

Master's Thesis, Division of Curriculum and Instruction

Advisor: Assoc. Prof. Dr. Levent GÖKREM

June 2023

In this thesis, a comparison was made with three different algorithms for path planning and obstacle avoidance behavior in swarm robotics. A path planning approach has been proposed so that the swarm robots can reach the target location from the starting location in the determined arenas. In addition, algorithms have been developed to enable the robots to avoid obstacles in the path planning process. In order to evaluate the developed path planning approach in terms of scalability and flexibility and to measure its performance, systematic simulations were applied with VFH, RRT and PRM algorithms in three different obstacle arenas with different robot numbers and different looking distances. With the developed algorithms, each swarm robot exhibits swarm behavior in three different arenas in an organized manner depending on the distance to look forward to the path to be followed to reach the target location. The developed VFH-Pure Pursuit-based collaborative algorithm directs the swarm robots to the direction where there are no obstacles or robots with the VFH method to avoid obstacles and collisions with robots. When obstacles and robots are not encountered, it follows the predetermined path with Pure Pursuit. In the RRT-Pure Pursuit-based collaborative algorithm and the PRM-Pure Pursuit-based collaborative algorithms, an unobstructed path is created in the arenas between the predetermined starting and target location. The path created is intended to be followed with the Pure Pursuit algorithm. Four different criteria used in the literature were used to measure and compare the performance of the path planning and obstacle avoidance approach. As a result of the encounters, it was seen that the PRM-Pure Pursuit-based collaborative obstacle avoidance and path planning approach for swarm robots was more successful than other algorithms.

Keywords: Path Planning, Obstacle Avoidance, Swarm robotics, Mobile Robots, Path tracking

İÇİNDEKİLER

ETİK SÖZLEŞMESİ.....	1
JÜRİ KABUL VE ONAY.....	2
TEŞEKKÜR.....	3
ÖZET	4
ABSTRACT	5
İÇİNDEKİLER.....	6
ŞEKİLLER LİSTESİ.....	8
ÇİZELGELER LİSTESİ	13
KISALTMALAR ve SİMGELER	14
1. GİRİŞ	15
2. GENEL BİLGİLER ve LİTERATÜR TARAMASI.....	18
2.1. Olasılıksal Yol Haritası Algoritması (PRM).....	19
2.2. Hızla Keşfedilen Rastgele Ağaç Algoritması (RRT)	20
2.3. Pure Pursuit Algoritması.....	22
2.4. Vektör Alan Histogram (VFH)	23
2.5. Sürü Robot Modelleri.....	24
2.6. Simülasyon.....	26
3. MATERYAL METOD.....	27
3.1. Yol Planlama ve Engelden Kaçınma Problemi	27
3.2. Robot Kinematik Modeli.....	29
3.3. Sürü Robotları İçin Pure Pursuit-VFH İşbirliğine Dayalı Engelden Kaçınma ve Yol Planlama.....	30
3.4. Sürü Robotları İçin Pure Pursuit-RRT İşbirliğine Dayalı Engelden Kaçınma ve Yol Planlama.....	33
3.5. Sürü Robotları için Pure PRM- Pursuit Tabanlı İş Birliğine Dayalı Engelden Kaçınma ve Yol Planlama	34
3.6. Performans Ölçütleri.....	35
3.6.1. Simülasyon Süresi	36
3.6.2. Kat edilen Mesafe.....	36
3.6.3. İzleme Hatası.....	36
3.6.4. En Kısa Yol.....	37
4. BULGULAR ve TARTIŞMA.....	37
4.1.Sürü Robotları için VFH- Pure Pursuit Tabanlı İş Birliğine Dayalı Engelden Kaçınma ve Yol Planlama	37
4.1.1. Simülasyon Kurulumu	38

4.1.2.Simülasyon Sonuçları.....	40
4.2.Sürü Robotları için RRT- Pure Pursuit Tabanlı İş Birliğine Dayalı Engelden Kaçınma ve Yol Planlama	62
4.2.1.Simülasyon Kurulumu	63
4.2.2 Simülasyon Sonuçları.....	64
4.3.Sürü Robotları için PRM- Pure Pursuit Tabanlı İş Birliğine Dayalı Engelden Kaçınma ve Yol Planlama	87
4.3.1.Simülasyon Kurulumu	88
4.3.2 Simülasyon Sonuçları.....	89
4. SONUÇ	114
5. KAYNAKÇA	116



ŞEKİLLER LİSTESİ

Şekil 1.1: Doğada bulunan canlı sürüler (, 2013).....	15
Şekil 2.1: (a)Jasmine , (b) Kilobots (Rubenstein vd., 2012),(c) E-Puck (Nemec vd., 2017), (d)Colias robot sürüsü (Arvin vd., 2014), (e)Mona robot platformu(Bush, 2017)	25
Şekil 3.1: İki tekerlekli diferansiyel tahrikli mobil robot modeli.....	29
Şekil 3.2: Sürü robot modeli	31
Şekil 3.3: ld mesafesinin yol takibine etkisi(<i>Saf Takip Denetleyicisi - MATLAB & Simulink</i> , t.y.).....	31
Şekil 3.4: Pure Pursuit-VFH iş birliğine dayalı engelden kaçınma ve yol planlama akış diyagramı	32
Şekil 3.5: (a) Arena(1), (b) Arena(2ve (c) Arena(3) RRT ile oluşturulan optimal yol sonuçları	34
Şekil 3.6: (a) Arena(1), (b) Arena(2) ve (c) Arena(3) PRM ile oluşturulan optimal yol sonuçları.....	35
Şekil 4.1: 50x50 boyutlarındaki (a) Arena (1), (b) Arena (2) ve (c) Arena (3)	39
Şekil 4.2: Simülasyon deney platformu ve sürü robotu.....	39
Şekil 4.3: İleri bakma mesafesi ld	40
Şekil 4.4: 0.3 birim değer VFH algoritması için uygulanan sistematik test sonuçları ...	41
Şekil 4.5: 0.5 birim değer VFH algoritması için uygulanan sistematik test sonuçları ...	41
Şekil 4.6: 0.8 birim değer VFH algoritması için uygulanan sistematik test sonuçları ...	42
Şekil 4.7: 1 birim değer VFH algoritması için uygulanan sistematik test sonuçları	42
Şekil 4.8: Sürü robotları için VFH- Pure Pursuit tabanlı iş birliğine dayalı engelden kaçınma ve yol planlama algoritmasının Arena (1) için sonuçları.....	43
Şekil 4.9: (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan simülasyon sonuçları	44
Şekil 4.10: (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan simülasyon sonuçları	45
Şekil 4.11: (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan simülasyon sonuçları	46
Şekil 4.12 : (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan izleme hatası simülasyon sonuçları	47

Şekil 4.13 : (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan izleme hatası simülasyon sonuçları	48
Şekil 4.14: (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan izleme hatası simülasyon sonuçları	49
Şekil 4.15: Sürü robotları için VFH- Pure Pursuit tabanlı iş birliğine dayalı engelden kaçınma ve yol planlama algoritmasının Arena (2) için sonuçları.....	50
Şekil 4.15'te arena (2) olarak isimlendirilen 50x50 boyutları için sistematik simülasyonlarda sırasıyla seçilmiş 3,5 ve 7 sürü robotları ile uygulanan örnek deneyler gösterilmiştir. Bu simülasyonlarda yol planlama ve engellere çarpmadan başlangıç konumundan hedef konuma ulaştığı sürü robotlarının deney sonuçları gösterilmiştir. Şekil 4.16: (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan simülasyon sonuçları	50
Şekil 4.17: (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan simülasyon sonuçları	51
Şekil 4.18: (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan simülasyon sonuçları	52
Şekil 4.19: (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan izleme hatası simülasyon sonuçları	53
Şekil 4.20: (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan izleme hatası simülasyon sonuçları	54
Şekil 4.21:(a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan izleme hatası.....	55
Şekil 4.22: Sürü robotları için VFH- Pure Pursuit tabanlı iş birliğine dayalı engelden kaçınma ve yol planlama algoritmasının arena (3) için sonuçları.....	56
Şekil 4.23:(a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan simülasyon sonuçları	57
Şekil 4.24:(a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan simülasyon sonuçları	58
Şekil 4.25: (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan simülasyon sonuçları	59
Şekil 4.26: (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan izleme hatası.....	60
Şekil 4.27: (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan izleme hatası.....	61

Şekil 4.28: (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan izleme hatası.....	62
Şekil 4.29:50x50 boyutlarındaki arenalarda RRT algoritma ile planlanan yollar	64
Şekil 4.30: RRT- Pure Pursuit algoritması için arena (1)'de simülasyon deney sonuçları (a) 3 sürü robotu, (b) 5 sürü robotu ve (c) 7 sürü robotu	65
Şekil 4.31: RRT -Pure Pursuit algoritması arena (1), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan simülasyon sonuçları	67
Şekil 4.32:Pure Pursuit- RRT algoritması arena (1), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan simülasyon sonuçları	67
Şekil 4.33:Pure Pursuit- RRT algoritması arena (1), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan simülasyon sonuçları	68
Şekil 4.34: Pure Pursuit -RRT algoritması arena (1), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan izleme hatası.....	69
Şekil 4.35: Pure Pursuit -RRT algoritması arena (1), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan izleme hatası.....	70
Şekil 4.36:Pure Pursuit -RRT algoritması arena (1), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan izleme hatası.....	71
Şekil 4.37:Pure Pursuit- RRT algoritması için arena (2)'de simülasyon deney sonuçları (a) 3 sürü robotu, (b) 5 sürü robotu ve (c) 7 sürü robotu	72
Şekil 4.38:Pure Pursuit- RRT algoritması arena (2), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan simülasyon sonuçları	73
Şekil 4.39: Pure Pursuit- RRT algoritması arena (2), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan simülasyon sonuçları	74
Şekil 4.40: Pure Pursuit- RRT algoritması arena (2), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan simülasyon sonuçları	75
Şekil 4.41:Pure Pursuit -RRT algoritması arena (2), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan izleme hatası.....	76
Şekil 4.42:Pure Pursuit -RRT algoritması arena (2), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan izleme hatası.....	77
Şekil 4.43:Pure Pursuit -RRT algoritması arena (2), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan izleme hatası.....	78
Şekil 4.44:Pure Pursuit- RRT algoritması için arena (3)'de simülasyon deney sonuçları (a) 3 sürü robotu, (b) 5 sürü robotu ve (c) 7 sürü robotu	79

Şekil 4.45: Pure Pursuit- RRT algoritması arena (3), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan simülasyon sonuçları	79
Şekil 4.46: Pure Pursuit- RRT algoritması arena (3), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan simülasyon sonuçları	80
Şekil 4.47: Pure Pursuit- RRT algoritması arena (3), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan simülasyon sonuçları	81
Şekil 4.48: Pure Pursuit -RRT algoritması arena (3), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan izleme hatası.....	82
Şekil 4.49: Pure Pursuit -RRT algoritması arena (3), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan izleme hatası.....	83
Şekil 4.50: Pure Pursuit -RRT algoritması arena (3), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan izleme hatası.....	84
Şekil 4.51: 0.3 birim değer için RRT algoritması için uygulanan sistematik test sonuçları	85
Şekil 4.52: 0.5 birim değer için RRT algoritması için uygulanan sistematik test sonuçları	85
Şekil 4.53: 0.8 birim değer için RRT algoritması için uygulanan sistematik test sonuçları	86
Şekil 4.54: 1 birim değer için RRT algoritması için uygulanan sistematik test sonuçları	86
Şekil 4.55: 50x50 boyutlarındaki arenalarda PRM algoritma ile planlanan yollar	88
Şekil 4.56: Pure Pursuit- PRM algoritması için arena (1)'de simülasyon deney sonuçları (a) 3 sürü robotu, (b) 5 sürü robotu ve (c) 7 sürü robotu	90
Şekil 4.57: Pure Pursuit- PRM algoritması arena (1), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan simülasyon sonuçları	91
Şekil 4.58: Pure Pursuit- PRM algoritması arena (1), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan simülasyon sonuçları	92
Şekil 4.59: Pure Pursuit- PRM algoritması arena (1), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan simülasyon sonuçları	93
Şekil 4.60: Pure Pursuit -RRT algoritması arena (1), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan izleme hatası deney sonuçları.....	94
Şekil 4.61: Pure Pursuit -RRT algoritması arena (1), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan izleme hatası.....	95
Şekil 4.62: Pure Pursuit -RRT algoritması arena (1), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan izleme hatası.....	96
Şekil 4.63: Pure Pursuit- PRM algoritması için arena (2)'de simülasyon deney sonuçları (a) 3 sürü robotu, (b) 5 sürü robotu ve (c) 7 sürü robotu	97

Şekil 4.64: Pure Pursuit- PRM algoritması arena (2), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan simülasyon sonuçları	98
Şekil 4.65: Pure Pursuit- PRM algoritması arena (2), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan simülasyon sonuçları	99
Şekil 4.66: Pure Pursuit- PRM algoritması arena (2), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan simülasyon sonuçları	100
Şekil 4.67: Pure Pursuit -PRM algoritması arena (2), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan izleme hatası.....	101
Şekil 4.68: Pure Pursuit -PRM algoritması arena (2), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan izleme hatası.....	102
Şekil 4.69: Pure Pursuit -PRM algoritması arena (2), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan izleme hatası.....	102
Şekil 4.70: Pure Pursuit- PRM algoritması için arena (3)'te simülasyon deney sonuçları (a) 3 sürü robotu, (b) 5 sürü robotu ve (c) 7 sürü robotu	103
Şekil 4.71: Pure Pursuit- PRM algoritması arena (3), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan simülasyon sonuçları	105
Şekil 4.72: Pure Pursuit- PRM algoritması arena (3), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan simülasyon sonuçları	106
Şekil 4.73: Pure Pursuit- PRM algoritması arena (3), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan simülasyon sonuçları	107
Şekil 4.74: Pure Pursuit -PRM algoritması arena (3), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan izleme hatası.....	108
Şekil 4.75: Pure Pursuit -PRM algoritması arena (3), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan izleme hatası.....	109
Şekil 4.76: Pure Pursuit -PRM algoritması arena (3), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan izleme hatası.....	110
Şekil 4.77: 0.3 birim değer için PRM algoritması için uygulanan sistematik test sonuçları	111
Şekil 4.78: 0.5 birim değer için PRM algoritması için uygulanan sistematik test sonuçları	112
Şekil 4.79: 0.8 birim değer için PRM algoritması için uygulanan sistematik test sonuçları	112
Şekil 4.80: 1 birim değer için PRM algoritması için uygulanan sistematik test sonuçları	112

ÇİZELGELER LİSTESİ

Çizelge 1: Sürü robotu özellikleri	28
Çizelge 2: VFH Simülasyon özellikleri.....	38
Çizelge 3: RRT simülasyon özellikleri	63
Çizelge 4: Pure Pursuit – RRT algoritma için sürü robotlarının katettikleri ortalama mesafe	87
Çizelge 5: PRM Simülasyon özellikleri	88
Çizelge 6: Pure Pursuit – PRM algoritma için sürü robotlarının katettikleri ortalama mesafe	113



KISALTMALAR ve SİMGELER

Kısaltmalar

Açıklama

ACOKarınca Kolonisi Optimizasyonu

PSOParçacık Sürü Optimizasyonu

PRMOlasılıklı Yol Haritası

RRTHızla Keşfedilen Rastgele Ağaç Algoritması

PIDOrantılı İntegral Türev

VFFSanal Kuvvet Alanı

VFHVektör Alan Histogramı

ld İleriye Bakma Mesafesi

Simgeler

Açıklama

x Robotun yatay pozisyonu

y Robotun dikey pozisyonu

R Robot tekerlek yarıçapı

ω_L Sol tekerlek açısal hızı

ω_R Sağ tekerlek açısal hızı

L Robot çapı

a Robot kinematik modeli

v Robot doğrusal hızı

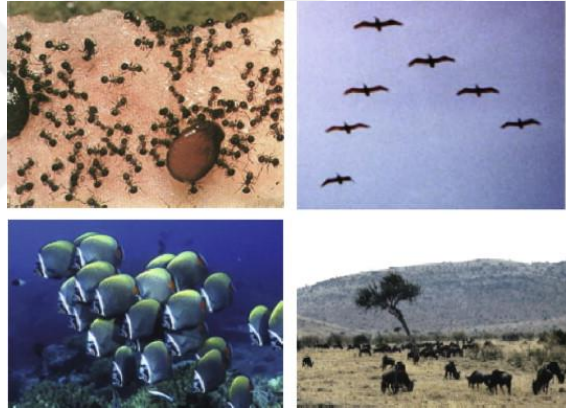
w Açısal hız

TE İzleme hatası

1. GİRİŞ

Robotlar hayatımızın her evresinde yer almaktadır. Savunma sanayi, üretim, sağlık sektörlerinde ve arama kurtarma çalışmalarında yaygın olarak rastlanmaktadır. Ancak robotların fiziksel koşulları robotları karmaşık ve kullanılmaz hale getirmektedir. Karmaşık alanlarda kullanılmak üzere daha kullanışlı ve küçük sürü (swarm) robot sistemleri ortaya çıkmıştır.

Sürü robot sistemleri, belirlenen hedefi karmaşık ve yetenekli robotların görevlendirilmesinin yerine basit özelliklere sahip birçok robotların görevlendirilmesi amaçlanmaktadır. Bu yaklaşım, böcekler, kuşlar ve arılar gibi canlıların sürü davranışlarından ilham almaktadır. (Misir ve Gökrem, 2020)



Şekil 1.1: Doğada bulunan canlı sürüler (Tan Zheng, 2013)

Sürü davranışı, benzer boyutlarda olan ve bir araya kümelenmiş doğadaki canlıların aynı yerde gezmeleri, grup halinde hareket etmeleri ya da aynı yöne doğru göç etmeleri ile sergilenen toplu bir davranıştır. (Levitis vd., 2009) Bu sürü davranışını, sergileyebilmeleri için her robotun çevresini algılayabilen ve kendi kendini organize edebilen basit bir özelliklere sahip olması gerekir. (Misir ve Gökrem, 2021b) Sürü robot sistemleri de doğada bulunan canlı sürülerinin davranışlarından esinlenerek yiyecek arama (foraging) (Campo & Dorigo, 2007), kapsama (coverage) (Panerati vd., 2018), akın etme (flocking), düzen alma (formation), nesnelere etki etme, çoklu hedef gözleme, sürü trafiği ve yol planlama gibi davranışlar modellenmiştir. (Seçkin vd., 2019)

Sürü robotların sürü halinde kullanılması, tek ve karmaşık bir robotun kullanımına göre çeşitli avantajlara sahiptir.

Bu avantajlar şunlardır:

- Kendi kendini koordine edebilme özelliği
- Hata toleransı
- Sağlamlık
- Esneklik
- Ölçeklenebilirlik

Sürü robotlarının hata toleransı avantajı, sürüdeki bir veya birden çok robotun arıza durumuna geçmesi veya sürüden çıkarılmasına rağmen atanan görevin gerçekleştirilmesidir.(Çayirpunar, 2009) Esneklik, sürünün değişken görev şartlarına ve ortam şartlarına uyum sağlayabilmesidir. Sağlamlık, sürü robotlarının atanan görevin gerçekleştirme sürecinde robotların tamamının aktif çalışmayacak durumda olsa dahi görevi gerçekleştirmesidir.(Misir ve Gökrem, 2021b)

Sürü robotiği, yerel etkileşimleri kullanarak çok sayıda robotun kendi kendileri organize ettiği bir alandır. Sürü robotiği belirli bir görevi yerine getirmek için iş birliği içerisinde çalışan birçok bireysel robotu koordine ederek ve etkileşime girerek istenilen işlevleri uygulamaktadır.Sürü robotlarının iş birliği içinde bir görevi yapabilmesi için hem çevreleriyle hem de kendi aralarında iletişim kurabilmeleri için gerekli özelliklere sahip olmalıdırlar. (Mısırd vd., 2020)

Bireysel robotlar genellikle sınırlı yeteneklere sahiptir bu sebeple tek bir robotun göreve ulaşması zordur. Robotlar arasındaki koordinasyon ve iş birliği göreve daha sağlıklı ve kolay ulaşılabilmesini sağlamaktadır. Bu incelemeler neticesinde robotların birbirlerine bütünleşmiş bir şekilde sürü oluşturup sürü iş birliği ile hareket etmesi sürü robotiği alanını ön plana çıkarmaktadır.(Çayirpunar, 2009)

Doğadaki kolektif canlılarda olduğu gibi sürü robotlarının da bir merkezi kontrolcü tarafından kontrol edilmemesi ve tek başına robotun bir görevi üstlenmeyecek yetenekte olması da

benimsenmektedir. (Coppola vd., 2019)(Şahin, 2005)Sürü robotları verilen görevleri yaparken herhangi bir merkezi olmadan sadece yerel bir etkileşim ile kollektif karar vermektedirler.

Sürü robotlarının uygulama alanları son yıllarda oldukça önem arz etmektedir. Tehlikeli ya da ulaşılamayan alanlar için tercih edilir hale gelmiştir. Sürü robotik araştırmalarındaki değinilen problemler iki sınıfa ayrılmaktadır. İlk olarak toplanma, konumlanma, kendi kendine organize edebilme ve kapsam alanı gibi modellere dayanmaktadır. Diğer problem ise çevredeki varlıklara dayanmaktadır, örnek olarak hedef aramak, (Stormont, 2005) yiyecek aramak ve arama- kurtarma çalışmaları uygulama kapsamaları mevcuttur.(Tan ve Zheng, 2013)

Sürü robotik sistemlerin kullanışlı hale getirilebilmesi ve basit hale indirgenebilmesi için sürü robotlarının birbirleri ile değişen çevre koşullarına göre hareket etmesi ve iletişim halinde olması gerekmektedir. Robotların hareketleri haberleşme özellikleri ile gerçekleşmektedir. (Coppola vd., 2019)Ortamdaki robotlara ve engellere çarpmadan kaçınması için bir engel algılama algoritmaları bulunur.

Mobil robotların çoklu robot sisteminde koordineli bir şekilde iş birliği içerisinde çalışmaktadırlar. Sürü robotların uygulama alanlarında en önemli görevlerinden biri yol planlama ve engellerden kaçınmadır.(Mısıır, 2023) Bir mobil robot için yol planlama başlangıç konumundan hedef konuma engelleri aşarak gideceği yolun belirlenmesidir.Mobil robotların yol planlama algoritmaları, robotu engelsiz en kısa ve en hızlı biçimde başlangıç konumdan hedef konumuna varmasına sağlamaktadır. Robotun izleyeceği yol haritası ne kadar karmaşık ise, robotun engelsiz yolu bulması ve hedefe ulaşmasını zorlaştıracak olup ve sonuç olarak maliyetli olacaktır. (Mısıır, 2023)Bu engeli aşmak için, daha verimli yol planlama algoritması oluşturmak gerekmektedir. Robotlara kesin ve maliyetsiz bir yol planladığımızda, daha verimli ve daha kısa sürede hiçbir engele çarpmadan hedef konuma ulaşacaktır. (Hosseini, 2016)Dolayısıyla, bu tez çalışmasında çoklu mobil robotlarda engelden kaçınarak yol planlama algoritmaları incelenmektedir.

Yol planlamada2 temel problem mevcuttur.Birincisi başlangıç ve hedef noktaları arasındaki optimum yolun oluşturulmasıdır. Yolun planlandığı ortamda herhangi bir engel var ise çarpışma olmaksızın robotun hedefe ulaşması da diğer bir problemdir. Bu problemlerini çözüme ulaştırmak için iki önemli özelliğe dikkat edilmesi gerekir. Bunlar yol planlama çevrenin özellikleri ve yol planlama algoritmalarıdır. Mobil robotların yol planlaması ve

engelden kaçınması oldukça karmaşık ve zor bir problemdir. Statik bir çevrede mobil robotun dışında her şey sabittir ve yol planlama önceden belirlenebilir. Dinamik çevrelerde ise robot ve robotun dışındaki her şey hareketli olabilir. Yol planlama algoritmaları global ve yerel yol planlama olmak üzere de sınıflandırılabilirler. Global yol planlama algoritması, arena hakkında tam bilgi gerektirir ve arena statik yapıda olmalıdır. Bu yaklaşımda, robotlar harekete başlamadan önce, başlangıç noktasından hedef noktaya kadar olan bütün yolu üretir. Yerel yol planlama ise yol bulma robot hareket halinde iken gerçekleşir. Çevresel değişimler söz konusu olduğunda robot yeni bir yol üretecektir. (Garip, 2018)

Literatürde mobil robot üzerine yapılan çalışmalar konumlandırma, çevredeki nesneleri algılama, gezinme, engellere çarpmadan kaçınma hedefe ulaşma ve yol planlama şeklinde olduğu görülmektedir. Mobil robot çalışma konularını çevrenin belirlenmesi, yol planlama ve konumlandırma olarak sınıflandırmak mümkündür. Bu tez çalışmasının amacı, mobil robotların yol planlama ve engelden kaçınma problemlerine ilişkin çözümler üretmektir.

Mobil robotlar çevre koşullarını algılayıp bu koşullara uygun şekilde hareket etmesini ve engelden kaçınmasını sağlamalıdır. Bu problemi çözüm sağlamak için pek çok çalışma gerçekleştirilmiştir. Bu çalışmalar 1969 yılında Stanford üniversitesi “Shakey” isimli hareket planlama yapabilen mobil bir robot sistemi ile başlamıştır. (Yaşar, 2020) 1979 yılında Lozano ve Wesley mobil robotlara güvenli bir yol planlamak için çarpışmadan kaçınma algoritmasına öncülük etmiştir. (Lozano-Pérez ve Wesley, 1979)

Günümüze kadar birçok algoritma geliştirilmiştir. Bu algoritmalar Sezgisel (heuristic) algoritmalar ve klasik (conventional) şeklinde 2 grupta incelenebilir. Sezgisel algoritmalar genetik algoritmalar, ACO (Karıncalar Kolonisi Optimizasyonu), PSO (Parçacık Sürü Optimizasyonu) vb. algoritmalar örnek olarak verilebilir. Klasik algoritmalar ise Potansiyel alanlar (Potential Fields), PRM (Probabilistic Roadmap), RRT (Rapidly Exploring Random Tree) ve böcek (bug) algoritmaları örnek verilebilir. Klasik algoritmaların avantajı kesin olarak hesaplamasıdır ancak yavaş hesaplamaktadır.

2. GENEL BİLGİLER ve LİTERATÜR TARAMASI

Sürü robotikte robotların bir noktadan bir başka noktaya varması ile ilgili problem yol planlaması olarak tanımlanır. Mobil robotlar kapalı bir ortamda gezinerek atanan bir görevi

yerine getirirler. Bir robottan ortamın haritasını oluřturması, haritadaki yerini kesin olarak belirlemesi ve engellere arpmadan belirlenen hedefe varması istenmektedir. Yol planlama ařamasındaki uygun kořullar, robot hızı, engellerden ve robot arpıřmalarından kaınma, robot kinematięi ve dinamięine uygun yrnge planlaması gibi kriterleri iermektedir (Chen vd., 2018).

Yol planlama algoritmaları global ve yerel yol planlama olmak zere de sınıflandırılabilirler. Global yol planlama algoritmalarında robot hari tm evre bilgileri sabittir. Bu nedenle robotlar harekete bařlanamadan yol belirlenebilir. Ancak deęiřken evre řartlarında ve dinamik engellerde robot yol planlamada yetersizdir. Yerel yol planlama algoritmaları evre kořullarını nceden bilmeye gerek duymaz. Yol bulma robot hareket halinde iken gerekleřir. evresel deęiřimler sz konusu olduęunda robot yeni bir yol retecektir. Hareketli ortamlarda verimli alıřırlar. Hedeflenen konum uzakta iken yntem verimsizleřir. Normal olarak her iki yntemin iř birlięi ierisinde alıřtırılması daha verimli olacaktır. (Bi vd., 2008)(H. Zhang vd., 2012)

2.1. Olasılıksal Yol Haritası Algoritması (PRM)

Bir mobil robotun, alıřma ortamı engellerinde bařlangı konumdan hedef konuma giden yola sahip olması gerekir. Karmařık bir ortamda ne olduęunu, engellerin o statik ortamda nereye yerleřtirildięini bilmek sorunlardan biridir. Bu sorunu ozmek iin pek ok algoritma yntemi vardır ve yaklařımlardan biri Olasılıksal Yol Haritasıdır (PRM). PRM engellerden kaınan bir mobil robotunPRM eřzamanlı haritalaması ve navigasyonu, bařlangı noktasından hedef konuma varması ve optimize etmeye alıřan robotik alanında bir yol planlama algoritmasıdır. (Alarabi vd., 2022)

PRM (ProbabilisticRoadmap) algoritması, belirlenen bir haritada engellerden kaınarak robotun bařlangı konumundan hedef konumuna varması iin haritadaki olası yolların aę grafięidir. PRM yol planlayıcısı 2 blmden oluřmaktadır. Bu algoritma ęrenme ařaması ve sorgulama ařamasından oluřmaktadır.Belirlenen harita zerinde, rastgele noktalar oluřturulur. Algoritma retilen noktaların boř alanda olması iin kontrol eder.Boř alanda ise aę grafięine eklenir.Yeni oluřan noktalar en yakın noktalara baęlanarak, baęlantının boř alanda olup olmadıęı kontrol edilir. Bořalanda ise baęlantı grafięine eklenir. Bu iřlem hedef konuma ulařana kadar devam eder. Sorgulama ařamasında, bařlangı ve hedef konumları grafięe

bağlanır yol bir Dijkstra'nın en kısa yol sorgusuyla elde edilir. Rastgele oluşan noktaların sayısı artıkça, PRM algoritmasının yol bulamama olasılığı sıfıra yaklaşır. (Dr. Lydia E. Kavraki, 2013)

Gang ve Wang, 2015 çalışmalarında mobil robot yol planlamasında optimum yolu hesaplamak için PRM (olasılıklı yol haritası yöntemi) algoritması kullanılmaktadırlar. Geliştirilmiş düğüm yöntemi ve geometrik yöntemeye dayalı bir planlama algoritması önerilmiştir. Düğüm geliştirme yöntemi, temel PRM algoritması ile ilk planlamayı optimize etmek için kullanılır, orijinal yol düğümleri kademeli olarak bazı yeni düğümlerle değiştirilir ve yoldaki bükülme noktalarının sayısı büyük ölçüde azaltılır, böylece yol uzunluğu azalır. Aynı zamanda, planlanmış yolu yumuşatmak için yeni bir geometrik optimizasyon stratejisi kullanılır, yol yumuşatma elde edilir. Çalışma sonucunda algoritmanın optimum yolun uzunluğunu azaltabileceğini ve yolun düzgünlüğünün büyük ölçüde geliştiği görülmektedir. (Gang ve Wang, 2015)

Sanchez, 2022 ve arkadaşlarının yaptığı çalışmanın amacı, holonomik olmayan yol planlaması için yarı rastgele örnekleme tekniklerinin kullanımını karşılaştırmaktır. Deneyler, olasılıklı yol haritası yöntemleri (PRM) bağlamında değerlendirilir. PRM tabanlı planlayıcıların yarı rastgele iki yöntem önerilmiştir. Birinci yöntem olarak yarı rastgele örnekleme klasik bir PRM ve ikinci yöntem olarak yarı rastgele tembel bir PRM planlayıcılar önerilmiştir. (Sanchez vd., 2022)

Alarabi, 2022 ve arkadaşları geliştirdikleri yaklaşımda, engellerden kaçınmak ve bilinen bir ortamda bir hedef konuma gitmek için PRM kullanarak robot yolu planlama yeteneğine sahiptir. Önerilen PRM tabanlı yaklaşım, karmaşıklığı artan simülasyon ortamlarında test edilmiş ve diğer genetik algoritma yöntemleriyle karşılaştırılmıştır. Karşılaştırma sonuçları, PRM'nin mesafe ve dönüşlere göre çeşitli ortam türleri için uygun yollar bulabildiğini göstermektedir. Bu yaklaşımla elde edilen sonuçlar, PRM algoritmasının çok etkili ve verimli olduğunu göstermektedir. Algoritmalar, bilinen bir ortamda diğer yaklaşımlara kıyasla çok kısa bir mesafede engellerden kaçınarak bir optimal yol bulabilmektedir. (Alarabi vd., 2022)

2.2. Hızla Keşfedilen Rastgele Ağaç Algoritması (RRT)

Bir mobil robotun yol planlaması için kullanılan algoritmalarından biri de Hızla Keşfedilen Rastgele Ağaç (RRT) algoritmasıdır. Belirlen haritada rastgele örnekler kullanılarak haritadaki

boşlukları verimli bir şekilde aramak için RRT algoritması, Steven M. LaValle ve James J. Kuffner Jr. tarafından geliştirilmiştir.(Nemec vd., 2017)(Randomized Kinodynamic Planning - Steven M. LaValle, James J. Kuffner, 2001.)RRT, belirli bir durum uzayından rastgele alınan örneklerden artımlı olarak bir arama ağacı oluşturan ağaç tabanlı bir hareket planlayıcıdır. Ağaç sonunda arama alanını kapsar ve başlangıç konumundan hedef konuma bir yol oluşturur. (Randomized Kinodynamic Planning- Steven M. LaValle, James J. Kuffner, 2001.)RRT başlangıç konumundan hedef konumuna yol alırken rastgele noktalar seçerek bu düğümlere dallanır ve hedef konuma belli bir mesafede yaklaşıncaya dek bu işlemi tekrarlayan bir algoritmadır. RRT diğer hareket planlayıcı algoritmalarından farklı olarak bir sonraki düğüm seçiminde daha önce bulunmuş düğümleri işleme katmamaktadır. Yakındaki bir düğümü hesaba katmaktansa harita üzerinde rastgele bir düğüm seçilmekte ve bu düğüme yakın, daha önce bulunmuş farklı bir düğümden hareket edilmektedir. Ağacın genişlemesi, başlangıçtan bitiş konumuna kadar tek yönlüdür.(Garip, 2018)

Yijing Li, 2021'nin yaptığı çalışmasında, hareketli engellere ve ortama RRT tabanlı bir yol planlama algoritması sunmaktadır. Bu çalışmada hedef bilgisi veya hareketli engel bilgisi önceden belirlenmemiştir. Gerçek zamanlı algoritma, engellerden kaçınma yaklaşımına ve holonomik olmayan engellere dayalı olarak yeni yol noktaları belirler. Yalnızca tüm yolun yeniden planlanması ve hareket eden engellerin hesaplanmasını azaltmaktadır ve aynı zamanda orijinal yoldan planlanmış yola nispeten kısa bir kayma sağlar. Bu yol planlama stratejisini kullanarak robot, karmaşık hareketli bir ortamda daha az hesaplama maliyetiyle hedefe kendi kendine gidebilir.(Li, 2021)

Gao Zhang, 2020 ve arkadaşları tarafından geliştirilen bir RRT algoritması önerilmiştir. Geliştirilen algoritma, temel RRT algoritmasının optimal olmayan yol, yerel minimuma düşme kolaylığı ve karmaşık ortamlarda başarı oranı artırmak için kullanılmıştır. Bu yöntem, rastgele ağacı içeriden dışarıya doğru büyümeye zorlamak için komşuluk genişletme yöntemini kullanır, bu da RRT* algoritmasında son derece zaman alan yeniden kablolama sürecini ve en uygun yolu sağlamayı önler. Rastgele ağacın büyümesini yönlendirmek için sezgisel metot kullanılır, bu da rastgele ağacın kapsamını azaltır ve planlama verimliliğini artırır. Vebu geliştirilen algoritma farklı haritalarda uygulanarak gözlemlenmiştir.(W. Zhang vd., 2020)

Mobil robotlar yol planlama için kullanılan RRT algoritması kullanıldığında arama süresi ve optimum olmayan oluşturulmuş yol sorunları ele almak için bir RRT algoritması geliştirilmiştir. Geliştirilen algoritma uyarlanabilir örnekleme alanı ve düğüm yaklaşımı ile birleştirilmiş açgözlü stratejiye dayalı geliştirilmiş bir RRT algoritması önermektedir. Öncelikli olarak düğüm genişletme stratejisini iyileştirmek için RRT algoritmasının temelinde açgözlü fikir ortaya atılır, ikinci olarak, örnekleme alanı, arama süresini optimize etmek ve verimliliği artırmak için uyarlamalı örnekleme faktörü getirilerek sınırlandırılır. Son geliştirme ise, yolun gereksiz düğümlerini kaldırarak ve düğümlerin engellere yaklaşmasını sağlamak için optimizasyon algoritmasını kullanarak yol uzunluğu daha da optimize edilir.(Zhou vd., 2022)

2.3.PurePursuit Algoritması

PurePursuit bir yol takip algoritmasıdır. PurePursuit mantığı ve kullanımı basittir. Uzakta belirlenen bir noktaya odaklanarak o noktaya hareket eden ve yörüngeyi takip eden bir algoritmadır. Robotu hareket ettiren açısal hız hesaplanarak robotu mevcut konumdan ileriye bakma mesafesine ulaştırır. Doğrusal hızın sabit olduğu varsayılır, bu nedenle robotun doğrusal hızı değişebilir. Algoritma daha sonra robotun mevcut konumuna bağlı olarak yoldaki ileri bakma noktasını hedef noktasına kadar hareket ettirir. Kısaca bu algoritma robotun belirlenen bir yolu takip etmesidir yani robotun sürekli olarak ileriye bakma mesafesini izlemesidir.(Coulter, R., 1990)

Yol planlama ve takip sürecinde PurePursuit algoritmasının problemlerini ele almak için HuangTian, 2020 ve arkadaşları tarafından bir çalışma yapılmıştır. Yol planlama hatası problemini hedefleyerek, ileri bakma mesafesini hesaplamak için robotun hızı ve yol bilgisi kullanılır. PID denetleyicisi ile birleştirilmiş PurePursuit yöntem tarafından hesaplanan hedef yönlendirme açısı optimize eder ve yol planlamayı uygulamak için sabit açı çıkışı verilmektedir. Bu çalışma sonucunda yol hatasını azalmıştır ve kararlı kontrol açısı verilmektedir.(Huang vd., 2020)

Otonom araçların, burada yol izleme problemlerinin ele alındığı çeşitli çalışma alanları mevcuttur. Yol izleme kontrolü, yanal ve uzunlamasına kontrol olarak sınıflandırılabilir. Bu çalışmada Selvaraj, Lal ve Vive PurePursuit algoritması kullanılarak yanal veya direksiyon

açısı kontrolü ele alınmaktadır. Bu algoritmanın etkinliği farklı eğrilik yolları için test edilmiştir. (Lal vd., 2017)

Mobil robotlar alanında son yıllarda yaygın olarak kullanılan PurePursuit algoritmasını geliştirilmiştir. PurePursuit yönteminde ileriye bakma mesafesi belirlenerek yolu takip eder. Ancak ileri bakma mesafesi sabit olarak seçildiği için yanlış seçilme ihtimalinde düşük performanslı yol izleme neden olacaktır. Bu problemi çözmek için Horvarth ve ark. çoklu hedef noktalarının seçilmesine dayalı ileriye bakma mesafesinin dinamik değişimini önerir, böylece referans yörüngeye daha çok uyan bir eğriliği hedefler. (Horvath vd., 2019)

2.4. Vektör Alan Histogram (VFH)

Mobil robot sistemlerinde engellerden ve çarpışmalardan kaçınarak yol planlamak en önemli görevlerden biridir. Bu nedenle mobil robotlar için Johann Borenstein ve Yoram Koren tarafından gerçek zamanlı bir Sanal Kuvvet Alan (VFF) algoritması geliştirilmiştir. Sunulan bu algoritma, hedefe doğru hareket eden bir mobil robotun bilinmeyen engelleri tespit ederek çarpışmalardan kaçınmasını sağlamak için robot (Jie vd., 2010) hücre, robot üzerinde sanal bir kuvvet oluşturuyor. Oluşan sanal kuvvetler doğrultusunda mobil robot kuvvetin düşük olduğu yönde dönüş açısı hesaplayarak engellere çarpmadan ilerlemeyi hedeflemektedir. VFF algoritma da robot birbirine yakın engeller arasında hareket edememektedir. Bu problemi çözmek için Borenstein ve Koren VFF algoritmasını geliştirerek VFH algoritmasını önermişlerdir. (Mısır vd., 2022) VFH, engellerden ve çarpışmalar için, hesaplama açısından verimli ve sağlam olma amaçlarıyla geliştirilmiştir. VFH algoritmasının, çok engelli ortamlarda uygulanırken hızlı ve güvenilir olduğu ispatlanmıştır. (Ulrich ve Borenstein, 2000)

Vektör alan histogramı (VFH), mobil robotları bulunduran ortamı temsil etmek için kullanılır ve bir sonraki hareket yönü, her yönün maliyet fonksiyonuna göre seçilir. Histogram, robot bakış açısından görülen engel yoğunluğunu gösterir. Mümkün olan her yönde engellerin yoğunluğu sürekli olarak hesaplanır. Robotun hareket yönü, engel yoğunluğunun düşük olduğu yöndedir. (Mısır vd., 2022) VFH algoritması, robotun engellerden kaçınması ve bir hedef yönü takip etmesi için bir hareket yönü oluşturmak için, robot boyutunu ve dönüş yarıçapını hesaba katar. (Ulrich ve Borenstein, 1998)

Yim ve Park, 2014 yapıldıkları çalışmada çeşitli ortamlarda VFH yöntemi ile mobil robot yol planlama analizleri yapılmıştır. VFH algoritmasını analiz etmek için sektör sayısını, robot hızlarını ve farklı engelli ortamlarda simülasyonlar gerçekleştirilmiştir. (Yim ve Park, 2014)

USARSim ortamında mobil robotlara engelden kaçınma özelliğinin uygulamak için araştırma yapılmıştır. Amacı, robot yolda bir engelle karşılaştığında minimum hareketi yapmasının istenilmesidir. Bu problemi çözmek için VFH algoritması kullanılmıştır. Deneyler sonucunda robotların engelden kaçınmada başarılı bir yöntem olduğu ortaya çıkmıştır. Ve VFH yöntemi ile robotun engellerden kaçınarak otonom gezinim yöntemine göre harita içerisinde daha çok dolaştığı gözlemlenmiştir. (Sahin vd., 2014)

Bu makalede VFH algoritmasını geliştirmek amaçlı mobil robotlar için yerel yol planlaması ve engellerden kaçınma için geliştirilmiş bir IVFH*(Real-time dynamic obstacle avoidance for mobile robots) algoritması sunar. Sunulan algoritmada iyileştirme amaçlı değişiklikler yapılmıştır. Bu değişikliklerden birisi dar alanda kapasiteyi genişletmek için yeniden eşik değerleri belirler. Algoritma, robotun yerel engellerden kaçınma yeteneklerini bir dereceye kadar geliştirerek, çevreye uyum sağlama yeteneğini de geliştiriyor. Simülasyon deneyleri, algoritmanın dinamik engellerden kaçınmayı etkili bir şekilde önleyebildiğini göstermektedir. (Jie vd., 2010)

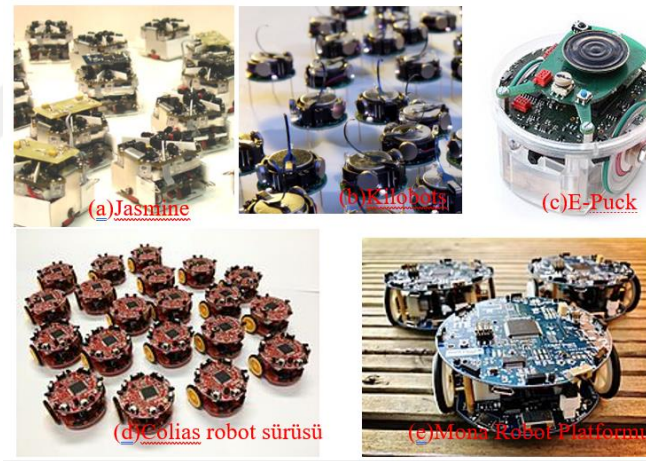
(Qu vd., 2015) Farklı ortamlarda mobil aracın başarımını iyileştirmek için sınırlı bir VFH algoritması sunulmaktadır. Hareket planlamasının güvenliğini artırmak için alçak geçiren bir filtre ve iyileştirilmiş bir maliyet fonksiyonu sunulmuştur. Sunulan algoritmada yapılan deneyler sonucunda VFH algoritmasının çarpışmasız yörüngelerin verimli bir şekilde üretilmesini gösterilmektedir.

