

GEZGİN ROBOT SİSTEMLERİNDE KONUMLANDIRMA VE HARİTALAMA

YUNUS ATAŞ

YÜKSEK LİSANS TEZİ  
ELEKTRİK VE ELEKTRONİK MÜHENDİSLİĞİ

TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

MART 2010

ANKARA

Fen Bilimleri Enstitü onayı

---

Prof. Dr. Ünver KAYNAK

Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylarım.

---

Prof. Dr. M. Önder EFE

Anabilim Dalı Başkanı

Yunus ATAŞ tarafından hazırlanan GEZGİN ROBOT SİSTEMLERİNDE KONUMLANDIRMA VE HARİTALAMA adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

---

Doç. Dr. Veysel GAZİ

Tez Danışmanı

Tez Jüri Üyeleri

Başkan : Yrd. Doç. Dr. A. Murat Özbayoğlu \_\_\_\_\_

Üye : Doç. Dr. Veysel Gazi \_\_\_\_\_

Üye : Yrd. Doç. Dr. Melda Yüksel \_\_\_\_\_

## **TEZ BİLDİRİMİ**

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Yunus ATAŞ

**Üniversitesi** : TOBB Ekonomi ve Teknoloji Üniversitesi  
**Enstitüsü** : Fen Bilimleri  
**Anabilim Dalı** : Elektrik ve Elektronik Mühendisliği  
**Tez Danışmanı** : Doç. Dr. Veysel GAZİ  
**Tez Türü ve Tarihi** : Yüksek Lisans – Mart 2010

**Yunus ATAŞ**

## **GEZGİN ROBOT SİSTEMLERİNDE KONUMLANDIRMA VE HARİTALAMA**

### **ÖZET**

Bu tez çalışmasında gezgin robot sistemleri için konumlandırma iyileştirmesi ve haritalama uygulamaları gerçekleştirilmiştir. Yapılan ilk çalışmada IMU (ataletsel ölçüm birimi) özellikleri incelenerek kör konumlandırma sistemlerine sahip gezgin yer robotları üzerindeki etkileri tespit edilmiş ve başarılı bir şekilde kör konumlandırma sistemlerinde iyileştirici etken olarak uygulanmıştır. Daha sonra bu sistem üzerinde parçacık filtresi uygulanarak sistem üzerindeki etkileri incelenmiş fakat parçacık filtresinin çalışılan sistem düzeneği için uygun olmadığı çeşitli uygulamalar ile gözlemlenmiştir. Kör konumlandırma sistemi üzerinde yapılan iyileştirme çalışmalarından sonra bu iyileşmenin tek robotlu haritalama sistemlerine olan etkisi incelenmiş ve konumlandırma iyileştiğinden dolayı oluşturulan haritanın gerçek alan ile daha tutarlı hale geldiği çeşitli karşılaştırmalar ile ortaya konulmuştur. Ayrıca lazer algılayıcılarının ve kör konumlandırma sistemlerinin kullanıldığı haritalamada kör konumlandırma sistemlerinde zamanla artan bir hızla büyüyen hataların, oluşan harita üzerindeki etkileri uygulamalı olarak ortaya konulmuştur ve gerçek alan ile oluşturulan haritanın daha tutarlı bir hale getirilmesi için nasıl bir yöntem izlenmesi gerektiği konusunda önerilerde bulunulmuştur. Devam eden çalışmalarda ise kör konumlandırma sistemleri kullanılarak çeşitli yaklaşımlar ile yardımlaşmalı çok robotlu haritalama sistemleri üzerinde çalışmalar yapılmış ve çok robotlu haritalama sistemlerinin daha hızlı ve sistemde oluşan bozucu etkenlere karşı daha gürbüz olduğu uygulamalar ile ortaya konulmuştur.

**Anahtar Kelimeler:** Kör Konumlandırma Sistemleri, IMU, Lazer Algılayıcı, Sürü Robotlar, Tek Robotlu Haritalama Sistemleri, Yardımlaşmalı Çok Robotlu, Haritalama Sistemleri, Yapay Potansiyel Fonksiyonlar, Parçacık Süzgeçleri

**University** : TOBB University of Economics and Technology  
**Institute** : Institute of Natural and Applied Sciences  
**Science Programme** : Electrical and Electronics Engineering  
**Supervisor** : Associate Professor Dr. Veysel GAZI  
**Degree Awarded and Date** : M.Sc. – March 2010

**Yunus ATAŞ**

## **LOCALIZATION AND MAPPING IN MOBILE ROBOT SYSTEMS**

### **ABSTRACT**

In this thesis improvement in localization and mapping in mobile robot systems have been performed. In the first study, the characteristics of IMU (Inertial Measurement Unit) sensors have been analyzed and their effect on mobile robot systems that use odometry for localization has been investigated. Moreover, they have been successfully applied to the odometric localization systems. As a second study, particle filter is applied to the odometry localization system. However, the experiments show that for the experimental set-up available in the laboratory, in which the experiments were performed, the particle filter is not an applicable method for the subjected standalone odometry based localization. After the improvement in odometry based localization in mobile robot system with IMU sensor, improved odometry based localization is applied to the single robot mapping. It is shown that the map also becomes more consistent and more accurate because of the improvement in localization. Furthermore, the effect of cumulative error in odometry based localization is studied in laser sensor based mapping systems to investigate different mapping methods for improving the consistency of the maps. Following that odometry based cooperative multi-robot map building is performed with different algorithms. Experiments show that cooperative multi-robot map building needs less time to construct the consistent map and is more robust than single-robot map building.

**Keywords:** Odometry Based Localization, IMU, Laser Sensor, Swarm Systems, Single - Robot Map Building, Cooperative Multi-robot Map Building, Artificial Potential Functions, Particle Filters

## TEŐEKKÜR

Çalıőmalarım boyunca deęerli yardım ve katkılarıyla beni yönlendiren deęerli hocam Doç. Dr. Veysel GAZI'ye ve yine kıymetli tecrübelerinden faydalandığım TOBB Ekonomi ve Teknoloji Üniversitesi Elektrik ve Elektronik Mühendislięi Bölümü öğretim üyelerine,

Her türlü zorluęa birlikte göęüs gerdiğim Sürü Sistemler Araőtırma Laboratuvarındaki çalıőma arkadaşlarım olan Murat İlter Köksal, Salih Burak Akat, Ömer Çayırpunar, Mirbek Turduev, Engin Karataő, Abdel-Razzak Merheb, Andaç Töre Őamiloęlu, Esmā Gul ve Sabahat Duran'a,

Ve her zaman beni destekleyen ve bugünlere getiren aileme teőekkürlerimi sunarım.

Bu çalıőma Avrupa Komisyonu tarafından 045269 sözleşme numaralı 6. Çerçeve Programı özel amaçlı araőtırma projesi kapsamında ve TÜBİTAK (Türkiye Bilimsel ve Teknolojik Araőtırma Kurumu) tarafından 104E170 ve 106E122 sayılı projeler kapsamında desteklenmiőtir.

## İÇİNDEKİLER

	<b>Sayfa</b>
ÖZET	iv
ABSTRACT	v
TEŞEKKÜR	vi
İÇİNDEKİLER	vii
1. GİRİŞ	1
2. GEZGİN ROBOT SİSTEMLERİNDE KÖR KONUMLANDIRMANIN İYİLEŞTİRİLMESİ	5
2.1. Kör Konumlandırma İle Güzergâh Takibi	5
2.1.1. Kör Konumlandırma Hakkında Genel Bilgi	5
2.1.2. Kör Konumlandırma İle Güzergâh Takibi Deney Sonuçları	8
2.2. IMU Destekli Kör Konumlandırma İle Güzergâh Takibi	13
2.2.1. IMU Destekli Kör Konumlandırma Hakkında Genel Bilgi	13
2.2.2. IMU Destekli Kör Konumlandırma İle Güzergâh Takibi Deney Sonuçları	19
2.3. Parçacık Süzgeci Uygulanarak Güzergâh Takibi	22
2.3.1. Parçacık Süzgeci Hakkında Genel Bilgi	22
2.3.2. Parçacık Süzgeci Parametrelerinin Hesaplanması	26
2.3.3. Parçacık Süzgeci Uygulanarak Güzergâh Takibi Deney Sonuçları	27
2.4. Güzergâh Takip Sonuçlarının Karşılaştırılması	30
3. TEK-ROBOTLU HARİTALAMA SİSTEMLERİNDE KONUM TUTARLILIĞININ HARİTA ÜZERİNE ETKİLERİ	37
3.1. Kör Konumlandırma Sistemi İle Tek-Robotlu Haritalama	37
3.2. IMU Destekli Kör Konumlandırma İle Tek-Robotlu Haritalama	47
4. YARDIMLAŞMALI ÇOK-ROBOTLU HARİTALAMA	52
4.1. Robot Modeli ve Denetimi	53
4.2. Boşluklara Yönelerek Yardımlaşmalı Haritalama	58
4.3. Yapay Potansiyel Fonksiyonlar Kullanarak Yardımlaşmalı Haritalama	63
5. SONUÇ	68
5.1. Yorumlar	68

5.2. Gelecek Çalışmalar	70
EKLER	72
A. KHEPERAIII ROBOTU	72
B. TEPE KAMERALI DENEY DÜZENEĞİ	73
B.1. Deney alanı	73
B.2. Kullanılan Görüntü İşleme Yöntemleri	74
B.2.1. Robot Konumlarının Bulunması	75
B.2.2. Robotların Yönelimlerinin Bulunması	76
C. IMU ALGILAYICI	76
C.1.1. Ara Haberleşme Kartı	77
C.1.2. IMU Algılayıcısı Linux Sürücüsü	78
D. LAZER MESAFE ALGIYACI	86
D.1. Lazer Mesafe Algılayıcısı USB Bağlantısı	86
D.2. Lazer Mesafe Algılayıcısı Linux Sürücüsü	87
E. UDP HABERLEŞME PROTOKOLÜ	88
KAYNAKLAR	91
ÖZGEÇMİŞ	96



## TABLULARIN LİSTESİ

<b>Tablo</b>	<b>Sayfa</b>
Tablo 2.1. Parçacık süzgecinde kullanılan standart sapma ve ortalama değerler.	27
Tablo 2.2. Güzergâh takibi ortalama hataları ve yüzdellik iyileşmeler tablosu.	34

## ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. KheperaIII mini gezgin robotu.	8
Şekil 2.2. KheperaIII destek tekerleği.	9
Şekil 2.3. Güzergâh takibi için tepe kamerası bulunan deney düzeneği.	10
Şekil 2.4. Tepe kamerasından alınan anlık bir görüntü.	10
Şekil 2.5. Önceden belirlenmiş güzergâh.	11
Şekil 2.6. Kör konumlandırma ile güzergâh takibi örnekleri.	12
Şekil 2.7. Kör konumlandırma ile robotun son konum ve yönelim açıları.	13
Şekil 2.8. 3DM-GX2 IMU algılayıcısı.	15
Şekil 2.9. Robot hareketsizken kaydedilen IMU fark açıları grafiği.	16
Şekil 2.10. Robot hareket halinde iken kaydedilen IMU fark açıları grafiği.	17
Şekil 2.11. IMU bağlı KheperaIII robotu.	19
Şekil 2.12. IMU destekli kör konumlandırma ile güzergâh takibi örnekleri.	20
Şekil 2.13. IMU destekli kör konumlandırma ile robotun bulunduğu son konum ve yönelim açıları.	21
Şekil 2.14. Parçacıkların köşe noktalarındaki konum ve yönelim tahminleri.	28
Şekil 2.15. Parçacık süzgeci uygulanarak IMU destekli kör konumlandırma ile güzergâh takibi örnekleri.	29
Şekil 2.16. Parçacık süzgeci uygulanarak IMU destekli kör konumlandırma ile robotun bulunduğu son konum ve yönelim açıları.	30
Şekil 2.17. Kör konumlandırma ile güzergâh takibinde son konum ve yönelim hataları grafiği.	32
Şekil 2.18. IMU destekli kör konumlandırma ile güzergâh takibinde son konum ve yönelim hataları grafiği.	33
Şekil 2.19. Parçacık süzgeci uygulanarak IMU destekli kör konumlandırma ile güzergâh takibi son konum ve yönelim hataları grafiği.	34
Şekil 2.20. IMU destekli kör konumlandırma ile güzergâh takibinin iyileştirilmesi grafiği.	36
Şekil 3.1. Hokuyo URG-04LX lazer mesafe ölçer.	38
Şekil 3.2. Lazer algılayıcı bağlı KheperaIII robotu.	39
Şekil 3.3. Robot deney alanı.	40
Şekil 3.4. Kör konumlandırma sistemi ile tek robotlu haritalamada tam görüş mesafesinde zamanla büyüyen haritalar.	42
Şekil 3.5. Kör konumlandırma sistemi ile tek robotlu haritalamada tam görüş mesafesinde oluşturulan son harita.	43
Şekil 3.6. Robot yönelim açısındaki hataların harita üzerine etkisi.	44
Şekil 3.7. Kör konumlandırma sistemi ile tek robotlu haritalamada kısa görüş mesafesinde zamanla büyüyen haritalar.	45

