YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ROBOT TAKIMI İLE EŞ ZAMANLI KONUM BELİRLEME VE HARİTALAMA

Bilgisayar Mühendisi Ozan ÖZIŞIK

FBE Bilgisayar Mühendisliği Anabilim Dalında Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Yrd. Doç. Dr. Sırma YAVUZ

İSTANBUL, 2010

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ iv

KISALTMA LİSTESİ v

ŞEKİL LİSTESİ vi

ÇİZELGE LİSTESİ xii

ÖNSÖZ xiii

ÖZET xiv

ABSTRACT xv

1. GİRİŞ 1

1.1 Tezin Amacı ve Kapsamı 1
1.2 Robotikte Belirsizlik 2
1.3 Harita Temsili 3
1.4 Mobil Robotlarda Konum Belirleme 5
1.4.1 Olasılıksal Konum Belirleme Metotları 6
1.5 Eş Zamanlı Konum Belirleme ve Haritalama 7
1.5.1 Eş Zamanlı Konum Belirleme ve Haritalama Metotları 8
1.6 Doğru Ayrıştırma ve Çıkarım Algoritmaları 8
1.6.1 Ardışıl Kenar Takibi (*Successive Edge Following*) 9
1.6.2 Doğru İzleme (*Line Tracking*) 9
1.6.3 İteratif Uç Nokta Uydurma (*Iterative End Point Fit*) 10
1.6.4 Ayırma-ve-Birleştirme (*Split-and-Merge*) 10
1.6.5 Prototip Tabanlı Bulanık Kümeleme Algoritması (*Prototype-based Fuzzy Clustering Algorithm*) 10
1.6.6 Bulanık Ayırma-ve-Birleştirme (*Split-and-Merge Fuzzy*) 11
1.6.7 Doğru Bağlanımı (*Line Regression*) 12
1.6.8 Rastgele Örnek Uzlaşımı (*Random Sample Consensus – RANSAC*) 12
1.6.9 Hough Dönüşümü (*Hough Transform*) 12
1.6.10 Basit Doğrusal Bağlanım (*Simple Linear Regression*) 14
1.7 Önceki Çalışmalar 15

2. ÖNERİLEN YÖNTEM 21

2.1 Kullanılan Robot Modeli 21
2.2 Robot Takımı Kullanımı ve Avantajları 22
2.3 Keşif Stratejisi 22
2.4 Özellik çıkarımı 23
2.4.1 İki Noktadan Doğru Çıkarımı 25
2.4.2 Hough Dönüşümü ile Doğru Çıkarımı 27

2.4.3 Basit Doğrusal Bağlanım ile Doğru Çıkarımı 27
2.5 Genişletilmiş Kalman Filtresi 29
2.6 Döngü Kapatma 30
2.7 Webots Simülasyon Ortamı, Uygulama ve Yapılan Deneylerin Detayları 33
3. TEK ROBOTLA İKİ NOKTADAN DOĞRU ÇIKARIMI YAPARAK HARİTALAMA 48
3.1 Deneysel Sonuçlar 48
3.2 Sonuçlar 53
4. TEK ROBOTLA HOUGH DÖNÜŞÜMÜ İLE DOĞRU ÇIKARIMI YAPARAK HARİTALAMA 54
4.1 Deneysel Sonuçlar 54
4.2 Sonuçlar 63
5. ROBOT TAKIMI İLE HARİTALAMA 64
5.1 İki Robotla Deneysel Sonuçlar 64
5.2 Dört Robotla Deneysel Sonuçlar 69
5.3 Sonuçlar 70
6. GENİŞLETİLMİŞ KALMAN FİLTRESİ VE HAREKET GÜRÜLTÜSÜ 71
7. SONUÇ 78
KAYNAKLAR 97
EKLER 100
Ek 1 Genişletilmiş Kalman Filtresi Genel Algoritması 101
Ek 2 Genişletilmiş Kalman Filtresi Algoritması Uygulaması 108
Ek 3 Kitapta Söz Edilen Terimlerin Türkçe ve İngilizce Adları 121
ÖZGEÇMİŞ 123

SİMGE LİSTESİ

x	Robot konumu
z	Algılayıcı ve özellik çıkarım modülü tarafından elde edilen gözlem
u	Kumanda modülünden gelen kontrol işareti
m	Harita
t	Zaman değişkeni

KISALTMA LİSTESİ

EZKBH	Eş Zamanlı Konum Belirleme ve Haritalama
GKF	Genişletilmiş Kalman Filtresi
KLD	Kullback-Leibler Distance
SLAM	Simultaneous Localization and Mapping

ŞEKİL LİSTESİ

Şekil 1.1 Özellik tabanlı harita örneği (Bu örnekte özellik olarak doğrular kullanılmıştır)..... 3
Şekil 1.2 Konum tabanlı harita örneği (Bu örnekte her hücrenin koyuluğu o hücrenin kaç defa dolu görüldüğüne bağlı olarak belirlenmiştir)..... 4
Şekil 1.3 Bir ofis ortamının topolojik harita ile gösterimi..... 5
Şekil 1.4 Kartezyen düzlemde bir doğrunun (r,θ) parametreleri ile gösterimi..... 13
Şekil 1.5 Kartezyen düzlemde aynı hizada bulunan noktalar ve bunlardan geçen doğru 13
Şekil 1.6 Şekil 1.5'te görülen noktaların Hough uzayındaki temsili..... 14
Şekil 2.1 Kullanılan robot modeli 21
Şekil 2.2 Hough uzayı parametreleri ile doğru temsili..... 24
Şekil 2.3 Duvar takibi yapan robotun konumundaki hata her adımda artacaktır. 24
Şekil 2.4 İç köşede aynı taraftaki algılayıcıların farklı duvarları görmesi durumu..... 26
Şekil 2.5 Dış köşede aynı taraftaki algılayıcıların farklı duvarları görmesi durumu 26
Şekil 2.6 Aynı doğru parametreleri ile ifade edildikleri için aynı duvar olarak ele alınabilecek farklı iki duvar..... 30
Şekil 2.7 Döngü kapatma olmaksızın haritalama..... 31
Şekil 2.8 Aynı duvara karşılık gelen iki yer işaretçisi..... 31
Şekil 2.9 Döngü kapatma işlemi sırasında ilişkili oldukları bulunan iki yer işaretçisi 32
Şekil 2.10 Döngü kapatma işlemi sonrası yer işaretçilerinin durumu..... 33
Şekil 2.11 Webots Simülasyon ortamı 34
Şekil 2.12 Çok sayıda iç ve dış köşeden oluşan Ortam 1'in görüntüsü..... 35
Şekil 2.13 Oda ve koridorlardan oluşan Ortam 2'nin görüntüsü..... 35
Şekil 2.14 Birbirleriyle 10, 20, 30 ve 45 derecelik açılar yapan duvarlar içeren Ortam 3'ün görüntüsü..... 36
Şekil 2.15 Ortam 1 için duvar takibi ile çıkarılması beklenen harita 37
Şekil 2.16 Ortam 1 için keşif stratejisi ile çıkarılması beklenen harita..... 37
Şekil 2.17 Ortam 2 için beklenen harita 38
Şekil 2.18 Ortam 3 için duvar takibi ile çıkarılması beklenen harita 38
Şekil 2.19 Ortam 3 için keşif stratejisi ile çıkarılması beklenen harita..... 39
Şekil 2.20 Ortam 1'de, standart sapması 2 cm olan algılayıcı gürültüsü varken duvar takibi ile elde edilen ham harita..... 40
Şekil 2.21 Ortam 2'de, standart sapması 2 cm olan algılayıcı gürültüsü varken duvar takibi ile elde edilen ham harita..... 40
Şekil 2.22 Ortam 3'te, standart sapması 2 cm olan algılayıcı gürültüsü varken duvar takibi ile

elde edilen ham harita 41
Şekil 2.23 Ortam 1'de, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi ile elde edilen ham harita 41
Şekil 2.24 Ortam 2'de, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi ile elde edilen ham harita 42
Şekil 2.25 Ortam 3'te, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi ile elde edilen ham harita 42
Şekil 2.26 Ortam 1'de, standart sapması 4 cm olan algılayıcı gürültüsü varken duvar takibi ile elde edilen ham harita 43
Şekil 2.27 Ortam 2'de, standart sapması 4 cm olan algılayıcı gürültüsü varken duvar takibi ile elde edilen ham harita 43
Şekil 2.28 Ortam 3'te, standart sapması 4 cm olan algılayıcı gürültüsü varken duvar takibi ile elde edilen ham harita 44
Şekil 2.29 Ortam 1'de, standart sapması 4 cm olan algılayıcı gürültüsü varken keşif stratejisi ile elde edilen ham harita 44
Şekil 2.30 Ortam 2'de, standart sapması 4 cm olan algılayıcı gürültüsü varken keşif stratejisi ile elde edilen ham harita 45
Şekil 2.31 Ortam 3'te, standart sapması 4 cm olan algılayıcı gürültüsü varken keşif stratejisi ile elde edilen ham harita 45
Şekil 2.32 Örnek harita 46
Şekil 2.33 Örnek işgal edilmiş alan haritası 47
Şekil 2.34 Örnek keşif haritası 47
Şekil 3.1 Ortam 1'de, gürültü yokken duvar takibi ile elde edilen harita 48
Şekil 3.2 Ortam 2'de, gürültü yokken duvar takibi ile elde edilen harita 49
Şekil 3.3 Ortam 1'de, standart sapması 2 cm olan algılayıcı gürültüsü varken duvar takibi ile çıkarılan harita 49
Şekil 3.4 Ortam 2'de, standart sapması 2 cm olan algılayıcı gürültüsü varken duvar takibi ile çıkarılan harita 50
Şekil 3.5 Ortam 1'de, standart sapması 4 cm olan algılayıcı gürültüsü varken duvar takibi ile çıkarılan harita 50
Şekil 3.6 Ortam 2'de, standart sapması 4 cm olan algılayıcı gürültüsü varken duvar takibi ile çıkarılan harita 51
Şekil 3.7 Ortam 1'de, gürültü yokken keşif stratejisi kullanıldığında çıkarılan harita 51
Şekil 3.8 Ortam 2'de, gürültü yokken keşif stratejisi kullanıldığında çıkarılan harita 52

Şekil 3.9 Ortam 1'de, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında çıkarılan harita 52
Şekil 3.10 Ortam 2'de, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında çıkarılan harita 53
Şekil 4.1 Ortam 1'de, gürültü yokken duvar takibi ile elde edilen harita 54
Şekil 4.2 Ortam 2'de, gürültü yokken duvar takibi ile elde edilen harita 55
Şekil 4.3 Ortam 3'te gürültü yokken duvar takibi ile elde edilen harita 55
Şekil 4.4 Ortam 1'de, standart sapması 2 cm olan algılayıcı gürültüsü varken duvar takibi ile çıkarılan harita 56
Şekil 4.5 Ortam 2'de, standart sapması 2 cm olan algılayıcı gürültüsü varken duvar takibi ile çıkarılan harita 56
Şekil 4.6 Ortam 3'te, standart sapması 2 cm olan algılayıcı gürültüsü varken duvar takibi ile elde edilen harita 57
Şekil 4.7 Ortam 1'de, standart sapması 4 cm olan algılayıcı gürültüsü varken duvar takibi ile çıkarılan harita 57
Şekil 4.8 Ortam 2'de, standart sapması 4 cm olan algılayıcı gürültüsü varken duvar takibi ile çıkarılan harita 58
Şekil 4.9 Ortam 3'te, standart sapması 4 cm olan algılayıcı gürültüsü varken duvar takibi ile çıkarılan harita 58
Şekil 4.10 Ortam 1'de, gürültü yokken keşif stratejisi kullanıldığında çıkarılan harita 59
Şekil 4.11 Ortam 2'de, gürültü yokken keşif stratejisi kullanıldığında çıkarılan harita 59
Şekil 4.12 Ortam 3'te gürültü yokken keşif stratejisi ile elde edilen harita 60
Şekil 4.13 Ortam 1'de, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında çıkarılan harita 60
Şekil 4.14 Ortam 2'de, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında çıkarılan harita 61
Şekil 4.15 Ortam 3'te, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi ile elde edilen harita 61
Şekil 4.16 Ortam 1'de, standart sapması 4 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında çıkarılan harita 62
Şekil 4.17 Ortam 2'de, standart sapması 4 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında çıkarılan harita 62
Şekil 4.18 Ortam 3'te, standart sapması 4 cm olan algılayıcı gürültüsü varken keşif stratejisi ile elde edilen harita 63

Şekil 5.1 Ortam 1'de, gürültü yokken keşif stratejisi kullanıldığında iki robot tarafından çıkarılan harita 64
Şekil 5.2 Ortam 2'de, gürültü yokken keşif stratejisi kullanıldığında iki robot tarafından çıkarılan harita 65
Şekil 5.3 Ortam 3'te, gürültü yokken keşif stratejisi kullanıldığında iki robot tarafından çıkarılan harita 65
Şekil 5.4 Ortam 1'de, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında iki robot tarafından çıkarılan harita 66
Şekil 5.5 Ortam 2'de, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında iki robot tarafından çıkarılan harita 66
Şekil 5.6 Ortam 3'te, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında iki robot tarafından çıkarılan harita 67
Şekil 5.7 Ortam 1'de, standart sapması 4 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında iki robot tarafından çıkarılan harita 67
Şekil 5.8 Ortam 2'de, standart sapması 4 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında iki robot tarafından çıkarılan harita 68
Şekil 5.9 Ortam 3'te, standart sapması 4 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında iki robot tarafından çıkarılan harita 68
Şekil 5.10 Ortam 1'de, gürültü yokken keşif stratejisi kullanıldığında dört robot tarafından çıkarılan harita 69
Şekil 5.11 Ortam 2'de, gürültü yokken keşif stratejisi kullanıldığında dört robot tarafından çıkarılan harita 69
Şekil 5.12 Ortam 3'te, gürültü yokken keşif stratejisi kullanıldığında dört robot tarafından çıkarılan harita 70
Şekil 6.1 GKF'nin hareket gürültüsünde yaptığı iyileştirmeyi gözlemlemek amacıyla robotun dolaştırıldığı ortam 71
Şekil 6.2 Robotun ortamda attığı dört tur boyunca konumda x ekseni doğrultusunda meydana gelen değişim. 72
Şekil 6.3 GKF kullanılmazken robotun x eksenindeki konum hatasının değişimi 72
Şekil 6.4 GKF kullanılırken robotun x eksenindeki konum hatasının değişimi 73
Şekil 6.5 Robotun ortamda attığı dört tur boyunca konumda y ekseni doğrultusunda meydana gelen değişim. 74
Şekil 6.6 GKF kullanılmazken robotun y eksenindeki konum hatasının değişimi 74
Şekil 6.7 GKF kullanılırken robotun y eksenindeki konum hatasının değişimi 75

Şekil 6.8 Robotun ortamda attığı dört tur boyunca yönelme açısında meydana gelen değişim.76
Şekil 6.9 GKF kullanılmazken robotun yönelme açısındaki hatanın değişimi 76
Şekil 6.10 GKF kullanılırken robotun yönelme açısındaki hatanın değişimi..... 77
Şekil 7.1 Ortam 1'de, gürültü yokken duvar takibi ile elde edilen haritalar. a. Tek Robotla İki Noktadan Doğru Çıkarımı Yaparak Haritalama b. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama 81
Şekil 7.2 Ortam 2'de, gürültü yokken duvar takibi ile elde edilen haritalar. a. Tek Robotla İki Noktadan Doğru Çıkarımı Yaparak Haritalama b. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama 82
Şekil 7.3 Ortam 1'de, standart sapması 2 cm olan algılayıcı gürültüsü varken duvar takibi ile çıkarılan haritalar. a. Tek Robotla İki Noktadan Doğru Çıkarımı Yaparak Haritalama b. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama..... 83
Şekil 7.4 Ortam 2'de, standart sapması 2 cm olan algılayıcı gürültüsü varken duvar takibi ile çıkarılan haritalar. a. Tek Robotla İki Noktadan Doğru Çıkarımı Yaparak Haritalama b. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama..... 84
Şekil 7.5 Ortam 1'de, standart sapması 4 cm olan algılayıcı gürültüsü varken duvar takibi ile çıkarılan haritalar. a. Tek Robotla İki Noktadan Doğru Çıkarımı Yaparak Haritalama b. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama..... 85
Şekil 7.6 Ortam 2'de, standart sapması 4 cm olan algılayıcı gürültüsü varken duvar takibi ile çıkarılan haritalar. a. Tek Robotla İki Noktadan Doğru Çıkarımı Yaparak Haritalama b. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama..... 86
Şekil 7.7 Ortam 1'de, gürültü yokken keşif stratejisi kullanıldığında çıkarılan haritalar. a. Tek Robotla İki Noktadan Doğru Çıkarımı Yaparak Haritalama b. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama c. Robot Takımı ile Haritalama87
Şekil 7.8 Ortam 2'de, gürültü yokken keşif stratejisi kullanıldığında çıkarılan haritalar. a. Tek Robotla İki Noktadan Doğru Çıkarımı Yaparak Haritalama b. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama c. Robot Takımı ile Haritalama88
Şekil 7.9 Ortam 3'te gürültü yokken keşif stratejisi ile elde edilen haritalar. a. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama b. Robot Takımı ile Haritalama 89
Şekil 7.10 Ortam 1'de, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında çıkarılan haritalar. a. Tek Robotla İki Noktadan Doğru Çıkarımı Yaparak Haritalama b. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama c. Robot Takımı ile Haritalama 90

Şekil 7.11 Ortam 2'de, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında çıkarılan haritalar. a. Tek Robotla İki Noktadan Doğru Çıkarımı Yaparak Haritalama b. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama c. Robot Takımı ile Haritalama 91
Şekil 7.12 Ortam 3'te, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi ile elde edilen haritalar. a. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama b. Robot Takımı ile Haritalama..... 92
Şekil 7.13 Ortam 1'de, standart sapması 4 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında çıkarılan haritalar. a. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama b. Robot Takımı ile Haritalama 93
Şekil 7.14 Ortam 2'de, standart sapması 4 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında çıkarılan haritalar. a. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama b. Robot Takımı ile Haritalama 94
Şekil 7.15 Ortam 3'te, standart sapması 4 cm olan algılayıcı gürültüsü varken keşif stratejisi ile elde edilen haritalar. a. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama b. Robot Takımı ile Haritalama..... 95

ÇİZELGE LİSTESİ

Çizelge 6.1 Haritalardaki dolaşım süreleri 79

ÖNSÖZ

Akıllı bir robot düşüncesi beni her zaman büyülemiştir. Küçüklükte belki bir oyun arkadaşı olarak hayatımda yer alabileceğini düşündüğüm robotların, büyüdükçe insanlık için önemli işler yapabildiğini, daha fazlasını da yapabileceğini gördüm, şimdi bu yolda yapılan çalışmalar arasında benim çalışmamın da bir yeri olacağını bilmek bana büyük mutluluk veriyor. Robotik, her yönüyle karmaşık, oldukça zor bir uğraş fakat mesleğimi seçmemdeki en büyük etken olan bir şey yaratma, uğraştığım şeyin sonuçlarını karşımda görme hazzını fazlasıyla yaşatan bir alan. Bana bu alanda çalışma fırsatı veren, bilimle, robotlarla, yapay zekayla uğraşmak gibi hayallerimi gerçekleştirmemi sağlayan değerli hocam Yrd. Doç. Dr. Sırma Yavuz'a sonsuz teşekkürlerimi sunuyorum. Çalışmam boyunca bana yardımcı olan, fikirlerini sunan Yrd. Doç. Dr. M. Fatih Amasyalı'ya ve matematik bilgisini ve değerli fikirlerini her zaman benimle paylaşan Muhammet Balcılar'a teşekkürü bir borç bilirim. Son olarak, beni hiçbir zaman yalnız bırakmayan, her zaman destekleyen, bana huzur veren anneme tüm emekleri için çok teşekkür ediyorum.

ÖZET

Bilinmeyen bir ortama bırakılan bir robotun ortamın haritasını çıkarırken aynı anda çıkardığı haritayı ve algılayıcı verilerini kullanarak kendi konum bilgisinde oluşan hatayı düzeltmesi literatürde Eş Zamanlı Konum Belirleme ve Haritalama problemi olarak geçmektedir. Bu problemde algılayıcıların doğruluğu büyük önem taşıdığından çözüm olarak önerilen sistemlerde genelde çok yüksek maliyetli lazer tarayıcılar kullanılmaktadır fakat bu tip pahalı sistemler her uygulama için uygun olmamaktadır. Örneğin evlerde kullanılacak temizlik robotlarının ya da tehlikeli ortamlarda çalışan gözden çıkarılabilir robotların düşük maliyetli olması beklenir. Robotlarda düşük maliyeti sağlamak için oldukça düşük fiyatlı olan kızılötesi ya da sonar algılayıcı kullanmak uygundur. Bu nedenle bu çalışmada altı adet kızılötesi algılayıcıya sahip düşük maliyetli robotlardan oluşan bir robot takımı ile ortamdaki duvarlar veya mobilyaların doğrusal özellikleri kullanılarak Genişletilmiş Kalman Filtresi metoduyla haritalama yapılması önerilmiştir. Denemeler Cyberbotics firması tarafından geliştirilen Webots simülasyon ortamında gerçekleştirilmiştir.

ABSTRACT

The problem of a robot to create a map of an unknown environment while correcting its own position based on the same map and sensor data is called Simultaneous Localization and Mapping problem. As the accuracy and precision of the sensors have an important role in this problem, most of the proposed systems use high cost laser scanners but this kind of expensive systems are not suitable for some implementations. For example, the robots built for house cleaning or disposable robots which will be used for hazardous material detection or mobile sensor networks must be low-cost. In order to build a low-cost robot, it is appropriate to use low-cost sensors (like infrared and sonar). In this project, a robot team that consists of low-cost robots creates a map of an unknown environment using line features like walls and furnitures and using Extended Kalman Filter method. Experiments are performed in Webots Simulation environment which is developed by Cyberbotics company.

1. GİRİŞ

Günümüzde sanayide, uzay araştırma projelerinde, kurtarma görevlerinde ve daha pek çok alanda robot kullanımı oldukça yaygındır. Sanayide kullanılan robotlar genellikle sabit konumlu, önceden adım adım belirlenmiş hareketleri yapan makinelerdir, oysa uzay araştırma projelerinde ve kurtarma görevlerinde kullanılan robotlar mobil robotlardır ve bunlara önceden sabit davranışlar atanması mümkün değildir. Bunlar, ya bir insan operatör tarafından kumanda edilirler ya da otonom olarak çalışırlar. Bir mobil robotun bir ortamda kendisinden beklenen görevleri otonom olarak yerine getirebilmesi için ortamın bir modeline ihtiyacı vardır fakat bu modeli robota sağlamak her zaman mümkün olmamaktadır. Bu durumda robotun ortam modelini kendisinin oluşturması beklenmektedir. Bu modeli oluştururken robotun ortamda gezinmesi, algılayıcıları aracılığı ile ortam hakkında bilgi toplayarak bu bilgiyi uygun formatta saklaması ve bu bilgiyi kullanarak ortamda gezinmeye devam etmesi gerekmektedir. Robotun gerçekte yaptığı eylem ile kendisinden yapması beklenen eylem arasında her zaman bir hata payı olması ve robotun gerçekte yaptığı eylemi ölçmek için kullanılan iç algılayıcı verisinin yani odometri bilgisinin de gürültülü ya da düşük çözünürlüklü olabilmesi nedeniyle gezinme sırasında robotun konumuna dair belirsizlik oluşmakta, bu belirsizlik her adımda artmaktadır. İşte bu nedenle robotun ortam haritasını çıkarırken bir yandan da bu haritayı kullanarak kendi konumunu belirlemesi gerekmektedir. Literatürde bu problem Eş Zamanlı Konum Belirleme ve Haritalama (EZKBH) problemi olarak geçmektedir. EZKBH probleminde robot yalnızca algılayıcıları aracılığıyla elde ettiği z gözlemi ve kumanda tarafından gelen u kontrol işareti bilgisine sahiptir. Çalışmalarda genellikle kontrol işareti olarak iç algılayıcıların döndürdüğü geri besleme, yani odometri verisi kullanılmaktadır.

1.1 Tezin Amacı ve Kapsamı

Eş Zamanlı Konum Belirleme ve Haritalama probleminde algılayıcıların doğruluğu büyük önem taşıdığından çözüm olarak önerilen sistemlerde genelde çok yüksek maliyetli lazer tarayıcılar kullanılmaktadır fakat bu tip pahalı sistemler her uygulama için uygun olmamaktadır. Örneğin evlerde kullanılan temizlik robotlarının ya da tehlikeli ortamlarda çalışan gözden çıkarılabilir robotların düşük maliyetli olması beklenir. Robotlarda düşük maliyeti sağlamak için oldukça düşük fiyatlı olan kızılötesi ya da sonar (ultrasonik) algılayıcı kullanmak uygundur. Bu nedenle bu çalışmada altı adet kızılötesi algılayıcıya sahip düşük maliyetli robotlardan oluşan bir robot takımı ile ortamdaki duvar, mobilya gibi doğrusal

özellikler kullanılarak “Çevrimiçi Haritalama” yapılması önerilmiştir. Doğusal özelliklerin çıkarımı için uygun bir metot seçimi de tezin kapsamı dahilindedir. EZKBH algoritması olarak Genişletilmiş Kalman Filtresi tercih edilmiştir. Denemeler Cyberbotics firması tarafından geliştirilen Webots simülasyon ortamında (Michel, 2004) gerçekleştirilmiştir.

1.2 Robotikte Belirsizlik

Gerçek dünyada bir görevi yerine getirmek üzere oluşturulan bir robotik sistemin başa çıkması gereken büyük miktarda belirsizlik bulunmaktadır. Ortamdaki belirsizliği oluşturan etkenler şöyle sıralanabilir (Siegwart ve Nourbakhsh, 2004; Thrun vd., 2005):

- Algılayıcılar mükemmel değildir: Algılayıcılar menzil ve çözünürlük olarak sınırlıdırlar. Algılayıcının özelliklerinden kaynaklanan kısıtlar ve belirsizlikler mevcuttur. Örneğin yansıma tabanlı çalışan mesafe algılayıcılarda yansımanın gerçekleştiği nesnenin rengi ve yapıldığı malzeme başarıyı etkiler, düzgün ve açılı bir nesne yansımayı algılayıcıya değil de başka bir tarafa doğru gönderebilir, kamera kullanılan ve renklere bağlı işlem yapılan bir sistemde ışıklandırmadaki değişim başarısızlığa yol açabilir. Mesafe algılayıcı verileri, hatta bazı durumlarda kamera verileri de, ayırt edici değildir, robot ortam içinde pek çok yerde benzer bir algılayıcı verisi alabilmektedir, dolayısı ile bir okumada robotun nerede olduğunun anlaşılması mümkün olmamaktadır.
- Robot kendisinden beklenen hareketi mükemmel şekilde yapamamaktadır: Hareketleri gerçekleştiren motorlar kontrol gürültüsü ve yıpranma nedeniyle hatalı çalışabilmektedirler. Tekerlek çaplarında farklılık olabilmekte, tekerleklerden biri olması gerekenden ufak bir miktar önde ya da geride olabilmekte, bozuk yüzey ya da tekerlekteki şekil bozuklukları nedeniyle tekerleğin temas yüzeyi değişebilmekte, tekerlek kayabilmektedir.
- İçsel algılayıcı verisi olarak kabul edilebilecek odometri bilgisi de dışsal algılayıcılar gibi gürültüye maruz kalmakta, sınırlı çözünürlük nedeniyle belirsizliği artırmaktadır. Ayrıca bu bilgi her adımda robotun konumunu belirlemek için kullanıldığından kümülatif hata katedilen mesafe ile orantılı olarak ciddi boyutlara ulaşmaktadır.
- Robotun bulunduğu ortamda olan biten her zaman tahmin edilebilir değildir. Montaj hattı gibi özel yapım alanlarda belirsizlik oldukça düşükken, ev, ofis gibi ortamlar dinamik bir yapıya sahiptir (nesnelerin yeri değişebilir, ortamda gezinen insanlar bulunabilir) dolayısı ile buralarda belirsizlik fazladır.
- Bir robotik sistemde konumun gerçek zamanlı olarak belirlenmesi gerektiğinden, konum

belirleme modülünün sonuç üretmesi gereken süre kısıtlıdır, bu da yapılabilecek işlemleri kısıtlar. Bu nedenle kullanılan algoritmalar genellikle zamanında cevap üretebilmek için yaklaşımsal sonuçlar üretmektedir.

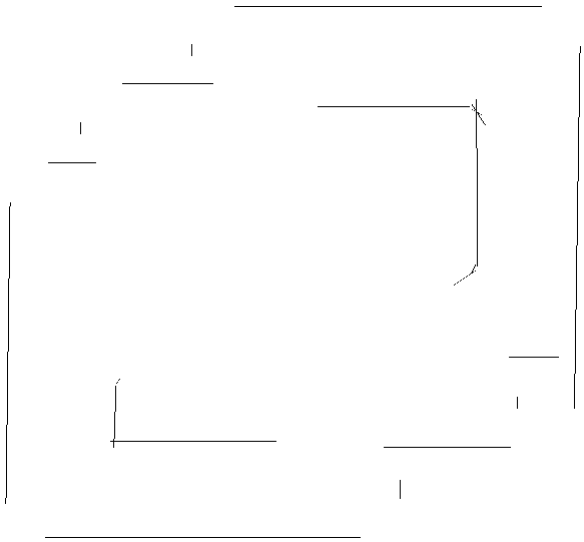
- Robotun ve ortamın mükemmel bir şekilde modellenmesi mümkün değildir. Bir model fiziksel dünyayı ancak belli bir ölçüde ifade edebilir.

Bu belirsizliklerin üstesinden gelebilmek için literatürde olasılıksal yöntemler önerilmiştir. Bu yöntemlerde eldeki bilgiler tek bir değer ile değil, bir olasılık dağılımı ile ifade edilmektedir.

1.3 Harita Temsili

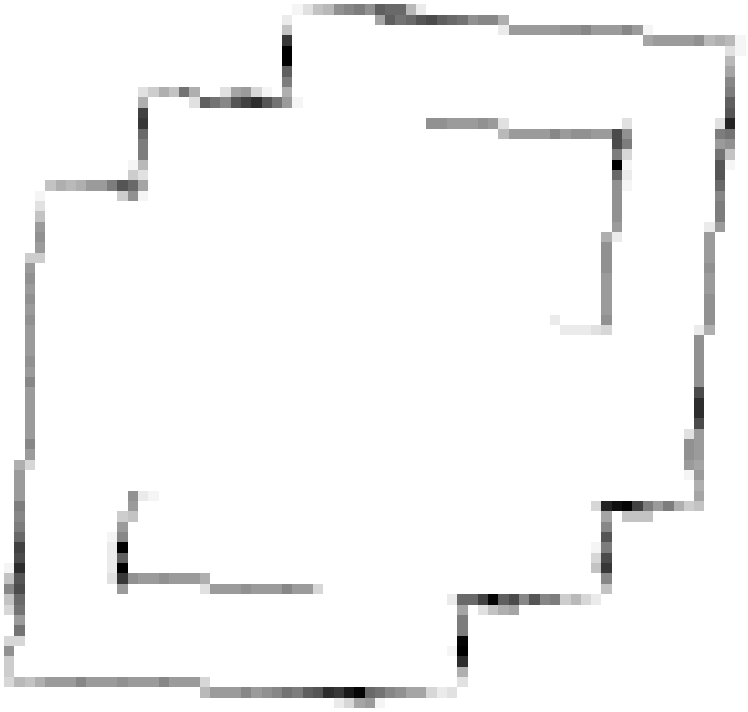
Robotikte, ortam modelinin ifade edilmesi için temel olarak kullanılan iki tip harita vardır. Bunlar özellik tabanlı harita (*feature-based map*) ve konum tabanlı haritadır (*location-based map*). Bunlara ek olarak bir de topolojik harita bulunmaktadır.

Özellik tabanlı haritalar, ortamı belirli yer işaretçileri (*landmark*) ve bunların küresel koordinat düzlemindeki konumlarıyla ifade ederler (Şekil 1.1). Sürekli (*continuous*) harita temsili mümkündür.



Şekil 1.1 Özellik tabanlı harita örneği (Bu örnekte özellik olarak doğrular kullanılmıştır)

Konum tabanlı haritalarda, harita ayrık (*discrete*) hücrelere bölünür ve ortam bu hücrelerin doluluğu, boşluğu, doluluk oranı, hatta içeriği ile ifade edilebilir (Şekil 1.2).

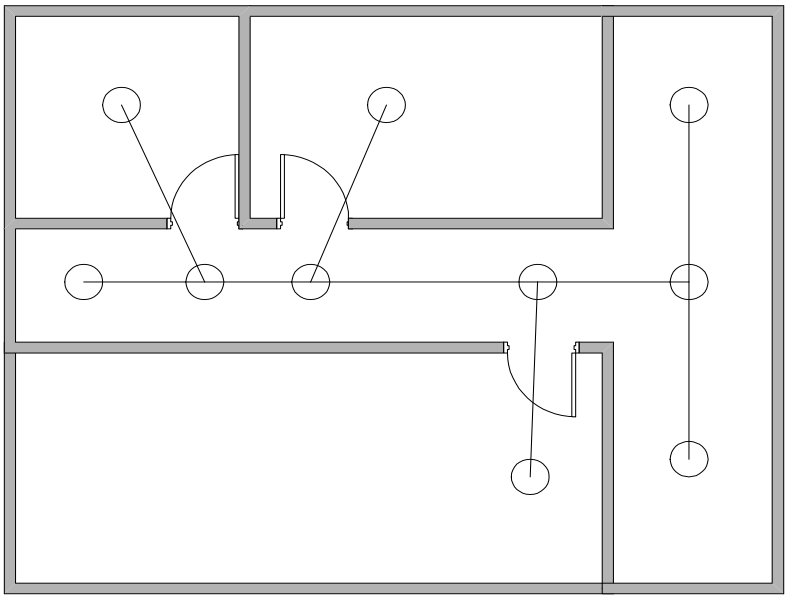


Şekil 1.2 Konum tabanlı harita örneği (Bu örnekte her hücrenin koyuluğu o hücrenin kaç defa dolu görüldüğüne bağlı olarak belirlenmiştir)

Özellik tabanlı haritaların bir avantajı, bir nesnenin yerinin değiştirilmesinin oldukça kolay olmasıdır, o yer işaretçisine ait konum parametrelerinin değiştirilmesi yeterlidir. Ayrıca özellik tabanlı haritaların kapladıkları bellek miktarı da genellikle düşüktür çünkü bir ortamı ifade etmek için yalnızca yer işaretçilerine ait birkaç parametre tutulmaktadır.

Konum tabanlı haritaların avantajı herhangi bir konumun dolu mu boş mu olduğunun kolayca anlaşılabilmesi, dolayısı ile dolaşım için planlama yapılmasına imkan verilmesidir. Yalnız her hücreye ait bilginin tutulması, haritanın kullandığı bellek miktarının özellik tabanlı haritalara göre çok fazla olmasına neden olmaktadır.

Diğer bir harita tipi olan topolojik haritalar düğümler ve bağlantılardan oluşan graflardır (Şekil 1.3). Düğümler, bir özelliğin varlığı ya da yokluğuyla kolay ayırt edilebilen belirli konumlardır, bağlantılar ise bu konumlar arasında yol olup olmadığını ifade etmektedir. Topolojik haritalarla konum tabanlı haritaları birlikte kullanan, topolojik haritadaki her düğümün kendi içinde bir konum tabanlı harita olduğu çalışmalar da bulunmaktadır (Chang vd. 2007).



Şekil 1.3 Bir ofis ortamının topolojik harita ile gösterimi

1.4 Mobil Robotlarda Konum Belirleme

Robot dolaşımı (*navigation*), bir mobil robotun sahip olması gereken en zorlayıcı özelliklerdendir. Dolaşımdaki başarı, dolaşımı mümkün kılan dört bileşenin başarısına bağlıdır. Bu bileşenler dış dünya algısı, robotun konumunun belirlenmesi, karar mekanizması ve hareket kontrolüdür (Siegwart ve Nourbakhsh, 2004). Bu bileşenlerden olan konum belirleme, daha önce ifade edilen belirsizlikler nedeniyle oldukça güç bir problemdir, bu problemde ortamın bir modeli mevcuttur, odometri bilgisi, algılayıcı verileri ve bu model kullanılarak robotun nerede olduğu bulunmaya çalışılır.

Konum belirleme problemi yerel konum belirleme ve küresel konum belirleme olarak ikiye ayrılır:

- Yerel konum belirleme probleminde robotun başlangıç konumu bilinmektedir ve hareket gürültüsü hesaba katılarak robot konumu her adımda takip edilir. Robotun konumuna dair belirsizlik robotun gerçek konumu çevresindeki küçük bir alanla sınırlı olduğu için bu problem yerel konum belirleme problemi olarak adlandırılır.
- Küresel konum belirleme probleminde robotun ilk konumu bilinmemektedir. Robot ortam içinde herhangi bir yerde bulunabileceği için belirsizlik robotun çevresi ile sınırlı değildir. Bu nedenle bu probleme küresel konum belirleme problemi denir. Küresel konum

belirleme probleminin daha zorlaştırılmış hali kaçırılmış robot problemidir. Bu problemde robotun konumunun farklı zamanlarda aniden ve büyük miktarda değiştirildiği varsayılmaktadır. Küresel konum belirlemede robot kendi konum bilgisine dair belirsizliğin farkındadır, robotun konumuna dair *inancına* göre hareket etmesinde hiçbir sakınca yoktur. Fakat kaçırılmış robot probleminde robot tam kendi konumuna dair güçlü bir inanca sahipken robotun kaçırılması olasılığı vardır ve robotun bu durumda kaçırıldığını anlaması ve yeniden kendi konumunu belirlemesi gerekmektedir.

Olasılıksal konum belirleme metotlarında algılayıcı gürültüsünün, hareket gürültüsünün ve robotun konumuna dair inancının temsil edilmesinde olasılık dağılımları kullanılmaktadır.

1.4.1 Olasılıksal Konum Belirleme Metotları

Olasılıksal konum belirleme metotlarında yapılan işlemler hareket güncellemesi ve algı güncellemesi şeklinde ikiye ayrılabilir:

- Hareket güncellemesi, robotun kendi konumuna dair inancında odometri bilgisi kullanılarak yapılan güncellemedir. Konumuna dair $t-1$ anındaki x_{t-1} inancına sahip bir robotun t anındaki u_t kontrol işareti ile nereye gitmiş olabileceği (x'_t) hesaplanır.
- Algı güncellemesi, robotun inancında algılayıcı bilgileri, yani gözlemler kullanılarak yapılan güncellemedir. t anında z_t gözlemini yapan robotun, bu gözlemi yaptığına göre nerede olabileceği (x_t) hesaplanır. Hesaplamada ortam modeli, algılayıcı modeli ve hareket güncellemesi sonucu bulunan konum olasılıkları (x'_t) kullanılır.

Olasılıksal konum belirleme için kullanılan iki temel metot Markov Konum Belirleme Algoritması ve Kalman Filtresidir.

Markov Konum Belirleme Algoritması'nda inanç durumu robotun ayrık bir harita üzerindeki olası her konumuna ayrı ayrı atanan olasılıklarla ifade edilir. Hareket ve algı güncellemesi işlemleri her hücrenin olasılığını günceller.

Kalman Filtresi ise inanç durumunu bir Gauss olasılık dağılımı ile ifade eder, dolayısı ile hareket ve algı güncellemesi işleminde yalnızca dağılıma ait μ ve σ parametreleri güncellenir.

Markov Konum Belirleme Algoritması ile Kalman Filtresi'nin belirsizliği temsil etme yöntemlerindeki farklılık, bunların kullanılabileceği alanların da farklı olmasına yol açar.

Markov Konum Belirleme Algoritması'nda olası her konum için bir olasılık tutulduğundan robot haritanın neresinde olursa olsun konum belirleme yapılabilir, dolayısı ile küresel konum belirleme problemine çözüm getirilebilir. Kalman Filtresi'nde ise belirsizlik robotun gerçek konumu çevresine yayılmış bir Gauss dağılımı ile ifade edildiğinden ancak konum takibi yapılabilir, yani yerel konum belirleme problemine çözüm getirilebilir.

Markov Konum Belirleme Algoritması'nda olası her konumun olasılığında güncelleme yapıldığından bu algoritmanın işlem yükü fazladır. Bu yükü azaltmak üzere Parçacık Filtresi ya da Monte Carlo algoritması olarak adlandırılan yöntem önerilmiştir. Bu yöntemde inanç durumu olası konumların bir alt kümesi için hesaplanan olasılık değerleriyle ifade edilir. Bu alt kümenin her bir elemanı parçacık olarak adlandırılır. Her güncelleme sonrası olası konumlar arasından yeniden örnekleme yapılır; örnekleme, konumların olasılıklarıyla ağırlıklandırılarak, robotun bulunma olasılığı yüksek yerlerden daha fazla örnek alınması sağlanır. Genellikle konum belirlemede bir sapma olması olasılığı göz önünde bulundurularak, birbiçimli (*uniform*) bir dağılımdan rastgele örnekler de eklenir. Fox (2003) tarafından önerilen KLD-Örnekleme metodunda da örnek sayısının her adımda yeniden belirlenmesi sağlanarak performans artışı sağlanmıştır.

1.5 Eş Zamanlı Konum Belirleme ve Haritalama

Eş Zamanlı Konum Belirleme ve Haritalama, robotun bir yandan ortam haritasını çıkarırken bir yandan da bu haritayı kullanarak kendi konumunu belirlemesi problemidir.

Temel olarak "Çevrimiçi EZKBH" ve "Tam EZKBH" olmak üzere iki tür Eş Zamanlı Konum Belirleme ve Haritalama problemi vardır (Thrun vd., 2005):

Çevrimiçi EZKBH'de algılayıcı ölçümleri ve kontrol işaretleri kullanılarak robotun o andaki konumunun ve ortam haritasının elde edilmesi amaçlanır. Olasılık modeli (1.1) eşitliğindeki gibidir.

$$p(x_t, m \mid z_{1:t}, u_{1:t}) \tag{1.1}$$

Burada x_t robotun t anındaki pozu yani noktasal konumu ve yönü, m harita olmak üzere $z_{1:t}$ ve $u_{1:t}$ başlangıçtan t anına kadar olan ölçümler ve kontrol işaretleridir. Bu probleme çevrimiçi denmesinin nedeni yalnızca t anındaki değişkenleri içermesidir. Çevrimiçi EZKBH problemi için geliştirilmiş algoritmaların çoğu artımsaldır, yani kullandıktan sonra geçmiş ölçümleri ve kontrol işaretlerini saklamazlar, yalnızca son gelen ölçümleri ve kontrol

işaretlerini kullanırlar.

Tam EZKBH'de harita ile birlikte yalnızca t anındaki poz değil, başlangıçtan t anına kadarki tüm pozların belirlenmesi amaçlanır. Olasılık modeli (1.2) eşitliğindeki gibidir.

$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t}) \tag{1.2}$$

1.5.1 Eş Zamanlı Konum Belirleme ve Haritalama Metotları

Eş Zamanlı Konum Belirleme ve Haritalama probleminin çözümü için kullanılan en eski metot Genişletilmiş Kalman Filtresi'dir (*Extended Kalman Filter*) (Thrun vd., 2005). Çevrimiçi EZKBH problemine çözüm getiren bu yöntemde algı ve hareketteki gürültünün Gauss dağılımına sahip olduğu varsayılmaktadır. Durum değişkenleri de yine Gauss dağılımı ile ifade edilmektedir; örneğin robotun konumu tutulurken ortalama değer (μ) ve kovaryans (Σ) matrisleri kullanılmaktadır. Hareket güncellemesinde robot konumu güncellenip hareket gürültüsü de konum belirsizliğine eklenirken, algı güncellemesinde yapılan gözlem ile bunun ilişkilendirildiği yer işaretçisinin görülmesi beklenen konum arasındaki farka ve bunların belirsizliklerine bağlı olarak tüm durum matrisinde güncelleme yapılmakta, belirsizlikler azaltılmaktadır. GKF algoritmasına EK 1'de yer verilmiştir.

EZKBH problemine çözüm olarak önerilen diğer bir önemli yöntem FastSLAM yöntemidir. Parçacık filtresine dayanan ve Tam EZKBH problemini çözebilen bu yöntem, değişkenlerin bir kısmını parçacıklarla ifade eden, kalanını ise Gauss dağılımları ile ifade eden Rao-Blackwellized parçacık filtresi yaklaşımını kullanır çünkü tüm değişkenlerin parçacıklarla ifade edilmesi değişkenlerin fazlalığı ve değişkenlerin boyutları nedeniyle mümkün değildir. FastSLAM yönteminde parçacıklar içinde robot yolu ile yer işaretçileri bulunmaktadır, tüm yer işaretçileri ortalama değer (μ) ve kovaryans (Σ) parametreleri ile ifade edilmektedir. FastSLAM yönteminin bir avantajı veri ilişkilendirmenin parçacık tabanlı yapılabilmesi dolayısı ile sadece en olası ilişkilendirmenin değil pek çok olası ilişkilendirmenin izlenebilmesidir. Diğer bir avantajı ise parçacık filtrelerinin doğrusal olmayan hareket modelleri ile başa çıkabilmesidir (Thrun vd., 2005).

1.6 Doğru Ayrıştırma ve Çıkarım Algoritmaları

Literatürde yer alan pek çok özellik tabanlı Eş Zamanlı Konum Belirleme ve Haritalama çalışmasında özellik olarak doğrular seçilmiştir. Bunun nedeni ofis, ev gibi ortamların

çoğunlukla düz duvarlardan ve mobilyalardan oluşuyor olması ve doğru çıkarımının göreceli olarak kolay olmasıdır. Yapılan çalışmalarda lazer tarayıcı verisinden (Einsele, 1997; Borges ve Aldon, 2000; Pfister vd., 2003; Garulli vd., 2005a, 2005b) ya da kızıl ötesi ve sonar gibi kısıtlı algılayıcılardan elde edilen verilerden (Yun vd., 1998; Großmann ve Poli, 1999, 2001; Tardós vd., 2002; Beevers ve Huang, 2006; Choi vd., 2006; Choi vd., 2008; Yap ve Shelton, 2009) doğru çıkarımı yapılabilmektedir.

Bu bölümde bazı doğru ayrıştırma ve çıkarım metotlarına yer verilmiştir. Bu metotlar Nguyen vd. (2005) ile Borges ve Aldon (2000) tarafından yapılan çalışmalarda anlatılan metotlardan derlenmiştir.

Bu metotlarda kullanılan algılayıcı verileri lazer tarayıcısının açısal dönüşü ile veya robotun hareketi ile sıralı olarak elde edilmiş ya da sonradan sıralanmış noktasal verilerdir.

Bu bölümde yer verilen metotlardan Ardışıl Kenar Takibi (*Successive Edge Following*) ve Doğru İzleme'nin (*Line Tracking*) eşik değer seçimi konusunda oldukça hassas olduğu Borges ve Aldon (2000) tarafından ifade edilmiştir. Nguyen vd. (2005) yaptıkları karşılaştırmalı çalışmada lazer tarayıcı verisinden doğru çıkarımında Ayırma-ve-Birleştirme (*Split-and-Merge*) algoritması ile Doğru İzleme algoritmasının hız ve doğruluk açısından karşılaştırdıkları diğer metotlar olan Doğru Bağlanımı (*Line Regression*), Rastgele Örnek Uzlaşımı (*Random Sample Consensus* - *RANSAC*), Hough dönüşümü ve Beklenti Enbüyültme'ye (*Expectation Maximization*) göre üstün olduğunu söylemişlerdir.

1.6.1 Ardışıl Kenar Takibi (*Successive Edge Following*)

Bu yöntemde, bir doğru parçasındaki ardışık tüm noktalar arasındaki mesafenin bir eşik değerden küçük olması şartı göz önünde bulundurularak, sırayla tüm noktalar arasındaki mesafeye bakılır. Bir doğru parçasına son olarak (x_{n-1}, y_{n-1}) noktası eklendiyse, (x_n, y_n) noktasının eklenmesi için bu ikisi arasındaki mesafeye bakılır, eğer bu mesafe belli bir eşik değerden küçükse ya da bu değere eşitse, (x_n, y_n) noktası da doğru parçasına eklenir, aksi halde önceki doğru parçası (x_{n-1}, y_{n-1}) noktası ile sonlanır ve (x_n, y_n) noktası ile yeni bir doğru parçası başlatılır.