2.5. Sürü Robot Modelleri

Sürü robot sistemleri, bir görevi karmaşık ve üstün özelliklere sahip bir robotun yapmasından ziyade basit özelliklere sahip birçok robot ile yapılmasını amaçlamaktadır. Bu sistemler, böcekler, kuşlar ve arılar gibi canlıların sürü davranışlarından ilham almaktadır. (Misir ve Gökrem, 2020)

Tek bir robot için temel davranışlar, belirlenen hedefe hareket ve engellerden kaçınma vb. karmaşık görevleri gerçekleştirmek için bu davranışları verimli bir şekilde uygulamayabilmek için kullanılır. Sürü robotik davranışları ise hayvanlar tarafından sergilenen doğal sürü zekasından ilham alırlar. Akın etme, yiyecek arama vb. bu tür davranışlara örnektir. Akın etme davranışı, robotların bir gruptaki bir hedefe doğru hareketidir. Bu davranışı uygulamak için robotlar, sürü arkadaşlarını tespit etmek ve çarpışmayı önlemek için sensörlere sahip olmalıdır. (Anoopve Kanakasabapathy, 2017)

Sürü robot modellerinde tek bir görevi gerçekleştirmek yerine karmaşık görevleri de yerine getirmek ön planda tutulmaktadır. (Olaronke vd., 2020) Sürü robotları, karmaşık görevi gerçekleştirmek için tek bir robottan daha çok avantaj sağlarlar.(Rooban vd., 2022) Bu avantajı sürdürebilmek için ise robot tasarımları basitleştirerek porta tiplleşmiştir.Şekil 2.1’de literatürde yer alan sürü robot çalışmalarında kullanılan robot modeli örnekleri gösterilmektedir.



Şekil 2.1: (a)Jasmine , (b) Kilobots (Rubenstein vd., 2012),(c) E-Puck (Nemec vd., 2017), (d)Colias robot sürüsü (Arvin vd., 2014), (e)Mona robot platformu(Bush, 2017)

Kolibot,ölçeklenebilir çalışmalara sahip düşük maliyetli bir robottur. Bu robot, sürü robotları üzerinde kolektif algoritmaların test edilmesini içintasarlanmıştır. Bu robotlar düşük maliyetli olmakla birlikte, diğer kollektif robotlara benzer yeteneklere sahiptir. (Rubenstein vd., 2012)Jasmine,bir sürü robot platformudur.Doğadan ilham alansürü algoritmalarını gerçek fiziksel donanımda analiz etmek için Jasmine robot sürüsü kullanılmıştır.(Schmickl vd., 2009)

E-puck robotları robot sürülerini eğitim amaçlı tasarlanmıştır, ancak herhangi birçok aracılı robotik sistemde kullanılabilir. Her mobil robot, ses sensörleri (mikrofon, hoparlör), uzaklık sensörleri gibi mikrodenetleyiciler ile donatılarak sürü robot davranışları uygulanabilmektedir.(Nemec vd., 2017)

Mona robot platformu,sürekli sürü robot için test amaçlı tasarlanmıştır. Kablosuz iletişim gibi platformun üzerine eklenmiş ek modülün yerleştirilmesiyle oluşan modüler bir tasarımıdır. Son olarak Mona robot platformu eğitim ve araştırma amaçlı geliştirilmiştir.(Bush, 2017)

Colias,robotik eğitim ve araştırma için geliştirilmiş bir platform olarak tasarlanmıştır. Hem bireysel hem de sürü sistemlerinde test edilmiştir ve gözlemlenen sonuçlar, mikro boyutlu bir mobil robot olarak ve robot sürüsü uygulamaları için düşük maliyetli bir platform olarak kullanımının uygulanabilirliğini göstermektedir.(Arvin vd., 2014)

2.6.Simülasyon

Simülatörler, matematiksel modeller gibi soyutlama ile fiziksel robotların kullanıldığı gerçek doğrulama arasında bir ara adımdır. Simülasyonlar, sürü robot sistemlerini analiz etmek için uygulamaları hızlı bir şekilde yapmak hem de maliyetten tasarruf etmek için en çok kullanılan araçlardandır. Robot türlerine çeşitli amaçlara göre simüle etmek için pek çok türde platform ve algoritma vardır. Sürüleri tasarlamak, analiz etmek ve karşılaştırmak için özel algoritmalara sahip simülatörler vardır.(Calderón-Arce vd., 2022) Bu simülatörlere, Matlab (Mobile RoboticToolbox),TeamBots,Swarm-bots, Gazebo, USARSim ve ARGoS örnek olarak gösterilebilir.

3. MATERYAL METOD

Bu bölümde, sürü robotlarının birliğine içerisinde yol planlama ve engelden kaçınma algoritmaları incelenmektedir. Yol planlama ve engelden kaçınma problemi için VFH (Vektör Alan Histogram), RRT (Rastgele Ağaç Algoritması), PRM (Olasılıklı Yol Haritası) ve tüm algoritmalarda yol takibi için Pure Pursuit algoritmaları kullanılmıştır. İşbirliğine dayalı yol planlama ve engellerden kaçınma problemi için her bir sürü robotunun davranışını belirleyen bir yaklaşım önerilmektedir. İlk olarak önerilen algoritma ve metotların uygulanabileceği mobil robot kinematiği açıklanmaktadır.

3.1. Yol Planlama ve Engelden Kaçınma Problemi

Bu bölümde yol planlama ve engelden kaçınma davranışı açıklanmakta ve geliştirilen algoritmalar tarafından karşılaştırılacaktır. Sürü robotlarının yol planlama ve engelden kaçınma davranışı, 50X50 boyutundaki engellerle donatılmış 3 farklı arena içerisinde uygulanmaktadır. Burada amaç engellerle donatılmış bir arenada başlangıç konumundan hedef konuma güvenli bir şekilde ulaşabilmesi için etkin yol planlama algoritmasına sahip olması gerekir. İyi bir yol planlama algoritmasında üç ana amaç olması beklenir. Bunlar, düşük maliyetli hesaplama süresi, en kısa yol mesafesi ve engelsiz yoldur.

Sürü robotlarının, algılanan engellerden kaçınarak belirli arenanın dışına çıkamaması, statik engelleri algılayarak arena içerisinde hedef konuma göre hareket etmesi istenir. Arena içerisinde n kadar sürü robotu bulunur. Bu sürü robotları 1'den n'e kadar numaralandırılarak isimlendirilir. Sürü robotlarının arena içerisinde engelleri ve robotları algılayabilmek için sensörlerle donatılmıştır. Bu sensörler ve görevleri ve varsayımlar aşağıda belirtildiği gibidir:

- Sürü robotları 360° görüş alanına sahiptir.

- Sürü robotlarının kinematik modeli holonomik olmayan 2 tekerlekli ve robot yarıçapı 0,8 birim olarak belirlenmiştir.
- Sürü robotlarının doğrusal hızı 0.6'dır ve açısal hızı 0.75 olarak belirlenmiştir.
- Simülasyon örnekleme zamanı her senaryoda aynı olmalıdır.
- Arena alanını kenarlarını engel olarak algılanır.
- Bir sürü robotu herhangi bir engelle veya robotla karşılaştığında rastgele hareket ederek hedef konuma ilerlemektedir.
- Sürü robotların ileriye bakma mesafesine değiştirilerek yol takibinin etkileri gözlemlenmektedir.
- Sürü robotlarının yol planlama başlangıç konumları ve hedef konumları belirli ve her senaryoda aynı olmalıdır.
- Sürü robotlarının katettiği yol mesafeleri hesaplanabilir.
- Sürü robotlarının robot algılama sensör limiti 10 olarak belirlenmiştir.
- Sürü robotlarının engel algılama sensör limiti 10 olarak belirlenmiştir.

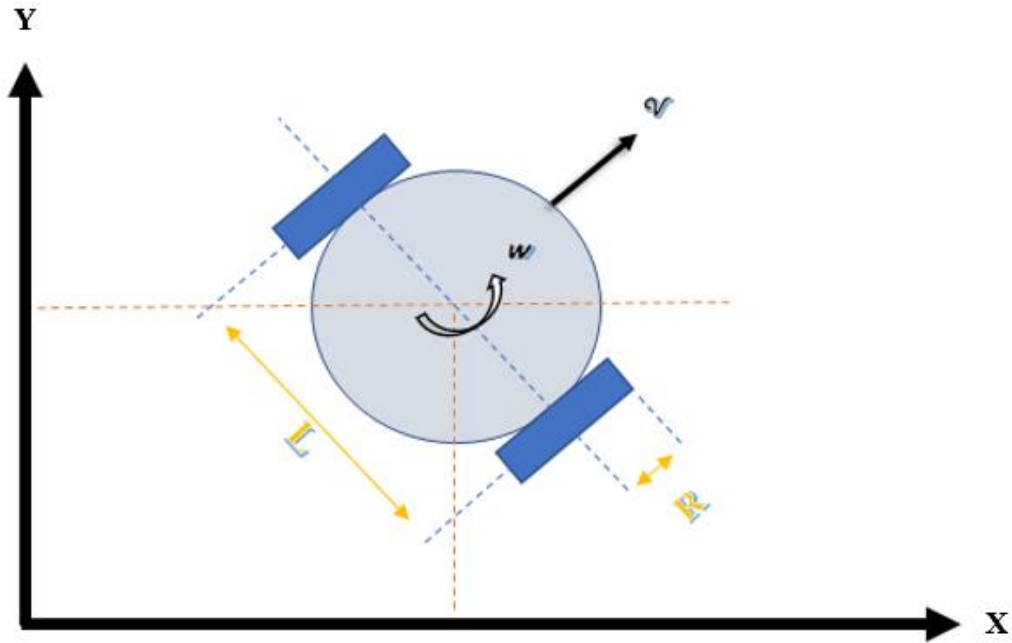
Sürü robotlarının özellikleri Çizelge 1'de belirtilmektedir.

Özellikleri	Ölçüm Aralıkları
Robot Modeli	DifferentialDrive
Robot Algılama açısı	360°
Robot Algılama uzaklığı	10 birim
Engel Algılama uzaklığı	10 birim
Engel Algılama açısı	360°

Çizelge 1: Sürü robotu özellikleri

3.2. Robot Kinematik Modeli

Bu çalışmada mobil robot modellerinden holonomik olmayan, iki tekerlekli diferansiyel tahrikli mobil robot referans alınmıştır. Şekil 3.1 de mobil robotun kinematik modeli gösterilmektedir. Şekil 3.1 de gösterildiği gibi $\frac{L}{2}$ yarıçapında dairesel bir yapıya sahiptir. Robot merkezinde $\frac{L}{2}$ uzağına yerleştirilmiş R yarıçapında paralel olarak gerçekleştirilmiş iki tekerleğe sahiptir. L ifadesi sürü robotunun iki tekerlek arası mesafesidir. ω ise açısal hızı ifade etmektedir.



Şekil 3.1:

İki
tekerlekli
diferansi
yel
tahrikli
mobil
robot
modeli

Sürü
robotu
Denkle
m
3.1'de

koordinatları temsil eder ve Denklem 3.2 de ise α ifadesi kullanılarak robot kinematik model gösterilmiştir.

$$\alpha = [x, y, \theta]^K (3.1)$$

$$\dot{\alpha} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega \quad (3.2)$$

Şekil 3.1 de görüldüğü gibi v , robotun doğrusal hızı ω ise açısal hızı olarak tanımlanmaktadır. Sürü robotu kinematik modelinde görüldüğü gibi iki tekerleğin açısal ve doğrusal hızları arasında bir ilişki bulunmaktadır. Tekerlekler için açısal ve doğrusal hızlar sağ (ω_R)- sol (ω_L) tekerlek olarak ayrılır. Sırasıyla aşağıdaki denklemlerde görülmektedir.

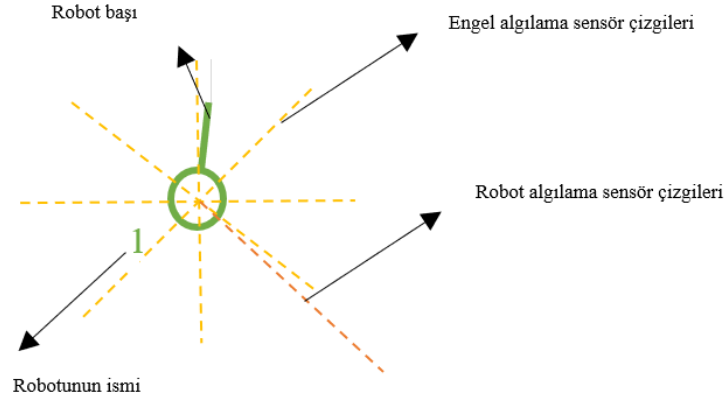
$$v = \frac{R (\omega_L + \omega_R)}{2} \quad \omega = \frac{R (\omega_L - \omega_R)}{L} \quad (3.3)$$

Sürü robotları ω_R ve ω_L açısal hızlarına bağlı olarak robotun yönünü belirlemektedir.

$$\omega_R = \frac{v + (\frac{L}{2})\omega}{R} \quad \omega_L = \frac{v - (\frac{L}{2})\omega}{R} \quad (3.4)$$

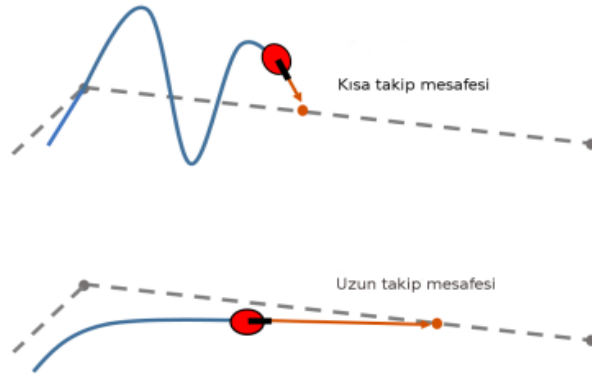
3.3.Sürü Robotları İçin PurePursuit-VFH İşbirliğine Dayalı Engelden Kaçınma ve Yol Planlama

Bu bölümde engelden kaçınma ve yol planlama davranışı için geliştirilen bir algoritma açıklanmaktadır. Bu algoritmada robot hareketi Bölüm 3.1’de belirtilen varsayımlar doğrultusunda gerçekleştirilmiştir. Geliştirilen algoritmalarda kullanılan bir sürü robotunun modeli Şekil 3.2’de gösterilmiştir. Turuncu kesik çizgilerle belirtilen sensör çizgilerinin taradığı alan hem engel hem de robot algılama alanını ifade etmektedir. Sürü robotlarını ayırt edebilmek için numaralandırılarak isimlendirilir.



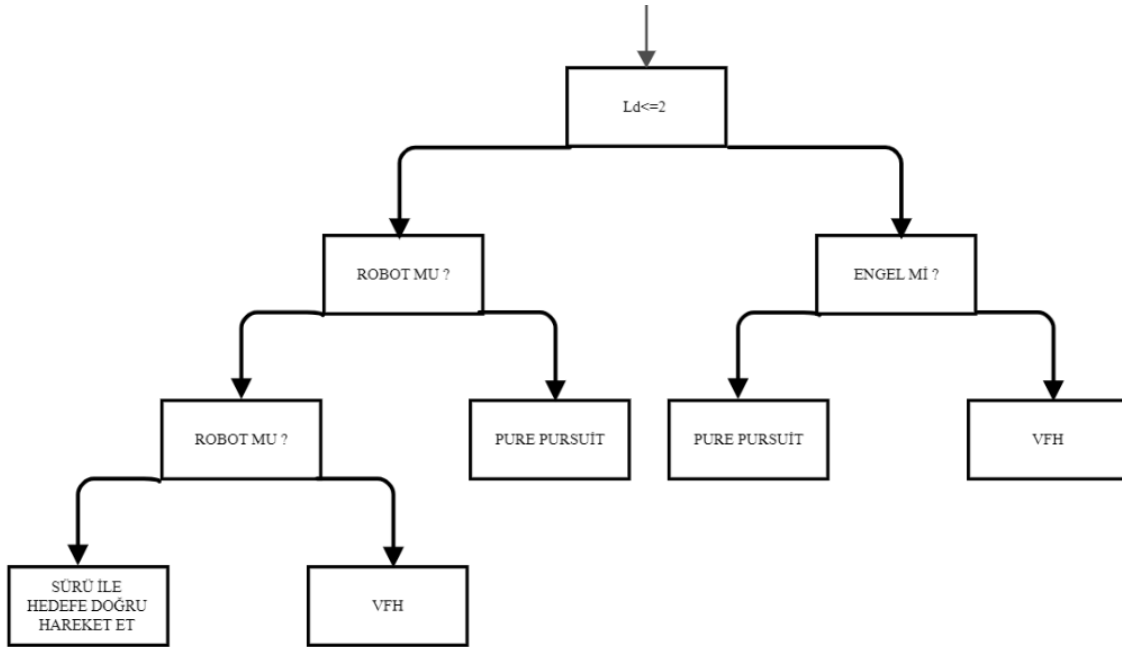
Şekil 3.2: Sürü robot modeli

PurePursuit, geometrik bir yol izleme denetleyicisidir. Yol izleme denetleyicisi, sürü robot kinematiğini ve referans yolunu izleyerek hedefe ulaşmaya çalışan bir denetleyicidir. PurePursuit sabit bir mesafe olan bir ileriye bakma mesafesini (l_d) kullanır. Robotun direksiyon açısı kullanılarak belirlenen hedef noktaya ilerlemektedir. Belirlenen ileriye bakma mesafeleri değiştirilerek algoritmalar gözlemlenmiştir. Bu mesafenin artması köşe noktalarından daha kavisli geçmesini, azaltılması ise daha dalgalı şekilde yolun takip edilmesine sebep olur.



Şekil 3.3: l_d mesafesinin yol takibine etkisi (Saf Takip Denetleyicisi - MATLAB & Simulink, t.y.)

Pure-Pursuit algoritmasında yol planlama ve takibi için ana parametre ileriye bakma mesafesidir (l_d). l_d ileriye bakma mesafesi, robotun mevcut konumdan yol boyunca ne kadar uzağa bakması gerektiğidir. Yolu takip eden aracın l_d yönüne bağlı olarak yön ve hız belirlenir. Pure-Pursuit ve VFH kullanan iş birliğine dayalı yol planlama algoritması, sürü robotlarının başlangıç konumdan hedef konuma organize bir şekilde hareket etmesini sağlar. Önerilen yaklaşım ile sürü robotlarının sayı değişimine bağlı olarak önceden belirlenmiş başlangıç konumundan hedef konumuna hareket eder. Yolları düzenli bir şekilde takip eden sürü robotları, çevredeki engellere ve koşullara bağlı olarak esnek davranışlar sergileyebilmektedir. Şekil 3.4'te PurePursuit-VFH iş birliğine dayalı engelden kaçınma ve yol planlama akış diyagramı gösterilmektedir.



Şekil 3.4: PurePursuit-VFH iş birliğine dayalı engelden kaçınma ve yol planlama akış diyagramı

Geliştirilen algoritmada, sürü robotlarının her biri l_d 'ye bağlı olarak nasıl bir yol izleyeceğine karar verir. Her robot, l_d 'nin 2 birimden az olup olmadığını kontrol eder. Sürü robotunun l_d değeri 2 birimden fazla ise çevresinde çarpışma mesafesinde bir engel veya robot olup olmadığını kontrol eder. Sürü robotu bu şartlar altında bir engel veya robotla karşılaşır, engel veya robotun olmadığı yöne VFH yöntemi ile yönlendirilir. Sürü robotu herhangi bir engelle veya robotla karşılaşmıyorsa Pure-Pursuit yöntemi ile yolu takip eder. Sürü robotunun l_d değeri 2 birimden küçük ise robot algılama alanında robot olup olmadığına bakılır. Sürü robotunun algılama alanında robot veya robotlar varsa bu robotların yönüne yaklaşır. Sürü

robotu bu durumda bir engelle veya robotla çarpışacak kadar yaklaşırsa VFH algoritması ile engelden uzaklaşır. Aksi durumda Pure-Pursuit algoritması ile önceden belirlenmiş yolu takip eder.

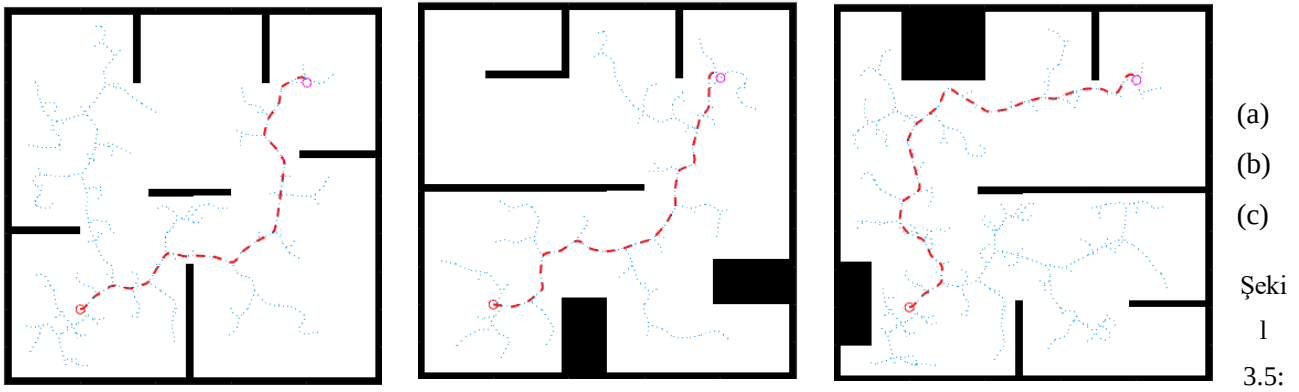
3.4.Sürü Robotları İçin PurePursuit-RRT İşbirliğine Dayalı Engelden Kaçınma ve Yol Planlama

Bu bölümde, RRT-PurePursuit algoritması açıklanmaktadır. RRT, belirli bir durum uzayından rastgele alınan örneklerden artımlı olarak bir arama ağacı oluşturan ağaç tabanlı bir hareket planlayıcıdır. (Randomized Kinodynamic Planning - Steven M. LaValle, James J. Kuffner, 2001.) Bu hareket planlayıcısı başlangıç konumunu düğüm olarak seçer ve ardından kendisine en yakın noktayla birleştirir. Böylece bu iki nokta arasında bir yol oluşturur. Oluşturulan yolun arena içerisinde engellere çarpmadığı kontrol edilir. Engelsiz bir yol oluşturulmuş ise nokta ağaca eklenir. Yolun engellerle karşılaşması sonucunda, oluşturulan yolda hareket mümkün olmayacağına karar verilir. Süreç tekrar başlar, yeni bir nokta seçilerek hesaplamalar yapılır. Hedef konuma ulaşılan kadar bu işlem tekrarlanır. Ağaç sonunda arama alanını kapsar ve başlangıç konumundan hedef konuma bir yol oluşturur. (Randomized Kinodynamic Planning - Steven M. LaValle, James J. Kuffner, 2001,)

RRT algoritmasının amacı, başlangıç konumundan itibaren rastgele noktalar seçerek yollar oluşturur, oluşturulan yollara göre hedef konuma giden en uygun rotayı oluşturmaktır. Bu amaçla arena içerisinde seçilen noktayı seçilmiş olanların dışında ve hedef yakın engelsiz alanlardan seçmesi işlem kolaylığı sağlar.

Geliştirilen PurePursuit-RRT iş birliğine dayalı engelden kaçınma ve yol planlama algoritması başlangıç ve hedef konumları aynı olan 3 farklı arena içerisinde deneyler gerçekleştirilmiştir. Bu algoritma ile arenada başlangıç ve konum noktaları atanmaktadır. Bu noktalar, startPose [10 10 0] ve goalPose [40 40 $-\pi/2$] olarak belirtilmiştir. Bu noktalar arasında optimal engelsiz bir yol oluşturulur. Sürü robotlarının her biri Pure-Pursuit yöntemi ile RRT algoritmasıyla oluşturulan yolu takip etmektedir. Sürü robotlarının her biri 4 farklı l_d değerine bağlı olarak nasıl bir yol izleyeceği gözlemlenmiştir.

Şekil 3.5’ te RRT algoritması ile oluşturulan optimal yollar gösterilmektedir.



(a) Arena (1), (b) Arena (2) ve (c) Arena(3) RRT ile oluşturulan optimal yol sonuçları

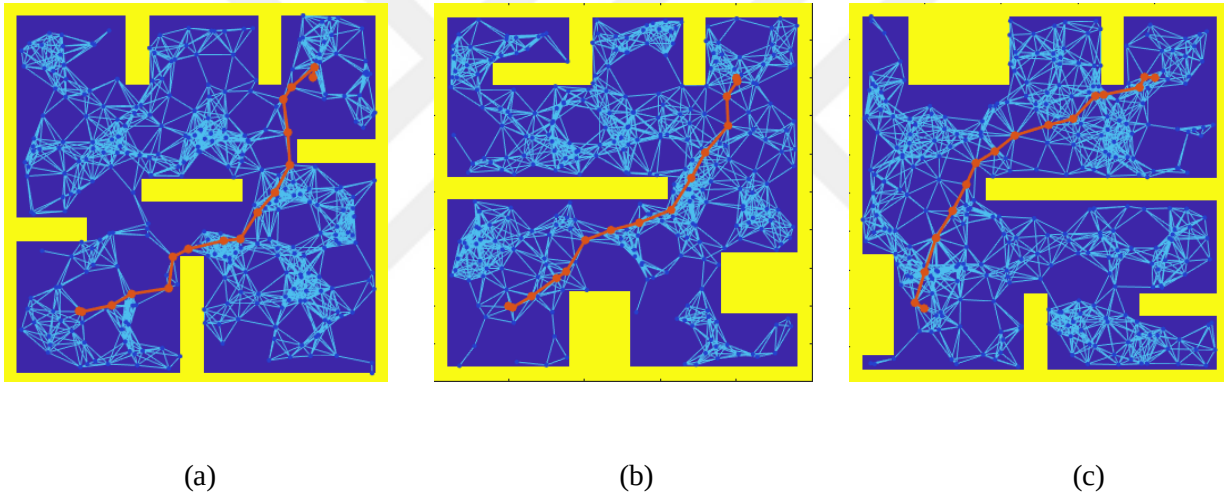
3.5.Sürü Robotları için PurePRM- Pursuit Tabanlı İşBirliğine Dayalı Engelden Kaçınma ve Yol Planlama

Bölüm 3.4'te açıklanan RRT algoritması, başlangıç konumundan optimal yolları bulmak için avantajlıydı. Ancak her yeni konum verildiğinde, tamamen yeni bir ağaç (grafik) oluşturulması gerekmektedir. Başlangıç ve hedef konum arasında tek engelsiz bir yol bulmak için tekrar tekrar yeni bir grafik oluşturmak verimsizdir. Bu verimsiz çalışmayı gidermek için alternatif bir yaklaşım olan, Olasılıksal Yol Haritası (PRM) algoritması geliştirilmiştir. PRM, yeni bir grafik oluşturmak yerine, arenanın tamamını kapsayarak tek bir yol haritası oluşturur. Oluşturulan yol haritası boyunca sürü robotlarının hedef konuma varması için bir en kısa yol planlama algoritması kullanılır.

PRM (Probabilistic Roadmap) algoritması, belirlenen bir arenada engellerden kaçınarak robotun başlangıç konumundan hedef konumuna varması için haritadaki olası yolların ağ grafiğidir. PRM yol planlayıcısı 2 bölümden oluşmaktadır. Bu algoritma öğrenme aşaması ve sorgulama aşamasından oluşmaktadır. Belirlenen harita üzerinde, rastgele noktalar oluşturulur. Algoritma üretilen noktaların boş alanda olması için kontrol eder. Boş alanda ise ağ grafiğine eklenir. Yeni oluşan noktalar en yakın noktalara bağlanarak, bağlantının boş alanda olup olmadığı kontrol edilir. Boş alanda ise bağlantı grafiğine eklenir. Bu işlem hedef konuma ulaşana kadar devam eder. Sorgulama aşamasında, başlangıç ve hedef konumları grafiğe bağlanır. Robotun hareketi için çarpışmasız bir yol Dijkstra'nın en kısa yol sorgusuyla elde edilir. Rastgele oluşan noktaların sayısı arttıkça, PRM algoritmanın yol bulama olasılığı sıfıra yaklaşır. (Park ve Yoon, 2018)

Geliştirilen PurePursuit- PRM algoritması ile 50X50 3 farklı arena içerisinde PRM algoritması ile arena içerisinde sürü robotlarının güvenlik faktörü olarak arena içerisindeki engeller robot yarıçapı kadar şişilir.Şişirilen arenada içerisinde rastgele 300 adet düğümler oluşturulur. Oluşturulan noktaların boş alanda olup olmadığı kontrol edilir. Rastgele oluşturulan düğümler arasında belirtilen ConnectionDistance değerine göre bağlantılar oluşturulur. Geliştirilen algoritmada bu değer 5 olarak belirtilmiştir. Başlangıç konumu [10 10], hedef konum ise [40 40] olarak oluşturulmuştur.

PRM algoritması ile oluşturulan engelsiz en kısa yolu sürü robotları PurePursuit algoritması ile yolu izleyip hedef konuma ulaşacaktır. RRT-PurePursuit algoritmasında uygulandığı gibi sürü robotlarının her biri 4 farklı l_d değerine bağlı olarak oluşturulan yolu izleyeceği gözlemlenmiştir.Şekil 3.6’da PRM algoritması ile oluşturulan yollar gösterilmiştir.



Şekil 3.6: (a) Arena (1), (b) Arena(2) ve (c) Arena(3) PRM ile oluşturulan optimal yol sonuçları

3.6.Performans Ölçütleri

Sürü robotlarının yol planlama ve engellerden kaçınma problemlerini ve algoritmalarının performanslarını incelemek için sürü robotlarının hedefe ulaştıkları süre(yineleme sayısı), farklı yol planlama ortamlarında simülasyon deneylerinde hedefe olan mesafe(arena içerisinde hedefe ulaşırken katedilen yol), sürü robotlarının izleme hatası, hedef ile başlangıç konum arasındaki en kısa yol, başlangıç ve hedef konuma dayalı önceden belirlenmiş yol izlenerek, farklı sayıda sürü robotları ve PurePursuit algoritması ile belirlenen değerlere göre deneyler

incelenmiştir. Deneyler 0,3, 0,5, 0,8 ve 1 birim l_d değerleri ile 3, 5 ve 7 robot için farklı sistematik uygulamaları içermektedir. Deneyler yoluyla elde edilen sonuçlarda, sürü robotlarının ölçeklenebilirliği ve esnekliği yol izleme ve engelden kaçınma davranışı açısından incelenmiştir.

3.6.1. Simülasyon Süresi

Geliştirilen algoritmalarda performans ölçütleri arasında deneyin gerçekleşmesi süresi deneyin performansını ölçecektir. Sürü robotlarının bulundukları ortamlar değişiklik gösterdikçe robotların güvenli bir şekilde hareket etmesi zorlaşır ve algoritmaların hesaplamamaliyetlerinde değişiklikler oluşur ve algoritmaların çalışma süreleri değişmektedir. Bu nedenle sürü robotları belirlenen bir arena içerisinde başlangıç konumdan hedef konuma ulaştıkların deneyler sonlanmış olur. Bu deneylerin sonlandırılmasın kadar geçen süre (yineleme sayısı) deneyini performansını ölçecektir.

3.6.2. Katedilen Mesafe

Geliştirilen algoritmalarda robotların başlangıç konumdan hedef konuma varması için katedilen yoldaki her sıralı durum çifti arasındaki mesafeleri toplayarak yolun toplam uzunluğunu hesaplanabilir. Yani sürü robotlarının başlangıç konumdan hedef konuma ulaşana kadar katettiği yolun gerçek uzunluğudur.

$$PathLength = curPose_{xy} - curPose \quad (3.5)$$

Burada $curPose_{xy}$ sürü robotlarının güncel pozisyonudur. $curPose$ ise sürü robotlarının başlangıç pozisyonudur.

3.6.3. İzleme Hatası

Bu çalışmada geliştirilen algoritmalar ile sürü robotlarının izleme hataları gözlemlenecektir. İzleme hatası, sürü robotunun planlanan yoldan geçerken bir engel ile karşılaştıklarında izleme hatası olarak gözlemlenir. Sürü robotlarının bir engelle veya keskin bir dönüş ile karşılaştıklarında takip edecekleri yola olan mesafesi artar. Dönen ve engellerden kaçan her sürü robotu izleyecekleri yola daha da yakınlaşmak için organize bir şekilde hareket

etmektedir. Denklem 3.6'da ifade edilen izleme hatası, sürü robotunun izleme yoluna dik mesafesidir.(Mısıır vd., 2022)

$$TE = \frac{\tan(\delta_{pp}(t)) \cdot l_d^2}{2L} \quad (3.6)$$

3.6.4. En Kısa Yol

En kısa yol problemi, belirlenen arenalar üzerinde başlangıç ve hedef konum arasında oluşan kenarların ağırlıklarının toplamı en aza indirilecek şekilde bir yol bulma problemidir. (Ahuja vd., 1990) geliştirilen algoritmalarda Dijkstra'nın en kısa yol sorgusuyla bir ölçüt hesaplanmış ve oluşturulan haritalarda ve algoritmalarda karşılaştırılmıştır.

4. BULGULAR ve TARTIŞMA

Bu bölümde, sürü robotları için PurePursuit- VFH iş birliğine dayalı engelden kaçınma ve yol planlama, PurePursuit- RRT iş birliğine dayalı engelden kaçınma ve yol planlama, PurePursuit- PRM iş birliğine dayalı engelden kaçınma ve yol planlama algoritmalarının sonuçları açıklanmaktadır. Geliştirilen metotlar MATLAB, “Mobile RoboticsSimulationToolbox” ortamında simüle edilmiştir. “Mobile RoboticSimulationToolbox” sürü robotu uygulamaları ve çoklu robot sistemlerinin geliştirilmesi için kullanılabilen bir araç kutusudur.

4.1.Sürü Robotları için VFH- PurePursuit Tabanlı İş Birliğine Dayalı Engelden Kaçınma ve Yol Planlama

Geliştirilen sürü robotları için PurePursuit- VFH iş birliğine dayalı engelden kaçınma ve yol planlama algoritması için Çizelge 2’deki gibi simülasyon ortamında sistematik deneyler uygulanmıştır. Sistematik deneyler arena boyutu, robot algılama yarıçapı, l_d değerine, robot sayısına bağlı olarak belirlenmiştir. Uygulanan deneylerindeki amaç sürü robotlarının, değişen arenalar (engeller) ve değişen l_d değerleri ile ölçeklenebilir

bir şekilde farklı robot sayılarında yol planlama ve engelden kaçınma davranışları uygulanabilir.

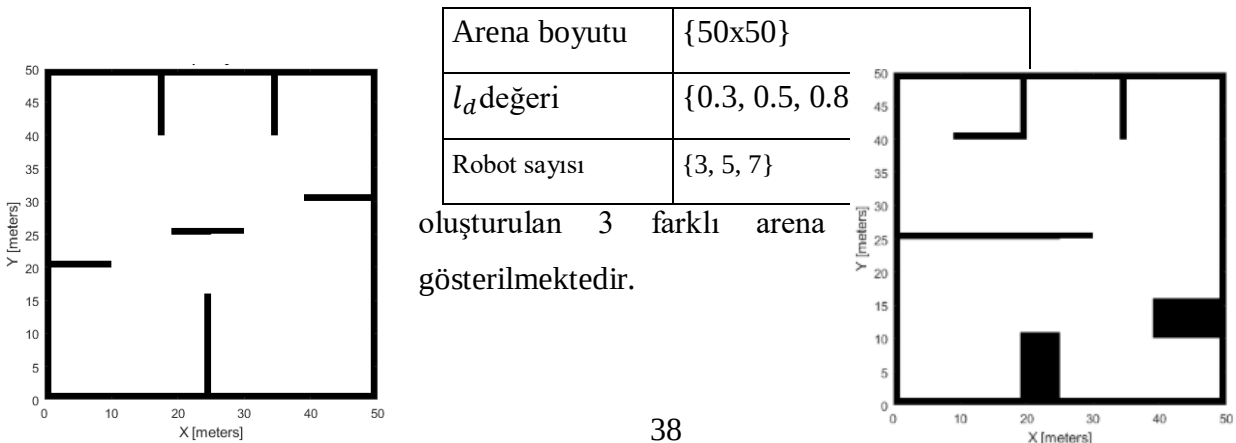
4.1.1. Simülasyon Kurulumu

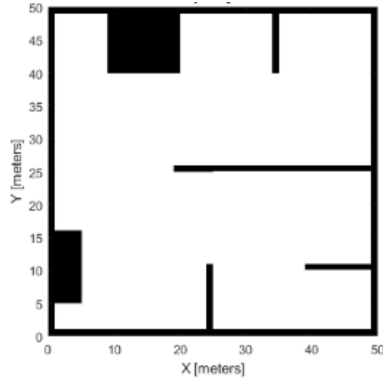
Çizelge 2’de için PurePursuit- VFH iş birliğine dayalı engelden kaçınma ve yol planlama algoritmasının simülasyon kurulumu

Çizelge 2: VFH Simülasyon özellikleri

Çizelge 2’de belirlenen robot sayıları {3, 5, 7}, PurePursuit yöntemi için l_d değeri {0.3, 0.5, 0.8, 1} ve arena boyutları aynıdır fakat 50x50 boyutlarında farklı arenalarda bağlı olarak 36 sistematik deney uygulanmıştır. Yol planlama ve engelden kaçınma probleminin performansını ölçmek için simülasyon süresi kullanılmıştır. Simülasyon süresi, deneylerin başlangıç konumundan hedef konuma ulaştığı ana kadar geçen simülasyon adım sayısıdır. Sistematik deneyler neticesinde elde edilen sonuçlarda, sürürobotlarının ölçeklenebilirliği ve esnekliği başlangıç ve hedef konum arasında engelsiz yol oluşturma ve PurePursuit ile yol izleme davranışı açısından incelenmiştir.

Şekil 4.1’de 50x50 boyutunda





(a)

(b)

(c)

Şekil 4.1: 50x50 boyutlarındaki (a) Arena (1), (b) Arena (2) ve (c) Arena

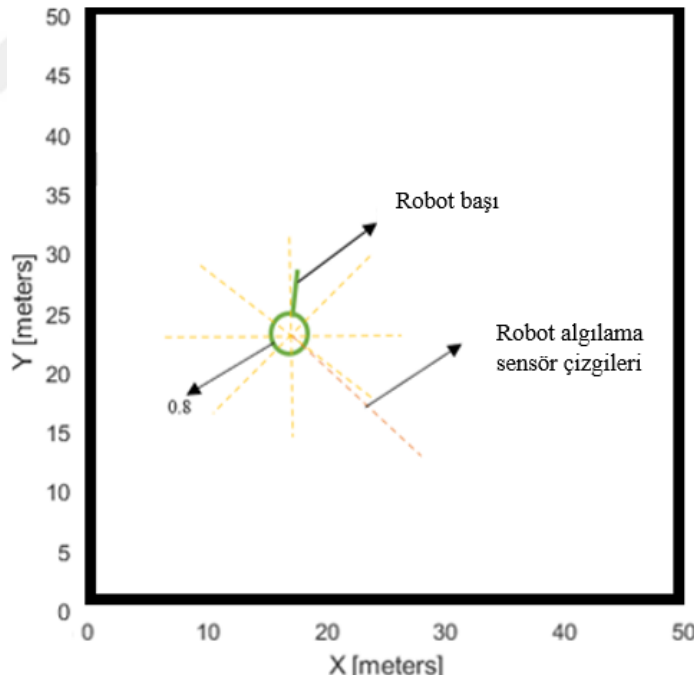
(3)

Simülasyon deneylerinde, robotlar belirlenen arenalara rastgele belirli başlangıç aralığında rastgele olarak dağıtılmaktadır. Rastgele dağıtılan robotlar birbirleriyle üst üste binecek şekilde aynı konumlarda yerleşmeleri için, rastgele dağılım tekrarlanır. Bu sayede sürü robotları aynı noktalarda ve konumlarda olmayacak şekilde rastgele olarak dağıtılmıştır.