Şekil 3.8. Kör konumlandırma sistemi ile tek robotlu haritalamada kısa görüş mesafesinde oluşturulan son harita.	46
Şekil 3.9. IMU destekli kör konumlandırma sistemi ile tek robotlu haritalamada tam görüş mesafesinde zamanla büyüyen haritalar.	48
Şekil 3.10. IMU destekli kör konumlandırma sistemi ile tek robotlu haritalamada tam görüş mesafesinde oluşturulan son harita.	49
Şekil 3.11. IMU destekli kör konumlandırma sistemi ile tek robotlu haritalamada kısa görüş mesafesinde zamanla büyüyen haritalar.	50
Şekil 3.12. IMU destekli kör konumlandırma sistemi ile tek robotlu haritalamada kısa görüş mesafesinde oluşturulan son harita.	51
Şekil 4.1. Lazer verileri küme şeması.	56
Şekil 4.2. Potansiyel kuvvet grafiği.	56
Şekil 4.3. Deney uygulama alanı.	57
Şekil 4.4. Boşluklara yönelerek yardımlaşmalı zamanla büyüyen haritalar.	60
Şekil 4.5. Boşluklara yönelerek yardımlaşma yöntemiyle olarak oluşturulan bireysel haritalar ve birleştirilen son harita.	62
Şekil 4.6. Potansiyel fonksiyonlar kullanarak yardımlaşma yöntemiyle zamanla büyüyen haritalar.	65
Şekil 4.7. Potansiyel fonksiyonlar kullanarak yardımlaşma yöntemiyle oluşturulan bireysel haritalar ve birleştirilen son harita.	66
Şekil A.1. KheperaIII robotu kızıl ötesi algılayıcıları dizilimi.	73
Şekil A.2. KheperaIII robotu ses ötesi algılayıcıları dizilimi.	73
Şekil B.1. Robot alanı, tepe kamerası ve masaüstü bilgisayardan oluşan deney düzeneği.	74
Şekil B.2. Robotların konum ve yönelimlerinin bulunmasını sağlayan kodlama sisteminin boyutları mm cinsinden olan örnek görüntüsü.	75
Şekil B.3. Robotların konum ve yönelimlerinin bulunmasını için kullanılan renkli noktalar.	76
Şekil C.1. KheperaIII haberleşme kartı.	77
Şekil C.2. KheperaIII haberleşme kartı şematik gösterimi.	78
Şekil D.1. USB Hub bağlı KheperaIII robotu.	86

## BÖLÜM 1

### 1. GİRİŞ

Birçok gezgin robot sistemleri iki basit konum tahmin yönetimi bir arada kullanılmaktadır: mutlak ve bağıl konumlandırma [1, 2, 3]. Mutlak konumlandırma genellikle gezinme işaretlerine, aktif veya pasif yer işaretlerine, harita eşleştirme veya uydu tabanlı gezinme işaretlerine bağlıdır. Bu mutlak konumlandırma yaklaşımlarından her biri çeşitli yöntem ve algılayıcılarla uygulanmaktadır. Fakat mevcut sistemlerin hiçbirisi mükemmel değildir ve genellikle bir bakıma pahalıdır. Gezgin robot sistemlerinde konumlandırma üzerine kapsamlı birer çalışma olarak [4, 5] gösterilebilir.

Bağıl konumlandırma genellikle kör konumlandırma tabanlıdır. Kör konumlandırma basit, pahalı olmayan ve gerçek zamanlı uygulanabilen bir konumlandırma yöntemidir. Kör konumlandırmanın dezavantajı ise sistemde oluşan hataların toplanarak sınırsız bir şekilde artmasıdır.

Fiber optik jiroskopların ortaya çıkmasıyla birlikte, jiroskoplar gezgin robot sistemlerinde de kullanılmaya başlanmıştır. Fakat bunun yanında jiroskoplardaki göreceli büyük sapma oranları yönelimde sınırı olmayan hata büyümelerine sebep olmaktadır. Bu sebepten dolayı jiroskopların kullanılabilmesi için kalman süzgeçleri gibi çeşitli süzgeçlere ihtiyaç duyulmaktadır. Bu süzgeçler sayesinde jiroskopların yönelim ölçme hata oranları kör konumlandırma sistemlerine göre büyük oranda azaltılabilmektedir. Bu süzgeçler günümüzde donanımsal olarak IMU (Ataletsel Ölçüm Birimi) algılayıcıları içinde uygulanabilmektedir.

Kör konumlandırma sistemlerinin ve jiroskopların sınırı olmayan hata büyüme oranlarına sahip olmalarından dolayı, bu konumlandırma sistemleri ancak dönemsel mutlak konumlandırma sistemleri ile beraber kullanılabilirler. Bununla birlikte kör konumlandırma ve jiroskoplardaki hata oranlarının azaltılması, iki mutlak konum güncellemesi arasındaki gezinme mesafesini artırmaktadır ve böylelikle mutlak konum güncellemesi için kullanılan yer işaretleri gibi donanımların

alan içerisindeki gereksinimleri azaldığından sistemin çalışma maliyeti düşürülebilmektedir.

Parçacık süzgeçleri kör konumlandırma sistemlerinde harita eşleştirme veya yer işaretleri ile kullanılan diğer konum tahmin yöntemlerinden birisidir. Parçacık süzgeçlerindeki ana fikir, olasılık dahilinde olan her bir konum tahmini için ayrı bir parçacık atanarak konum takibinin yapılmasıdır. Mutlak konum güncellemesi ulaşılabilir olduğunda ise her parçacığın takip ettiği konumun gerçek konuma göre hata payı hesaplanmaktadır. Böylelikle konum iyileştirmesi için hangi parçacığın daha az hata yaptığına bakılarak ve bir sonraki adımda hangi parçacıkların daha çok kullanılacağına karar verilerek kör konumlandırma sistemlerinde parçacık süzgeçleri ile iyileştirme yapılabilmektedir. Bu tez çalışmasında ise mutlak konumlandırma sistemleri olmadan sistemde kullanılan kör konumlandırmanın mutlak konumlandırma olduğu varsayımından yola çıkılarak parçacık süzgecinin konumlandırma sistemindeki etkileri üzerine bir inceleme yapılmıştır. Fakat deney sonuçlarında parçacık süzgecinin bizim uygulamamızdaki IMU destekli kör konumlandırma sistemine bir iyileştirme getirmediği anlaşılmıştır.

Robotların alan içinde gezinme yöntemleri hakkında geriye kalan iki soru: “Ben neredeyim?” ve “Nerelerden Geçtim?” sorularıdır. Bu iki soruya cevap genellikle konumlandırma ve haritalamadır. Konumlandırma ve harita çıkarma birbirine bağımlı işlemlerdir. Eğer bir robot alan içerisinde nerede olduğunu tam olarak bilemezse gerçek alan ile tutarlı bir harita çıkarması mümkün değildir. Bu sebepten dolayı haritalama işlemlerinde her şeyden önce konumlandırma çok önemlidir. Ancak ve ancak konumlandırmadaki hata oranı kabul edilebilir bir seviyenin altında ise haritalama işlemi gerçekleştirilebilir. Tutarlı bir haritanın çıkarılabilmesi ise gezgin robot sistemlerinde önemli bir yer teşkil etmektedir.

Harita çıkarılması işleminin önemli olmasının sebebi birçok gezgin robot sisteminin çalışabilmesi için çalıştıkları alanın haritasına ihtiyaç duymalarıdır. Örnek olarak Helpmate hastane servis robotu [6], Simmons’s Xavier Robot [7] ve çeşitli müze rehberi robotlar [8, 9, 10, 11] verilebilir.

Son yıllarda, gezgin robotlar için haritalama üzerine birçok çalışma yapılmıştır [12, 13, 14, 15]. Bazı çalışmalar robotların buldukları ortamların topolojik detaylarını

çıkarmak üzereyken diğer çalışmalar daha ayrıntılı olan metrik haritalar üzerine yoğunlaşmaktadırlar [12, 16, 17, 18]. Çoklu-robotlar eşgüdümlü olarak işbirliği içinde çalışarak bilinmeyen ortamda arama yapmakta ve herhangi bir robotun bozulması veya istenen şekilde çalışmaması durumunda diğer robotlar görevi tamamlayabilmektedirler. Bu özelliklerinden dolayı denetim ve koordinasyon yöntemlerinin uygun tasarlanmasıyla çok-robotlu haritalama sistemleri tek-robotlu haritalama sistemlerine göre artan bir şekilde daha verimli ve gürbüz hale gelmektedirler. Bu artan verim, haritalama işlemlerinde çoklu-robotların tekli-robotlara göre daha çok kullanılmalarının başlıca sebeplerinden birisidir. Ne kadar çok robot alanın haritasını çıkarmak için çalışırsa, bunun tersine bir oranla bütünsel haritanın çıkarılması için gereken zaman da robotlar arasında uygun bir eşgüdümlü çalışma ile azalabilmektedir. Ayrıca, haritalama işlemi sırasında bilinmeyen bir sebepten dolayı bir veya daha fazla robot görev dışı kalabilir. Bu durumda sistemdeki diğer robotlar haritalama işlemine devam edip bütünsel haritayı oluşturabilirler. Tek-robotlu sistemlerdeyse robotun görev dışı kalması durumunda tüm görev başarısızlık ile sonuçlanacaktır. Diğer yandan, çoklu-robotların tekli-robotlara göre bazı dezavantajları da bulunmaktadır; eşgüdümlü çalışmalarını sağlamak kolay olmayabilir, farklı robotlardan alınan verilerin tek bir haritada toplanması robotların konumlanma hatalarından dolayı ek işlemler gerektirmektedir, bu işlemler ise kolay değildir ve robotlar arasındaki sınırlı haberleşme mesafeleri de robotların eşgüdümlü çalışmalarını kısıtlayabilir.

Gerçek alan ile tutarlı bir harita oluşturulabilmesi için, farklı robotlardan alınan verilerin tek bir haritada birleştirilmesi gerekmektedir. Bunun yanı sıra, bu gibi harita birleştirme işlemlerinde haritanın tutarlılığının artırılması için işlem olabildiğince kısa sürede tamamlanmalıdır. Eğer robotların başlangıç konumları biliniyor ve robot konumlama sistemleri hatasız bir şekilde çalışıyorlarsa, harita birleştirme işlemi tek-robotlu haritalama işleminden daha fazla bir işlem yükü teşkil etmemektedir [19, 20, 21]. Bunun yanında, birçok uygulamada harita birleştirme işlemi hatalı konumlama sistemlerinden ve hatalı başlangıç konum verilerinden dolayı zor bir işlem haline gelebilmektedir.

Büyük ölçekli alanların haritalarının çıkarılması esnasında, sınırlı haberleşme mesafesine bağlı olarak kumanda merkezi ve robotlar arasındaki iletişim geçici olarak kopabilir. Bu gibi aksaklıklara karşı haritalama sistemini gürbüzleştirmek için, bütün robotlar algılayıcı ve konum verilerini daha sonra iletişim tekrar çalışır hale geldiğinde kontrol merkezine göndermek için kaydedebilmektedirler. Çoklu-robot sistemlerinde bu gibi aksaklıklarla baş edilebilmesi için, bazı robotlar görev esnasında kumanda merkezi ve robotlar arasında iletişim köprüsü olarak kullanılabilirler. Böylece bu iletişim köprüsü olarak görev yapan robotlar kumanda merkezi ile diğer robotlar arasında iletişimin sürekliliğini garantileyebilmektedirler.