1.6.2 Doğru İzleme (*Line Tracking*)

Bu yöntemde, o anda bakılan nokta olan (x_n, y_n) noktasının, doğru parçasını oluşturan

$p = \{(x_i, y_i), (x_{i+1}, y_{i+1}), ..., (x_{n-2}, y_{n-2}), (x_{n-1}, y_{n-1})\}$ noktalarına uydurulan doğruya olan T_n mesafesine bakılır ve $T_n > T_{maks}$ ise yeni bir doğru araması (x_{n+2}, y_{n+2}) ile başlatılır ve $p = \{(x_n, y_n)(x_{n+1}, y_{n+1})\}$ noktalarına uydurulan doğruya bakılır.

1.6.3 İteratif Uç Nokta Uydurma (*Iterative End Point Fit*)

Bu yöntemde, $p = \{(x_1, y_1), (x_2, y_2), ..., (x_{n-1}, y_{n-1}), (x_n, y_n)\}$ nokta kümesi daha fazla ayrılamaz oluncaya kadar $p' = \{(x_1, y_1), (x_2, y_2), ..., (x_{a-1}, y_{a-1}), (x_a, y_a)\}$ ve $p'' = \{(x_a, y_a), (x_{a+1}, y_{a+1}), ..., (x_{n-1}, y_{n-1}), (x_n, y_n)\}$ şeklinde iki kümeye ayrılır. Burada ayrılma noktası olan (x_a, y_a), p nokta kümesinin uç noktaları olan (x_1, y_1) ile (x_n, y_n) noktaları kullanılarak oluşturulan doğruya en uzakta bulunan noktadır. Doğru ile bu nokta arasındaki mesafenin bir eşik değeri geçmemesi durumunda doğruyu bölme işlemi sonlandırılır.

1.6.4 Ayırma-ve-Birleştirme (*Split-and-Merge*)

Nguyen vd. (2005) tarafından belirtildiğine göre görüntü işlemeye (Pavlidis ve Horowitz, 1974) dayanan bu yöntem, İteratif Uç Nokta Uydurma gibi iteratif bir yöntemdir. İteratif Uç Nokta Uydurma yönteminden farkı, ayrılma noktası bulunurken bu noktanın uç noktalar tarafından oluşturulan doğruya olan mesafesine değil de tüm noktalara uydurulan doğruya olan mesafesine bakılması ve bölme safhası bittikten sonra aynı hizadaki doğru parçalarının birleştirilmesidir.

1.6.5 Prototip Tabanlı Bulanık Kümeleme Algoritması (*Prototype-based Fuzzy Clustering Algorithm*)

Bu yöntemde bir maliyet fonksiyonunun minimize edilmesi amaçlanır. Genelde (1.3) eşitliğinde yer alana benzer bir fonksiyon kullanılır.

$$J(\beta, U, Z) = \sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij}^{m} d^2(x_j, y_j, \beta_i) \tag{1.3}$$

Bu eşitlikte yer alan semboller ve anlamları aşağıda listelenmiştir:

- C : Grup sayısı, yani noktaların kaç gruba ayrılacağı
- N : Nokta sayısı
- β_i : i . prototipin parametrelerini ifade eden bir sembol

- u_{ij}: j. noktanın β_i prototipine aitlik derecesi
- $d(x_j, y_j, \beta_i)$: j. nokta ile β_i prototipi arasındaki mesafeyi veren bir fonksiyon

Algoritma veri kümesi, grup sayısı ve başlangıç prototiplerinin verilmesiyle başlar. Öncelikle prototipler doğru kabul edilerek fonksiyonu minimize eden aitlik dereceleri (u_{ij}) hesaplanır. Daha sonra hesaplanan u_{ij} değerleri kullanılarak fonksiyonu minimize eden β_i prototipleri hesaplanır. Eğer herhangi bir prototip parametresinde bir eşik değerinden büyük bir değişim gerçekleşirse u_{ij} ve β_i güncellemeleri yeniden yapılır.

Bu yöntemin dezavantajı, ilk prototiplerin doğru seçimine, J 'nin yerel minimum değerlerine ve gürültüye karşı hassas olması ve başlangıçta grup sayısının verilmesi gerekliliğidir (Borges ve Aldon, 2000).

1.6.6 Bulanık Ayırma-ve-Birleştirme (*Split-and-Merge Fuzzy*)

Borges ve Aldon (2000) tarafından önerilen bu yöntem Prototip Tabanlı Bulanık Kümeleme Algoritması'nın bulanık kümeleme yaklaşımını iteratif bir şekilde uygulamaktadır. Bölme safhasını birleştirme safhası izlemektedir. Bölme safhasında, onay ölçütü olarak (1.4) eşitliği kullanılmaktadır.

$$\sigma_k^2 = \frac{1}{s_k} \sum_{j \in q_k} d^2(x_j, y_j, \beta_k) \tag{1.4}$$

Bu eşitlikte yer alan semboller ve anlamları aşağıda listelenmiştir:

- s_k: k. doğruyu oluşturan nokta sayısı
- β_k: k. prototipin parametrelerini ifade eden bir sembol
- q_k: k. doğruyu oluşturan noktaların indislerini tutan bir vektör

Bölme safhası eldeki tüm noktalara bir doğru uydurulmasıyla başlar. Eldeki doğruların onay ölçütünü sağlayıp sağlamadığına bakılır, $\sigma_k \leq \sigma_{maks}$ koşulunu sağlamayan doğruları oluşturan nokta kümesine $C = 2$ değeri için bulanık kümeleme yöntemi uygulanır, yani bu nokta kümelerinin her birinden iki ayrı doğru elde edilir ve elde edilen doğrular doğru listesine eklenir. Onay ölçütünü sağlamayan doğru var olduğu sürece bu işlem tekrarlanır.

Birleştirme safhasında her l_k doğrusu için, ağırlık merkezi koordinatları l_k doğrusunun ağırlık merkezi koordinatlarına en yakın olan iki doğru seçilir. l_k doğrusu iki doğruyla da ayrı ayrı

birleştirilerek oluşan doğruların onay ölçütünü sağlayıp sağlamadıklarına bakılır, onay ölçütünü sağlayan ve daha küçük bir σ değeri veren doğru kazanmış olur.

Birleştirme işlemi, birleştirilecek iki doğruyu oluşturan noktalar kümesine bir doğru uydurulmasıyla gerçekleştirilir.

1.6.7 Doğru Bağlanımı (*Line Regression*)

Nguyen vd. (2005) tarafından belirtildiğine göre Arras ve Siegwart (1997) tarafından önerilen bu yöntemde belli bir büyüklükte bir kayan pencere tanımlanarak sırasıyla bu pencere içinde kalan noktalara doğru uydurulur. Bir doğru güvenilirliği dizisi oluşturulur ve bu dizinin her gözüne ardışık üç pencere kayması sonucu oluşturulan doğrular arasındaki Mahalanobis mesafelerinin toplamı konur. Yığmalı Hiyerarşik Kümeleme (*Agglomerative Hierarchical Clustering*) ile doğru güvenilirliği dizisine bakılarak mesafesi eşik değerden düşük ardışık elemanlar birleştirilerek doğru parçaları elde edilir. Çakışan doğru parçaları birleştirilir ve parametreleri yeniden hesaplanır.

1.6.8 Rastgele Örnek Uzlaşımı (*Random Sample Consensus – RANSAC*)

Bu yöntemde nokta kümesinden rastgele iki nokta seçilip, diğer noktaların bu iki noktadan geçirilen doğruya olan mesafesine bakılır. Eğer bu doğruya dahil kabul edilebilecek yakınlıkta yeterli sayıda (en az C adet) nokta varsa doğruya dahil sayılan tüm noktalar kullanılarak bir doğru uydurulur. Doğru bir listeye eklenirken kullanılan noktalar ise kümeden çıkarılır. Bu işlemler C'den az nokta kalana kadar ya da belli sayıda iterasyon tamamlanana kadar tekrarlanır.

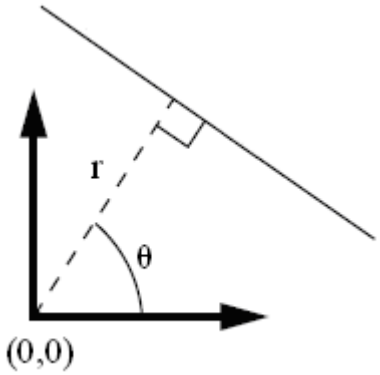
1.6.9 Hough Dönüşümü (*Hough Transform*)

Hough Dönüşümü, esas olarak Paul Hough tarafından bulunmuş fakat tüm dünyada kabul gören haline Duda ve Hart (1972) tarafından getirilmiş, doğru, çember ve elips gibi parametrik olarak ifade edilebilen şekillerin tanınmasında kullanılan bir özellik çıkarım tekniğidir.

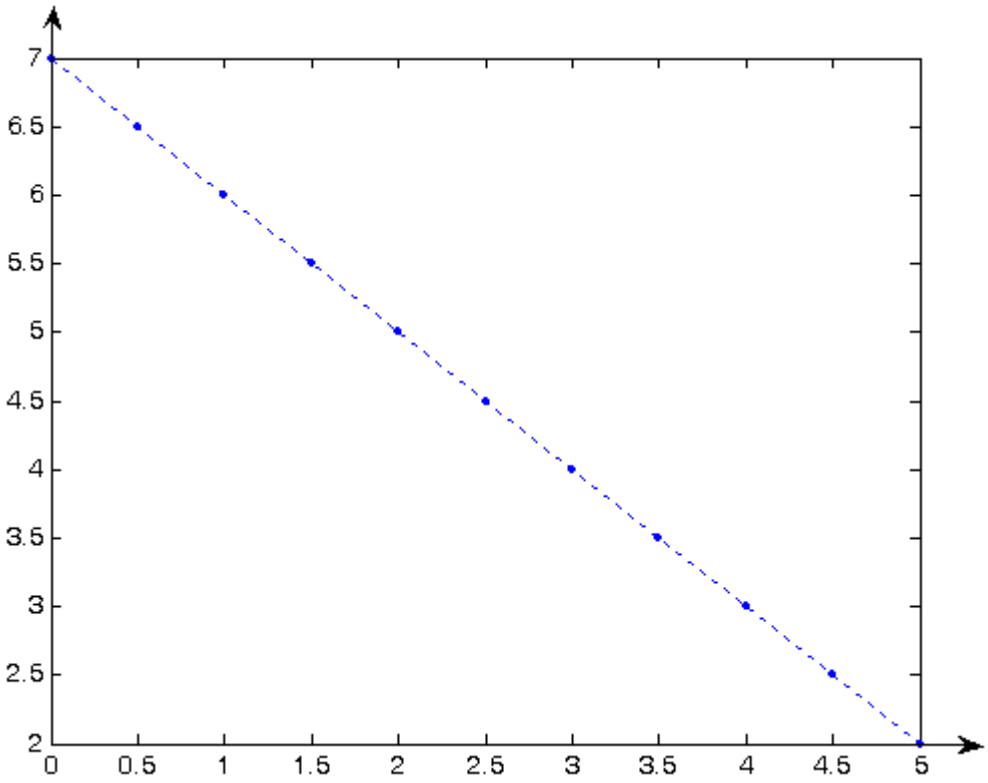
Kartezyen koordinat düzleminde bir doğruyu, orijine olan r mesafesi ve x ekseni ile yaptığı θ açısı kullanılarak ifade etmek mümkündür (Şekil 1.4). Bu doğruyu Hough uzayında ifade etmek istersek, doğru burada (r,θ) parametreleri ile ifade edilen bir nokta halini alır.

Şekil 1.5'teki noktaları ele alalım, her noktadan sonsuz sayıda doğru geçtiği için bu

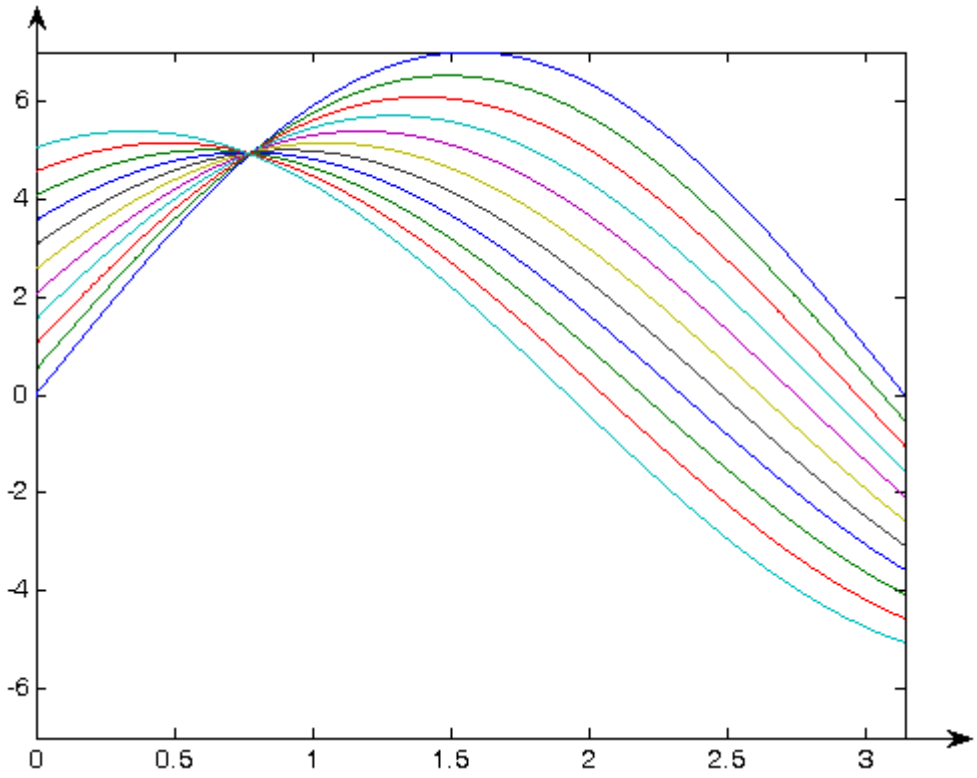
noktalardan geçen tüm doğruları Hough uzayında gösterdiğimizde ortaya nokta sayısı kadar sinüzoidal eğri çıkar (Şekil 1.6). Bu noktaların tümünden geçen doğru ise sinüzoidal eğrilerin kesişim noktasıdır.



Şekil 1.4 Kartezyen düzlemde bir doğrunun (r,θ) parametreleri ile gösterimi



Şekil 1.5 Kartezyen düzlemde aynı hizada bulunan noktalar ve bunlardan geçen doğru



Şekil 1.6 Şekil 1.5'te görülen noktaların Hough uzayındaki temsili

Hough Dönüşümü, sinüzoidal eğrilerin kesişim noktalarını bularak doğru parametrelerini elde eder. Kesişim noktaları bulunurken r ve θ değerleri belli bir çözünürlükte ayrıklaştırılır, Hough uzayındaki her ayrık hücre için bir akümülatör oluşturulur, her nokta için (1.5) eşitliği kullanılarak r ve θ değerleri hesaplanarak (r,θ) çiftleri için oylamalar yapılır. Oy sayıları belli bir eşik değeri geçen (r,θ) çiftleri bir doğru olarak kabul edilir.

$$r = x\cos\theta + y\sin\theta \tag{1.5}$$

Hough Dönüşümünün eksik yönleri olarak (r,θ) parametreleri için uygun çözünürlüğü bulmanın güç olduğu ve doğru parametreleri hesaplanırken Hough Dönüşümü'nün gürültüyü ve belirsizliği ele almaması söylenebilir (Nguyen vd., 2005).

İkinci sorunu çözmek için Nguyen vd. doğru parametrelerini bulmak için Hough Dönüşümü ile elde edilen doğruyu oluşturan (doğru parametrelerine oy veren) noktalara Toplam-En-Küçük-Kareler (*Total-Least-Squares*) metodu ile doğru uydurmuşlardır.

1.6.10 Basit Doğrusal Bağlanım (*Simple Linear Regression*)

Basit Doğrusal Bağlanım metodunda amaç $\{x_i, y_i\}$ şeklinde n adet noktadan oluşan veri

kümesine en iyi doğruyu, yani, en küçük kareler bakış açısı ile, karesel hata toplamını en küçük yapan doğruyu uydurmaktır.

Doğru denklemi olarak Eşitlik 1.6 kullanıldığında, uydurulan doğrunun hatası Eşitlik 1.7'deki gibi hesaplanabilir.

$$y = \alpha + \beta x \tag{1.6}$$

$$Q(\alpha, \beta) = \sum_{i=1}^{n} \hat{\varepsilon}_i^2 = \sum_{i=1}^{n} (y_i - \alpha - \beta x_i)^2 \tag{1.7}$$

Basit Doğrusal Bağlanım $Q(\alpha, \beta)$'yi minimize etmeye çalışmaktadır. $\min_{\alpha,\beta} Q(\alpha, \beta)$'yi sağlayan $\hat{\alpha}$ ve $\hat{\beta}$ değerleri Eşitlik 1.8 ve 1.9'daki gibi hesaplanabilir. Burada görülen $\bar{x}$ ve $\bar{y}$ değerleri noktaların sırasıyla x ve y koordinat değerlerinin ortalamalarıdır.

$$\hat{\beta} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \tag{1.8}$$

$$\hat{\alpha} = \bar{y} - \hat{\beta}\bar{x} \tag{1.9}$$

1.7 Önceki Çalışmalar

Eş Zamanlı Konum Belirleme ve Haritalama son 25 yıldır oldukça ilgi çeken bir konu olmuştur ve bu alanda birçok çalışma yapılmıştır. Yapılan çalışmalarda Eş Zamanlı Konum Belirleme ve Haritalama algoritmaları üzerinde durulduğu kadar özellik çıkarımı üzerinde de durulmuştur. Noktasal algılayıcı verisinden doğru çıkarımı, üzerinde fazlaca durulan bir konudur. Burada, genel hatlarıyla bu konuda yapılmış pek çok çalışmaya yer verilmiştir.

Einsele (1997) tarafından yapılan çalışmada bilinmeyen bir ortamda gezinen bir robotun kendi konumunda düzeltme yapması ve ortamın haritasını çıkarması amaçlanmıştır. İteratif Uç Nokta Uydurma ile lazer tarayıcı verileri doğru parçalarına ayrılmış, bir doğru parçasına ait nokta sayısı belirli bir değerin altındaysa o doğru parçası yok sayılarak gürültülü verilerden kurtulunmuş ve En Küçük Kareler Doğru Uydurma Metodu ile doğru parçalarını oluşturan noktalara doğru uydurulmuştur. Yeni çıkarılan doğrular ile daha önceden elde edilen doğruları eşleştirmek için bir miktar değiştirilmiş bir Dinamik Programlama metodu kullanılmıştır.

Yun vd. (1998) tarafından yapılan çalışmada doğrusal özellikler kullanılarak robotun kendi konumunu belirlemesi üzerine çalışılmıştır. Doğru çıkarımı için seyrek ve gürültülü sonar algılayıcı verisine Hough Dönüşümü uygulanmış, daha sonra da En Küçük Ortalama Kare (*Least Mean Square*) metodu ile doğru uydurulmuştur. Hough Dönüşümü'nün farklı parametre çözünürlükleri için başarısı değerlendirilmiştir.

Großmann ve Poli (1999, 2001) tarafından yapılan çalışmada seyrek ve gürültülü algılayıcı verisine Hough Dönüşümü uygulanarak doğru denklemleri elde edilmiş ve bu doğruların harita ile eşleştirilmesiyle robotun konumunda düzeltme yapılmıştır. Denemeler yedi adet sonar algılayıcıya sahip bir robot ile yapılmıştır.

Borges ve Aldon (2000) tarafından yapılan çalışmada, lazer tarayıcı verisinden doğru çıkarımı yapmak için Prototip Tabanlı Bulanık Kümeleme Algoritması'nı temel alan fakat doğru sayısının önceden verilmesinin gerekmediği iteratif bir metot olan Bulanık Ayırma-ve-Birleştirme algoritması önerilmiştir. Borges ve Aldon tarafından yapılan diğer bir çalışmada (2004) ise elde edilen noktasal verilerde kırılma noktası (*breakpoint*) tespit etmek amacıyla Uyumcul Kırılma Noktası Detektörü (*Adaptive Breakpoint Detector*) ve Kalman Filtresi Tabanlı Kırılma Noktası Detektörü (*Kalman Filter Based Breakpoint Detector*) denenmiştir. Uyumcul Kırılma Noktası Detektörü, iki nokta arasındaki mesafenin uyumcul bir eşik değerden küçük olup olmamasına bağlı olarak kırılma noktası tespit eden bir metottur. Kalman Filtresi Tabanlı Kırılma Noktası Detektörü ise Castellanos ve Tardós (1996) tarafından önerilen stokastik bir kırılma noktası detektörünün bir miktar değiştirilmiş halidir. Doğru çıkarımı için Doğru İzleme, İteratif Uç Nokta Uydurma ve Bulanık Ayırma-ve-Birleştirme metotları uygulanarak bu metotlar kıyaslanmış, Bulanık Ayırma-ve-Birleştirme metodunun diğerlerine göre oldukça başarılı olduğu ifade edilmiştir.

Tardós vd. (2002) tarafından yapılan çalışmada sonar verisine Hough Dönüşümü uygulanarak hem doğrusal hem de noktasal özellikler, yani duvarlar ve köşeler çıkarılmıştır. Birbirinden bağımsız olarak oluşturulan sınırlı büyüklükteki stokastik yerel haritaların birleştirilmesi için bir yöntem önerilmiştir. Harita birleştirirken öncelikle sadece dönüşüm (*transformasyon*) kullanılmış, bundan sonra yerel haritalarda ortak özellikler olabileceği için, bunların bulunması amacıyla Neira ve Tardós (2001) tarafından önerilen Bağlı Uyum Dal ve Sınır Algoritması (*Joint Compatibility Branch and Bound Algorithm*) kullanılmıştır.

Pfister vd. (2003) tarafından yapılan çalışmada doğru tabanlı harita oluşturabilmek için algılayıcı verilerinden doğru çıkarımı yapan bir metot önerilmiştir. Hough Dönüşümü ile

kabaca doğrular ve bu doğruları oluşturan noktalar bulunduktan sonra noktalar ile doğrular arasındaki farkın oluşturduğu hatayı minimize etmek için Maksimum Olabilirlik *(Maximum Likelihood)* yaklaşımından yararlanılmıştır. Doğrunun kendisini oluşturan noktalara olan mesafesi, yani hata, her nokta için ağırlıklandırılmış, bu ağırlıklar ise modellenmiş algılayıcı belirsizliğine göre belirlemiştir. Doğru uydurma işleminde belirsizlik de göz önünde bulundurulmuş ve doğrunun belirsizliğinin (kovaryans) hesabı için kapalı formda bir denklem elde edilmiştir. Aynı ya da farklı pozlarda alınmış iki tarama sonucunda üretilmiş doğruların birleştirilmesi için önce doğrular aynı ortak referans çerçevesine taşınmış, sonra Ki-Kare (*Chi-Square*) testi ile farklarına bakılmıştır, fark bir eşik değerden düşükse doğruların parametreleri ve kovaryans matrisleri kullanılarak birleştirme yapılmıştır. Denemelerde lazer tarayıcılı bir robot kullanılmıştır.

Burguera vd. (2005) tarafından yapılan çalışmada genelde lazer tarayıcı verisine uygulanan (ve aslında bilgisayarla görme konusundaki çalışmalardan alınan) bir tarama eşleştirme (*scan matching*) algoritması olan İteratif En Yakın Nokta (*Iterative Closest Point*) algoritması sonar algılayıcı verisine uygulanmıştır. Tarama eşleştirme yöntemine yenilik olarak, algılayıcı okumalarının gruplandığı bir önişlem ile gruplama işleminde kullanılan tüm robot yörüngesinde, yani gruplanacak veri toplanırken robotun aldığı yolda düzeltme yapılan bir sonişlem eklenmiştir. İteratif En Yakın Nokta metodu ile birlikte onun biraz farklı hali olan İteratif İkili Uyuşum (*Iterative Dual Correspondance*) metodu test edilmiştir. İteratif İkili Uyuşum metodunun İteratif En Yakın Nokta metodundan daha iyi sonuçlar verdiği tespit edilmiştir.

Garulli vd. (2005a; 2005b) tarafından yapılan çalışmalarda GKF kullanılarak doğru tabanlı haritalama ile ilgilenilmiştir. Lazer tarayıcı verisinden doğru çıkarımı için segmentasyon ve doğru uydurma peş peşe sürekli olarak yapılmıştır. Segmentasyon işleminde Ardışıl Kenar Takibi ile Doğru İzleme yöntemlerinde yer alan koşullar birleştirilmiştir, yani bir noktanın bir doğru parçasına olan uzaklığı ve o doğru parçasına ait son noktaya olan uzaklığı belirli değerlerden küçük değilse bu noktanın bu segmente ait olmadığı kabulü ile yeni bir doğru parçası başlatılmıştır. Yanlış doğru çıkarımını engellemek için L_0 uzunluğundan kısa ya da N_0'dan az noktadan oluşan doğrular güvenilmez kabul edilmiş ve reddedilmiştir. Verinin istatistiksel özelliklerinden kovaryans matrisleri elde edilmiştir. Gözlenen bir doğrunun eldekilerle kıyaslanması için doğruların açısal farkına, doğrular arasındaki mesafeye ve iki doğrunun çakışma oranına bakılmış, bunlara bakarken varyans kullanılarak ağırlıklandırma yapılmıştır. Yeni eklenecek bir yer işaretçisinin önce bir listeye eklenmesi, sonradan

güvenilirliği ortaya çıkınca, örneğin belli bir süre içinde belli sayıda görülünce, esas yer işaretçisi listesine eklenmesi önerilmiştir.

Nguyen vd. (2005) tarafından yapılan çalışmada lazer tarayıcı verisiyle kullanılan Ayırma-ve-Birleştirme, Doğru Bağlanımı, Doğru İzleme, Rastgele Örnek Uzlaşımı, Hough Dönüşümü ve Beklenti Enbüyültme doğru çıkarım algoritmaları kıyaslanmıştır. Doğru çıkarım algoritması ile doğruyu oluşturan noktalar bulunduktan sonra bu noktalara Toplam-En-Küçük-Kareler metodu ile doğru parametreleri ve kovaryans matrisi belirlenmiştir. Oldukça gürültülü ölçümleri elemek ve ham ölçüm verisini kümelemek için de Ardışıl Kenar Takibi benzeri bir yöntem kullanılmış, arka arkaya gelen noktalar arasındaki radyal fark bir eşik değerin üzerindeyse bu kırılma noktası kabul edilmiştir, böylece noktalar doğru parçalarına ayrılmıştır. Belli bir değerden az sayıda noktaya sahip doğru parçaları elenmiştir. Doğru parçalarına ayırma işlemi sonrası da Ki-Kare testi ile istatistiksel olarak benzer doğru parçaları yeniden birleştirilmiştir. Çalışmanın sonucunda Ayırma-ve-Birleştirme ve Doğru İzleme metotlarının hız ve doğruluk açısından diğerlerine göre daha başarılı olduğu tespit edilmiştir.

Choi vd. (2006) tarafından yapılan çalışmada 16 sonar algılayıcı ile bir stereo kamera birlikte kullanılarak haritalama yapılması amaçlanmıştır. Haritalama algoritması olarak GKF kullanılmıştır. Sonar algılayıcı verisinden nokta ve doğru özellikleri çıkarılırken kamera verisi kullanılarak da nesneler tanınmıştır. Sonar algılayıcı verisinden özellik çıkarımı için daha önceki çalışmalarında (Choi vd., 2005) kullandıkları yöntem kullanılmıştır. Nokta çıkarımı için bazı değişiklikler yapılarak Wijk ve Christensen (2000) tarafından önerilen Üçgenleme Tabanlı Füzyon (*Triangulation-Based Fusion*) algoritması kullanılmıştır. Doğru çıkarımı veri toplayarak değil de yalnızca o andaki sonar algılayıcı verileri kullanılarak yapılmıştır. Sonar verisinden doğru çıkarılırken şu koşullar kontrol edilmiştir:

- Art arda gelen üç sonar birbirine yakın veriler döndürmeli.
- Ortadaki sonar en küçük mesafe değerini döndürmeli.

Bu koşullar gerçekleşiyorsa doğru denklemlerini elde etmek için şunlar yapılmıştır:

- Merkez Doğrusu (*Centerline*) algılayıcı modeli (Choset vd., 2003) kullanılarak üç nokta elde edilmiştir. Bu modelde sonara ekonun açısal görüş alanının tam ortasından geldiği varsayılır.
- Elde edilen noktalara En Küçük Kareler Doğru Uydurma Metodu ile doğru uydurulmuştur.

Üç sonar da aynı doğruyu gördüğünden ve ortadaki sonarın açısal menzilinin tamamı tespit

edilen doğru özelliğinin içinde kaldığından bu sonarın açısal görüş alanı kullanılarak doğrunun uç noktaları tespit edilmiştir.

Beevers ve Huang (2006) tarafından yapılan çalışmada sınırlı algılayıcılara sahip bir robot ile Rao-Blackwellized Parçacık Filtresi kullanılarak Eş Zamanlı Konum Belirleme ve Haritalama yapılması amaçlanmıştır. Özellik olarak doğruların kullanıldığı bu çalışmada, belli sayıda adım boyunca ölçüm toplanıp daha sonra bu ölçümlerden doğru çıkarılmasına çalışılmıştır. Doğru çıkarımı için öncelikle eşik değeri ölçüm mesafesine göre belirlenen, Uyumcul Kırılma Noktası Detektörü benzeri bir kümeleme algoritması ile noktalar kümelere ayrılmıştır. Her kümeye İteratif Uç Nokta Uydurma uygulanarak gerektiği takdirde bunlar daha küçük kümelere ayrılmıştır. Bir eşik değerden daha fazla noktaya sahip kümelerin doğru parametrelerinin ve kovaryanslarının bulunması için ağırlıklandırılmış En Küçük Kareler Kestirimcisi (*Least Squares Estimator*) kullanılmıştır. Kovaryanslar hesaplanırken tüm veri yerine yalnızca doğru parçalarının uç noktaları kullanılmıştır. Denemelerde Radish'ten (Howard ve Roy, 2003) veri setleri kullanılmıştır. Çoğu veri 1° aralıklı lazer taraması olduğundan verinin büyük kısmı yok sayılıp 0°, 45°, 90°, 135° ve 180°'deki algılayıcı ölçümleri kullanılmıştır, maksimum algılayıcı okuması da 5 metre ile sınırlandırılmıştır (bazı durumlarda daha da az). Bu verilere gürültü eklendiğine dair bir şey yazılmadığından lazer tarayıcı doğruluğunda veriler ile çalışıldığı kabul edilebilir.

Choi vd. (2008) tarafından yapılan çalışmada seyrek ve gürültülü algılayıcı verisi ile haritalama yapılması amaçlanmıştır. Yer işaretçisi (landmark) olarak doğruların kullanıldığı bir Genişletilmiş Kalman Filtresi metodu tercih edilmiştir. Ortamdaki tüm doğruların birbirine paralel ya da dik olması koşulu öne sürülmüştür. Seyrek ve gürültülü algılayıcı verisinden doğru çıkarımı için sırası ile Sahte Yoğun Tarama (*Pseudo Dense Scan*), Yeniden Örnekleme (*Resampling*), İteratif Uç Nokta Uydurma, En Küçük Kareler Doğru Uydurma Metodu (*Least Square Line Fitting*), Kısıtlı Hough Dönüşümü (*Constrained Hough Transform*) yöntemleri kullanılmıştır. Algılayıcı gürültü modeli, Hough Dönüşümü'ndeki oylama sonuçları kullanılarak belirlenmiştir. Doğruları ilişkilendirmek için doğruların Hough uzayındaki parametrelerinin ağırlıklandırılmış Öklit uzaklıkları kullanılmıştır. Ayrıca doğrular arasında bir çakışma olup olmadığına bakılmıştır. Görünmeyen doğruların konumlarını tahmin eden aktif bir keşif stratejisi geliştirilmiştir. Bu stratejide, Choi vd. tarafından serbest nesne olarak adlandırılan, herhangi bir duvara dayanmayan nesnelerin paralellik ve diklik kısıtı nedeniyle dört ya da daha fazla doğru parçasından oluşması gerektiği kabul edilmiş, eğer robot tarafından keşfedilen böyle bir serbest nesne daha az doğruya sahipse nesnenin tüm kenarları

görülememiştir kabulü ile robotun serbest nesnenin etrafında tur atması sağlanmıştır. Denemeler 20 – 150 cm arasını ölçebilen yedi kızıl ötesi algılayıcıya sahip diferansiyel tekerli bir robot ile yapılmıştır.

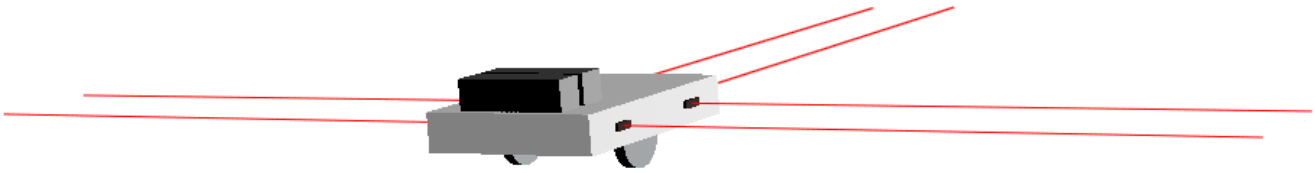
Yap ve Shelton (2009) tarafından yapılan çalışmada sonar algılayıcılara sahip bir robot ile Parçacık Filtresi kullanılarak doğru tabanlı haritalama yapılması amaçlanmıştır. Doğruların birbirine dik ya da paralel olduğu varsayılmıştır. Doğru çıkarımı için öncelikle Kultanen vd. (1990) tarafından önerilen Rastlantısal Hough Dönüşümü (*Randomized Hough Transform*) uygulanmış, böylece az çok aynı hizada olan noktalar bulunmuştur. Rastlantısal Hough Dönüşümü uygulanırken doğruların birbirine dik ya da paralel olması kısıtı da göz önünde bulundurularak açılarda düzeltme yapılmış ya da çıkarılan doğru elenmiştir. Daha sonra Ardışıl Kenar Takibi metodu ile kırılma noktaları belirlenip eldeki doğru parçaları daha küçük doğru parçalarına ayrılmıştır. Çıkarılan bir doğru parçası haritaya eklenmeden önce daha önce çıkarılan bir doğru ile birleştirilip birleştirilemeyeceğine bakılmış, bunun için iki doğrunun uç noktaları arasındaki mesafe ve iki doğru arasındaki dikey mesafe kontrol edilmiştir, bunlar belirli eşik değerlerden küçükse iki doğru birleştirilmiştir.

2. ÖNERİLEN YÖNTEM

Bu çalışmada Genişletilmiş Kalman Filtresi ile doğrusal özellikler kullanılarak haritalama yapılması amaçlanmıştır. Kullanılan metotların ayrıntılarına aşağıda yer verilmiştir.

2.1 Kullanılan Robot Modeli

Bu çalışmada, Yavuz vd. (2006) tarafından önerilen robot modeli temel alınarak oluşturulan, sağ ve sol yanları ile önünde ikişer adet olmak üzere toplam altı adet kızıl ötesi algılayıcıya sahip, kontrol işaretlerini algoritmayı yürüten bilgisayar ile haberleşerek alan diferansiyel tekerli bir robot modeli kullanılmıştır (Şekil 2.1). Yavuz vd. robotun sağ ve sol yanlarına ikişer algılayıcı yerleştirerek duvar takibi ile yapılan ortam keşfi sırasında robotun duvara paralel gidip gitmediğini anlayabilmeyi amaçlamışlardır. Bu çalışmada her yönde ikişer algılayıcı kullanılmasının nedeni, o yönde bulunan bir duvardan aynı anda iki okuma alınabilmesini ve elde edilen bu noktalara doğru uydurulabilmesini sağlamaktır.



Şekil 2.1 Kullanılan robot modeli

Yapılan pek çok EZKBH çalışmasında algılayıcı olarak lazer tarayıcı kullanılmaktadır (Pfister vd., 2003; Ko vd., 2003; Lakaemper vd., 2005; Garulli vd., 2005a, 2005b; Zhou ve Roumeliotis, 2006; Chang vd., 2007; Andersson ve Nygards, 2008; Je ve Kim, 2008) , bunun nedeni bu algılayıcıların yoğun ve yüksek doğruluğa sahip veri döndürmeleri ve uzun menzilli olmalarıdır. Lazer tarayıcıların dezavantajı ise bunların oldukça pahalı olması, dolayısı ile de temizlik robotu gibi tüketiciye yönelik robotlarda ya da robotun büyük olasılıkla bir daha kullanılamayacağı kurtarma görevleri veya tehlikeli madde tespiti için tasarlanan robotlarda kullanımlarının uygun olmamasıdır (Beevers, 2007). Kısıtlı algılayıcıların ilgi çekmesinin nedeni ise düşük fiyatlı olması, az güç tüketmesi ve az yer kaplamasıdır. Bu tip algılayıcılar düşük fiyatı nedeniyle robot takımının kullanıldığı çalışmalar için oldukça uygundur. Bu çalışmada da düşük maliyetli bir robot takımı ile haritalama yapmak amaçlandığından yalnızca 6 adet kızılötesi algılayıcı kullanılmıştır.

2.2 Robot Takımı Kullanımı ve Avantajları

Haritalamanın güvenilir bir şekilde yapılabilmesini ve hızlı olmasını sağlamak amacıyla bir robot takımı kullanılmasına karar verilmiştir. EZKBH problemi için tek robot yerine birden fazla robot kullanılması çeşitli avantajlar sağlamaktadır (Kobayashi vd. 2003; Zhao vd. 2006; Burgard vd. 2000; Zlot vd.2002). Bu avantajlar aşağıdaki gibi özetlenebilir:

- Çok sayıda robot bir ortamı tek robota göre daha hızlı keşfedebilir.
- Birden fazla ucuz robotun kullanıldığı durum, bir adet güçlü ve pahalı robotun kullanıldığı duruma göre hatalara daha dayanıklı olabilir, tek arıza noktası (*single point of failure*) sorunu engellenebilir.
- Robotlar algılayıcı verilerini, ortam hakkındaki bilgilerini paylaşarak kendi konumlarını daha iyi belirleyebilirler.
- Haritalarda çakışan, örtüşen bölgelerin değerlendirilmesiyle iyileştirme yapılabilir.

2.3 Keşif Stratejisi

Bilinmeyen bir ortama bırakılan robotların eksiksiz bir şekilde ortam haritasını çıkarmasını amaçlayan bir çalışmada keşif stratejisi çok önemlidir. Düzgün çalışan, verimli bir keşif stratejisi, haritalama süresinde ve haritanın başarısında doğrudan etkilidir. Bu çalışmada keşif stratejisi olarak iki farklı yöntem uygulanmıştır:

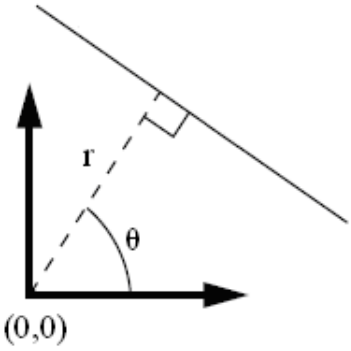
Kullanılan ilk keşif stratejisi duvar takibidir, bir duvarı görecek şekilde bırakılmış robot aynı taraftaki duvarları takip ederek ilerlemektedir. Burada takip edilen yer işaretçisinin duvar olması bir zorunluluk değildir, ortamdaki duvara yaslanmış mobilyalar da aynı şekilde takip edilmektedir. Duvar takibi sırasında yanlardaki algılayıcı ölçümleri kontrol edilerek robotun duvara fazla yaklaşması ya da duvardan fazla uzaklaşması önlenmekte, robotun önünde engel görülmesi durumunda da dönüş gerçekleşmektedir. Bu yöntemin avantajı robotun görüş alanı dahilinde her zaman bir yer işaretçisi bulunması, bu yer işaretçisinin haritaya dahil edilebilmesi ve aynı yer işaretçisi kullanılarak robotun konumunda düzeltme yapılabilmesidir. Bu keşif stratejisinin bir dezavantajı odanın ortasında bulunan ve “ada” olarak ifade edebileceğimiz mobilya, sütun gibi yer işaretçilerinin görülemeyecek olmasıdır. Diğer dezavantajı ise yalnızca tek robotla kullanıma uygun olmasıdır, birden fazla robot olduğunda bunların çarpışmasını, birbirlerini görmesini engelleyecek bir mekanizma bulunmamaktadır.

Kullanılan ikinci keşif stratejisinde, Yamauchi (1997, 1998) tarafından ortaya atılan sınır tabanlı keşif yönteminden esinlenilmiştir. Sınır tabanlı keşif yönteminde, keşfedilmiş alan ile

keşfedilmemiş alan arasındaki hücrelere sınır hücreleri denilmektedir. Robot bu hücrelere yönlendirilerek yeni alanların keşfedilmesi sağlanmaktadır. Bizim uygulamamızda ayrık hücrelere bölünmüş bir keşif haritası üzerinde Dijkstra algoritması (Dijkstra, 1959) ile en düşük maliyetli keşfedilmemiş, gidilebilecek alan bulunup robot o alana yönlendirilmektedir. Aramada iki farklı harita kullanılmaktadır. Yollar bulunup maliyet hesaplanırken algılayıcı ölçümleri sonucunda oluşturulan işgal edilmiş alan haritası kullanılmaktadır. Bu harita algılayıcı ölçümleri kullanılarak sürekli güncellenmekte, bir hücrede engel görülürse o hücrenin değeri adım adım sıfıra yaklaştırılmakta, aksi durumda hücrenin değeri 255'e yaklaştırılmaktadır. Bu mekanizma gürültülü algılayıcı verisiyle başa çıkılabilmesini sağlamaktadır. Bir alanın zaten keşfedilip keşfedilmediğine ise ikili değer alabilen bir başka haritadan bakılmaktadır. Bu keşif stratejisinde ortamdaki gidilmesi mümkün her alan keşfedilmekte, böylece kenarlardaki duvarlar da, ortada bulunan adalar da haritalanabilmektedir. Yamauchi'nin çalışmasında (1998) robot takımındaki tüm robotlar bağımsız olarak karar aldıkları için aynı hedeflere yönelebilmekte, dolayısı ile birbirlerini görebilmektedirler, robotların birbirini görmesi gürültü gibi ele alınmaktadır. Bizim uygulamamızda karar mekanizması merkezi olduğu için arama sonucu belirlenen hedefler kontrol edilerek robot takımındaki robotların birbirine yakın hedefler seçmesi önlenebilmekte, robotların birbirini görmesi engellenebilmektedir,. Bu keşif stratejisinin dezavantajı robotun yer işaretçisi görmeden uzun süre yol alabilmesi, dönüşler yapabilmesi, bunun sonucunda da robotun konumuna dair belirsizliğin artmasıdır.

2.4 Özellik çıkarımı

Bu çalışmada, haritalama için kullanılan robotların uygun maliyetli olabilmesi için kızılötesi algılayıcılar kullanılmıştır. Seyrek ve gürültülü veri döndüren algılayıcılar kullanılarak yapılan bir EZKBH çalışmasında, önceki çalışmalar da incelendikten sonra (Tardós vd., 2002; Beevers ve Huang, 2006; Choi vd., 2006; Choi vd., 2008; Yap ve Shelton, 2009) özellik olarak en uygun seçimin doğrular olduğuna karar verilmiştir. Doğruların temsilinde ise Hough Uzayı parametreleri (r, θ) kullanılmıştır (Şekil 2.2).

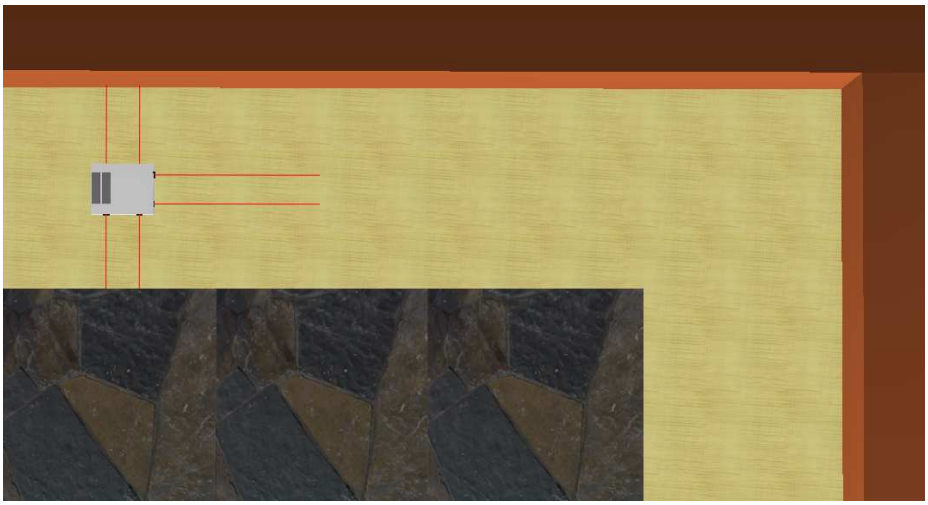


Şekil 2.2 Hough uzayı parametreleri ile doğru temsili

Hough Uzayı parametreleri ile ifade edilen doğrular için doğru denklemi (2.1) eşitliğinde verilmiştir.

$$r = x\cos\theta + y\sin\theta \tag{2.1}$$

Özellik olarak noktasal yer işaretçilerinin değil de doğruların kullanılmasının şöyle bir dezavantajı vardır; bu yer işaretçileri kullanılarak konumda yapılabilecek düzeltme sınırlıdır. Bir duvarı takip ederek ilerleyen robot ancak duvara olan mesafesine göre konumunda düzeltme yapabilir, iki yana kayıp kaymadığına bakabilir fakat ilerlediği yönde bir düzeltme yapabilmesi mümkün olmamaktadır (Şekil 2.3). İlerlediği yöndeki belirsizliği ancak daha önce gördüğü bir duvar karşısına çıkınca giderebilir, bir de karşısına ilk defa gördüğü bir duvar çıktığında bu duvarı yer işaretçisi listesine ekleyebilir ve duvara yaklaştığı süre boyunca (bu çalışma için, robotun duvarı görmesi ile dönüş yapması arasında geçen 40-50 santimetrelik ilerleme boyunca) konumunda oluşan hatada düzeltme yapabilir.



Şekil 2.3 Duvar takibi yapan robotun konumundaki hata her adımda artacaktır.

Doğru çıkarımı için iki yöntem kullanılmıştır. Bunlardan ilki robotun bir yanındaki iki algılayıcı aracılığıyla bir duvardan alınan iki noktaya doğru uydurulan İki Noktadan Doğru Çıkarımı metodudur. İkincisi ise Hough dönüşümüdür. İki Noktadan Doğru Çıkarımı metodunun denenmesinin nedeni basitliği ve uygulama kolaylığıdır, deneysel sonuçlarda yer verilen sorunlarla karşılaşıldıktan sonra farklı bir doğru çıkarım metodu arayışına gidilmiş, önceki bölümlerde yer verilen doğru çıkarım metotları gözden geçirilmiştir. Bu metotların çoğu ne yazık ki ancak yoğun ve doğruluğu oldukça yüksek olan lazer tarayıcı verisinde çalışabilecek yöntemlerdir, gürültüyü ele alacak mekanizmaları yoktur. Yalnızca Hough dönüşümü gürültülü ve seyrek algılayıcı verisi ile çalışabilecek, farklı duvarlardan alınan noktalara birden fazla doğru uydurabilecek bir yapıya sahip olduğundan, bu yöntem seçilmiştir.

2.4.1 İki Noktadan Doğru Çıkarımı

Kullanılan robot modelinin sağ ve sol yanlarında ikişer algılayıcı bulunduğu daha önce belirtilmişti. Robot duvar takibi yaparak ilerlerken robotun aynı tarafındaki iki algılayıcı da genellikle aynı duvarı göreceğinden, bu iki algılayıcının döndürdüğü noktasal verilerden geçen doğru bulunarak haritalama yapılabileceği düşünülmüştür. İki noktadan doğru çıkarımı, robotun bir duvardan fazla algılayıcı ölçümü alamayacağı durumlar için uygundur.