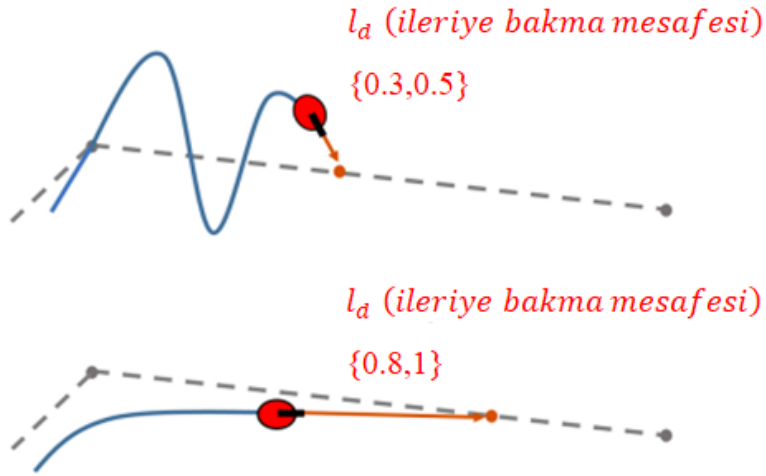
Simülasyon deneylerinde kullanılan sürü robotları holonomik olmayan diferansiyel tahrikli iki tekerlekli modele sahiptir. Robotların yarıçapı $R = 0.8$ birim olarak belirlenmiştir. Sürü robotu dairesel bir yapıya sahiptir. Bu dairesel yapının yarıçapı ile gösterilir. Simülasyon deney ortamı dört tarafı ve içerisi engellerle sınırlandırılmış kare yapıdadır. Kare ortamın duvarları engel olarak

belirtilen
4.1'de

belirtilmiştir. Bu
deney arenaları Şekil
gösterilmektedir.



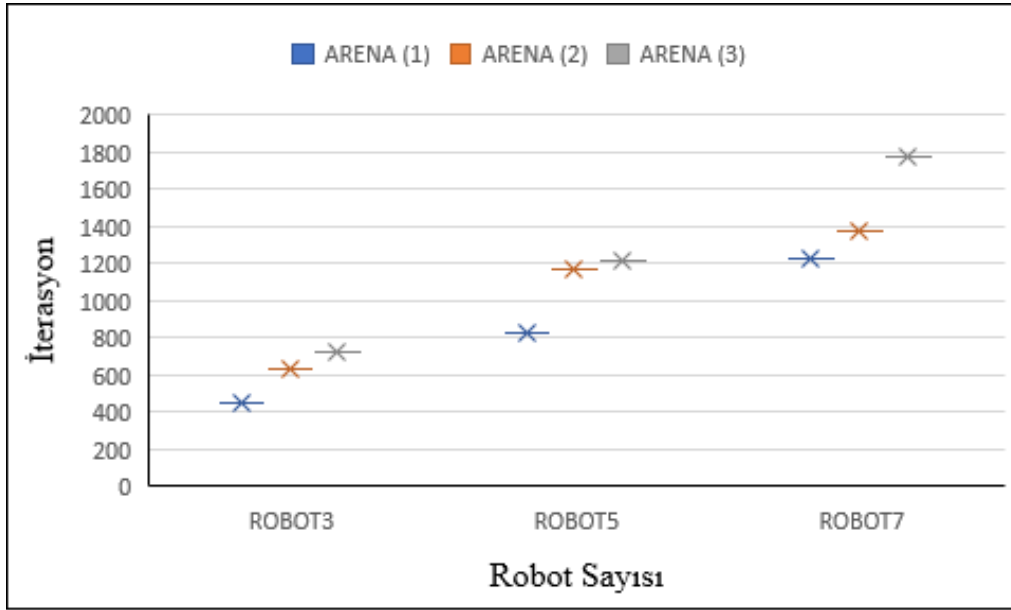
Şekil 4.2: Simülasyon deney platformu ve sürü robotu



Şekil 4.3: İleri bakma mesafesi l_d

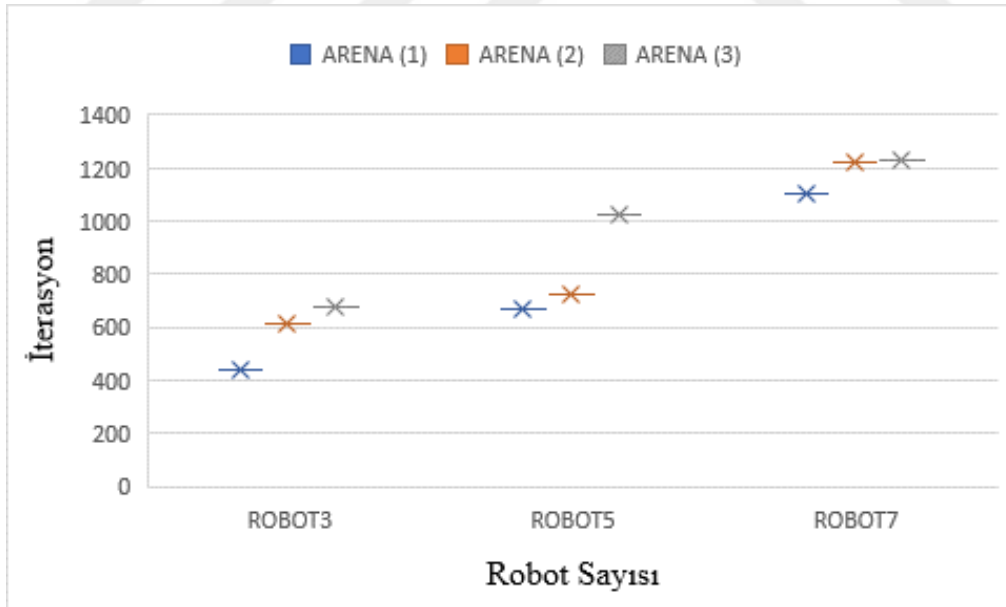
4.1.2.Simülasyon Sonuçları

Bu bölümde önerilen yol planlama ve engellerden kaçınma algoritması farklı arenalarda, farklı robot sayılarında ve farklı l_d değerinde simülasyon deney sonuçları incelenmiştir. Kutu grafiklerinde orta düz çizgi ile belirtilen değer yineleme sayılarının ortalama değeridir. 0.3 birim l_d değeri için arena (1), arena(2) ve arena (3) ile uygulanan sistematik deneylerde 3, 5, 7 sürü robotları için deney sonuçları, 827 iterasyon adımıyla en düşük ortalama değer arena (1)'de 5 sürü robotu ile, en yüksek ortalama değer ise 1782 iterasyon adımı olan arena(3)'de 7 sürü robotu ile gerçekleşmiştir.



Şekil 4.4: 0.3 birim değer VFH algoritması için uygulanan sistematik test sonuçları

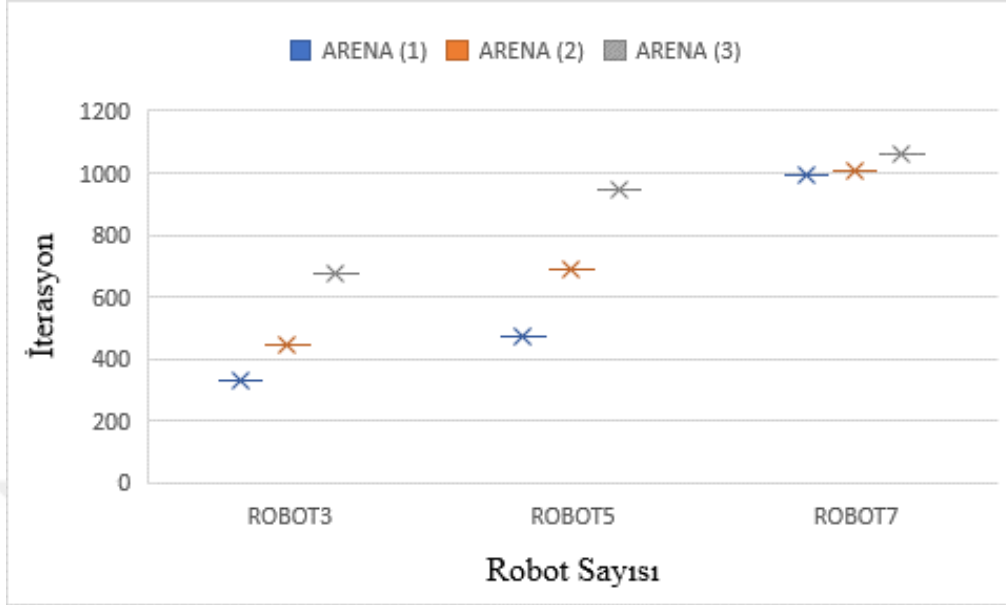
0.5 birim l_d değeri için arena (1), arena (2) ve arena (3) ile uygulanan sistematik deneylerde 3, 5, 7 sürü robotları için deney sonuçları, 665 iterasyon adımında en düşük ortanca değer arena (1)'de 5 sürü robotu ile, en yüksek ortanca değer ise 1025 iterasyon adımı olan arena (3)'de 5 sürü robotu ile gerçekleşmiştir.



Şekil 4.5: 0.5 birim değer VFH algoritması için uygulanan sistematik test sonuçları

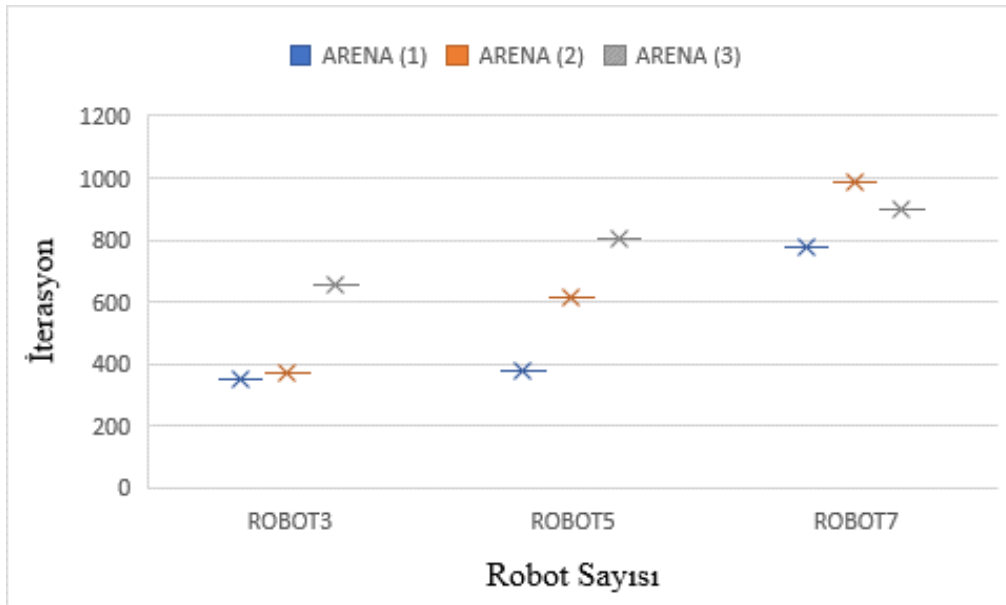
0.8 birim l_d değeri için arena (1), arena (2) ve arena (3) ile uygulanan sistematik deneylerde 3, 5, 7 sürü robotları için deney sonuçları, 473 iterasyon adımında en düşük ortanca değer arena

(1)'de 5 sürü robotu ile, en yüksek ortalanca değer ise 946 iterasyon adımı olan arena (3)' de 5 sürü robotu ile gerçekleşmiştir.



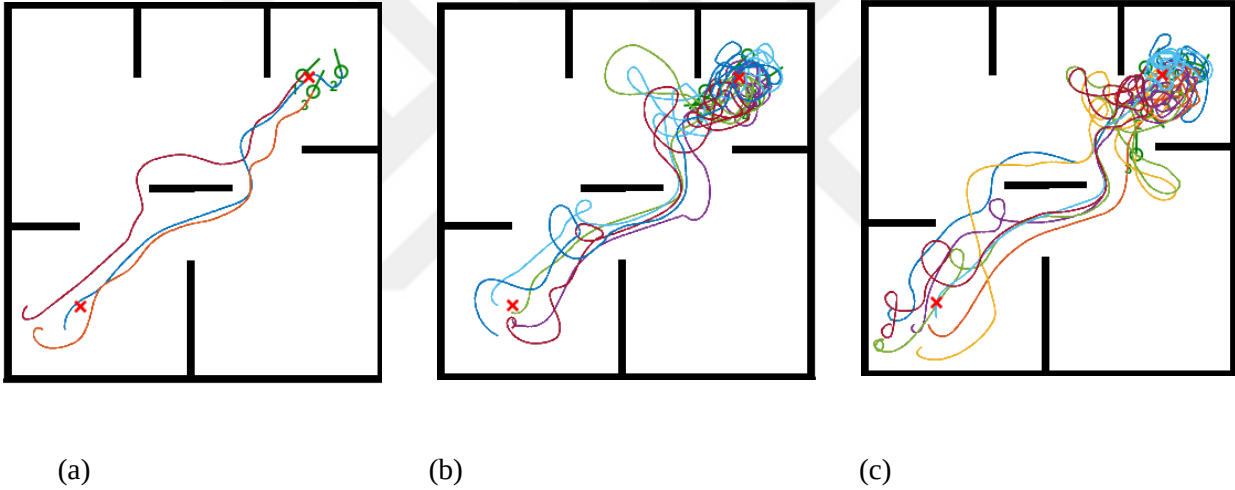
Şekil 4.6: 0.8 birim değer VFH algoritması için uygulanan sistematik test sonuçları

1 birim l_d değeri için arena (1), arena (2) ve arena (3) ile uygulanan sistematik deneylerde 3, 5, 7 sürü robotları için deney sonuçları, 378 iterasyon adımı en düşük ortalanca değer arena (1)'de 5 sürü robotu ile, en yüksek ortalanca değer ise 807 iterasyon adımı olan arena (3)' de 5 sürü robotu ile gerçekleşmiştir.



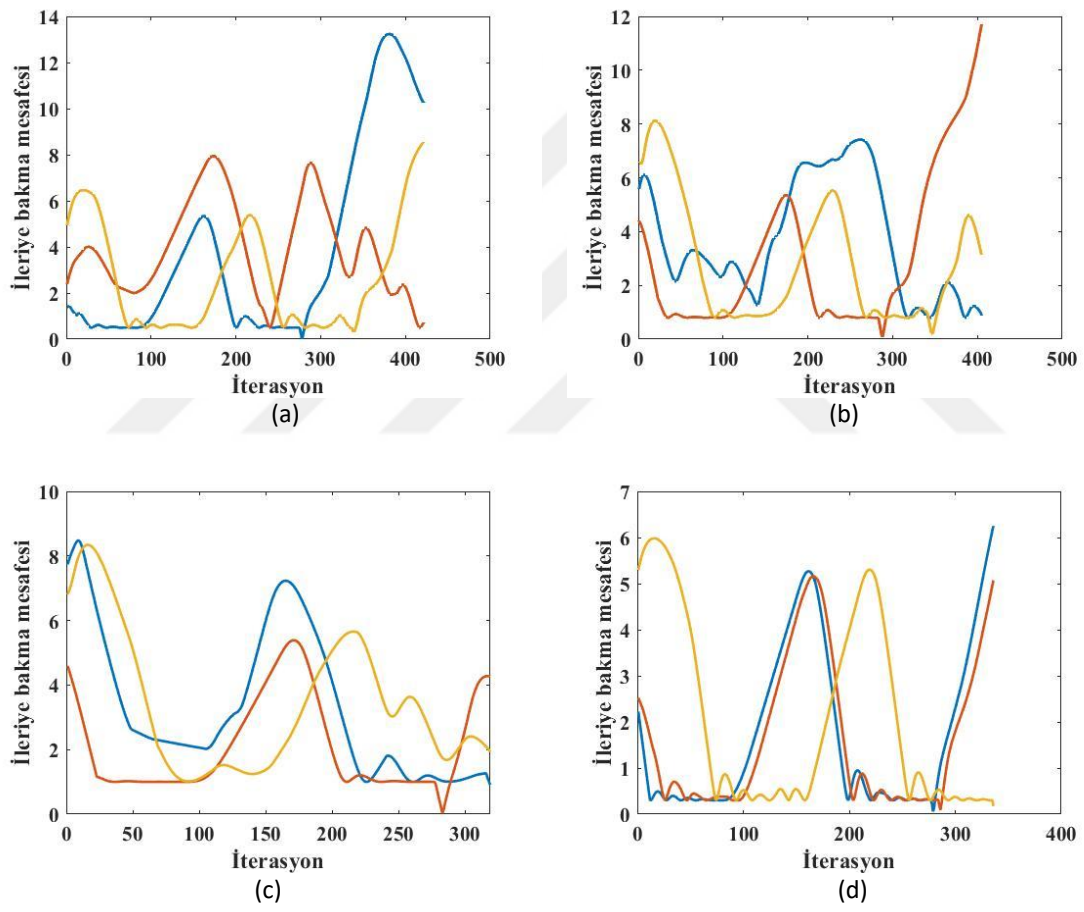
Şekil 4.7: 1 birim değer VFH algoritması için uygulanan sistematik test sonuçları

Bu değerler göz önüne alındığında hedef noktaya ulaşmak için daha çok engel olan arenada, simülasyon tamamlama süresi uzun sürmüştür. Bu sonucun temel nedeni, sürü robotlarının organize bir şekilde yolu takip ederken engelle karşılaştıklarında hem engellere çarpmadan hem de birbirlerine çarpmadan yolu belirlemeleridir. Uygulanan sistematik deneylerde ileriye bakma mesafesinin (l_d) değişmesine bağlı olarak değerlendirildiğinde mesafe arttıkça simülasyonun tamamlanma süresi azalmaktadır. İleriye bakma mesafesi arttıkça sürü robotları istenilen hedef noktaya varması için daha az salınım yapmaktadır. Robot sayısındaki artış ise simülasyonun tamamlanma süresinde artışa neden olmaktadır.

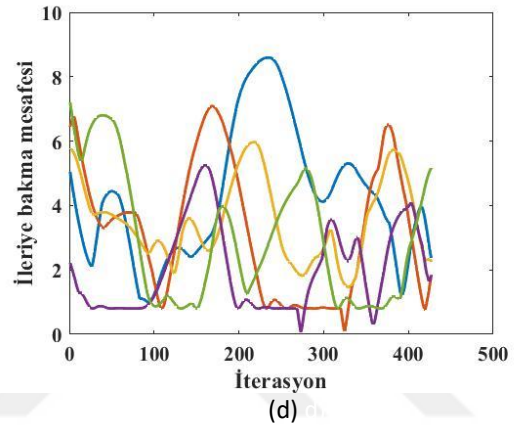
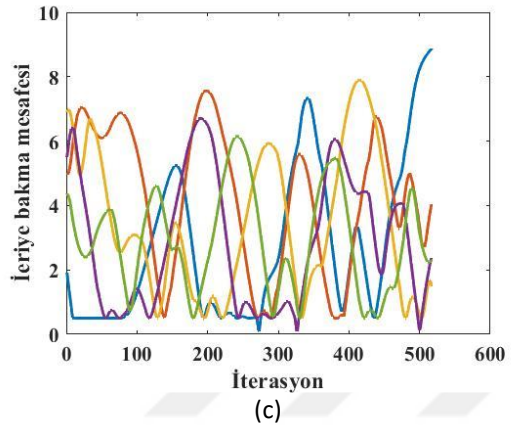
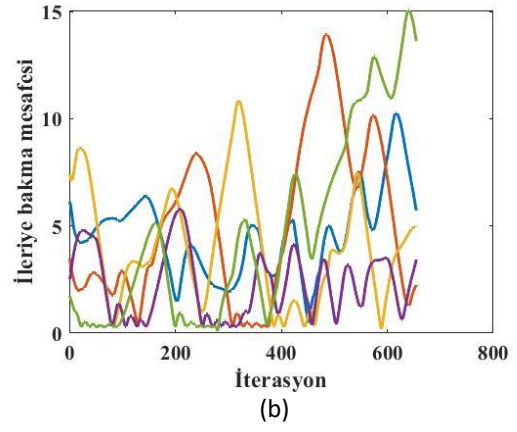
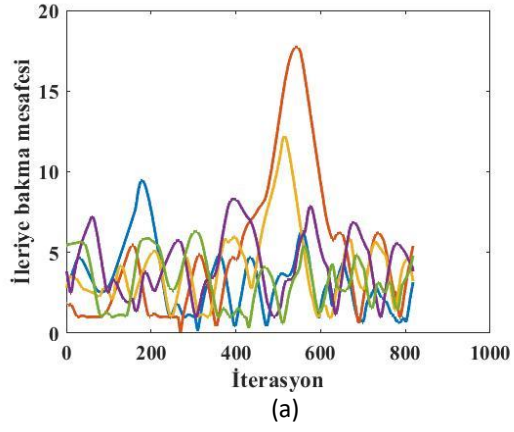


Şekil 4.8: Sürü robotları için VFH- PurePursuit tabanlı iş birliğine dayalı engelden kaçınma ve yol planlama algoritmasının Arena (1) için sonuçları

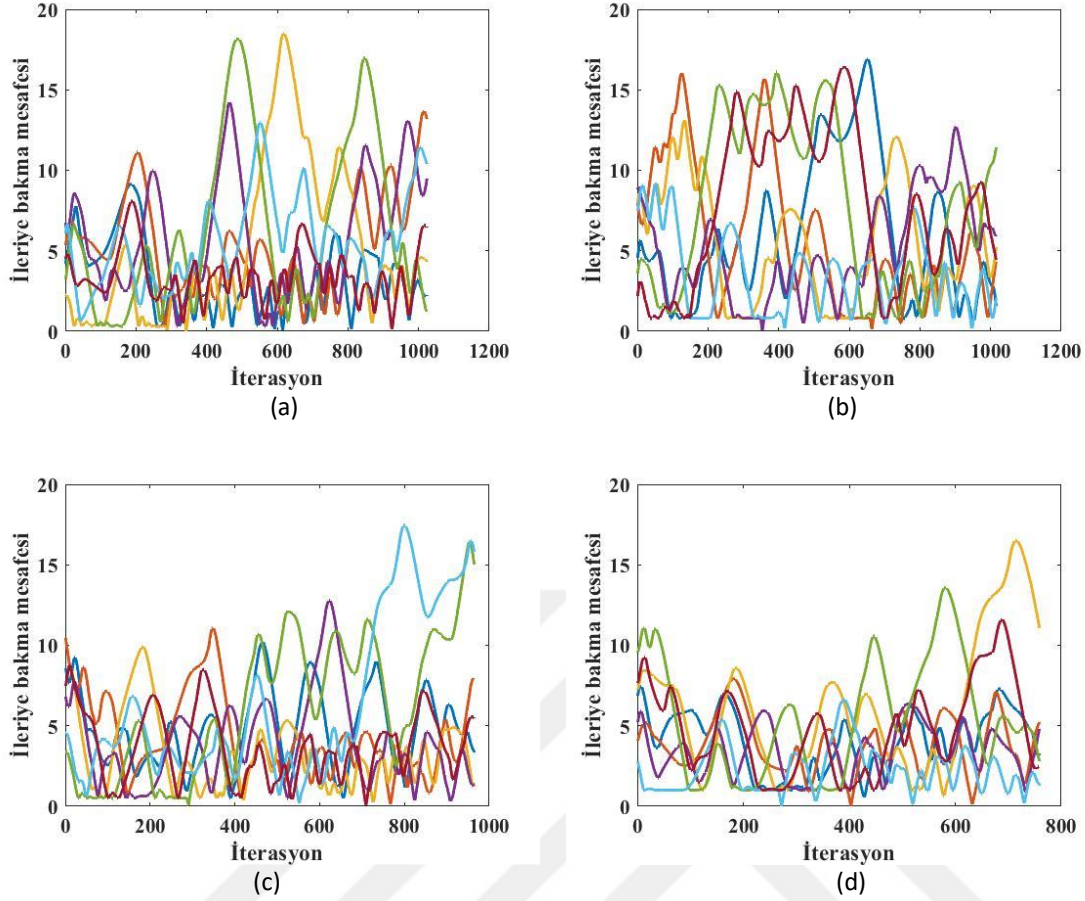
Şekil 4.8’de Arena (1) olarak isimlendirilen 50x50 boyutları için sistematik simülasyonlarda sırasıyla seçilmiş 3,5 ve 7 sürü robotları ile uygulanan örnek deneyler gösterilmiştir. Bu simülasyonlarda yol planlama ve engellere çarpmadan başlangıç konumundan hedef konuma ulaştığı sürü robotlarının deney sonuçları gösterilmiştir.



Şekil 4.9: (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan simülasyon sonuçları



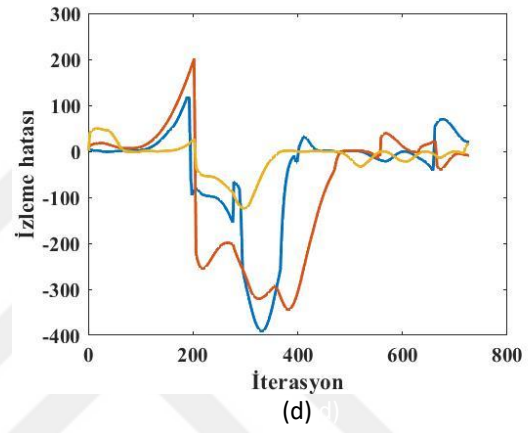
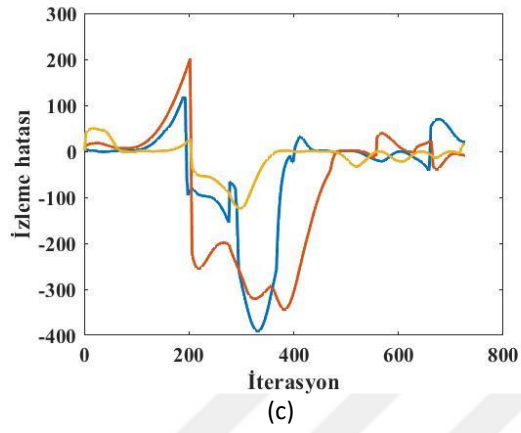
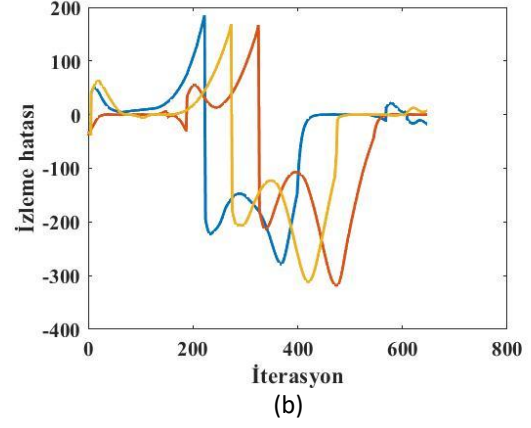
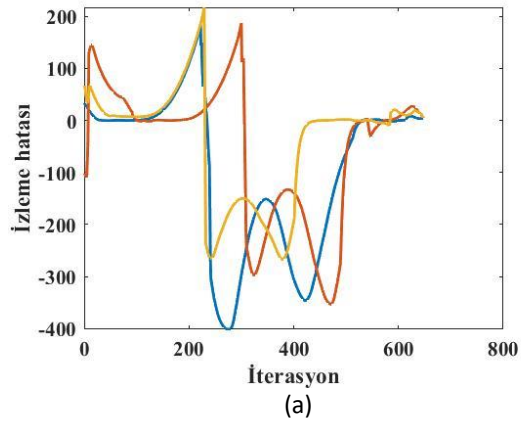
Şekil 4.10: (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan simülasyon sonuçları



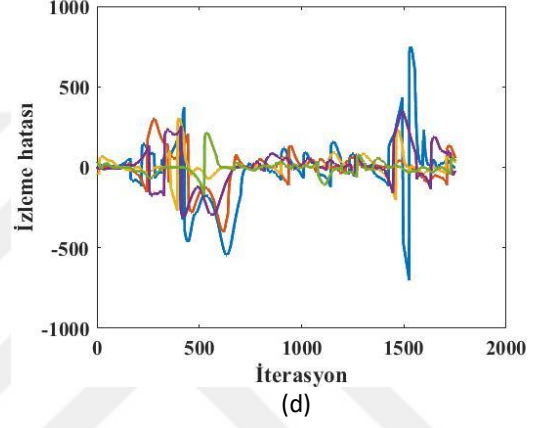
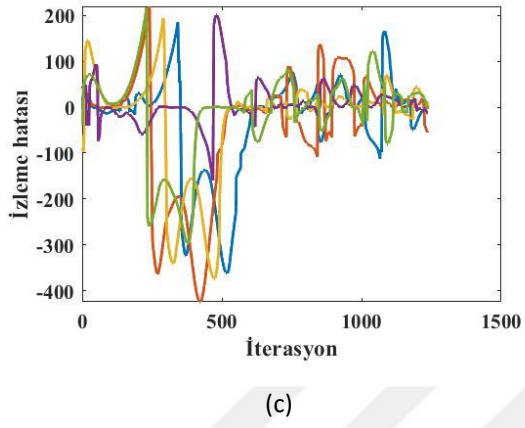
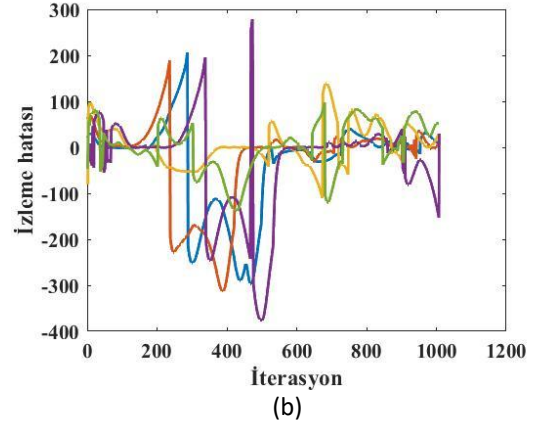
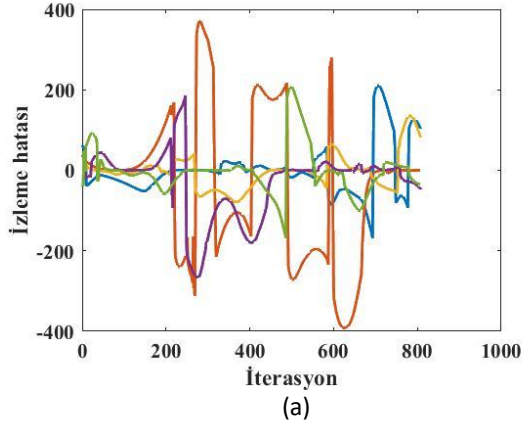
Şekil 4.11: (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan simülasyon sonuçları

Şekil 4.9, 4.10 ve 4.11 grafiklerinde Şekil 4.8 ‘deki arena (1)’de yapılan deneyden elde edilen değerler gösterilmektedir. Sürü robotlarının yol takibi sırasında bir engel veya keskin dönüş ile karşılaşması durumunda belirlenen yola olan mesafe artar ve bu durum mesafelere bakılarak görülebilir.Yapılan deneyler için 7 robot 0.3 değerli ileriye bakma mesafesi için en uzun simülasyon süresi (yineleme sayısı),en kısa simülasyon süresi ise 3 robot ile 1 değerli ileriye bakma mesafesi için görülmektedir. Bu sonuçların sebebi artan robot sayısı ve uzun mesafesi ileriye bakma mesafesidir.

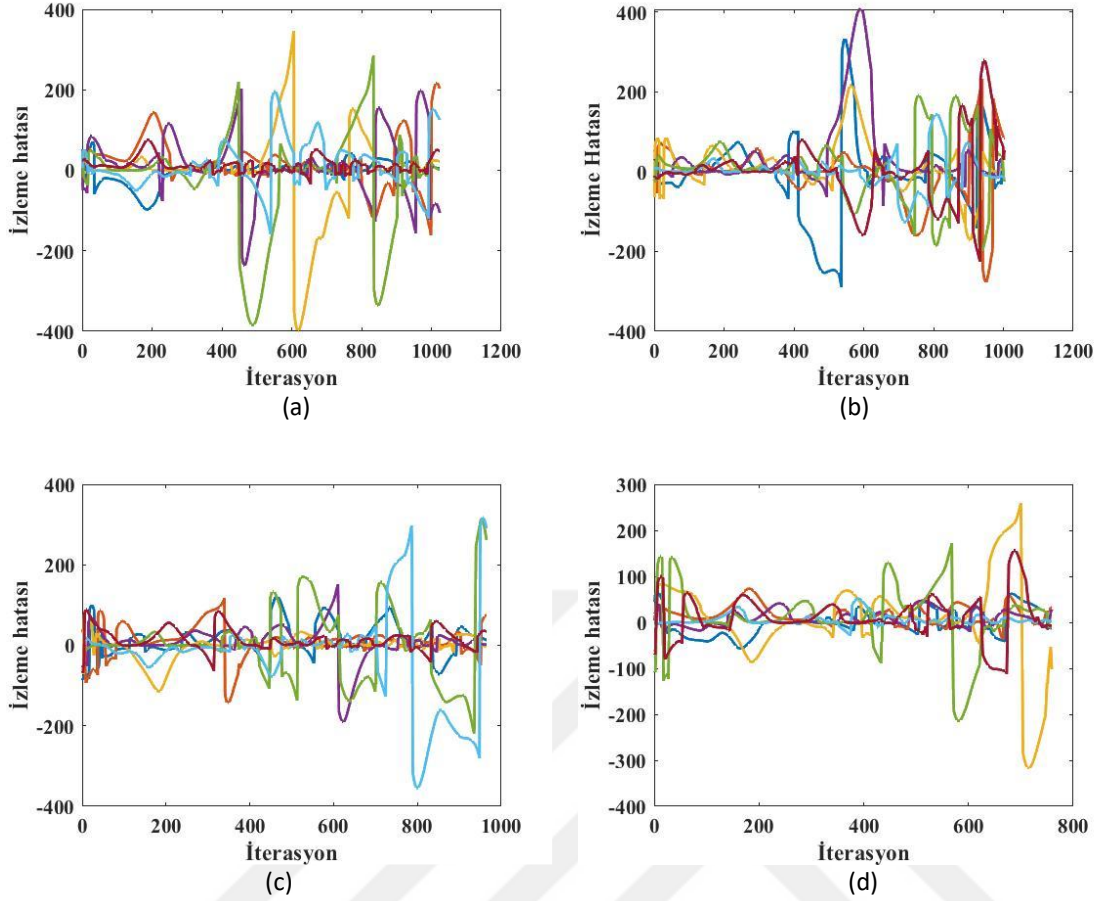
Şekil 4.8’de gösterilen arenalarda yapılan deneylerden elde edilen izleme oluşmaktadır. Denklem (3.6) ifade edilen izleme hatası, sürü robotun izleme yoluna dik mesafesidir.



Şekil 4.12 : (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan izleme hatası simülasyon sonuçları

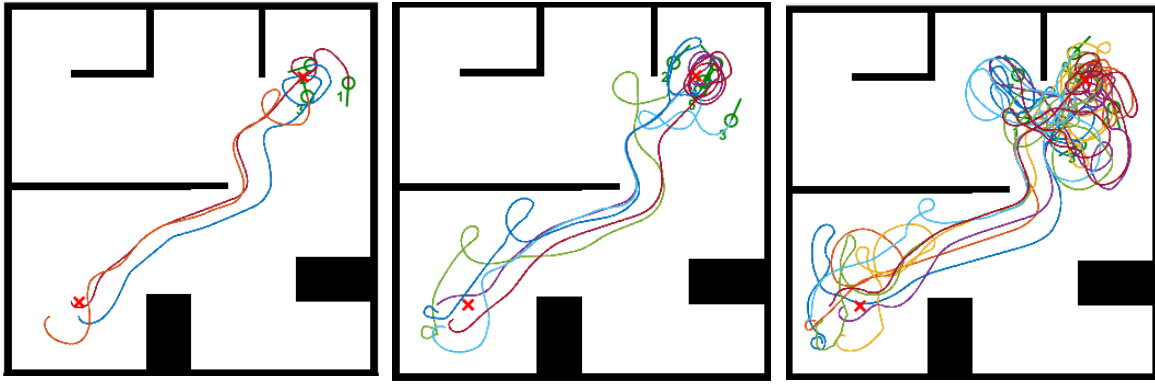


Şekil 4.13 : (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan izleme hatası simülasyon sonuçları



Şekil 4.14: (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan izleme hatası simülasyon sonuçları

Şekil 4.12, 4.13 ve 4.14 gösterildiği gibi, her sürü robotu belirtilen hedef konuma ulaşmaya çalışırken engel veya keskin dönüş anında bir izleme hatası gözlemlenir. Sürü robotlarının bir engelle veya keskin dönüş ile karşılaştıklarında hedef konuma ulaşmak için yol mesafesi artacaktır. Engellerden kaçan her sürü robotu hedef konuma daha da yaklaşmak için organize bir şekilde hareket etmektedir. Önerilen yaklaşımda, sürü robotlarının engellerle karşılaşması durumunda yolu takip etmekten ziyade engellerden kaçınmaya öncelik verilmektedir. Geliştirilen algorithmada, sürü robotlarının engellerle karşılaşması ile hedef konuma ulaşmaktan yerine engellerden kaçınma tercih edilmektedir.



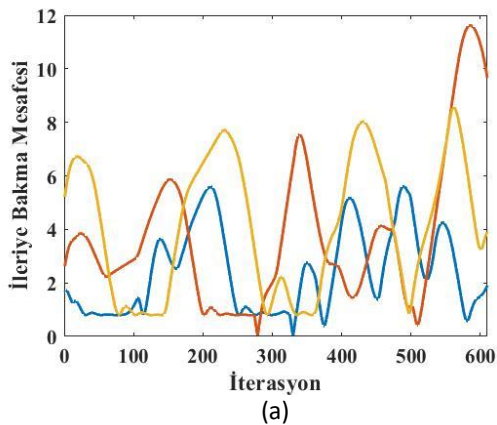
(a)

(b)

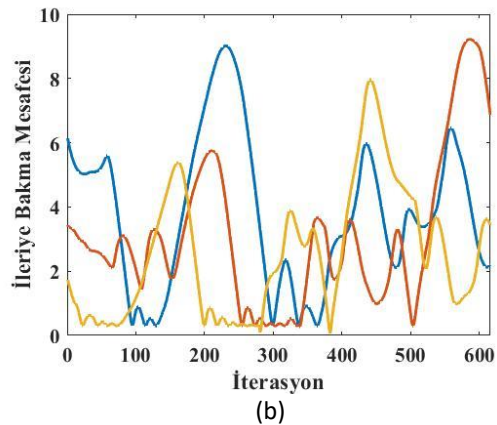
(c)

Şekil 4.15: Sürü robotları için VFH- PurePursuittabanlı iş birliğine dayalı engelden kaçınma ve yol planlama algoritmasının Arena (2) için sonuçları

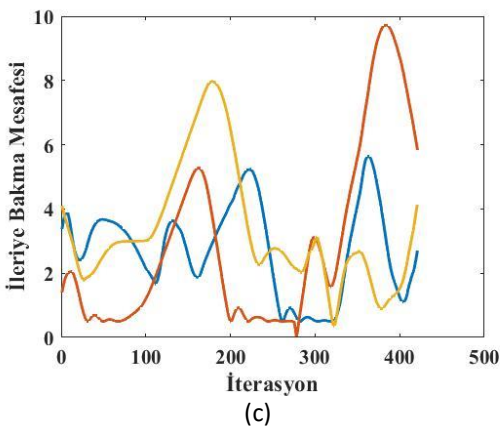
Şekil 4.15'te arena (2) olarak isimlendirilen 50x50 boyutları için sistematik simülasyonlarda sırasıyla seçilmiş 3,5 ve 7 sürü robotları ile uygulanan örnek deneyler gösterilmiştir. Bu simülasyonlarda yol planlama ve engellere çarpmadan başlangıç konumundan hedef konuma ulaştığı sürü robotlarının deney sonuçları gösterilmiştir.