Bu tez çalışmasında, IMU algılayıcıları kullanılarak robotların kör konumlandırma sistemlerinde iyileşmeler sağlanmış ve sonrasında lazer mesafe algılayıcıları kullanılarak robotlar hareket halinde iken çevrimiçi harita oluşturulması üzerine çalışmalar yapılmıştır. Çevrimiçi harita oluşturulması, haritalamanın gerekli olduğu bir sonraki adımda nereye gidileceğine karar vermesi gereken robot sistemleri ve tarama görevleri açısından oldukça önemlidir [22, 23]. Haritalamanın çevrimiçi olmasından dolayı, zaman ilerledikçe haritanın son hali daha tutarlı ve tamamlanmış hale gelmektedir.

Bu tez çalışması şu şekilde düzenlenmiştir. Bölüm 2’de, kör konumlandırma iyileştirmesi üzerine yapılan çalışmalar karşılaştırmalı olarak anlatılmıştır. Bölüm 3’te, IMU destekli kör konumlandırma sistemi kullanılarak tek robotlu haritalama yapılmış ve IMU iyileştirmesi olmadan çıkarılan harita ile farkları çeşitli lazer görüş mesafeleri için ortaya konulmuştur. Bunu takiben Bölüm 4’te, gezgin robotlarla yardımlaşmalı harita oluşturulması problemi iki farklı seyrüsefer yöntemi ile ele alınmış ve farklılıkları anlatılmıştır. Sonuç bölümünde ise, son yorumlar açıklanarak ve gelecek çalışmalar anlatılarak tez çalışması sonuçlandırılmıştır.

## BÖLÜM 2

### 2. GEZGİN ROBOT SİSTEMLERİNDE KÖR KONUMLANDIRMANIN İYİLEŞTİRİLMESİ

#### 2.1. Kör Konumlandırma İle Güzergâh Takibi

##### 2.1.1. Kör Konumlandırma Hakkında Genel Bilgi

Kör konumlandırma sistemlerinden bahsedilince ilk akla gelen diferansiyel sürürlü yani birbirinden bağımsız iki tekerleği ile hareket edebilen robotlardır. Genel itibariyle motor sürücülerinin üzerilerine tekerleklerin devirlerini sayan optik kodlayıcılar yerleştirilmektedir. Basit geometrik hesaplamalar ile bu devir sayılarından bilinen bir başlangıç noktasına göre anlık konum bilgisi doğrudan hesaplanmaktadır. Bu hesaplama kör konumlandırma adı verilmektedir.

Bu bölümdeki kör konumlandırmanın anlatımı Borenstein'in [4] çalışmasındaki anlatımı takip edilerek yapılmıştır. Bu hesaplamalar kör konumlandırma için aşağıdaki gibi yazılabilir (ayrıca bakınız, [24,25]). Örnekleme aralığının  $I$  ve kodlayıcıların bu aralıkta ölçtüğü sol ve sağ tekerlek devirlerinin ise  $N_{sl}$  ve  $N_{sg}$  olduğu varsayalım. Daha sonrasında

$$c_m = \pi \frac{D_n}{nc_e} \quad (2.1)$$

olduğunu varsayım. Burada

$c_m$  = kodlayıcı ölçümlerini tekerleklerin doğrusal yer değiştirmesine çeviren katsayı

$D_n$  = itibari<sup>1</sup> tekerlek çapı (mm)

$c_e$  = kodlayıcı çözünürlüğü (devir başına düşen ölçüm sayısı)

$n$  = kodlayıcıların bağlı olduğu motor ve tekerlek arasındaki dişli katsayısı

---

<sup>1</sup> İng: Nominal



Sol ve sağ tekerleklerin  $I$  ölçüm aralığı içinde kat ettikleri mutlak mesafeler  $\Delta U_{sl,i}$  ve  $\Delta U_{sg,i}$  aşağıdaki denklemlere göre hesaplanabilir.

$$\Delta U_{sl/sg,i} = c_m N_{sl/sg,i} \quad (2.2)$$

ve robot merkezi  $c$ 'nin doğrusal yer değiştirmesi  $\Delta U_i$  ise

$$\Delta U_i = (\Delta U_{sl} + \Delta U_{sg})/2 \quad (2.3)$$

denklemleri ile hesaplanabilir.

Daha sonra, robotun yön değişimi  $\Delta \theta_i$

$$\Delta \theta_i = (\Delta U_{sg} - \Delta U_{sl})/b \quad (2.4)$$

denklemleri kullanılarak hesaplanır. Burada  $b$  iki tekerlek arasındaki itibari mesafedir.

Robotun yeni bağlı yönelimi aşağıdaki gibi hesaplanabilir

$$\theta_i = \theta_{i-1} + \Delta \theta_i \quad (2.5)$$

ve merkez noktanın bağlı yer değiştirmesi

$$\begin{aligned} x_i &= x_{i-1} + \Delta U_i \cos \theta_i \\ y_i &= y_{i-1} + \Delta U_i \sin \theta_i \end{aligned} \quad (2.6)$$

burada  $x_i, y_i$  = robotun merkez noktası  $c$ 'nin  $i$  anındaki bağlı konumu.

Kör konumlandırma bahsederken üzerinde durulması gereken önemli özelliklerden birisi konumlandırma hatalarının genellikle yönelim açısındaki hatalardan kaynaklanmasıdır ve bunun sebebi ise yönelim açısında oluşan hataların yanal hatalara yani hareket edilen düzlem üzerinde hem  $x$  hem de  $y$  koordinatlarında hatalara sebep olarak hatanın sınırı olmadan toplanarak büyümesidir [26].

Kör konumlandırma sistemlerinde tekerlek devirlerinin hareket edilen düzleme göre doğrusal yer değiştirmeye dönüştürebileceği varsayılır. Bu varsayım sadece sınırlı şartlar altında geçerlidir. Uç bir örnek olarak tekerleklerin patinaj yapması, yağ veya benzeri bir maddenin üzerinde kayması gösterilebilir. Bu gibi durumlarda kodlayıcılar tekerleklerin doğrusal bir yer değiştirmeye ait olmayan devirlerini de ölçeceklerdir. Bu ölçümler ise konumlandırmada hatalara sebep olacaktır.

Bu gibi patinaj ve tekerlek kayma örneklerinin yanında kodlayıcı ölçümlerinin tutarsız doğrusal harekete dönüştürülmesinde birçok başka etken vardır. Bu etkenler iki gruba ayrılabilir: 1. sistematik hatalar ve 2. sistematik olmayan hatalar.

1. Sistematik hatalar

- a. Eşit olmayan tekerlek çapları
- b. Her iki tekerlek çapının ortalamasının itibari çaptan farklı olması
- c. Tekerleklerin yerinden kaymış olması
- d. İki tekerlek arasındaki mesafenin itibari mesafeden farklı olması
- e. Kısıtlı kodlayıcı çözünürlüğü
- f. Kısıtlı kodlayıcı örnekleme oranı

2. Sistematik olmayan hatalar

- a. Pürüzlü bir düzlemde hareket edilmesi
- b. Düzlem üzerinde beklenmeyen nesnelere üzerinden geçilmesi
- c. Tekerlek kayması

Sistematik ve sistematik olmayan hataların arasındaki en büyük fark kör konumlandırma sistemindeki hata oranlarının etkin bir şekilde düşürebilmesidir. Örneğin sistematik hatalar sabit bir şekilde toplanarak büyüdüğünden dolayı kör konumlandırma sistemlerinin hataları büyük oranda sistematik hatalardan oluşmaktadır. Düz zeminlerin birçoğunda sistematik hatalar sistematik olmayan hatalara göre daha fazla gözlemlenmektedir. Diğer yandan, engebeli veya pürüzlü bir zeminde ise sistematik olmayan hatalar baskın olmaktadır. Sistematik olmayan hatalardaki problem ise hatanın beklenmedik şekilde oluşabilmesidir. Örnek olarak robotun aniden bir nesneye çarpması verilebilir. Bu gibi durumlar kör konumlandırma sistemlerinde tahmin edilemeyen büyük hatalara sebep olabilmektedir.

Bu tez çalışmasında diferansiyel sürüş özelliğine sahip robotların kör konumlandırma sistemlerinde oluşan sistematik hataların IMU algılayıcısı kullanılarak azaltılması üzerine çalışmalar yapılmıştır.

### 2.1.2. K r Konumlandırma İle G zerg h Takibi Deney Sonuları

Daha  nceki b l mlerde de bahsedildiĐi  zere diferansiyel s r ş  zelliĐine sahip gezgin robotlarının k r konumlandırmalarındaki hataların temel sebebi iki tekerlek apının birbirinden farklı olması veya iki tekerlek arasındaki mesafenin itibari deĐerden farklı olmasıdır. Bu gibi hatalar robota has  zelliklerden kaynaklanmaktadır ve dolayısıyla robotların her alıřmasında yaklaşık aynı hata oranları oluřur. Bu oranlar dolayısı ile her robot iin farklı olabilmektedir.

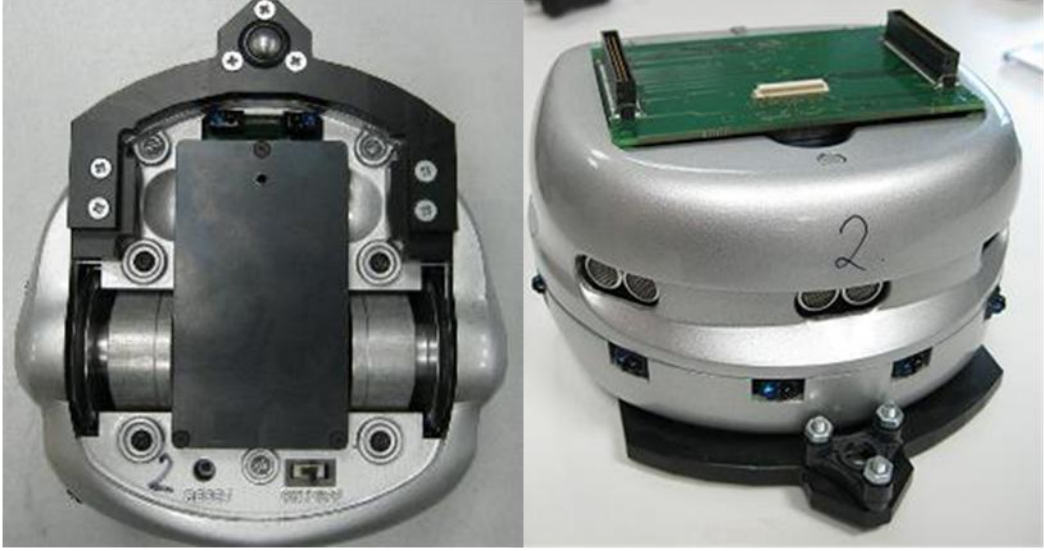
Bu deneyin amacı, diferansiyel s r ş  zelliĐine sahip bir robotun k r konumlandırma ile  nceden belirlenmiř bir g zerg hi ne derece takip edebildiĐini saptamaktır. Bunun iin deney esnasında diferansiyel s r ş  zelliĐine sahip KheperaIII robotları (řekil 2.1.) kullanılmıřtır. Dikkat edilmesi gereken unsur KheperaIII mini gezgin robotunun yerel konumlandırmasında robotun  n  x ekseni ve saĐ tarafı ise y ekseni olarak kabul edilmektedir. Bu  zellik g z  n n  alındıĐında bu tezde robot konumlarının kullanıldıĐı t m grafiklerde x ekseni deĐerleri saĐ tarafa doĐru artmakta iken y ekseni deĐerleri ařaĐı doĐru artmaktadır.



řekil 2.1. KheperaIII mini gezgin robotu.