Bu yöntemde aynı doğrudan alınan iki noktanın da (2.1) eşitliğinde verilen doğru denklemini sağlaması gerektiğinden (2.2) eşitliği yazılabilir.

$$x_1 \cos\theta + y_1 \sin\theta = x_2 \cos\theta + y_2 \sin\theta \tag{2.2}$$

Buradan da (2.3), (2.4) ve (2.5) eşitliklerinde ifade edilen işlemlerden sonra (r,θ) parametrelerini veren (2.6) ve (2.7) eşitlikleri elde edilir.

$$(x_1 - x_2)\cos\theta = (y_2 - y_1)\sin\theta \tag{2.3}$$

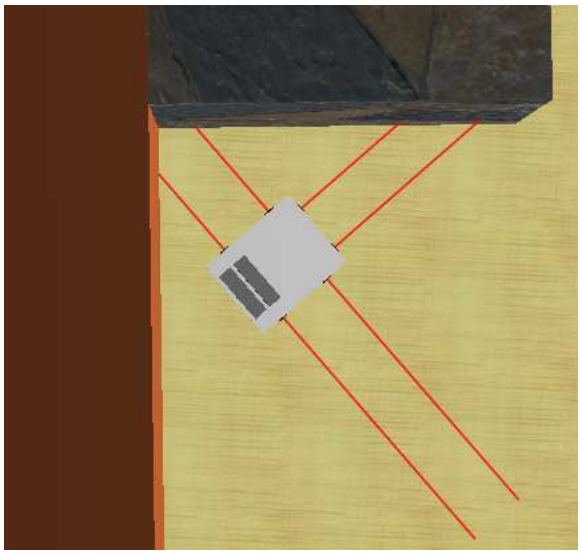
$$\frac{\sin\theta}{\cos\theta} = \frac{x_1 - x_2}{y_2 - y_1} \tag{2.4}$$

$$\tan\theta = \frac{x_1 - x_2}{y_2 - y_1} \tag{2.5}$$

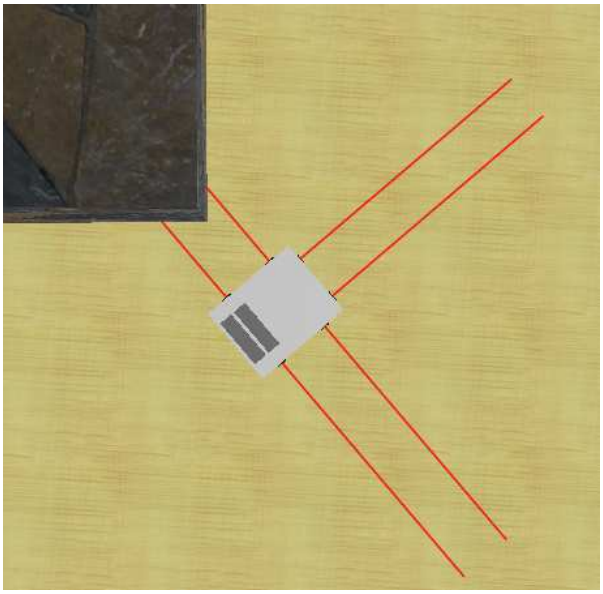
$$\theta = \arctan\left(\frac{x_1 - x_2}{y_2 - y_1}\right) \tag{2.6}$$

$$r = x_1 \cos\theta + y_1 \sin\theta \tag{2.7}$$

Bu yöntem kullanılarak yapılan denemelerde robotun aynı tarafındaki iki algılayıcının farklı duvarlar gördüğü durumlarda sorunla karşılaşılmıştır çünkü robot burada olması gerekenden farklı bir doğru bulmaktadır. Genellikle köşelerde karşılaşılan bu sorun Şekil 2.4 ve Şekil 2.5'te görülmektedir.



Şekil 2.4 İç köşede aynı taraftaki algılayıcıların farklı duvarları görmesi durumu



Şekil 2.5 Dış köşede aynı taraftaki algılayıcıların farklı duvarları görmesi durumu

Bu durumun ele alınması için iç köşelerde doğruların kesişim noktaları bulunarak köşe noktalarının elde edilmesi ve köşe noktası civarındaki ölçümlerin yok sayılması yolu

izlenmiştir. Dış köşelerin tespiti ise iç köşede olduğu gibi doğru kesişimleri ile bulunamaz çünkü ikinci doğru henüz görülmemiştir. Bu nedenle dış köşenin tespit edilmesi için şöyle bir yol izlenmiştir: Robot duvar takibi yaparken, takip edilen duvarın olduğu taraftaki algılayıcılardan öndeki bir engel göremez olduğunda robotun bir dış köşe ile karşılaştığı anlaşılır. Takip edilen duvarda görülen en son nokta köşe noktası kabul edilerek buna belli bir mesafeden daha yakın alınan okumalar göz ardı edilir.

2.4.2 Hough Dönüşümü ile Doğru Çıkarımı

Bu yöntemde, yanlardaki algılayıcılar aracılığıyla belli sayıda adım boyunca noktasal veri toplanmaktadır. Daha sonra bu nokta kümesini ifade edecek doğruların parametrelerini bulmak için her nokta (2.1) eşitliğinde yerine konularak belli bir çözünürlükteki (r,θ) ikililerinden hangilerinin sağlandığı bulunmakta, bu ikililer lehine oy kullanılmaktadır. Belli sayıda oydan fazlasını alan (r,θ) ikilileri o nokta kümesi içindeki doğrular olarak kabul edilmektedir.

Burada, nokta kümesinden birden fazla doğru çıkarılabiliyor olması önemlidir, bu sayede robotun bir köşeyi döndüğü durumda, dönüş boyunca görülen iki duvar için ayrı ayrı doğru elde edilebilmektedir, İki Noktadan Doğru Çıkarımı metodunda ifade edilen köşe sorunu ortadan kalkmaktadır.

2.4.3 Basit Doğrusal Bağlanım ile Doğru Çıkarımı

Bu çalışmada Basit Doğrusal Bağlanım, Hough dönüşümü sonucu bulunan parametrelerin iyileştirilmesi amacıyla kullanılmıştır. Hough dönüşümü sonucu bulunan doğrulara oy veren yani modele uygun olan noktalara (*inliers*) uygulanarak daha hassas sonuçlar elde edilmesi amaçlanmıştır.

Bu çalışmada doğruların temsilinde Hough Uzayı parametreleri (r,θ) kullanılmaktadır, Basit Doğrusal Bağlanım metodu doğru denklemini Eşitlik 2.8'deki gibi ele aldığından bunlar arasında bir dönüşüm gerekmektedir.

$$y = \alpha + \beta x \tag{2.8}$$

Dönüşüm için öncelikle Eşitlik 2.9'da yer alan doğru temsili, 2.8'deki temsile benzer bir biçime getirildi (Eşitlik 2.10).

$$r = x\cos\theta + y\sin\theta \tag{2.9}$$

$$y = \frac{r}{\sin\theta} - x\frac{\cos\theta}{\sin\theta} \tag{2.10}$$

Eşitlik 2.8 ile 2.10 birlikte ele alındığında 2.11 ve 2.12'de yer verilen eşitlikler açıkça görülmektedir.

$$\alpha = \frac{r}{\sin\theta} \tag{2.11}$$

$$\beta = -\frac{\cos\theta}{\sin\theta} \tag{2.12}$$

Eşitlik 2.11 ve 2.12 kullanılarak da (r,θ) parametrelerini veren Eşitlik 2.13 ile 2.14'e ulaşılmaktadır.

$$r = \alpha\sin\theta \tag{2.13}$$

$$\theta = \arctan\left(\frac{1}{-\beta}\right) \tag{2.14}$$

Basit Doğrusal Bağlanım kullanılırken noktaların Eşitlik 2.8'de verilen denkleme uyacağının garantisi yoktur. $x = sabit$ şeklinde doğrular uydurulması gerektiğinde Eşitlik 2.8'e denklemine göre doğru uydurmaya çalışmak başarısızlıkla sonuçlanmaktadır. Bu nedenle bu çalışmada noktaların x ve y koordinatlarındaki varyanslarına bakılmış, varyansı fazla olan bağımsız değişken seçilmiştir. Yani noktaların yayılımı x yönündeyse x bağımsız değişken kabul edilmiştir ve Eşitlik 2.8'e göre doğru uydurulmuştur. Noktaların yayılımı y yönündeyse y bağımsız değişken kabul edilmiştir ve Eşitlik 2.15'e göre doğru uydurulmuştur.

$$x = \alpha + \beta y \tag{2.15}$$

Eşitlik 2.15'teki (α,β) parametreleri ile (r,θ) parametreleri arasındaki dönüşüm için, daha önce kullanılan yöntem kullanılmış ve Eşitlik 2.9'da verilen doğru temsili 2.15'te verilene benzer bir biçime getirilmiş, Eşitlik 2.16 elde edilmiştir. Buradan da r ve θ parametrelerinin çekilmesiyle dönüşümde kullanılan Eşitlik 2.17 ile 2.18 elde edilmiştir.

$$x = \frac{r}{\cos\theta} - y\frac{\sin\theta}{\cos\theta} \tag{2.16}$$

$$r = \alpha\cos\theta \tag{2.17}$$

$$\theta = \arctan(-\beta) \tag{2.18}$$

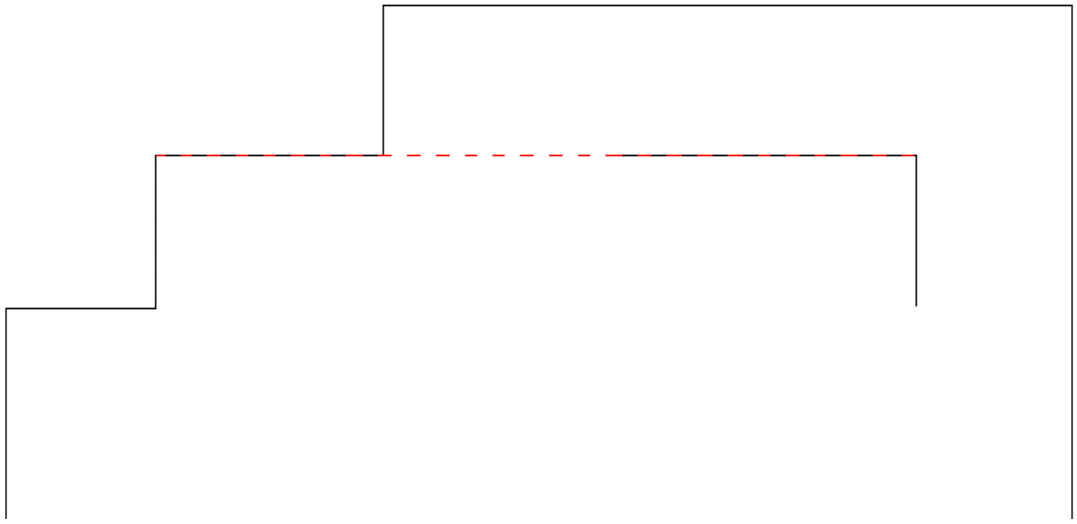
2.5 Genişletilmiş Kalman Filtresi

Genişletilmiş Kalman Filtresi'nde tüm durum değişkenlerinin parametreleri durum matrisi denilen bir sütun matrisinde tutulur. GKF, EZKBH problemine uygulandığında durum değişkenleri robot konumu ve yer işaretçilerinin konumu olmaktadır. Durum matrisinde başta robot konumu ile ilgili parametreler, daha sonra ise yer işaretçilerinin konumu ile ilgili parametreler tutulur. Dolayısı ile bu uygulamada durum matrisinin üst satırlarında robotların konum parametreleri olan (x, y, θ), sonraki satırlarda ise her bir yer işaretçisi için doğru parametreleri olan (r, θ) değerleri tutulmaktadır.

GKF algoritması da daha önce bahsedilen hareket güncellemesi ve algı güncellemesini içerir. Hareket güncellemesinde kinematik denklemler kullanılarak robotun konumunu ifade etmekte kullanılan Gauss dağılımının ortalama ve kovaryans değerleri güncellenir. Algı güncellemesinde ise en son görülen yer işaretçilerinin daha önceden görülen yer işaretçilerinden biri olup olmadığına bakılır, bunlar daha önceden görüldülerse gözlemler arasındaki farklılığa ve gözleme duyulan güvene bağlı olarak yer işaretçisi konumunda ve robot konumunda düzeltme yapılır, ilk defa görülüyorlarsa bu yer işaretçileri durum matrisine eklenirler. Yeni görülen bir yer işaretçisi ile durum matrisindeki bir yer işaretçisi arasındaki ilişkiyi saptamak için Mahalanobis mesafesi kullanılır. GKF algoritmasının adımları detaylı olarak Ek 1'de verilmiştir.

GKF'de hareket gürültüsü ve algılayıcı gürültüsü ortalaması sıfır olan Gauss dağılımlarıyla ifade edilir. Bu dağılımda ortamdaki gürültü miktarına göre kovaryans matrisindeki değerlerde değişiklik yapılarak gürültüyü daha doğru temsil etmek mümkündür. Kovaryans matrisinde yapılan değişiklikler yer işaretçileri karşılaştırılırken bakılan Mahalanobis mesafesinin sonucunu etkilemektedir, gürültü arttıkça birbirine belli bir uzaklıktaki yer işaretçilerinin aynı yer işaretçisi olma ihtimalinin daha fazla olduğu düşünülmektedir, gürültü azaldıkça bunların aynı kabul edilmesi için birbirlerine daha yakın görülmüş olmaları beklenir.

Bu çalışmada doğrular arasında ilişki kurulurken yalnızca (r, θ) parametrelerine bakıldığından aynı parametrelerle ifade edilebilen iki farklı duvarın tek bir duvar gibi algılanması mümkündür (Şekil 2.6). Bunun önlenmesi için duvarların görüldüğü noktalar arasındaki mesafeye bakılmış, bu mesafe belli bir değerden büyükse bunların ayrı iki duvar olduğu kabul edilmiştir.

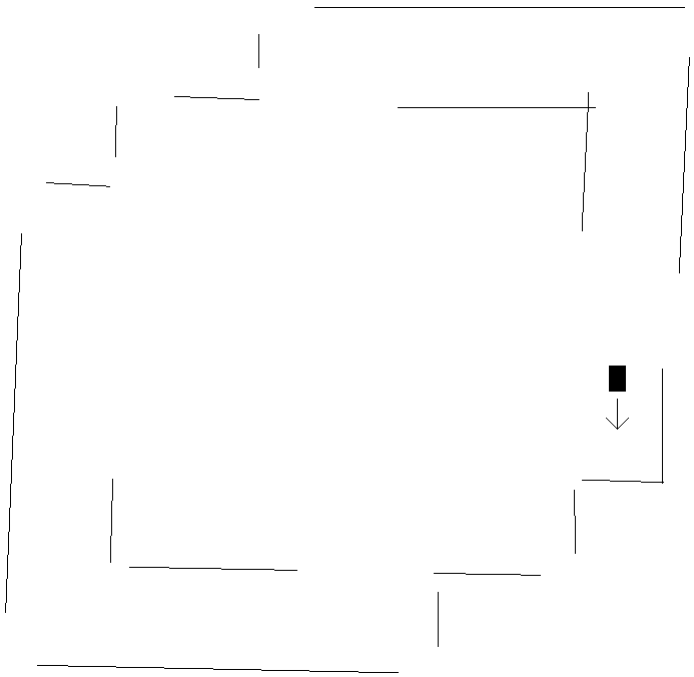


Şekil 2.6 Aynı doğru parametreleri ile ifade edildikleri için aynı duvar olarak ele alınabilecek farklı iki duvar

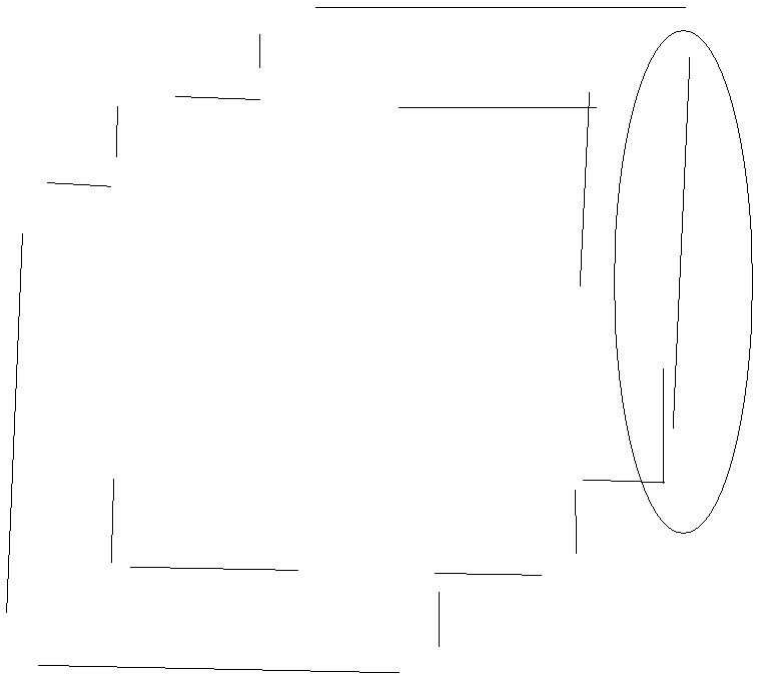
2.6 Döngü Kapatma

Döngü kapatma, dairesel ortamlarda robotun zaten daha önce bulunduğu bir noktaya döndüğünü anlaması problemidir.

Bu çalışmada, daha önce de belirtildiği gibi aynı parametrelere sahip farklı yer işaretçilerini ayırmak için uç noktalar kullanılmaktadır. Bu yaklaşım dairesel bir ortamda robot bir tur tamamlarken sorun çıkarmaktadır çünkü aynı duvarın farklı bir bölümünü gören robot uç noktalar arasındaki mesafe nedeniyle bunu farklı bir duvar olarak algılamaktadır. Şekil 2.7'de robotun başlangıç konumu içi dolu bir dikdörtgen ile ifade edilmiş, yönüne de yer verilmiştir. Haritalama başladığında robot solundaki duvarı görür ve bunu yeni bir yer işaretçisi olarak saklar. Ortamda bir tur attıktan sonra robot başlangıçta gördüğü duvarla benzer parametrelere sahip bir duvar görür fakat uç noktaları arasındaki mesafe nedeniyle bunu yeni bir yer işaretçisi olarak kabul eder. Robot ilerleyip de başlangıç noktasına döndüğünde bu iki yer işaretçisinin aslında aynı duvarı ifade ettiğini görürüz (Şekil 2.8).



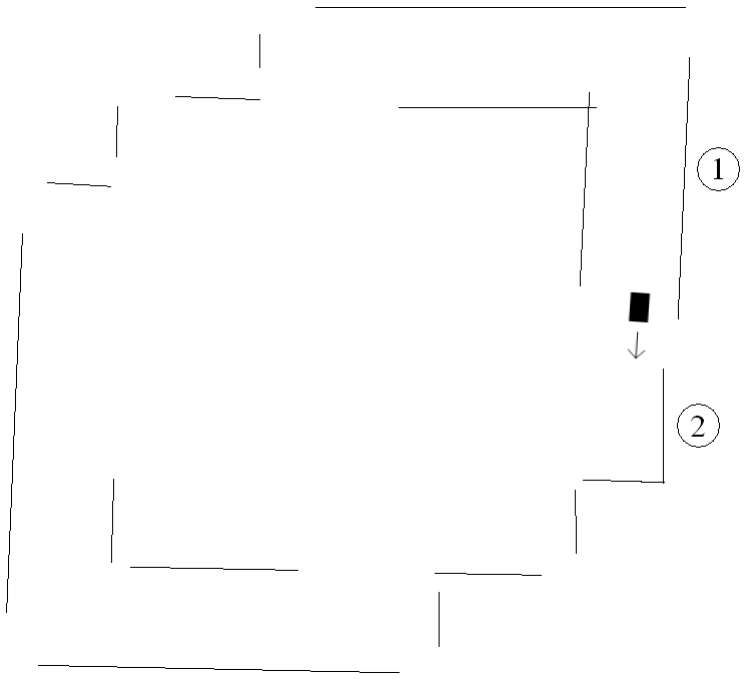
Şekil 2.7 Döngü kapatma olmaksızın haritalama



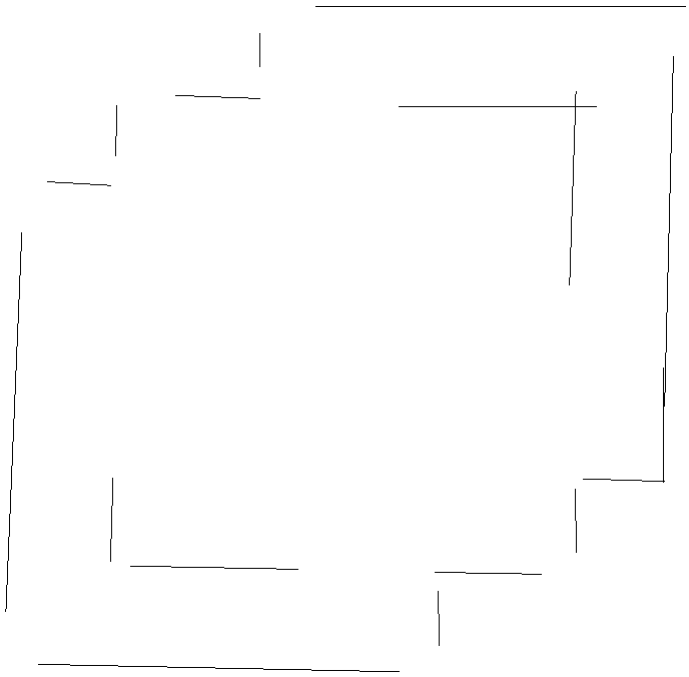
Şekil 2.8 Aynı duvara karşılık gelen iki yer işaretçisi

Döngü kapatırken iki duvarın ilişkilendirilmesi için, GKF'de olduğu gibi Mahalanobis mesafesi kullanılmıştır. İlişkili duvarları ararken tüm yer işaretçisi çiftlerini aramak tabi ki ek yük getirir. Buna çözüm olarak GKF'de gözlemlerle yer işaretçilerini eşleştirirken gözleme Mahalanobis mesafesi olarak en yakın bir değil iki yer işaretçisinin bulunması önerilmiştir

(Şekil 2.9). Son eklenen (1 numaralı) yer işaretçisi güncel inanç durumuna göre eklendiğinden gözlemle arasındaki mesafe minimumdur. Eğer robot döngü kapatma noktasına geldiyse ikinci en yakın yer işaretçisi ise başlangıçta eklenen ve son eklenen ile aynı duvara karşılık gelen (2 numaralı) yer işaretçisi olur. Gözleme en çok benzeyen iki yer işaretçisi bulunduktan sonra bunlar uç noktaları ve kendi aralarındaki Mahalanobis mesafesi kullanılarak karşılaştırılır. Eğer bunlar birbirlerine yeterince yakınlarsa bunların aynı duvarı temsil ettikleri anlaşılır. Bu noktada bu iki yer işaretçisi doğrudan birleştirilmemektedir çünkü bunların birleştirilmesinin önceki yer işaretçilerinin konumunun düzeltilmesi yönünde bir etkisi yoktur. Bunun yerine daha güvenilir olan yer işaretçisi tüm durumu güncellemek için kullanılır: Gözlem yok sayılır, 2 numaralı yer işaretçisi gözlem olarak, 1 numaralı yer işaretçisi ise bununla ilişkili yer işaretçisi olarak kullanılır. 2 numaralı yer işaretçisinin gözlem olarak seçilmesinin ve bir düzeltme faktörü olarak kullanılmasının nedeni bu yer işaretçisinin daha önce, konumla ilgili belirsizlik daha azken eklenmiş olmasıdır. Şekil 2.10’da döngü kapatma işlemi sonucunda yer işaretçilerinin yeni durumu görülmektedir.



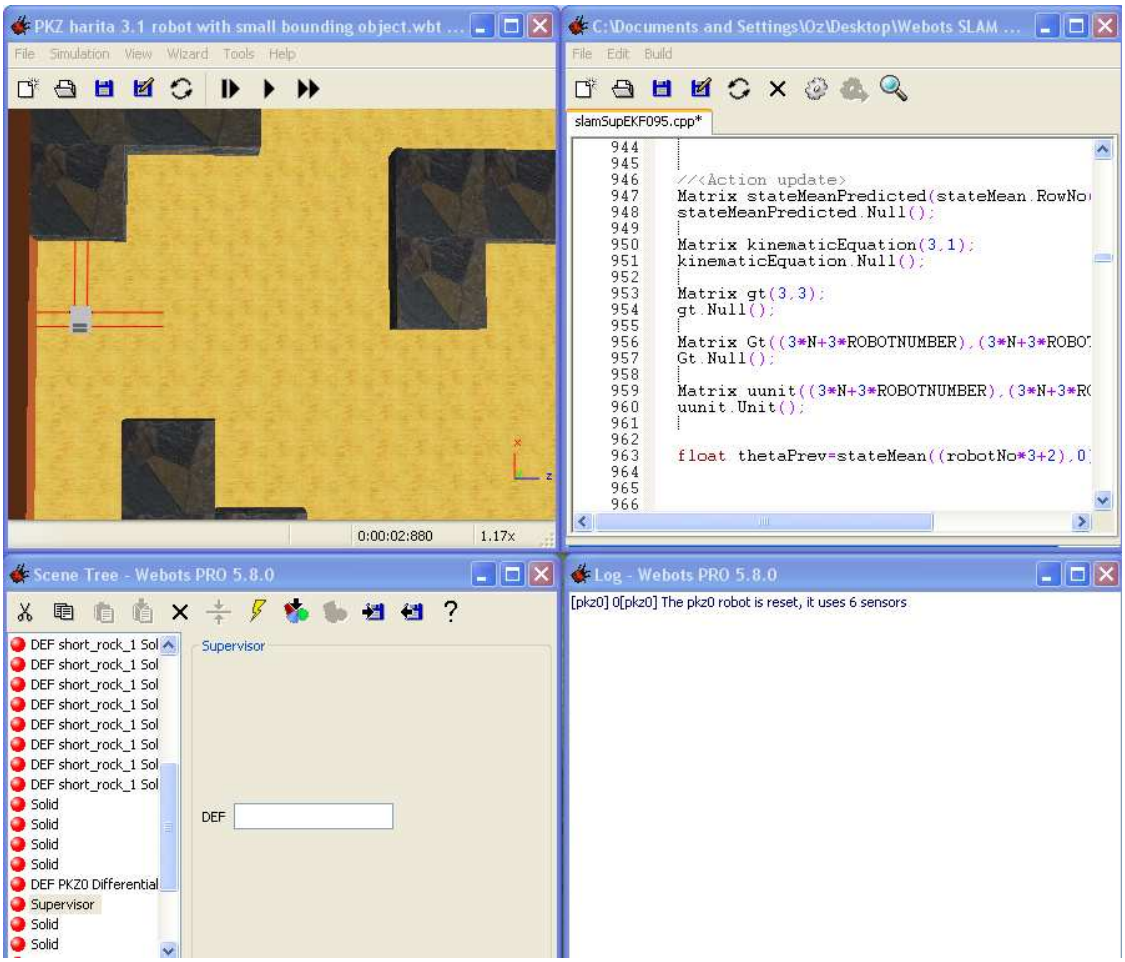
Şekil 2.9 Döngü kapatma işlemi sırasında ilişkili oldukları bulunan iki yer işaretçisi



Şekil 2.10 Döngü kapatma işlemi sonrası yer işaretçilerinin durumu

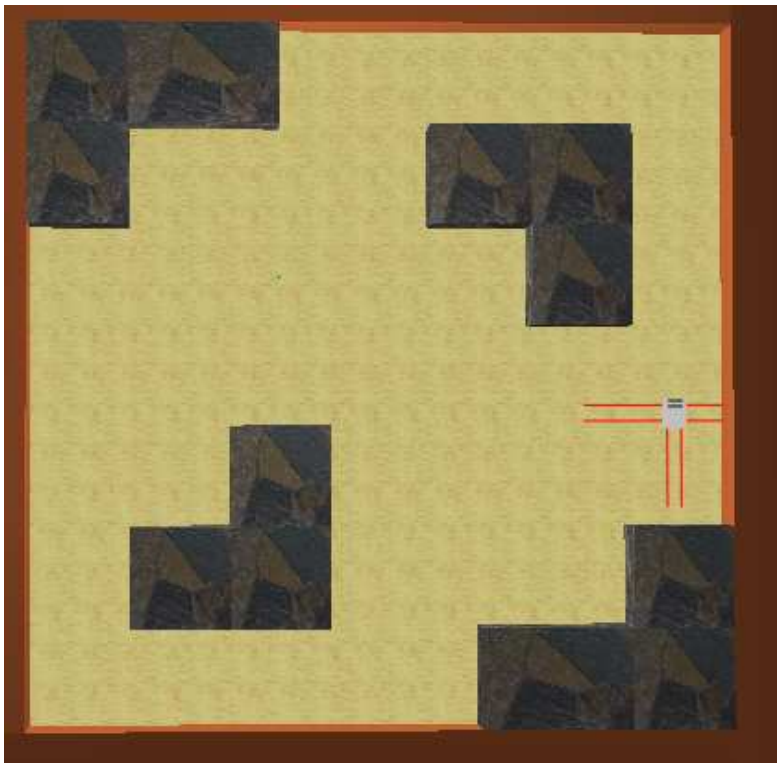
2.7 Webots Simülasyon Ortamı, Uygulama ve Yapılan Deneylerin Detayları

Bu çalışmada deneyler Cyberbotics firması tarafından geliştirilen Webots simülasyon ortamında gerçekleştirilmiştir. Webots, kolay bir arayüz aracılığıyla farklı robot modellerinin oluşturulabildiği, programlanabildiği ve çalıştırılabildiği bir simülasyon ortamıdır (Şekil 2.11). Pek çok algılayıcı, eylemci ve haberleşme cihazı modelini barındıran kapsamlı bir kütüphanesi bulunmaktadır. Ortama çeşitli şekillerde nesnelerin eklenebildiği Webots'ta, her nesnenin renk, doku, sürtünme katsayısı gibi özellikleri değiştirilebilmektedir. Ortamdaki her bir bileşen, Scene Tree'deki bir düğüm aracılığıyla yönetilmektedir. Her temel düğümün alt düğümleri aracılığıyla da bileşenlerin alt bileşenlerini ve özelliklerini kontrol etmek mümkündür. Webots'un dahili metin düzenleyicisi ve derleyicisi aracılığıyla robotları, C, C++, Java, Python, URBI ve Matlab dillerinde programlamak ve yürütmek mümkün olmaktadır. Ayrıca e-puck, Nao, Katana, Hoap-2 gibi robotlara doğrudan kontrol kodu aktarılması mümkündür.

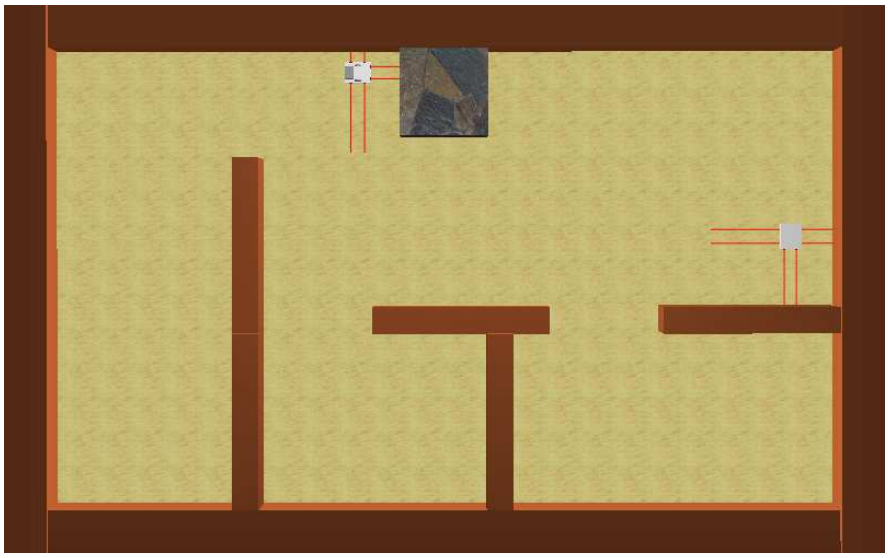


Şekil 2.11 Webots Simülasyon ortamı

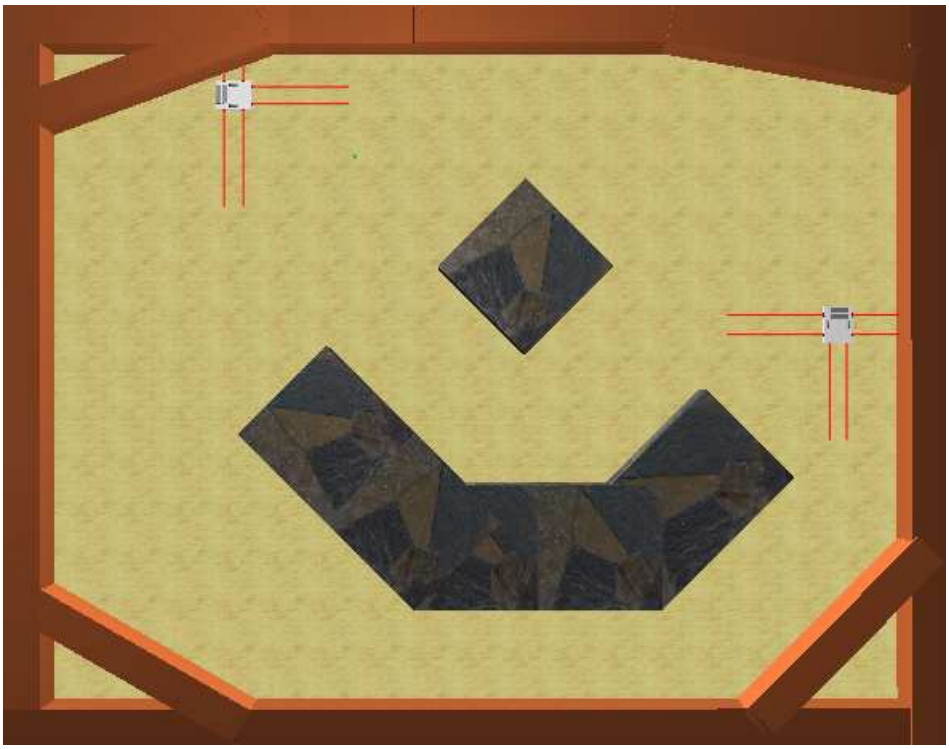
Deneylerin büyük çoğunluğu Webots'ta oluşturulan iki farklı ortamda gerçekleştirilmiştir. İlk ortam çok sayıda iç ve dış köşeden oluşan, "ada" şeklinde ifade edilebilecek herhangi bir duvara dayanmayan, ortada bulunan nesnelerin de yer aldığı karesel bir alandır (Şekil 2.12). İkinci ortam, ofis ortamına daha fazla benzeyen, oda ve koridorlardan oluşan bir ortamdır (Şekil 2.13). Uygulamanın birbirleriyle dik açı yapmayan duvarlar barındıran bir ortamda da başarılı olup olmayacağını test etmek için üçüncü ortam (Şekil 2.14) oluşturulmuştur, bu ortamda duvarlar birbirleriyle 10, 20, 30 ve 45 derecelik açılar yapmaktadırlar.



Şekil 2.12 Çok sayıda iç ve dış köşeden oluşan Ortam 1’in görüntüsü

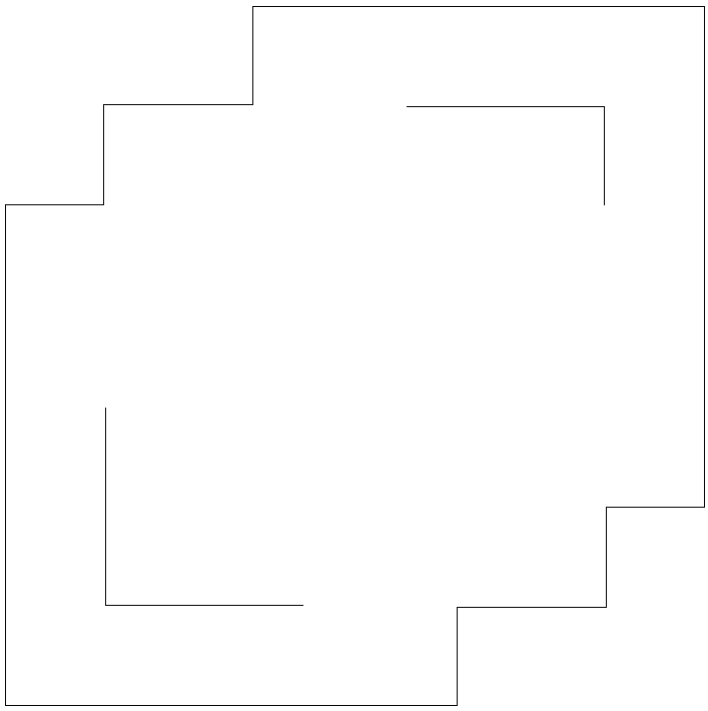


Şekil 2.13 Oda ve koridorlardan oluşan Ortam 2’nin görüntüsü

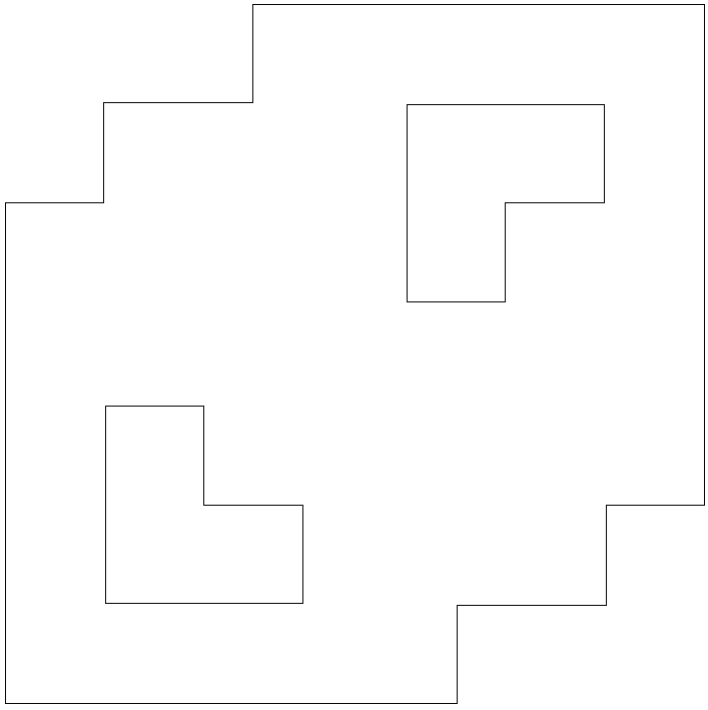


Şekil 2.14 Birbirleriyle 10, 20, 30 ve 45 derecelik açılar yapan duvarlar içeren Ortam 3'ün görüntüsü

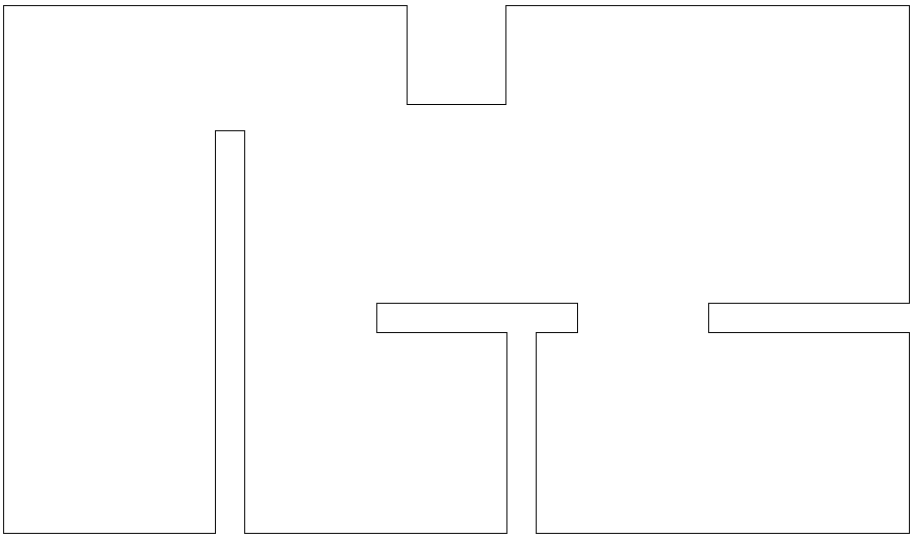
Bu ortamların çıkarılması beklenen haritaları aşağıda verilmiştir. Şekil 2.15'te Ortam 1'in duvar takibi ile çıkarılması beklenen haritası görülmektedir. Duvar takibinde robot sadece başlangıçta gördüğü duvarı takip ederek ilerlediği için ortadaki cisimlerin her tarafının görünmesi mümkün olmamaktadır, bu nedenle ortadaki cisimlerin tam olarak haritalanması beklenmemektedir. Şekil 2.16'da Ortam 1'in robot dolaşımı için keşif stratejisi kullanıldığında çıkarılması beklenen haritası görülmektedir. Burada robot her yeri gezeceğinden adaların da tam olarak haritada yer alması beklenmektedir. Şekil 2.17'de Ortam 2'de dolaşan robotun çıkarması beklenen harita görülmektedir. Ortam 2'de ada durumunda cisim yer almadığından burada duvar takibi ya da keşif stratejisi uygulanması, çıkarılması beklenen haritayı değiştirmemektedir. Şekil 2.18 ve 2.19'da da Ortam 3'te dolaşan robotun duvar takibi ile ve keşif stratejisi ile dolaşım yaptığında çıkaracağı haritalar görülmektedir.



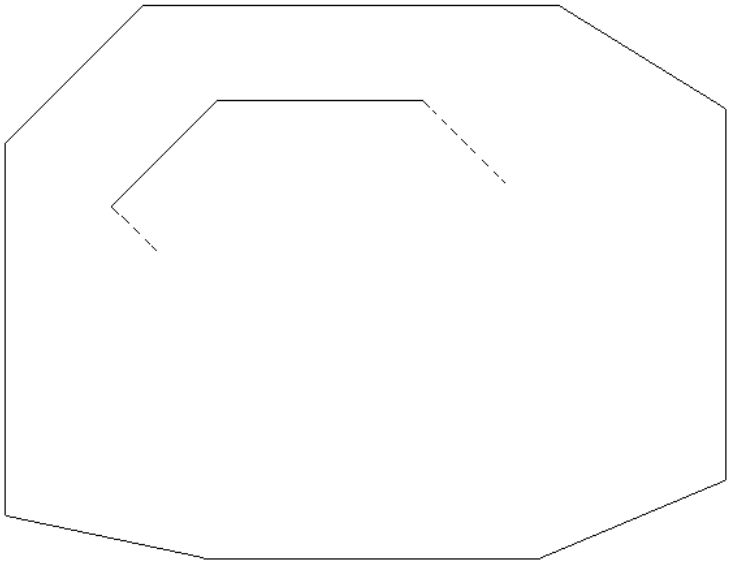
Şekil 2.15 Ortam 1 için duvar takibi ile çıkarılması beklenen harita



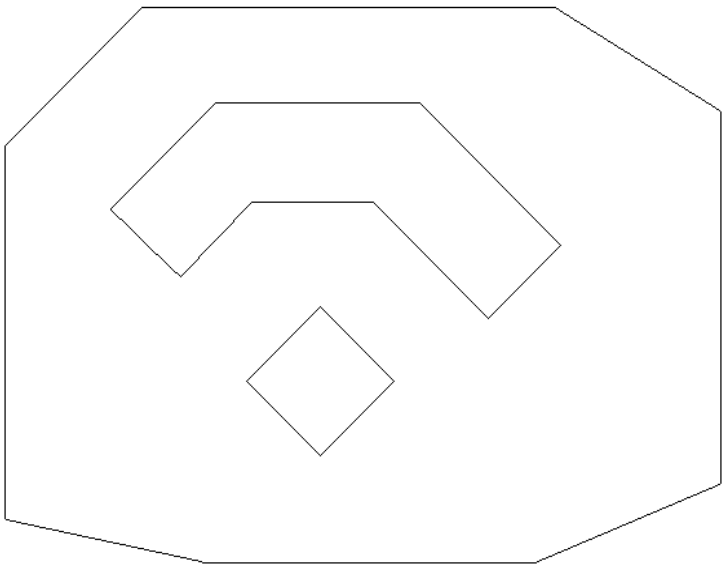
Şekil 2.16 Ortam 1 için keşif stratejisi ile çıkarılması beklenen harita



Şekil 2.17 Ortam 2 için beklenen harita



Şekil 2.18 Ortam 3 için duvar takibi ile çıkarılması beklenen harita

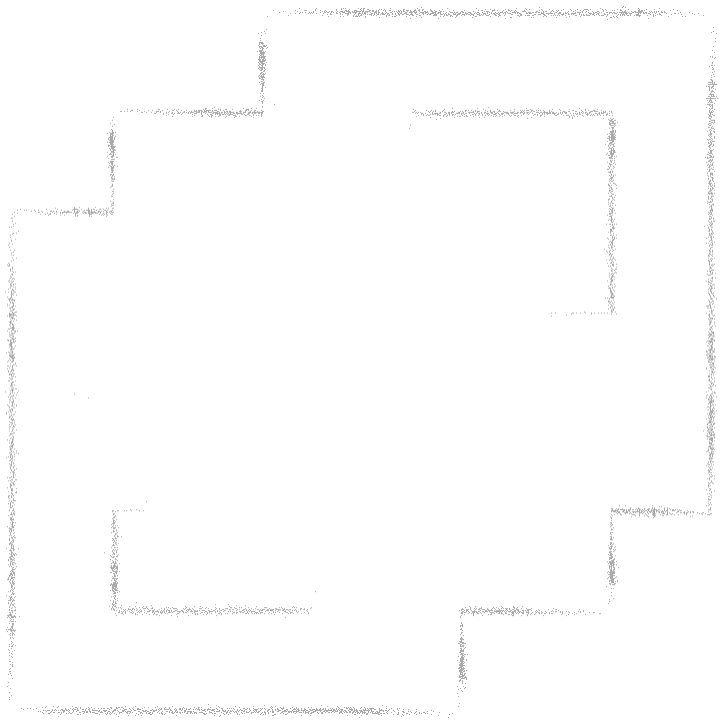


Şekil 2.19 Ortam 3 için keşif stratejisi ile çıkarılması beklenen harita

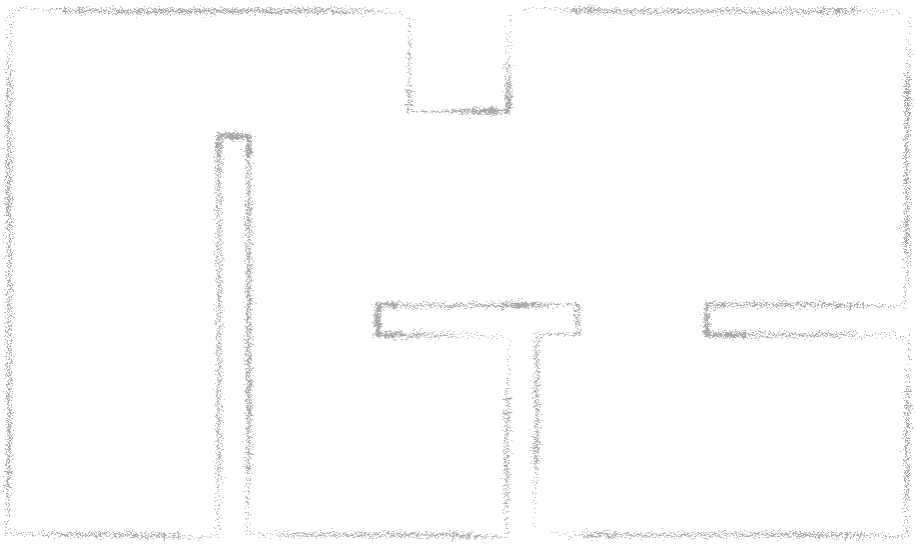
Bu ortamlarda yapılan denemelerde, gürültü bulunmadığında, algılayıcı gürültüsü eklendiğinde ve hareket gürültüsü eklendiğinde çıkarılan haritalara bakılmış, sistem başarısı gözlenmiştir. Denemelerde algılayıcı gürültüsünü gerçeklemek için robotun algılayıcı ölçümlerine Box-Muller dönüşümü kullanılarak oluşturulan Gauss gürültüsü eklenmiştir. Hareket gürültüsü ise Webots'ta robot özelliklerinde yer alan ve robotun iki tekerleğine ayrı ayrı beyaz gürültü ekleyen slipNoise parametresinde değişiklik yapılarak sağlanmıştır.