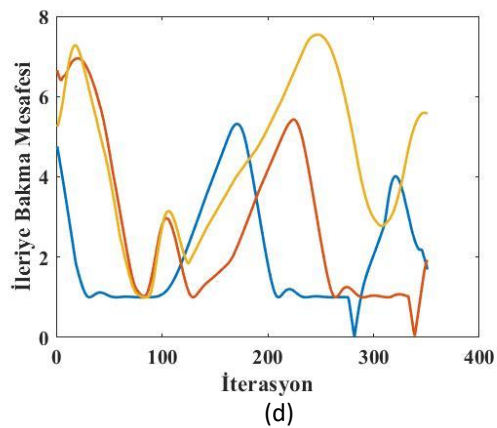
(a)



(b)



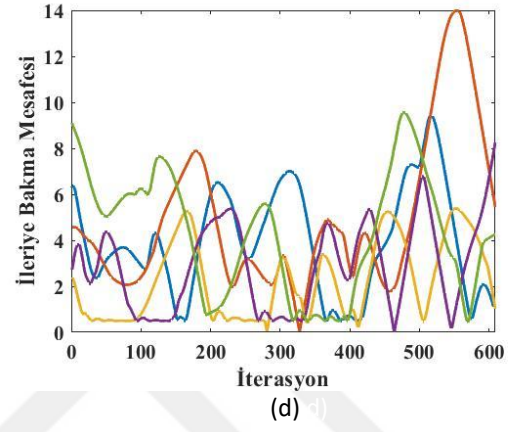
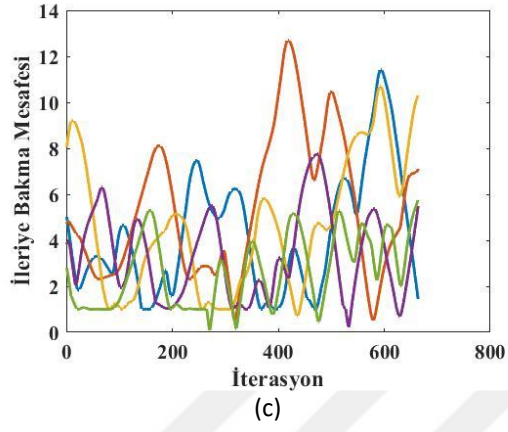
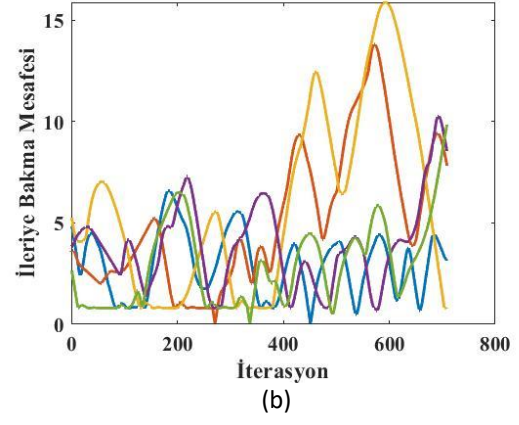
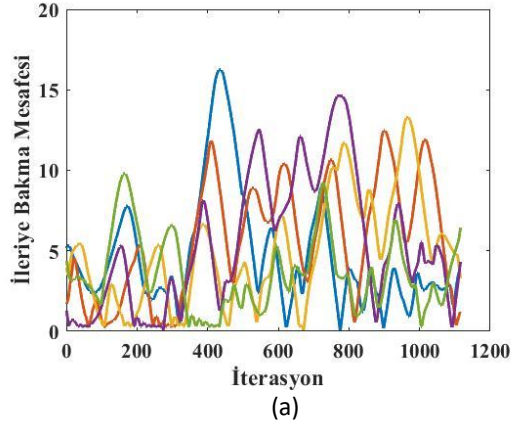
(c)



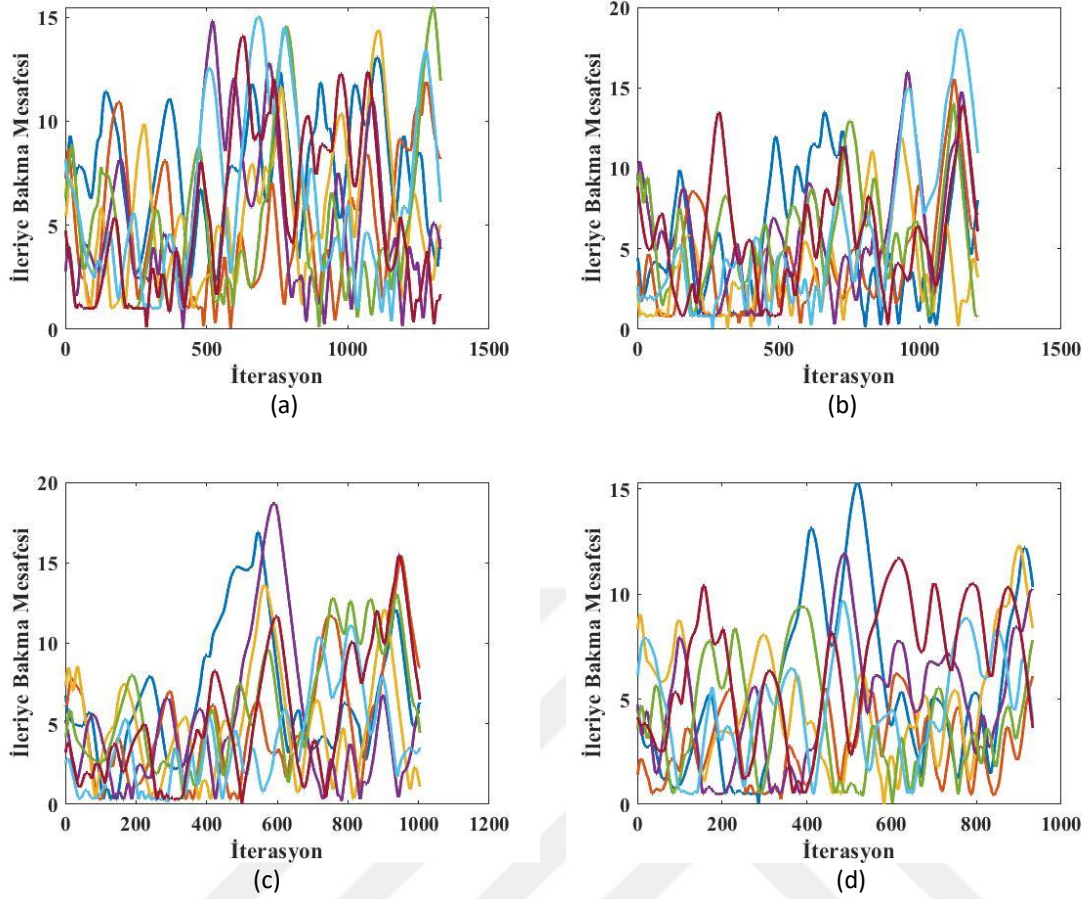
(d)

Şekil

4.16: (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan simülasyon sonuçları

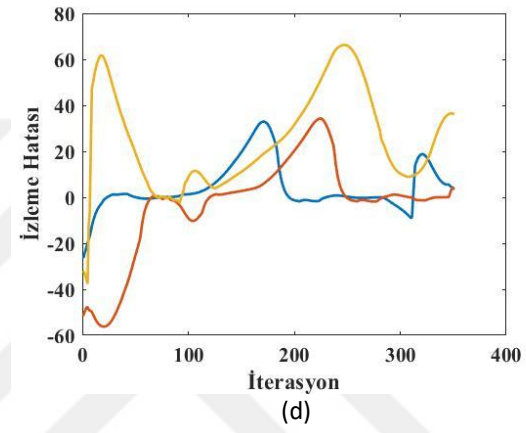
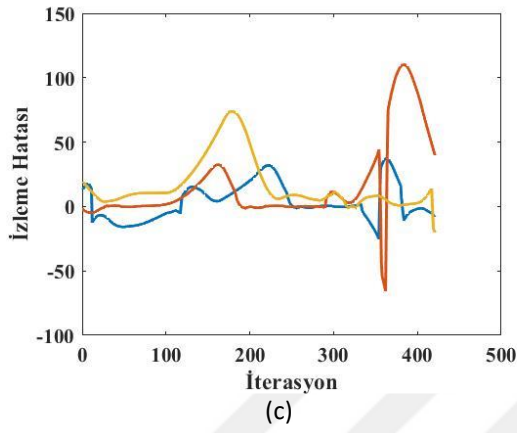
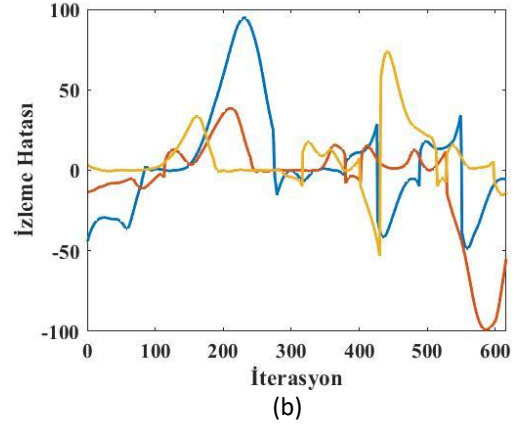
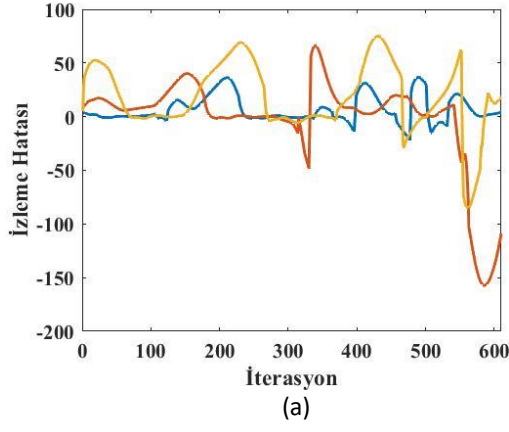


Şekil 4.17: (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan simülasyon sonuçları

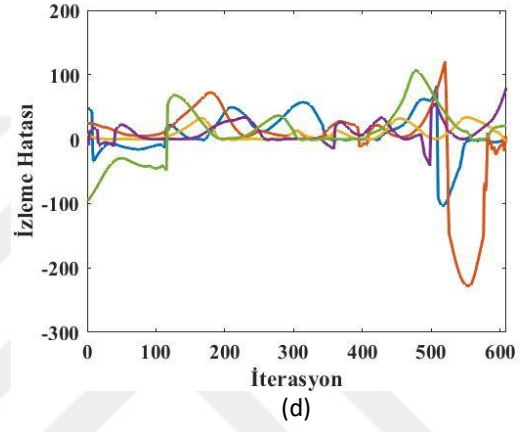
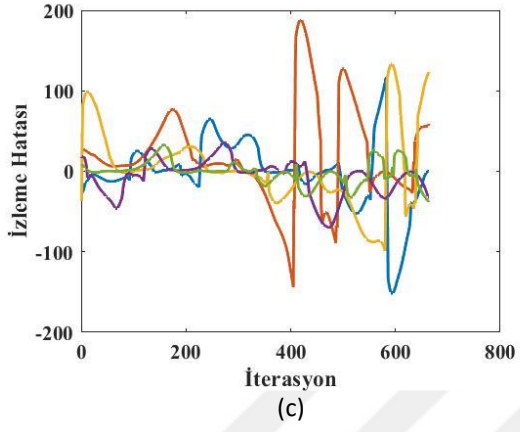
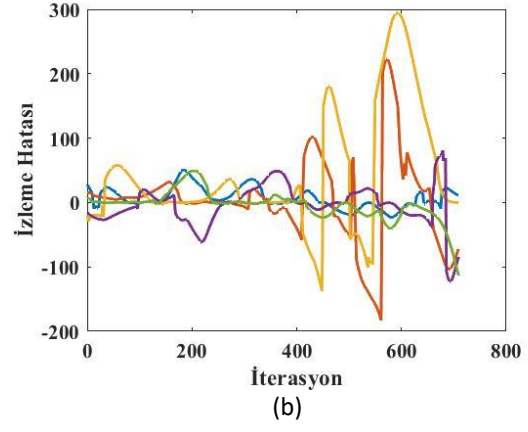
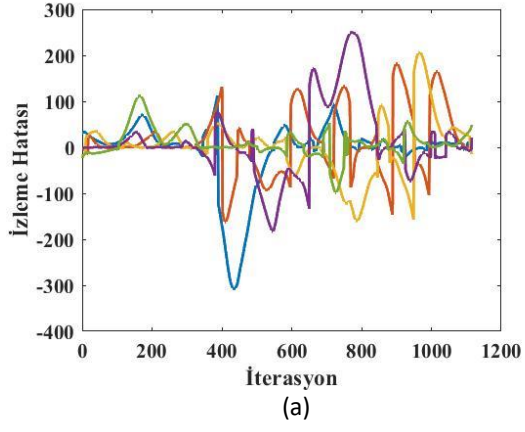


Şekil 4.18: (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan simülasyon sonuçları

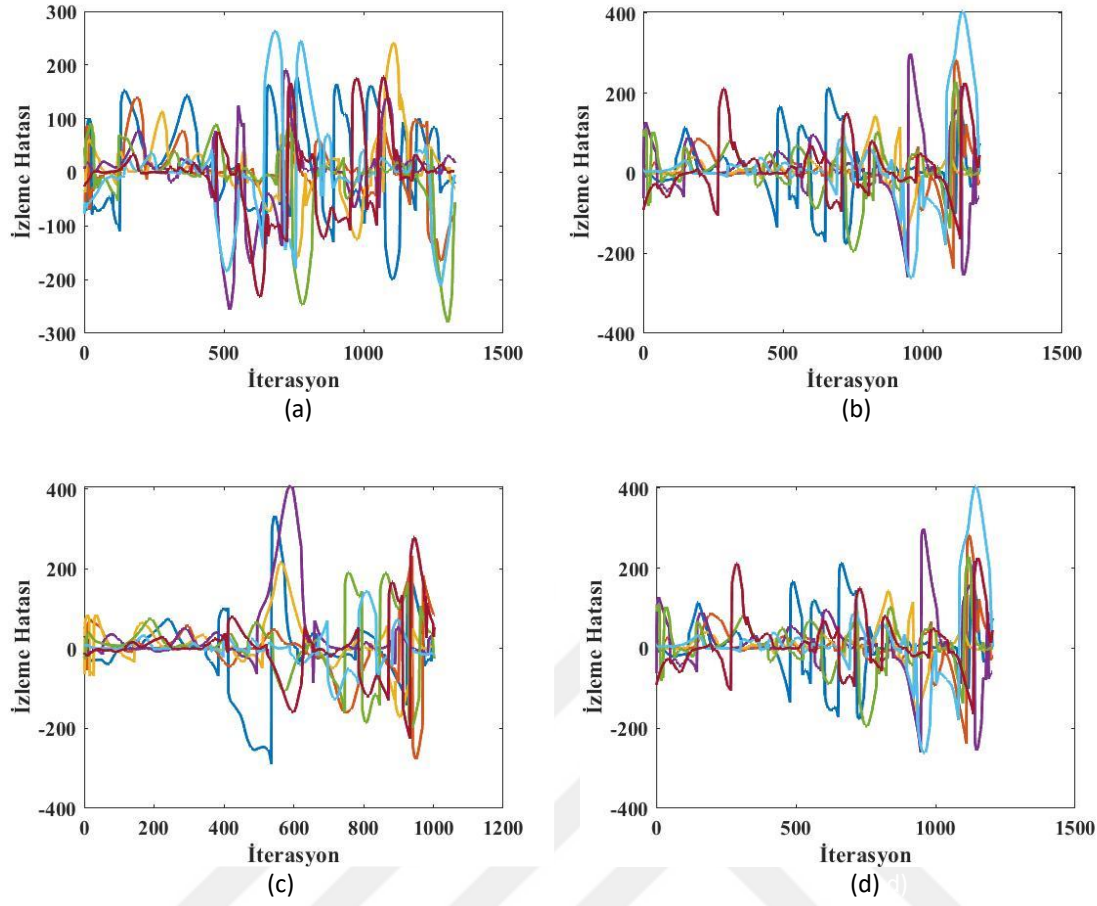
Şekil 4.16, 4.17 ve 4.18 grafiklerinde Şekil 4.15 ‘deki arena (2)’de yapılan deneyden elde edilen değerler gösterilmektedir. Sürü robotlarının yol takibi sırasında bir engel veya keskin dönüş ile karşılaşması durumunda belirlenen yola olan mesafe artar ve bu durum mesafelere bakılarak görülebilir. Şekil 4.15’teki arena (2)’de yapılan deneylerde oluşan izleme hataları incelenecektir. En uzun simülasyon tamamlanma süresi 7 robot ile 0,3 değerli ileriye bakma mesafesi yapılan deneyde, en kısa simülasyon süresi ise 3 robot ile 1 değerli ileriye bakma mesafesi yapılan deneyde görülmektedir. Bu deneyde elde edilen sonuçlar şu ifade edilmektedir. Robot sayısı arttıkça simülasyon tamamlama süresi doğru orantılıdır yani simülasyon süresi de artmaktadır. İleriye bakma mesafesi artış gösterdiğinde, yol boyunca sürü robotlarının salınımını azaltması amaçlanmaktadır. Bu sayede robotlarının yol takip etme mesafesi ve planlanan yolu takip etme süresi azalacaktır.



Şekil 4.19: (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan izleme hatası simülasyon sonuçları



Şekil 4.20: (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan izleme hatası simülasyon sonuçları



Şekil 4.21:(a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan izleme hatası

Şekil 4.19, 4.20 ve 4.21arena(2) için izleme hataları gösterilmektedir. Arena(1) için yapılan deney sonuçları ile benzerlik göstermektedir.



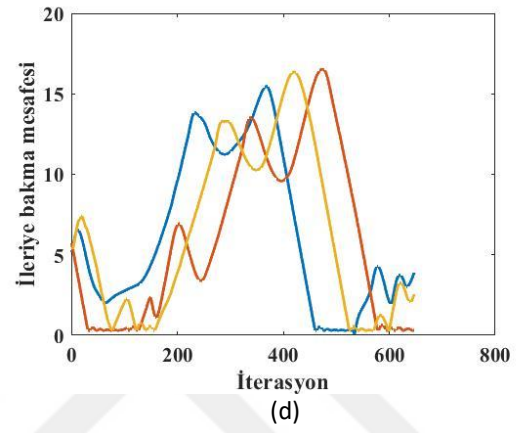
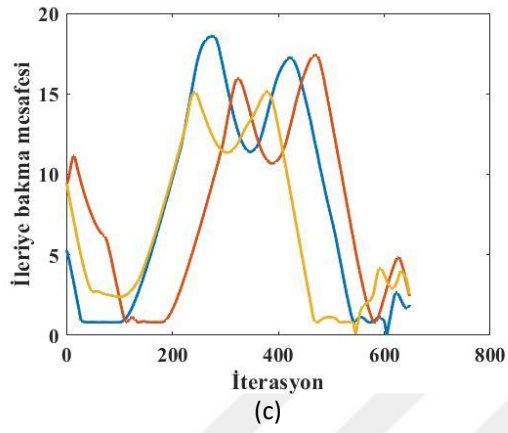
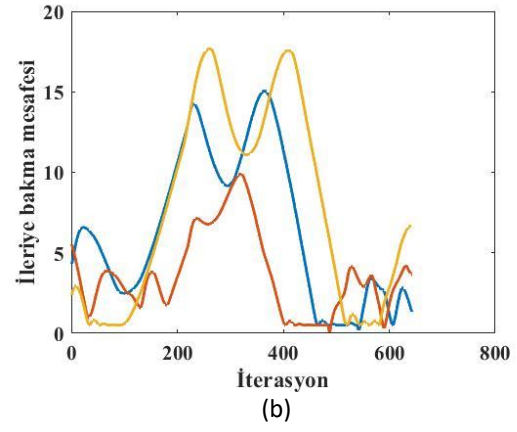
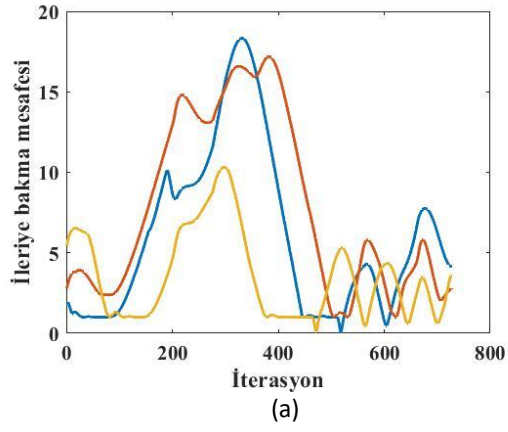
(a)

(b)

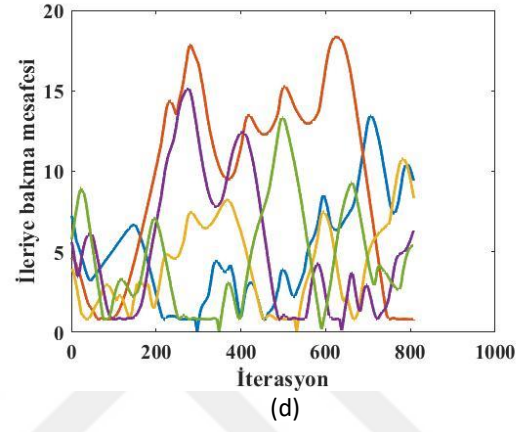
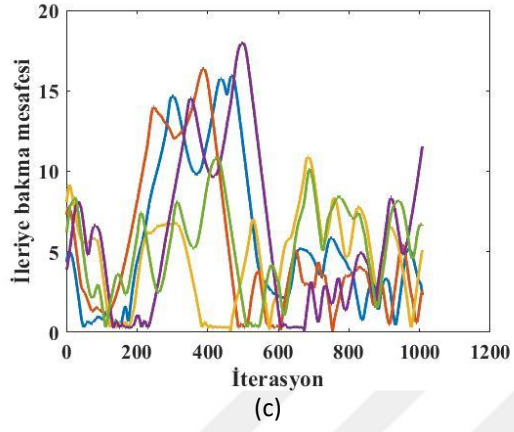
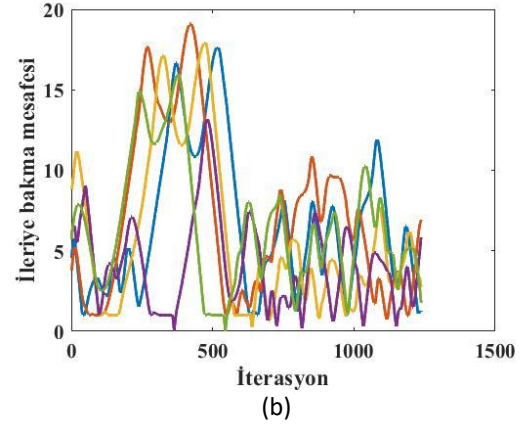
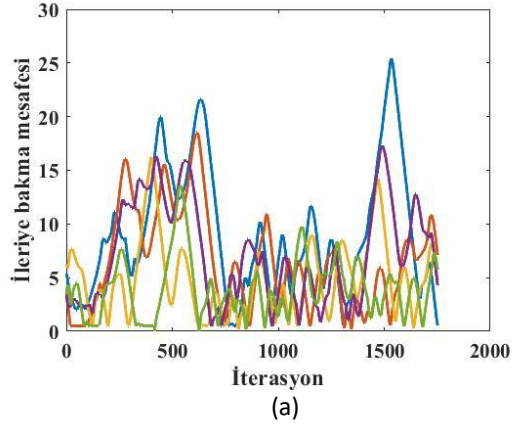
(c)

Şekil 4.22:Sürü robotları için VFH- PurePursuit tabanlı birliğine dayalı engelden kaçınma ve yol planlama algoritmasının arena (3) için sonuçları

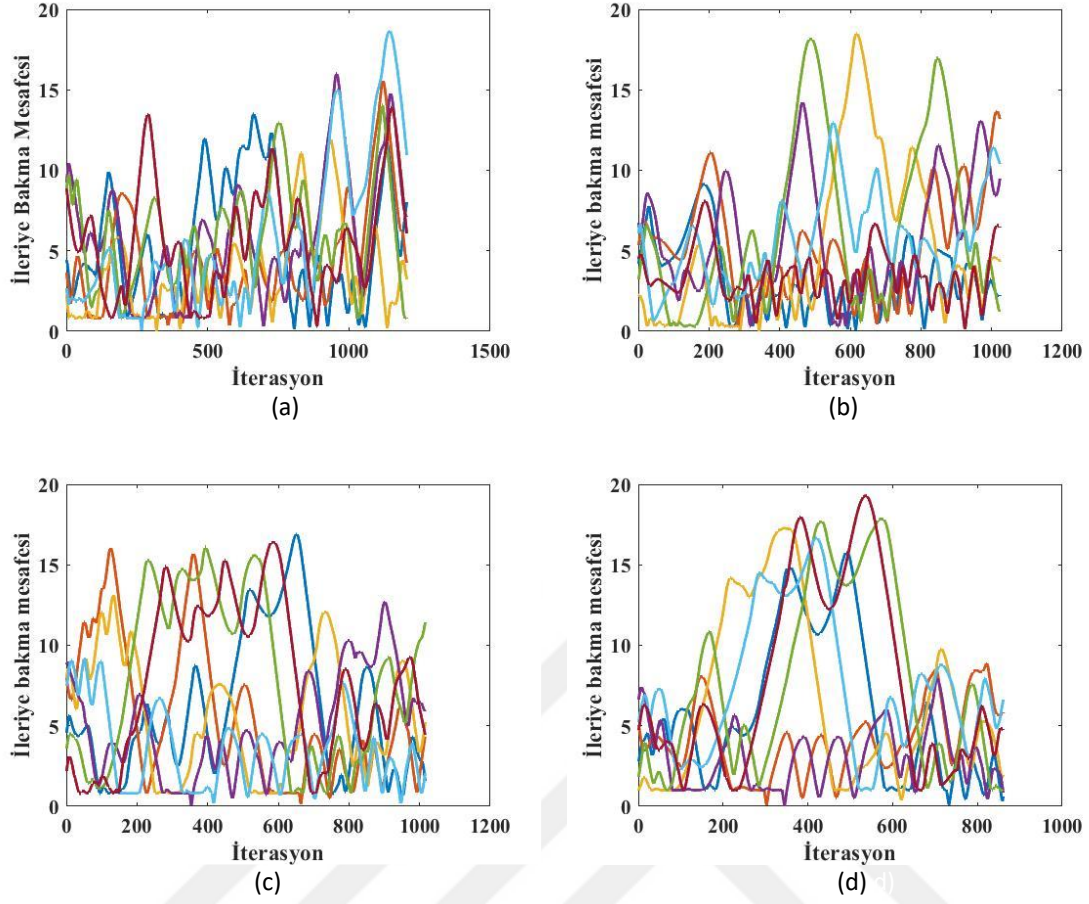
Şekil 4.22’de Arena (3) olarak isimlendirilen 50x50 boyutları için sistematik simülasyonlarda sırasıyla seçilmiş 3,5 ve 7 sürü robotları ile uygulanan örnek deneyler gösterilmiştir. Bu simülasyonlarda yol planlama ve engellere çarpmadan başlangıç konumundan hedef konuma ulaştığı sürü robotlarının deney sonuçları gösterilmiştir.



Şekil 4.23:(a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan simülasyon sonuçları

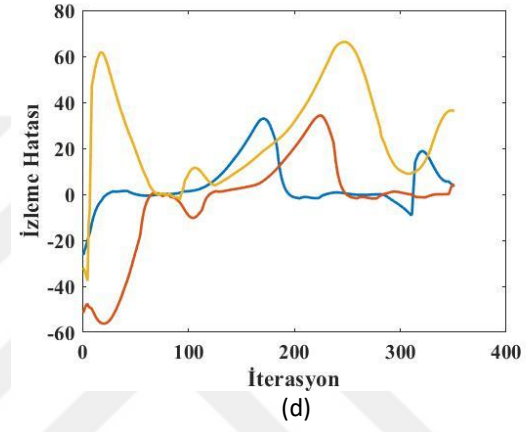
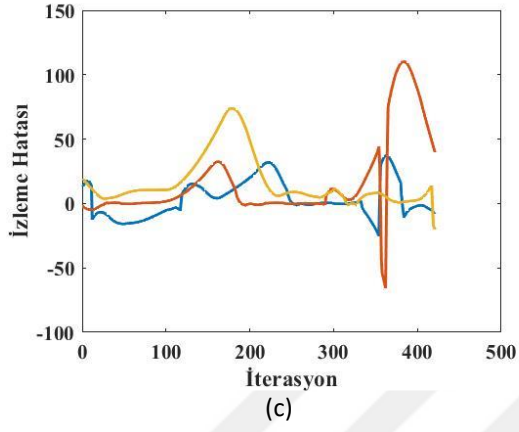
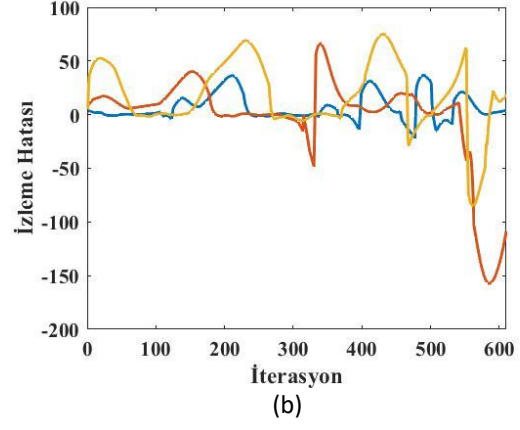
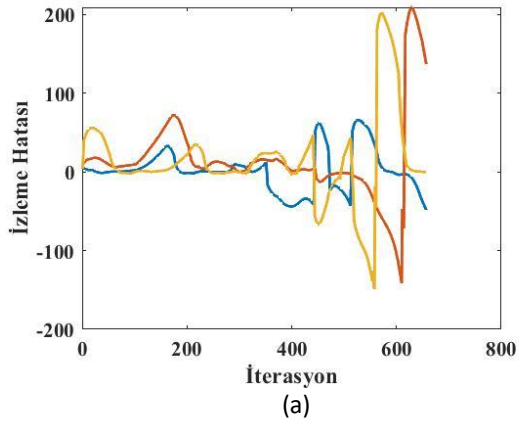


Şekil 4.24:(a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan simülasyon sonuçları

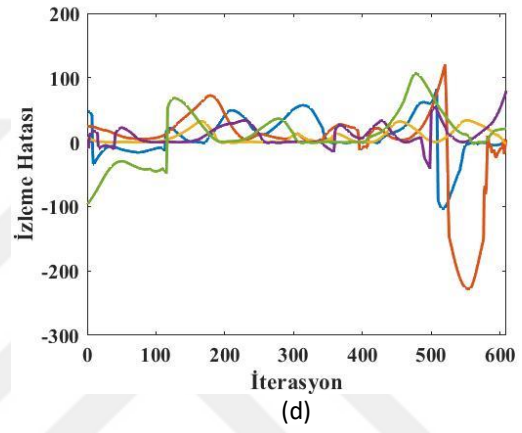
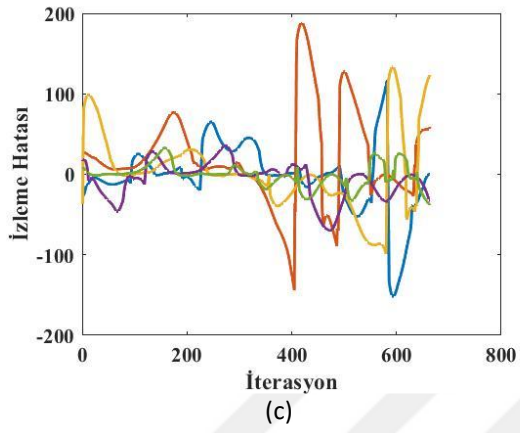
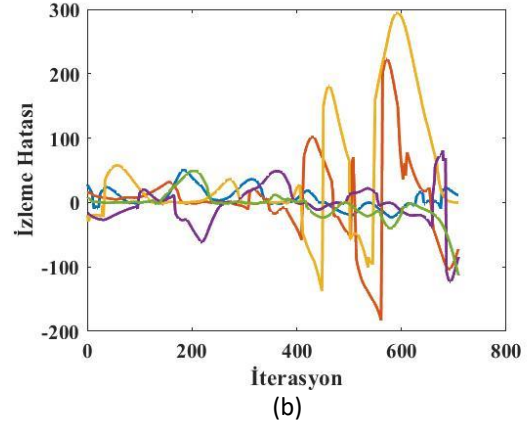
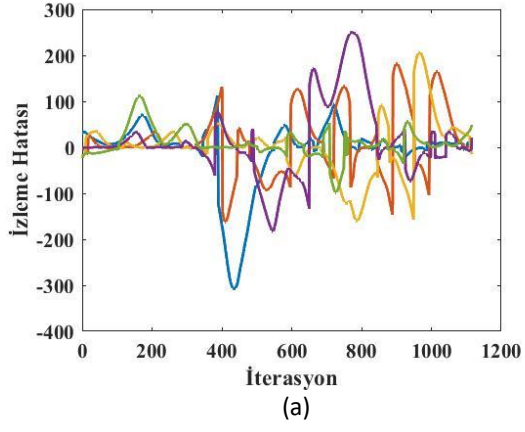


Şekil 4.25: (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan simülasyon sonuçları

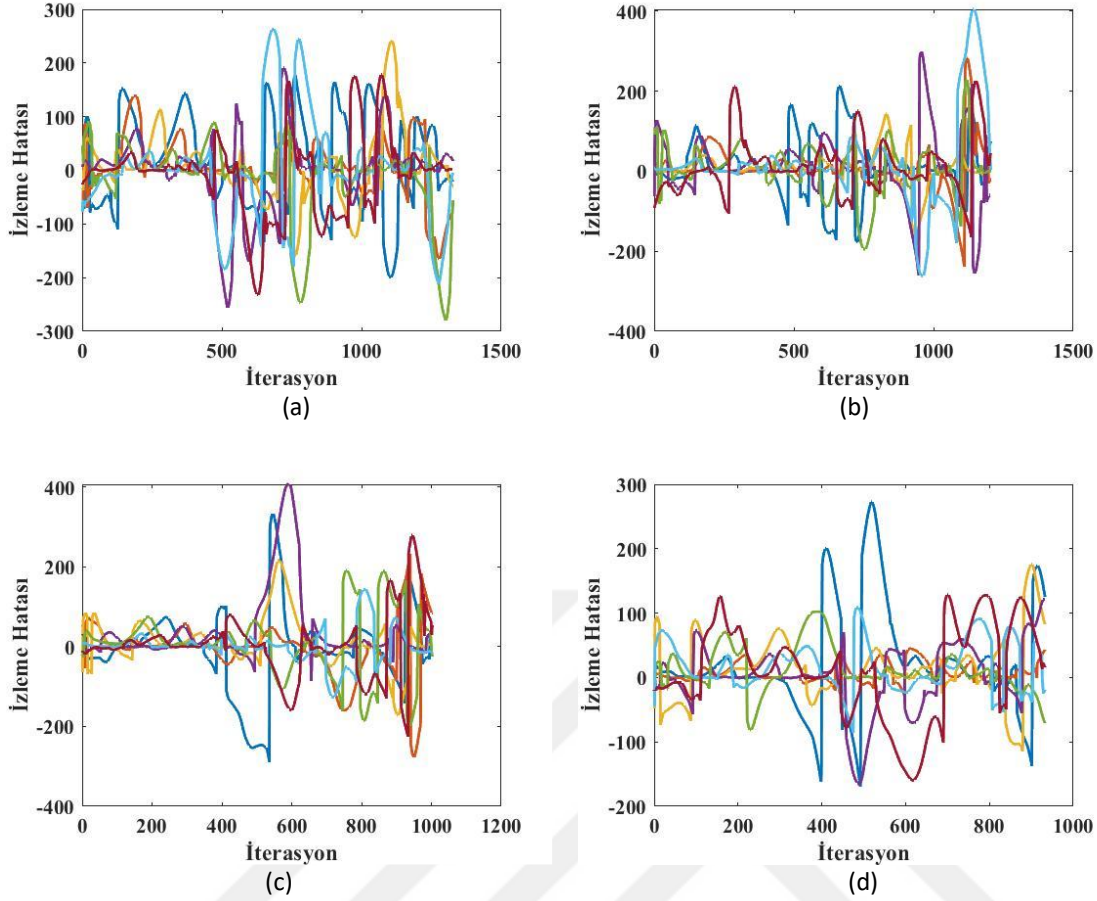
Şekil 4.23, 4.24 ve 4.25 grafiklerinde Şekil 4.22'deki arena (3)'de yapılan deneyden elde edilen değerler gösterilmektedir. Sürü robotlarının yol takibi sırasında bir engel veya keskin dönüş ile karşılaşması durumunda belirlenen yola olan mesafe artar ve bu durum mesafelere bakılarak görülebilir. Şekil4.22'deki arena (3)'de yapılan deneylerde oluşan izleme hataları incelenecektir.



Şekil 4.26: (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan izleme hatası



Şekil 4.27: (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan izleme hatası



Şekil 4.28: (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan izleme hatası

Şekil 4.26, 4.27 ve 4.28 arena (3) için sürü robotlarının izleme hatalarına yer verilmektedir. Burada arena(1) ve arena (2)'deki sonuçlara bakıldığında her 3 farklı arenada aynı sonuçlar elde edilmiştir. İzleme hatalarında grafiklerde görüleceği üzere sıfıra yakınsama görülmemektedir. Bunun sebebi oluşturulan algorithmada sürü robotları sadece başlangıç ve hedef konumları bilinmektedir. Başlangıç ve hedef konum arasında izleyeceği yol bilinmemektedir. Yol üzerinde bir engel ile karşılaşması durumunda hedefe ulaşmaktan ziyade engelden kaçınması önem arz etmektedir.

4.2.Sürü Robotları için RRT- PurePursuitTabanlı İş Birliğine Dayalı Engelden Kaçınma ve Yol Planlama

Geliştirilen sürü robotları için RRT- PurePursuit tabanlı iş birliğine dayalı engelden kaçınma ve yol planlama algoritması için Çizelge 3’teki gibi simülasyon ortamında sistematik deneyler uygulanmıştır. Sistematik deneyler arena boyutu ve farklı arenalarda, robot algılama yarıçapı, l_d değerine, robot sayısına bağlı olarak belirlenmiştir. Uygulanan deneylerindeki sürü robotlarının, değişen arenalar (engeller) ve değişen l_d değerleri ile ölçeklenebilir bir şekilde farklı robot sayılarında yol planlama ve engelden kaçınma davranışları uygulayarak başlangıç konumunda hedefe konuma en kısa yol oluşturarak simülasyonunun tamamlanması amaçlanmıştır. RRT algoritması ile farklı arenalarda başlangıç konum ile hedef konum arasında yol planlanmaktadır. Planlanan yolu takip edebilmek için PurePursuit algoritması kullanılmıştır. PurePursuit l_d değerinin değişikliği ile sürü robotlarının ölçeklenebilirliği ve esnekliği başlangıç konum ile hedef konum arasında engelsiz yol ve oluşturulan yolun izlenebilirliğini açısından gözlemlenecektir.

4.2.1. Simülasyon Kurulumu

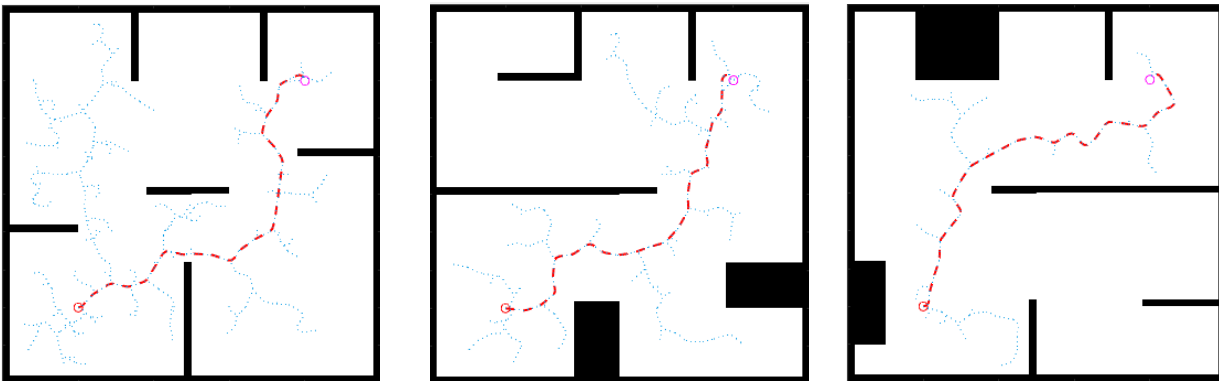
Çizelge 3’te RRT- PurePursuit tabanlı iş birliğine dayalı engelden kaçınma ve yol planlama simülasyon kurulumu için,

Arena boyutu	{50x50}
l_d Değeri	{0.3, 0.5, 0.8, 1}
Robot sayısı	{3, 5, 7}
Bağlantı Mesafesi	2,5
Minimum Dönüş Yarıçapı	0,75

Çizelge 3: RRT simülasyon özellikleri

Çizelge 3’te belirlenen robot sayıları {3, 5, 7}, PurePursuit yöntemi için l_d değeri {0.3, 0.5, 0.8, 1} ve arena boyutları aynıdır fakat 50x50 boyutlarında farklı arenalarda bağlı bağlantı

mesafesi 2,5



birim ve minimum dönüş yarıçapı olarak 0,75 birim için engelsiz yollar planlanmıştır. Planlanan bu yollar için 36 sistematik deney uygulanmıştır. Yol planlama ve engelden kaçınma probleminin performansını ölçmek için simülasyon süresi kullanılmıştır. Simülasyon süresi, deneylerin başlangıç konumundan hedef konuma ulaştığı ana kadar geçen simülasyon adım sayısıdır. Sistematik deneyler neticesinde elde edilen sonuçlarda, sürü robotlarının ölçeklenebilirliği ve esnekliği başlangıç ve hedef konum arasında engelsiz yol oluşturma ve PurePursuit ile yol izleme davranışı açısından incelenmektedir.

(a) (b) (c)

Şekil 4.29: 50x50 boyutlarındaki arenalarda RRT algoritma ile planlanan yollar

Simülasyon deneylerinde başlangıçta geliştirilen VFH-PurePursuit tabanlı algoritmasından farklı olarak yol başlangıç ve hedef konum arasında rastgele ağaç algoritması ile engelsiz bir yol planlanmıştır. Bu yol planlaması şu şekilde yapılır;

RRT algoritması rastgele bir düğüme odaklanır ve bu düğümü, kendisine en yakın bulunan düğüm ile birleştirir. Böylece her iki düğüm arasında bir yol oluşur. Oluşturulan bu yolun bir engelle çarpışmadığı kontrol edilir. Kontrol doğrultusunda herhangi bir engel ile temas olmazsa yol ve düğüm ağaca eklenir. Eğer engel ile teması var ise oluşturulan yolda hareketin mümkün olmadığına karar verilir ve yeni bir düğüm ve seçilen düğüme en yakın düğüm seçilerek aynı hesaplamalar tekrar yapılmaktadır.

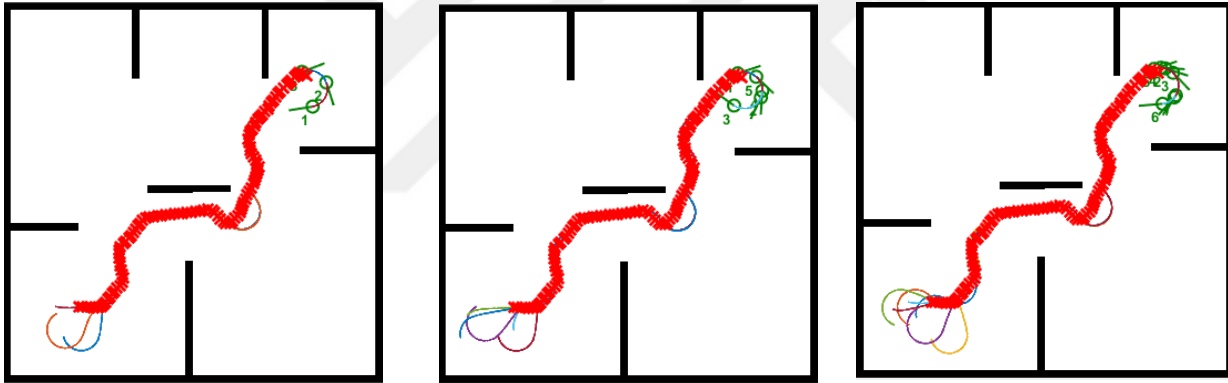
Bu işlem için algoritmada 30000 iterasyon sayısı ile bu yineleme yol bulunana kadar devam etmektedir.

Simülasyon deneylerinde robotlar, RRT algoritması ile oluşturulan yolda şekil 4.28’de gösterilen 50x50 boyutlu 3 farklı arenalarda belirli başlangıç aralığında rastgele olarak dağıtılmaktadır. Rastgele dağıtılan robotlar birbirleriyle üst üste binecek şekilde aynı konumlarda yerleşmeleri için, rastgele dağılım tekrarlanır. Bu sayede sürü robotları aynı noktalarda ve konumlarda olmayacak şekilde rastgele olarak dağıtılacaktır.

PurePursuit- RRT algoritması için simülasyon deneylerinde kullanılan sürü robotları PurePursuit- VFH algoritması için kullanılan sürü robotu ile aynıdır. Şekil 4.28’de görüldüğü gibi simülasyon deney ortamı dört tarafı ve içerisi engellerle sınırlandırılmış kare yapıdadır. Kare ortamın duvarları engel olarak belirtilmiştir.