KheperaIII robotu (ayrıntılı bilgi iin bakınız Ek 1.) iki tekerlek ve yerden destek almasını saĐlayan plastik bir para ile dengesini saĐlamaktadır. Robotun hareketi esnasında bu plastik para s rekli yer ile temas halinde olduĐundan dolayı, zamanla ařınarak k r konumlandırma sistemini olumsuz y nde etkilemektedir. Zamanla ařınan plastik paranın zeminle daha fazla s rt nmesinden dolayı zemin  zerindeki p r zlere veya ok k  k ıkıntılara takılarak konumlandırma sisteminde sistematik

olmayan hatalar oluşturmaktadır. Bu sistematik olmayan hataları en aza indirmek amacı ile KheperaIII robotlarının plastik destekleri çıkarılarak yerine metal, top şeklinde ve her yöne dönebilen birer destek yerleştirilmiştir (Şekil 2.2.). Böylelikle robotun zemin ile teması en aza indirgenmiştir.



Şekil 2.2. KheperaIII destek tekerleği.

Güzergâh takibi robot deneyleri TOBB Ekonomi ve Teknoloji Üniversitesi Sürü Sistemler Araştırma Laboratuvarı'nda bulunan, Şamiloğlu ve diğerleri tarafından [48] geliştirilen ve Şekil 2.3.'de görüntüsü verilen bir deney alanında gerçekleştirilmiştir (ayrıca bakınız Ek 2.). Deney düzeneği 120x180 cm boyutlarında sistematik olmayan hataların en aza indirilmesi için zemini pürüzsüz bir robot alanından, bu alanın tamamını görebilecek bir yüksekliğe yerleştirilmiş robotun gerçek konumunun belirlenip kaydedilebilmesi için bir USB web kamerasından, bir adet KheperaIII robotundan, görüntü işleme, konum algılaması ve kod geliştirme için Matlab yazılımının yüklü olduğu bir masaüstü bilgisayardan oluşmaktadır. Burada robotun konum ve yönelim açısı robotun üzerine yerleştirilmiş, dışsal üç turuncu renkli daire ile ve gerektiğinde robotun kimliği ise merkezdeki içsel siyah renkli dairelerden oluşan ikili bir kodlama sistemi ile belirlenebilmektedir (ayrıca bakınız Ek 2.). Tepe kamerasından alınan anlık bir görüntü örneği Şekil 2.4.'te gösterilmektedir.



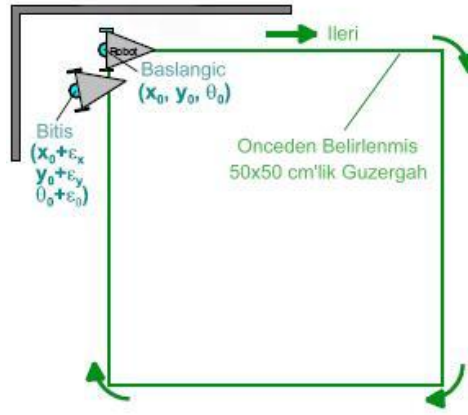
Şekil 2.3. Güzergâh takibi için tepe kamerası bulunan deney düzeneği.



Şekil 2.4. Tepe kamerasından alınan anlık bir görüntü.

Kör konumlandırma iyileştirmesi üzerine çalışma yapan Borenstein ve Feng [5]'in çalışmalarından elde edilerek değiştirilen Şekil 2.5.'te 50 x 50 cm'lik bir kare güzergâh gösterilmektedir. Robot başlangıç olarak gösterilen  $x_0$ ,  $y_0$ ,  $\theta_0$  konumundan 0.02 m/sn hızla hareketine başlamaktadır ve güzergâhın köşe noktalarına yaklaşmaya

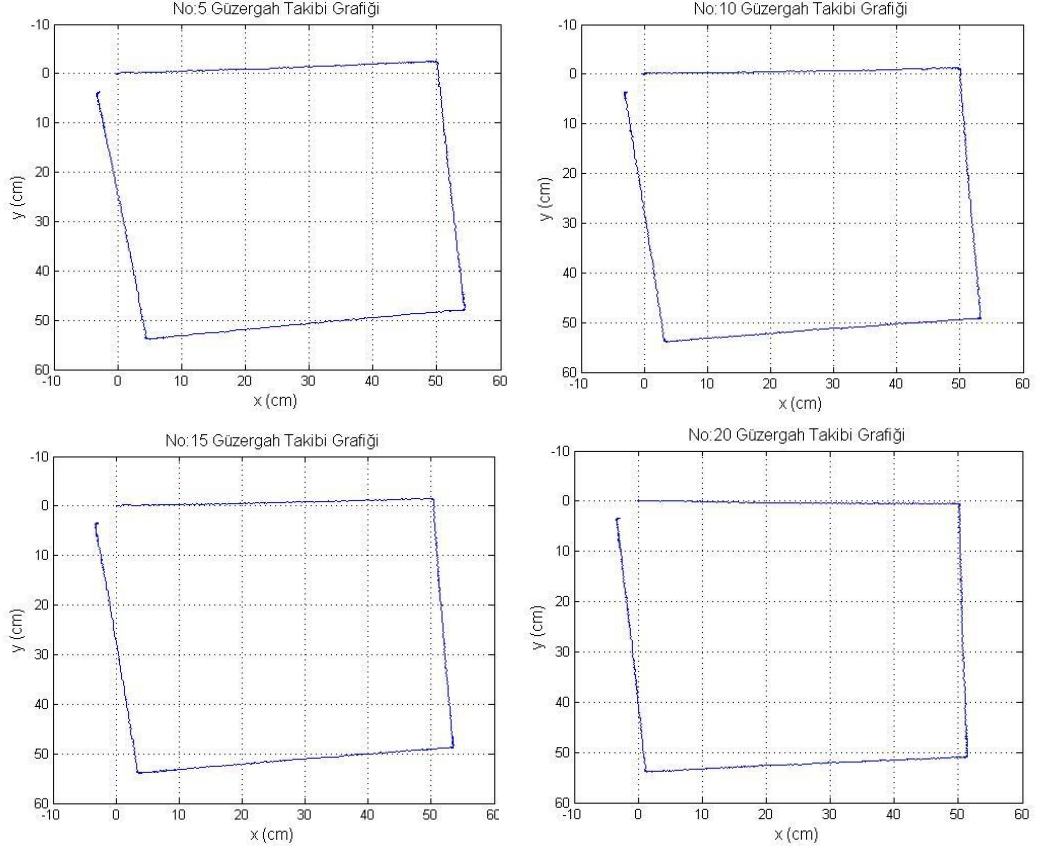
başladığında hızını yavaş yavaş azaltarak ulaştığında ise durup yönünü 25 derece/sn açısal hızla 90 derece değiştirerek dönüşünü gerçekleştirmektedir. Bu şekilde robot yaklaşık olarak 2 dakikada başladığı noktaya geri dönmektedir. Bu deney esnasında tepe kamerası robotun gerçek konumunu bulmak için kullanılmaktadır. Sonrasında kör konumlandırma sisteminde oluşan hata oranının hesaplanması için bu tepe kamerası yardımı ile bulunan konum robotun gerçek konumu olarak kabul edilmektedir.



Şekil 2.5. Önceden belirlenmiş güzergâh.

Robot kör konumlandırma sistemini kullanarak önceden belirlenen güzergâh üzerinden ayrılmadan başlangıç noktasına geri dönmesi için programlanmıştır. Robotun kör konumlandırma sisteminin hatasız çalışması durumunda robotun başlangıç konumundan başladıktan sonra güzergâhı takip ederek tekrar başlangıç konumuna dönmesi beklenmektedir. Fakat kör konumlandırmadaki sistematik hatalardan ve güzergâh takibi için kullanılan kontrol programındaki hatalardan dolayı robot başlangıç konumuna tekrar dönememektedir. Robotun hareketi esnasında tepe kamerası ile robotun gerçek konumu sürekli kaydedilmektedir. Güzergâh tamamlandıktan sonra kamera yardımı ile robotun son konum ve yönelimi hesaplanmaktadır. Bu son hesaplamalar daha sonrasında robotun kendi konumlandırma sisteminin hesapladığı son konum ve yönelim açısıyla karşılaştırılmaktadır.

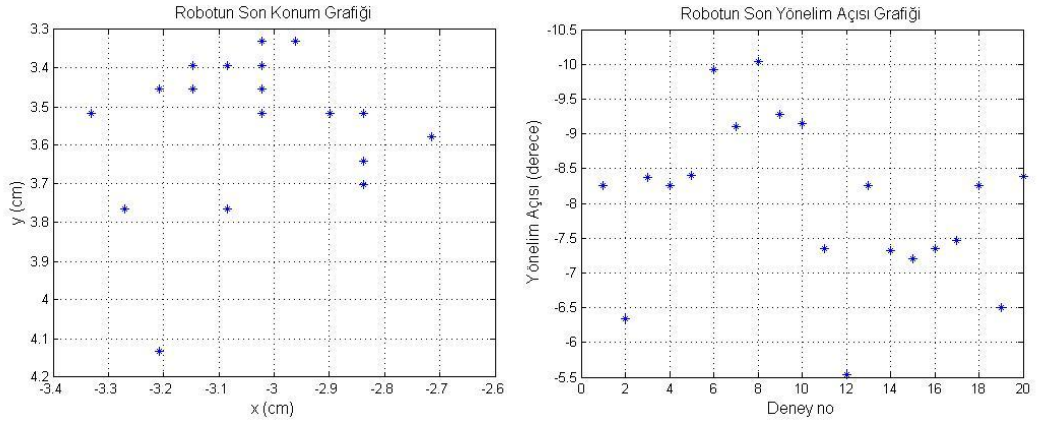
Bu deney 20 defa tekrarlanarak tüm veriler daha sonra incelenmek üzere kaydedilmiştir. Robotun kör konumlandırma sistemini kullanarak takip işlemi boyunca gerçekte izlediği yol, tepe kamerası yardımıyla hesaplanan ve bu deney esnasında robotun gerçek konum ve yönelimi olarak varsayılan çeşitli güzergâh örnekleri Şekil 2.6.'da gösterilmektedir.



Şekil 2.6. Kör konumlandırma ile güzergâh takibi örnekleri.

Şekil 2.6.'dan da anlaşılacağı gibi robot kör konumlandırma sistemini kullanarak verilen güzergâhı takip edememektedirler. Bunun en büyük sebebi daha önceden de belirtildiği gibi robotun sistematik hatalarıdır. Bir diğer sebep ise kontrol programındaki hatalardır. Şekil 2.6.'da robotun gerçekte kat ettiği yollar incelendiğinde, robotun hareketine başladıktan kısa bir süre sonra güzergâhtan ayrıldığı gözlemlenmektedir. Özellikle köşe noktalara yaklaştığında güzergâhtan oldukça uzaklaştığı rahatlıkla gözlemlenebilmektedir. Bunun başlıca sebebi ise iki tekerlek çapının birbirine eşit olmamasıdır.

Deney sonuçlarının incelenmesinde robotların sadece son konum ve yönelimleri üzerinde işlem yapılmaktadır çünkü robotun gerçek güzergâhı takip edememesinin diğer bir sebebi kontrol programındaki hataların yanı sıra robotu başlangıç konumuna istenilen açı ve konumda tam olarak yerleştirmenin her zaman mümkün olamamasıdır. İlerideki analizlerde kullanılacak olan robotların son konum ve yönelim açıları Şekil 2.7.'de grafiksel olarak gösterilmektedir.



Şekil 2.7. Kör konumlandırma ile robotun son konum ve yönelim açıları.

Şekil 2.7.'de robotun son konum grafiğinde gösterilen konumların x eksenindeki ortalaması -3.0199 cm, varyansı 0.0278, y eksenindeki ortalaması 3.55 cm, varyansı 0.0354 ve son yönelim açısı grafiğinde gösterilen açıların ortalaması ise -8.04 derece, varyansı ise 1.3695'dir. Eğer robotun kör konumlandırma sisteminde hatalar oluşmasaydı robot her zaman (0, 0) konumuna sıfır derece yönelim açısıyla dönebilecekti ve böylelikle tüm eksenlerdeki ortalama ve varyanslar sifıra eşit olacaktı.