Burada ortamların farklı gürültü seviyelerinde çıkarılmış ham haritalarına yer verilmiştir. Ham harita ile kastedilen, robot dolaşırken GKF kullanılmadan algılayıcı verilerinin doğrudan basılmasıdır. Farklı gürültü seviyeleri için ham haritalar verildiği gibi farklı dolaşım stratejileri kullanıldığında elde edilen ham haritalara da yer verilmiştir çünkü farklı stratejilerde robotun ortam içindeki dolaşım miktarı değişmektedir ve harita üzerinde biriken gürültülü veri miktarı değişmektedir, bu durumun gösterilmesi amaçlanmıştır.

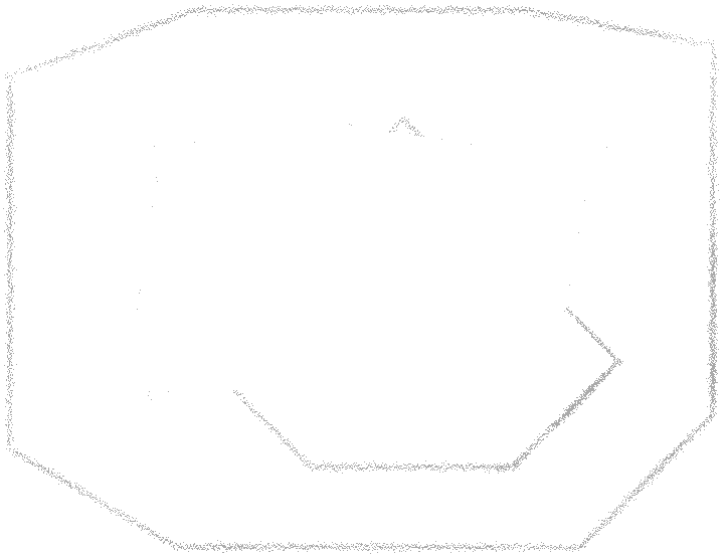
Şekil 2.20, 2.21 ve 2.22'de üç ortamda standart sapması iki santimetre olan algılayıcı gürültüsü varken duvar takibi ile dolaşıldığında çıkarıla haritalar yer almaktadır. Şekil 2.23, 2.24 ve 2.25'te aynı gürültü varken keşif stratejisi uygulandığında çıkarılan haritalar görülebilir. Şekil 2.26, 2.27, 2.28, 2.29, 2.30 ve 2.31'de de standart sapması 4 santimetre olan algılayıcı gürültüsü varken üç ortamın duvar takibi ve keşif stratejisi ile çıkarılan ham haritaları yer almaktadır.



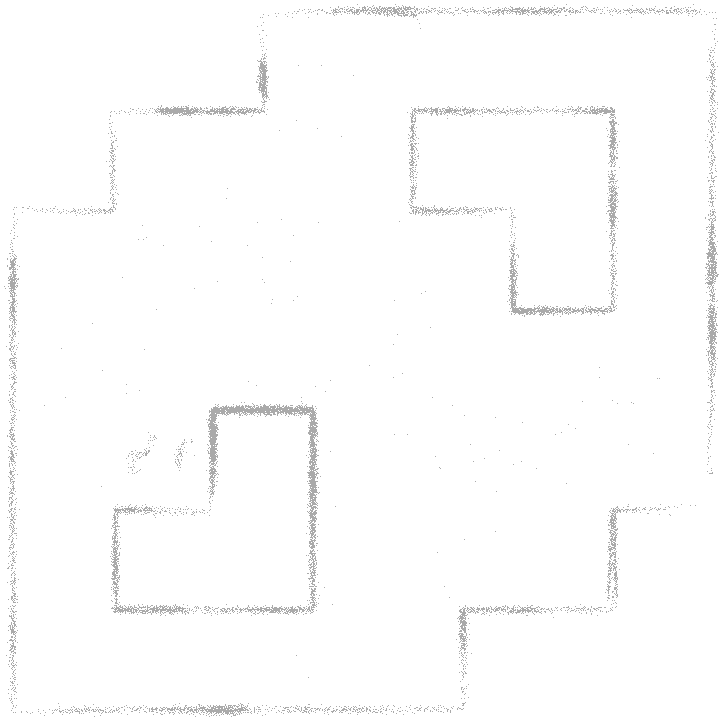
Şekil 2.20 Ortam 1’de, standart sapması 2 cm olan algılayıcı gürültüsü varken duvar takibi ile elde edilen ham harita



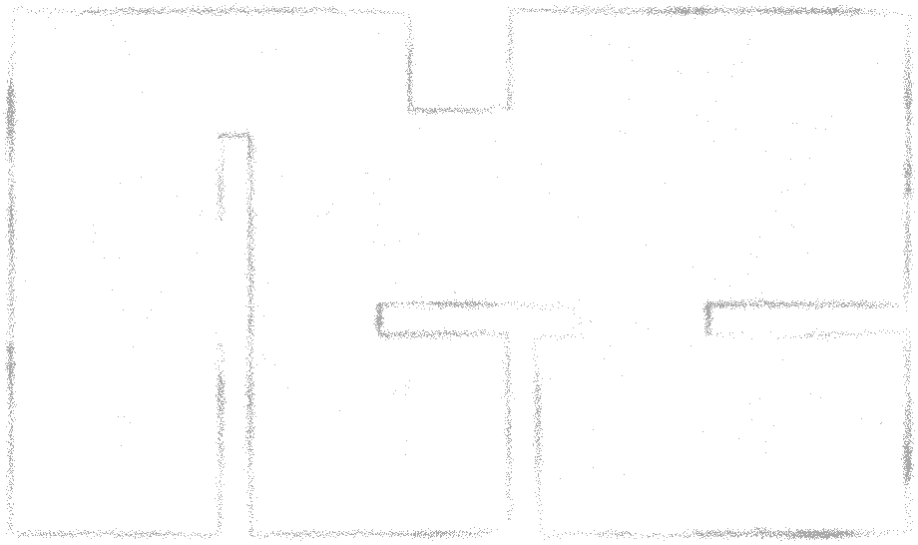
Şekil 2.21 Ortam 2’de, standart sapması 2 cm olan algılayıcı gürültüsü varken duvar takibi ile elde edilen ham harita



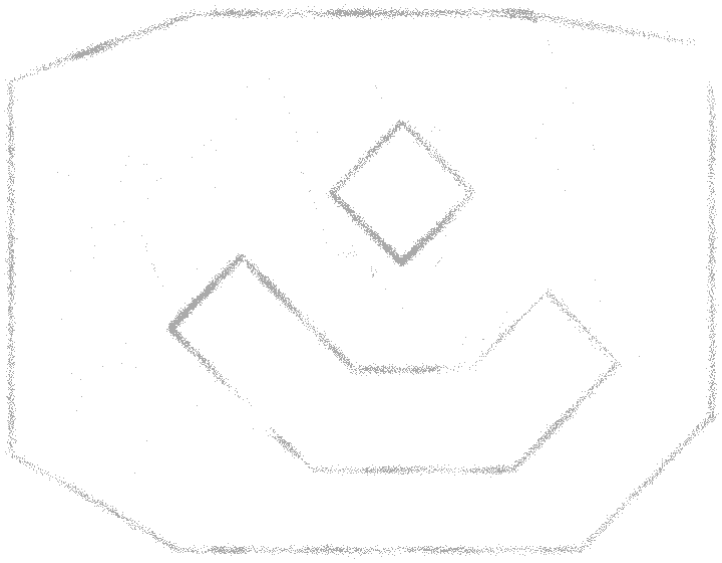
Şekil 2.22 Ortam 3’te, standart sapması 2 cm olan algılayıcı gürültüsü varken duvar takibi ile elde edilen ham harita



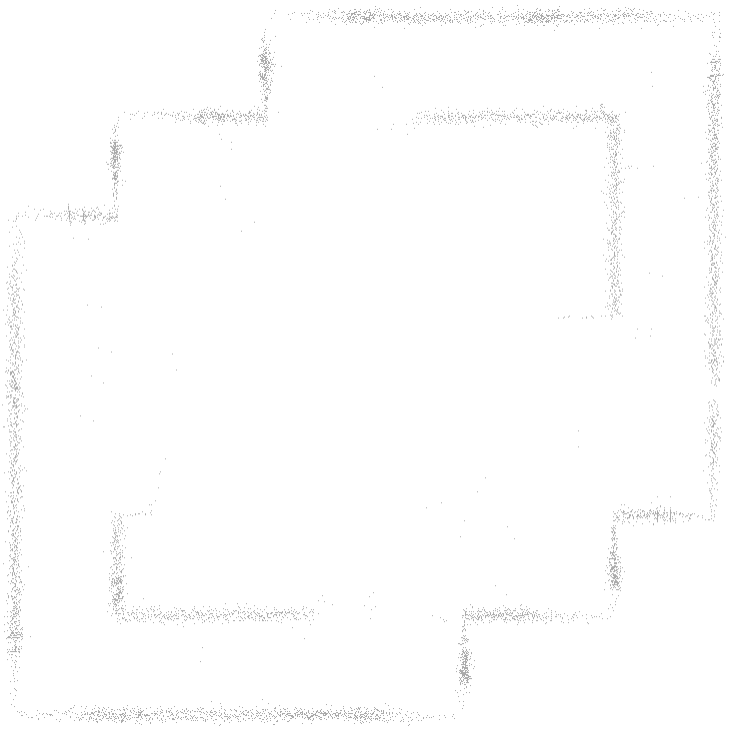
Şekil 2.23 Ortam 1’de, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi ile elde edilen ham harita



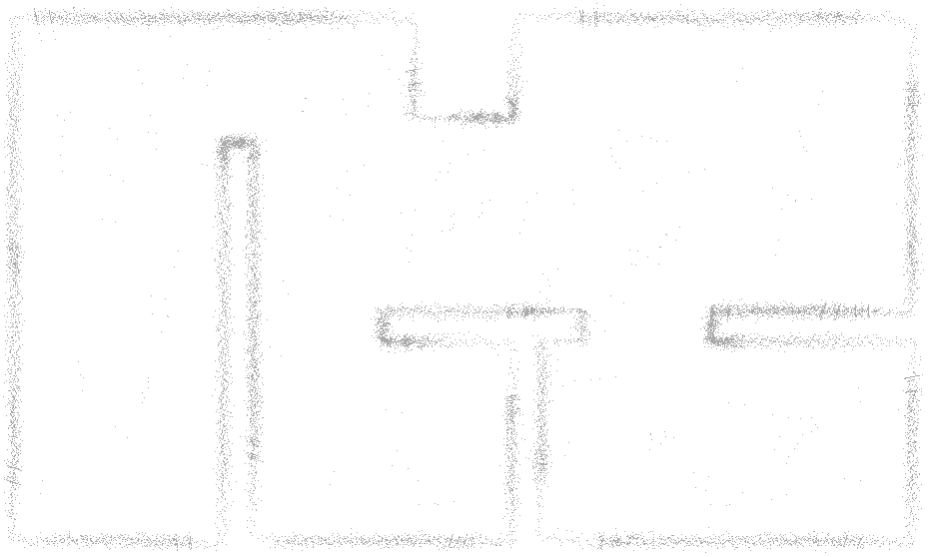
Şekil 2.24 Ortam 2’de, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi ile elde edilen ham harita



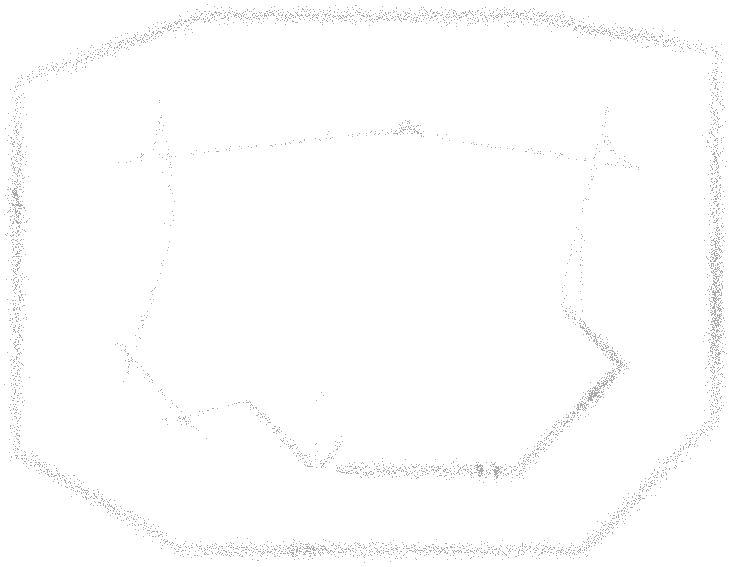
Şekil 2.25 Ortam 3’te, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi ile elde edilen ham harita



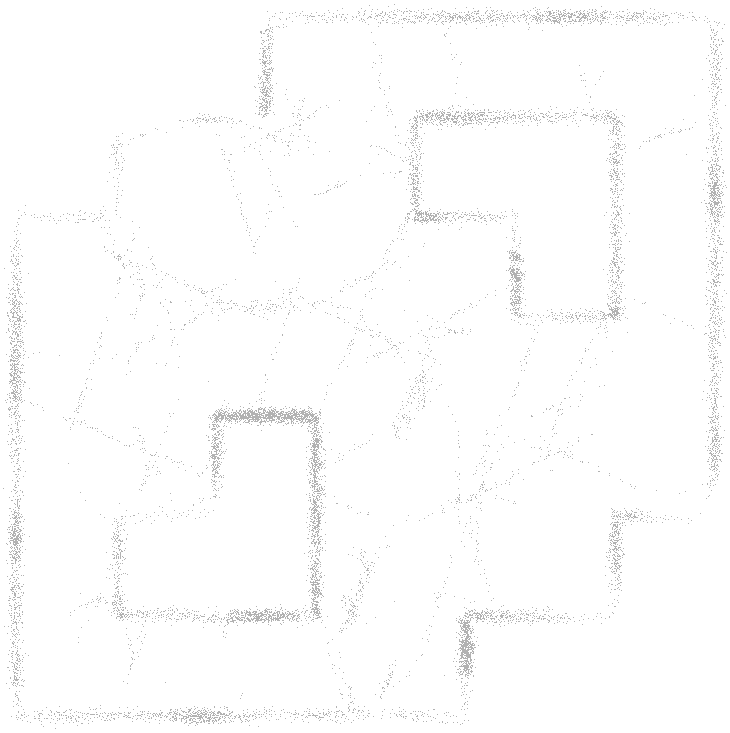
Şekil 2.26 Ortam 1’de, standart sapması 4 cm olan algılayıcı gürültüsü varken duvar takibi ile elde edilen ham harita



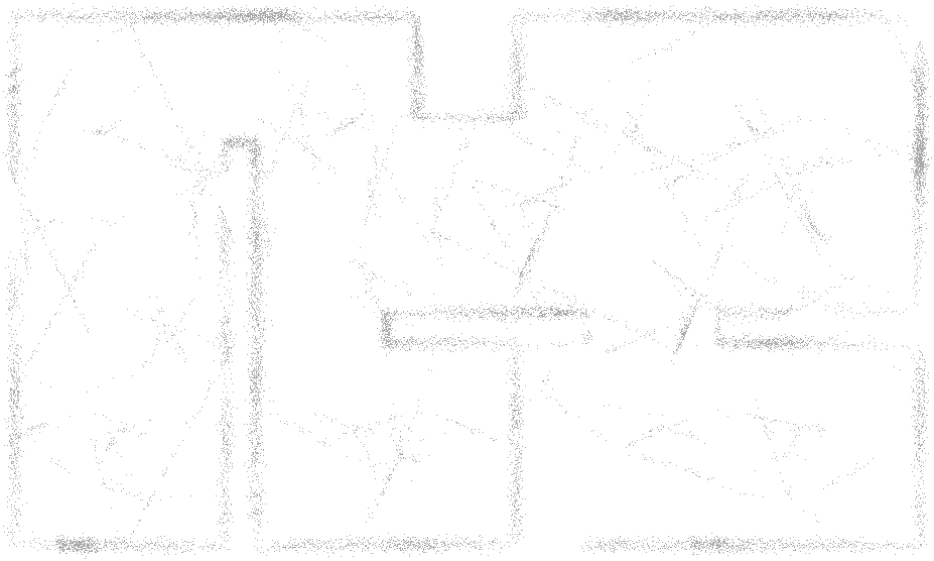
Şekil 2.27 Ortam 2’de, standart sapması 4 cm olan algılayıcı gürültüsü varken duvar takibi ile elde edilen ham harita



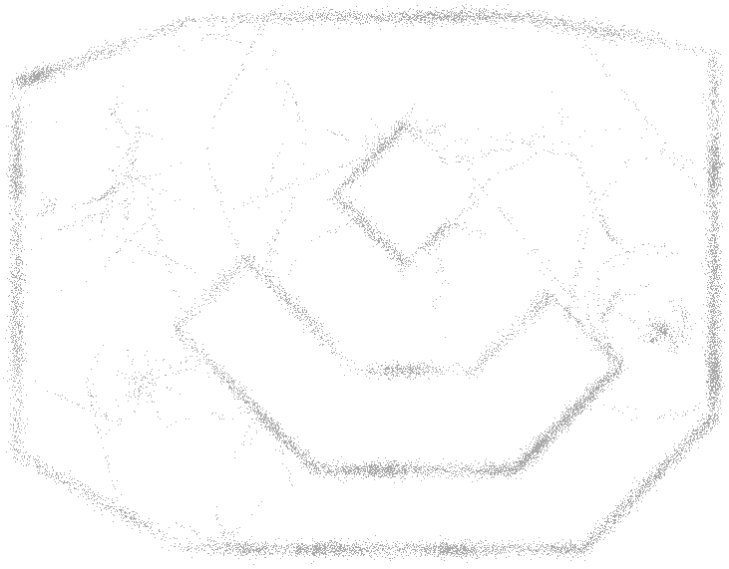
Şekil 2.28 Ortam 3’te, standart sapması 4 cm olan algılayıcı gürültüsü varken duvar takibi ile elde edilen ham harita



Şekil 2.29 Ortam 1’de, standart sapması 4 cm olan algılayıcı gürültüsü varken keşif stratejisi ile elde edilen ham harita

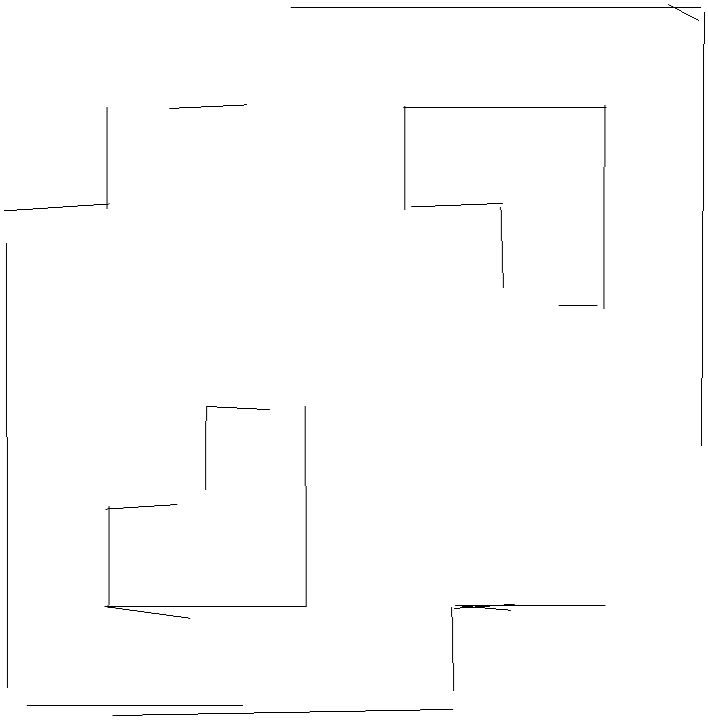


Şekil 2.30 Ortam 2’de, standart sapması 4 cm olan algılayıcı gürültüsü varken keşif stratejisi ile elde edilen ham harita

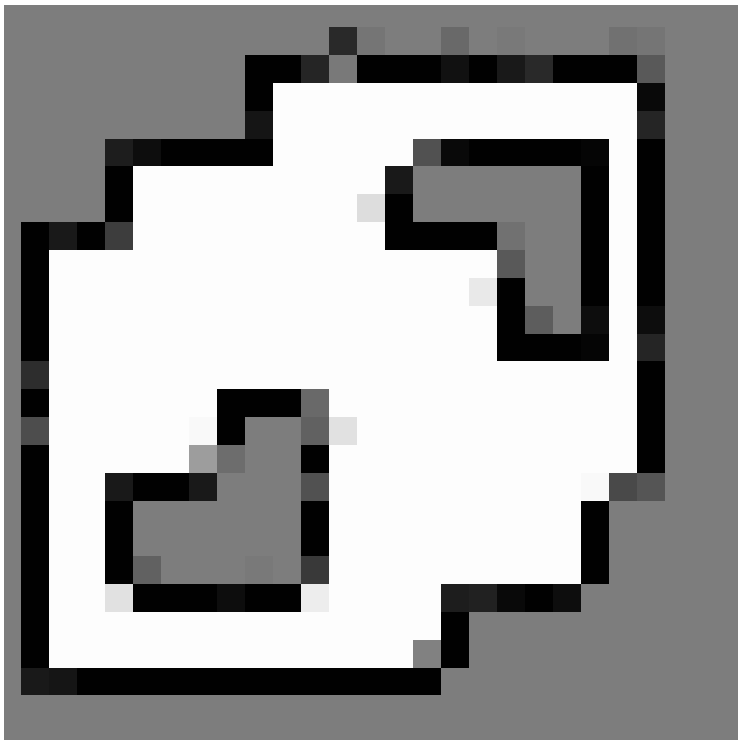


Şekil 2.31 Ortam 3’te, standart sapması 4 cm olan algılayıcı gürültüsü varken keşif stratejisi ile elde edilen ham harita

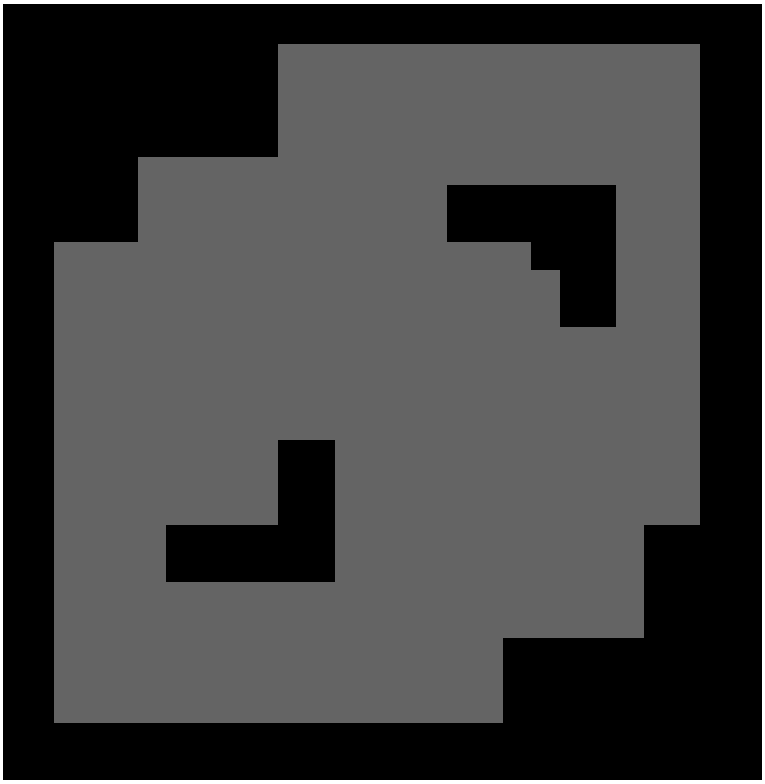
Uygulamada harita çizdirme işi kullanıcı talebi ile yapılmaktadır, herhangi bir anda PGM formatında harita oluşturulabilmektedir. Keşif stratejisi kullanıldığında, keşfedilmemiş gidilebilecek başka bir alan kalmadığında GKF algoritmasını durduran ve haritayı çizdiren bir mekanizma da bulunmaktadır. Duvar takibinde sadece uç noktaları ve (r,θ) parametreleri kullanılarak çizilen doğrulardan oluşan harita (Şekil 2.32) çıktı olarak verilirken keşif stratejisi kullanıldığında buna ek olarak işgal edilmiş alan haritası (Şekil 2.33) ile keşfedilmiş yerleri gösteren bir harita da (Şekil 2.34) çıktı olarak verilmektedir.



Şekil 2.32 Örnek harita



Şekil 2.33 Örnek işgal edilmiş alan haritası



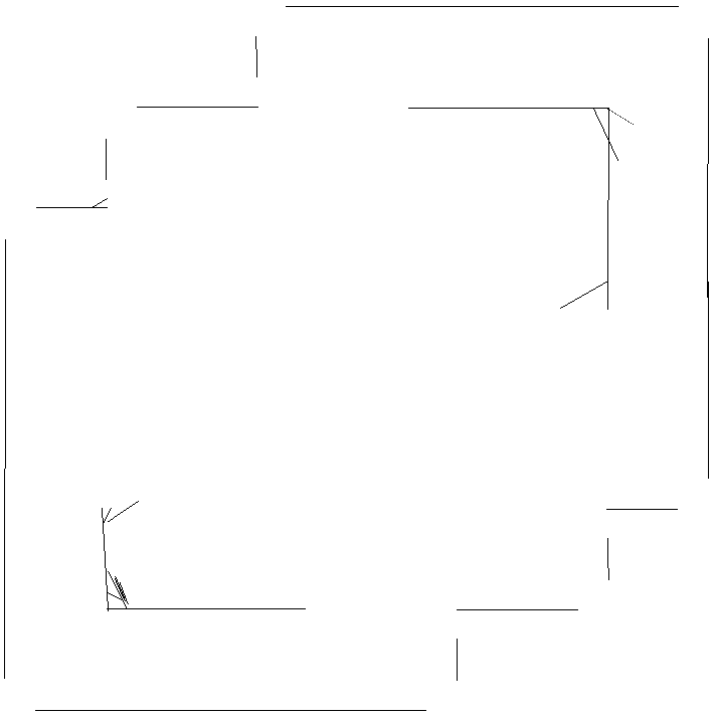
Şekil 2.34 Örnek keşif haritası

3. TEK ROBOTLA İKİ NOKTADAN DOĞRU ÇIKARIMI YAPARAK HARİTALAMA

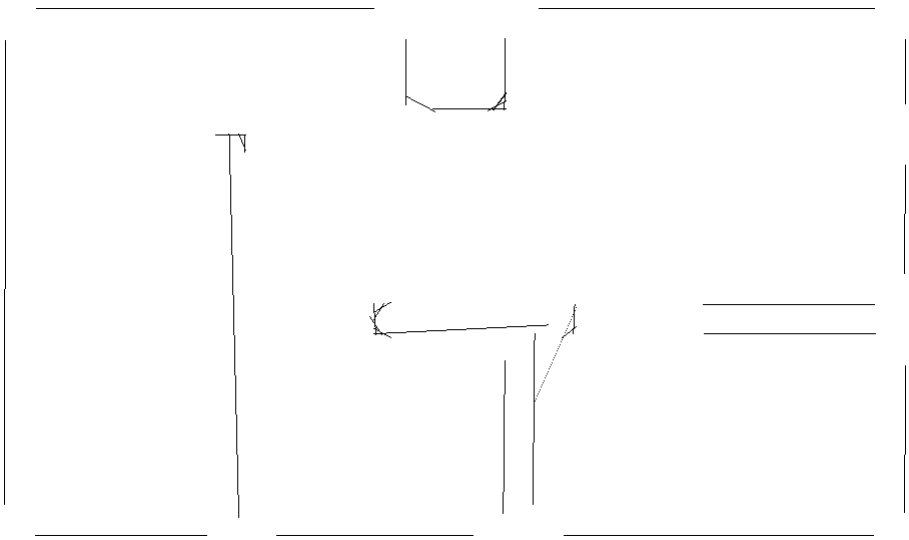
Bu çalışmanın nihai hedefine ulaşmak amacıyla ilerlerken yapılan denemelerden biri tek robotla iki noktadan doğru çıkarımı yaparak haritalama olmuştur. Burada, tüm çalışmada olduğu gibi, haritalama algoritması olarak Genişletilmiş Kalman Filtresi, özellik olarak da doğrular kullanılmıştır.

3.1 Deneysel Sonuçlar

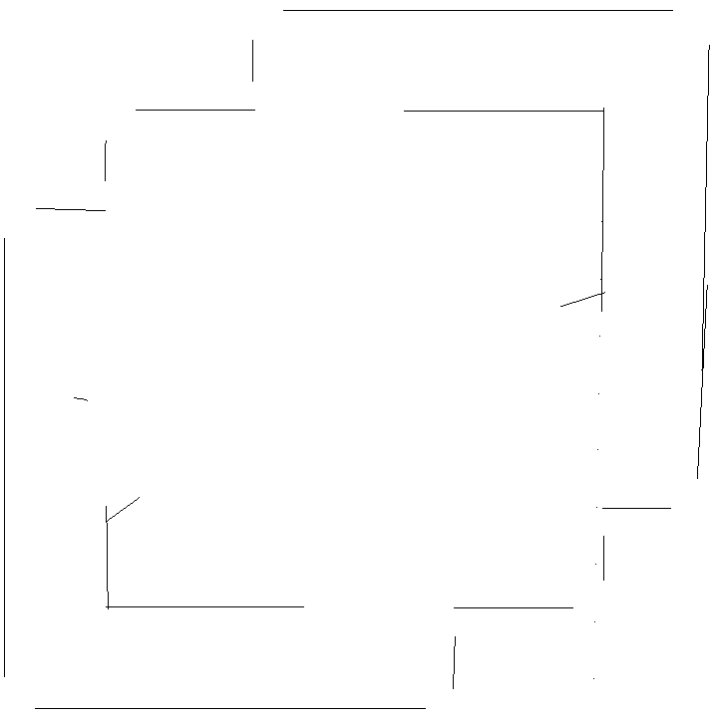
Simülasyon ortamında farklı haritalarla pek çok deneme yapılmıştır. Burada bu denemelerin bir kısmına ve elde edilen sonuçlara yer verilmiştir.



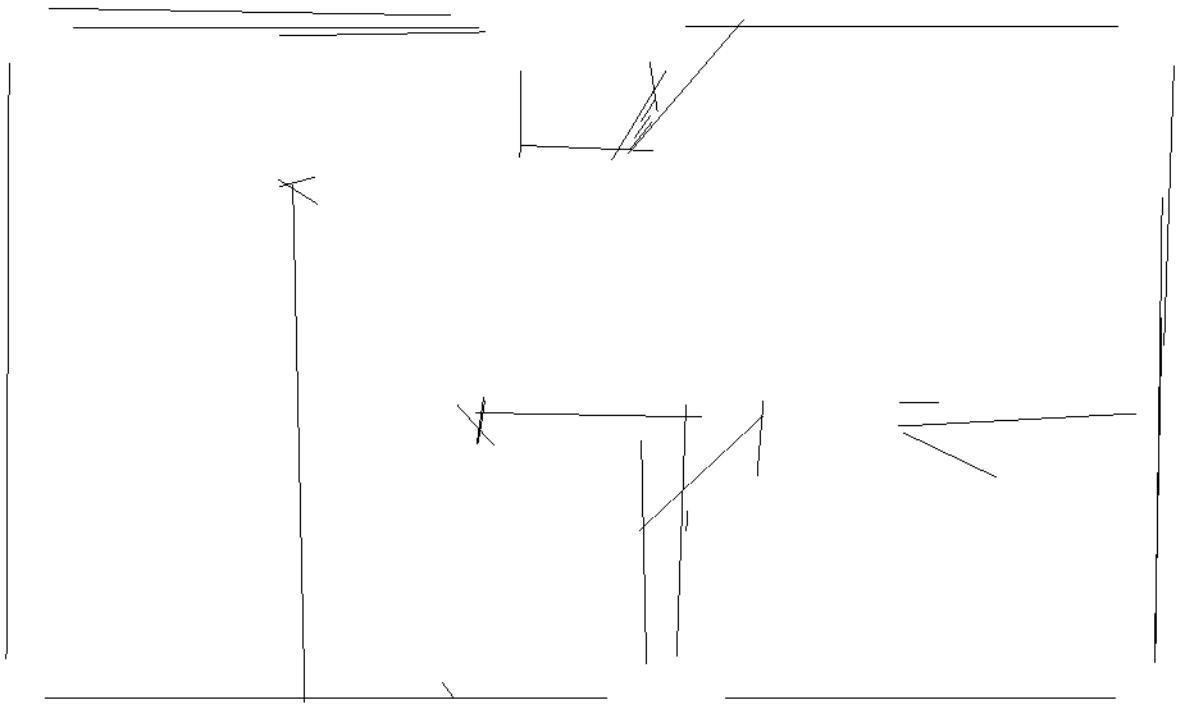
Şekil 3.1 Ortam 1’de, gürültü yokken duvar takibi ile elde edilen harita



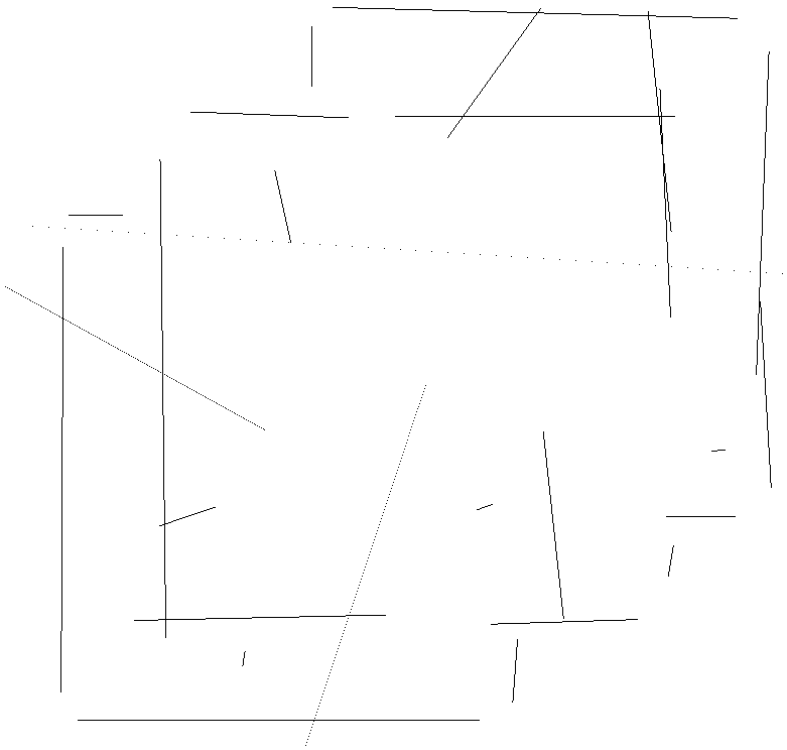
Şekil 3.2 Ortam 2’de, gürültü yokken duvar takibi ile elde edilen harita



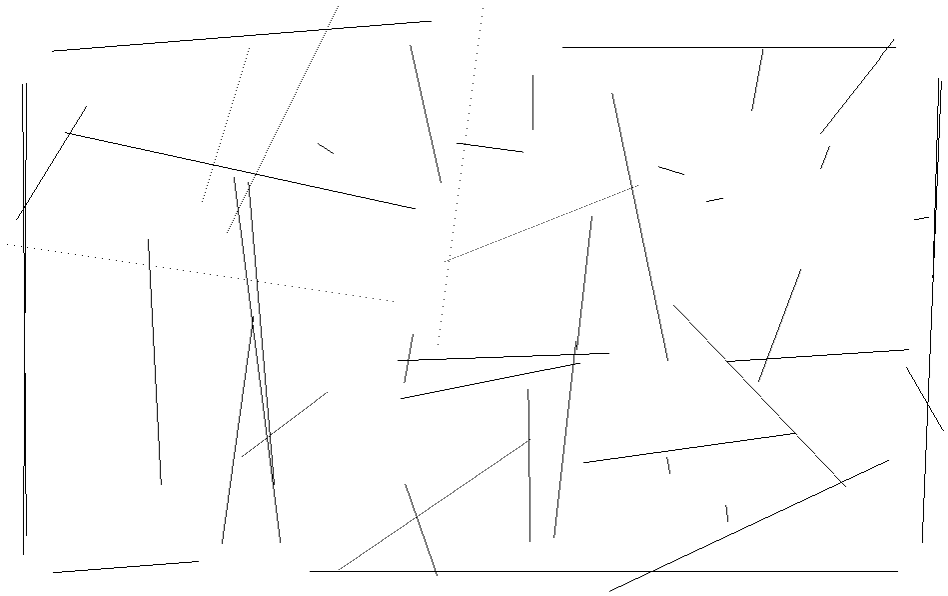
Şekil 3.3 Ortam 1’de, standart sapması 2 cm olan algılayıcı gürültüsü varken duvar takibi ile çıkarılan harita



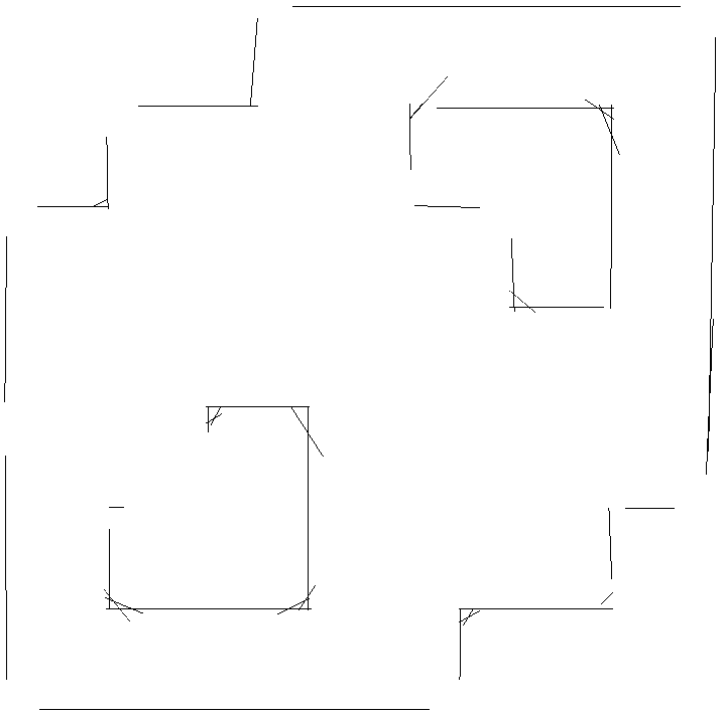
Şekil 3.4 Ortam 2’de, standart sapması 2 cm olan algılayıcı gürültüsü varken duvar takibi ile çıkarılan harita



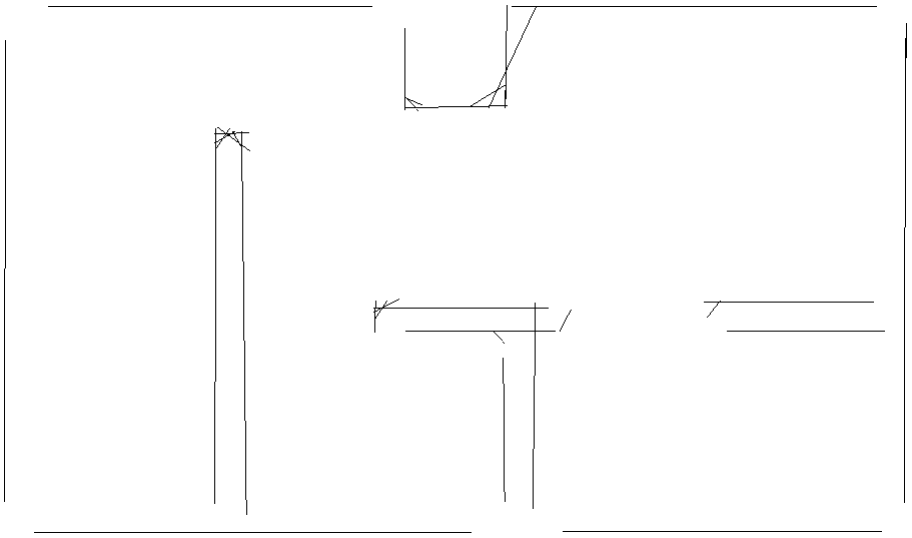
Şekil 3.5 Ortam 1’de, standart sapması 4 cm olan algılayıcı gürültüsü varken duvar takibi ile çıkarılan harita



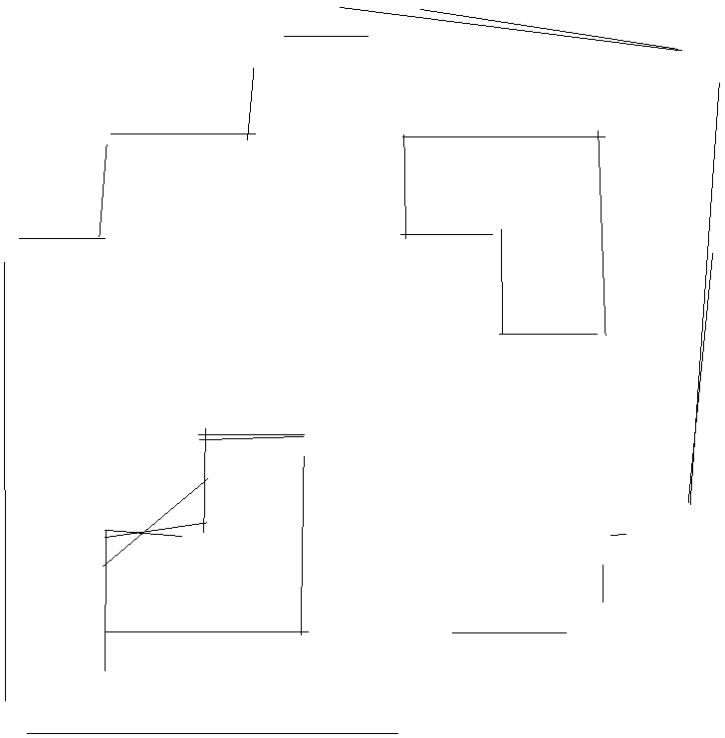
Şekil 3.6 Ortam 2’de, standart sapması 4 cm olan algılayıcı gürültüsü varken duvar takibi ile çıkarılan harita



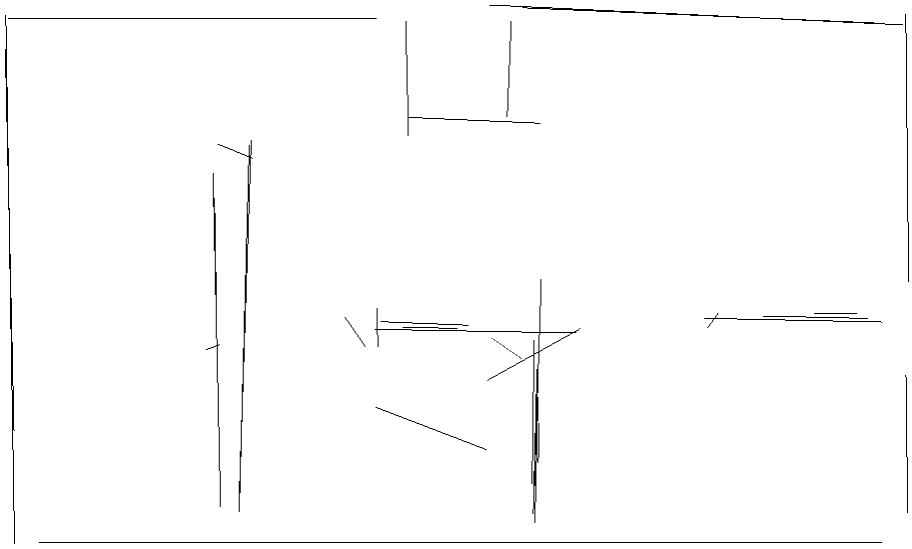
Şekil 3.7 Ortam 1’de, gürültü yokken keşif stratejisi kullanıldığında çıkarılan harita



Şekil 3.8 Ortam 2’de, gürültü yokken keşif stratejisi kullanıldığında çıkarılan harita



Şekil 3.9 Ortam 1’de, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında çıkarılan harita



Şekil 3.10 Ortam 2’de, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında çıkarılan harita

3.2 Sonuçlar

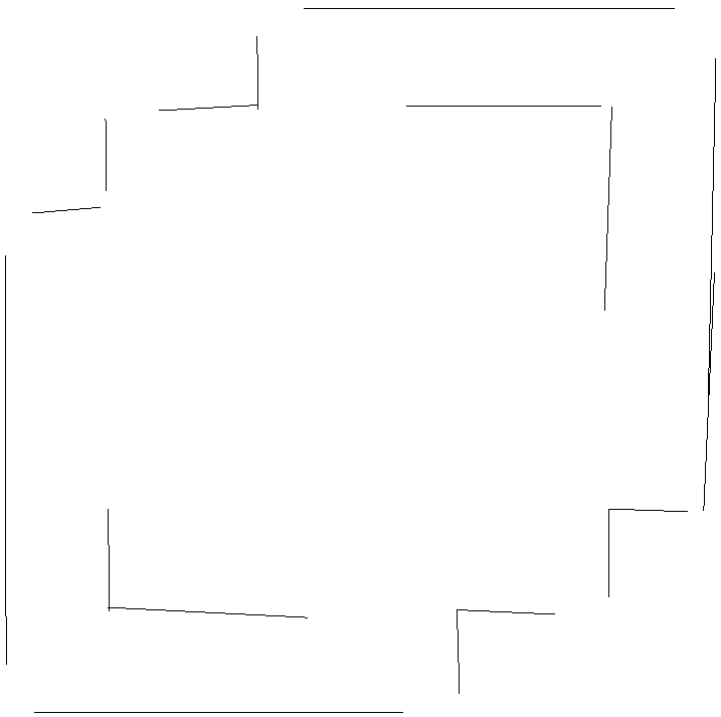
İki noktadan doğru çıkarımı ile yapılan haritalamanın, bir robotun basit, gürültüsüz bir ortamda duvar izleyerek yaptığı bir haritalama işleminde işe yarayabileceği görülmüştür fakat genel bir çözüm olarak uygun değildir. Köşe noktalarında karşılaşılan farklı duvarlardan okuma alma sorunu ciddi bir sorundur ve köşe noktalarının tespiti her durumda mümkün olmayacağından farklı bir çözüm gerekmektedir. Sistemin gürültüye toleransı da çok düşüktür, algılayıcı gürültüsü olduğunda pek çok hatalı yer işaretçisi görülmektedir. Ortamda hareket gürültüsü olduğunda da sistem köşelerden alınan yanlış okumalardan ciddi şekilde etkilenmekte ve robot konumunu tamamen şaşırmakta, çok başarısız haritalar çıkarılmaktadır.

4. TEK ROBOTLA HOUGH DÖNÜŞÜMÜ İLE DOĞRU ÇIKARIMI YAPARAK HARİTALAMA

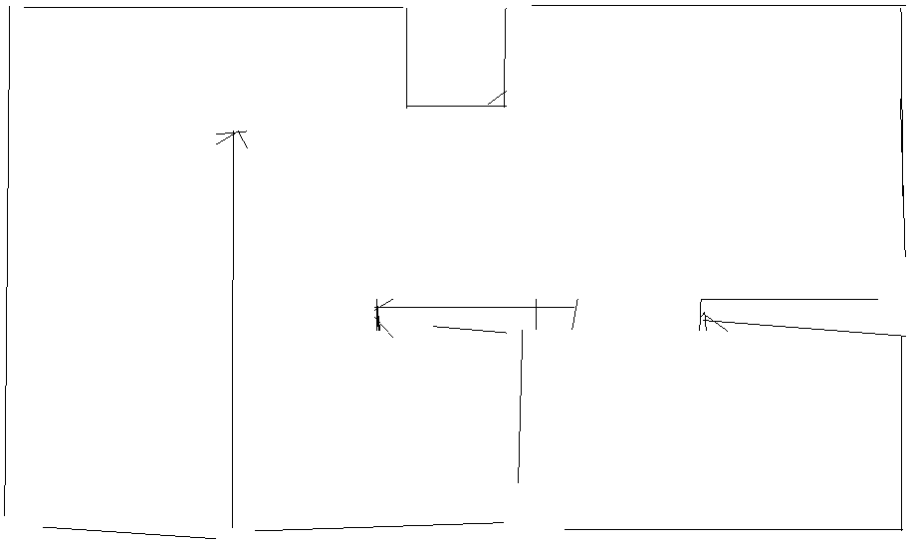
Çalışmanın bu aşamasında, yan algılayıcılar aracılığıyla toplanan noktasal verilere Hough dönüşümü uygulanarak doğrusal özellikler çıkarılmış, bu özellikler kullanılarak Genişletilmiş Kalman Filtresi ile EZKBH problemine çözüm getirilmesi amaçlanmıştır. Hough dönüşümünde maksimum oyu alan değil, belli bir eşik değeri geçen tüm (r,θ) ikilileri alınmıştır.