4.2.2 Simülasyon Sonuçları

Bu bölümde önerilen yol planlama ve engellerden kaçınma algoritması farklı arenalarda, farklı robot sayılarında ve farklı l_d değerinde simülasyon deney sonuçları incelenmiştir. Sırasıyla arena (1), arena (2) ve arena (3) için gerçekleşen sonuçlar gösterilecektir. İleriye bakma mesafesi (l_d) ile oluşturulan grafikler ve izleme hatası grafikleri gösterilecektir. Kutu grafikleri ile ise simülasyon deneylerindeki yinleme sayıları karşılaştırılarak gösterilecektir. Kutu grafiklerinde orta düz çizgi ile belirtilen değer yinleme sayılarının ortanca değeridir. Robotların katettikleri mesafelerin ortalamaları gösterilecek bir çizelgede gösterilmektedir.

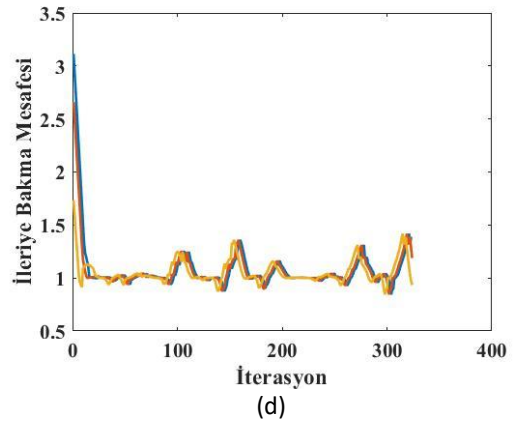
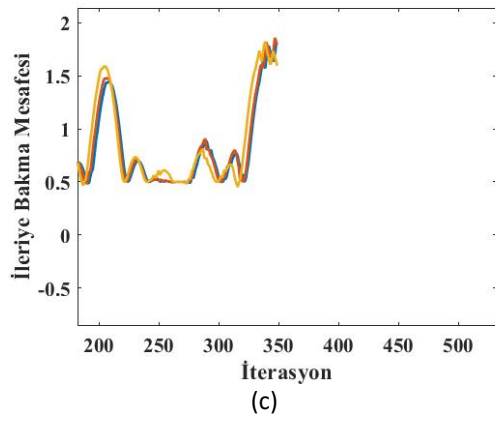
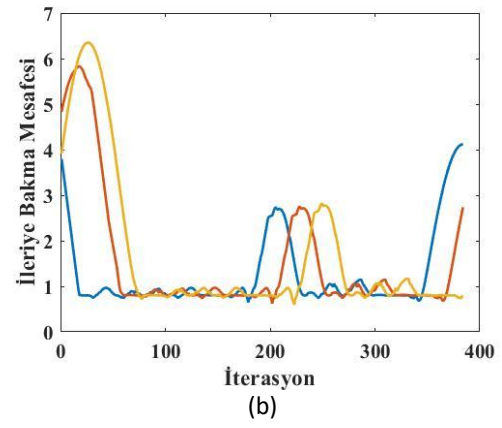
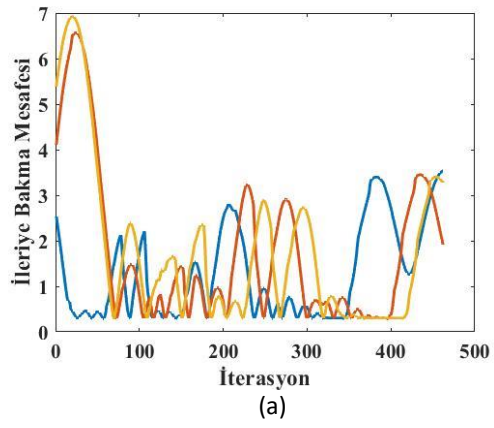


(a) (b)

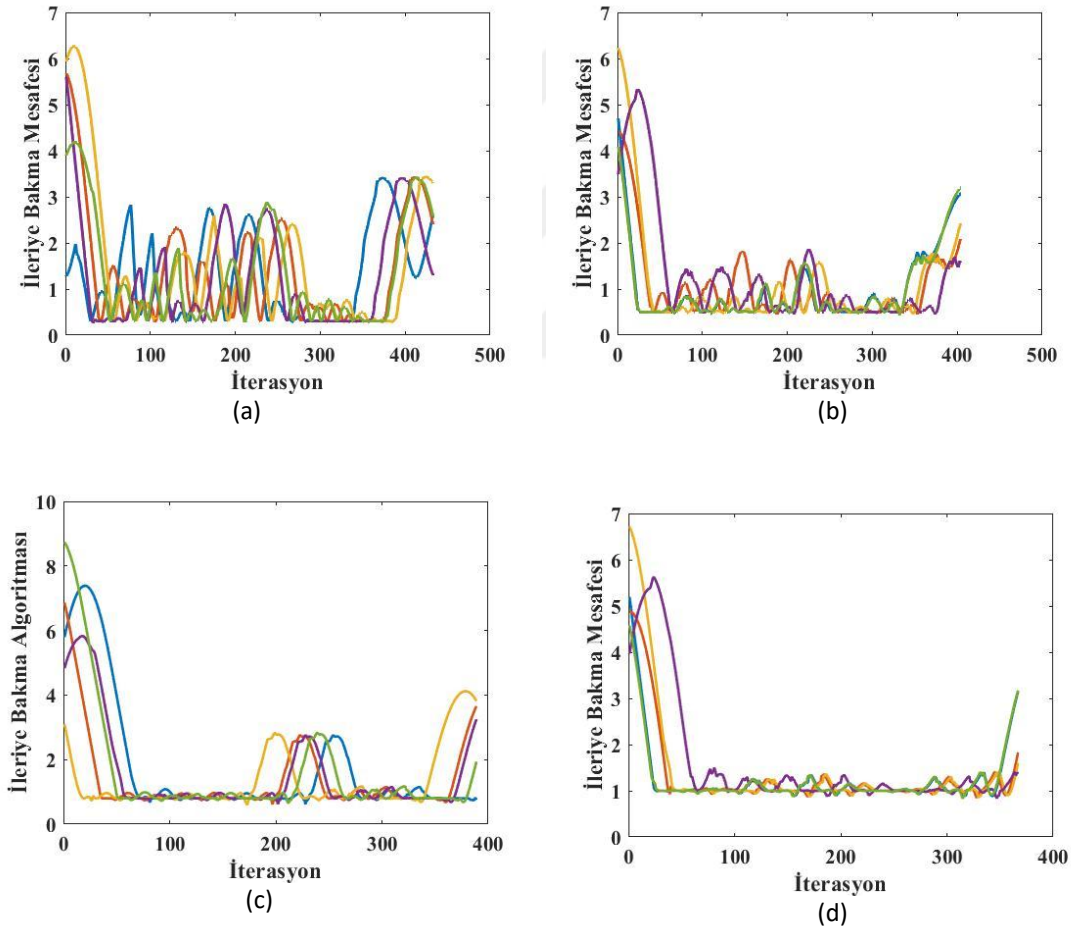
(c)

Şekil 4.30: RRT- PurePursuit algoritması için arena (1)'de simülasyon deney sonuçları (a) 3 sürü robotu, (b) 5 sürü robotu ve (c) 7 sürü robotu

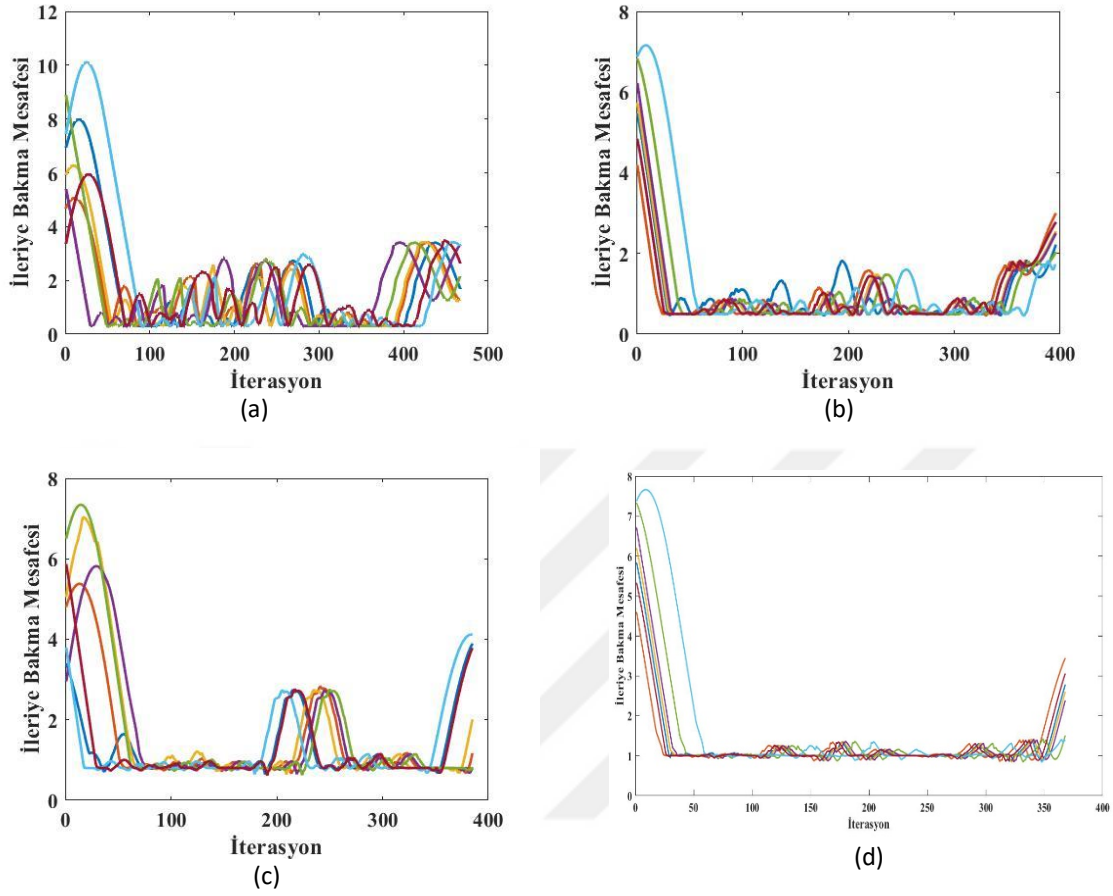
Arena (1)'de gerçekleştirilen simülasyon deney sonuçları şekil 4.30'da gösterilmiştir. Sırasıyla 3,5 ve 7 sürü robotu ile gerçekleştirilen başlangıç konumdan başlayarak planlanan ve engelsiz yol boyunca giderek hedef konuma varmaktadır. Yapılan deneylerin ileriye bakma mesafesine göre iterasyon sayıları Şekil 4.31, 4.32 ve 4.33 gösterilecektir.



Şekil 4.31: RRT -PurePursuit algoritması arena (1), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan simülasyon sonuçları



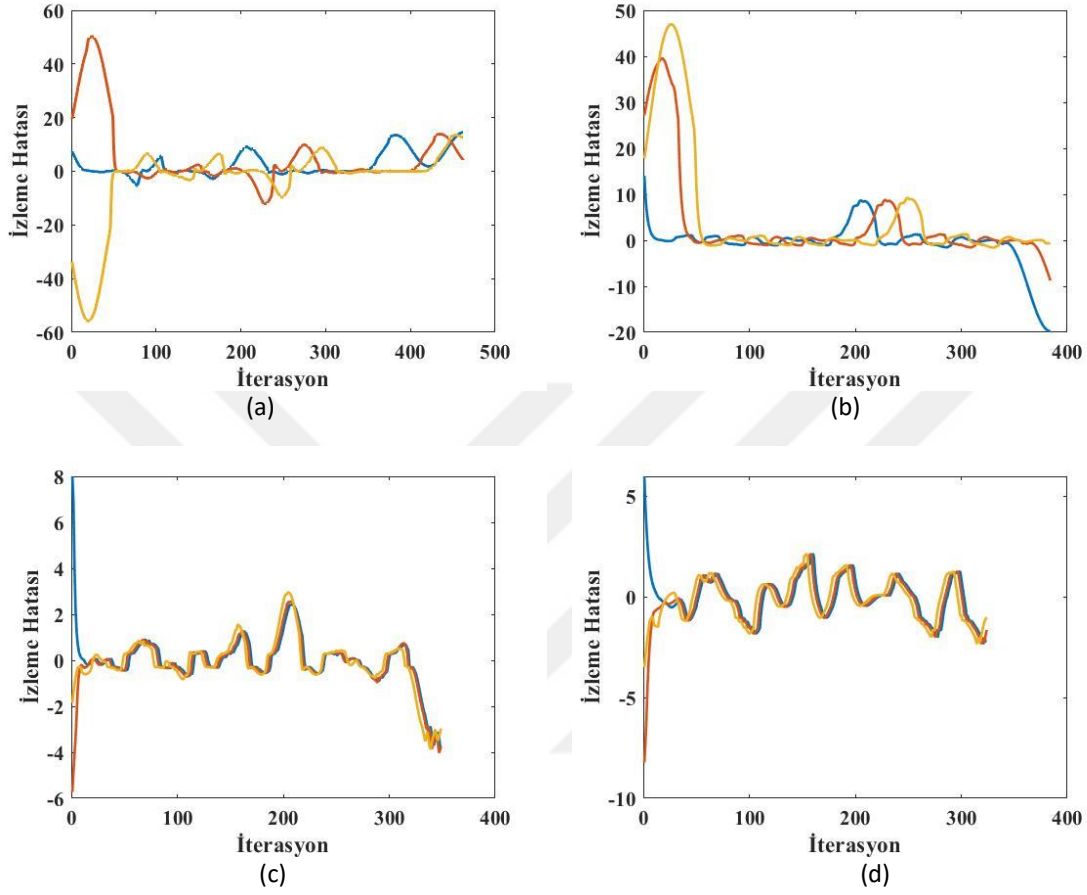
Şekil 4.32: Pure Pursuit- RRT algoritması arena (1), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan simülasyon sonuçları



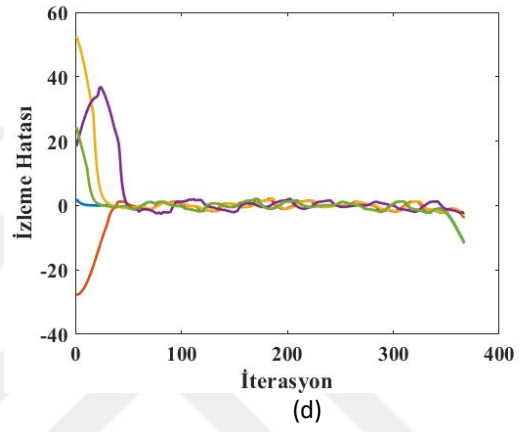
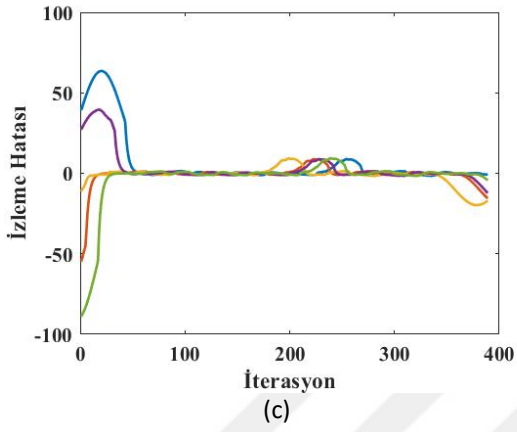
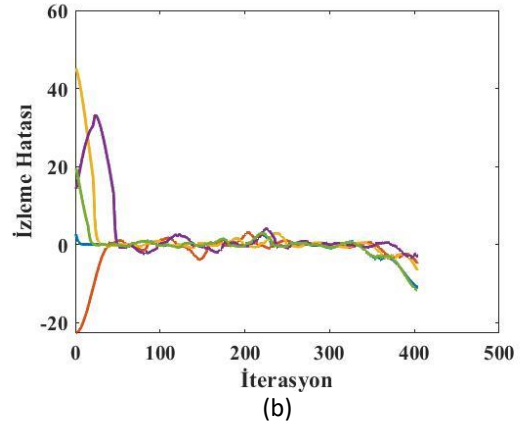
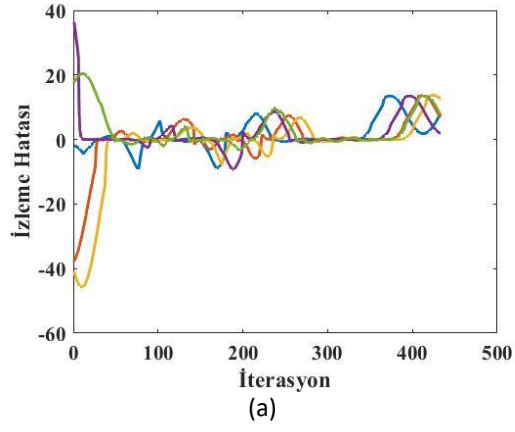
Şekil 4.33: Pure Pursuit- RRT algoritması arena (1), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan simülasyon sonuçları

Şekil 4.31,4.32 ve 4.33 grafiklerinde sırasıyla ileriye bakma mesafesi (l_d) 0.3, 0.5, 0.8 ve 1 olarak gösterilmektedir. İleriye bakma mesafesi arttıkça deneyin tamamlanma süresi (yineleme sayısı) azalacaktır. PurePursuit- RRT geliştirilen algoritmada VFH algoritmasından farklı olarak deney tamamlanma süresi daha azdır bunun sebebi sürü robotları harekete başlamadan başlangıç ve hedef konum arasındaki yolun belirli ve planlanmış engelsiz bir yol olmasıdır.

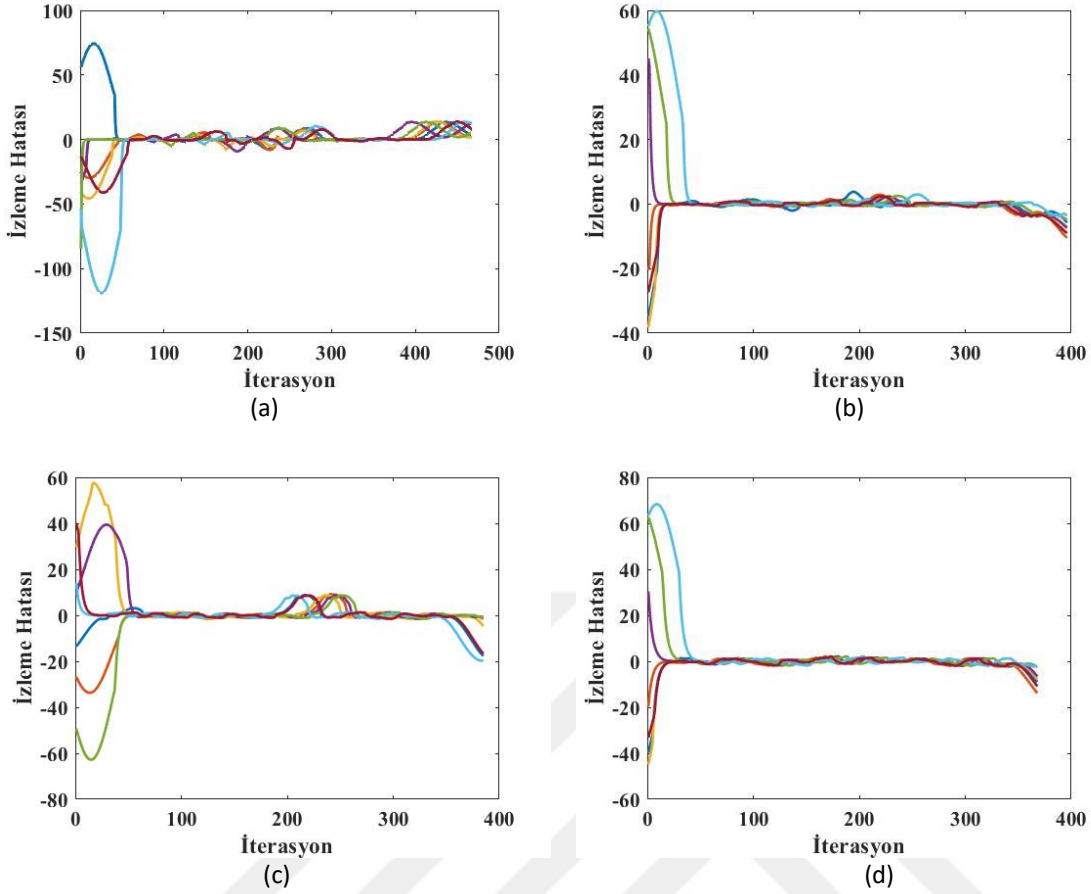
Şekil 4.30'daki arenada yapılan simülasyon deneylerinin sırasıyla 3,5 ve 7 robotlarının izleme hatası grafikleri gösterilecektir.



Şekil 4.34:PurePursuit -RRT algoritması arena (1), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan izleme hatası

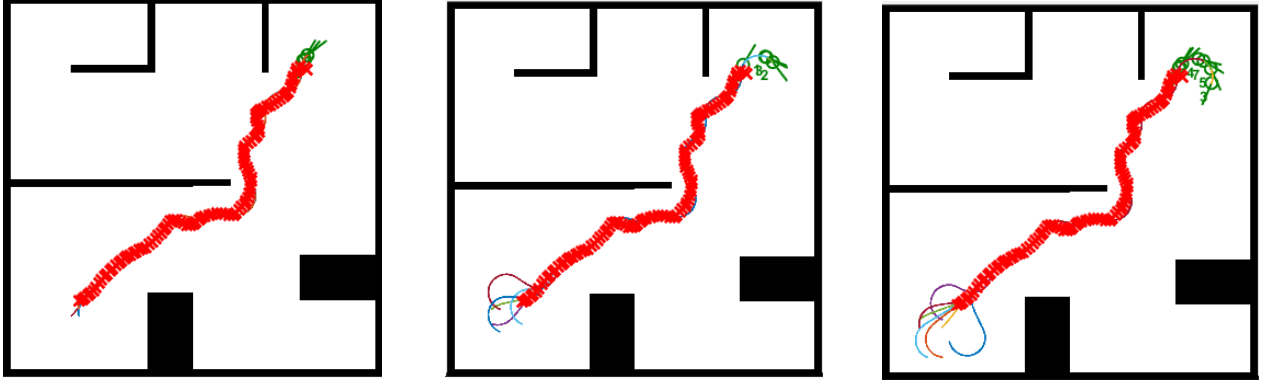


Şekil 4.35: PurePursuit -RRT algoritması arena (1), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan izleme hatası



Şekil 4.36:Pure Pursuit -RRT algoritması arena (1), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan izleme hatası

Şekil 4.34, 4.35 ve 4.36’ deki grafiklerde izleme hataları gösterilmektedir. Sürü robotları başlangıç konum ile hedef konum arasındaki planlanan yolu takip ederekten sürü robotlarının herhangi bir engelle yaklaşması ile izleme hataları gözlemlenir. Ancak izleme hataları PurePursuit- VFH algoritması ile karşılaştırıldığında izleme hatası 0’a yakınsamaktadır. Bunun sebebi arena üzerinde bir engelle çarpışma ihtimalinin olmamasıdır. Ancak iterasyon200-250 aralığında arenada bulunan orta engelle yaklaşmasının sonucu olmuştur. Robotların öncelik amacı bir engel ile çarpışmayı önlemektir.



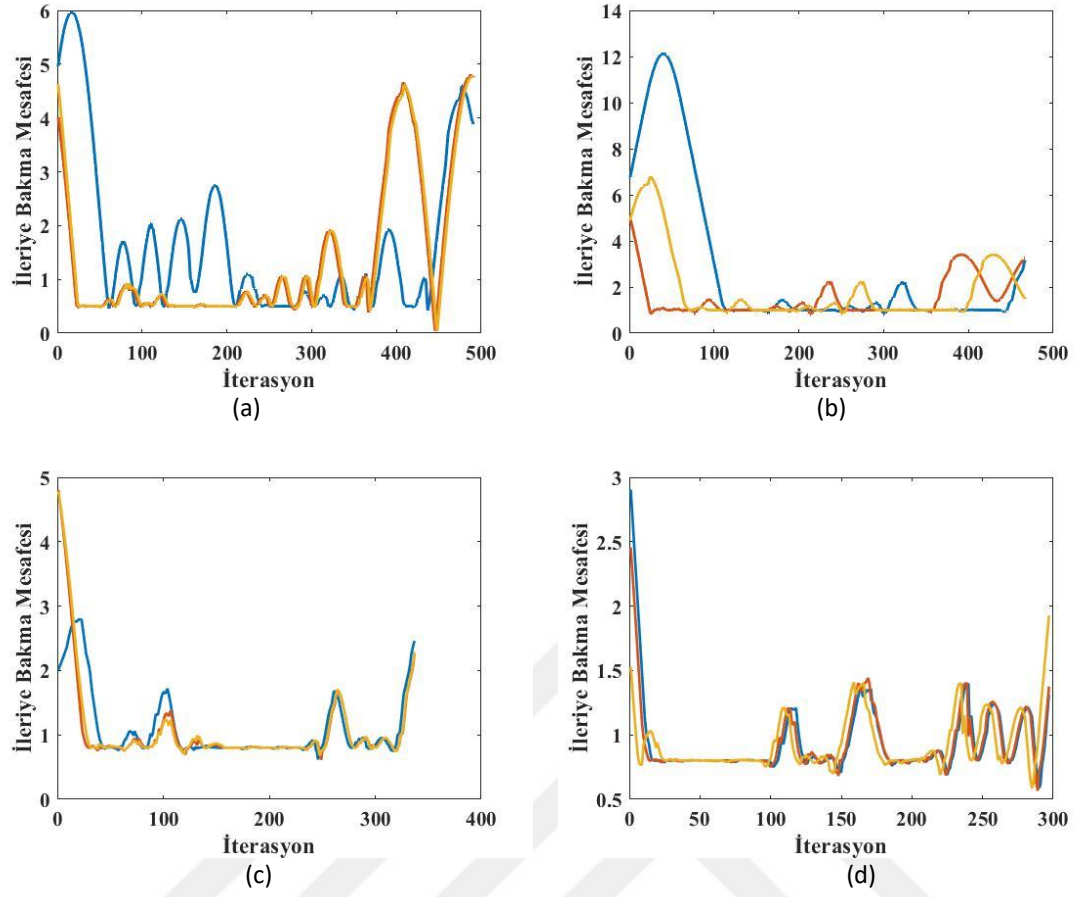
(a)

(b)

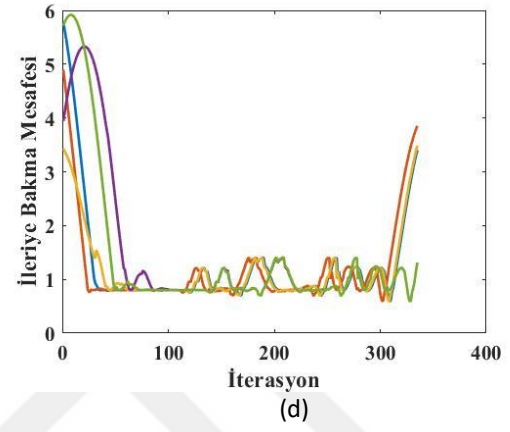
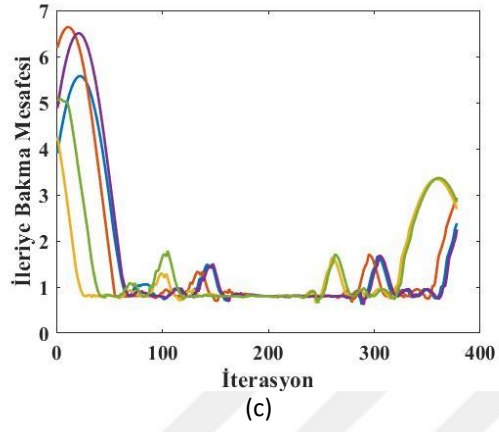
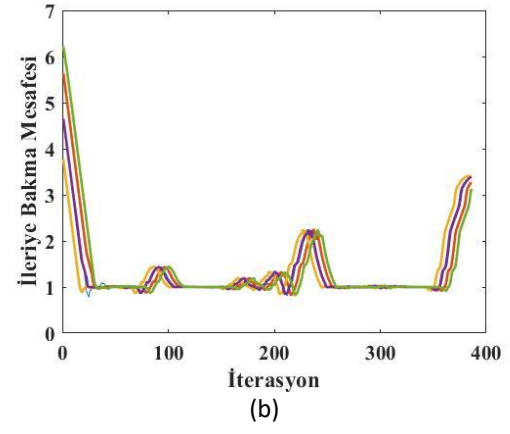
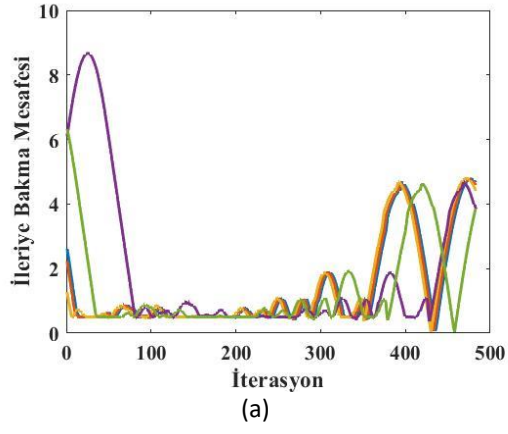
(c)

Şekil 4.37:Pure Pursuit- RRT algoritması için arena (2)’de simülasyon deney sonuçları (a) 3 sürü robotu, (b) 5 sürü robotu ve (c) 7 sürü robotu

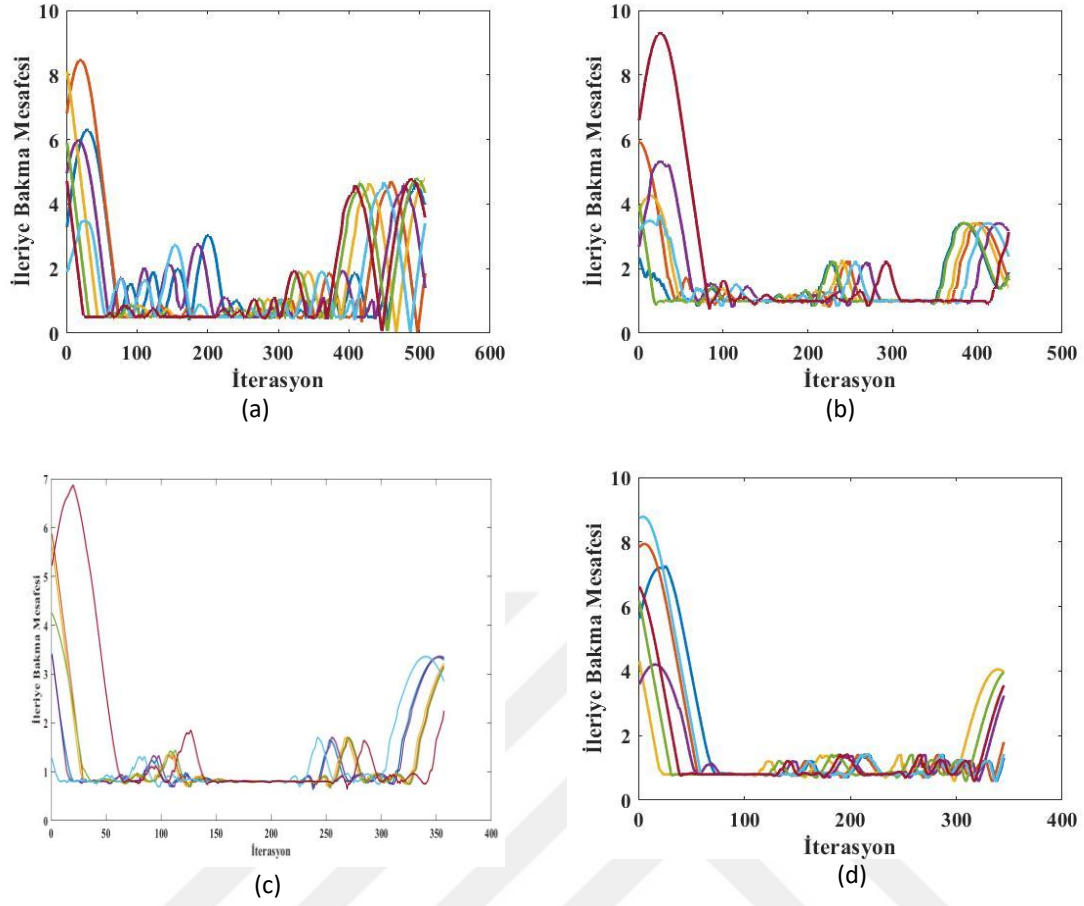
Arena (2)’de gerçekleştirilen simülasyon deneylerinin hedef konuma ulaştığı sonuçlar şekil 4.37’da gösterilmiştir. Sırasıyla 3,5 ve 7 sürü robotu ile gerçekleştirilen başlangıç konumdan başlayarak planlanan ve engelsiz yol boyunca giderek hedef konuma varmaktadırlar. İleriye bakma mesafeleri ve simülasyon tamamlanma süreleri (yineleme sayısı) ile oluşturulan grafikler gösterilecektir.



Şekil 4.38:Pure Pursuit- RRT algoritması arena (2), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan simülasyon sonuçları

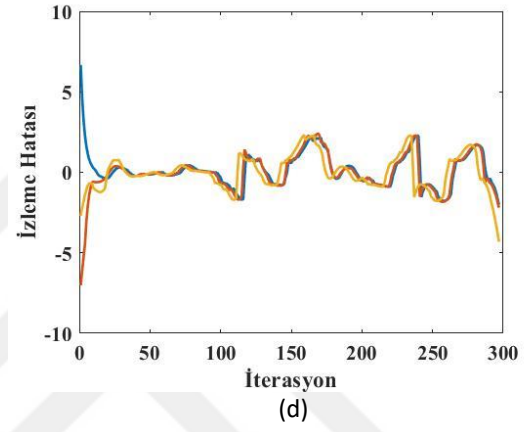
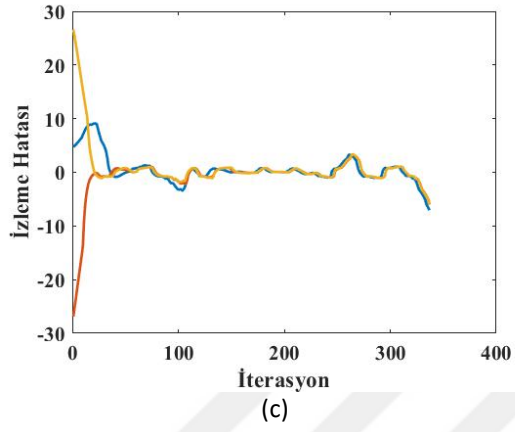
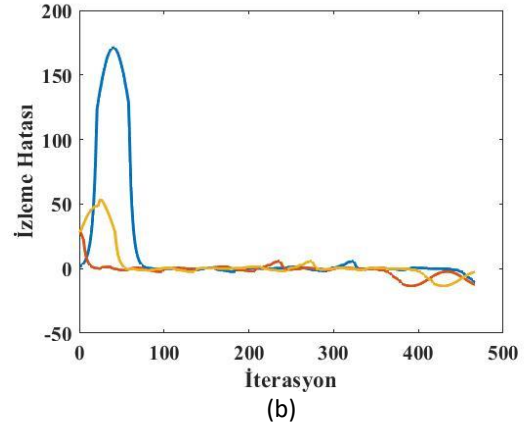
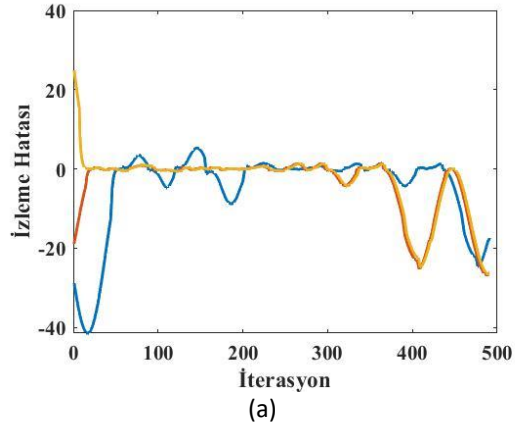


Şekil 4.39: PurePursuit- RRT algoritması arena (2), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan simülasyon sonuçları

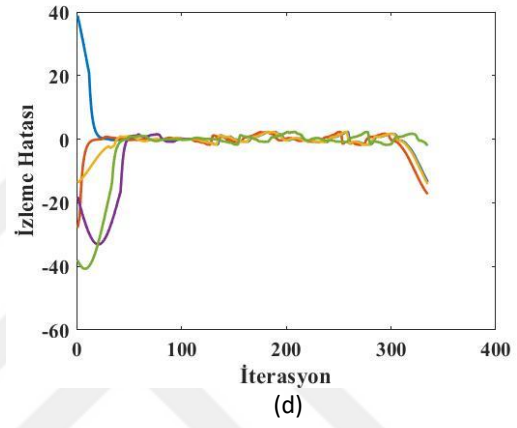
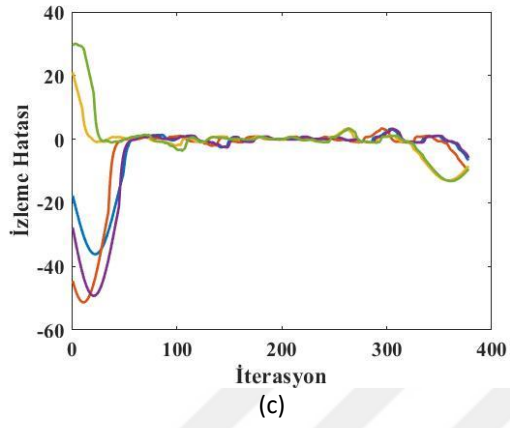
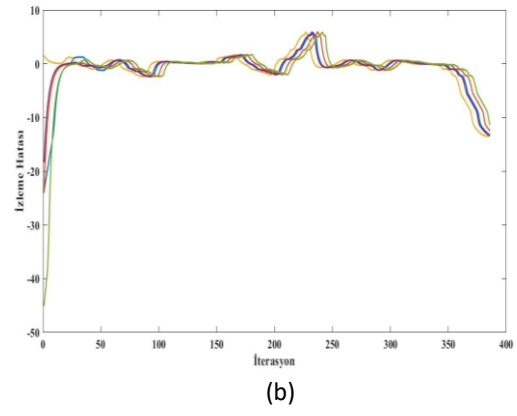
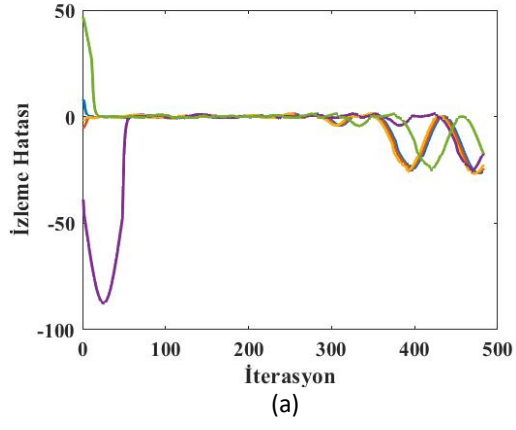


Şekil 4.40: PurePursuit- RRT algoritması arena (2), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan simülasyon sonuçları

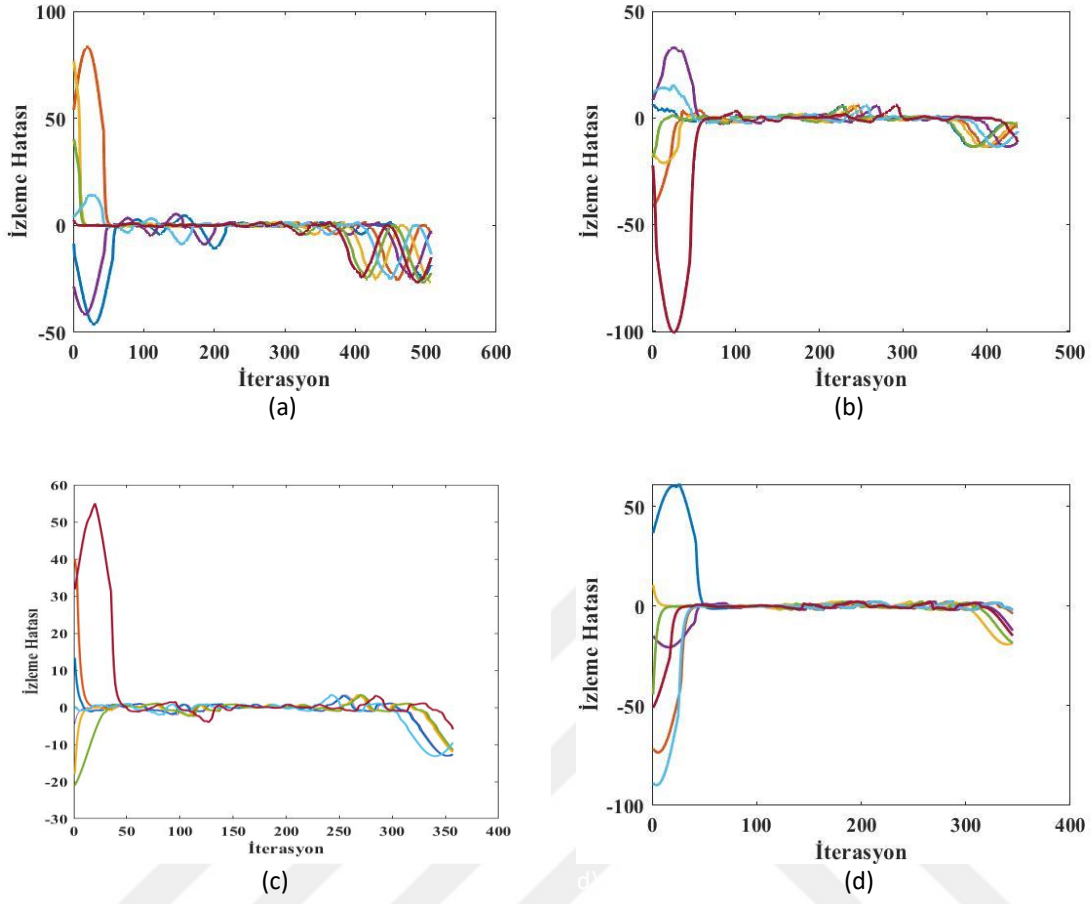
Şekil 4.38, 4.39 ve 4.40'ta grafikler göz önüne alındığında hedef konuma varmak için 7 robot ve ileriye bakma mesafesi (l_d) 0.3 değeri ile yapılan simülasyonlarda deney tamamlanma süresi daha fazladır. En kısa simülasyon tamamlanma süresi ise 3 robot ve ileriye bakma mesafesi (l_d) 1 değeri ile gerçekleşen deneyde görülmüştür. Bu sonucun sebebi sürü robotlarının sayısının artması ve ileriye bakma mesafesinin artış göstermesidir.