## 2.2. IMU Destekli Kör Konumlandırma İle Güzergâh Takibi

### 2.2.1. IMU Destekli Kör Konumlandırma Hakkında Genel Bilgi

Gezgin robot sistemlerinde jiroskopların kullanımı hakkında kapsamlı bir çalışma olarak Barshan ve Durrant-Whyte'nin çalışmaları [27] gösterilebilir. Gezgin robot sistemleri üzerinde test edilen jiroskoplardan birisi MURATA [28] tarafından üretilen ENV-O5S Gyrostar bir diğeri ise GEC [29] tarafından üretilen Solid State Angular Rate Transducer (START)'dır. Barshan ve Durrant-Whyte bu iki



algılayıcının performanslarını test etmişlerdir ve dakikada 5 dereceden 15 dereceye kadar büyük sapmalar meydana geldiğini bulmuşlardır. Daha sonrasında aynı araştırmacılar jiroskoplar için Genişletilmiş Kalman Süzgeçlerinde kullanılmış olan karmaşık hata modelini geliştirdiler. 5 dakikalık bir deney sonucunda START yönelim açısında yaklaşık olarak -70.8 derecelik bir hata yaparken Gyrostar -59 derecelik bir hata yapmıştır. Genişletilmiş Kalman Süzgeçleri ile bu hatalar START için 12 derecenin altına ve Gyrostar için ise 3.8 derecenin altına düşürülmüştür. Yapılan deneyler şunu göstermiştir ki, Genişletilmiş Kalman Süzgeci'nin uygulanmasıyla 5 dakikalık bir deney sonucunda açısız yönelim hatalarının 5 – 6 kata kadar azaldığı gözlemlenmiştir. Bununla beraber Genişletilmiş Kalman Süzgeçleri'nin hala dakikada 1 ile 3 derece arasında hataları olabilmektedir.

Günümüz teknolojisi sayesinde Genişletilmiş Kalman Süzgeçleri IMU algılayıcıları içinde donanımsal olarak uygulanabilmektedir. Bu amaçla Microstrain [30] tarafından üretilen 3DM-GX2 IMU algılayıcısı (Şekil 2.8.) robot seyrüseferleri için kullanılabilir uygun bir algılayıcıdır. Bu algılayıcı minyatür MEMS algılayıcı teknolojisi kullanılan jiroskoplara sahip yüksek performanslı bir yönelim algılayıcıdır. İçinde 3 eksenli doğrusal ivmeölçer, 3 eksenli jiroskop, 3 eksenli manyetometre, sıcaklık ölçerler ve üzerinde gelişmiş bir algılayıcı okuma algoritması çalışan dahili bir işlemci bulundurmaktadır. 3DM-GX2 kalibrasyonu yapılmış ataletsel ölçümlerden (doğrusal ivme, açısız oran ve manyetik alan veya fark açı ve fark hız vektörleri) hesaplanmış yönelim tahminlerine (yalpa<sup>2</sup>, baş kaldırma<sup>3</sup> ve savrulma<sup>4</sup> matrisi) kadar birçok veri çıkışı verebilmektedir. Bütün veri çıkışları sıcaklığa dayanıklı ve algılayıcının yanlış yerleştirilmelerine karşı iyileştirilmektedir. Özellikle açısız ölçümler yer çekim ivmesi hassaslığı için ek olarak iyileştirilmektedir. 3DM-GX'nin haberleşme arabirimi farklı bir birim içinde muhafaza edilmektedir ve böylelikle kablosuz, USB 2.0, RS232 ve RS422 gibi haberleşme yöntemlerinden herhangi birisi algılayıcıya kolaylıkla eklenebilmektedir. 3DM-GX2 algılayıcısının en önemli özelliklerinden birisi veri çıkışının tek tek

---

<sup>2</sup> İng : roll

<sup>3</sup> İng : pitch

<sup>4</sup> İng : yaw

okuma veya sürekli okuma şeklinde olabilmesidir. Algılayıcı tek tek okuma ayarında kullanıcıdan hangi verinin okunacağına dair komut bekler ve komut gelmesi durumunda istenen algılayıcıdan gerekli ölçümleri yapar ve kullanıcıya sonucu iletir. Algılayıcı sürekli okuma ayarında çalıştığında ise kullanıcının hangi verinin okunacağına dair komut göndermesine gerek yoktur, kullanıcı algılayıcıyı sürekli veri gönderme ayarına geçirirken ölçülecek olan algılayıcı çıktılarını IMU algılayıcısına bildirir ve algılayıcı sürekli olarak bu özelliğe ait ölçümleri hesaplar ve kullanıcıya gönderir. Bu algılayıcının diğer bir önemli özelliği ise 1 Hz'den 250 Hz'ye kadar hızda çalışabilmesidir.

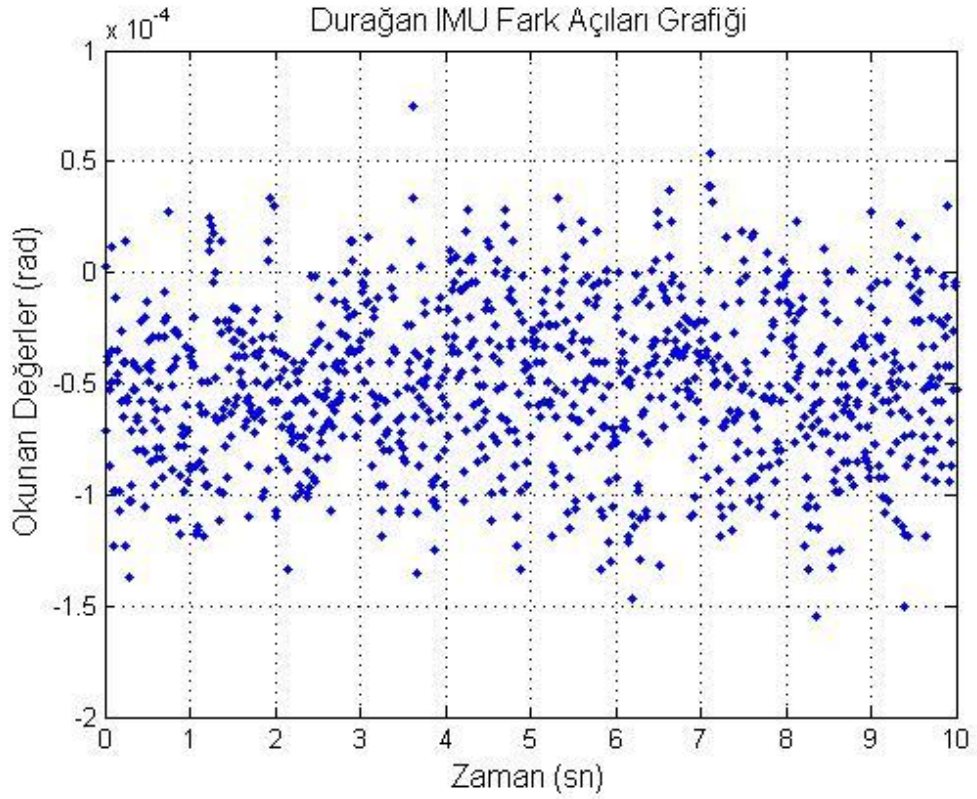
Daha önceden de bahsedildiği gibi kör konumlandırma sistemlerindeki toplanarak büyüyen hata oranlarının ana sebeplerinden birisi yönelim açılarındaki hatalardır. Yönelim açılarındaki hata oranları azaltıldığı takdirde kör konumlandırma sistemlerinin de tutarlılığı aynı oranda artmaktadır. Bu gerçekten yola çıkılarak kör konumlandırma sistemlerinin yönelim açılarındaki hata oranlarının azaltılabilmesi için bu bölümde IMU ve kör konumlandırma sisteminin beraber kullanıldığı yeni bir sistem olan IMU destekli kör konumlandırma sistemi üzerine çalışma ve uygulamalar yapılmıştır.



Şekil 2.8. 3DM-GX2 IMU algılayıcısı.

IMU destekli kör konumlandırma ile güzergâh takibi çalışmasında 3DM-GX2 algılayıcısının fark açıları verisi 100 Hz hızında ve sürekli veri çıkış ayarında kullanılmıştır. Burada dikkat edilmesi gereken, fark açıları IMU algılayıcısının bir önceki yaptığı ölçüm ile şimdiki yaptığı ölçüm arasında geçen sürede yönelim

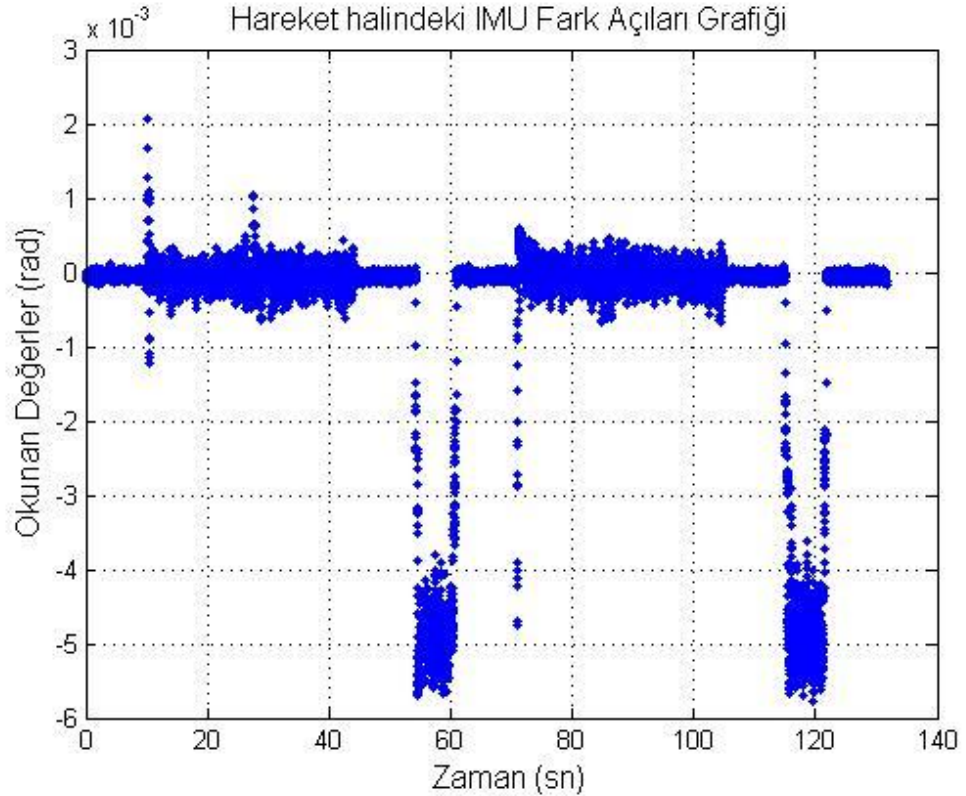
açısındaki değişimi veren kesikli zaman veri çeşididir. Bu özelliğinden dolayı kullanıcı hızının algılayıcı tarafından gönderilen verilerin hepsine yetişebilecek seviyede olması gerekmektedir çünkü kullanıcı algılayıcıdan okunan fark açılarını kendisi toplayarak başlangıç konumuna göre yönelim açısını hesaplamaktadır. Genişletilmiş Kalman Süzgecinin donanımsal olarak uygulanmasına ve birçok düzeltmelerin otomatik olarak algılayıcı içinde yapılmasına rağmen algılayıcıdan okunan veriler gürültülüdür ve bu gürültü yönelim açılarının yanlış hesaplanmasına sebep olmaktadır. Bu gürültünün ölçülebilmesi için robot üzerinde bir deney yapılmıştır. Bu amaçla robot hareketsizken IMU'dan 10 saniye boyunca okunan fark açı verileri analiz edilmek üzere kaydedilmiştir. Bu veriler grafiksel olarak Şekil 2.9.'da gösterilmektedir.



Şekil 2.9. Robot hareketsizken kaydedilen IMU fark açıları grafiği.