4.1 Deneysel Sonuçlar

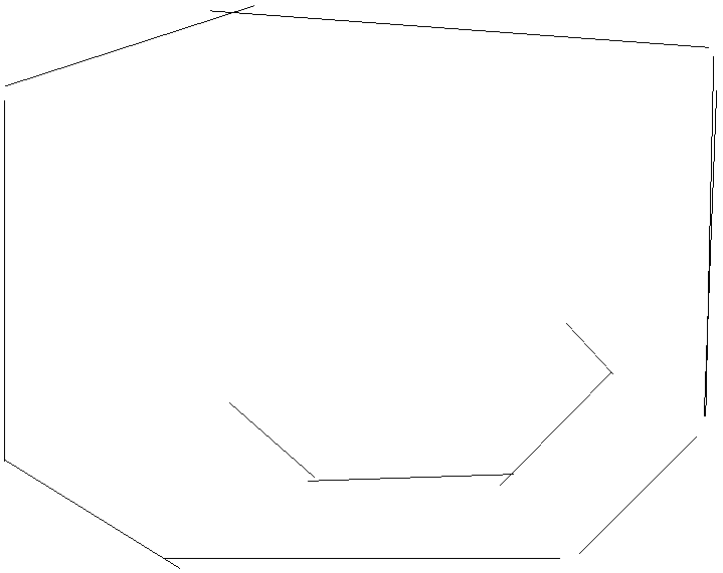
Literatürde yer alan çalışmalarda, özellikle kısıtlı algılayıcılar kullanıldığında duvarların birbirine dik olduğunun varsayıldığı görülmüştür (Choi, 2008; Yap, 2009). Bu çalışmada böyle bir varsayımda bulunulmamıştır.



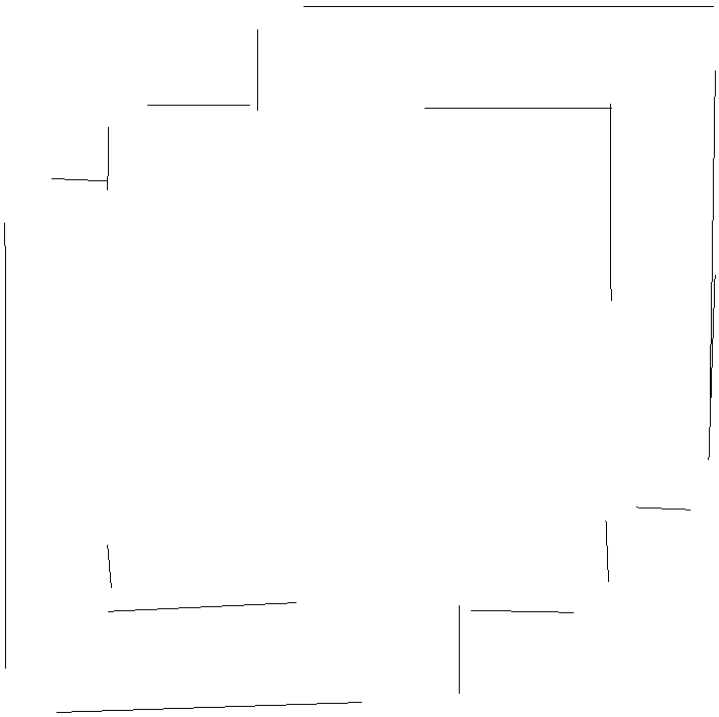
Şekil 4.1 Ortam 1'de, gürültü yokken duvar takibi ile elde edilen harita



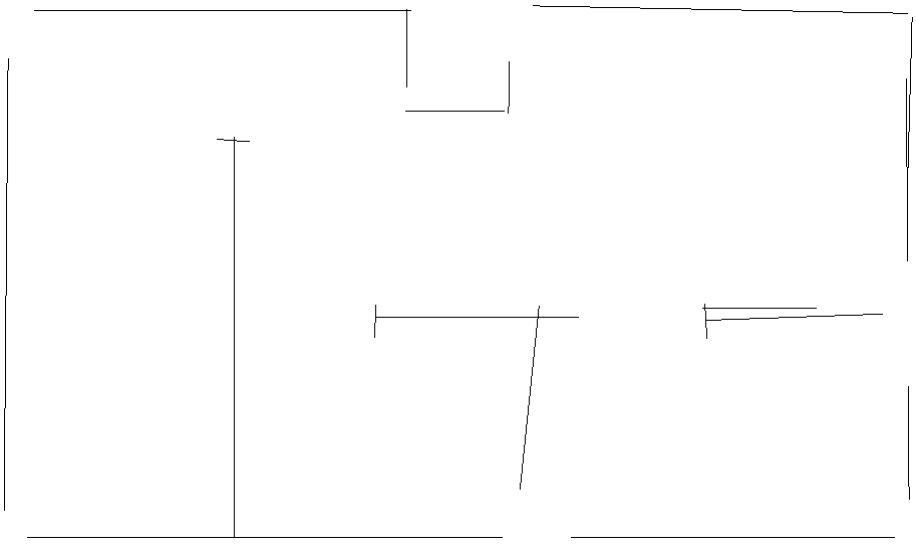
Şekil 4.2 Ortam 2’de, gürültü yokken duvar takibi ile elde edilen harita



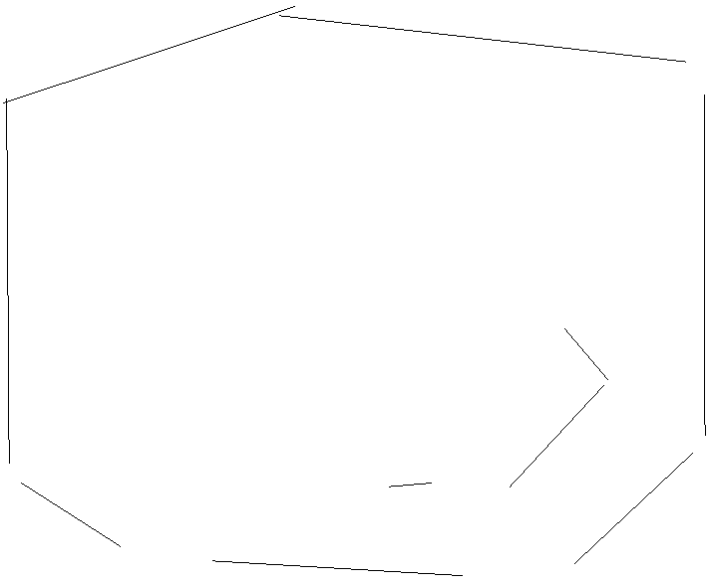
Şekil 4.3 Ortam 3’te gürültü yokken duvar takibi ile elde edilen harita



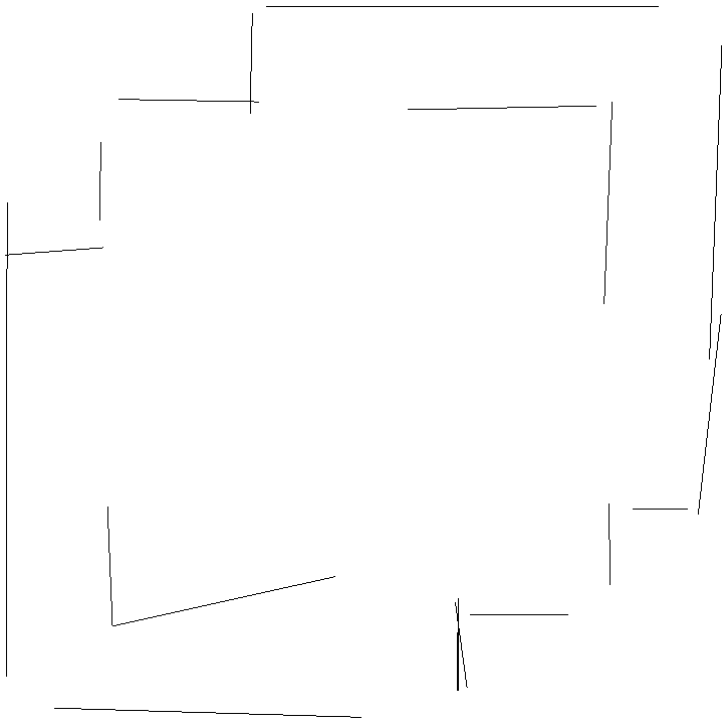
Şekil 4.4 Ortam 1’de, standart sapması 2 cm olan algılayıcı gürültüsü varken duvar takibi ile çıkarılan harita



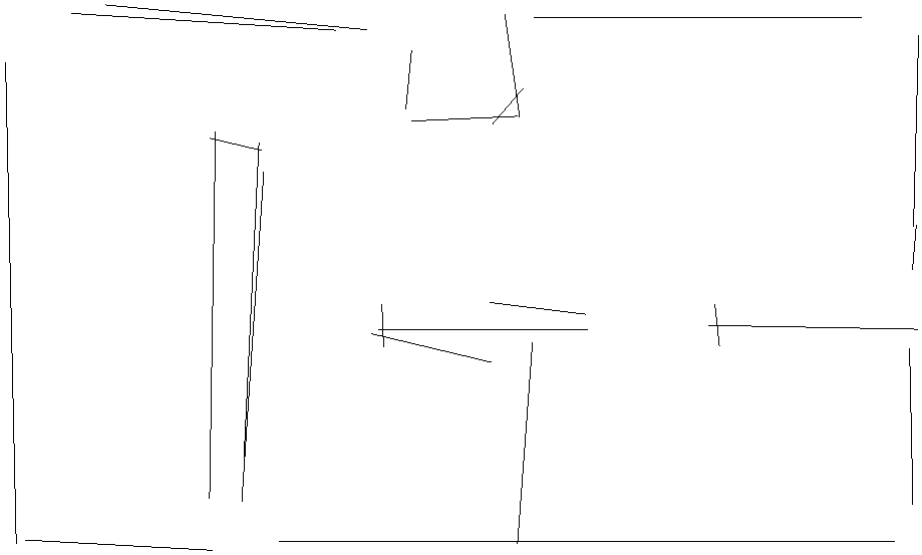
Şekil 4.5 Ortam 2’de, standart sapması 2 cm olan algılayıcı gürültüsü varken duvar takibi ile çıkarılan harita



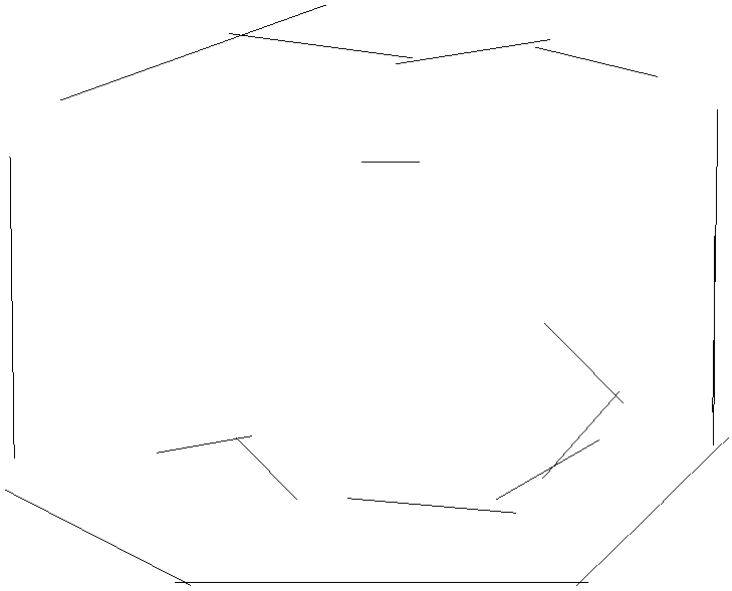
Şekil 4.6 Ortam 3’te, standart sapması 2 cm olan algılayıcı gürültüsü varken duvar takibi ile elde edilen harita



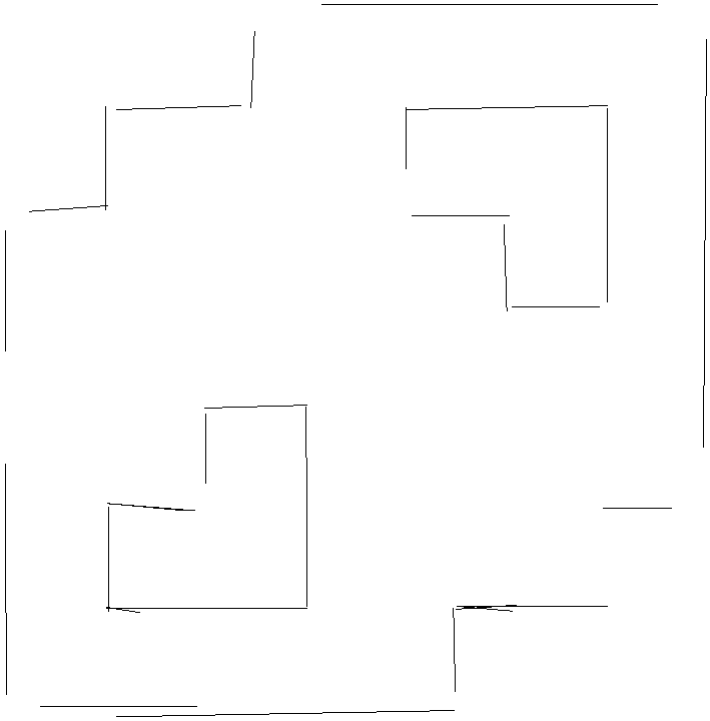
Şekil 4.7 Ortam 1’de, standart sapması 4 cm olan algılayıcı gürültüsü varken duvar takibi ile çıkarılan harita



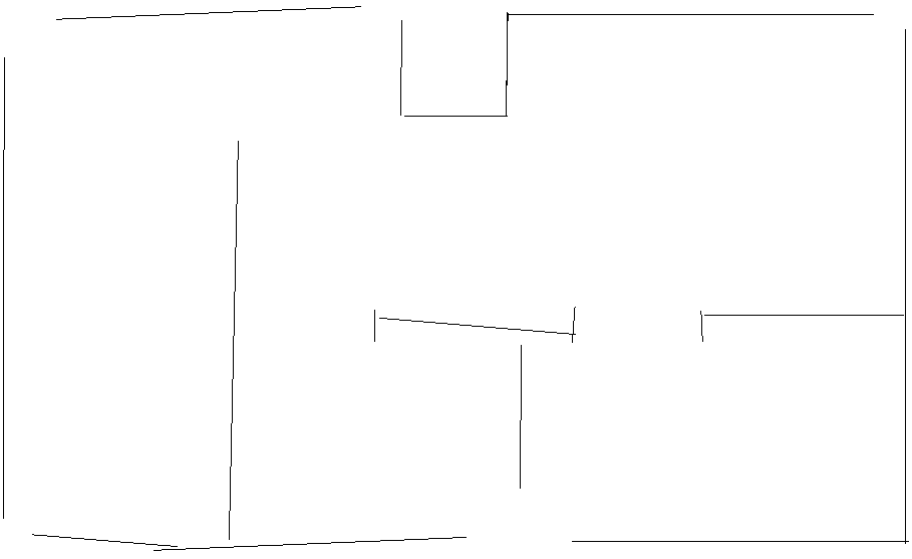
Şekil 4.8 Ortam 2’de, standart sapması 4 cm olan algılayıcı gürültüsü varken duvar takibi ile çıkarılan harita



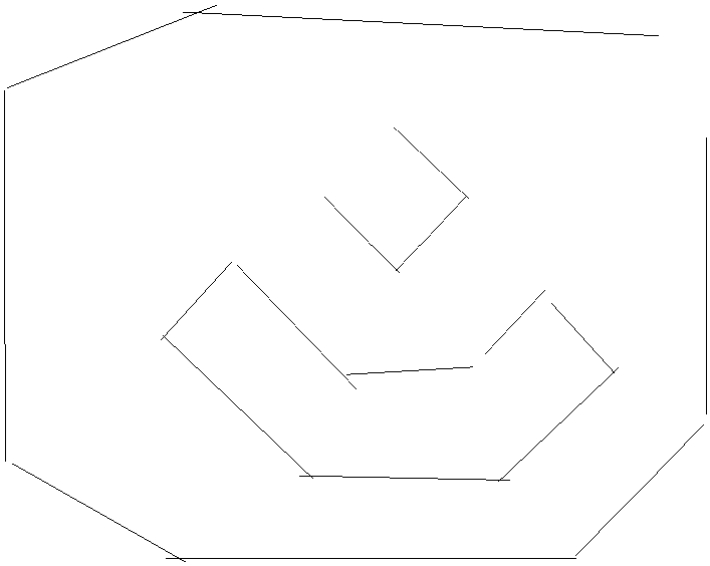
Şekil 4.9 Ortam 3’te, standart sapması 4 cm olan algılayıcı gürültüsü varken duvar takibi ile çıkarılan harita



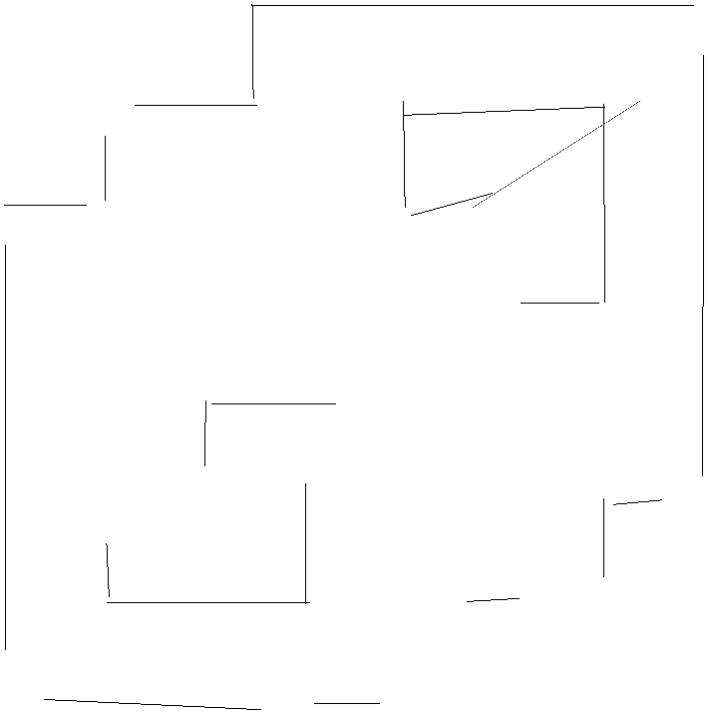
Şekil 4.10 Ortam 1’de, gürültü yokken keşif stratejisi kullanıldığında çıkarılan harita



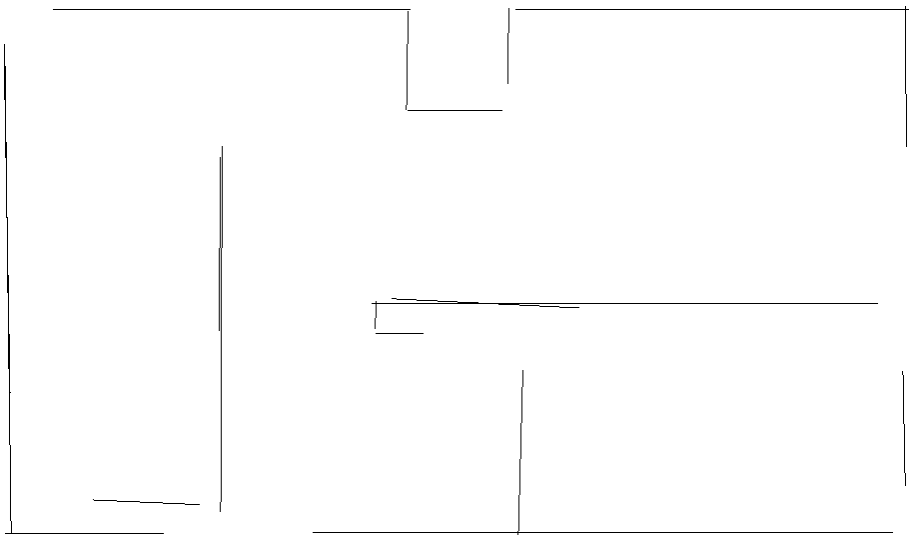
Şekil 4.11 Ortam 2’de, gürültü yokken keşif stratejisi kullanıldığında çıkarılan harita



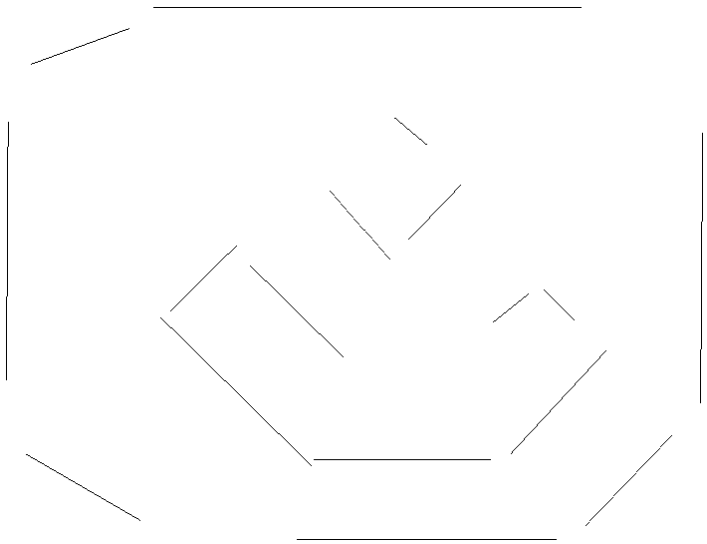
Şekil 4.12 Ortam 3’te gürültü yokken keşif stratejisi ile elde edilen harita



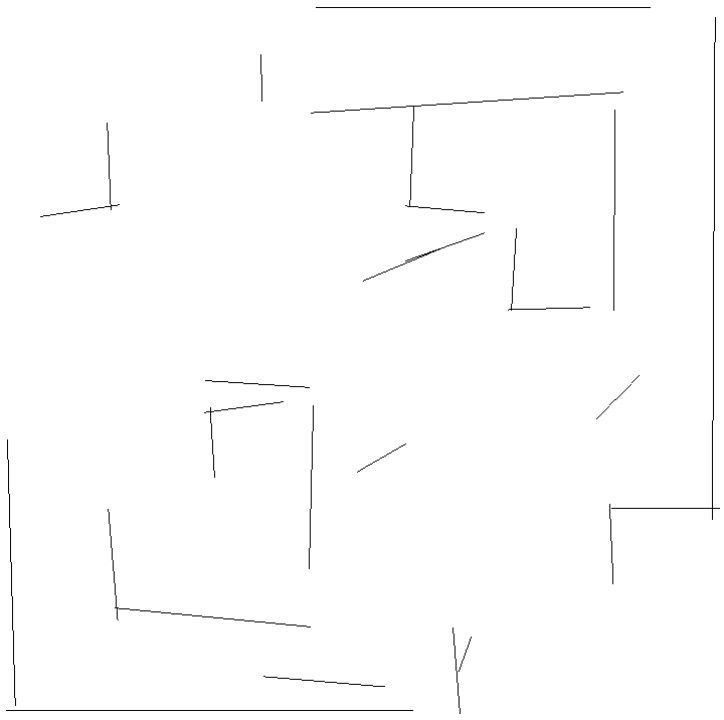
Şekil 4.13 Ortam 1’de, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında çıkarılan harita



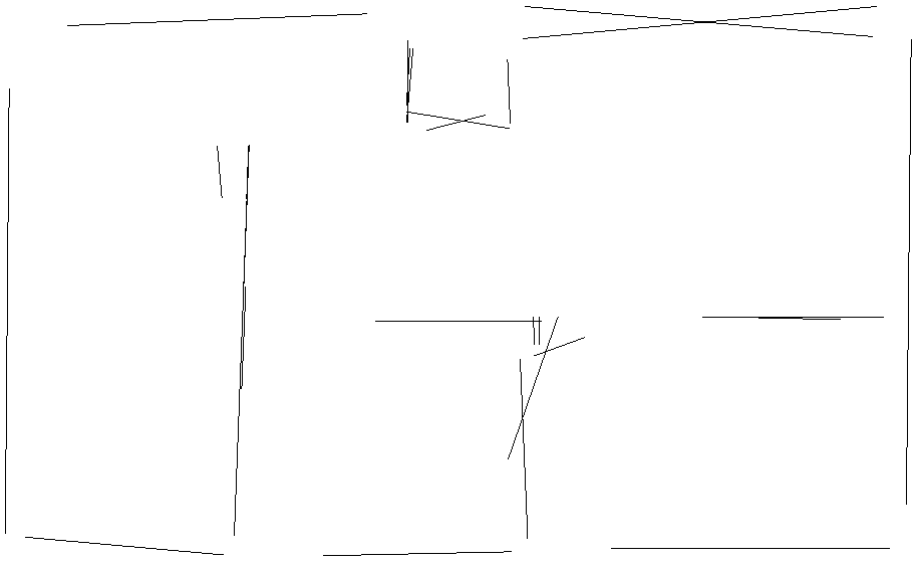
Şekil 4.14 Ortam 2'de, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında çıkarılan harita



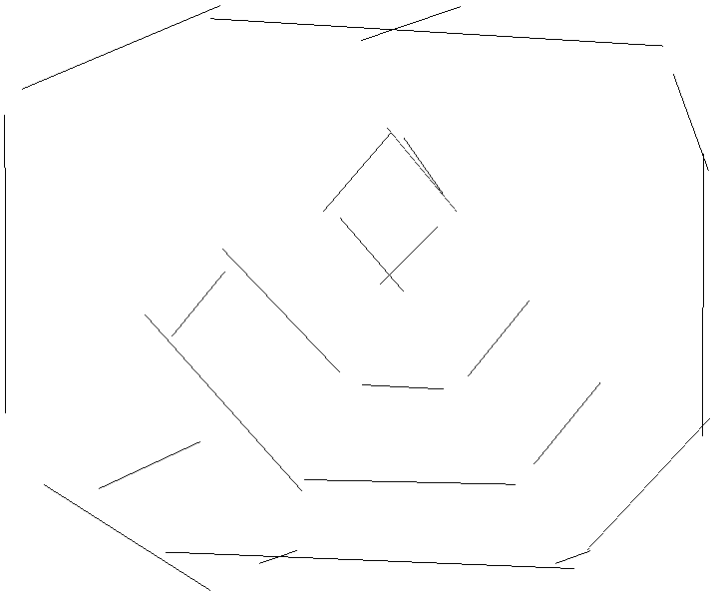
Şekil 4.15 Ortam 3'te, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi ile elde edilen harita



Şekil 4.16 Ortam 1’de, standart sapması 4 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında çıkarılan harita



Şekil 4.17 Ortam 2’de, standart sapması 4 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında çıkarılan harita



Şekil 4.18 Ortam 3’te, standart sapması 4 cm olan algılayıcı gürültüsü varken keşif stratejisi ile elde edilen harita

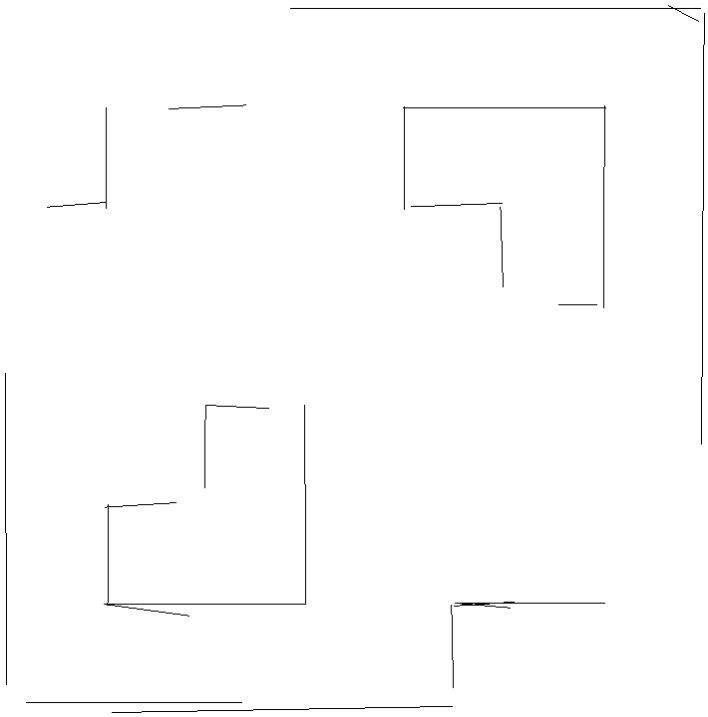
4.2 Sonuçlar

Bu aşamada tek robotla GKF kullanarak, doğrusal özellikleri de Hough Dönüşümü ile çıkararak haritalama yapılması amaçlanmıştır. Yapılan denemelerde bu yöntemin algılayıcı gürültüsüne karşı dayanıklı olduğu görülmüştür.

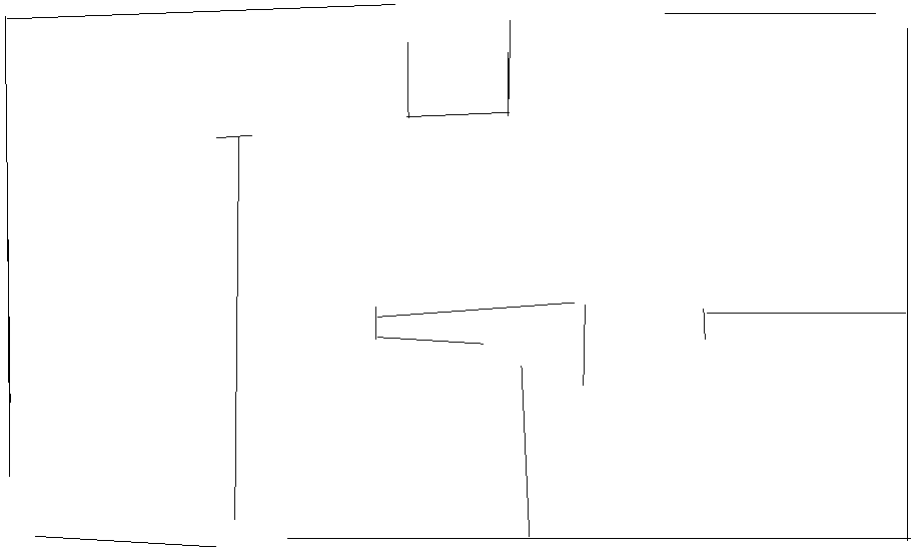
5. ROBOT TAKIMI İLE HARİTALAMA

Çalışmanın son aşamasında, iki robottan oluşan bir robot takımının Şekil 2.12, 2.13 ve 2.14'te gösterilen ortamlarda farklı gürültü seviyelerindeki haritalama performansı değerlendirilmiştir. Buna ek olarak dört robottan oluşan bir robot takımı ile de denemeler yapılmış, bunların çıkardıkları haritalar ve ortamda dolaşım başarıları değerlendirilmiştir. Eş Zamanlı Konum Belirleme ve Haritalama algoritması olarak Genişletilmiş Kalman Filtresi, yer işaretçisi olarak ise doğrusal özellikler kullanılmıştır. Doğrusal özelliklerin çıkarımında Hough Dönüşümü uygulanmıştır. Robotların ortamda gezinmesi için daha önce de bahsedilen sınır tabanlı keşif stratejisi kullanılmıştır.

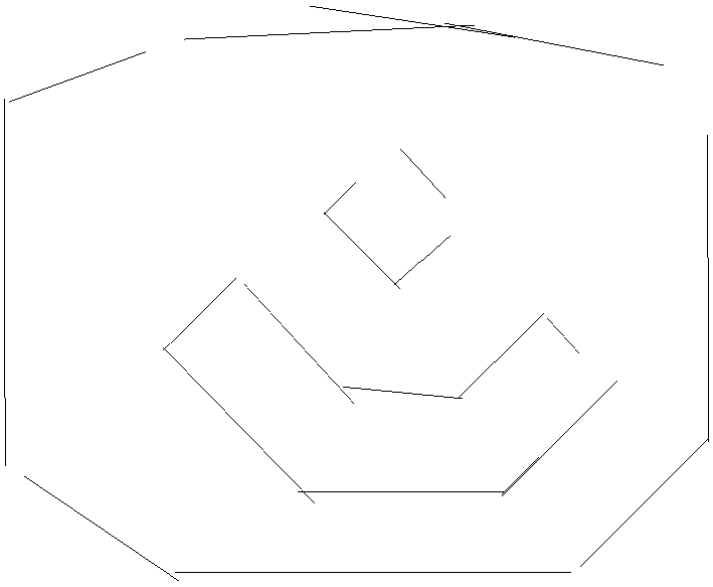
5.1 İki Robotla Deneysel Sonuçlar



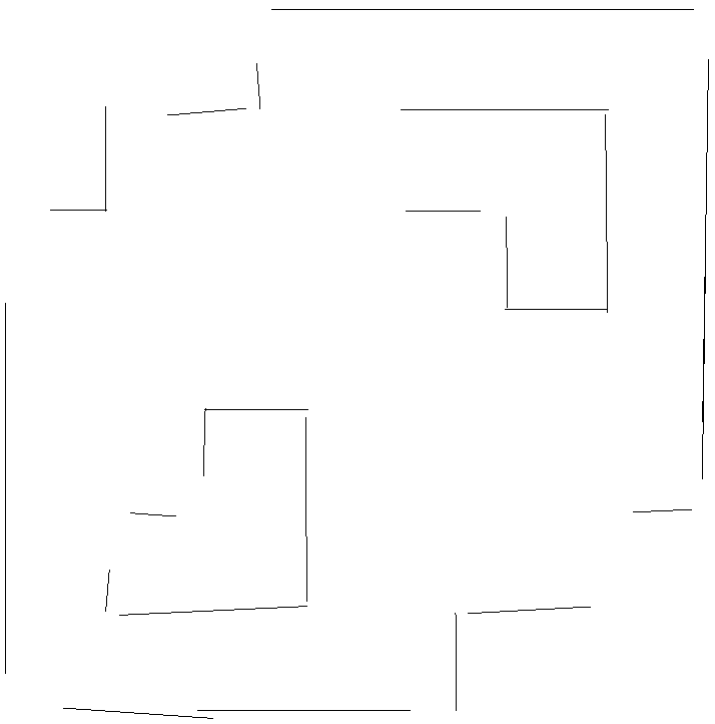
Şekil 5.1 Ortam 1'de, gürültü yokken keşif stratejisi kullanıldığında iki robot tarafından çıkarılan harita



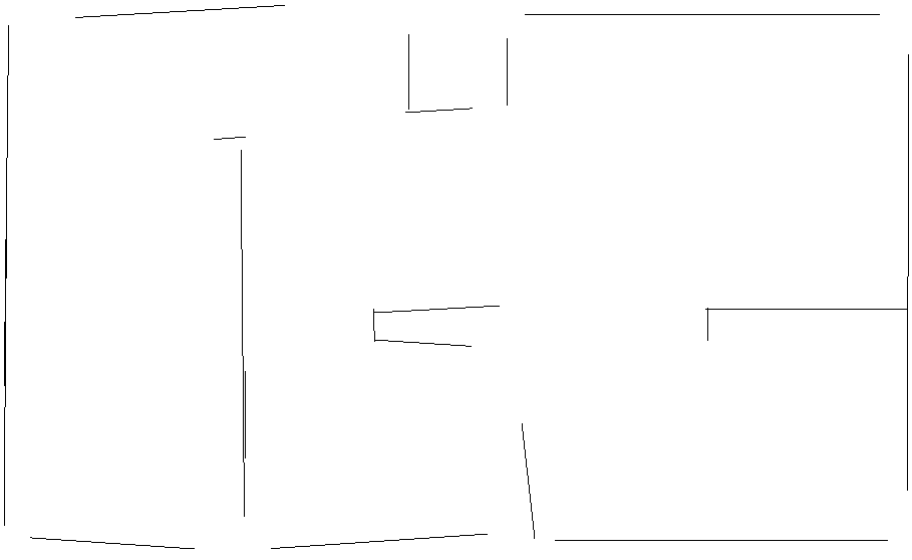
Şekil 5.2 Ortam 2’de, gürültü yokken keşif stratejisi kullanıldığında iki robot tarafından çıkarılan harita



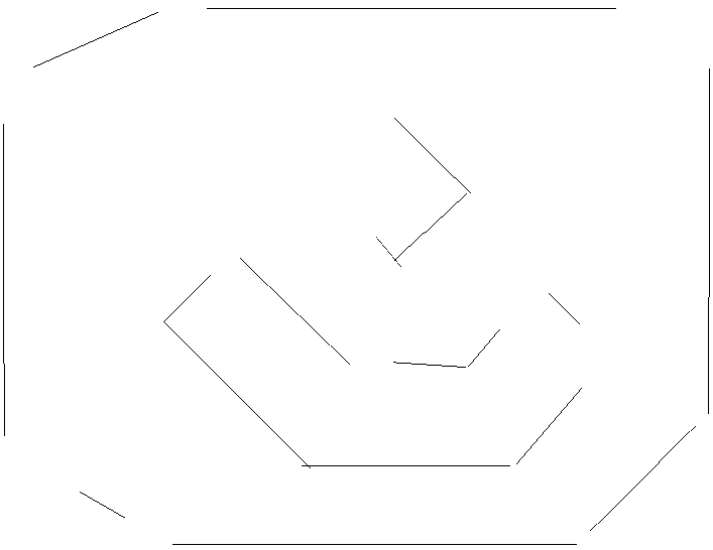
Şekil 5.3 Ortam 3’te, gürültü yokken keşif stratejisi kullanıldığında iki robot tarafından çıkarılan harita



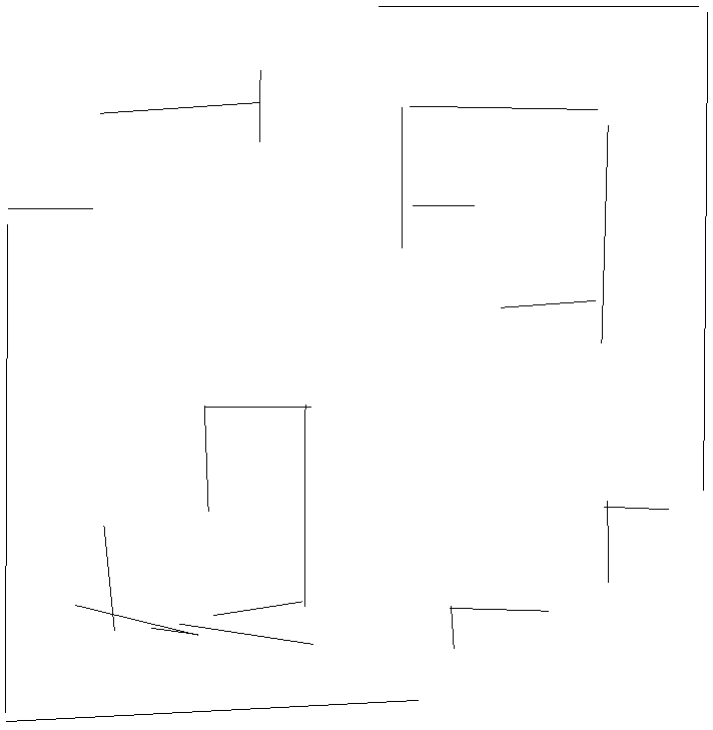
Şekil 5.4 Ortam 1’de, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında iki robot tarafından çıkarılan harita



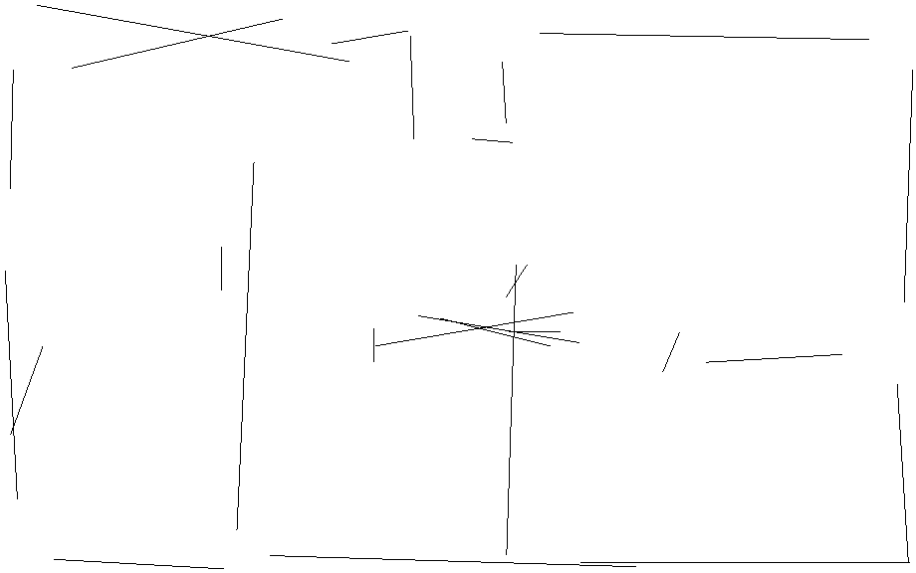
Şekil 5.5 Ortam 2’de, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında iki robot tarafından çıkarılan harita



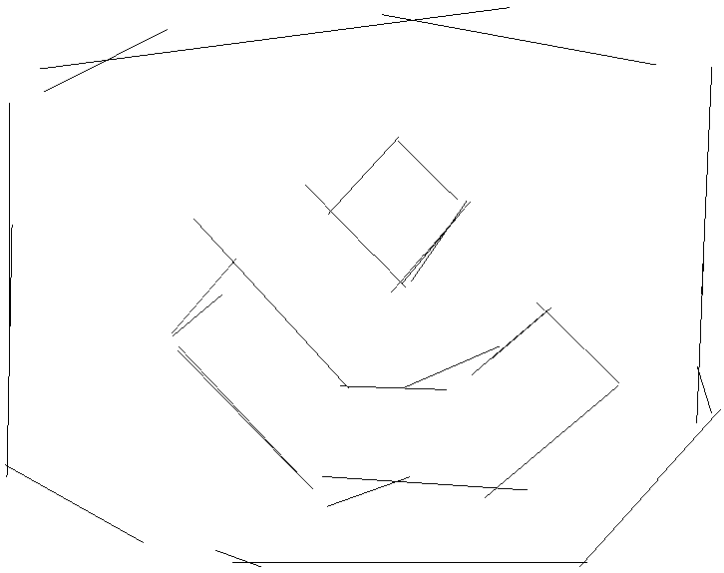
Şekil 5.6 Ortam 3’te, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında iki robot tarafından çıkarılan harita



Şekil 5.7 Ortam 1’de, standart sapması 4 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında iki robot tarafından çıkarılan harita

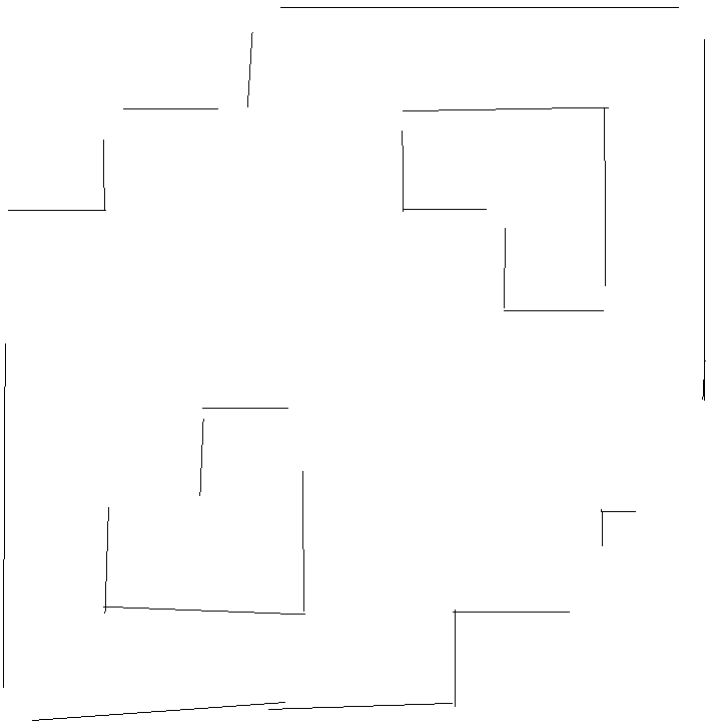


Şekil 5.8 Ortam 2'de, standart sapması 4 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında iki robot tarafından çıkarılan harita

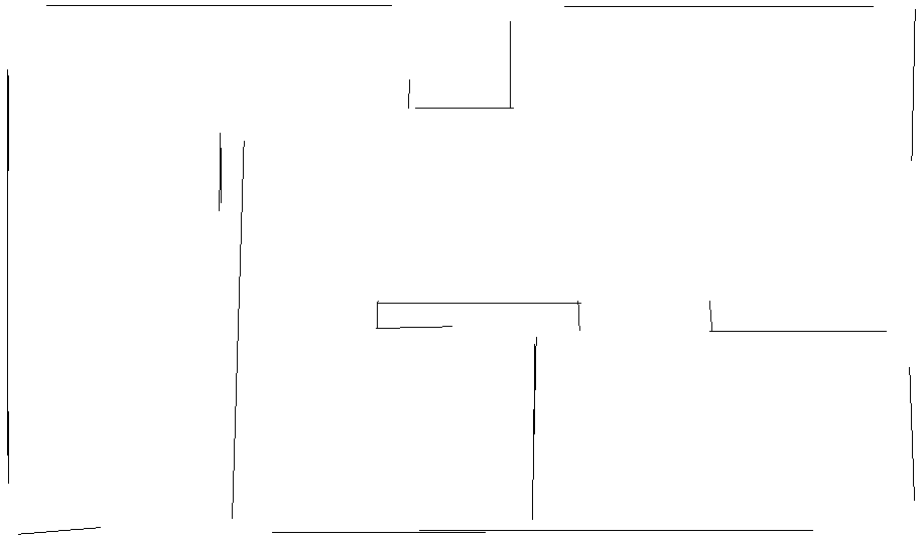


Şekil 5.9 Ortam 3'te, standart sapması 4 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında iki robot tarafından çıkarılan harita

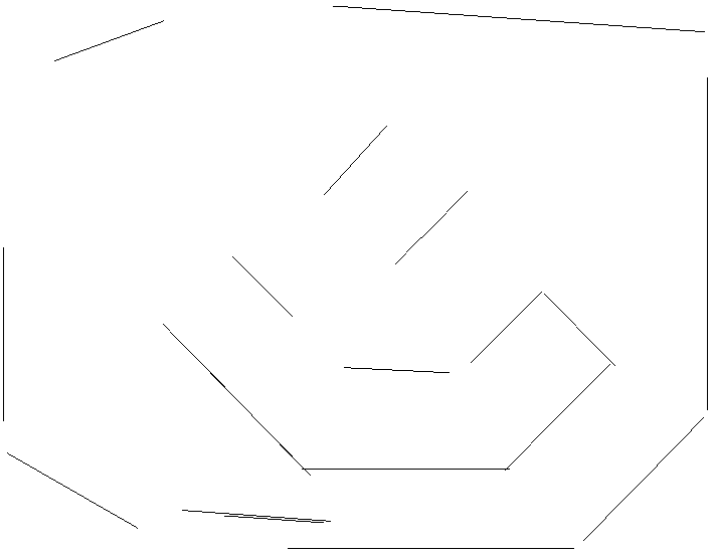
5.2 Dört Robotla Deneysel Sonuçlar



Şekil 5.10 Ortam 1’de, gürültü yokken keşif stratejisi kullanıldığında dört robot tarafından çıkarılan harita



Şekil 5.11 Ortam 2’de, gürültü yokken keşif stratejisi kullanıldığında dört robot tarafından çıkarılan harita



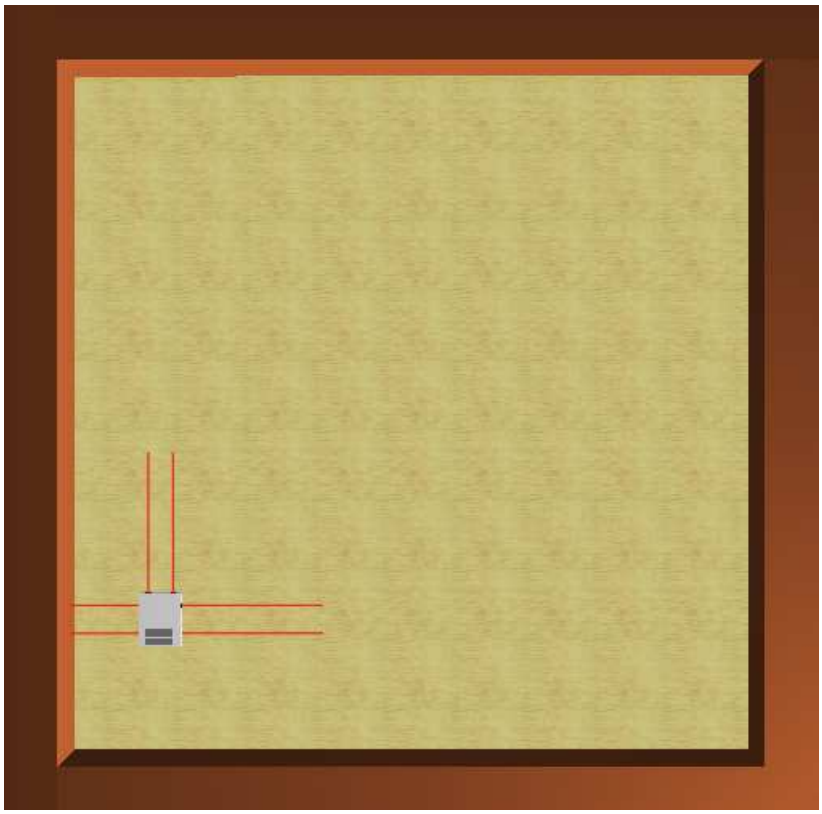
Şekil 5.12 Ortam 3’te, gürültü yokken keşif stratejisi kullanıldığında dört robot tarafından çıkarılan harita

5.3 Sonuçlar

Bu aşamada iki robottan oluşan bir robot takımı ile haritalama yapılması amaçlanmıştır. İki robot kullanılmasının sistem üzerindeki en belirgin iyileştirmesi keşif süresinde gerçekleşen azalmadır. Farklı ortamlarda farklı dolaşım yöntemleri kullanıldığında robotun haritayı tamamlamasının ne kadar sürdüğüne dair bilgiler son bölümde yer alan Çizelge 6.1’de görülebilir. Aynı yer işaretçisi birden fazla robot tarafından tekrar tekrar görüleceği için algılayıcı gürültüsüne karşı dayanaklılığın artacağı bekleniyordu fakat yapılan denemelerde buna dair bir işaret görülememiştir. Dört robot kullanılarak yapılan denemelerde de keşif süresinde kısalma görülmüştür.