Şekil 4.41: Pure Pursuit -RRT algoritması arena (2), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan izleme hatası

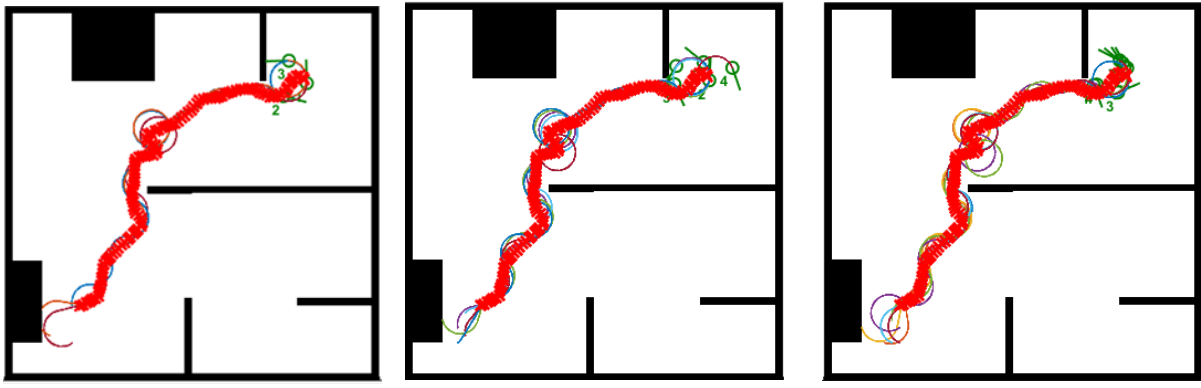


Şekil 4.42:Pure Pursuit -RRT algoritması arena (2), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan izleme hatası



Şekil 4.43:Pure Pursuit -RRT algoritması arena (2), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan izleme hatası

Şekil 4.41, 4.42 ve 4.43'te arena (2) için izleme hataları gözlemlenmiştir. Grafiklerden görüleceği üzere başlangıç belirlenen yolu her robotun bulması ve takip etmesinde ve simülasyon sonunda bir izleme hatası tespit edilmiştir. Bunun sebebinde planlanan ve takip edilecek yola olan mesafesi artacaktır.



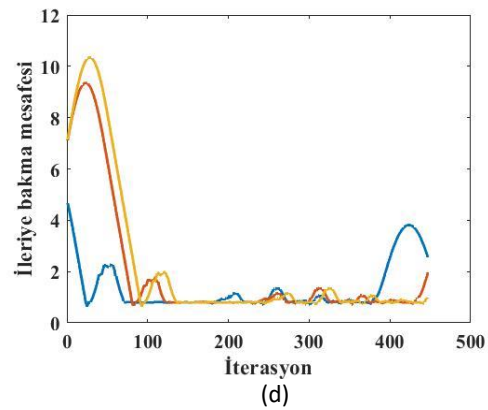
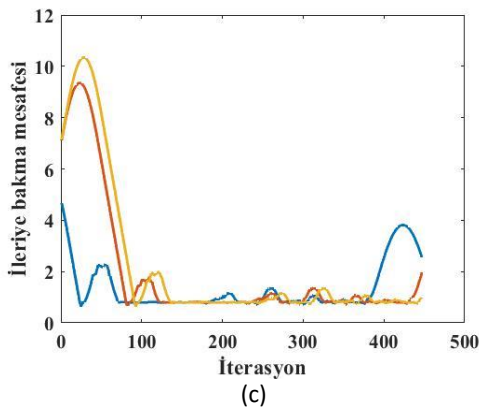
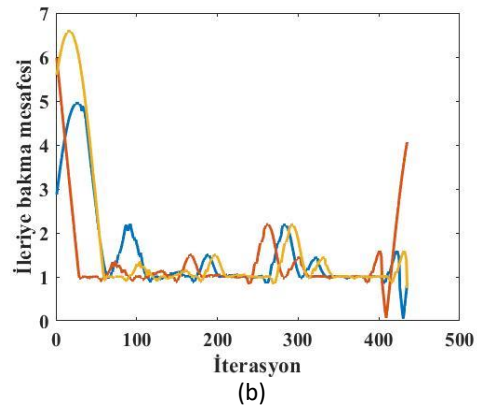
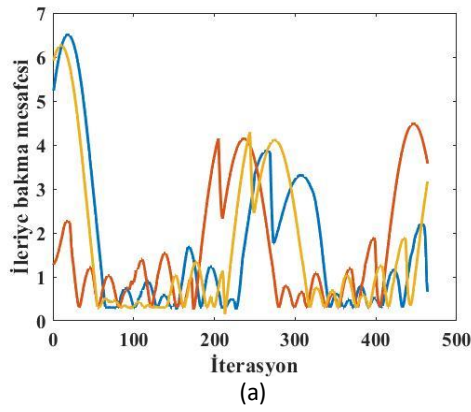
(a)

(b)

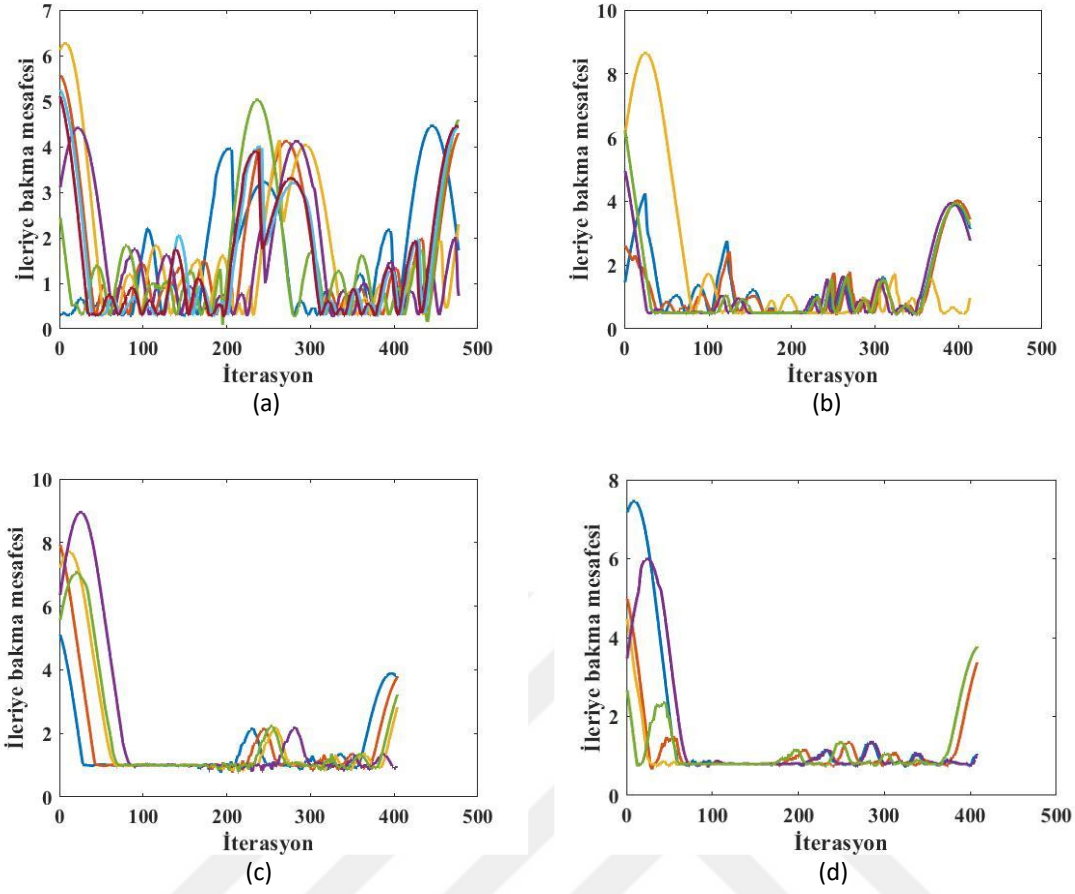
(c)

Şekil 4.44: Pure Pursuit- RRT algoritması için arena (3)'de simülasyon deney sonuçları (a) 3 sürü robotu, (b) 5 sürü robotu ve (c) 7 sürü robotu

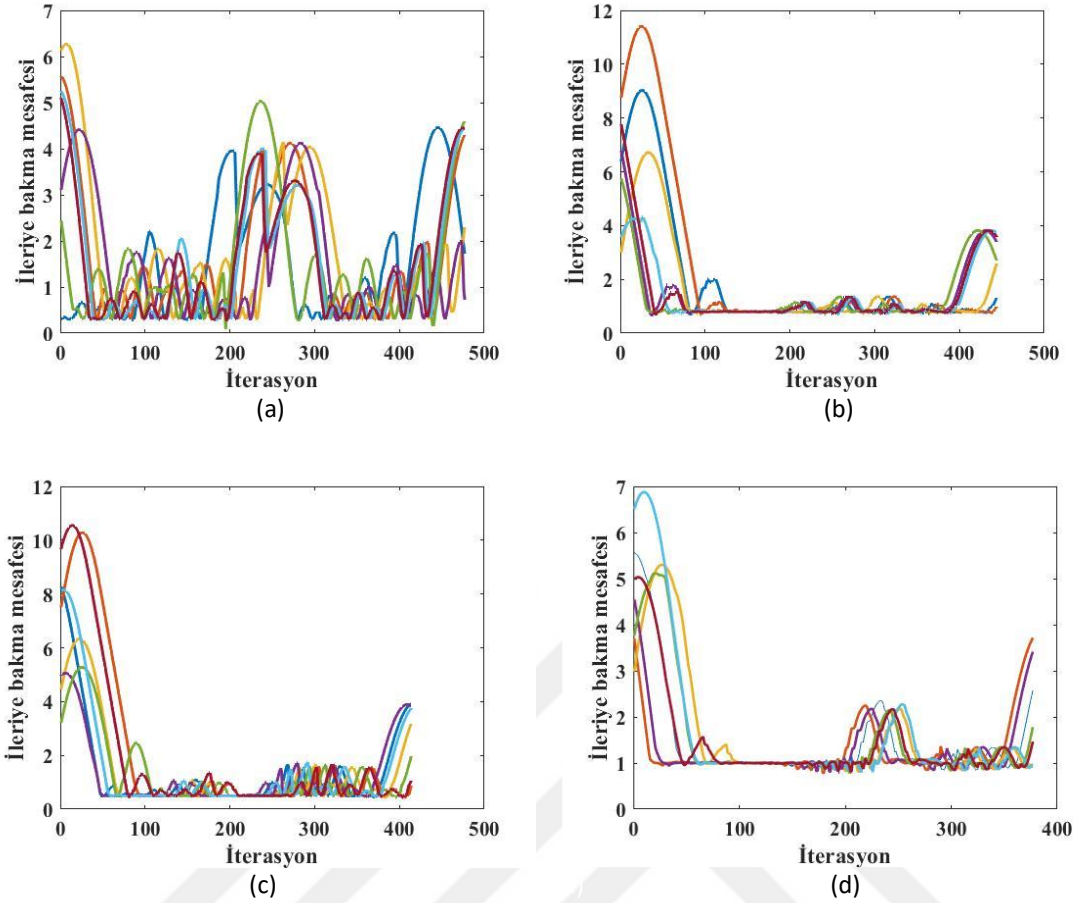
Arena (3)'de gerçekleştirilen simülasyon deneylerinin hedef konuma ulaştığı sonuçlar şekil 4.44'te gösterilmiştir. Sırasıyla 3,5 ve 7 sürü robotu ile gerçekleştirilen başlangıç konumdan başlayarak planlanan ve engelsiz yol boyunca giderek hedef konuma ulaşmaktadır.



Şekil 4.45: PurePursuit- RRT algoritması arena (3), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan simülasyon sonuçları



Şekil 4.46:Pure Pursuit- RRT algoritması arena (3), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan simülasyon sonuçları

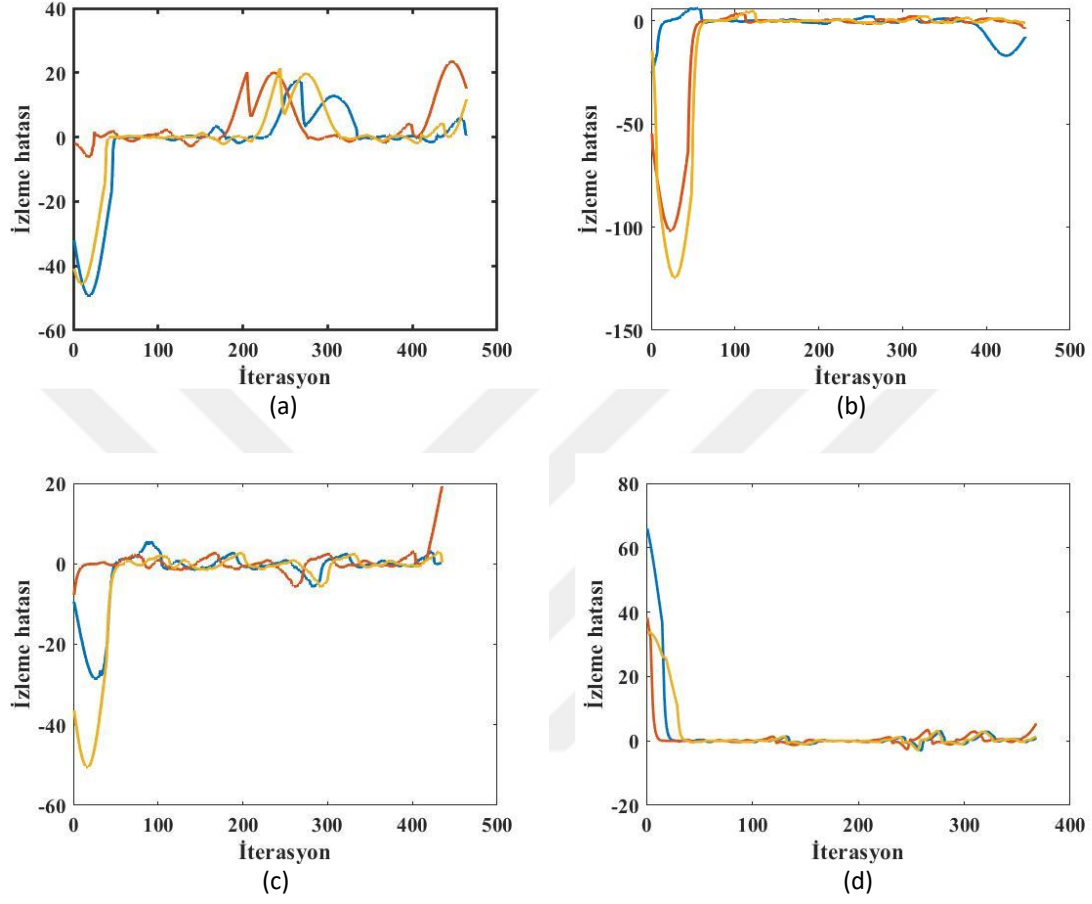


Şekil 4.47: Pure Pursuit- RRT algoritması arena (3), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan simülasyon sonuçları

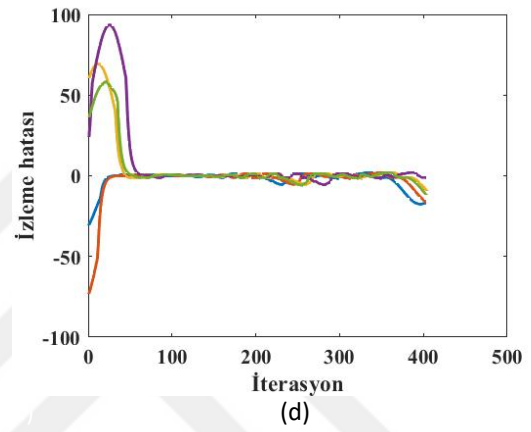
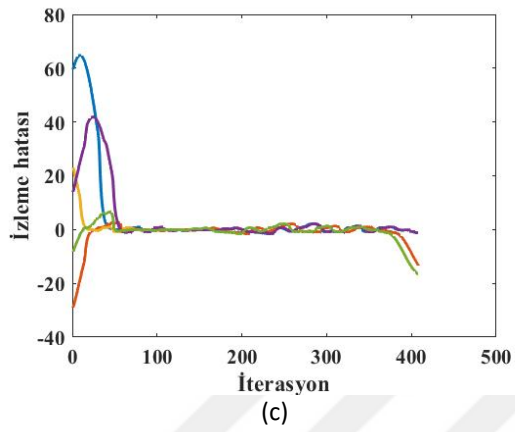
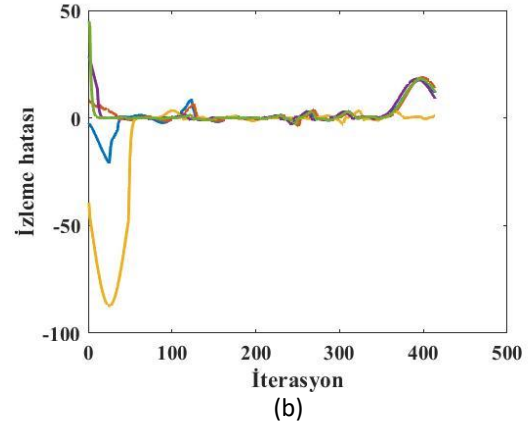
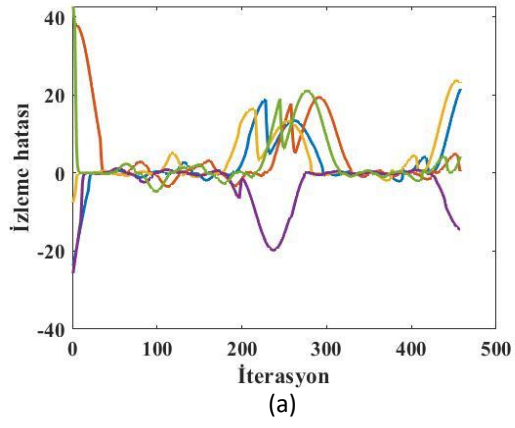
Şekil 4.45, 4.46 ve 4.47’de grafikler göz önüne alındığında hedef konuma varmak için 7 robot ve ileriye bakma mesafesi (l_d) 0.3 değeri ile yapılan simülasyonlarda deney tamamlanma süresi daha fazladır. En kısa simülasyon tamamlanma süresi ise 3 robot ve ileriye bakma mesafesi (l_d) 1 değeri ile gerçekleşen deneyde görülmüştür. Bu sonucun sebebi sürü robotlarının sayısının artması ve ileriye bakma mesafesinin artış göstermesidir.

Geliştirilen PurePursuit- VFH algoritmalarında RRT algoritması ile karşılaştırıldığında ileriye bakma mesafesi ve robot sayısında artış simülasyon süresinde (yineleme sayısı) çok fazla değişim göstermemektedir. Bunun sebebi RRT algoritması ile belirlenen arena üzerinde başlangıç konum ile hedef konum arasında engelsiz bir yol planlanması ve geliştirilen algoritma ile sürü robotlarının bu planlanmış yolu takip etmeleri amaçlanmıştır. Bu nedenle engelsiz yol sürü robotları harekete başlamadan planlandığı için sürü robotlarının arena içerisinde hedef konumu bulmalarında kolaylık sağlanmıştır.

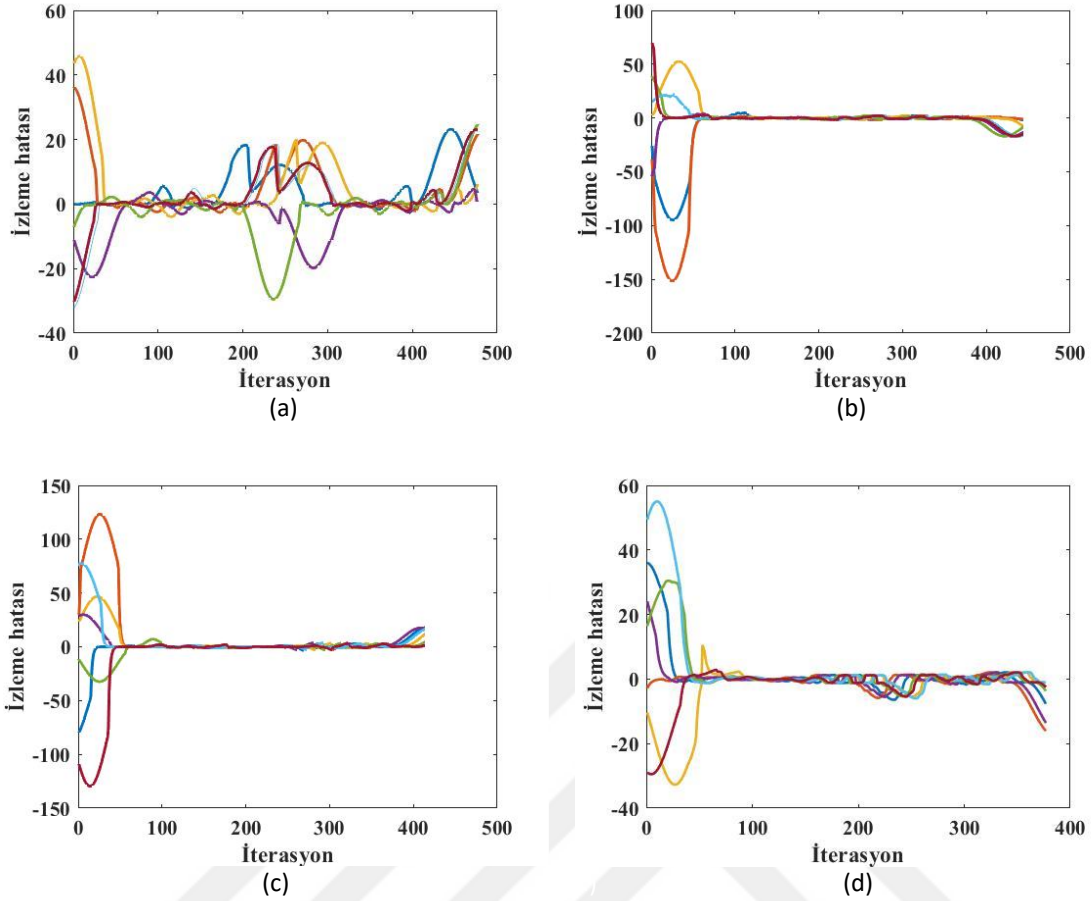
Geliştirilen PurePursuit- RRT iş birliğine dayalı algoritmanın arena (3) simülasyon deney izleme hata sonuçları şekil 4.48, 4.49 ve 4.50 gösterilmiştir.



Şekil 4.48:Pure Pursuit -RRT algoritması arena (3), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan izleme hatası



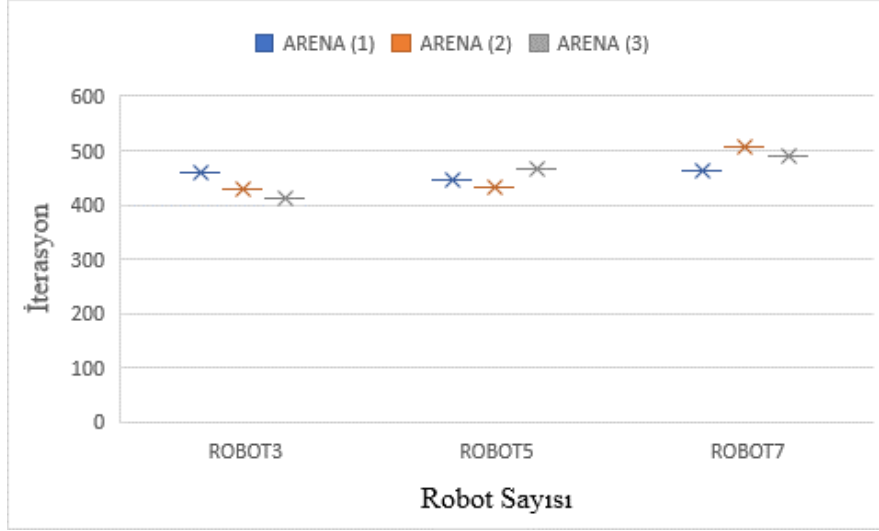
Şekil 4.49: Pure Pursuit -RRT algoritması arena (3), (a) 0.3, (b) 0.5, (c) 0.8, (d) 1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan izleme hatası



Şekil 4.50: PurePursuit -RRT algoritması arena (3), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan izleme hatası

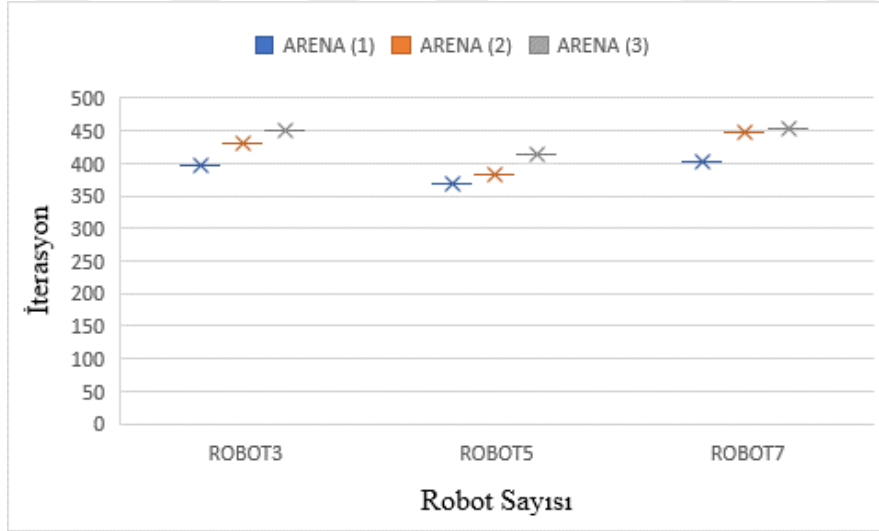
Şekilde görüldüğü izleme hataları gözlenmektedir. Planlanan yolda kıvrım oluşmuş, keskin bir dönüş ile robotlar yolu takip etmiştir. Bu planlanan yolda robotlar birbirlerine çarpmadan yolu keskin dönüş ile takip ettikleri yola olan mesafeleri artmıştır. Keskin dönüş yapan ve birbirleri ile çarpışmadan kaçınan sürü robotları takip edecekleri yola tekrardan organize şekilde yaklaşarak hedef konuma doğru hareket etmektedirler.

Sürü robotları için RRT- PurePursuit tabanlı iş birliğine dayalı engelden kaçınma ve yol planlama algoritması sonuçları için kutu grafiği şeklinde incelenmiştir. Kutu grafiklerinde orta düz çizgi ile belirtilen değer yineleme sayılarının ortanca değeridir. 0.3 birim l_d değeri için arena (1), arena (2) ve arena (3) ile uygulanan sistematik deneylerde 3, 5, 7 sürü robotları için deney sonuçları, 433 iterasyon adımımda en düşük ortanca değer arena (2)'de 5 sürü robotu ile, en yüksek ortanca değer ise 467 iterasyon adımı olan arena (3)' de 5 sürü robotu ile gerçekleşmiştir.



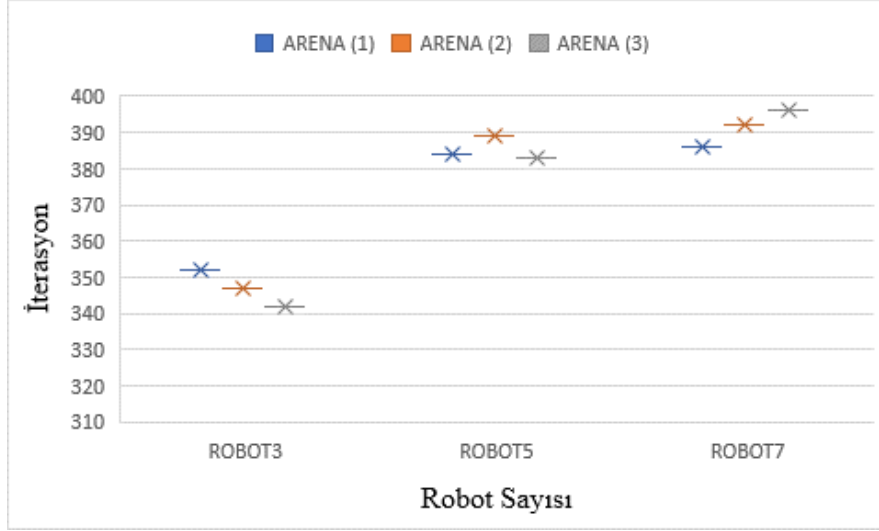
Şekil 4.51: 0.3 birim değer için RRT algoritması için uygulanan sistematik test sonuçları

Şekil 4.52’de geliştirilen PurePursuit – RRT algoritması için ileriye bakma mesafesi (l_d) 0.5 değerli sistematik deney sonuçlarının kutu grafiği yer almaktadır. En düşük ortalama değer 368 iterasyon adımı arena (1) 5 robot için, en yüksek ortalama değer ise 414 iterasyon adımı arena (3)’de 5 robot için gerçekleşmiştir.

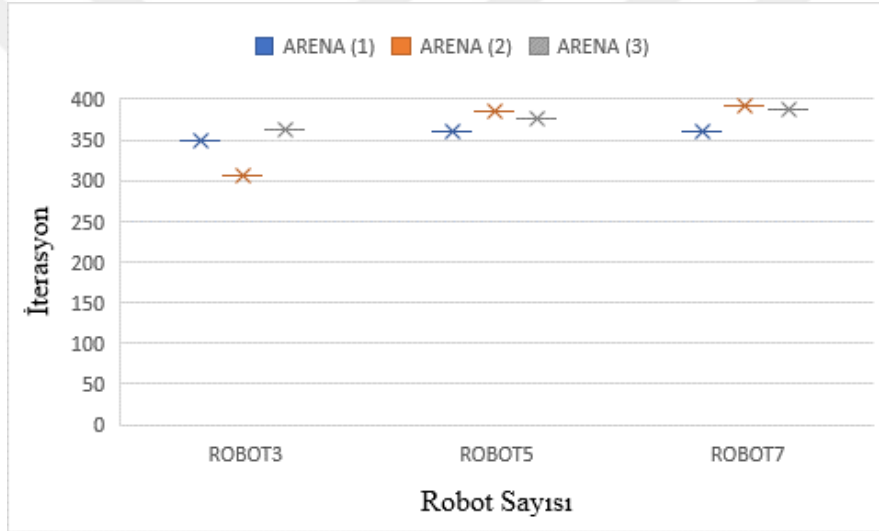


Şekil 4.52: 0.5 birim değer için RRT algoritması için uygulanan sistematik test sonuçları

0.8 birim değerli ileriye bakma mesafesi için arena (1), arena (2) ve arena (3) ile uygulanan sistematik deneylerde 3, 5, 7 sürü robotları için deney sonuçları, 384 iterasyon adımı en düşük ortalama değer arena (1) 5 robot ile, 389 iterasyon adımı ise en yüksek ortalama değer arena (2) 5 robot gerçekleşmiştir.



Şekil 4.53: 0.8 birim değer için RRT algoritması için uygulanan sistematik test sonuçları



Şekil 4.54: 1 birim değer için RRT algoritması için uygulanan sistematik test sonuçları

1 birim değerli ileriye bakma mesafesi için arena (1), arena (2) ve arena (3) ile uygulanan sistematik deneylerde 3, 5, 7 sürü robotları için deney sonuçları, 377 iterasyon adımı en düşük ortalama değer arena (3) 5 robot ile, 386 iterasyon adımı en yüksek ortalama değer ise arena (2) 5 robot gerçekleşmiştir.

Geliştirilen PurePursuit- RRT algoritması ile 4.1’de açıklanan PurePursuit-VFH iş birliğine dayalı algoritma karşılaştırıldığında simülasyon süreleri arenalar arası ve robotlar arasında yakınlık göstermektedirler. En yüksek ve en düşük ortalama iterasyon değerleri üzerinde görülmektedir. Geliştirilen 2 algoritma karşılaştırıldıklarında VFH algoritmasında simülasyon tamamlanma süresine göre daha uzun sürmektedir.

PurePursuit- RRT İş birliğine Dayalı Algoritma için Katedilen Mesafe	
ARENA (1)	62 cm
ARENA (2)	54 cm
ARENA (3)	57 cm

Çizelge 4:Pure Pursuit – RRT algoritma için sürü robotlarının katettikleri ortalama mesafe

Çizelge 4’te geliştirilen sürü robotları için PurePursuit- RRT iş birliğine dayalı algorithmada her bir arena için 36 sistematik deney sonuçlarında sürü robotlarında her robotunun başlangıç konumdan hedef konuma ulaşmak için katettikleri yolun ortalama değeri gösterilmektedir. Çizelgeye göre arena (1)’ de en uzun mesafe, en kısa mesafe ise arena (2)’de görülmektedir.

4.3.Sürü Robotları için PRM- PurePursuit Tabanlı İş Birliğine Dayalı Engelden Kaçınma ve Yol Planlama

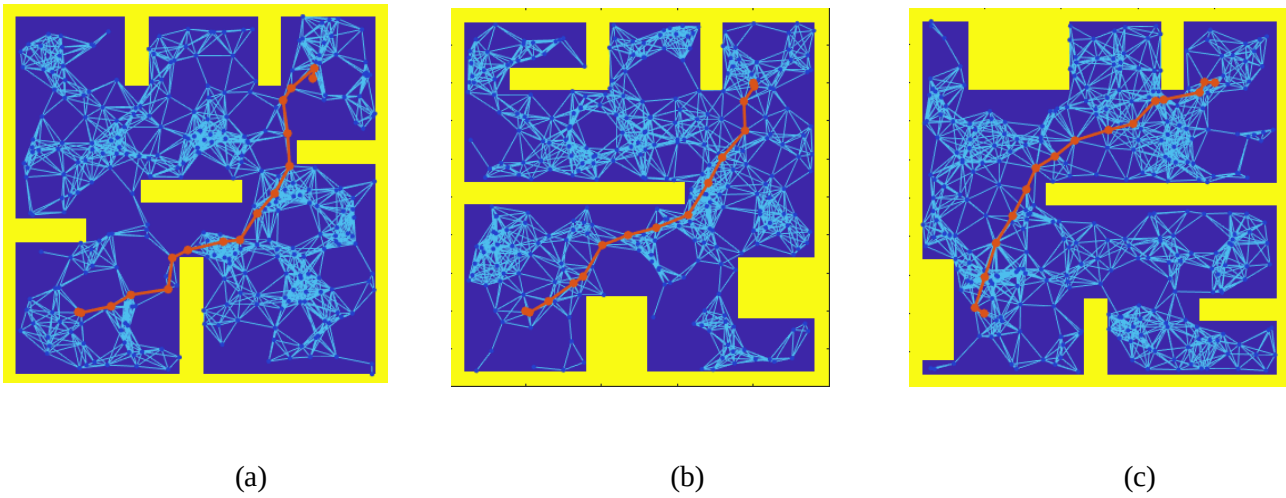
Geliştirilen sürü robotları içinPRM-PurePursuit tabanlı iş birliğine dayalı engelden kaçınma ve yol planlama algoritması için Çizelge 5’teki gibi simülasyon ortamında sistematik deneyler uygulanmıştır. Sistematik deneyler arena boyutu ve farklı arenalarda, robot algılama yarıçapı, l_d değerine, robot sayısına bağlı olarak belirlenmiştir. Uygulanan deneylerindeki sürü robotlarının, değişen arenalar (engeller) ve değişen l_d değerleri ile ölçeklenebilir bir şekilde farklı robot sayılarında yol planlama ve engelden kaçınma davranışları uygulayarak başlangıç konumunda hedefe konuma en kısa yol oluşturarak simülasyonunun tamamlanması amaçlanmıştır. PRM algoritması ile farklı arenalarda rastgele noktalar üreterek bu noktalar arasında bağlantılar oluşturarak başlangıç konum ile hedef konum arasında yol planlanmaktadır.Robotun hareketi için çarpışmasız bir yol Dijkstra’nın en kısa yol sorgusuyla elde edilir. Planlanan yolu takip edebilmek için Pure-Pursuit algoritması kullanılmıştır. PurePursuit l_d değerinin değişikliği ile sürü robotlarının ölçeklenebilirliği ve esnekliği başlangıç konum ile hedef konum arasında engelsiz yol ve oluşturulan yolun izlenebilirliğini açısından gözlemlenmektedir.

4.3.1.Simülasyon Kurulumu

Arena boyutu	{50x50}
l_d Değeri	{0.3, 0.5, 0.8, 1}
Robot sayısı	{3, 5, 7}
Düğüm sayısı	300
Bağlantı mesafesi	5

Çizelge 5: PRM Simülasyon özellikleri

Çizelge 5’te belirlenen robot sayıları {3, 5, 7}, PurePursuit yöntemi için l_d değeri {0.3, 0.5, 0.8, 1} ve arena boyutları aynıdır fakat 50x50 boyutlarında farklı arenalarda bağlı düğüm sayısı 300 tane olarak belirlenmiş ve düğümler arenalar içerisinde rastgele dağılmıştır. Bağlantı mesafesi olarak 5 birim olarak belirlenmiş ve bu özellikler doğrultusunda engelsiz yollar planlanmıştır. Planlanan bu yollar için 36 sistematik deney uygulanmıştır. Yol planlama ve engelden kaçınma probleminin performansını ölçmek için simülasyon süresi kullanılmıştır. Simülasyon süresi, deneylerin başlangıç konumundan hedef konuma ulaştığı ana kadar geçen simülasyon adım sayısıdır. Sistematik deneyler neticesinde elde edilen sonuçlarda, sürü robotlarının ölçeklenebilirliği ve esnekliği başlangıç ve hedef konum arasında engelsiz yol oluşturma ve PurePursuit ile yol izleme davranışı açısından incelenmektedir.



Şekil 4.55: 50x50 boyutlarındaki arenalarda PRM algoritma ile planlanan yollar

Simülasyon deneylerinde başlangıçta geliştirilen VFH-PurePursuit algoritmasından farklı olarak ancak RRT algoritması ile benzerlik göstererek yol başlangıç ve hedef konum arasında rastgele noktalar üreterek engelsiz bir yol planlanmıştır. Bu yol planlaması şu şekilde yapılır;

PRM yol planlayıcısı 2 bölümden oluşmaktadır. Bu algoritma öğrenme aşaması ve sorgulama aşamasından oluşmaktadır. İlk olarak güvenlik faktörü oluşturmak için sürü robotu yarıçapı kadar arena genişletilir. Güvenlik faktörü kadar genişletilen arena üzerinde, 300 adet rastgele noktalar oluşmakta ve üretilen noktaların boş alanda olması kontrol edilmektedir. Üretilen noktalar boş alanda ise 5 birim bağlantı mesafesi ile ağ grafiğine eklenmektedir. Oluşturulan noktalar arası bağlantının boş alanda olup olmadığı kontrol edilmektedir ve boş alanda ise bağlantı grafiğine eklenir. Bu işlem hedef konuma ulaşana kadar devam etmektedir. Sorgulama aşamasında ise başlangıç ve hedef konumları grafiğe bağlanmaktadır. Robotun hareketi için çarpışmasız bir yol Dijkstra algoritmasının en kısa yol sorgusuyla elde edilir. Rastgele üretilen noktaların sayısı arttıkça, PRM algoritmanın yol bulamama olasılığı sıfıra yaklaşır. (Park & Yoon, 2018)

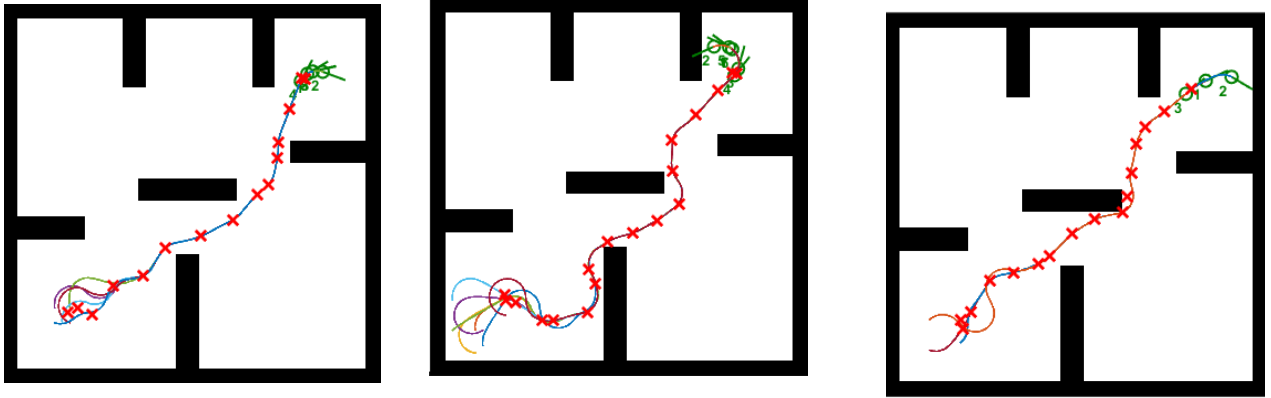
Simülasyon deneylerinde robotlar, PRM algoritması ile oluşturulan yolda şekil 4.53'te gösterilen 50x50 boyutlu 3 farklı arenalarda belirli başlangıç aralığında rastgele olarak dağıtılmaktadır. Rastgele dağıtılan robotlar birbirleriyle üst üste binecek şekilde aynı konumlarda yerleşmeleri için, rastgele dağılım tekrarlanır. Bu sayede sürü robotları aynı noktalarda ve konumlarda olmayacak şekilde belirli aralıkta rastgele olarak dağıtılacaktır.

PurePursuit- PRM algoritması için simülasyon deneylerinde kullanılan sürü robotları geliştirilen VFH ve RRT algoritmaları için kullanılan sürü robotu ile aynıdır. Şekil 4.53'te görüldüğü gibi simülasyon deney ortamı dört tarafı ve içerisi engellerle sınırlandırılmış kare yapıdadır. Kare ortamın duvarları engel olarak belirtilmiştir.