Şekil 2.9.'daki verilerin ortalaması  $-0.50732 \text{ rad}$  ve varyansı ise  $0.13309$ 'dur. Ayrıca burada dikkat edilmesi gereken robot hareketsiz olduğu halde IMU'dan okunan verilerin büyük bir kısmı  $-0.5 \text{ rad}$  civarındadır. Okunan bu değerlerin sıfır civarında

değil de genellikle sıfırın altında olması IMU'nun üzerine etki den yer çekim ivmesinden kaynaklanmaktadır. Şekil 2.9.'dan anlaşılacağı üzere robot hareketsiz olduğunda dahi algılayıcıdan değeri  $-1.5 \times 10^{-4} \text{ rad}$  ve  $0.5 \times 10^{-4} \text{ rad}$  aralığında değişen veri geldiğini bir başka deyişle sistemde gürültü olduğunu ifade etmektedir. Şekil 2.10.'da ise bir robot hareket senaryosu boyunca IMU algılayıcısından okunan veriler gösterilmektedir. Bu senaryoda robot 10 saniye hareketsiz kaldıktan sonra 0.02 m/sn hareket etmeye başlamaktadır. 30 saniye düz gittikten sonra 10 saniye durup sonrasında 25 derece/sn açılal hız ile 180 derece yönünü değiştirmektedir. 10 saniye daha durduktan sonra tekrar 0.02 m/sn hızla 30 saniye düz gitmekte ve sonrasında tekrar 10 saniye duraksamaktadır. Daha sonrasında yönünü yine 25 derece/sn açılal hız ile 180 derece değiştirerek eski konum ve yönelimine tekrar gelmektedir bundan sonrasında ise IMU'dan okunan veriler 10 saniye daha kaydedildikten sonra kaydetme işlemi sonlandırılmaktadır. Bu senaryo boyunca robot konumlandırması için sadece kör konumlandırma kullanılmış ve IMU verileri ise sadece kaydedilme amacıyla kullanılmıştır.



Şekil 2.10. Robot hareket halinde iken kaydedilen IMU fark açıları grafiği.

Şekil 2.10.'da görüldüğü gibi robot hareketsizken ve IMU'dan okunan fark açıları  $-1.5 \times 10^{-4} \text{ rad}$  ve  $0.5 \times 10^{-4} \text{ rad}$  aralığında değişirken robot harekete başladığı

anda motorlara ani güç verilmesinden dolayı robot üzerinde bir sarsıntı oluşmakta ve okunan verilerde ani bir artma gözlemlenmektedir. Sarsıntı sona erdikten sonra hareket esnasında okunan veriler  $-0.5 \times 10^{-3} \text{ rad}$  ve  $0.5 \times 10^{-3} \text{ rad}$  aralığında olmaktadır. Bunun sebebi ise robot hareket halinde iken motorların oluşturduğu sarsıntı ve robot tekerleklerinin ideal boyutlarda olmamasından kaynaklanan robot yönelimindeki küçük değişimlerdir. Robotun dönüş hareketi esnasında ise verilerin  $-4 \times 10^{-3} \text{ rad}$  ve  $5.7 \times 10^{-3} \text{ rad}$  aralığında olduğu görülmektedir.

Yapılan bu deneylerle çok küçük açısız dönme hızlarında yani robot düz hareket ettiğinde IMU'dan gelen verinin gürültü oranının çok fazla olduğu açıkça görülmektedir. Diğer yandan ise kör konumlandırmada sistematik hataların oluşma oranının düşük açısız hızlarda yüksek açısız hızlara oranla daha az olduğu bilinmektedir. Kör konumlandırma sistemlerinin yüksek açısız hızlarda hata oranlarının artmasının ana sebebi her kodlayıcı okumasında tekerleklerin daha hızlı dönmesine bağlı olarak daha fazla tur sayısının ölçülmesi ve bunun sonucu olarak sistematik hata oranlarının artmasıdır. Bu sebepten dolayı küçük açısız hızlarda kör konumlandırma hata oranları daha az olmaktadır. Yönelim açısında oluşan hataların da sistematik hataların temelini oluşturduğu düşünüldüğünde yönelim açısındaki hataların azalmasıyla konumlandırmadaki hatalar da azalacaktır.

Bu veriler ışığında IMU verilerinin mutlak değerinin belirli bir eşik değerinin üzerine çıktığı durumlarda yani IMU verilerinin üzerindeki gürültü oranı azaldığında yönelim açısının hesaplanmasında IMU verileri, diğer durumlarda ise kodlayıcı verilerinin kullanılmasıyla kör konumlandırma sistemi hata oranlarının azaltılması sağlanmaktadır. Böylelikle kör konumlandırma sistemi IMU ile desteklenmiş olmakta ve yönelim açısındaki hata oranları azaltılmaktadır. Grafikler dikkate alındığında ve yapılan deneyler sonucunda fark açısı eşikleme<sup>5</sup> değerinin  $4 \times 10^{-3} \text{ rad}$  olarak seçilebileceği uygun görülmüştür.

---

<sup>5</sup> İng : treshholding

### 2.2.2. IMU Destekli K r Konumlandırma İle G zerg h Takibi Deney Sonuları

Bu deneyin amacı, diferansiyel s r ş  zelliğine sahip bir robotun IMU destekli k r konumlandırma ile  nceden belirlenmiř bir g zerg hı ne derece takip edebildiğini saptamaktır. Bunun iin deney esnasında diferansiyel s r ş  zelliğine sahip KheperaIII robotları kullanılmıştır.

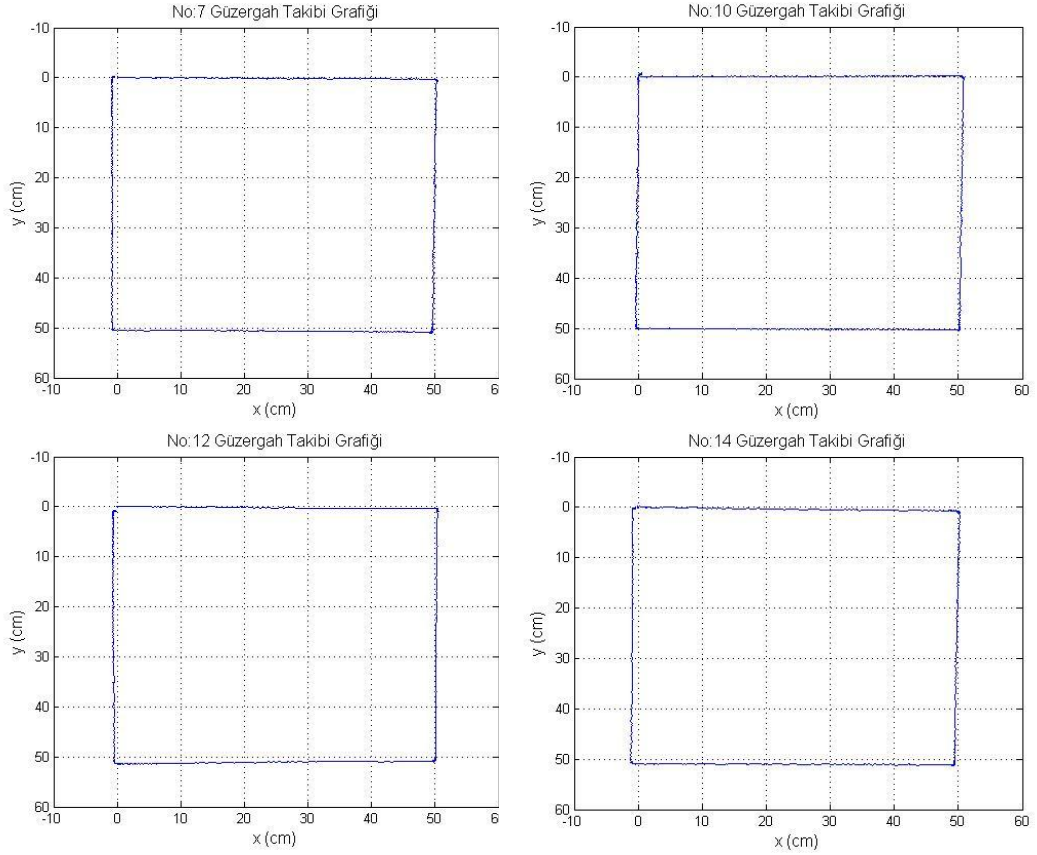
KheperaIII robotunun B l m 2.1.'de anlatılan  zelliklerinin yanı sıra robota 3DM-GX2 IMU algılayıcısı baėlanmıř ve algılayıcıdan RS232 haberleřme y ntemi ile Linux iřletim sisteminde veri okunabilmesi iin gerekli Linux s r c s  ise C programlama ile geliřtirilmiřtir (s r c  hakkında detaylı bilgi Ek 3.2.'de verilmiřtir). Bu baėlantının gerekleřtirilebilmesi ve IMU algılayıcısının g  gereksiniminin karřılanabilmesi iin TOBB Ekonomi ve Teknoloji  niversitesi S r  Sistemler Arařtırma Laboratuvarı'nda geliřtirilen  zel bir seri haberleřme ve besleme kartı (bakınız Ek 3.1) kullanılmıştır. Ek olarak bu karta baėlanan lityum polimer pil sayesinde hem robot ve IMU arasında haberleřme saėlanırken hem de IMU iin g  gereksinimi karřılanabilmektedir. IMU algılayıcı baėlanmıř bir KheperaIII robotu Őekil 2.11'de g sterilmektedir.



Őekil 2.11. IMU baėlı KheperaIII robotu.

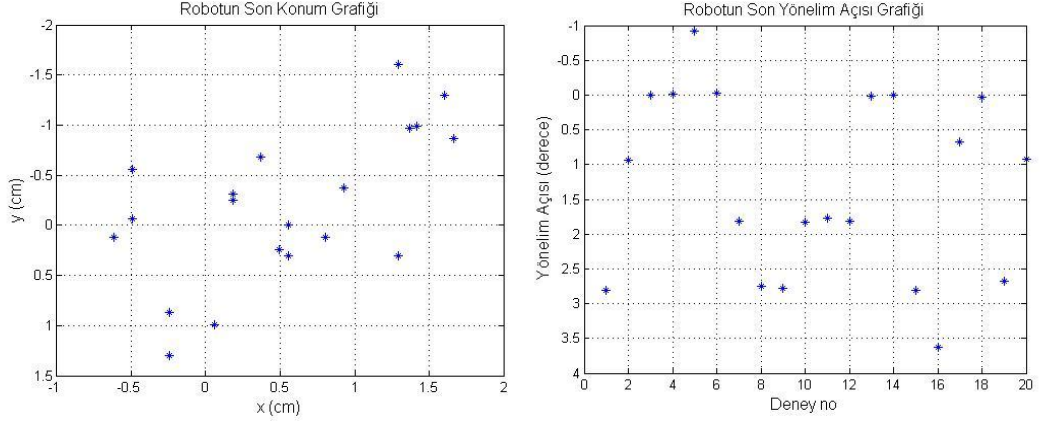
Bölüm 2.2.'de yapılan deneyler aynı şartlar altında IMU destekli kör konumlandırma sistemiyle tekrarlanmıştır. Şekil 2.12.'de robotun IMU destekli kör konumlandırmayla gerçekte takip ettiği güzergâh gösterilmektedir.

Şekil 2.12.'de görüldüğü gibi yönelim açılarındaki iyileşmeler ile robot başladığı konuma geri dönebilmekte ve güzergâh takibi yapabilmektedir. Robotun 20 deney sonunda geldiği son konum ve yönelim açıları Şekil 2.13.'de gösterilmektedir.



Şekil 2.12. IMU destekli kör konumlandırma ile güzergâh takibi örnekleri.

Şekil 2.13.'de robotun son konum grafiğinde gösterilen konumların x eksenindeki ortalaması 0.5341 cm, varyansı 0.5493, y eksenindeki ortalaması -0.1839 cm, varyansı 0.5774 ve son yönelim açısı grafiğinde gösterilen açılarının ortalaması ise 1.3161 derece ve varyansı ise 1.7380'dir.



Şekil 2.13. İMU destekli kör konumlandırma ile robotun bulunduğu son konum ve yönelim açıları.