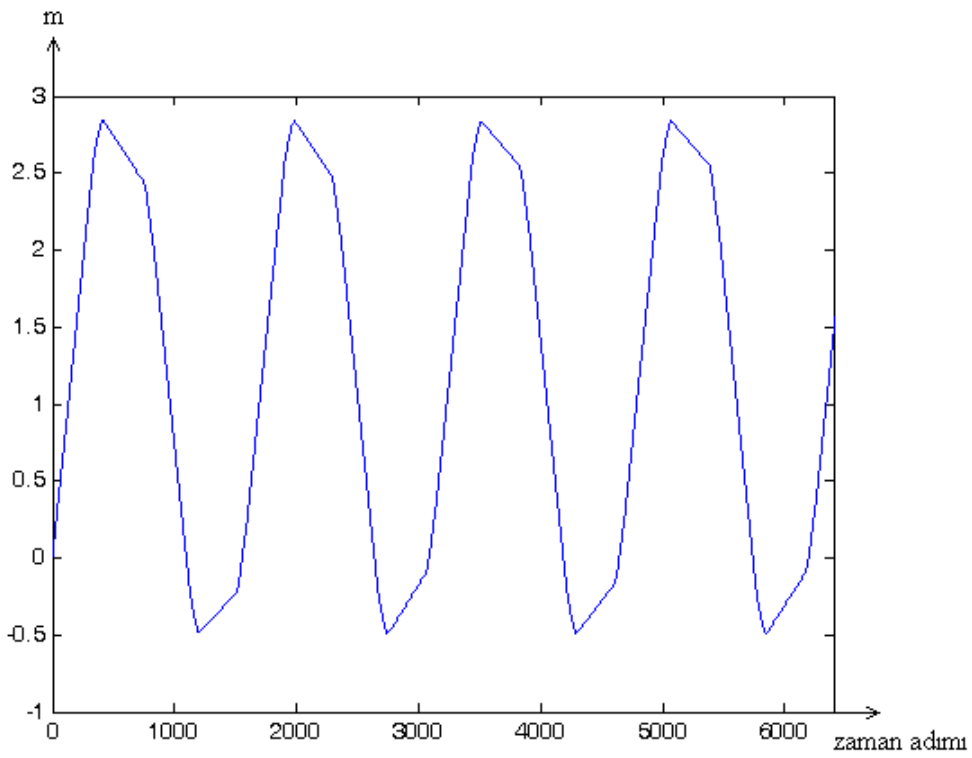
6. GENİŞLETİLMİŞ KALMAN FİLTRESİ VE HAREKET GÜRÜLTÜSÜ

Hareket gürültüsü olduğunda Genişletilmiş Kalman Filtresi'nin bir iyileştirme yapıp yapamadığını görmek için Şekil 6.1'de görülen bir kenarı 185 santimetre olan kare şeklindeki ortamda robota dört tur attırılmış, robotun gerçek konumu ile hesaplanan konumu arasındaki fark, yani konumdaki hata kaydedilmiştir. GKF kullanıldığında ve kullanılmadığında bu hatanın nasıl bir hal aldığı gözlenmiştir.

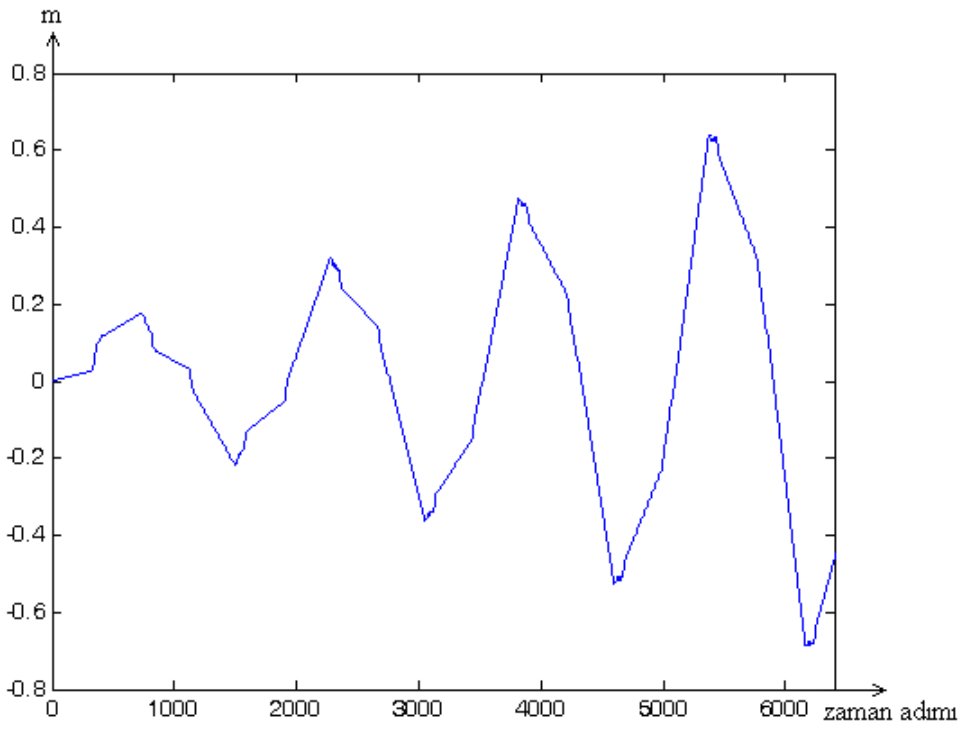


Şekil 6.1 GKF'nin hareket gürültüsünde yaptığı iyileştirmeyi gözlemlemek amacıyla robotun dolaştırıldığı ortam

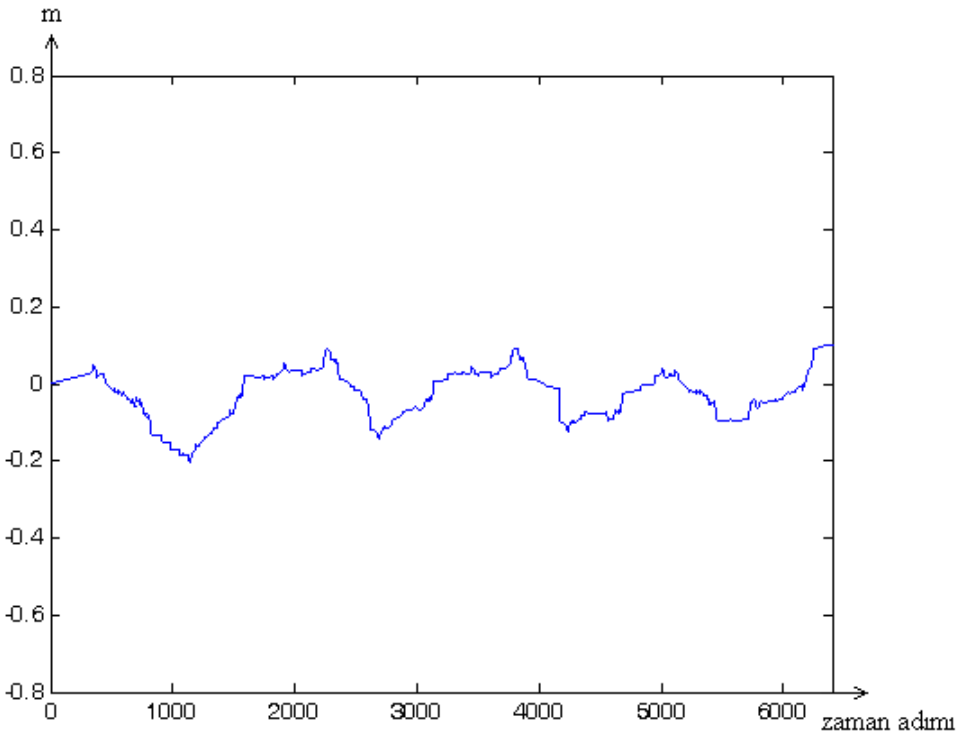
Şekil 6.2'de dört tur boyunca robotun konumunun x eksenindeki değişimi görülmektedir. Bu değerler simülasyondan alınan gerçek konum bilgileridir. Şekil 6.3'te GKF kullanılmadığında, Şekil 6.4'te ise GKF kullanıldığında x ekseninde oluşan konum hatası görülmektedir.



Şekil 6.2 Robotun ortamda attığı dört tur boyunca konumda x ekseni doğrultusunda meydana gelen değişim.

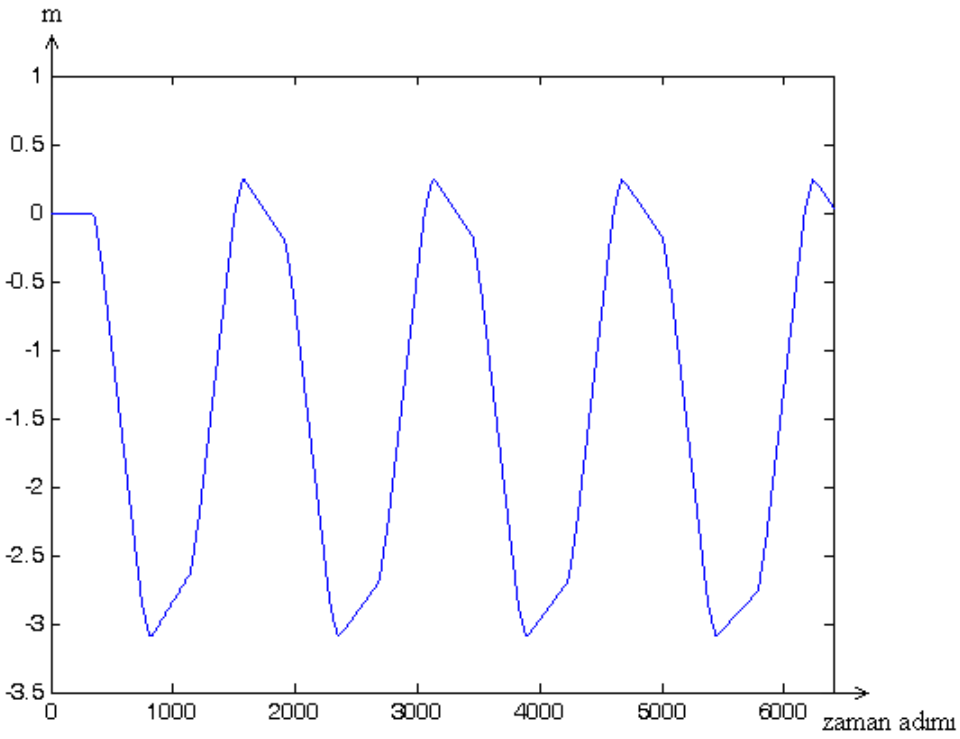


Şekil 6.3 GKF kullanılmazken robotun x eksenindeki konum hatasının değişimi

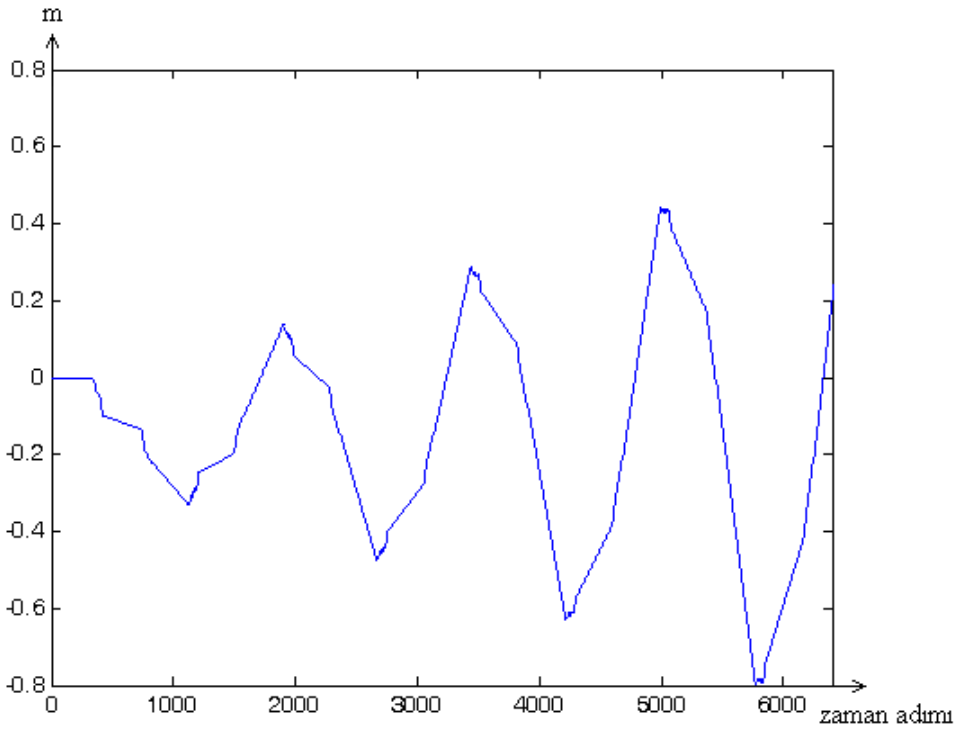


Şekil 6.4 GKF kullanılırken robotun x eksenindeki konum hatasının değişimi

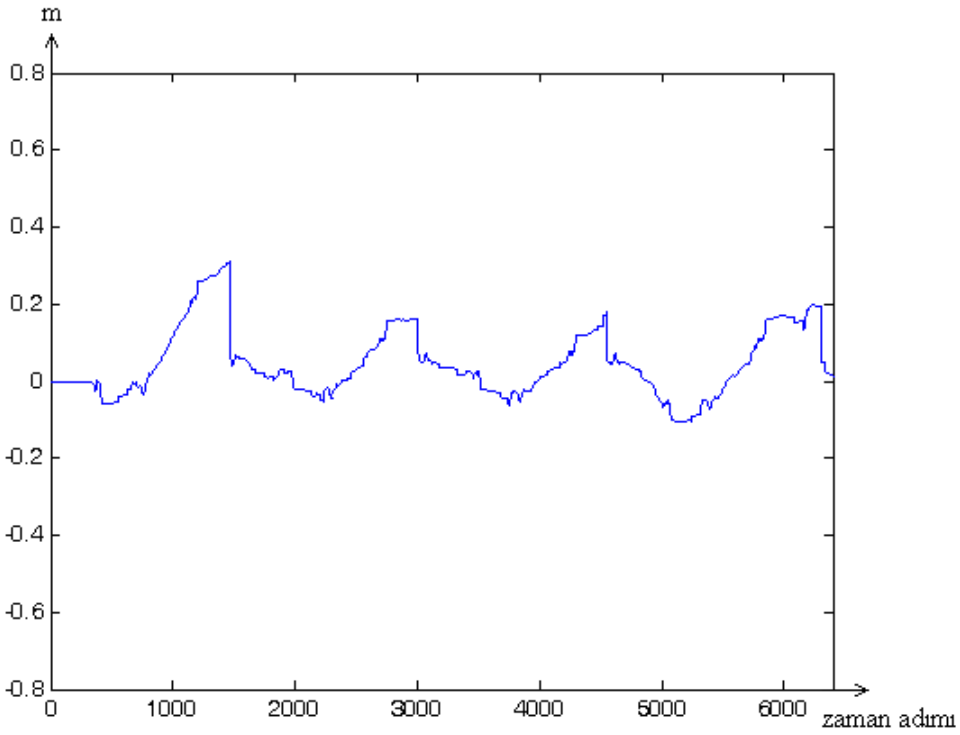
Şekil 6.5'te dört tur boyunca robotun konumunun y eksenindeki değişimi görülmektedir. Şekil 6.6'da GKF kullanılmadığında, Şekil 6.7'de ise GKF kullanıldığında y ekseninde oluşan konum hatası görülmektedir.



Şekil 6.5 Robotun ortamda attığı dört tur boyunca konumda y ekseni doğrultusunda meydana gelen değişim.

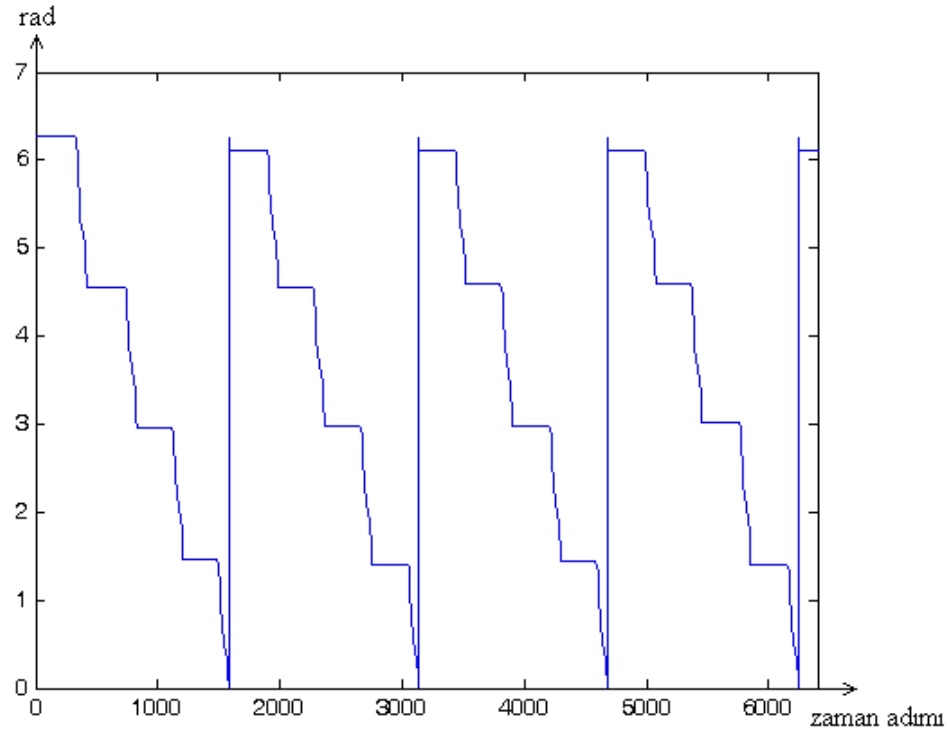


Şekil 6.6 GKF kullanılmazken robotun y eksenindeki konum hatasının değişimi

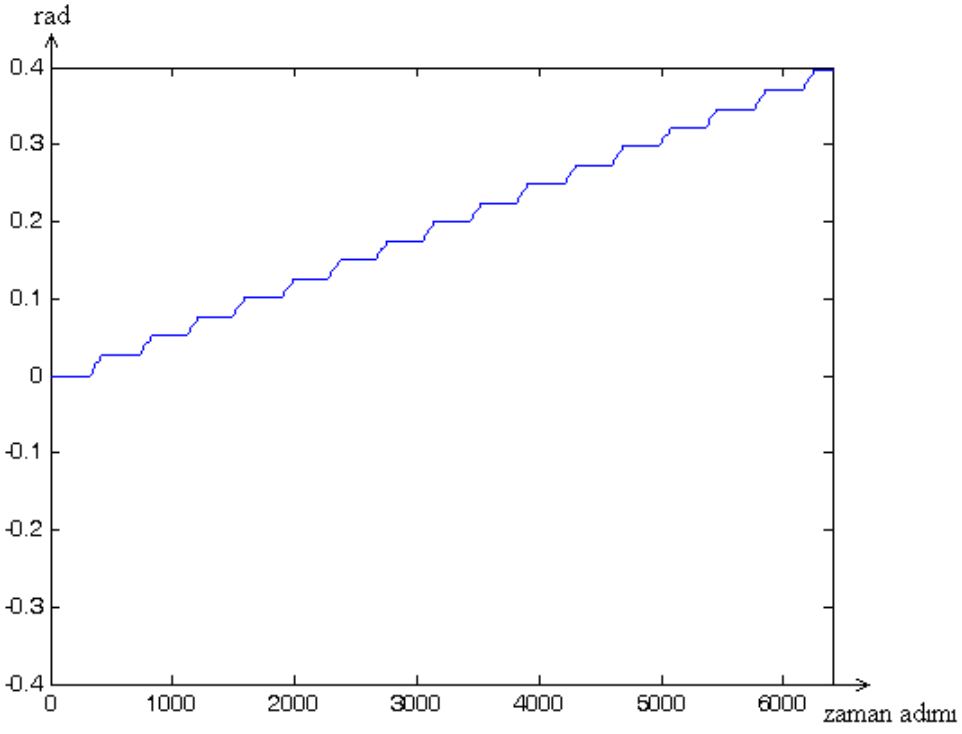


Şekil 6.7 GKF kullanılırken robotun y eksenindeki konum hatasının değişimi

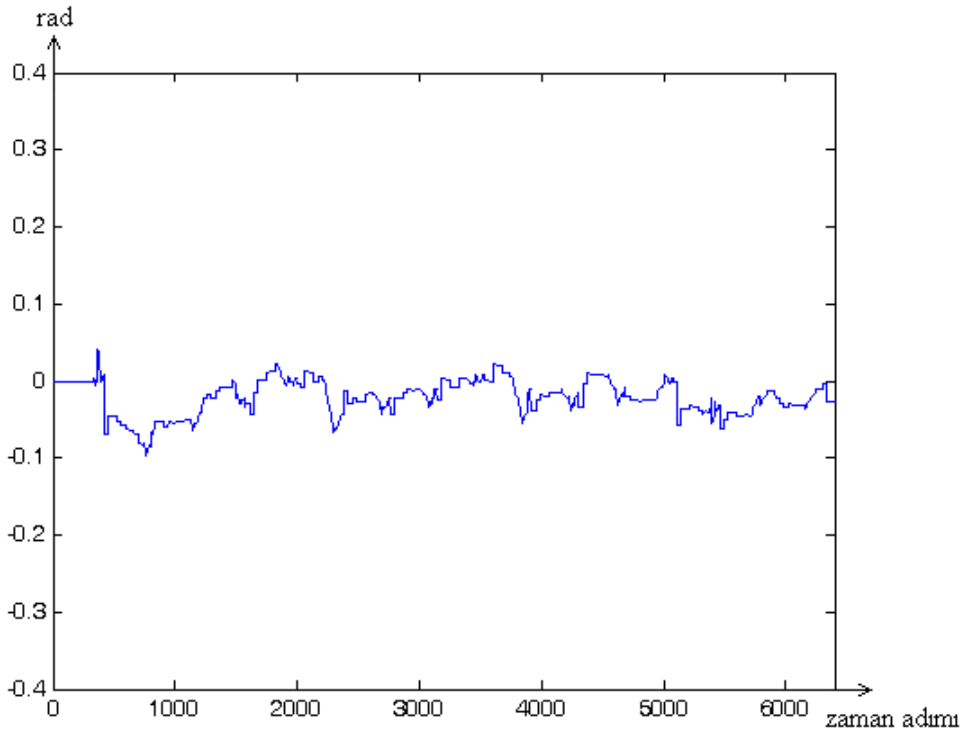
Şekil 6.8'de dört tur boyunca robotun yönelme açısının (θ) değişimi görülmektedir. Değerlerdeki 0'dan 6.28'e ani sıçramalar robot dönerken açının 0'dan 2π'ye geçişinden kaynaklanmaktadır. Şekil 6.9'da GKF kullanılmadığında, Şekil 6.10'da ise GKF kullanıldığında yönelme açısında oluşan hata görülmektedir. Hata hesaplanırken 0-2π geçişleri özel olarak ele alınmıştır.



Şekil 6.8 Robotun ortamda attığı dört tur boyunca yönelme açısında meydana gelen değişim.



Şekil 6.9 GKF kullanılmazken robotun yönelme açısındaki hatanın değişimi



Şekil 6.10 GKF kullanılırken robotun yönelme açısındaki hatanın değişimi

Şekil 6.3, 6.6 ve 6.9'da GKF kullanılmazken hatanın zaman içinde git gide arttığı, Şekil 6.4, 6.7 ve 6.10'da ise GKF kullanıldığında hatada böyle bir artış gerçekleşmediği, hatanın bir aralıkta tutulduğu gözlenebilmektedir.

7. SONUÇ

Bu çalışmada kısıtlı algılayıcıya sahip robotlardan oluşan bir robot takımı ile bir ortamın haritasının çıkarılması amaçlanmıştır. Eş Zamanlı Konum Belirleme ve Haritalama algoritması olarak Genişletilmiş Kalman Filtresi seçilmiş, yer işaretçileri olarak ise ortamdaki gözlenebilir doğrular seçilmiştir. Literatürde yer alan çalışmalarda, özellikle kısıtlı algılayıcılar kullanıldığında duvarların birbirine dik olduğunun varsayıldığı görülmüştür (Choi, 2008; Yap, 2009). Bu çalışmada böyle bir varsayımda bulunulmamıştır.

Doğru çıkarımı için iki metot düşünülmüştür, ilki aynı taraftaki iki algılayıcının aldığı noktasal veriye doğru uydurma, ikincisi ise robotun ilerleyişi boyunca nokta toplayıp bu noktalara Hough Dönüşümü ile doğru uydurma şeklindedir. Bu metotlardan ilkinin gürültüden fazlaca etkilendiği, köşe dönüşlerinde iki ayrı duvardan alınan noktalar nedeniyle de pek çok yanlış yer işaretçisi oluşturabildiği görülmüştür. İkinci metot gürültüye karşı daha dirençlidir ve yapısı gereği iki ayrı duvardan alınan noktaları ayırt edebilme ya da göz ardı edebilme becerisine sahiptir. Bununla birlikte Nguyen vd. (2005) tarafından da ifade edildiği gibi Hough Dönüşümünde kullanmak üzere uygun (r,θ) parametrelerinin belirlenmesinin güç olduğu görülmüştür. Bundan sonra yapılacak çalışmalarda da ikinci metodun kullanılması, belki bu metot üzerinde iyileştirmeler yapılması, ölçüm belirsizliklerinin doğru çıkarımında göz önünde bulundurulması düşünülmektedir.

Genişletilmiş Kalman Filtresi'nin algılayıcı gürültüsünü düzgün bir şekilde ele alabildiği, yer işaretçilerinin konumunda zaman içinde düzeltme yapabildiği görülmüştür. Daha önce görülen bir doğrunun bir süre sonra yeniden görülmesi durumunda, örneğin döngü kapatılırken yapılan düzeltmelerin de etkili olduğu görülmüştür. Yalnız GKF'deki matris işlemleri nedeniyle çok fazla yer işaretçisi bulunan ortamlarda bu yöntemin kullanılmasının uygun olmadığı, algılayıcı gürültüsü nedeniyle hatalı pek çok yer işaretçisi tespit edildiğinde uygulamanın fazlasıyla yavaşladığı görülmüştür.

GKF'nin hareket gürültüsü bulunduğunda konum tahmininde iyileştirme yapabildiği 6. bölümde açıkça görülmüştür. Fakat önceki bölümlerde denenen haritalarda hareket gürültüsü bulunduğunda aynı başarı gözlenememiştir. Bunda haritaların büyüklüğü ve dış köşelerin fazlalığı oldukça etkilidir. Hareket gürültüsü ile ilgili denemeler yapılırken bir yayında yapılan tavuk-yumurta problemi benzetmesinin ne kadar doğru olduğu görülmüştür; çeşitli durumlarda algılayıcılardan elde edilen mesafe bilgileri noktasal verilere dönüştürülürken robotun yanlış hesaplanan konumu kullanıldığından bu noktalara uydurulan doğru da yanlış

olmaktadır ve robotun konumunda bir düzeltme gerçekleşmesi mümkün olamamaktadır. Hareket gürültüsü ile algılayıcı gürültüsünün birlikte ele alınması pek mümkün görülmemektedir.

Yer işaretçisi olarak noktasal özellikler değil de doğrular seçildiği için bu yer işaretçileri kullanılarak konumda yapılabilecek düzeltme sınırlıdır. Robot takip ettiği duvara göre ancak iki yana kayıp kaymadığını anlayabilmektedir, ilerlediği yönde düzeltme yapması ancak daha önce gördüğü bir duvar karşısına çıkınca mümkün olmaktadır.

Yer işaretçileri karşılaştırılırken iki yer işaretçisi aynı doğru parametrelerine bile sahip olsa arada boşluk olabileceği ihtimali ile uç noktaları arasında mesafe varsa bunlar ayrı yer işaretçileri olarak değerlendirilmektedirler. Duvar takibinde robot ardışık gözlemleri aynı duvardan alabildiği ve çıkarılan doğruların uç noktaları neredeyse bitişik olduğu için bu sorun oluşturmamaktadır. Fakat keşif stratejisinde robot en düşük maliyetli keşfedilmemiş bölgeye ilerlediği için bir duvardan alınan gözlemler birbirinden uzak olabilmekte ve bunlar ayrı yer işaretçileri olarak değerlendirilebilmektedirler. Aynı duvardan bu şekilde kesik kesik okumalar almak hatalı yer işaretçisi oluşumuna neden olabilmektedir. Bundan sonraki çalışmalarda duvar takibi ile keşif stratejisi arasında geçiş yapılan bir dolaşım yöntemi kullanılması düşünülmektedir.

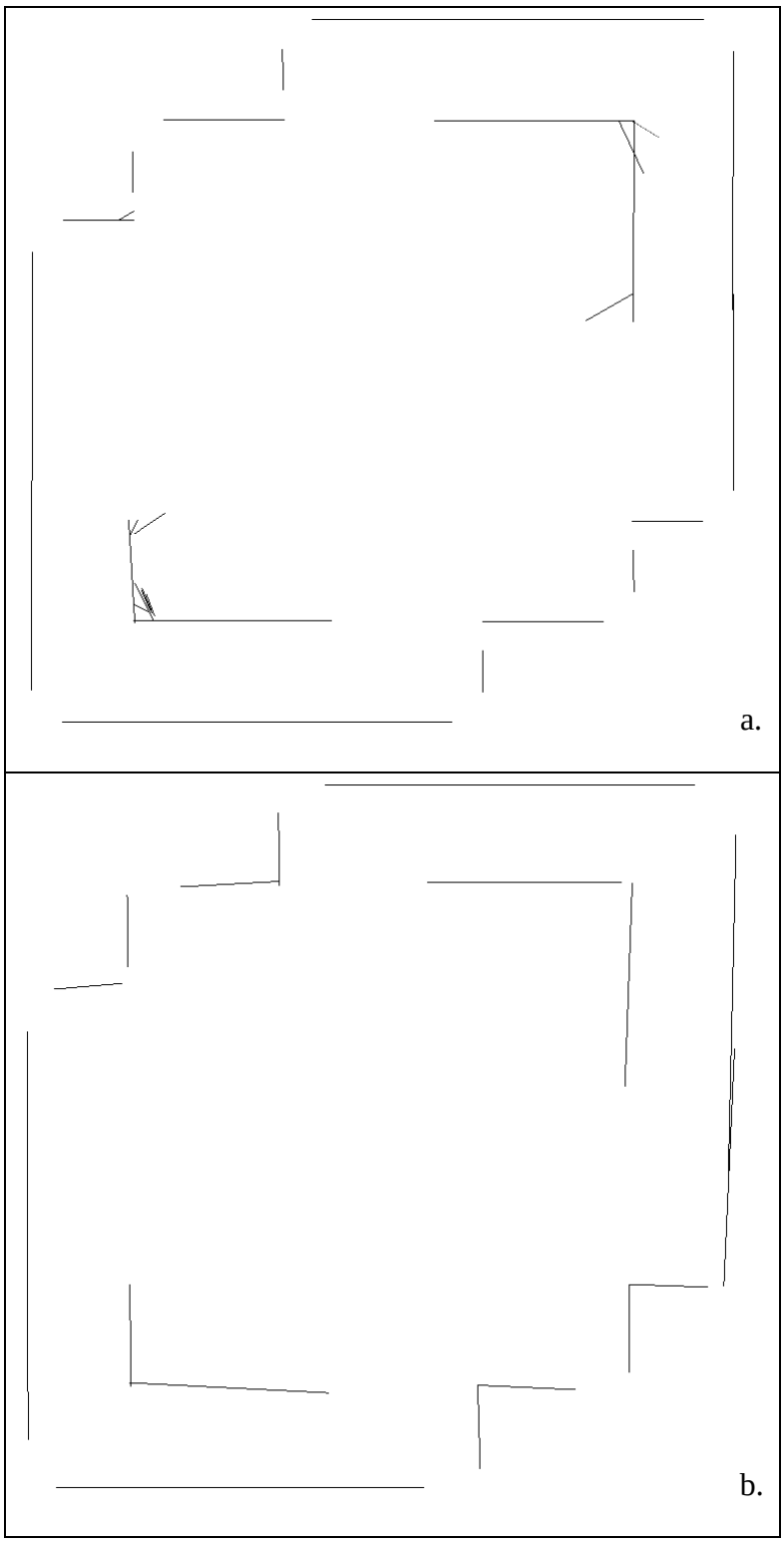
Robot takımı ile haritalama yapmanın en belirgin başarısı keşif süresinin kısalması olmuştur. Çizelge 6.1'de dolaşım süreleri görülmektedir.

Robot takımı ile haritalama yapılırken robotların ortama düzgün bir şekilde dağılması önemlidir, yapılan denemelerde robotların birbirlerine çok yakın konumlardan başlatılmış olsalar dahi ortama düzgün bir şekilde dağılabildikleri görülmüştür.

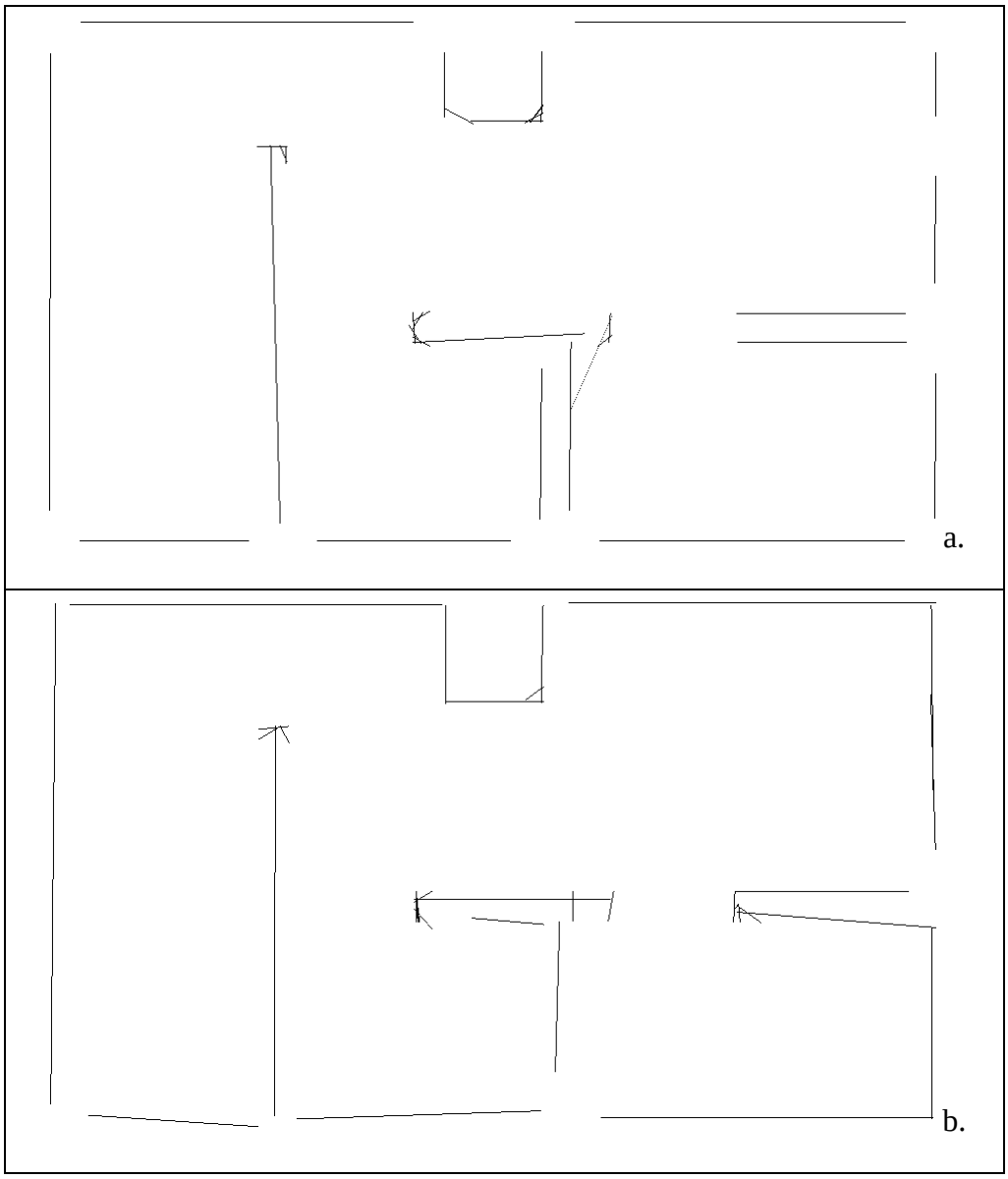
Çizelge 6.1 Haritalardaki dolaşım süreleri

	Tek robot duvar takibi	Tek robot keşif	İki robot keşif	Dört robot keşif
Ortam 1	3:25	10:30	4:35	3:05
Ortam 2	6:40	11:35	5:10	3:11
Ortam 3	3:02	7:15	4:19	2:10

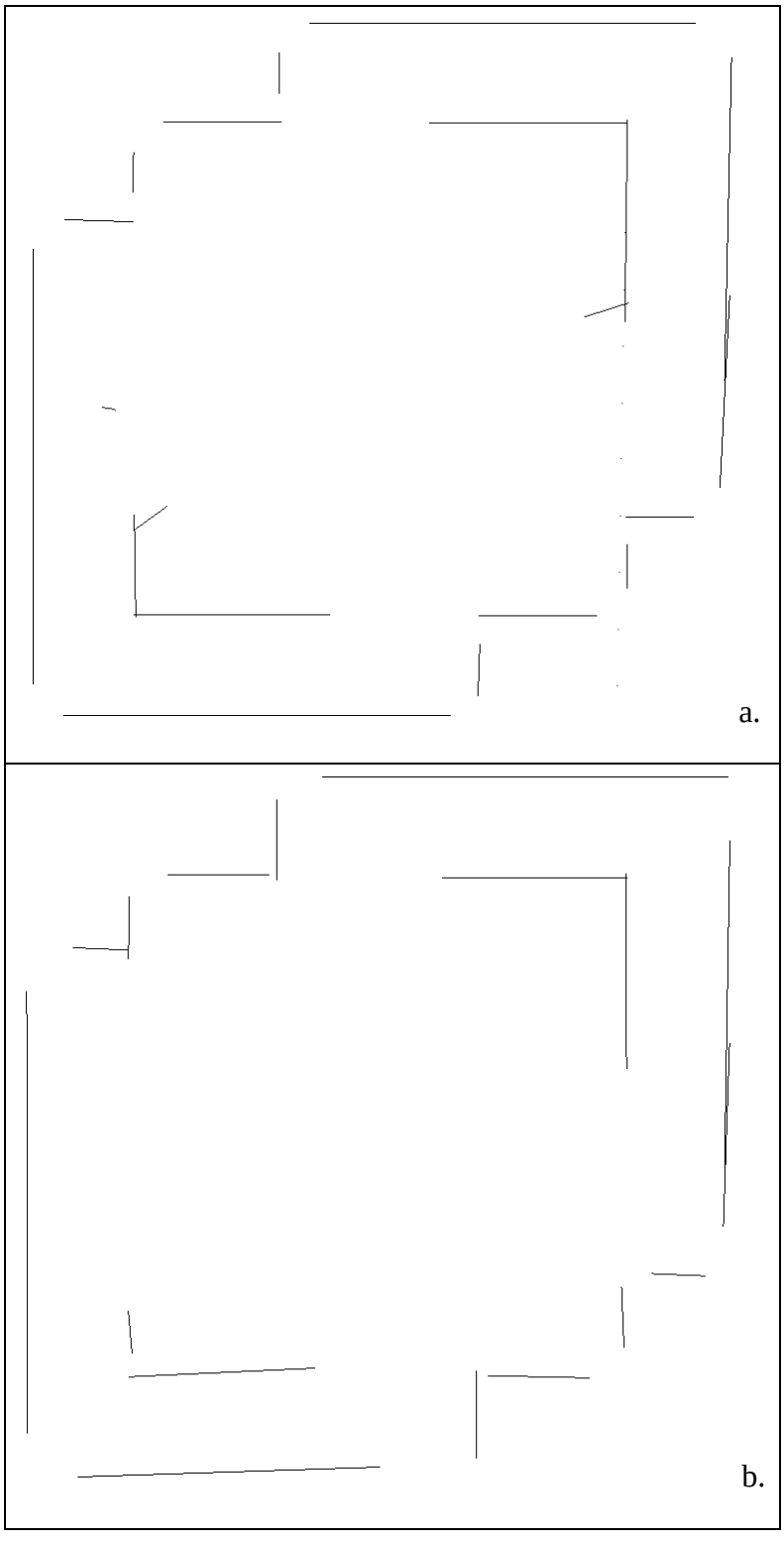
Tek Robotla İki Noktadan Doğru Çıkarımı Yaparak Haritalama, Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama ve Robot Takımı ile Haritalama yöntemleriyle yapılan denemelerin sonuçları daha rahat karşılaştırma yapılabilmesi için Şekil 7.1 – 7.15 arasında birlikte verilmiştir. Hough Dönüşümü kullanıldığında çıkarılan haritaların İki Noktadan Doğru Çıkarımı kullanılarak elde edilen haritalara göre çok daha başarılı olduğu rahatlıkla görülmektedir. Robot Takımı kullanımının ise keşif süresinin kısalması dışında başarıya açık bir etkisi görülmemektedir.