4.3.2 Simülasyon Sonuçları

Bu bölümde önerilen yol planlama ve engellerden kaçınma algoritması farklı arenalarda, farklı robot sayılarında ve farklı l_d değerinde simülasyon deney sonuçları incelenmiştir. Sırasıyla arena (1), arena (2) ve arena (3) için gerçekleşen sonuçlar gösterilecektir. İleriye bakma mesafesi (l_d) ile oluşturulan grafikler ve izleme hatası grafikleri gösterilecektir. Kutu grafikleri ile ise simülasyon deneylerindeki yinelenme sayıları karşılaştırılarak gösterilecektir.

Kutu grafiklerinde orta düz çizgi ile belirtilen değer yineleme sayılarının ortanca değeridir. Robotların katettikleri mesafelerin ortalamaları gösterilecek bir çizelgede gösterilecektir.

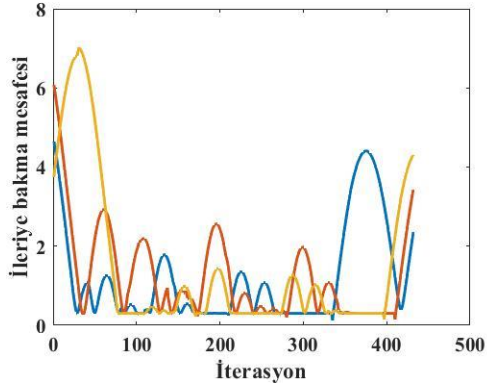


(a) (b) (c)

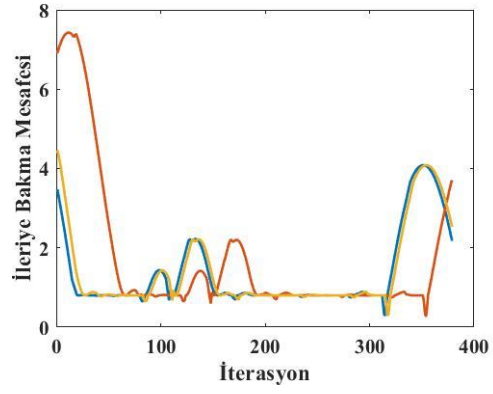
Şekil 4.56: PurePursuit- PRM algoritması için arena (1)'de simülasyon deney sonuçları (a) 3 sürü robotu, (b) 5 sürü robotu ve (c) 7 sürü robotu

Şekil 4.55 (a)'da görülen arena (1) oluşturulan yol ve simülasyon sonuçları şekil 4.56sırasıyla 3,5 ve7 sürü robotları için gösterilmektedir. Arena (1) üzerinde rastgele oluşturulan 300 adet noktalar üzerinde engelsiz alanlar üzerindeki noktalar arasında bağlantılar ile oluşturulan yollar gözlemlenmektedir. Sürü robotları başlangıç ve hedef konum arasında oluşturulan yolu, geliştirilenPurePursuit- RRT iş birliğine dayalı algoritma ile benzerlik göstererek arenada sürü robotları hareket başlamadan gideceği yol belirlenir ve belirlenen yol PurePursuit algoritması ileriye bakma mesafe (l_d) değeri sırası ile 0.3, 0.5, 0.8 ve 1 değiştirilerek sürü robotlarının ölçeklenebilirliği ve esnekliği yol takip etme davranışı ile incelenecektir.

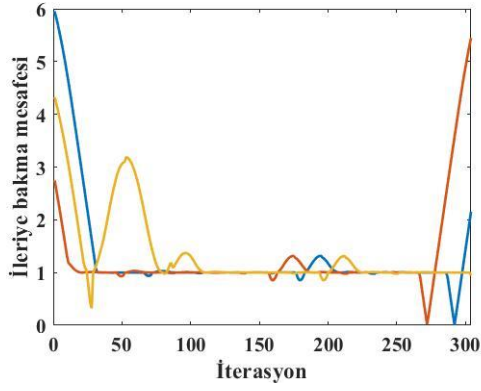
Şekil 4.57, 4.58 ve 4.59 ile sırasında 3, 5 ve 7 sürü robotlarının simülasyon tamamlanma süreleri (yineleme sayıları) gösterilmiştir.



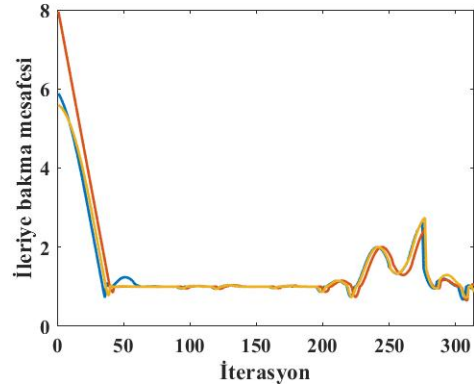
(a)



(b)

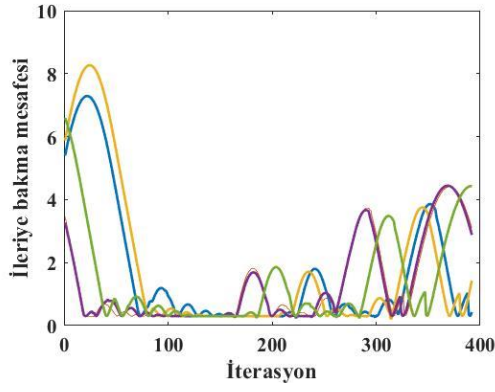


(c)

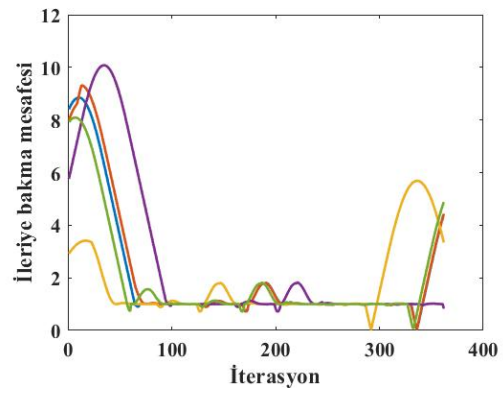


(d)

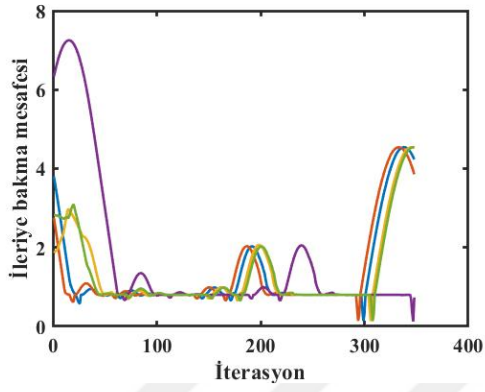
Şekil 4.57: PurePursuit- PRM algoritması arena (1), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan simülasyon sonuçları



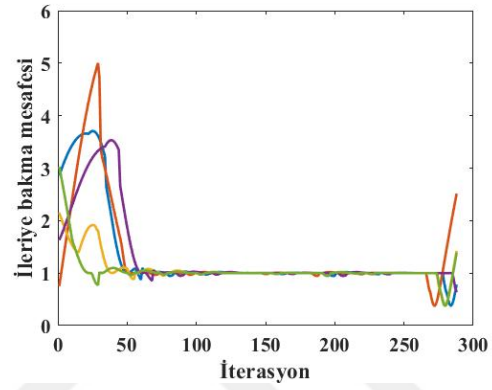
(a)



(b)

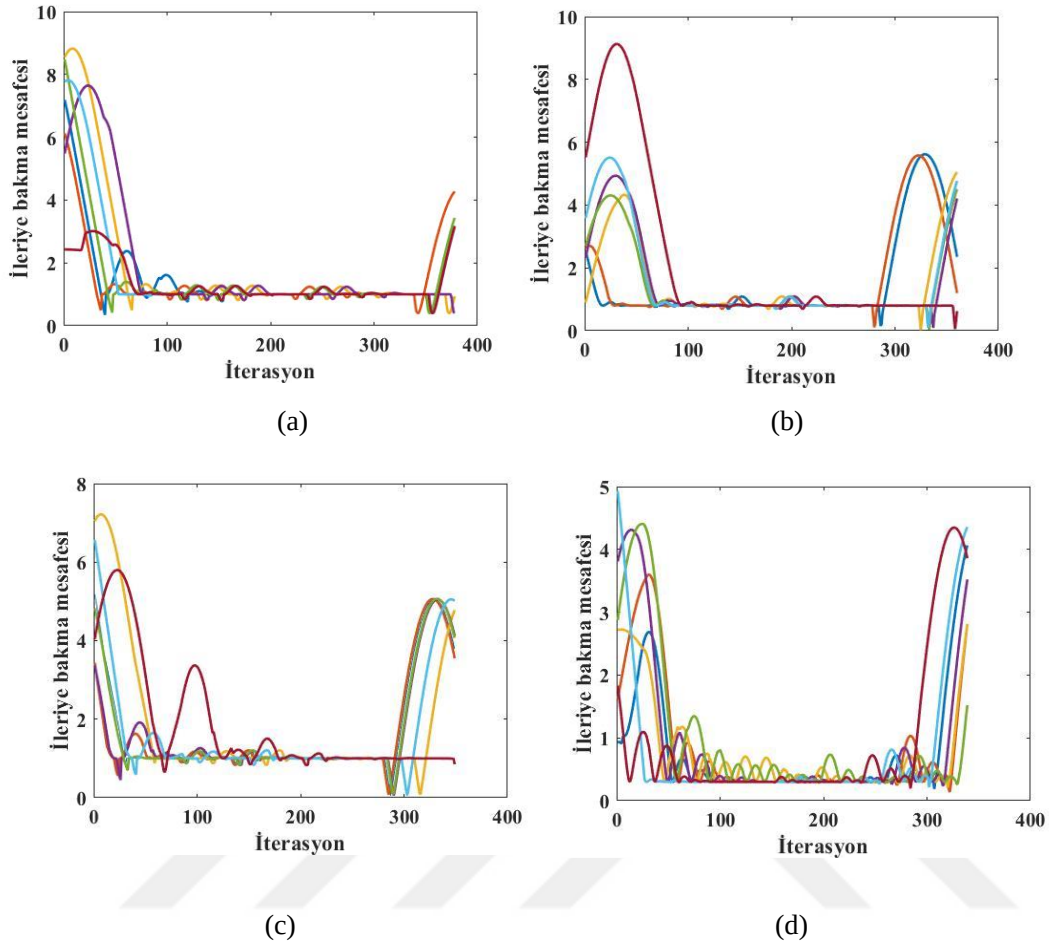


(c)



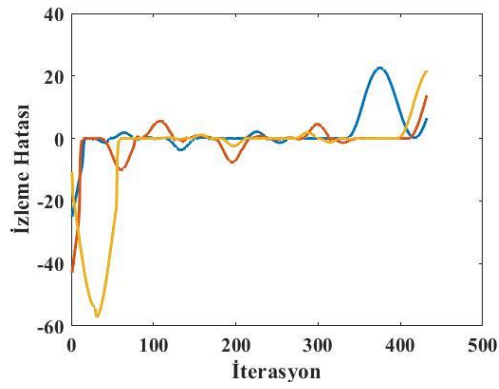
(d)

Şekil 4.58: PurePursuit- PRM algoritması arena (1), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan simülasyon sonuçları

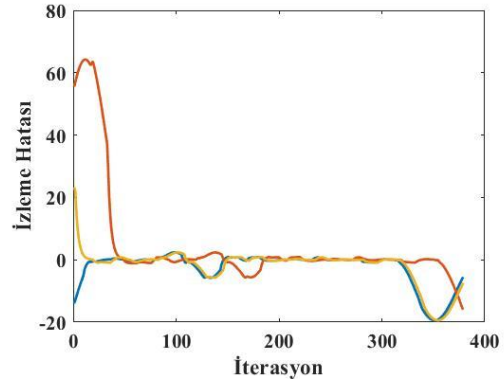


Şekil 4.59: PurePursuit- PRM algoritması arena (1), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan simülasyon sonuçları

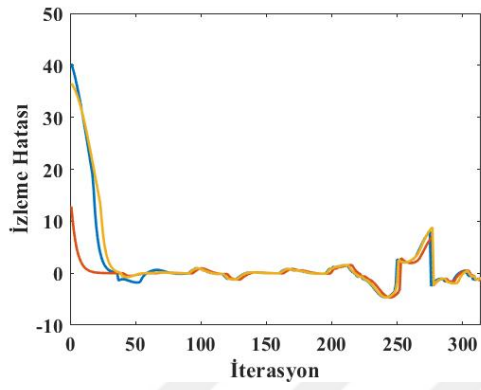
Belirlenen yol PurePursuit algoritması ileriye bakma mesafe (l_d) değeri sırası ile 0.3, 0.5, 0.8 ve 1 değiştirilerek sürü robotlarının ölçeklenebilirliği ve esnekliği yol izleme davranışıyla ile grafikler incelendiğinde l_d değeri 1 birim olarak görülen grafiklerde simülasyon tamamlanma süresi enazdır. Ancak arena üzerinde engelsiz yollar önceden belirlendiği için ileriye bakma mesafesi ve sürü robot sayıları değişiklikleri simülasyon süresini etkilememektedir.



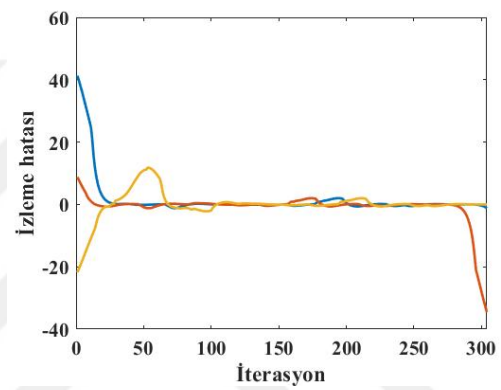
(a)



(b)

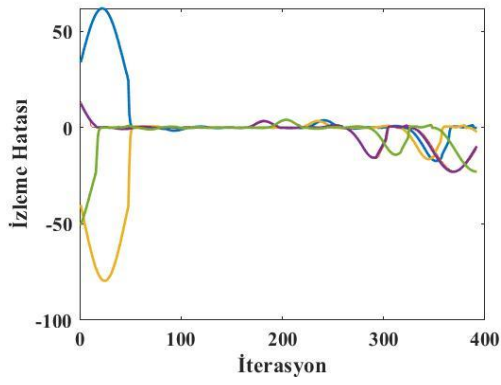


(c)

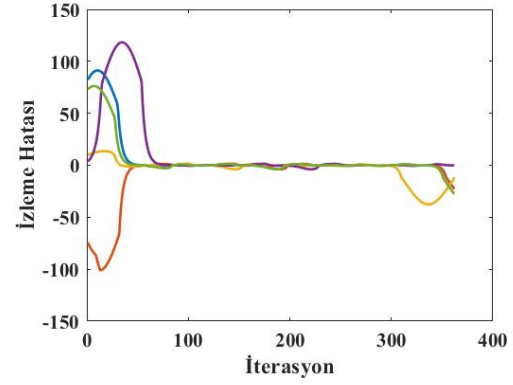


(d)

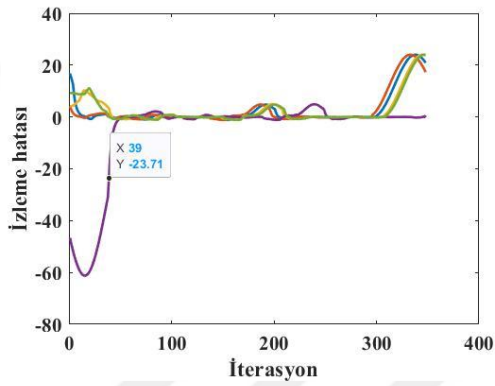
Şekil 4.60:Pure Pursuit -RRT algoritması arena (1), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan izleme hatası deney sonuçları



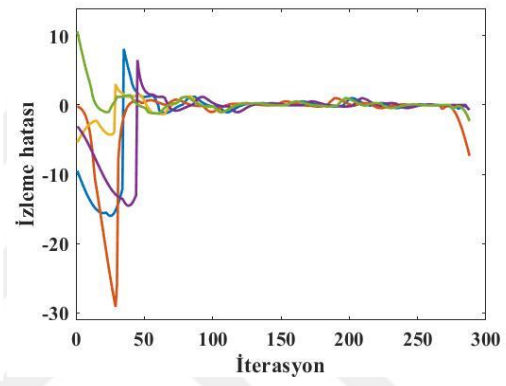
(a)



(b)

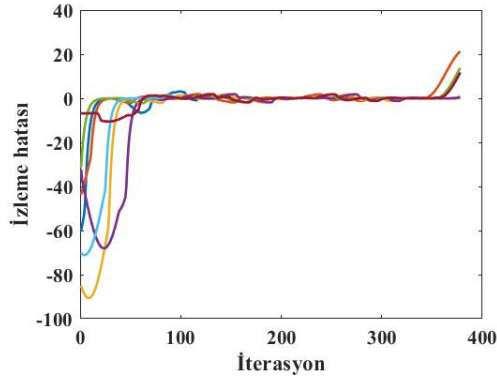


(c)

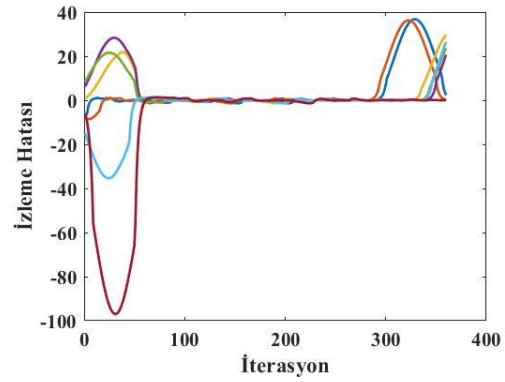


(d)

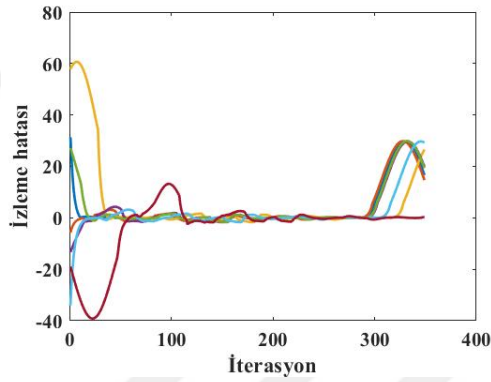
Şekil 4.61: Pure Pursuit -RRT algoritması arena (1), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan izleme hatası



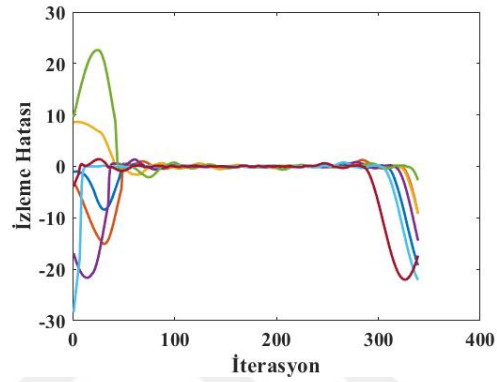
(a)



(b)

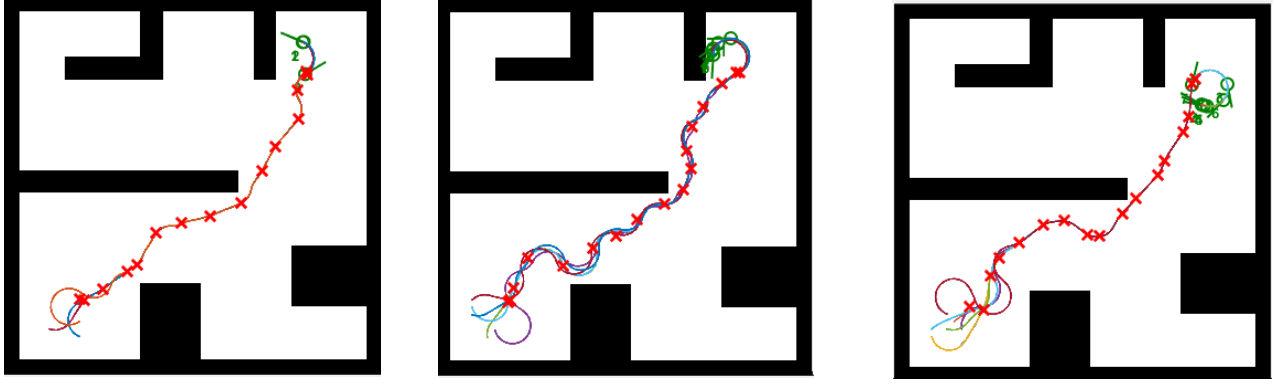


(c)



(d)

Şekil 4.62: Pure Pursuit -RRT algoritması arena (1), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan izleme hatası

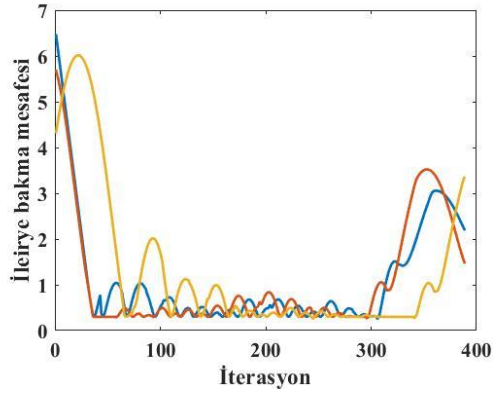


(a) (b) (c)

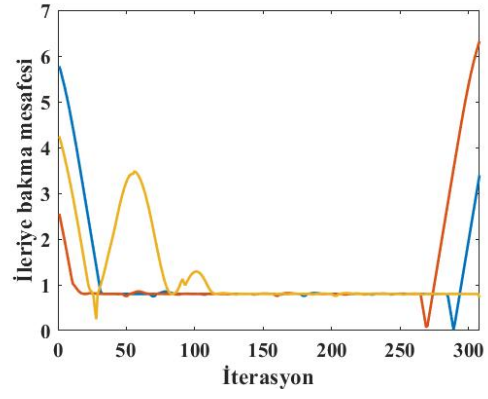
Şekil 4.63: PurePursuit- PRM algoritması için arena (2)'de simülasyon deney sonuçları (a) 3 sürü robotu, (b) 5 sürü robotu ve (c) 7 sürü robotu

Şekil 4.63 (a)'da görülen arena (2) oluşturulan yol ve simülasyon sonuçları şekil 4.63 sırasıyla 3,5 ve 7 sürü robotları için gösterilmektedir. Arena (2) üzerinde rastgele oluşturulan 300 adet noktalar üzerinde engelsiz alanlar üzerindeki noktalar arasında bağlantılar ile oluşturulan yollar gözlemlenmektedir. Sürü robotları başlangıç ve hedef konum arasında oluşturulan yolu, PurePursuit algoritması ileriye bakma mesafe (ld) değeri sırası ile 0.3, 0.5, 0.8 ve 1 değiştirilerek sürü robotlarının ölçeklenebilirliği ve esnekliği yol takip etme davranışı ile incelenecektir.

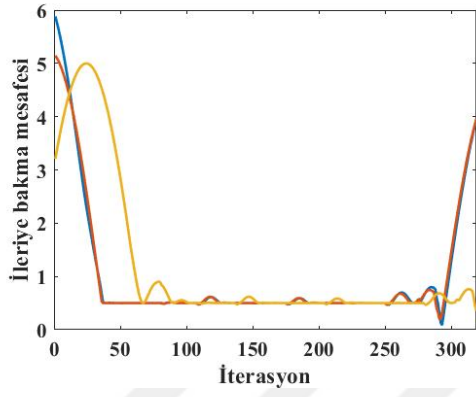
Şekil 4.64, 4.65 ve 4.66'de ileriye bakma mesafesine göre simülasyon zamanları (iterasyon sayıları) gösterilmektedir.



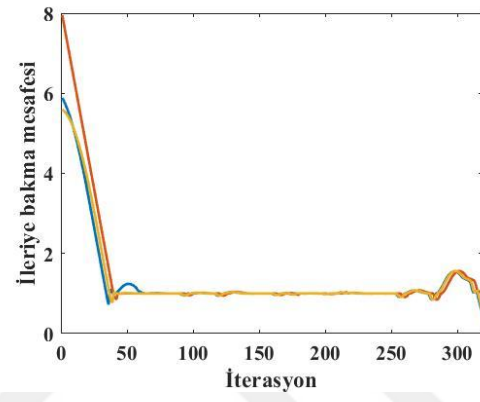
(a)



(b)

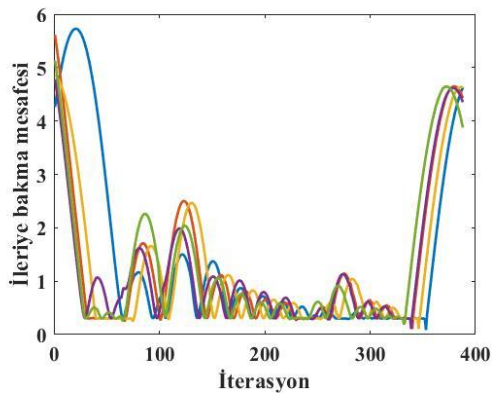


(c)

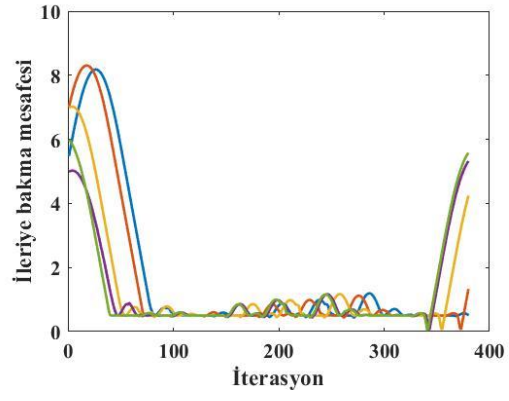


(d)

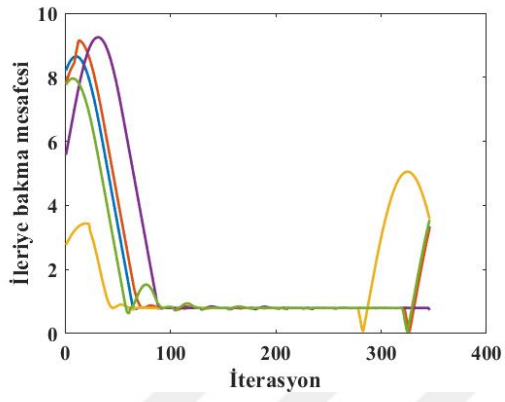
Şekil 4.64: PurePursuit- PRM algoritması arena (2), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan simülasyon sonuçları



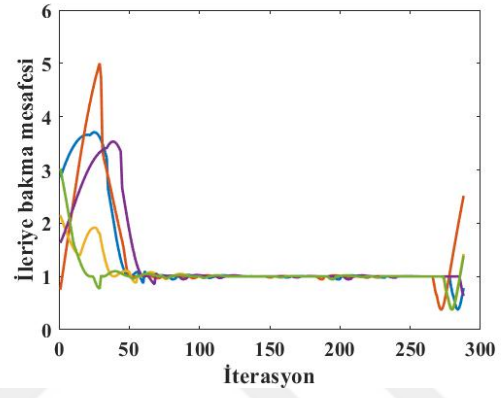
(a)



(b)

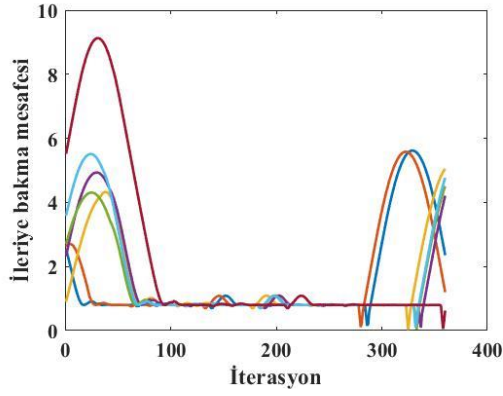


(c)

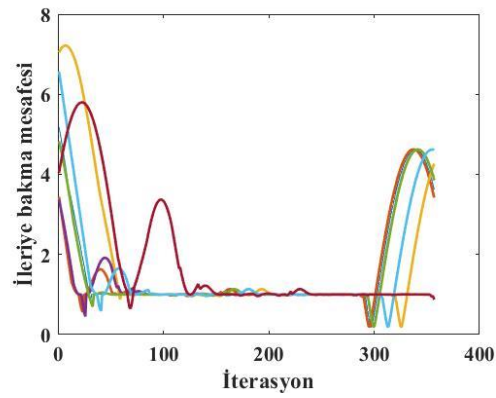


(d)

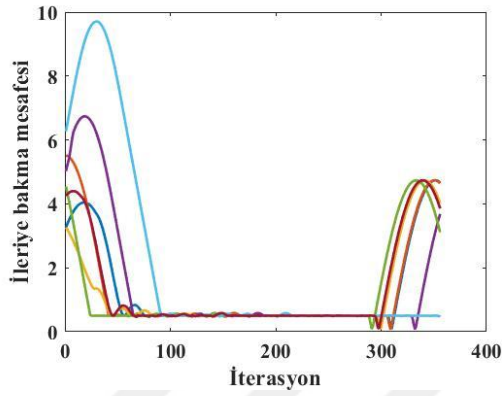
Şekil 4.65:Pure Pursuit- PRM algoritması arena (2), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan simülasyon sonuçları



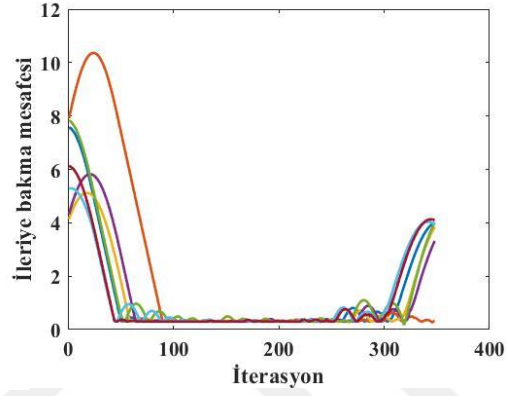
(a)



(b)



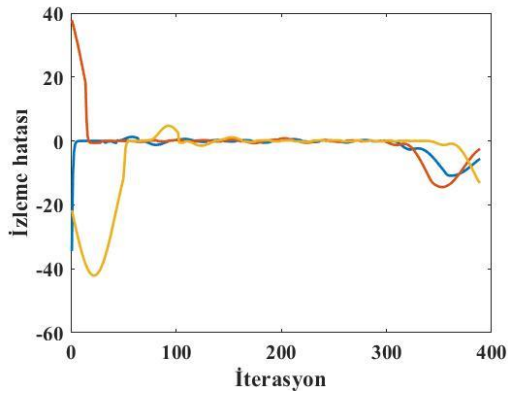
(c)



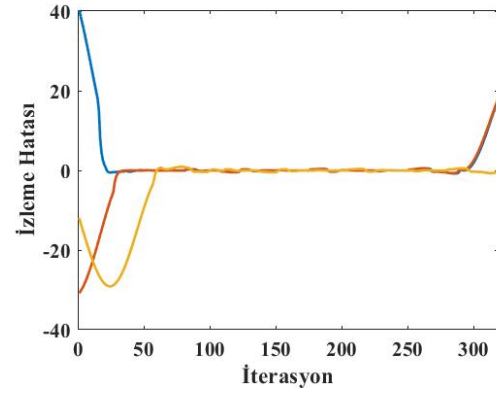
(d)

Şekil 4.66: Pure Pursuit- PRM algoritması arena (2), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan simülasyon sonuçları

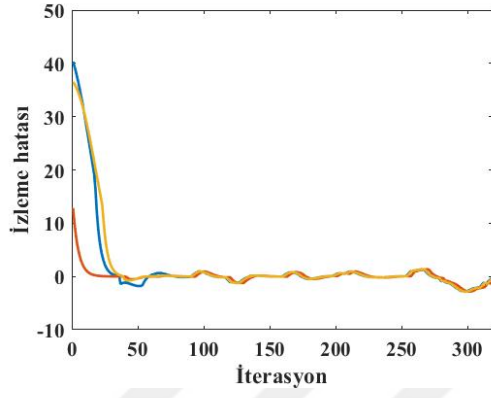
İzleme hatası, sürü robotunun planlanan yoldan geçerken bir engel ile karşılaştıklarında izleme hatası olarak gözlemlenir. Sürü robotlarının bir engelle veya keskin bir dönüş ile karşılaştıklarında takip edecekleri yola olan mesafesi artar. Dönen ve engellerden kaçan her sürü robotu izleyecekleri yola daha da yaklaşmak için organize bir şekilde hareket etmektedir. İleriye bakma mesafesine ve robot sayısına göre izleme hataları şekil 4.67,4.68 ve 4.69gösterilen grafiklerde gösterilmektedir.



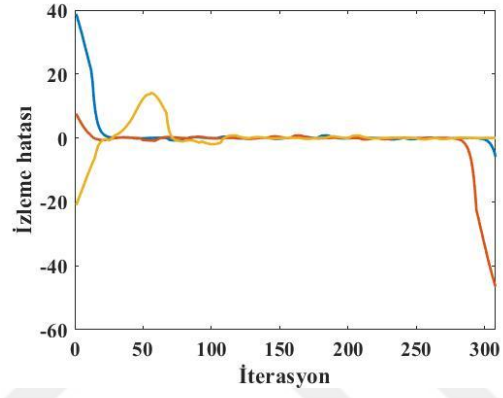
(a)



(b)

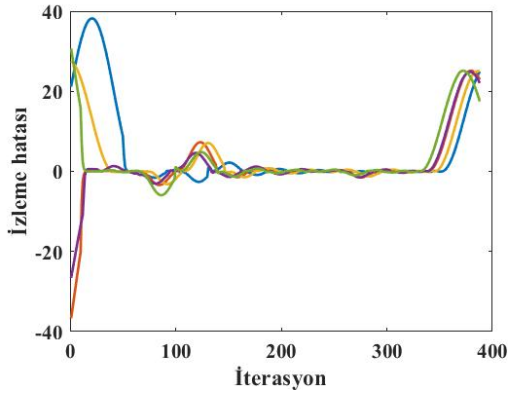


(c)

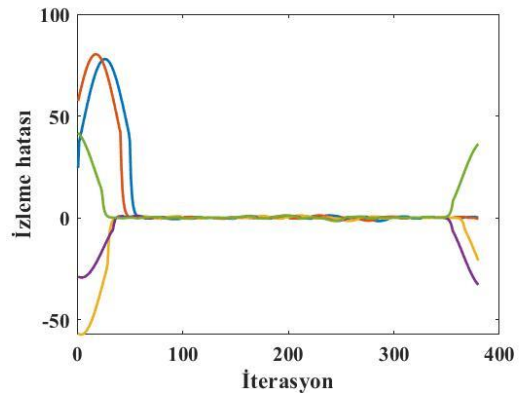


(d)

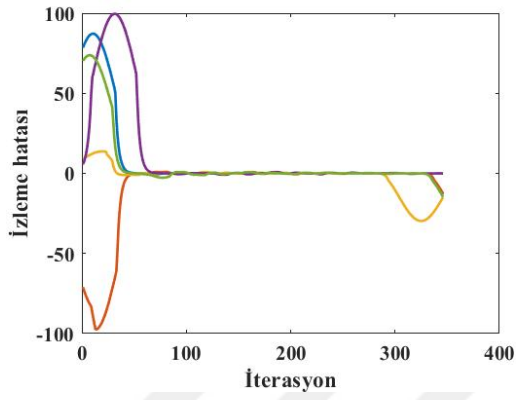
Şekil 4.67: PurePursuit -PRM algoritması arena (2), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan izleme hatası



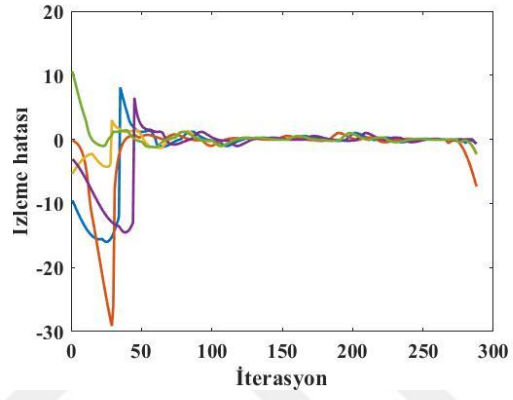
(a)



(b)



(c)



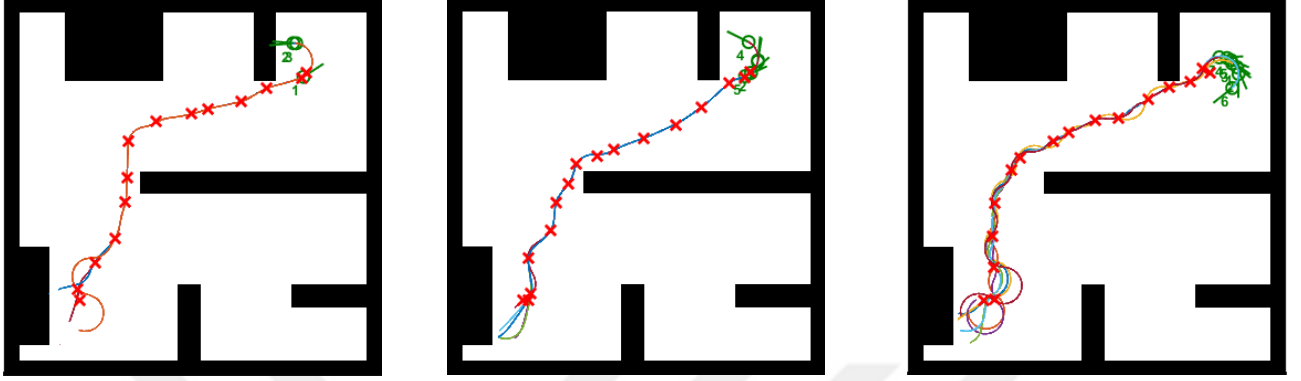
(d)

Şekil 4.68: Pure Pursuit -PRM algoritması arena (2), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan izleme hatası

Şekil 4.69: PurePursuit -PRM algoritması arena (2), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan izleme hatası

İzleme hataları incelendiğinde PRM (Olasılıklı Yol Haritası) algoritması ile arenalarda oluşturulan engelsiz yollarda robotların izleme hataları grafiklerden görüleceği üzere genellikle başlangıç ve hedef konumlarda izleme hataları gözlemlenir. Sürü robotları

oluřturulan engelsiz yol boyunca organize bir řekilde hedef konuma doęru hareket etmektedirler.

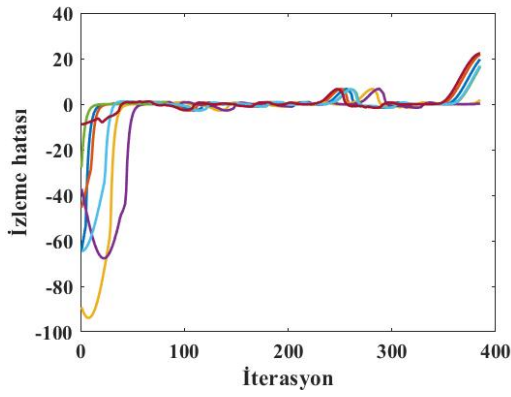


(a) (b)

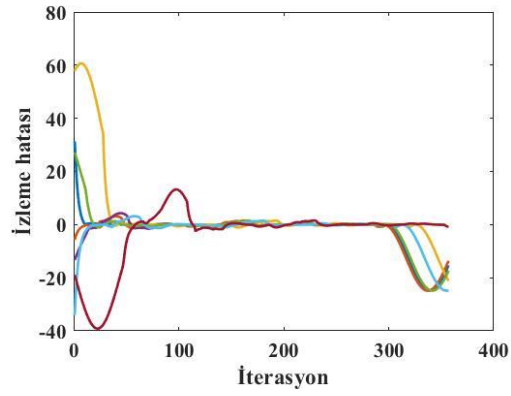
(c)

řekil 4.70: PurePursuit- PRM algoritması iin arena (3)'te simlasyon deney sonuları (a) 3 sr robotu, (b) 5 sr robotu ve (c) 7 sr robotu

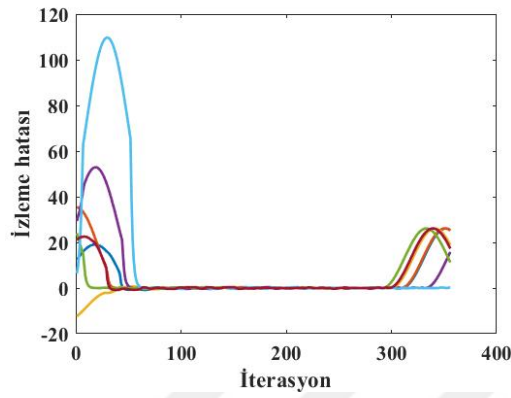
řekil 4.70 (a)'da grlen arena (3) oluřturulan yol ve simlasyon sonuları řekil 4.70 sırasıyla 3,5 ve 7 sr robotları iin



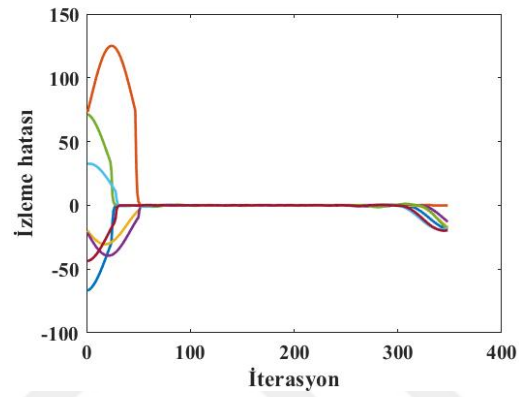
(a)



(b)

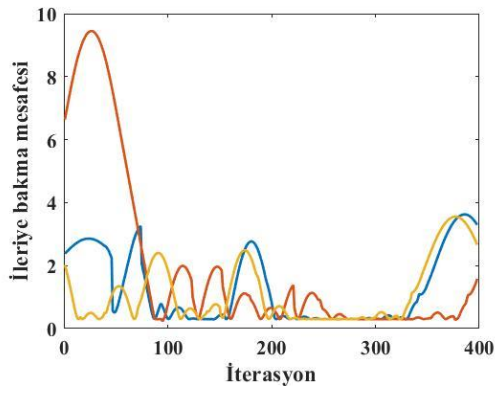


(c)

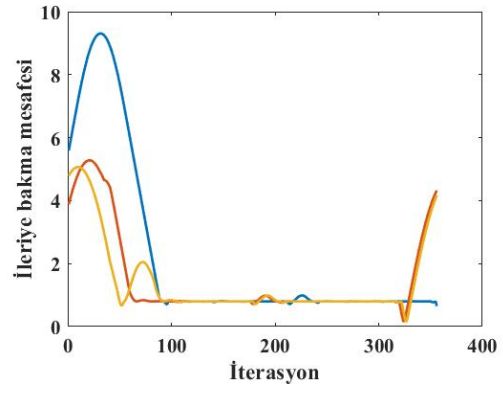


(d)

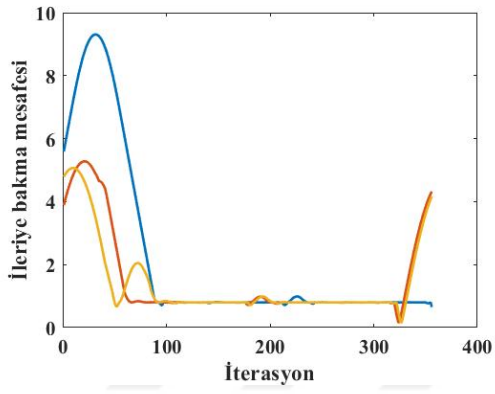
gösterilmektedir. Arena (3) robot yarıçapı kadar şişirilir. Arena (1), (2) ve (3)'deki gibi arena üzerinde rastgele oluşturulan 300 adet noktalar üzerinde engelsiz alanlar üzerindeki noktalar arasında bağlantılar ile oluşturulan yollar gözlemlenmektedir. Sürü robotları başlangıç ve hedef konum arasında oluşturulan yolu, PurePursuit algoritması ileriye bakma mesafe (l_d) değeri sırası ile 0.3, 0.5, 0.8 ve 1 değiştirilerek 36 sistematik deney yapılmış ve sürü robotlarının ölçeklenebilirliği ve esnekliği yol takip etme davranışı ile incelenecektir.