Daha önce de belirtildiği gibi eğer robotun konumlandırma sisteminde hatalar oluşmasaydı robot her zaman (0, 0) konumuna sıfır derece yönelim açısıyla dönebilecekti ve böylelikle tüm eksenlerdeki ortalama ve varyanslar sıfıra eşit olacaktı. Kör konumlandırma sistemine göre x eksenindeki ortalama yaklaşık % 82, y eksenindeki ortalama yaklaşık % 95 ve yönelim açısında ise yaklaşık % 84 iyileşme olmaktadır. Şekil 2.13.'de görüldüğü gibi İMU destekli kör konumlandırma ile konumlandırmada oluşan sistematik hatalar önemli ölçüde azaltılmıştır. Fakat kör konumlandırma sistemine göre x eksenindeki varyans yaklaşık % 1875, y eksenindeki varyans yaklaşık % 1500 ve yönelim açısı varyansı yaklaşık % 27 artmıştır. Kör konumlandırma sistemindeki hataların büyük çoğunluğu sistematik hatalardan kaynaklandığından dolayı robot her deney sonunda yaklaşık aynı hata değerleri ile çalışmaktadır. Bu sebeple kör konumlandırma sistemindeki varyanslar İMU destekli kör konumlandırma sistemindeki varyanslara göre çok küçük olmaktadır. İMU destekli kör konumlandırma sisteminde ise İMU verileri üzerinde gürültü olduğundan dolayı varyanslar büyük olmaktadır. İMU destekli kör konumlandırma sistemi ile iki mutlak konum güncellemesi zamanı artırabilir ve böylelikle çalışma sisteminin maliyeti düşürülebilir. Detaylı analiz, açıklama ve kör konumlandırma ile karşılaştırmalar Bölüm 2.4.'de mevcuttur.



## 2.3. Parçacık Süzgeci Uygulanarak Güzergâh Takibi

### 2.3.1. Parçacık Süzgeci Hakkında Genel Bilgi

Parçacık süzgecinin ana fikri, ilgili değişkenin<sup>6</sup> (bu tez çalışmasında robotun konumu ve yönelim açısı) diğer bir deyişle parçacıkların ilgili değişken için tahminleri zamanla yayıldıkça genellikle Gaussian olmayan çok-modelli bir olasılık dağılım fonksiyonu ile takip edilmesidir. Bu yöntemin temeli ilgili değişkenin tüm olasılık dağılım fonksiyonunun örnekleme tabanlı bir temsilini oluşturmaktır. Bunun için bir dizi eylem yapılır ve her bir eylem ilgili değişkenin durumunu birkaç modele göre değiştirir. Ayrıca bazı zamanlarda ilgili değişkenin o andaki durumu kısıtlayan gözlemler yapılmaktadır.

İlgili değişkenin birçok kopyası (parçacık) kullanılmaktadır ve bunların her biri parçacığa özgü bir ağırlıkla ilişkilendirilir. Tüm bu parçacıkların ağırlıklı ortalaması alınarak ilgili değişkenin değeri hakkında bir tahminde bulunulur. Parçacık süzgeci özyinelemeli bir algoritmadır ve iki safhadan oluşur: tahmin ve güncelleme. Her eylemden (robotun bir konumdan diğer bir konuma hareketi) sonra gürültünün ilgili değişkenin üzerindeki etkisini temsil etmesi için rastgele bir gürültü de dahil edilerek kullanılan modele göre ilgili değişken değiştirilmektedir (tahmin safhası). Daha sonra her bir parçacığın ağırlıkları en son alınan algılayıcı verilerine göre tekrardan hesaplanır (güncelleme safhası). Diğer bir deyişle  $t = k, k \in \mathbb{Z}$  zamanında ilgili değişkeni (bu robot konumlandırması için  $\mathbf{x}^k = [x^k, y^k, \theta^k]^T$ ) içinde  $M$  adet örnek bulunduran bir küme tarafından temsil edilmektedir ( $S_j^k = [\mathbf{x}_j^k, w_j^k]: j = 1 \dots M$ ), burada  $j$  indisi parçacığı temsil etmektedir, her parçacık ilgili değişkenin bir kopyası ve değişkenin yayılan tahminine olan katkısını ifade eden ağırlık ( $w_j^k$ )'tan oluşmaktadır.

Eğer  $t = k$  zamanında sistemin bir önceki zamandaki ( $t = k-1$ ) olasılık dağılım fonksiyonu biliniyorsa  $t = k$  zamanında tahmini bir ön olasılık dağılım fonksiyonu bulunabilmesi için eylemin etkileri modellenabilir (tahmin). Diğer bir deyişle, tahmin

---

<sup>6</sup> İng : variable of interest

safhası ilgili deęişkenin parçacıklar üzerindeki etkisini temsil eden gerekli gürültü eklenmiş bir model kullanmaktadır. Güncelleme safhası hareket eden robotun olasılık dağılım fonksiyonunu tam olarak tanımlanması amacı ile parçacık ağırlıklarının güncellenmesi için algılayıcılardan alınan verileri kullanır.

Bir parçacık dağılımı verildiğinde, genellikle robotun konumuna göre bazı işlemler yapmamız gerekmektedir. Konum tahmini için 3 farklı deęerleme yöntemi mevcuttur. Birincisi ağırlıklı ortalama ( $P_{tah} = \sum_{j=1}^M w_j x_j$ ) kullanılabilir, ikincisi en iyi parçacık ( $P_j$  öyle ki  $w_j = \max(w_k) : k = 1 \dots M$ ) ve üçüncüsü gürbüz ortalama olarak da bilenen en iyi parçacığın etrafında küçük bir çerçevede ağırlıklı ortalama kullanılabilir. Her yöntemin kendine has avantajları ve dezavantajları vardır: en iyi parçacık yöntemi ayırıklaştırma hatalarına<sup>7</sup> sebep olurken, çok modelli dağılımlarda ağırlıklı ortalama sonuç vermemektedir. En iyi yöntem gürbüz ortalama yöntemidir fakat bunun yanında çok fazla işlem yükü getirmektedir.

Robotun bir yerden bir yere hareketinden sonra konumunun olasılık dağılımını tahmin etmek için gürültünün son konumun üzerindeki etkisinin bir modeli bulunması gerekmektedir. Bunun için birçok deęişik yöntem kullanılmaktadır (bazı yöntemlerin detaylı anlatımı için bakınız [35, 36]), bu yöntemlerin çoğunda robot hareketi için katkılı Gaussian gürültü modeli kullanılmaktadır. Herhangi bir isteğe baęlı hareket  $[\Delta x, \Delta y]^T$ , yönelim deęiştirmeyi takiben yer deęiştirme olarak yapılabilmektedir. Burada robotun hareketine başladığı konumu  $[x, y, \theta]^T$ 'dir. İlk önce robot yönelim açısını  $\delta\theta = \theta_k - \theta$  kadar deęiştirmektedir, burada robotun hedef konum yönüne dönmesi için  $\theta_k = \arctan(\Delta y / \Delta x)$  olarak ifade edilmektedir ve daha sonra robot  $\rho = \sqrt{\Delta x^2 + \Delta y^2}$  kadar ileri yönde bir yer deęiştirme yapmaktadır. Eđer başlangıç konumu  $[x, y, \theta]^T$  ise robotun son konumu Denklem 2.7'de verilen  $[x', y', \theta_k]^T$  olmaktadır. Sonuç olarak gürültü modeli iki çeşit hareket biçimine ayrı ayrı uygulanmaktadır çünkü iki hareketin birbirinden bağımsız olduğu varsayılmaktadır. Yönelim ve yer deęiştirme ile ilgili tüm denklemler ve detaylar I.

---

<sup>7</sup> İng : discretization error

Rekleitis tarafından [37] 'de anlatılmaktadır ve bu tez çalışmasındaki uygulama [37]'deki anlatım doğrultusunda yapılmıştır.

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x + \rho \cos(\theta^k) \\ y + \rho \sin(\theta^k) \\ \theta^k \end{bmatrix} \quad (2.7)$$

Parçacık süzgeçlerinin kullanımlarında ortaya çıkan diğer bir problem ise sadece birkaç yinelemeden sonra parçacık sayısının tükenmesidir. Birçok parçacığın konum tahminleri robotun gerçek konumundan oldukça uzaklaşıp ağırlıklarının hareket eden robotun olasılık dağılım fonksiyonuna katkıda bulunamayacak kadar küçülmesidir. Eğer kullarımdaki parçacıkların kümesi  $S_k = \{x_i^k, w_i^k\} : k = 1 \dots M$  hareket eden robotun konumunun olasılık dağılım fonksiyonunun kesikli ifadesi olarak kabul edilirse, aynı olasılık dağılım fonksiyonunu ifade eden  $x'_k = x_i^{l'}$ ,  $k, l$  ise  $[1, M]$  aralığında ve ağırlıkları  $w_i^{l'k} = 1/M$  olan yeni bir  $S'_k = \{x_i^{l'k}, w_i^{l'k}\} : k = 1 \dots M$  gerekmektedir.

Liu [38] ağırlıkları sıfıra yakın olan parçacıkları belirlemek için iki farklı ölçüm önermektedir. Birincisi değişken katsayısı  $cv_t^2$  (bakınız Denklem 2.8), ve ikincisi ise etkin örnekleme büyüklüğü  $ESS_t$  (bakınız Denklem 2.9)'dir.

$$cv_t^2 = \frac{\text{var}(w_t(i))}{E^2(w_t(i))} = \frac{1}{M} \sum_{i=1}^M (Mw(i) - 1)^2 \quad (2.8)$$

$$ESS_t = \frac{1}{1+cv^2} \quad (2.9)$$

$w(i)$  : i. parçacığa ait ağırlık

$t$  : zaman

$w_t(i)$  : t zamanında i. parçacığın ağırlığı

$M$  : toplam parçacık sayısı

Denklem 2.8 ve 2.9 için gerekli detaylı açıklamalar Liu [38] tarafından yapılmaktadır. Etkin örnekleme büyüklüğü belirli bir eşik değerinin altına düştüğünde, genellikle  $M$  adet parçacık sayısının belirli bir yüzdesi, olasılık yöntemleri ile küçük ağırlıklı parçacıklar elenmekte ve büyük ağırlıklı parçacıkların bir kopyası daha üretilerek sistemdeki parçacıklar yeniden örneklenmektedir.

Yeniden örneklendirme için çeşitli yöntemler önerilmektedir [37]. Tüm yöntemlerde girdi parçacıkların ağırlıklarından oluşan bir dizi, çıktı ise neslini devam ettirecek olan parçacıkların indislerinden oluşan bir dizidir. Tüm yöntemlerde ihtiyaç duyulan yeniden örnekleme öncesindeki çok benzer olan yeniden örneklenmiş parçacık kümesi tarafından oluşturulmuş olasılık dağılım fonksiyonudur.