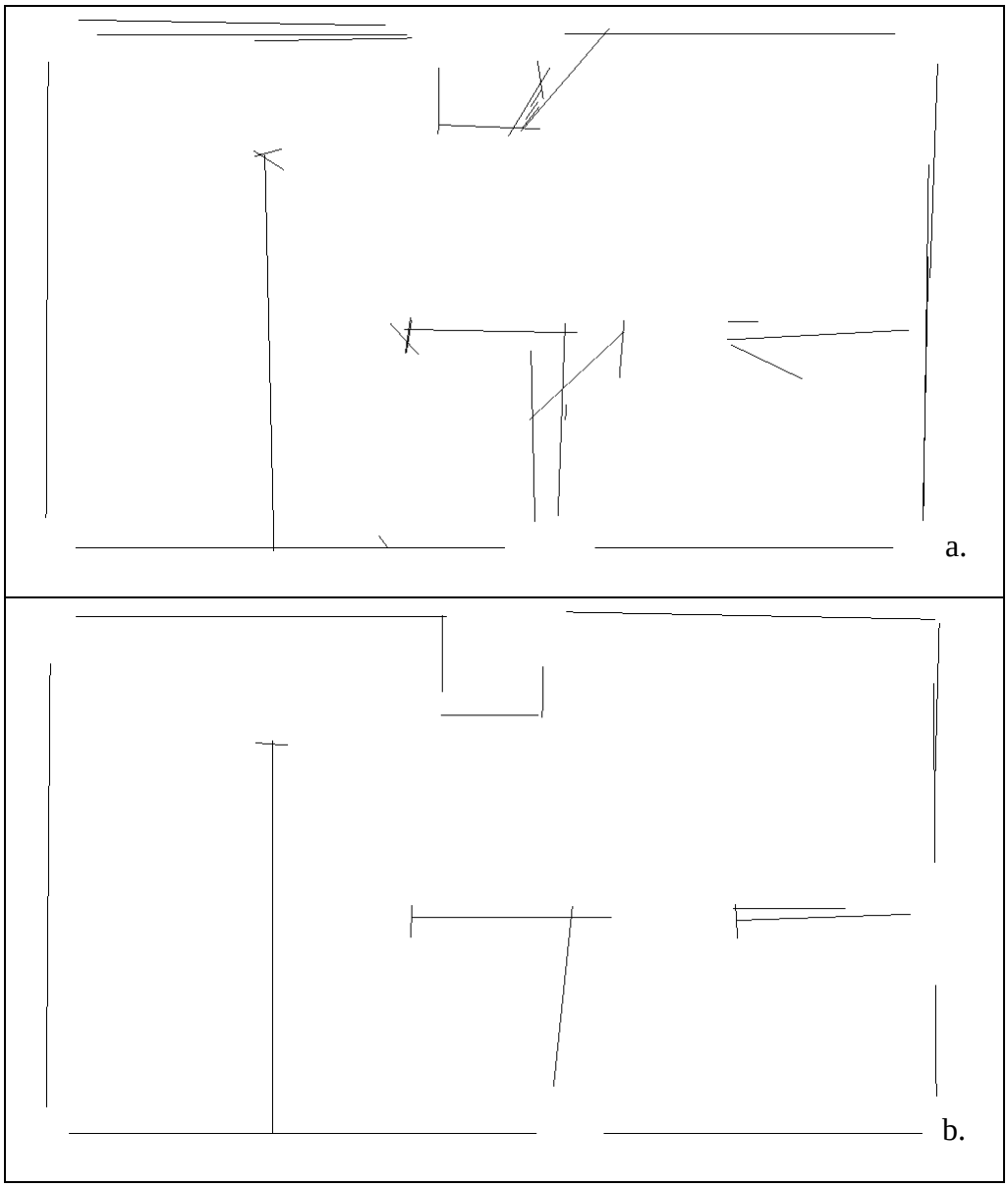
Şekil 7.1 Ortam 1’de, gürültü yokken duvar takibi ile elde edilen haritalar.
a. Tek Robotla İki Noktadan Doğru Çıkarımı Yaparak Haritalama
b. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama



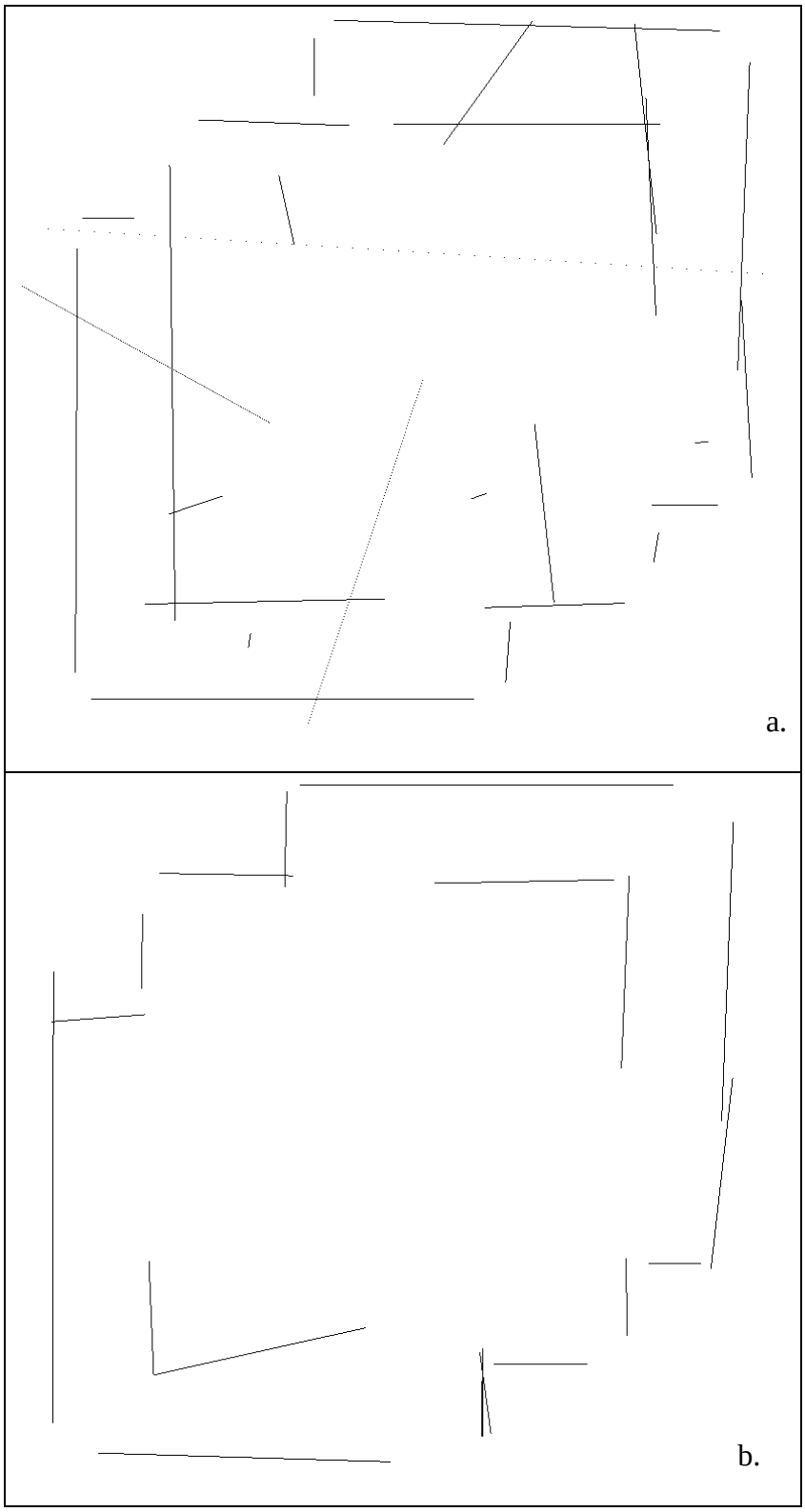
Şekil 7.2 Ortam 2'de, gürültü yokken duvar takibi ile elde edilen haritalar.
a. Tek Robotla İki Noktadan Doğru Çıkarımı Yaparak Haritalama
b. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama



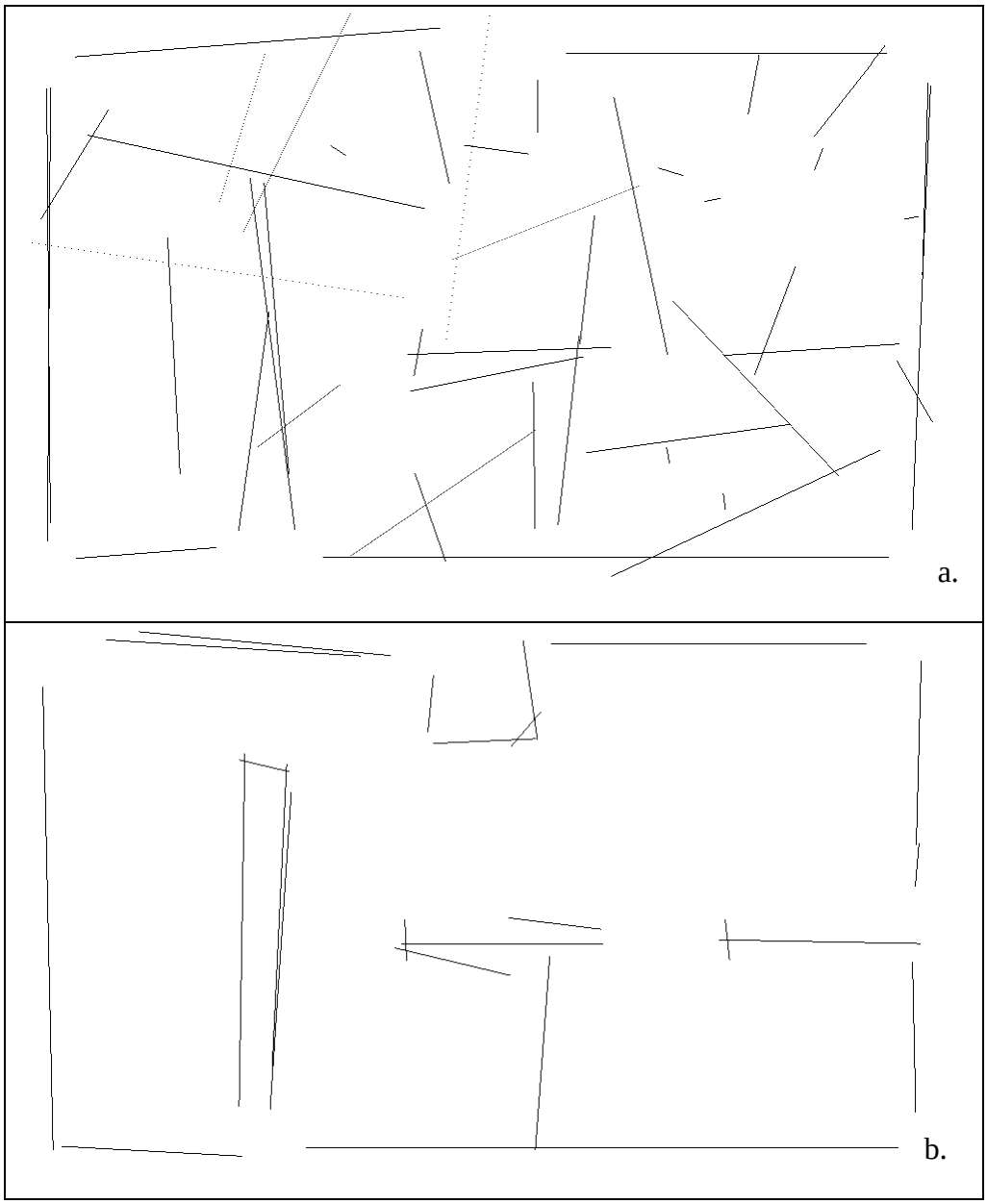
Şekil 7.3 Ortam 1'de, standart sapması 2 cm olan algılayıcı gürültüsü varken duvar takibi ile çıkarılan haritalar.
a. Tek Robotla İki Noktadan Doğru Çıkarımı Yaparak Haritalama
b. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama



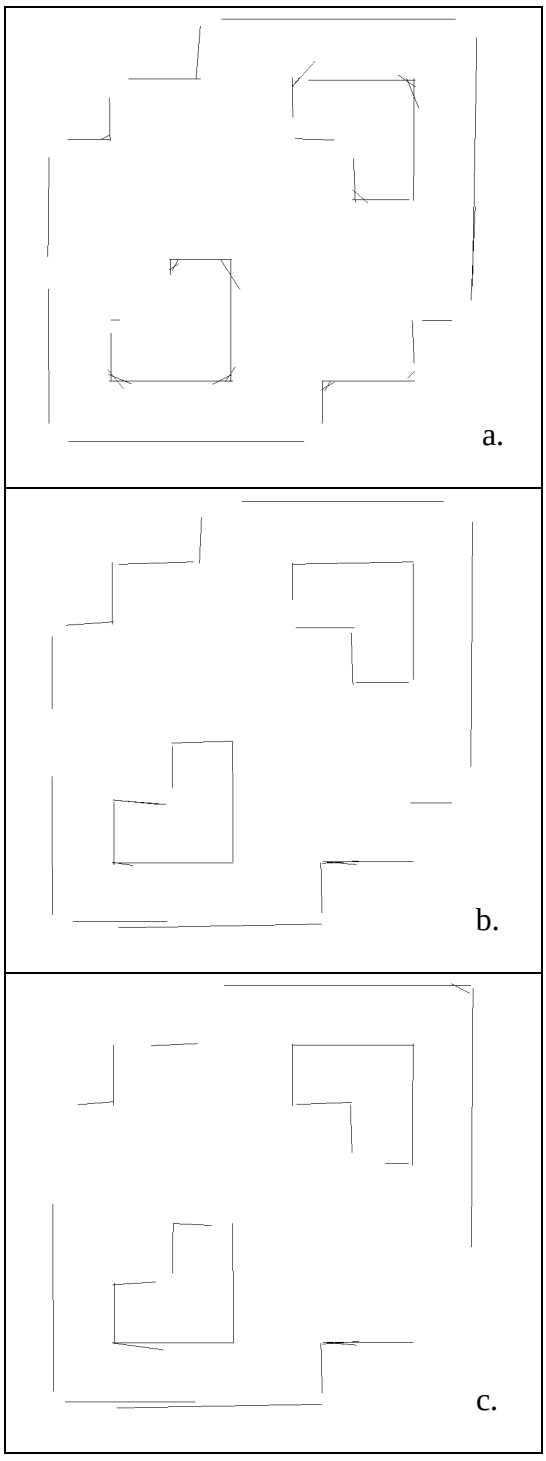
Şekil 7.4 Ortam 2'de, standart sapması 2 cm olan algılayıcı gürültüsü varken duvar takibi ile çıkarılan haritalar.
a. Tek Robotla İki Noktadan Doğru Çıkarımı Yaparak Haritalama
b. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama



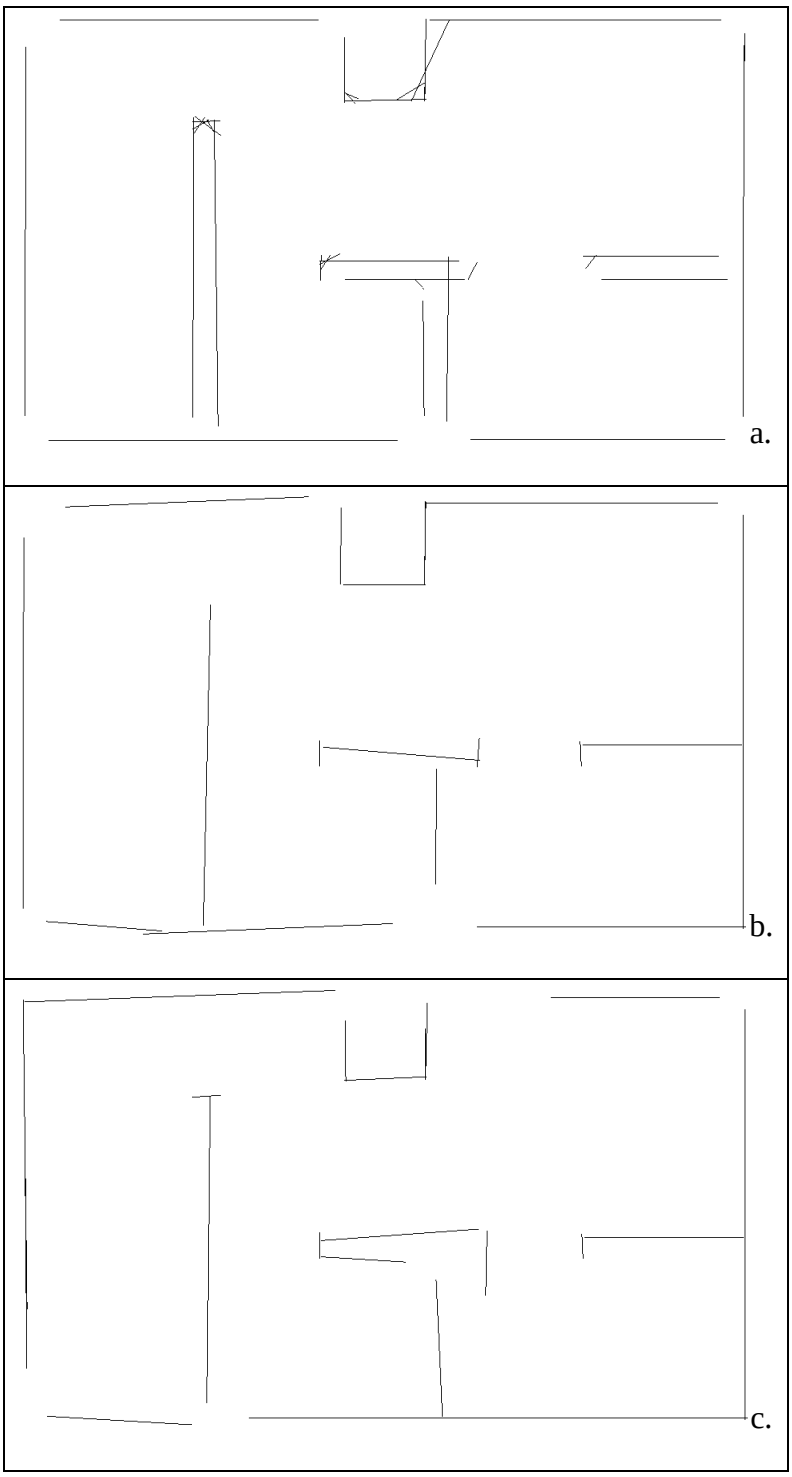
Şekil 7.5 Ortam 1'de, standart sapması 4 cm olan algılayıcı gürültüsü varken duvar takibi ile çıkarılan haritalar.
a. Tek Robotla İki Noktadan Doğru Çıkarımı Yaparak Haritalama
b. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama



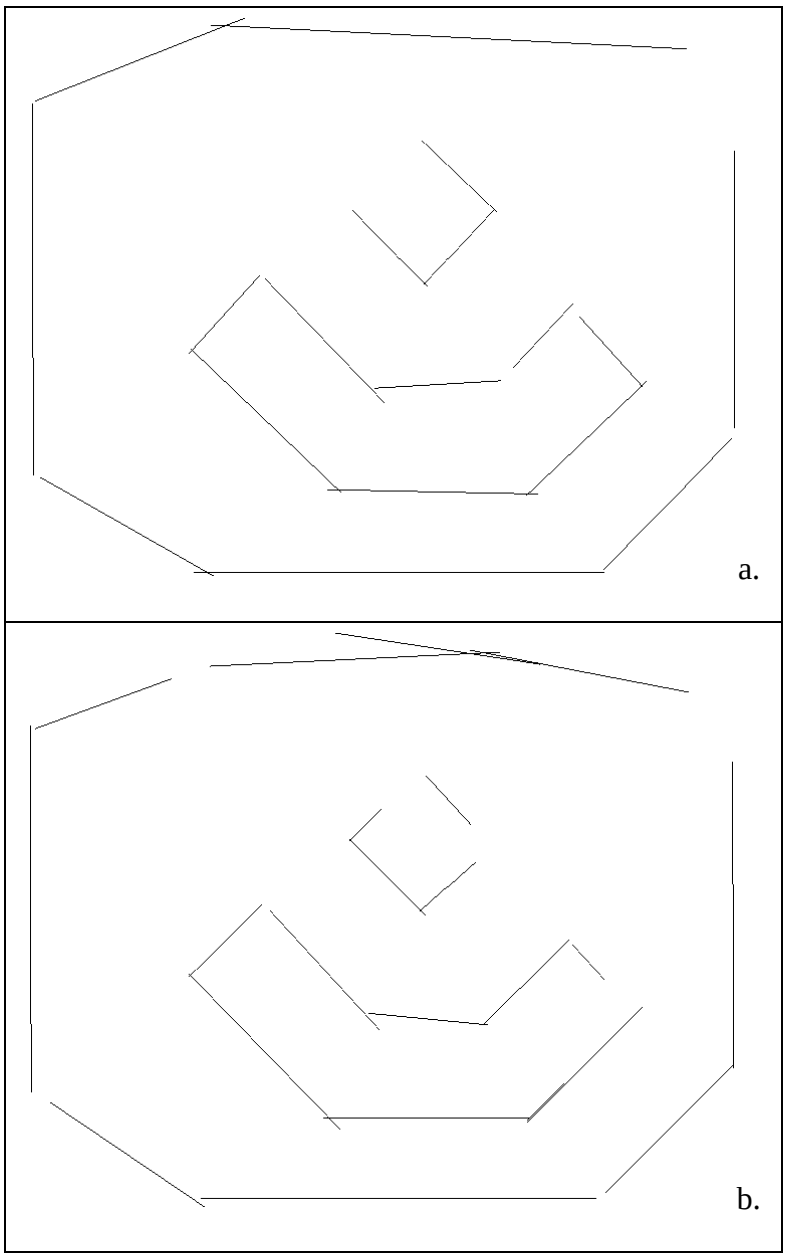
Şekil 7.6 Ortam 2'de, standart sapması 4 cm olan algılayıcı gürültüsü varken duvar takibi ile çıkarılan haritalar.
a. Tek Robotla İki Noktadan Doğru Çıkarımı Yaparak Haritalama
b. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama



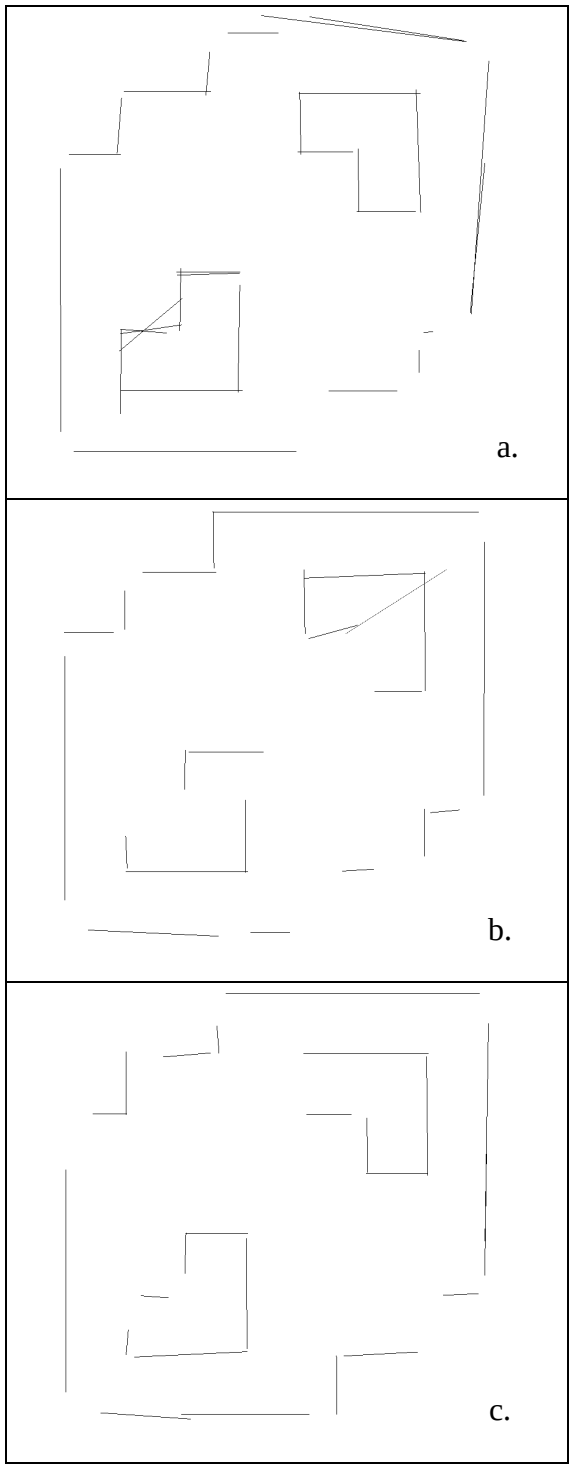
Şekil 7.7 Ortam 1'de, gürültü yokken keşif stratejisi kullanıldığında çıkarılan haritalar.
a. Tek Robotla İki Noktadan Doğru Çıkarımı Yaparak Haritalama
b. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama
c. Robot Takımı ile Haritalama



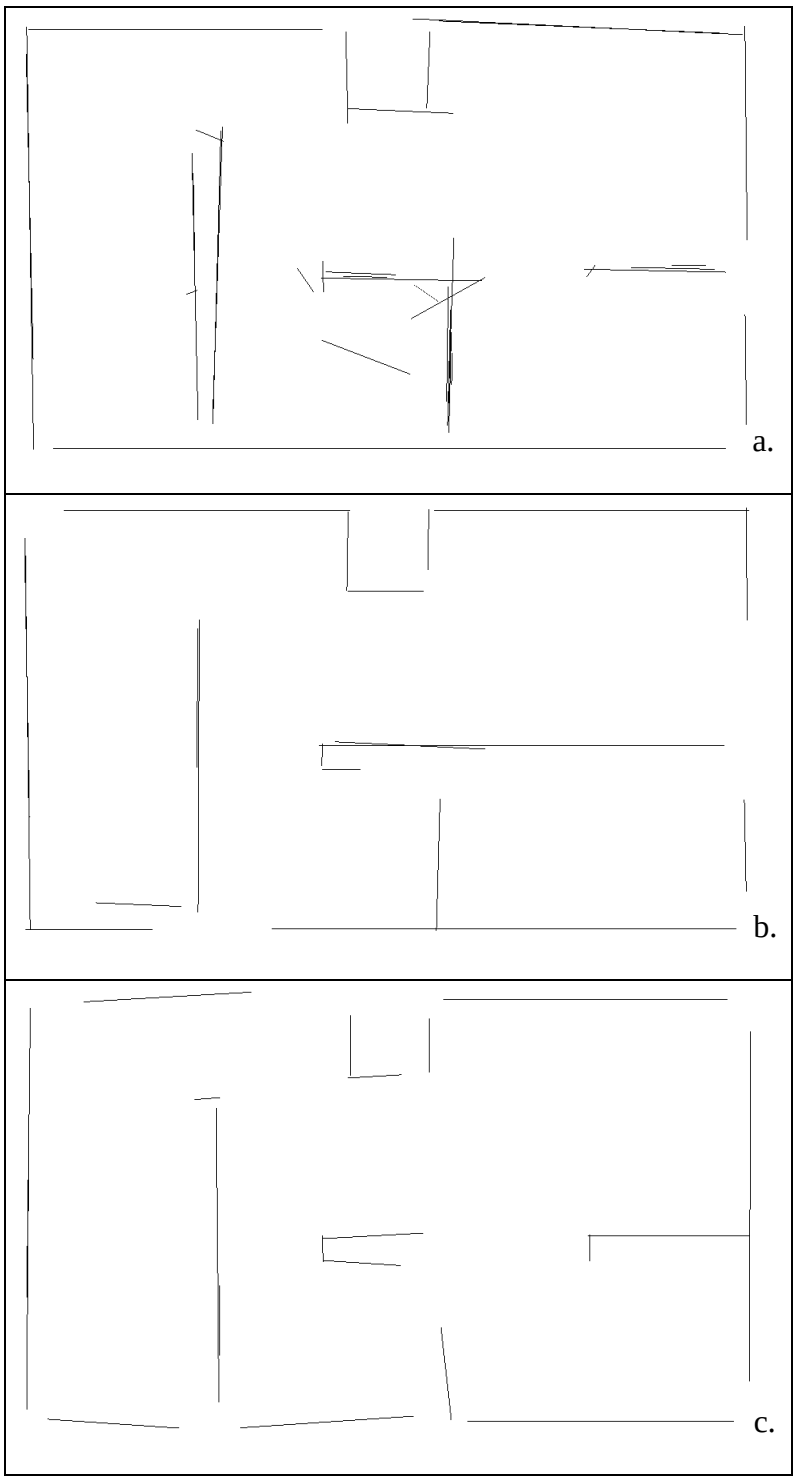
Şekil 7.8 Ortam 2'de, gürültü yokken keşif stratejisi kullanıldığında çıkarılan haritalar.
a. Tek Robotla İki Noktadan Doğru Çıkarımı Yaparak Haritalama
b. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama
c. Robot Takımı ile Haritalama



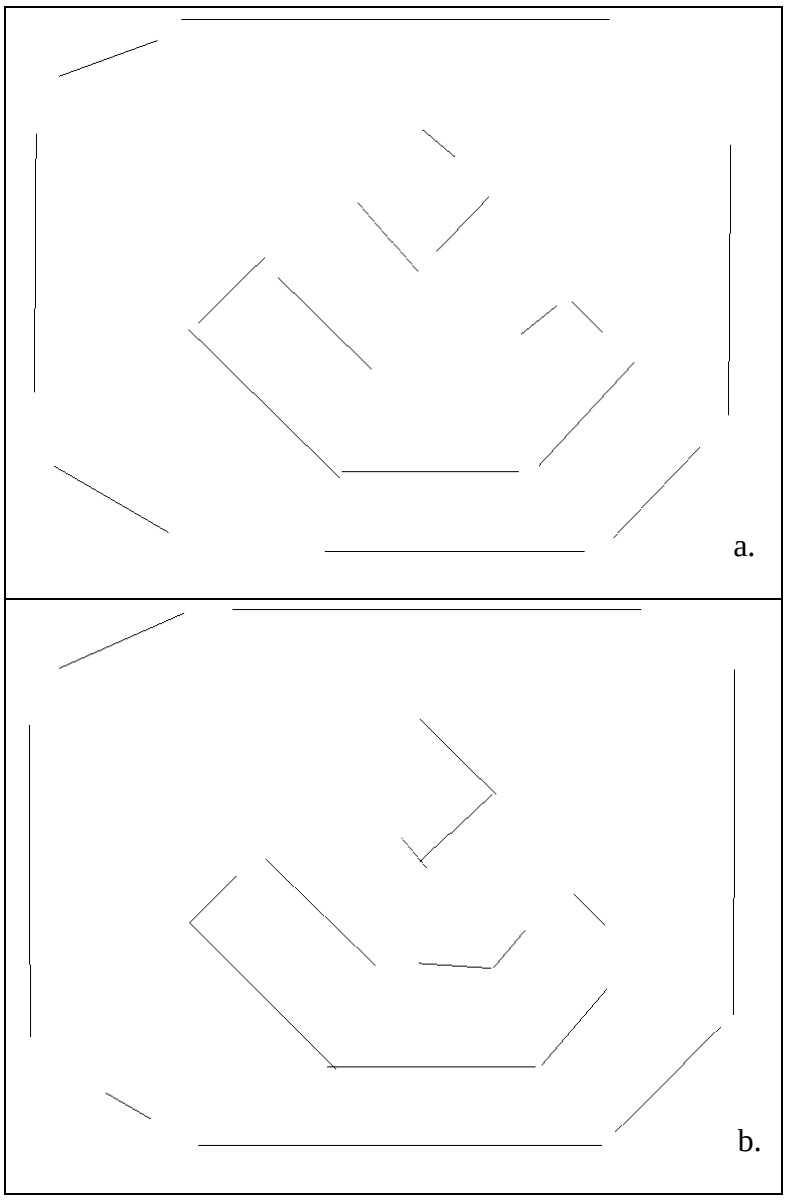
Şekil 7.9 Ortam 3'te gürültü yokken keşif stratejisi ile elde edilen haritalar.
a. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama
b. Robot Takımı ile Haritalama



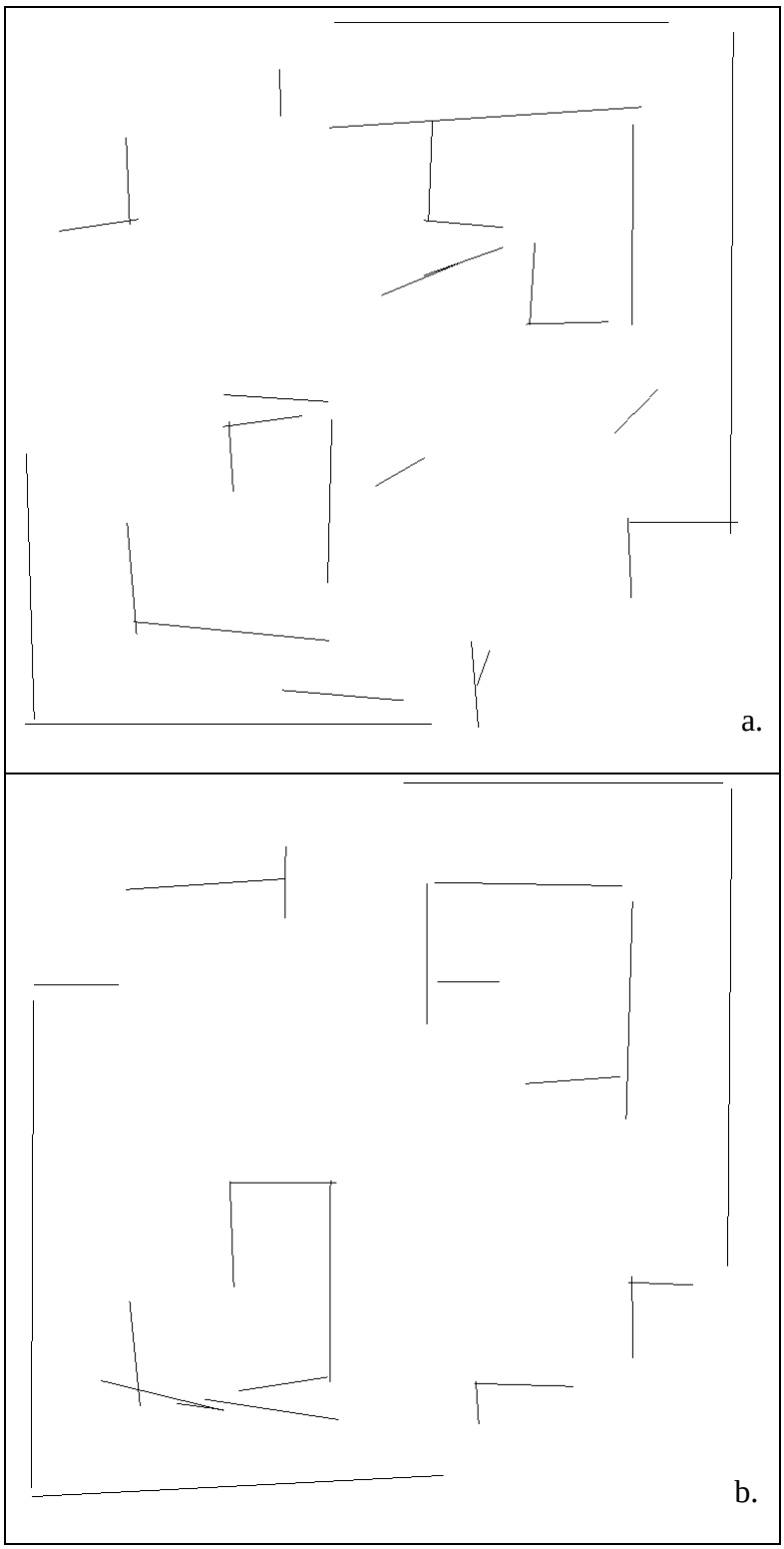
Şekil 7.10 Ortam 1’de, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında çıkarılan haritalar.
a. Tek Robotla İki Noktadan Doğru Çıkarımı Yaparak Haritalama
b. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama
c. Robot Takımı ile Haritalama



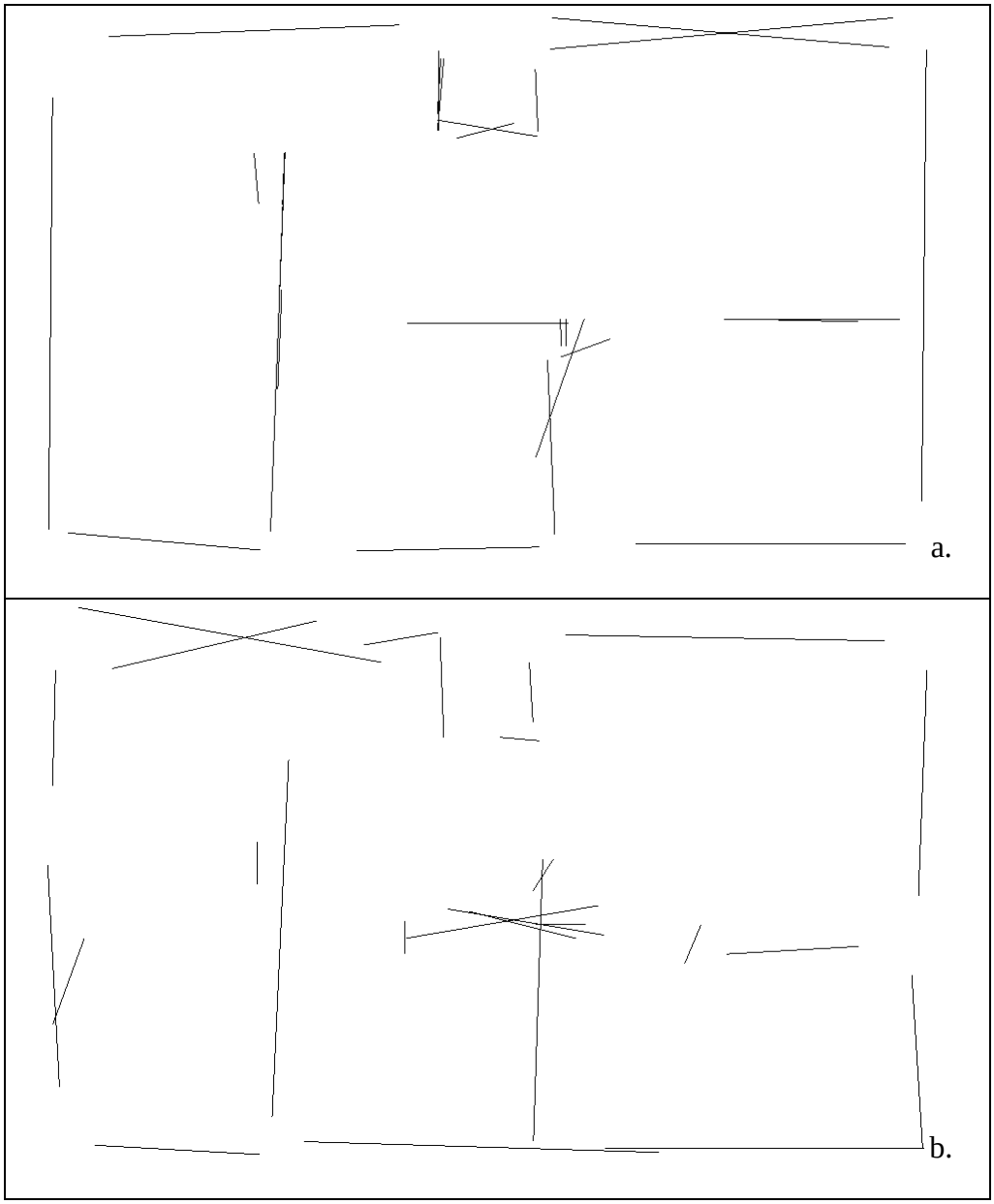
Şekil 7.11 Ortam 2'de, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında çıkarılan haritalar.
a. Tek Robotla İki Noktadan Doğru Çıkarımı Yaparak Haritalama
b. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama
c. Robot Takımı ile Haritalama



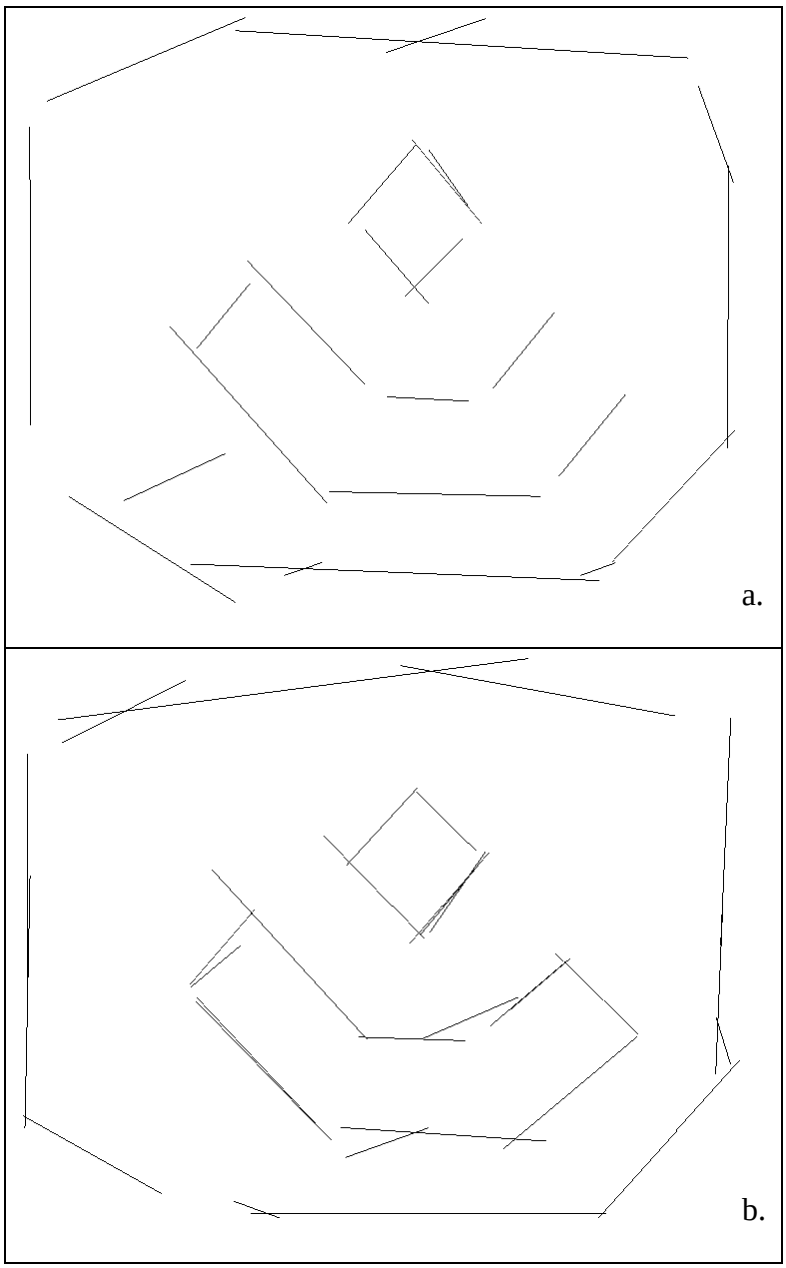
Şekil 7.12 Ortam 3'te, standart sapması 2 cm olan algılayıcı gürültüsü varken keşif stratejisi ile elde edilen haritalar.
a. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama
b. Robot Takımı ile Haritalama



Şekil 7.13 Ortam 1'de, standart sapması 4 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında çıkarılan haritalar.
a. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama
b. Robot Takımı ile Haritalama



Şekil 7.14 Ortam 2'de, standart sapması 4 cm olan algılayıcı gürültüsü varken keşif stratejisi kullanıldığında çıkarılan haritalar.
a. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama
b. Robot Takımı ile Haritalama



Şekil 7.15 Ortam 3'te, standart sapması 4 cm olan algılayıcı gürültüsü varken keşif stratejisi ile elde edilen haritalar.
a. Tek Robotla Hough Dönüşümü ile Doğru Çıkarımı Yaparak Haritalama
b. Robot Takımı ile Haritalama

Bundan sonra yapılacak çalışmalarda keşfin daha verimli olabilmesi, çıkarılan haritanın da daha düzgün olması için duvar takibi ile keşfedilmemiş alana yönelme yöntemlerini birleştiren bir keşif stratejisi oluşturulması; sınırlı algılayıcı ve doğrusal yer işaretçileri kullanılırken robot konumunda gerçekleşen hatayı düzeltmede en büyük etkiyi gösterebilecek olan döngü kapatma metodunun iyileştirilmesi; noktasal verilere doğru uydururken belirsizliğin de ele alınması başarıda önemli bir artışa neden olacaktır.

KAYNAKLAR

Andersson, L.A.A. ve Nygards, J., (2008), "C-SAM: Multi-Robot SLAM using Square Root Information Smoothing", IEEE International Conference on Robotics and Automation, May 19-23, CA, USA.

Arras, K. O. ve Siegwart, R., (1997), "Feature Extraction and Scene Interpretation for Map-Based Navigation and Map Building", Symposium on Intelligent Systems and Advanced Manufacturing, October 14, Pittsburgh, USA.

Beevers, K.R., Huang, W.H., (2006), "SLAM with Sparse Sensing", IEEE International Conference on Robotics and Automation (ICRA 2006), May 15-19, Orlando.

Beevers, K.R., (2007), Mapping With Limited Sensing, PhD Thesis, Rensselaer Polytechnic Institute.

Borges, G.A. ve Aldon, M.-J., (2000), "A Split-and-Merge Segmentation Algorithm for Line Extraction in 2-D Range Images", International Conference on Pattern Recognition, September 3-8, Barcelona.

Borges, G.A. ve Aldon, M.-J., (2004), "Line Extraction in 2D Range Images for Mobile Robotics", Journal of Intelligent and Robotic Systems, 40:267–297.

Burgard, W., Fox, D., Moors, M., Simmons, R. ve Thrun, S., (2000), "Collaborative Multi-Robot Exploration", IEEE International Conference on Robotics and Automation, April 24-28, San Francisco.

Burguera, A., Oliver, G. ve Tardós, J. D., (2005), "Robust Scan Matching Localization Using Ultrasonic Range Finders", IEEE/RSJ International Conference on Intelligent Robots and Systems, August 2-6, Edmonton, Canada.

Castellanos, J. A. ve Tardós, J. D, (1996), "Laser-based Segmentation and Localization for a Mobile Robot", Robotics and Manufacturing: Recent Trends in Research and Applications, 6, M. Jamshidi, F. Pin ve P. Dauchez (Derl.), ASME Press.

Chang, H.J., Lee, C.S.G., Hu, Y.C. ve Lu,Y.-H., (2007), "Multi-Robot SLAM with Topological/Metric Maps", IEEE/RJS International Conference on Intelligent Robots and Systems, San Diego, CA, USA.

Choi, J., Ahn, S. ve Chung, W.K., (2005), "Robust Sonar Feature Detection for the SLAM of Mobile Robot," IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, Alberta, Canada.

Choi, J., Ahn, S., Choi, M. ve Chung, W.K., (2006), "Metric SLAM in Home Environment with Visual Objects and Sonar Features", IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China.

Choi, Y.-H., Lee, T.-K. ve Oh S.-Y., (2008), "A Line Feature Based SLAM with Low Grade Range Sensors Using Geometric Constraints and Active Exploration for Mobile Robot", Autonomous Robots, 24:13–27.

Choset, H., Nagatani, K. ve Lazar, N. A., (2003), "The Arc-Transversal Median Algorithm: A Geometric Approach to Increasing Ultrasonic Sensor Azimuth Accuracy", IEEE Transactions on Robotics and Automation, 19(3):513–523.

Dijkstra, E. W., (1959), "A note on two problems in connexion with graphs", Numerische

Mathematik 1:269–271.

Duda, R. O. ve Hart, P. E., (1972), "Use of the Hough Transformation to Detect Lines and Curves in Pictures," Comm. ACM, 15:1–15.

Einsele, T., (1997), “Real-Time Self-Localization in Unknown Indoor Environments using a Panorama Laser Range Finder”, IEEE/RSJ International Conference on Intelligent Robots and Systems, Grenoble, France.

Fox, D., (2003), “Adapting the Sample Size in Particle Filters Through KLD-Sampling”, International Journal of Robotics Research, 22:985-1003.

Garulli, A., Giannitrapani, A., Rossi, A., Vicino, A., (2005a) “Simultaneous Localization and Map Building using Linear Features”, European Conference on Mobile Robots, Ancona, Italy.

Garulli, A., Giannitrapani, A., Rossi, A., Vicino, A., (2005b) “Mobile Robot SLAM for Line-based Environment Representation”, IEEE Conference on Decision and Control, and the European Control Conference, December 12-15, Sevile, Spain.

Großmann, A. ve Poli, R., (1999), "Robust Mobile Robot Localisation from Sparse and Noisy Proximity Readings", International Joint Conference on Artificial Intelligence, Workshop on Reasoning with Uncertainty in Robot Navigation (RUR-99), Stockholm, Sweden.

Großmann, A. ve Poli, R., (2001), "Robust Mobile Robot Localisation From Sparse and Noisy Proximity Readings using Hough Transform and Probability Grids", Robotics and Autonomous Systems, 37:1-18.

Howard A. ve Roy N., (2003), The Robotics Data Set Repository (Radish).

Je, H. ve Kim, D., (2008), "Simultaneous Localization and Mapping Based on PF-MDS", ECSIS Symposium on Learning and Adaptive Behaviors for Robotic Systems, Scotland, UK.

Ko, J., Stewart, B., Fox, D., Konolige, K. ve Limketkai, B., (2003), "A Practical, Decision-theoretic Approach to Multi-Robot Mapping and Exploration", IEEE/RSJ International Conference on Intelligent Robots and Systems,Las Vegas, Nevada, USA.

Kobayashi, F., Sakai, S. ve Kojima, F., (2003), “Determination of Exploration Target Based on Belief Measure in Multi-Robot Exploration”, IEEE International Symposium on Computational Intelligence in Robotics and Automation, Kobe, Japan.

Kultanen, P., Xu, L. ve Oja, E., (1990), “Randomized Hough transform (RHT),” International Conference on Pattern Recognition, Atlantic City, USA.

Lakaemper, R., Latecki, L.J. ve Wolter, D., (2005), "Incremental Multi-Robot mapping", IEEE/RSJ International Conference on Intelligent Robots and Systems, Alberta, Canada.

Michel, O., (2004), “Webots: Professional Mobile Robot Simulation”, Journal of Advanced Robotics Systems, 1(1):39-42.

Neira, J. ve Tardós, J.D., (2001), “Data Association in Stochastic Mapping using the Joint Compatibility Test”, IEEE Transactions on Robotics and Automation, 17(6):890–897.

Nguyen, V., Martinelli, A., Tomatis, N. ve Siegwart, R, (2005), “A Comparison of Line Extraction Algorithms using 2D Laser Rangefinder for Indoor Mobile Robotics”, IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, Alberta, Canada.

Pavlidis, T. ve Horowitz, S.L., (1974), "Segmentation of Plane Curves", IEEE Transactions on Computers, C-23(8):860–870.

Pfister, S.T., Roumeliotis, S.I. ve Burdick, J.W., (2003), "Weighted Line Fitting Algorithms for Mobile Robot Map Building and Efficient Data Representation", IEEE International Conference on Robotics and Automation, Taipei, Taiwan.

Siegwart, R. ve Nourbakhsh, I.R., (2004), Introduction to Autonomous Mobile Robots, The MIT Press, London, England.

Tardós, J.D., Neira, J., Newman, P. M. ve Leonard, J. J. (2002), "Robust Mapping and Localization in Indoor Environments using Sonar Data", The International Journal of Robotics Research, 21(4):311–330.

Thrun, S., Burgard, W. ve Fox, D., (2005), Probabilistic Robotics, The MIT Press, Cambridge, MA, USA.

Wijk, O. ve Christensen, H.I., (2000), "Triangulation-Based Fusion of Sonar Data with Application in Robot Pose Tracking", IEEE Transactions on Robotics and Automation, 16(6):740–752.

Yamauchi, B., (1997), "A Frontier-Based Approach for Autonomous Exploration", IEEE International Symposium on Computational Intelligence in Robotics and Automation, Monterey, Kaliforniya, ABD.

Yamauchi, B., (1998), "Frontier-Based Exploration Using Multiple Robots", International Conference on Autonomous Agents, Minneapolis, Minnesota, ABD.