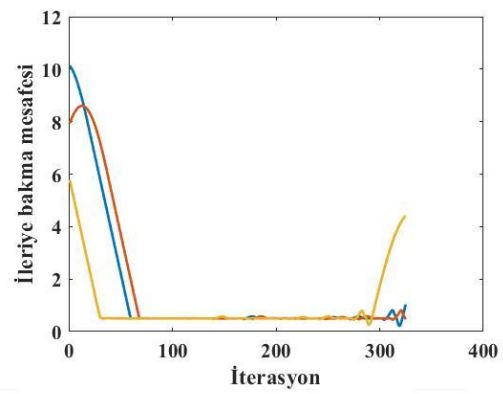
(a)



(b)

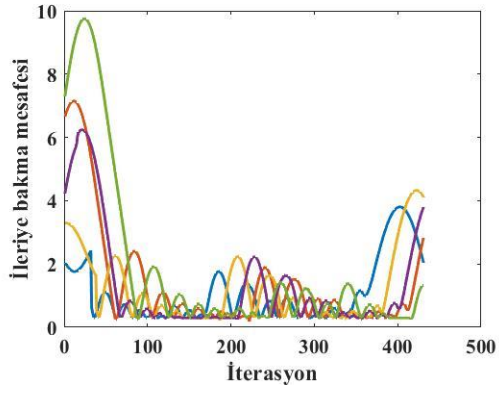


(c)

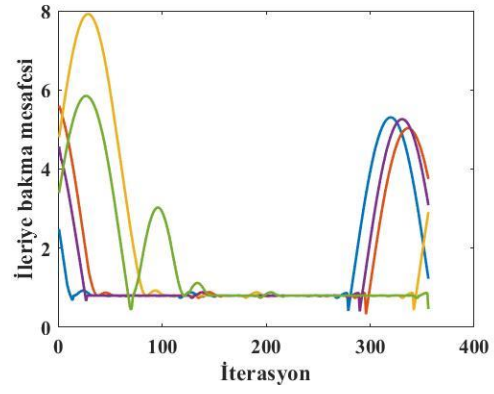


(d)

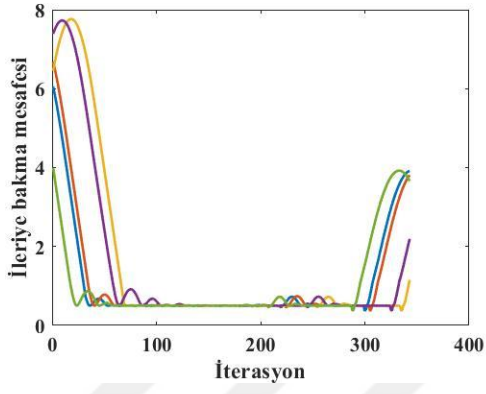
Şekil 4.71: Pure Pursuit- PRM algoritması arena (3), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan simülasyon sonuçları



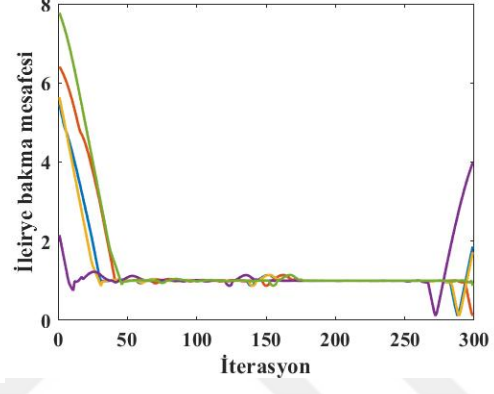
(a)



(b)

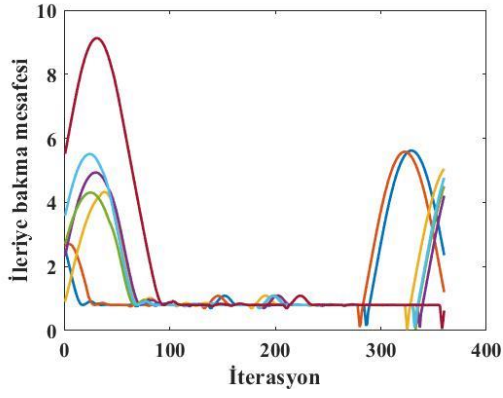


(c)

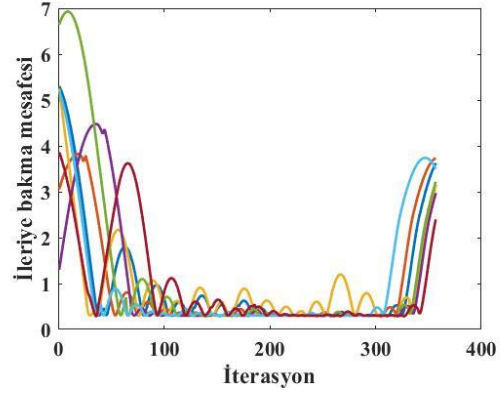


(d)

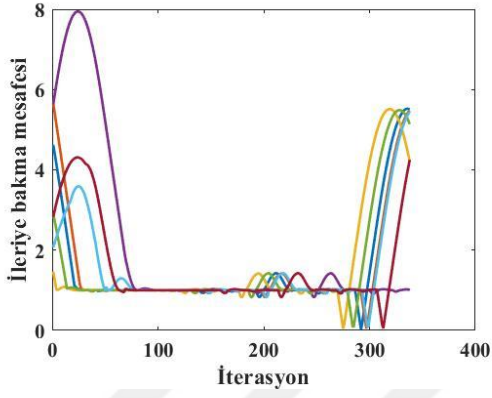
Şekil 4.72: PurePursuit- PRM algoritması arena (3), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan simülasyon sonuçları



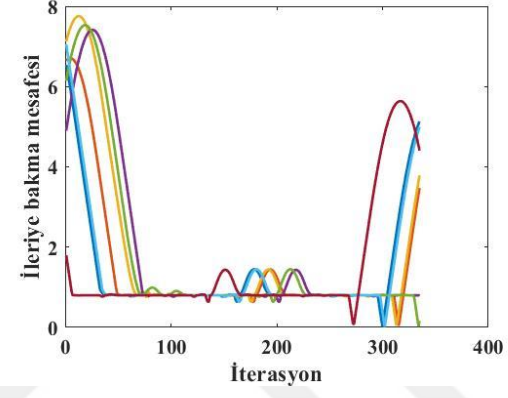
(a)



(b)



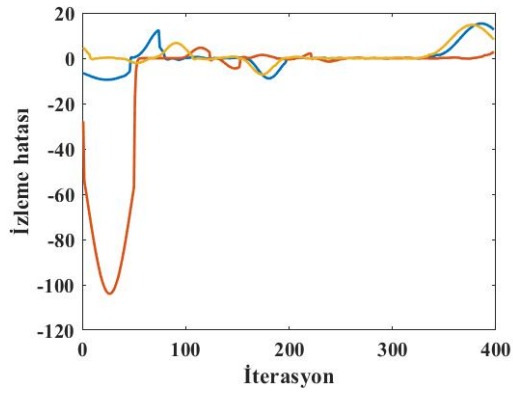
(c)



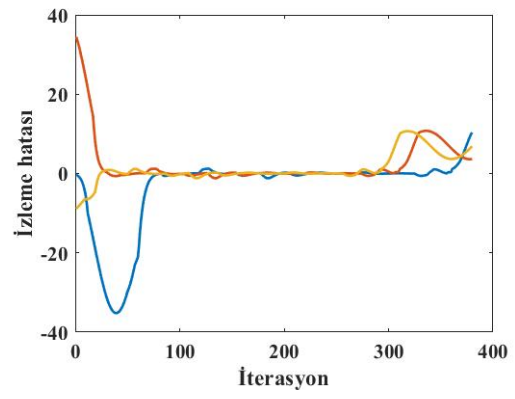
(d)

Şekil 4.73: PurePursuit- PRM algoritması arena (3), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan simülasyon sonuçları

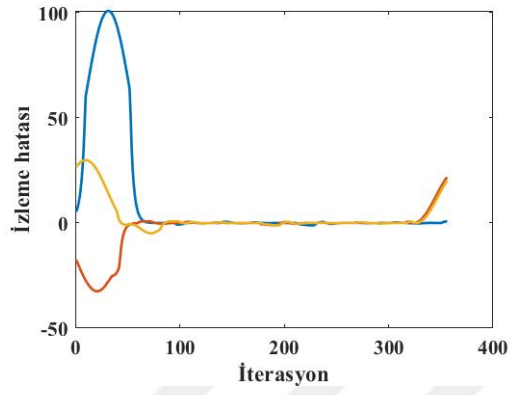
Geliştirilen Pursuit- PRM iş birliğin dayalı algoritma arena(3) için izleme hataları şekil 4.74,4.75 ve 4.76’ de grafiklerde gösterilmiş ve incelenmiştir.



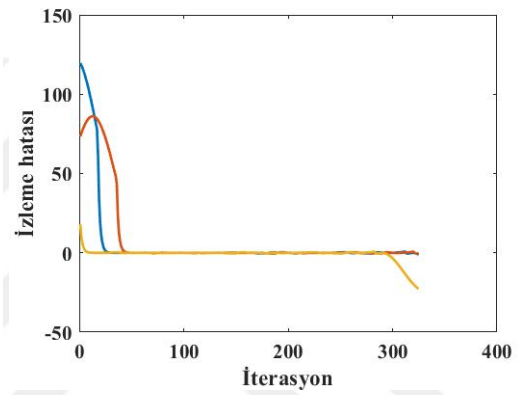
(a)



(b)

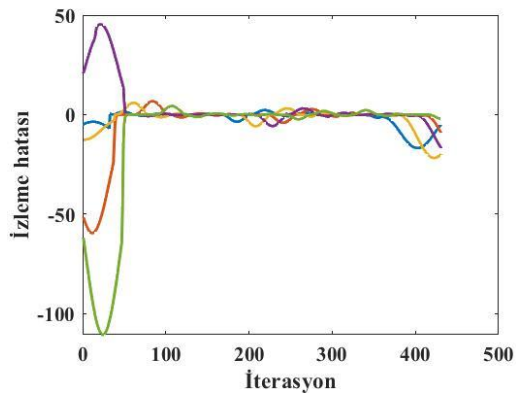


(c)

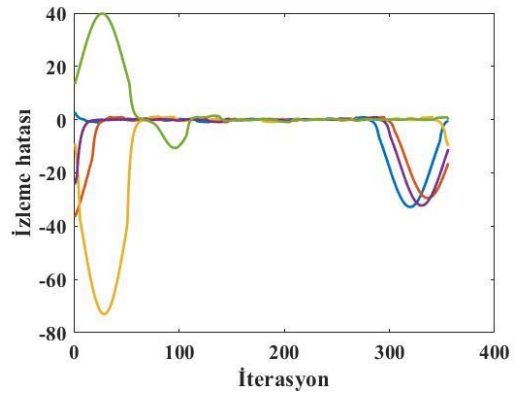


(d)

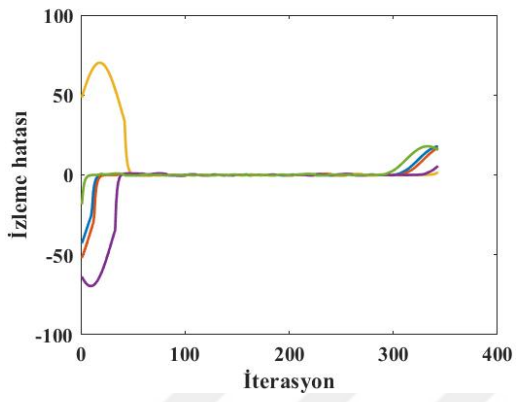
Şekil 4.74: PurePursuit -PRM algoritması arena (3), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 3 sürü robotu ile yapılan izleme hatası



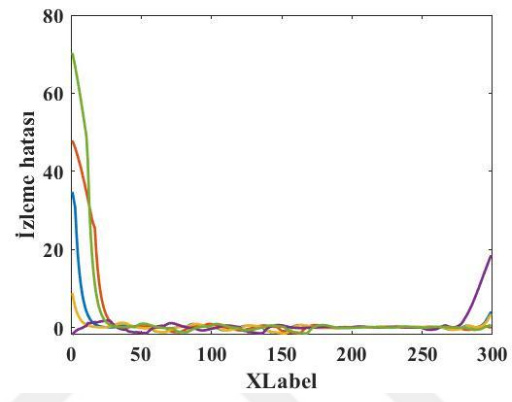
(a)



(b)

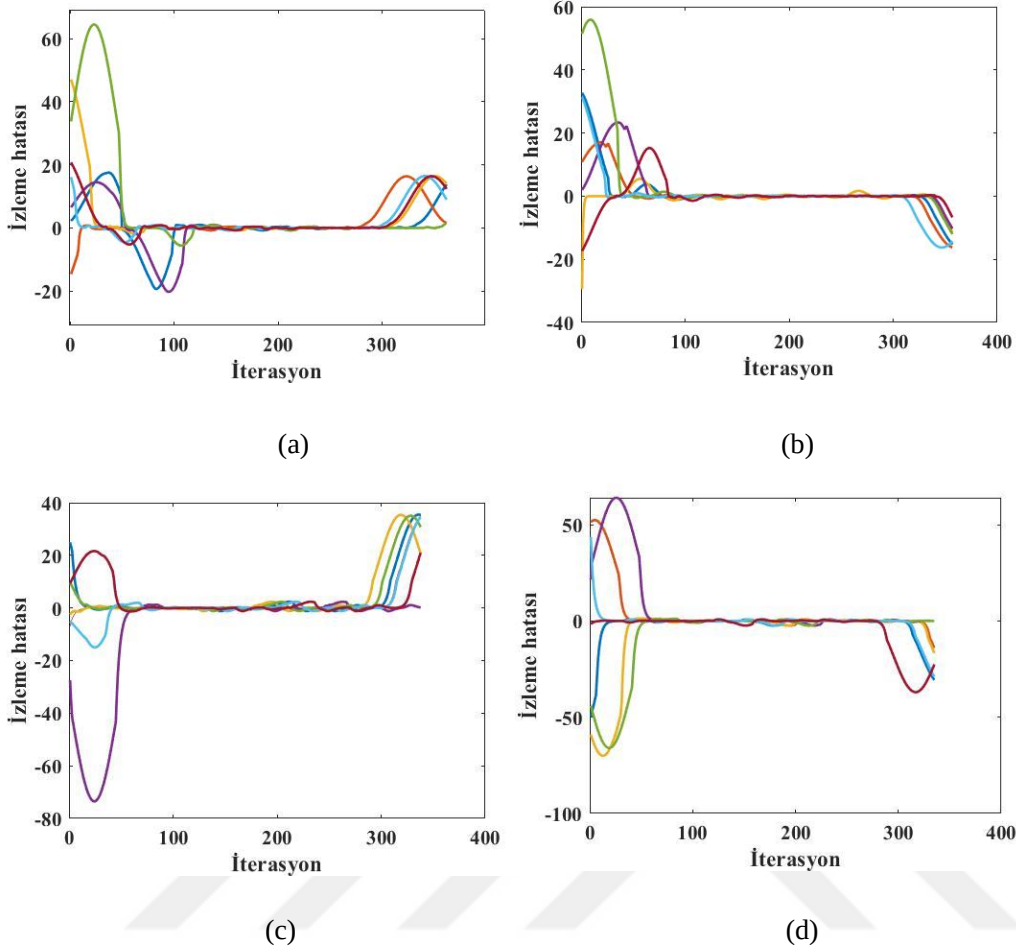


(c)



(d)

Şekil 4.75: PurePursuit -PRM algoritması arena (3), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 5 sürü robotu ile yapılan izleme hatası

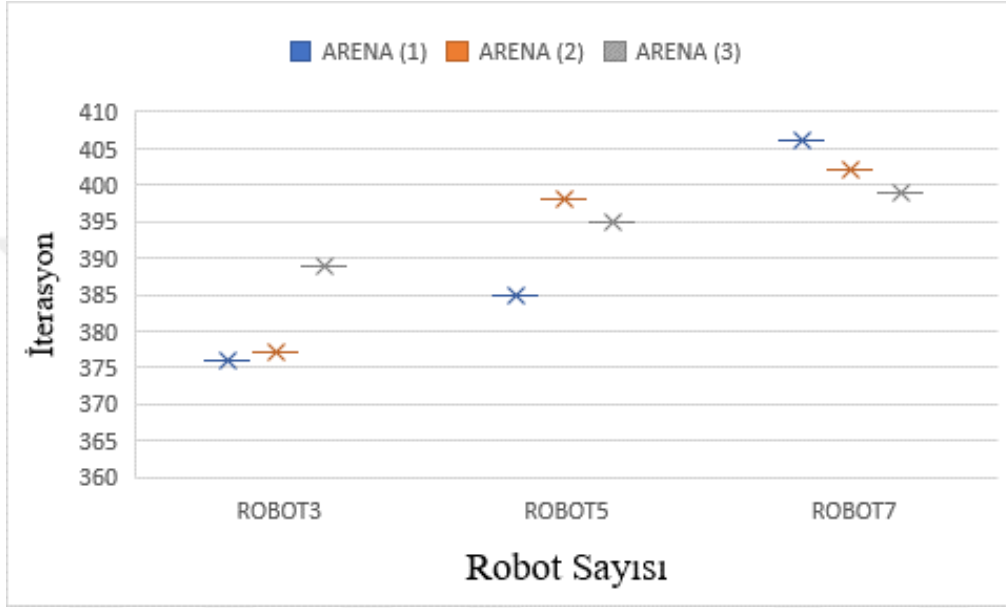


Şekil 4.76: PurePursuit -PRM algoritması arena (3), (a)0.3, (b)0.5, (c) 0.8, (d)1 ileriye bakma mesafesi için 7 sürü robotu ile yapılan izleme hatası

İzleme hatası VFH, RRT algoritmalarında açıklandığı gibi sürü robotlarının izleme yoluna dik mesafesidir. Yani sürü robotları bir engelle veya engele yakınlığında keskin dönüş anında izleme hataları gözlemlenir. Sürü robotları engelle karşılaştıklarında takip edilecek yolun mesafesi artar çünkü birinci önceli engellere kaçmamaktır. İzleme hatası grafiklerinde incelendiğinde 0-100 iterasyon aralığında izleme hataları gözlemlenmektedir. 0-100 iterasyon aralığı arena içerisindeki başlangıç konumdan planlanan yola göre hareket planlamasının gerçekleştiği andır.

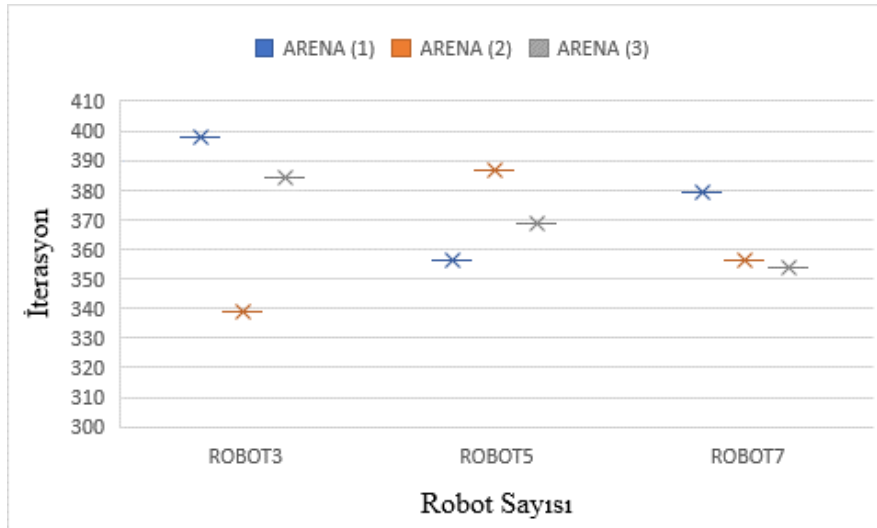
Sürü robotları için PurePursuit- PRM iş birliğine dayalı engelden kaçınma ve yol planlama algoritması sonuçları için kutu grafiği şeklinde incelenmiştir. Kutu grafiklerinde orta düz çizgi

ile belirtilen değer yineleme sayılarının ortalanca değeridir. 0.3 birim l_d değeri için arena (1), arena (2) ve arena (3) ile uygulanan sistematik deneylerde 3, 5, 7 sürü robotları için deney sonuçları, 385 iterasyon adımıyla en düşük ortalanca değer arena (1)'de 5 sürü robotu ile, en yüksek ortalanca değer ise 395 iterasyon adımı olan arena (3)'de 5 sürü robotu ile gerçekleşmiştir.



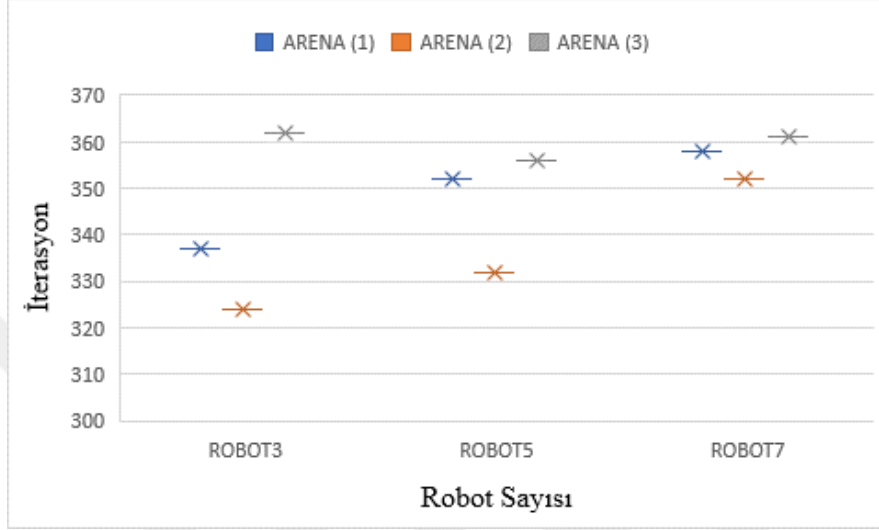
Şekil 4.77: 0.3 birim değer için PRM algoritması için uygulanan sistematik test sonuçları

Şekil 4.78'de geliştirilen PurePursuit – PRM algoritması için ileriye bakma mesafesi (l_d) 0.5 değerli sistematik deney sonuçlarının kutu grafiği yer almaktadır. En düşük ortalanca değer 356 iterasyon adımıyla arena (1) 5 robot için, en yüksek ortalanca değer ise 369 iterasyon adımıyla arena (3)'de 5 robot için gerçekleşmiştir.



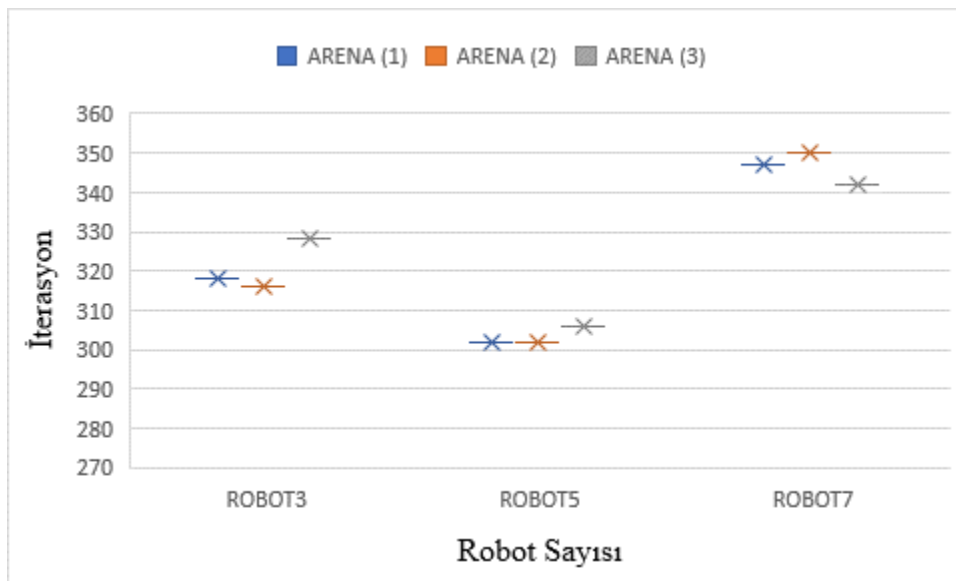
Şekil 4.78: 0.5 birim değer için PRM algoritması için uygulanan sistematik test sonuçları

Şekil 4.79’de geliştirilen PurePursuit – PRM algoritması için ileriye bakma mesafesi (l_d) 0.8 değerli sistematik deney sonuçlarının kutu grafiği yer almaktadır. En düşük ortalama değer 332 iterasyon adımı arena (2) 5 robot için, en yüksek ortalama değer ise 356 iterasyon adımı arena (3)’de 5 robot için gerçekleşmiştir.



Şekil 4.79: 0.8 birim değer için PRM algoritması için uygulanan sistematik test sonuçları

Şekil 4.80’de geliştirilen PurePursuit – PRM algoritması için ileriye bakma mesafesi (l_d) 1 değerli sistematik deney sonuçlarının kutu grafiği yer almaktadır. En düşük ortalama değer 319 iterasyon adımı arena (2) 5 robot için, en yüksek ortalama değer ise 338 iterasyon adımı arena (3)’de 5 robot için gerçekleşmiştir.



Şekil 4.80: 1 birim değer için PRM algoritması için uygulanan sistematik test sonuçları

Geliştirilen PurePursuit- PRM algoritmalarında robotların başlangıç konumdan hedef konuma varması için katettikleri yol için toplam uzunluk hesaplanmaktadır. Gerçekleştirilen 36 sistematik deney için ortalama katedilen yol hesaplanmıştır. Bu deneyler her robot için 12 sistematik deney şeklinde gerçekleşmiştir. Çizelge 6’da PurePursuit-PRM algoritma için sürü robotlarının katettikleri ortalama mesafeler gösterilmektedir. PRM algoritması ile geliştirilen, planlanan yol için sürü robotların ortalama en uzun katettikleri mesafe arena (1) de gerçekleşmiştir. En kısa mesafesi ise arena (3) gerçekleşmektedir.

PurePursuit- PRM İş birliğine Dayalı Algoritma için Katedilen Mesafe	
ARENA (1)	57 cm
ARENA (2)	54 cm
ARENA (3)	56 cm

Çizelge 6: PurePursuit – PRM algoritma için sürü robotlarının katettikleri ortalama mesafe

4. SONUÇ

Bu tez çalışmasında sürü robotlarının yol planlama ve engelden kaçınma davranışı için 3 farklı algoritma deneyler gerçekleştirilmiştir. Geliştirilen algoritmalarda yol planlama ve engelden kaçınma davranışındaki performansları incelenmiştir. Her bir sürü robotu, yol takibi sırasında ayrı ayrı algıladıkları komşu robotların davranışlarına göre hareket eder. Önerilen yol izleme yaklaşımını ölçeklenebilirlik ve esneklik açısından değerlendirmek için, 3 farklı arenada farklı sayıda robotlarla ve farklı bakış mesafelerine bağlı olarak sistematik deneyler yapılmıştır. Geliştirilen algoritmalarda, her bir sürü robotu, 3 farklı arenalarda izlenecek yola bakış mesafesine bağlı olarak organize bir şekilde sürü davranışı sergilemektedir. Sistematik deneylerden elde edilen sonuçlar, sürü robotlarının hedefe ulaşmak için esnek ve ölçeklenebilir bir şekilde yol takip ettiğini göstermektedir.

Yol planlama yöntemleri için geliştirilen algoritma ile sürü robotlarının esnek ve ölçeklenebilir olarak bu davranışı sergilemeleri amaçlanmıştır. Algoritmanın performansını incelemek için sürü robotlarına farklı arenalarda, farklı sayıda robotlarla ve farklı uzunluktaki ileriye bakma mesafelerinde sistematik simülasyon deneyleri uygulanmıştır. Uygulanan sistematik deneylerde sürü robotlarının sayısı değişse de engelden kaçınma ve hedef konuma ulaşma davranışı sergilenmektedir. Robot sayısının artmasına bağlı olarak sürü robotlarının başlangıç konumdan hedef konuma varma süreleri farklılıklar göstermiştir. Uygulanan sistematik deneylerde 3 robot için PurePursuit -PRM algoritması yapılan deneylerde en az sürede (iterasyon sayısı) hedef konuma varmışlardır. En uzun sürede (iterasyon sayısı) 7 robot ile PurePursuit- VFH algoritmasında gerçekleştirilmiştir. Sırasıyla 3, 5 ve 7 sürü robotları için her arena ve her algoritma için elde edilen deney sonuçlarında göre başlangıç konumdan hedef konuma ulaşma süresi uzamıştır.

Sistematik deney alanları 50x50 boyutlarında 3 farklı arenada, her arena için engel konumları bağlı olarak başlangıç konumdan hedef konuma varma süreleri ve sürü robotlarının katettikleri mesafeler farklılık göstermektedir. Uygulanan sistematik deneylerde 50x50 boyutlarındaki farklı arenadaki engel konumlarına bağlı olarak değişmektedir. Her arena için robotların rastgele başlangıç konum aralığı ve belirlediğimiz başlangıç ve hedefleri konumlarımız aynıdır. Elde edilen deney sonuçlarına göre arena(1)'de en uzun simülasyon süresi ve en uzun katedilen mesafe yer almaktadır. Arena(2)'de ise en kısa simülasyon süresi ve katedilen mesafesi gerçekleşmektedir.

Elde edilen sonuçları ileriye bakma mesafesine için gözlemlendiğinde hedef konuma ulaşma süresi değişmektedir. İleriye bakma mesafesi artıkça sürü robotlarının izleyeceği yola olan mesafesi de artar. Bu nedenle sürü robotlarının arenalar içerisinde bulunan engellere çarpışma ve keskin dönüş olasılıkları azalır. VFH algoritması için arena içerisindeki engelden kaçınarak ve hedef konuma ilerleyerek ileriye bakma mesafesini artırttıkça tamamlanma süreside azalır. PRM-RRT algoritmalarında ise sürü robotları harekete başlamadan, başlangıç ve hedef konum arasında engelsiz bir yol oluşturur ve oluşturulan yolu ileriye bakma mesafesine göre takip ederek hedefe ulaşır. Deney sonuçlarına göre, değer arttıkça sürü robotlarının yol takibini tamamlama süreleri azalmaktadır.

Uygulanan sistematik deney sonuçlarında en kısa hedefe ulaşma süresi PurePursuit- PRM algoritması l_d değeri 1 ile arena(2)'de 3 sürü robotu ile gerçekleştirilen deney sonucunda 288 iterasyon adımı ile görülmektedir. Sürü robotlarının katettikleri en kısa mesafe ise PRM algoritması ile arena (2) 53 cm olarak gözlemlenmektedir. En uzun simülasyon süresi ise PurePursuit- VFH algoritması l_d değeri 0.3 ile arena (1)'de 7 sürü robotu 1148 iterasyon adımı ile gerçekleşmiştir. En uzun kat edilen mesafe ise VFH algoritması ile gözlemlenmiştir. Her bir sürü robotuna özerk olarak engelden kaçınma ve yol takibi ile hedefe ulaşma davranışı uygulamıştır.

İzleme hataları gözlemlendiğinde VFH algoritması arena içerisinde sadece başlangıç ve hedef konum belli olduğu için yani hedefe ulaşmak için bir yol veya herhangi bir ara nokta belirlenmediği için sürü robotları keskin dönüşler ve engellere rastlar ve izleme hataları gözlemlenir. PRM ve RRT algoritmaları için ise arenalar üzerinde engelsiz çarpışmasız başlangıç ve hedef konum arasında bir yol oluşturur. Bu neden sürü robotları herhangi bir engelden kaçınmaz böylece izleme hataları az olarak gözlemlenmektedir.

En kısa yol ölçütüne göre ise, RRT ve PRM algoritmaları karşılaştırılacaktır. RRT algoritması belirli bir durum uzayından rastgele alınan örneklerden artımlı olarak bir arama ağacı oluşturan ağaç tabanlı bir hareket planlayıcıdır. Bu hareket planlayıcısı ile başlangıç ve hedef konum arasında engelsiz bir yol oluşturulur. PRM algoritması ise, Dijkstra'nın en kısa yol sorgusuyla başlangıç hedef konum arasında bir engelsiz çarpışmasız bir yol oluşturur. Her arena için en kısa yol uzunluğu PRM algoritması ile gözlemlenmektedir.

Bu tez çalışmasında, incelenen bütün yöntemler, sürü robotiği uygulamalarının yanı sıra, iş birliğine dayalı özerk robot yol planlama uygulamalarında da kullanılmaktadır. Sürü homojen çoklu robotlar üzerinde çeşitli ortamlar ve görevler için kullanılmaktadır.

5. KAYNAKÇA

- Ahuja, R. K., Mehlhorn, K., Orlin, J., & Tarjan, R. E. (1990). Faster algorithms for the shortest path problem. *Journal of the ACM*, 37(2), 213-223. <https://doi.org/10.1145/77600.77615>
- Alarabi, S., Luo, C., & Santora, M. (2022). A PRM Approach to Path Planning with Obstacle Avoidance of an Autonomous Robot. 2022 8th International Conference on Automation, Robotics and Applications (ICARA), 76-80. <https://doi.org/10.1109/ICARA55094.2022.9738559>
- Anoop, A. S., & Kanakasabapathy, P. (2017). Review on swarm robotics platforms. 2017 International Conference on Technological Advancements in Power and Energy (TAP Energy), 1-6. <https://doi.org/10.1109/TAPENERGY.2017.8397275>
- Arvin, F., Murray, J., Zhang, C., & Yue, S. (2014). Colias: An Autonomous Micro Robot for Swarm Robotic Applications. *International Journal of Advanced Robotic Systems*, 11, 1. <https://doi.org/10.5772/58730>
- Bi, Z., Yimin, Y., & Wei, Y. (2008). Hierarchical path planning approach for mobile robot navigation under the dynamic environment. 2008 6th IEEE International Conference on Industrial Informatics, 372-376. <https://doi.org/10.1109/INDIN.2008.4618127>
- Bush, C. (2017, Eylül 20). Mona: An Affordable Open-source Mobile Robot for Research and Education (Workshop) [Text]. <https://www.edinburgh-robotics.org/events/mona-affordable-open-source-mobile-robot-research-and-education-workshop>
- Calderón-Arce, C., Brenes-Torres, J. C., & Solis-Ortega, R. (2022). Swarm Robotics: Simulators, Platforms and Applications Review. *Computation*, 10(6), Article 6. <https://doi.org/10.3390/computation10060080>
- Campo, A., & Dorigo, M. (2007). Efficient Multi-foraging in Swarm Robotics. İçinde F. Almeida e Costa, L. M. Rocha, E. Costa, I. Harvey, & A. Coutinho (Ed.), *Advances in Artificial Life* (ss. 696-705). Springer. https://doi.org/10.1007/978-3-540-74913-4_70
- Coppola, M., Guo, J., Gill, E., & de Croon, G. C. H. E. (2019). Provable self-organizing pattern formation by a swarm of robots with limited knowledge. *Swarm Intelligence*, 13(1), 59-94. <https://doi.org/10.1007/s11721-019-00163-0>
- Çayırpunar, Ö. (t.y.). ÇOKLU ROBOT SİSTEMLERİNDE ROBOTLAR ARASI HABERLEŞME VE İŞ BİRLİĞİ KULLANILARAK ARAMA VERİMLİLİĞİNİN ARTIRILMASI.

- Dr. Lydia E. Kavraki: A Woman Making Robots Work. (2013, Ekim 14). Mental Floss.
<https://www.mentalfloss.com/article/53176/dr-lydia-e-kavraki-woman-making-robots-work>
- Gang, L., & Wang, J. (2015). PRM path planning optimization algorithm research. 10, 148-153.
- Garip, Z. (2018). Mobil robotların yol planması için metasezgisel hibrit algoritmalar geliştirilmesi ve uygulanması [DoctoralThesis, Sakarya Üniversitesi].
<https://acikerisim.sakarya.edu.tr/handle/20.500.12619/74245>
- Horvath, E., Hajdu, C., & Koros, P. (2019). Enhancement of pure-pursuit path-tracking algorithm with multi-goal selection. 2019 1st IEEE International Conference on Gridding and Polytope Based Modeling and Control (GPMC), 13-18.
<https://doi.org/10.1109/GPMC48183.2019.9106958>
- Hosseini, S. (2016). Path Tracking Methodologies For Mobile Robots [Fen Bilimleri Enstitüsü]. <http://hdl.handle.net/11527/15511>
- Huang, Y., Tian, Z., Jiang, Q., & Xu, J. (2020). Path Tracking Based on Improved Pure Pursuit Model and PID. 2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT), 359-364.
<https://doi.org/10.1109/ICCASIT50869.2020.9368694>
- Jie, D., Xueming, M., & Kaixiang, P. (2010). IVFH^{∗}: Real-time dynamic obstacle avoidance for mobile robots. 2010 11th International Conference on Control Automation Robotics & Vision, 844-847. <https://doi.org/10.1109/ICARCV.2010.5707283>
- Lal, D. S., Vivek, A., & Selvaraj, G. (2017). Lateral control of an autonomous vehicle based on Pure Pursuit algorithm. 2017 International Conference on Technological Advancements in Power and Energy (TAP Energy), 1-8.
<https://doi.org/10.1109/TAPENERGY.2017.8397361>
- Levitis, D. A., Lidicker, W. Z., & Freund, G. (2009). Behavioural biologists do not agree on what constitutes behaviour. *Animal Behaviour*, 78(1), 103-110.
<https://doi.org/10.1016/j.anbehav.2009.03.018>
- Li, Y. (2021). An RRT-Based Path Planning Strategy in a Dynamic Environment. 2021 7th International Conference on Automation, Robotics and Applications (ICARA), 1-5.
<https://doi.org/10.1109/ICARA51699.2021.9376472>
- Lozano-Pérez, T., & Wesley, M. A. (1979). An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22(10), 560-570.
<https://doi.org/10.1145/359156.359164>

- Mısır, O. (2023). Dynamic local path planning method based on neutrosophic set theory for a mobile robot. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 45(3), 1-13. <https://doi.org/10.1007/s40430-023-04048-6>
- Mısır, O., Çelik, M., & Gökrem, L. (2022). Waypoint-Based Path Tracking Approach For Self-Organized Swarm Robots. *Uluslararası Muhendislik Arastirma ve Gelistirme Dergisi*, 14, 799-815. <https://doi.org/10.29137/umagd.1118039>
- Mısır, O., Gökrem, L., & Serhat Can, M. (2020). Fuzzy-based self organizing aggregation method for swarm robots. *Biosystems*, 196, 104187. <https://doi.org/10.1016/j.biosystems.2020.104187>
- Mısır, O., & Gökrem, L. (2020). Sürü Robotları için Esnek ve Ölçeklenebilir Toplanma Davranışı Metodu. *European Journal of Science and Technology*, 100-109. <https://doi.org/10.31590/ej>
sat.779162
- Mısır, O., & Gökrem, L. (2021a). Dynamic interactive self organizing aggregation method in swarm robots. *Biosystems*, 207, 104451. <https://doi.org/10.1016/j.biosystems.2021.104451>
- Mısır, O., & Gökrem, L. (2021b). Flocking-Based Self-Organized Aggregation Behavior Method for Swarm Robotics. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, 45(4), 1427-1444. <https://doi.org/10.1007/s40998-021-00442-9>
- Nemec, D., Janota, A., Hruboš, M., Gregor, M., & Pirnik, R. (2017). Mutual acoustic identification in the swarm of e-puck robots. *International Journal of Advanced Robotic Systems*, 14, 172988141771079. <https://doi.org/10.1177/1729881417710794>
- Olaronke, I., Rhoda, I., Gambo, I., Oluwaseun, O., & Janet, O. (2020). A Systematic Review of Swarm Robots. *Current Journal of Applied Science and Technology*, 79-97. <https://doi.org/10.9734/cjast/2020/v39i1530719>
- Panerati, J., Gianoli, L., Pincioli, C., Shabah, A., Nicolescu, G., & Beltrame, G. (2018). From Swarms to Stars: Task Coverage in Robot Swarms with Connectivity Constraints. 2018 IEEE International Conference on Robotics and Automation (ICRA), 7674-7681. <https://doi.org/10.1109/ICRA.2018.8463193>
- Park, J.-H., & Yoon, T.-W. (2018). Maximizing the Coverage of Roadmap Graph for Optimal Motion Planning. *Complexity*, 2018, e9104720. <https://doi.org/10.1155/2018/9104720>
- Qu, P., Xue, J., Ma, L., & Ma, C. (2015). A constrained VFH algorithm for motion planning of autonomous vehicles. 2015 IEEE Intelligent Vehicles Symposium (IV), 700-705. <https://doi.org/10.1109/IVS.2015.7225766>