Yap T.N. ve Shelton C.R., (2009), "SLAM in Large Indoor Environments with Low-Cost, Noisy, and Sparse Sonars", IEEE International Conference on Robotics and Automation, Kobe, Japan.

Yavuz, S., Amasyalı, M. F., Balcılar, M., Bilgin, G., Dinç, T. ve Kurt, Z., (2006), "Eş Zamanlı Konum Belirleme ve Harita Oluşturma Amaçlı Otonom Bir Robot", Eleco 2006 Elektrik, Elektronik, Bilgisayar Mühendisliği Sempozyumu ve Fuarı, 6-10 Aralık, Bursa, Türkiye.

Yun, X., Latt, K. ve Glennon, J.S, (1998), "Mobile Robot Localization using the Hough Transform and Neural Networks", IEEE International Symposium on Intelligent Control (ISIC/CIRA/ISAS Joint Conf.), Gaithersburg, Maryland, USA.

Zhao, J., Jiang, J. ve Zang, X., (2006), "Cooperative Multi-Robot Map-building based on Genetic Algorithms", IEEE International Conference on Information Acquisition, Weihai, Shandong, China.

Zhou, X.S. ve Roumeliotis, S.I., (2006), "Multi-Robot SLAM with Unknown Initial Correspondence: The Robot Rendezvous Case", IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China.

Zlot, R., Stentz, A., Dias, M.B. ve Thayer, S., (2002), "Multi-Robot Exploration Controlled by a Market Economy", IEEE International Conference on Robotics and Automation, Washington DC, USA.

EKLER

Ek 1 Genişletilmiş Kalman Filtresi Genel Algoritması

Ek 2 Genişletilmiş Kalman Filtresi Algoritması Uygulaması

Ek 3 Kitapta Söz Edilen Terimlerin Türkçe ve İngilizce Adları

Ek 1 Genişletilmiş Kalman Filtresi Genel Algoritması

GKF_Algoritmasi($\mu_{t-1}, \sum_{t-1}, u_t, z_t, N_{t-1}$)

1: $N_t = N_{t-1}$

2: $F_x = \begin{pmatrix} 1 & 0 & 0 & 0...0 \\ 0 & 1 & 0 & 0...0 \\ 0 & 0 & 1 & 0...0 \end{pmatrix}$

3: $\overline{\mu}_t = \mu_{t-1} + F_x^T Mtr_{kinematik}$

4: $G_t = I + F_x^T g_t F_x$

5: $R_t = V_t M_t V_t^T$

6: $\overline{\sum}_t = G_t \sum_{t-1} G_t^T + F_x^T R_t F_x$

7: $\overline{\mu}_t$ *ve* $\overline{\sum}_t$ *matrislerinin boyutunu eklenecek yeni yer işaretçisi için büyüt*

8: Döngü_1: *gözlenen tüm* z_t^i *yer işaretçileri için yap*

8.1: *yer işaretçisinin mutlak (küresel koordinat düzlemine göre) koordinatını bul.*

8.2: *yer işaretçisini* $\overline{\mu}_t$ *matrisine ekle,* $\overline{\sum}_t$ *matrisinde de bu yer işaretçisi için ilklendirme yap.*

8.3: Döngü_2: *k=1'den* $(N_t + 1)$*'e* $\overline{\mu}_t$ *matrisindeki tüm yer işaretçileri için yap*

8.3.1: *robotun k. yer işaretçisini görüyor olsa nasıl bir ölçüm değeri elde edeceğine dair hesaplanan değerlerden oluşan beklenen ölçüm matrisini hesapla (* $\hat{z}_t^k$ *).*

8.3.2: $F_{x,k} = \begin{pmatrix} 1 & 0 & 0 & 0...0 & 0 & 0 & 0...0 \\ 0 & 1 & 0 & 0...0 & 0 & 0 & 0...0 \\ 0 & 0 & 1 & 0...0 & 0 & 0 & 0...0 \\ 0 & 0 & 0 & 0...0 & 1 & 0 & 0...0 \\ 0 & 0 & 0 & 0...0 & 0 & 1 & 0...0 \end{pmatrix}$

8.3.3: $H_t^k = h_t F_{x,k}$

8.3.4: $\psi_k = H_t^k \overline{\sum}_t \left[H_t^k\right]^T + Q_t$

8.3.5: $\pi_k = \left(z_t^i - \hat{z}_t^k\right)^T \psi_k^{-1} \left(z_t^i - \hat{z}_t^k\right)$

8.4: $\pi_{N_t+1} = \alpha$

8.5: $j(i) = \arg\min_k \pi_k$

8.6: Koşullu Kontrol:

8.6.1: *Eğer* $j(i) > N_t$ *ise*

8.6.1.1: $N_t = N_t + 1$

8.6.1.2: $\overline{\mu}_t$ *ve* $\overline{\sum}_t$ *matrislerinin boyutunu eklenecek yeni yer işaretçisi için büyüt*

8.6.2: *Eğer* $j(i) > N_t$ *değilse*

8.6.2.1: $K_t^i = \overline{\sum}_t \left[H_t^{j(i)}\right]^T \psi_{j(i)}^{-1}$

8.6.2.2: $\overline{\mu}_t = \overline{\mu}_t + K_t^i (z_t^i - \hat{z}_t^{j(i)})$

8.6.2.3: $\overline{\sum}_t = \left(I - K_t^i H_t^{j(i)}\right) \overline{\sum}_t$

9: $\mu_t = \overline{\mu}_t$

10: $\sum_t = \overline{\sum}_t$

11: μ_t *ile* $\sum_t$ *'yi geri döndür.*

Açıklamalar:

Genişletilmiş Kalman Filtresi (GKF), Kalman Filtresi'nin doğrusal olmayan fonksiyonlarda da kullanılmasını sağlayan bir metottur, robotun hareket ve algı fonksiyonları doğrusal olmadığından robotik problemlerinde de kullanılır.

GKF'de doğrusal olmayan fonksiyonları doğrusallaştırmak için birinci dereceden Taylor açılımı kullanılır. Aşağıda Taylor açılımının genel hali görülmektedir:

$$f(x+a) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \dots$$

Burada kullanılan a, komşuluğu dahilinde f fonksiyonunun sonsuz sayıda türevinin alınabildiği reel ya da karmaşık bir sayıdır. GKF'de fonksiyon parametresi olarak en olası değer olan μ_{t-1} değeri kullanılır.

Robotun hareket fonksiyonu olan $g(u_t, x_{t-1})$ fonksiyonunu ele alırsak:

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + g'(u_t, \mu_{t-1})(x_{t-1} - \mu_{t-1})$$

$$g(u_t, x_{t-1}) = g(u_t, \mu_{t-1}) + G_t(x_{t-1} - \mu_{t-1})$$

olur. Aynı işlemi algı fonksiyonu için uygularsak:

$$h(x_t) \approx h(\overline{\mu}_t) + h'(\overline{\mu}_t)(x_t - \overline{\mu}_t)$$

$$h(x_t) = h(\overline{\mu}_t) + H_t(x_t - \overline{\mu}_t)$$

olur.

Burada yer verilen Genişletilmiş Kalman Filtresi algoritmasında adımlar mümkün olduğunca matematiksel olarak ifade edilmeye çalışılmıştır fakat bazı hesaplar tamamen kullanılan algılayıcı, efektör vs. (ve bunların nasıl modellendiği) ile ilgili olduğu için bu hesaplar sözel olarak ifade edilmiştir.

Algoritmada giriş olarak μ_{t-1}, $\sum_{t-1}$, u_t, z_t ve N_{t-1} değerleri alınmaktadır. μ_{t-1}, robotların ve yer işaretçilerinin konum bilgilerini ifade eden Gauss olasılık dağılımlarının ortalama değerini

tutan sütun matristir. Üst kısımda robotların (x, y, θ) bilgileri tutulur, daha sonra ise yer işaretçilerinin bilgileri gelir. $\sum_{t-1}$, robotların ve yer işaretçilerinin konum bilgilerini ifade eden Gauss olasılık dağılımlarının kovaryanslarının tutulduğu matristir. u_t, aslında robota gönderilen kontrol işaretidir fakat pek çok uygulamada robota gönderilen komutlar değil, robottan dönen odometri verisi kullanılır. z_t, robotun yaptığı gözlemlerdir. N_{t-1} ise bir önceki adımda elde tutulan yer işaretçisi sayısıdır.

Bu algoritmada 1. adım ile 6. adım arası hareket güncellemesidir. Burada ortalama değerler matrisi (μ_t) ile kovaryans matrisi $(\sum_t)$ üzerinde kinematik denklemler ve hareket belirsizlikleri kullanılarak güncellemeler yapılmaktadır. Bu matrislerin hareket güncellemesi yapılmış halleri üzerleri çizili olarak ifade edilmektedir ($\overline{\mu}_t$, $\overline{\sum}_t$). 7. ve 8. adımlarda algı güncellemesi yapılmaktadır. Bu adımlarda z_t gözlemleri ile daha önceden görülmüş ve durum uzayına eklenmiş yer işaretçileri karşılaştırılmakta, eşleşme bulunduğunda yer işaretçisinin görülmesi beklenen yer ile görüldüğü yer arasındaki fark kullanılarak durum güncellenmekte, eşleşme bulunamazsa bu yer işaretçisi yeni bir yer işaretçisi olarak mevcut duruma eklenmektedir.

Algoritmada her adımda yapılan işlemler aşağıda anlatılmaktadır:

1: Bir önceki adımdaki yer işaretçisi sayısı şimdiki yer işaretçisi sayısı olarak alınmaktadır.

2: F_x matrisi tanımlanmaktadır. Bu matris, $Mtr_{kinematik}$ matrisini μ_{t-1} matrisi ile toplanabilecek şekilde uygun boyutlu hale getiren ve çarpma işlemi sonucunda μ_{t-1}'de yalnızca ilgili yerlerin (ele alınan robotun konum bilgilerinin) değişmesini sağlayan matristir. Boyutları $3\times(3\times RobotSayisi + 2\times N_t)$ şeklindedir. Burada $Mtr_{kinematik}$ matrisi, ilgili robotun konum parametrelerinin her birinde hareket sonucu gerçekleştiği kinematik denklemler kullanılarak hesaplanan değişimdir.

3: μ_{t-1} matrisine hareket sonucu robotun konumunda gerçekleşen değişimler eklenmekte ve $\overline{\mu}_t$ matrisi elde edilmektedir.

4: Robotun hareketi sonucu kovaryans matrisinde ($\sum_{t-1}$) gerçekleşen dönüşümü (*transformasyon*) hesaplamak için kullanılan G_t matrisi hesaplanmaktadır. Burada G_t, daha

önce de ifade edildiği gibi $g(u_t, \mu_{t-1})$ fonksiyonunun türevidir.

$$g(u_t, \mu_{t-1}) = \mu_{t-1} + Mtr_{kinematik}$$

olduğu için bunun μ_{t-1}'e (yani x, y ve θ parametrelerine) göre türevi

$$g'(u_t, \mu_{t-1}) = I + Mtr'_{kinematik}$$

şeklinde olmaktadır. $Mtr'_{kinematik}$ de burada g_t ile ifade edilmektedir. F_x yine matrislerin tüm durum matrisi boyutuna uygun olması için kullanılmaktadır. I ise boyutu kullanıldığı yerdeki diğer matrislerle uyumlu olacak şekilde ayarlanan birim matristir.

5: Hareket sonucu durum uzayında oluşan belirsizliği ifade eden R_t matrisi hesaplanmaktadır. Burada M_t, kontrol uzayındaki hareket gürültüsünü ifade eden kovaryans matrisidir. Bu gürültünün durum uzayına aktarımı doğrusal bir yaklaşımla yapılır. Bu yaklaşım için gereken jacobian matrisi olan V_t, hareket fonksiyonunun u_t ve μ_{t-1}'de hareket parametrelerine göre türevidir.

6: Duruma ait kovaryans matrisinde hareket güncellemesi yapılmaktadır. Önceki adımdan gelen kovaryans matrisinde dönüşüm uygulanmakta, buna hareket gürültüsü eklenmektedir.

7: Bu algoritmada gözlemler elde tutulan yer işaretçileri ile kıyaslanırken öncelikle mutlaka yeni bir yer işaretçisi gibi durum matrisine eklenmektedir, gözlem ile (aslında zaten gözlemin kendisi olan) yer işaretçisi arasındaki mesafe olarak ise belli bir eşik değer atanmaktadır. Eğer gözleme daha yakın bir yer işaretçisi bulunursa gözlemle bu yer işaretçisi eşleşmiş olur, güncellemeler yapılır, son eklenen yer işaretçisini dışarıda bırakacak şekilde matris boyutları küçültülür. Eğer gözleme en yakın yer işaretçisi son eklenen yer işaretçisi olursa yeni bir yer işaretçisi görülmüş demektir, bu yer işaretçisi korunur. Bu adımda gerçekleştirilen $\bar{\mu}_t$ ve $\bar{\Sigma}_t$ matrislerinin büyütülmesi işte bu yer işaretçisinin yerleştirilmesi içindir.

8: Döngü başlamaktadır, tüm gözlemler için kıyaslama yapılacaktır.

8.1: Robot tarafından yapılan gözlemler robotun yerel koordinat düzlemine göredir. Bu adımda gözlenen yer işaretçisinin küresel koordinat düzlemine göre koordinatı bulunmaktadır.

8.2: Yer işaretçisi $\bar{\mu}_t$ matrisine eklenir. $\bar{\Sigma}_t$ kovaryans matrisinde de bu yer işaretçisi için belirsizlik ilklendirilir.

8.3: Döngü başlamaktadır, elde tutulan tüm yer işaretçilerine bakılacaktır.

8.3.1: Robotun, k. yer işaretçisini görüyor olsa nasıl bir ölçüm değeri elde edeceğine dair hesaplanan değerlerden oluşan beklenen ölçüm matrisini hesapla ($\hat{z}_t^k$).

8.3.2: Matris çarpımında boyutların tutmasını ve yalnızca ilgili robot konumu ve yer işaretçisinde değişiklik yapılmasını sağlayan $F_{x,k}$ matrisi tanımlanmaktadır.

8.3.3: Durum uzayındaki belirsizliğin ölçüm uzayındaki karşılığını hesaplamada kullanılan H_t^k matrisi hesaplanmaktadır. H_t^k matrisi, yer işaretçilerinin mutlak ve göreceli konumları arasında dönüşüm yapan fonksiyonun tüm durum vektörüne göre kısmi türevidir. Bu fonksiyon yalnızca ilgili robot ile yer işaretçisine bağlı olduğundan türev, h_t jacobian'ı ile h_t jacobian'ının durum uzayı boyutlarında matrise aktarılmasını sağlayan $F_{x,k}$ matrisinin çarpımına dönüşür. Burada h_t, yer işaretçilerinin mutlak ve göreceli konumları arasında dönüşüm yapan fonksiyonun robot konumu (x, y, θ) ve mutlak yer işaretçisi konumu parametrelerine $(r,\ \theta)$ göre kısmi türevidir.

8.3.4: Gözlem ile elde tutulan yer işaretçisi karşılaştırılırken kullanılacak toplam belirsizlik hesaplanmaktadır. Burada Q_t ölçüm gürültüsüdür.

8.3.5: z_t^i gözlemi ile $\hat{z}_t^k$ beklenen ölçümü arasındaki Mahalanobis mesafesi hesaplanmaktadır.

8.4: Gözlem ile aslında zaten gözlemin kendisi olan yeni eklenen yer işaretçisi arasındaki mesafe olarak bir eşik değer kabul ediliyor.

8.5: Gözleme en yakın yer işaretçisi bulunuyor.

8.6: Koşullu kontrol

8.6.1: Eğer en yakın yer işaretçisi yeni eklenen yer işaretçisi ise

8.6.1.1: Yer işaretçisi sayısı bir artırılıyor.

8.6.1.2: $\bar{\mu}_t$ ve $\overline{\sum}_t$ matrislerinin boyutları bundan sonra eklenecek yer işaretçisi için büyütülür.

8.6.2: Eğer en yakın yer işaretçisi daha önce eklenen yer işaretçilerinden biri ise

8.6.2.1: Yeni gözlemden elde edilen bilginin durum üzerinde ne kadar etkili olacağını

belirleyen Kalman kazanımı hesaplanır.

8.6.2.2: Kalman kazanımı ve yer işaretçisinin beklenen yeri ile gözlenen yeri arasındaki fark kullanılarak durum ortalama değer matrisinde güncelleme yapılır.

8.6.2.3: Kalman kazanımı kullanılarak durum kovaryans matrisinde güncelleme yapılır, belirsizlikler azalır.

9: $\overline{\mu}_t$ matrisi μ_t matrisine atanır.

10: $\overline{\sum}_t$ matrisi $\sum_t$ matrisine atanır.

11: μ_t ve $\sum_t$ matrisleri geri döndürülür.

Ek 2 Genişletilmiş Kalman Filtresi Algoritması Uygulaması

Burada GKF'nin bu çalışmadaki uygulama detayları verilmiştir.

1. Giriş:

Uygulama ortalama değer matrisinin r robot sayısı olmak üzere $(r\times 3)\times 1$ şeklinde boyutlandırılması ve robotların ilk konumları ile ilklendirilmesi ile başlamaktadır. Kovaryans matrisi de $(r\times 3)\times(r\times 3)$ şeklinde boyutlandırılmaktadır ve robotların ilk konumlarını kesin olarak bildiğimiz için 0 ile ilklendirilmektedir.

$$\mu_0 = \begin{pmatrix} x_1 & y_1 & \theta_1 & x_2 & y_2 & \theta_2 & \dots & x_r & y_r & \theta_r \end{pmatrix}^T$$

$$\sum_0 = \begin{pmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{pmatrix}$$

Algılayıcı konumları da robot konumlarına göre her adımda yeniden hesaplanıp tutulmaktadır.

Bu çalışmada merkezi kontrol mekanizması kullanılmıştır, tüm robotlar 6 kızılötesi algılayıcılarından aldıkları verileri ve iki tekerlerinden ayrı ayrı aldıkları odometri verilerini merkez bilgisayara göndermektedirler.

u_t kontrol işareti olarak robotların gönderdiği odometri verisi kullanılmaktadır.

GKF'ye algılayıcı verileri doğrudan gönderilmektedir, bu verilerden doğrusal özelliklerin çıkarımı, yani z_t gözlemlerinin elde edilmesi GKF içinden çağrılan bir metot ile halledilmektedir. GKF'nin bir gözlem olsun ya da olmasın her durumda çağrılmasının nedeni hareket güncelleme işleminin de GKF içinde yer almasıdır. Ayrıca Hough Dönüşümü uygulanacak noktasal verilerin her adımda elde edilmesi, kümeye eklenmesi gerekmektedir. GKF her robot için ayrı ayrı çalıştırılır.

2. Hareket Güncellemesi

Hareket güncellemesinde ortalama değer matrisi odometri verileri ve kinematik denklemler kullanılarak güncelleştirilmekte, kovaryans matrisi de dönüşüm matrisi ve hareket gürültüsü

ile güncellenmektedir.

2.1 $Mtr_{kinematik}$ matrisi:

Uygulamada kullanılan kinematik denklemler robotun düz ya da eğimli gitmesine göre değişmektedir. Robotun sağ ve sol tekerlerden aldığı odometri verileri eşitse yani robot düz gidiyorsa $Mtr_{kinematik}$ matrisi şu değerleri almaktadır:

$$Mtr_{kinematik} = \begin{pmatrix} odometri[sol]\cos(\theta_{t-1}) \\ odometri[sol]\sin(\theta_{t-1}) \\ 0 \end{pmatrix}$$

Robotun eğimli gittiği durumda $Mtr_{kinematik}$ matrisi şöyle hesaplanmaktadır:

$$alfa = \frac{odometri[sag] - odometri[sol]}{aksUzunlugu}$$

$$donusYaricapi = \frac{odometri[sag]}{alfa}$$

$$dd = 2 \times \left(donusYaricapi - \frac{aksUzunlugu}{2} \right) \times \sin\left(\frac{alfa}{2} \right)$$

$$Mtr_{kinematik} = \begin{pmatrix} dd \times \cos(\theta_{t-1} + alfa/2) \\ dd \times \sin(\theta_{t-1} + alfa/2) \\ alfa \end{pmatrix}$$

Burada şu belirtilmelidir, tüm uygulama boyunca açılarla ilgili yapılan her işlemden sonra açıların $0 - 2\pi$ aralığında kalması için gerekli işlem yapılmaktadır. Burada da alfa değeri hesaplandıktan sonra bu değerin $0 - 2\pi$ aralığında olup olmadığı kontrol edilmiş, değilse bu aralığa çekilmiştir.

2.2 g_t matrisi:

$Mtr_{kinematik}$ matrisinin (x, y, θ) robot konumu parametrelerine göre kısmi türevi olan g_t matrisi robot düz gidiyorsa

$$g_t = \begin{pmatrix} 0 & 0 & odometri[sol]\sin(\theta_{t-1}) \\ 0 & 0 & -odometri[sol]\cos(\theta_{t-1}) \\ 0 & 0 & 0 \end{pmatrix},$$

eğimli gidiyorsa

$$g_t = \begin{pmatrix} 0 & 0 & dd \times \sin(\theta_{t-1} + alfa/2) \\ 0 & 0 & -dd \times \cos(\theta_{t-1} + alfa/2) \\ 0 & 0 & 0 \end{pmatrix}$$

değerlerini almaktadır. Bu matrislerde sütunlar $Mtr_{kinematik}$ matrisinin ilgili satırının sırasıyla x'e, y'ye ve θ'ya göre kısmi türevini tutmaktadır.

2.3 V_t matrisi:

Hareket uzayındaki gürültünün durum uzayına taşınmasını sağlayan ve $Mtr_{kinematik}$ matrisinin hareket parametrelerine yani sağ ve sol teker için odometri değerlerine göre kısmi türevi olan V_t matrisinin değerleri de robotun düz ya da eğimli gitmesine göre kinematik denklemler değiştiğinden her iki durumda farkı şekillerde hesaplanmaktadır.

Robot düz gidiyorsa

$$V_t = \begin{pmatrix} \cos(\theta_{t-1}) & \cos(\theta_{t-1}) \\ \sin(\theta_{t-1}) & \sin(\theta_{t-1}) \\ 0 & 0 \end{pmatrix}$$

olmaktadır. Robotun eğimli gittiği durumda kinematik denklemler sağ ve sol odometri değerlerine bağlı pek çok değişken içerdiğinden bu hesaplar bira daha karmaşıktır, eşitlikler parçalara ayrılarak türev alınmıştır, türev alma adımları ayrıntılı şekilde aşağıda verilmiştir.

$$Mtr_{kinematik}(0,0) = dd \times \cos\left(\theta_{t-1} + \frac{alfa}{2}\right)$$

$$Mtr_{kinematik}(1,0) = dd \times \sin\left(\theta_{t-1} + \frac{alfa}{2}\right)$$

$$Mtr_{kinematik}(2,0) = alfa$$

$$f = \frac{aksUzunlugu \times (odometri[sag] + odometri[sol])}{odometri[sag] - odometri[sol]}$$

$$g = \sin\left(\frac{odometri[sag] - odometri[sol]}{2 \times aksUzunlugu}\right)$$

$$h = \cos\left(\theta_{t-1} + \frac{odometri[sag] - odometri[sol]}{2 \times aksUzunlugu}\right)$$

$$j = \sin\left(\theta_{t-1} + \frac{odometri[sag] - odometri[sol]}{2 \times aksUzunlugu}\right)$$

$$r = \frac{odometri[sag] - odometri[sol]}{aksUzunlugu}$$

olmak üzere

$$Mtr_{kinematik}(0,0) = f \times g \times h$$

$$Mtr_{kinematik}(1,0) = f \times g \times j$$

$$Mtr_{kinematik}(2,0) = r$$

olur.

Eğer

$$e = odometri[sag] - odometri[sol]$$

$$k = odometri[sag] + odometri[sol]$$

denirse f, g, h, j ve r değerleri de şöyle ifade edilebilir:

$$f = \frac{aksUzunlugu \times k}{e}$$

$$g = \sin\left(\frac{e}{2 \times aksUzunlugu}\right)$$

$$h = \cos\left(\theta_{t-1} + \frac{e}{2 \times aksUzunlugu}\right)$$

$$j = \sin\left(\theta_{t-1} + \frac{e}{2 \times aksUzunlugu}\right)$$

$$r = \frac{e}{aksUzunlugu}$$

Tüm değerlerin kısmi türevleri aşağıdaki gibidir:

$$eTur0 = \frac{\partial e}{\partial odometri[sol]} = -1$$

$$eTur1 = \frac{\partial e}{\partial odometri[sag]} = 1$$

$$kTur0 = \frac{\partial k}{\partial odometri[sol]} = 1$$

$$kTur1 = \frac{\partial k}{\partial odometri[sag]} = 1$$

$$fTur0 == \frac{\partial f}{\partial odometri[sol]} = \frac{aksUzunlugu \times kTur0 \times e - aksUzunlugu \times k \times eTur0}{e^2}$$

$$fTur1 = \frac{\partial f}{\partial odometri[sag]} = \frac{aksUzunlugu \times kTur1 \times e - aksUzunlugu \times k \times eTur1}{e^2}$$

$$gTur0 = \frac{\partial g}{\partial odometri[sol]} = \frac{eTur0}{2 \times aksUzunlugu} \times \cos\left(\frac{e}{2 \times aksUzunlugu}\right)$$

$$gTur1 = \frac{\partial g}{\partial odometri[sag]} = \frac{eTur1}{2 \times aksUzunlugu} \times \cos\left(\frac{e}{2 \times aksUzunlugu}\right)$$

$$hTur0 = \frac{\partial h}{\partial odometri[sol]} = -\frac{eTur0}{2 \times aksUzunlugu} \times \sin\left(\theta_{t-1} + \frac{e}{2 \times aksUzunlugu}\right)$$

$$hTur1 = \frac{\partial h}{\partial odometri[sag]} = -\frac{eTur1}{2 \times aksUzunlugu} \times \sin\left(\theta_{t-1} + \frac{e}{2 \times aksUzunlugu}\right)$$

$$jTur0 = \frac{\partial j}{\partial odometri[sol]} = \frac{eTur0}{2 \times aksUzunlugu} \times \cos\left(\theta_{t-1} + \frac{e}{2 \times aksUzunlugu}\right)$$

$$jTur1 = \frac{\partial j}{\partial odometri[sag]} = \frac{eTur1}{2 \times aksUzunlugu} \times \cos\left(\theta_{t-1} + \frac{e}{2 \times aksUzunlugu}\right)$$

$$rTur0 = \frac{\partial r}{\partial odometri[sol]} = \frac{eTur0}{aksUzunlugu}$$

$$rTur1 = \frac{\partial r}{\partial odometri[sag]} = \frac{eTur1}{aksUzunlugu}$$

$$dx0 = \frac{\partial Mtr_{kinematik}(0,0)}{\partial odometriDegeri[sol]} = fTur0 \times g \times h + f \times gTur0 \times h + f \times g \times hTur0$$

$$dx1 = \frac{\partial Mtr_{kinematik}(0,0)}{\partial odometriDegeri[sag]} = fTur1 \times g \times h + f \times gTur1 \times h + f \times g \times hTur1$$

$$dy0 = \frac{\partial Mtr_{kinematik}(1,0)}{\partial odometriDegeri[sol]} = fTur0 \times g \times j + f \times gTur0 \times j + f \times g \times jTur0$$

$$dy1 = \frac{\partial Mtr_{kinematik}(1,0)}{\partial odometriDegeri[sag]} = fTur1 \times g \times j + f \times gTur1 \times j + f \times g \times jTur1$$

$$dth0 = \frac{\partial Mtr_{kinematik}(2,0)}{\partial odometriDegeri[sol]} = rTur0$$

$$dth1 = \frac{\partial Mtr_{kinematik}(2,0)}{\partial odometriDegeri[sag]} = rTur1$$

$$V_t = \begin{pmatrix} dx0 & dx1 \\ dy0 & dy1 \\ dth0 & dth1 \end{pmatrix}$$

3. Algı Güncellemesi

Algı güncellemesinde öncelikle robotun hareket güncellemesi sonucu elde edilen konumu ve algılayıcı ölçümleri bir doğru çıkarım fonksiyonuna gönderilir. Bu fonksiyonda kullanıcı seçimine göre İki Noktadan Doğru Çıkarımı ile ya da Hough Dönüşümü ile noktalardan doğru elde edilir. Hough Dönüşümü uygulanıyorsa tabi ki önceden belirlenmiş bir sayıda nokta toplanana kadar doğru uydurma işlemi yapılmaz ve kümeye nokta eklemeye devam edilir. Doğru çıkarım fonksiyonunun döndürdüğü gözlemler algı güncellemesi için kullanılır, eğer bir gözlem elde edilemezse algı güncellemesi yapılmaz.

Genel algoritmadan farklı olarak bu uygulamada gözlemler robotun yerel koordinat düzlemine göre değil, küresel koordinat düzlemine göre elde edilmektedir. Çünkü robot hareket halindeyken noktasal verileri toplamaktadır ve bu nokta kümesinden doğru çıkarılmaktadır, bu işlem küresel koordinat düzleminde yapılabilmektedir. Dolayısı ile genel algoritmadakinden farklı olarak gözlemin mutlak koordinatının yani küresel koordinat düzlemindeki konumunun elde edilmesine gerek yoktur fakat göreceli koordinatının yani yerel koordinat düzlemine göre konumunun elde edilmesi gerekmektedir. Gözlemlerin göreceli koordinatlarını ifade etmek için de bu gözlemi yapan algılayıcıdan

3.1 h_t matrisi

Tüm gözlemler için eldeki yer işaretçileri ile karşılaştırma yapılarak bunların daha önceden görülen bir yer işaretçisi mi yoksa yeni bir yer işaretçisi mi olduğunun bulunması gerekmektedir. Karşılaştırma yapılırken yer işaretçilerinin robotun yerel koordinat düzlemindeki konumları kullanılır. Yer işaretçilerinin belirsizliklerinin de yine ölçüm uzayında ifade edilmesi gerektiğinden burada h_t jacobian'ı kullanılarak bir dönüşüm yapılır. h_t jacobian'ı, yer işaretçilerinin mutlak ve göreceli konumları arasında dönüşüm yapan fonksiyonun (x, y, θ) robot konumu ve (r, θ) mutlak yer işaretçisi konumu parametrelerine göre kısmi türevidir.

$$h_t = \begin{pmatrix} \frac{\partial r_{goreceli}}{\partial x} & \frac{\partial r_{goreceli}}{\partial y} & \frac{\partial r_{goreceli}}{\partial \theta_{robot}} & \frac{\partial r_{goreceli}}{\partial r_{mutlak}} & \frac{\partial r_{goreceli}}{\partial \theta_{mutlak}} \\ \frac{\partial \theta_{goreceli}}{\partial x} & \frac{\partial \theta_{goreceli}}{\partial y} & \frac{\partial \theta_{goreceli}}{\partial \theta_{robot}} & \frac{\partial \theta_{goreceli}}{\partial r_{mutlak}} & \frac{\partial \theta_{goreceli}}{\partial \theta_{mutlak}} \end{pmatrix}$$

Bir gözlemin göreceli konumu bulunurken aşağıdaki eşitlik kullanılmaktadır. Bu eşitlik, noktanın doğruya uzaklığı formülünden çıkarılmıştır. Burada x_0 ve y_0, İki Noktadan Doğru Çıkarımı kullanılıyorsa aynı taraftaki iki algılayıcının orta noktası, Hough Dönüşümü kullanılıyorsa doğru uydurulan nokta kümesini oluşturan algılayıcının konumudur. θ_{mutlak}, doğrunun küresel koordinat düzlemine göre, Hough uzayı parametreleri ile ifade edilmiş açısı, $\theta_{goreceli}$ ise robota göre açısıdır. Noktanın doğruya uzaklığı formülü aradaki dik uzaklığı vermektedir fakat bizim bulmak istediğimiz uzaklık algılayıcının ışını göndereceği açıyla ilgilidir, bu nedenle mesafe $\cos\theta_{goreceli}$'ye bölünmektedir.

$$r_{goreceli} = \frac{\left| \frac{\cos\theta_{mutlak}}{\sin\theta_{mutlak}} x_0 + y_0 - \frac{r_{mutlak}}{\sin\theta_{mutlak}} \right|}{\cos\theta_{goreceli} \sqrt{\cot^2\theta_{mutlak} + 1}}$$

$$f = \left| \frac{\cos\theta_{mutlak}}{\sin\theta_{mutlak}} x_0 + y_0 - \frac{r_{mutlak}}{\sin\theta_{mutlak}} \right|$$

$$g = \cos\theta_{goreceli} \sqrt{\cot^2\theta_{mutlak} + 1}$$

$$i = \frac{\cos\theta_{mutlak}}{\sin\theta_{mutlak}} x_0 + y_0 - \frac{r_{mutlak}}{\sin\theta_{mutlak}}$$

$$m = \frac{\cos\theta_{mutlak}}{\sin\theta_{mutlak}} x_0$$

$$n = \frac{\cos\theta_{mutlak}}{\sin\theta_{mutlak}}$$

$$p = \frac{r_{mutlak}}{\sin\theta_{mutlak}}$$

$$j = \sqrt{\cot^2\theta_{mutlak} + 1}$$

$$k = \cos\theta_{goreceli}$$

$$i = n \times x_0 + y_0 - p$$

$$f = |i|$$

$$g = j \times k$$

$$r_{goreceli} = \frac{f}{g}$$

$$r'_{goreceli} = \frac{f'g - fg'}{g^2}$$

$$f' = \mathrm{sgn}(i) \times i'$$

$$i' = m' + y'_0 - p'$$

$$m' = n'x_0 + nx_0'$$

$$n' = \frac{\cos'\theta_{mutlak}\sin\theta_{mutlak} - \cos\theta_{mutlak}\sin'\theta_{mutlak}}{\sin^2\theta_{mutlak}}$$

$$p' = \frac{r_{mutlak}'\sin\theta_{mutlak} - r_{mutlak}\sin'\theta_{mutlak}}{\sin^2\theta_{mutlak}}$$

$$g' = j' \times k + j \times k'$$

$$j' = \left(\left(\cot^2\theta_{mutlak} + 1\right)^{\frac{1}{2}}\right)' = \frac{1}{2}\left(\cot^2\theta_{mutlak} + 1\right)^{-\frac{1}{2}} \times \left(\cot^2\theta_{mutlak} + 1\right)'$$

$$j' = \frac{1}{2}\left(\cot^2\theta_{mutlak} + 1\right)^{-\frac{1}{2}} \times 2 \times \cot\theta_{mutlak} \times \cot'\theta_{mutlak} = \left(\cot^2\theta_{mutlak} + 1\right)^{-\frac{1}{2}} \times \cot\theta_{mutlak} \times \cot'\theta_{mutlak}$$

$$k' = \cos'\theta_{goreceli}$$

$$x_0 = x + a\cos\theta_{robot} + b\sin\theta_{robot}$$

$$y_0 = y + c\cos\theta_{robot} + d\sin\theta_{robot}$$

Burada x ve y robot konumu parametreleridir. a, b, c ve d parametreleri ise algılayıcının robot üzerindeki konumu ile ilgili sabit mesafe bilgileridir.

$$x_0' = x' + a\cos'\theta_{robot} + b\sin'\theta_{robot}$$

$$y_0' = y' + c\cos'\theta_{robot} + d\sin'\theta_{robot}$$

Bu noktada eşitliklerdeki tüm değişkenler temel bileşenlerine ayrılmış durumda. Bu bileşenlerin x, y, θ_{robot}, r_{mutlak} ve θ_{mutlak} parametrelerine göre kısmi türevleri aşağıda sıralanmıştır. Bu değerlerin yerlerine konulmasıyla h_t matrisi elde edilmektedir.

$$\frac{\partial\cos\theta_{goreceli}}{\partial x} = 0 \qquad \frac{\partial\cot\theta_{mutlak}}{\partial x} = 0 \qquad \frac{\partial\cos\theta_{robot}}{\partial x} = 0 \qquad \frac{\partial\sin\theta_{robot}}{\partial x} = 0 \qquad \frac{\partial\cos\theta_{mutlak}}{\partial x} = 0$$

$$\frac{\partial \sin\theta_{mutlak}}{\partial x}=0 \qquad \frac{\partial r_{mutlak}}{\partial x}=0 \qquad \frac{\partial x}{\partial x}=1 \qquad \frac{\partial y}{\partial x}=0$$

$$\frac{\partial \cos\theta_{goreceli}}{\partial y}=0 \qquad \frac{\partial \cot\theta_{mutlak}}{\partial y}=0 \qquad \frac{\partial \cos\theta_{robot}}{\partial y}=0 \qquad \frac{\partial \sin\theta_{robot}}{\partial y}=0 \qquad \frac{\partial \cos\theta_{mutlak}}{\partial y}=0$$

$$\frac{\partial \sin\theta_{mutlak}}{\partial y}=0 \qquad \frac{\partial r_{mutlak}}{\partial y}=0 \qquad \frac{\partial x}{\partial y}=0 \qquad \frac{\partial y}{\partial y}=1$$

Duvar robotun solunda görüldüyse $\frac{\partial \cos\theta_{goreceli}}{\partial \theta_{robot}}=-\sin\theta_{goreceli}$

Duvar robotun önünde görüldüyse $\frac{\partial \cos\theta_{goreceli}}{\partial \theta_{robot}}=-\sin\theta_{goreceli}\times \mathrm{sgn}\left(\theta_{robot}-\theta_{mutlak}\right)$

Duvar robotun sağında görüldüyse $\frac{\partial \cos\theta_{goreceli}}{\partial \theta_{robot}}=\sin\theta_{goreceli}$

$$\frac{\partial \cot\theta_{mutlak}}{\partial \theta_{robot}}=0 \qquad \frac{\partial \cos\theta_{robot}}{\partial \theta_{robot}}=-\sin\theta_{robot} \qquad \frac{\partial \sin\theta_{robot}}{\partial \theta_{robot}}=\cos\theta_{robot}$$

$$\frac{\partial \cos\theta_{mutlak}}{\partial \theta_{robot}}=0 \qquad \frac{\partial \sin\theta_{mutlak}}{\partial \theta_{robot}}=0 \qquad \frac{\partial r_{mutlak}}{\partial \theta_{robot}}=0 \qquad \frac{\partial x}{\partial \theta_{robot}}=0 \qquad \frac{\partial y}{\partial \theta_{robot}}=0$$

$$\frac{\partial \cos\theta_{goreceli}}{\partial r_{mutlak}}=0 \qquad \frac{\partial \cot\theta_{mutlak}}{\partial r_{mutlak}}=0 \qquad \frac{\partial \cos\theta_{robot}}{\partial r_{mutlak}}=0 \qquad \frac{\partial \sin\theta_{robot}}{\partial r_{mutlak}}=0 \qquad \frac{\partial \cos\theta_{mutlak}}{\partial r_{mutlak}}=0$$

$$\frac{\partial \sin\theta_{mutlak}}{\partial r_{mutlak}}=0 \qquad \frac{\partial r_{mutlak}}{\partial r_{mutlak}}=1 \qquad \frac{\partial x}{\partial r_{mutlak}}=0 \qquad \frac{\partial y}{\partial r_{mutlak}}=0$$

Duvar robotun solunda görüldüyse $\frac{\partial \cos\theta_{goreceli}}{\partial \theta_{mutlak}}=\sin\theta_{goreceli}$

Duvar robotun önünde görüldüyse $\frac{\partial \cos\theta_{goreceli}}{\partial \theta_{mutlak}} = \sin\theta_{goreceli} \times \mathrm{sgn}(\theta_{robot} - \theta_{mutlak})$

Duvar robotun sağında görüldüyse $\frac{\partial \cos\theta_{goreceli}}{\partial \theta_{mutlak}} = -\sin\theta_{goreceli}$

$$\frac{\partial \cot\theta_{mutlak}}{\partial \theta_{mutlak}} = -\left(\frac{1}{\sin^2\theta_{mutlak}}\right) \qquad \frac{\partial \cos\theta_{robot}}{\partial \theta_{mutlak}} = 0 \qquad \frac{\partial \sin\theta_{robot}}{\partial \theta_{mutlak}} = 0 \qquad \frac{\partial \cos\theta_{mutlak}}{\partial \theta_{mutlak}} = -\sin\theta_{mutlak}$$

$$\frac{\partial \sin\theta_{mutlak}}{\partial \theta_{mutlak}} = \cos\theta_{mutlak} \qquad \frac{\partial r_{mutlak}}{\partial \theta_{mutlak}} = 0 \qquad \frac{\partial x}{\partial \theta_{mutlak}} = 0 \qquad \frac{\partial y}{\partial \theta_{mutlak}} = 0$$

$$\frac{\partial r_{goreceli}}{\partial x} = \frac{\mathrm{sgn}(i) \times \frac{\cos\theta_{mutlak}}{\sin\theta_{mutlak}}}{\sqrt{\cot^2\theta_{mutlak} + 1} \times \cos\theta_{goreceli}}$$

$$\frac{\partial r_{goreceli}}{\partial y} = \frac{\mathrm{sgn}(i)}{\cos\theta_{goreceli}\sqrt{\cot^2\theta_{mutlak} + 1}}$$

$$\frac{\partial r_{goreceli}}{\partial \theta_{robot}} = \frac{(\mathrm{sgn}(i) \times ((nb+d)\cos\theta_{robot} - (na+c)\sin\theta_{robot}))g - \left|nx_0 + y_0 - \frac{r_{mutlak}}{\sin\theta_{mutlak}}\right|(j \times \cos'\theta_{goreceli})}{g^2}$$

$$\frac{\partial r_{goreceli}}{\partial r_{mutlak}} = \frac{\left(\frac{\mathrm{sgn}(i)}{\sin\theta_{mutlak}}\right)}{\cos\theta_{goreceli}\sqrt{\cot^2\theta_{mutlak} + 1}}$$

$$\frac{\partial r_{goreceli}}{\partial \theta_{mutlak}} = \frac{\mathrm{sgn}(i) \times \left(-x_0 - \frac{n(\cos\theta_{mutlak} x_0 - r_{mutlak})}{\sin\theta_{mutlak}}\right) \times g - f\left(\frac{-\cot\theta_{mutlak} \times \cos\theta_{goreceli}}{\sin^2\theta_{mutlak} \times j} + j\cos'\theta_{goreceli}\right)}{g^2}$$

$\theta_{goreceli}$, θ_{mutlak} ile algılayıcının duruş açısının farkı alınarak bulunur. Algılayıcının duruş açısı θ_{robot} kullanılarak hesaplanır. $\theta_{goreceli}$ hesabı ile ilgili kısmi türevler aşağıda verilmiştir.

$$\frac{\partial \theta_{goreceli}}{\partial x} = 0 \qquad \frac{\partial \theta_{goreceli}}{\partial y} = 0 \qquad \frac{\partial \theta_{goreceli}}{\partial \theta_{robot}} = -1 \qquad \frac{\partial \theta_{goreceli}}{\partial r_{mutlak}} = 0 \qquad \frac{\partial \theta_{goreceli}}{\partial \theta_{mutlak}} = 1$$

$$h_t = \begin{pmatrix} \frac{\partial r_{goreceli}}{\partial x} & \frac{\partial r_{goreceli}}{\partial y} & \frac{\partial r_{goreceli}}{\partial \theta_{robot}} & \frac{\partial r_{goreceli}}{\partial r_{mutlak}} & \frac{\partial r_{goreceli}}{\partial \theta_{mutlak}} \\ \frac{\partial \theta_{goreceli}}{\partial x} & \frac{\partial \theta_{goreceli}}{\partial y} & \frac{\partial \theta_{goreceli}}{\partial \theta_{robot}} & \frac{\partial \theta_{goreceli}}{\partial r_{mutlak}} & \frac{\partial \theta_{goreceli}}{\partial \theta_{mutlak}} \end{pmatrix}$$

h_t matrisi hesabı ile ilgili son olarak şu belirtilmelidir, θ_{mutlak} sıfır derece olduğunda

$$r_{goreceli} = \frac{\left| x_0 - \frac{r_{mutlak}}{\cos \theta_{mutlak}} \right|}{\cos \theta_{goreceli}}$$

haline gelmektedir, dolayısı ile h_t matrisi de değişmektedir. Burada bu eşitliğin türevine yer verilmemiştir.

3.2 Gözlem – Yer İşaretçisi Karşılaştırması

Bu uygulamada gözlemlerle yer işaretçileri karşılaştırılırken öncelikle görüldükleri konumlara bakılmaktadır, gözlenen doğru parçasının uç noktaları yer işaretçisinin uç noktalarından uzaktaysa bu direkt yeni bir yer işaretçi kabul edilmektedir.

Karşılaştırmada kullanılan bir yer işaretçisinin robota göre açısı olan $\theta_{goreceli}$ hesaplandığında 90 dereceden büyük çıkarsa robotun bu yer işaretçisini o konumda görmesi zaten imkansız olduğundan doğrudan gözlemin bu yer işaretçisi olmadığı kabul edilmektedir.

Bu bahsedilen durumlar ortaya çıkmadığında gözlem ile yer işaretçisi arasındaki Mahalanobis mesafesine bakılmaktadır.

Ek 3 Kitapta Söz Edilen Terimlerin Türkçe ve İngilizce Adları

Ardışıl Kenar Takibi: *Successive Edge Following*

Ayırma-ve-Birleştirme: *Split-and-Merge*

Bulanık Ayırma-ve-Birleştirme: *Split-and-Merge Fuzzy*

Ayrık: *Discrete*

Basit Doğrusal Bağlanım: *Simple Linear Regression*

Beklenti Enbüyültme: *Expectation Maximization*

Birbiçimli: *Uniform*

Dinamik Programlama: *Dynamic Programming*

Doğru İzleme: *Line Tracking*

Doğru Parçası: *Line Segment*

Doğru Bağlanımı: *Line Regression*

Dolaşım: *Navigation*

Dönüşüm: *Transform, Transformasyon*

Bağlı Uyum Dal ve Sınır Algoritması: *Joint Compatibility Branch and Bound Algorithm*

En Küçük Kareler Doğru Uydurma Metodu: *Least Square Line Fitting*

En Küçük Kareler Kestirimcisi: *Least Squares Estimator*

En Küçük Ortalama Kare: *Least Mean Square*

Eş Zamanlı Konum Belirleme ve Haritalama: *Simultaneous Localization and Mapping*

Genişletilmiş Kalman Filtresi: *Extended Kalman Filter*

Hough dönüşümü: *Hough Transform*

İteratif En Yakın Nokta: *Iterative Closest Point*

İteratif İkili Uyuşum: *Iterative Dual Correspondance*

İteratif Uç Nokta Uydurma: *Iterative End Point Fit*

Kalman Filtresi Tabanlı Kırılma Noktası Detektörü: *Kalman Filter Based Breakpoint Detector*

Kırılma Noktası: *Breakpoint*

Kısıtlı Hough Dönüşümü: *Constrained Hough Transform*

Ki-Kare: *Chi-Square*

Konum Tabanlı Harita: *Location-based Map*

Maksimum Olabilirlik: *Maximum Likelihood*

Merkez Doğrusu: *Centerline*

Modele Uygun Nokta: *Inlier*

Özellik Tabanlı Harita: *Feature-based Map*

Prototip Tabanlı Bulanık Kümeleme Algoritması: Prototype-based fuzzy clustering algorithm

Rastgele Örnek Uzlaşımı: *Random Sample Consensus – RANSAC*

Rastlantısal Hough Dönüşümü: *Randomized Hough Transform*

Sahte Yoğun Tarama: *Pseudo Dense Scan*

Sürekli: *Continuous*

Tarama Eşleştirme: *Scan Matching*

Toplam-En-Küçük-Kareler: *Total-Least-Squares*

Uyumcul: *Adaptive*

Uyumcul Kırılma Noktası Detektörü: *Adaptive Breakpoint Detector*

Üçgenleme Tabanlı Füzyon: *Triangulation-Based Fusion*

Yeniden Örnekleme: *Resampling*

Yer İşaretçisi: *Landmark*

Yığmalı Hiyerarşik Kümeleme: *Agglomerative Hierarchical Clustering*

ÖZGEÇMİŞ

Doğum tarihi	17.08.1985	
Doğum yeri	İstanbul	
Lise	2000-2003	Vatan Anadolu Lisesi
Lisans	2003-2007	Yıldız Teknik Üniversitesi Elektrik-Elektronik Fakültesi Bilgisayar Mühendisliği Bölümü

Çalıştığı kurum

2007-Devam ediyor YTÜ Bilgisayar Mühendisliği Araştırma Görevlisi