Parçacık süzgecinde yeniden örneklendirme işleminden farklı olarak yapılan diğer bir işlem ise güncelleme safhasıdır. Bu safhada mutlak konum bilgisi ulaşılabilir olduğu durumlarda kullanılan her bir parçacık için bu mutlak konuma göre her parçacığın doğruluk olasılığı hesaplanmakta (Denklem 2.10) ve bu bilgilere göre kullanılan her parçacığın ağırlığı olasılıklarına eşitlenerek yani  $w_i^k = P(\mathbf{x}_i^k)$  denklemi ile güncellenmektedir.

$$P(\mathbf{x}_i^k) = \frac{1}{\sqrt{2\pi}\sigma_\rho} \frac{e^{-\frac{\rho^2}{2\sigma_\rho^2}}}{\sqrt{2\pi}\sigma_d} \frac{e^{-\frac{\Delta\theta_d^2}{2\sigma_d^2}}}{\sqrt{2\pi}\sigma_r} \frac{e^{-\frac{\Delta\theta_r^2}{2\sigma_r^2}}}{\sqrt{2\pi}\sigma_r}$$

$$\Delta x = x_k - x_{k-1}$$

$$\Delta y = y_k - y_{k-1}$$

$$\Delta\theta = \theta_k - \theta_{k-1} \tag{2.10}$$

$$\rho = \sqrt{\Delta x^2 + \Delta y^2}$$

$$\Delta\theta_r = \arctan(\Delta y / \Delta x)$$

$$\Delta\theta_d = \Delta\theta - \Delta\theta_r$$

Burada

$\Delta x$  : robotun son konumu ile bir önceki konumu arasındaki x eksenleri farkı

$\Delta y$  : robotun son konumu ile bir önceki konumu arasındaki y eksenleri farkı

$\Delta\theta$  : robotun son konumu ile bir önceki konumundaki yönelim açıları farkı

$\rho$  : robotun hareketi sonunda değiştirdiği yer miktarı

$\Delta\theta_r$  : robotun son konumuna yönelmek için yaptığı dönüş miktarı

$\Delta\theta_d$  : robotun başlangıç konumundan son konumuna gelirken yöneliminde meydana gelen kaymalar

$\sigma_p$  : yer deęiřtirme standart sapması

$\sigma_r$  : ynelim deęiřtirme standart sapması

$\sigma_d$  : ynelim aısında meydana gelen aı kaymaları standart sapması

### 2.3.2. Paracak Szgeci Parametrelerinin Hesaplanması

Bir nceki blmde anlatılan paracak szgelerinin uygulanabilmesi iin bir takım parametrelere ihtiya duyulmaktadır. Bu parametreler paracıkların konumlarına ve ynelim aılarına eklenen rastgele Gaussian grlt fonksiyonunun ve paracıkların aęırlıklarını gncellemek iin kullanılan olasılık daęılım fonksiyonlarının girdilerinden bazılarını oluřturmaktadırlar. Robota has zellikler olan  $\sigma_p$  (yer deęiřtirme standart sapması),  $\sigma_r$  (ynelim deęiřtirme standart sapması),  $\sigma_d$  (ynelim aısında meydana gelen aı kaymaları standart sapması),  $M_p$  (yer deęiřtirme hata ortalaması),  $M_{\theta_r}$  (ynelim deęiřtirme hata ortalaması) ve  $M_{\theta_d}$  (ynelim aısında meydana gelen aı kaymaları ortalaması) gibi parametreler paracak szgeci uygulanmadan nce Rekleitis'in [37]'de anlattıęı Őekilde bulunması gerekmektedir.

Bu deęerlerin bulunabilmesi iin mutlak konumlandırma sistemlerine ihtiya duyulmaktadır. Bunun iin laboratuvar ortamında bulunan tepe kamerası yardımıyla deney esnasında robotun gerek konumu tespit edilerek standart sapmalar ve ortalamalar bulunmuřtur. Robotun  $\sigma_r$  ve  $M_{\theta_r}$  deęerlerinin bulunabilmesi iin EK 2.'de anlatılan dzenek kullanılmıřtır. Bu amala robot bir bařlangı ynelim aısında bařlayıp kendi eksenini etrafında IMU destekli kr konumlandırma sistemini kullanarak 360 derece dnř yapıp durmaktadır. Kamera yardımıyla bulunan robotun dnře bařlamadan nceki ynelim aısı ve dnř tamamladıktan sonraki ynelim aısı kullanıcı tarafından kaydedilmektedir. Bu deney 100 kez tekrarlandıktan sonra kaydedilen deęerler ile  $\sigma_r$  ve  $M_{\theta_r}$  deęerleri hesaplanmıřtır.

Aynı dzenek paracak szgecinin dięer parametrelerinin de bulunması iin kullanılmıřtır. Bu amala robot IMU destekli kr konumlandırma sisteminin kullanarak 1 metre dz gidip durmaktadır. Deney esnasında kamera yardımıyla bulunan robotun bařlangı konumu ve ynelim aısı, son geldięi konum ve ynelim aısı kullanıcı tarafından kaydedilmektedir. Bu deney 100 defa tekrarlandıktan sonra

kaydedilen deęerler üzerinden  $\sigma_\rho$ ,  $M_\rho$ ,  $\sigma_d$  ve  $M_{\theta_d}$  deęerleri tespit edilmiřtir. Deneyle sonucunda hesaplanan ilgili deęerler Tablo 2.1.'de gsterilmektedir.

$\sigma_r = 1.7844$	$\sigma_\rho = 1.3052$	$\sigma_d = 1.3807$
$M_{\theta_r} = -0.9888$	$M_\rho = 100.3734$	$M_{\theta_d} = 2.4224$

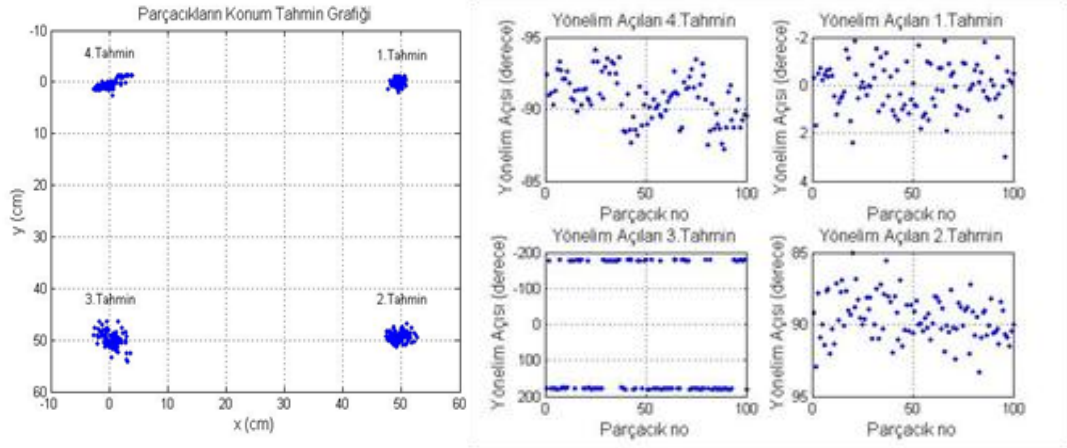
Tablo 2.1. Paracık szgecinde kullanılan standart sapma ve ortalama deęerler.

### 2.3.3. Paracık Szgeci Uygulanarak Gzergh Takibi Deney Sonuları

Bu blmde paracık szgelerinin gncelleme safhasında mutlak konumlandırma verileri yerine IMU destekli kr konumlandırma sistemi verilerinin gerek konum verileri olduęu kabul edilerek paracık szgecinin sisteme olan etkisi incelenmiřtir. Bu inceleme ile paracık szgecinin mutlak konumlandırma sistemleri olmadıęı durumlarda IMU destekli kr konumlandırma sistemine bir iyileřtirme getirip getiremeyeceęi arařtırılmıřtır.

Bu deneyin amacı, diferansiyel srř zellięine sahip bir robotun paracık szgeci uygulanan IMU destekli kr konumlandırma ile nceden belirlenmiř bir gzerghı ne derece takip edebildięini saptamaktır. Bunun iin deney esnasında diferansiyel srř zellięine sahip KheperaIII robotları kullanılmıřtır.

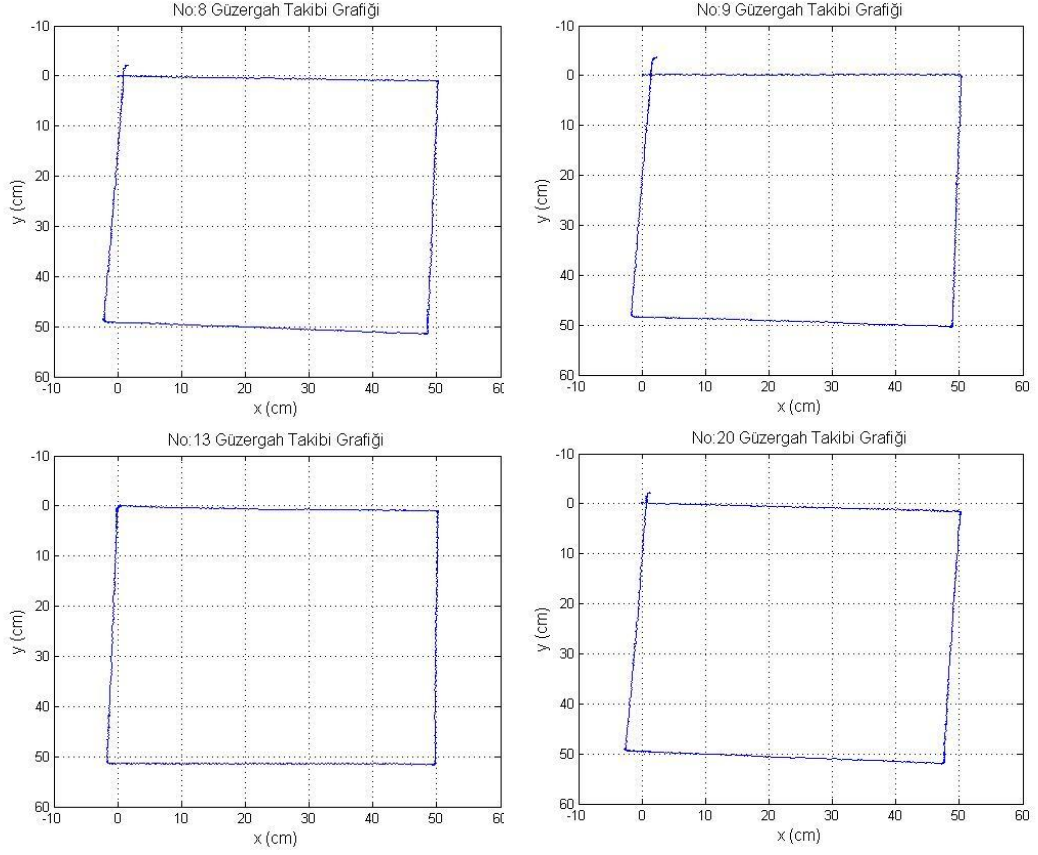
Blm 2.2.'de yapılan deneyler aynı řartlar altında 100 adet paracıęı olan bir paracık szgeci uygulanarak IMU destekli kr konumlandırma sistemi ile tekrarlanmıřtır. Robot gzerghin kře noktalarına ulařtıęında IMU destekli kr konumlandırma verileri mutlak konumlandırma verileri yerine paracık szgecinde girdi olarak verilerek paracıkların aęırlık gncellemeleri ve gerektięinde paracıkların yeniden rneklendirmeleri yapılmaktadır. řekil 2.14.'te robotun paracık szgeci uygulanan IMU destekli kr konumlandırma ile gzergh takibi yaparken paracıkların aldıęı deęerler, gncellemeler gsterilmektedir.



Şekil 2.14. Parçacıkların köşe noktalarındaki konum ve yönelim tahminleri.

Şekil 2.14.'te sol taraftaki büyük şekilde parçacık güzergâh takibi sırasında parçacıkların konum tahminleri görülmektedir. Şekildeki küçük grafiklerden sağ üst kısımdaki veri kümesi parçacıkların konum hakkındaki ilk tahminlerini, sağ alt kısımdaki ikinci, sol alt kısımdaki üçüncü ve sol üst kısımdaki veri kümesi ise parçacıkların konum hakkındaki son tahminlerini ifade etmektedirler. Parçacıkların ilk tahminlerinde tahmin kümesi diğerlerine göre daha dar bir alan içerisinde olurken diğer tahminlerde ise giderek artmaktadır. Bunun sebebi ise konumdaki hataların toplanarak artması ve dolayısıyla tahminlerin giderek yayılmasıdır. Son tahmin veri kümesine dikkat edilirse parçacıkların tekrardan örneklenerek çok küçük ağırlığı olan parçacıkların elenmesi ve yerine ağırlığı büyük parçacıkların kopyasının getirilmesinden dolayı bir önceki tahmin kümesine göre daha az yayılma göstermektedir. Böylelikle parçacıkların ağırlıklı ortalamaları kullanılarak gerçek konuma en yakın tahminin yapılması amaçlanmaktadır.

Şekil 2.14.'te sağ taraftaki küçük şekillerde parçacıkların güncellemelerden sonra tahminde buldukları yönelim açıları gösterilmektedir. Şekiller dikkatle incelendiğinde yönelim açılarındaki giderek artan yayılmalar yeniden örnekleme olduktan sonra azalmaktadır. Diğer bir deyişle yönelim güncellemesi için ağırlıklı ortalama kullanıldığından, ortalamadaki küçük ağırlıklı parçacıklar azaldığından ve büyük ağırlıklı parçacıklar kopyalandığından yönelim dağılımında bir toplanma görülmektedir. Aynı toplanma konum dağılımlarında da görülmektedir.



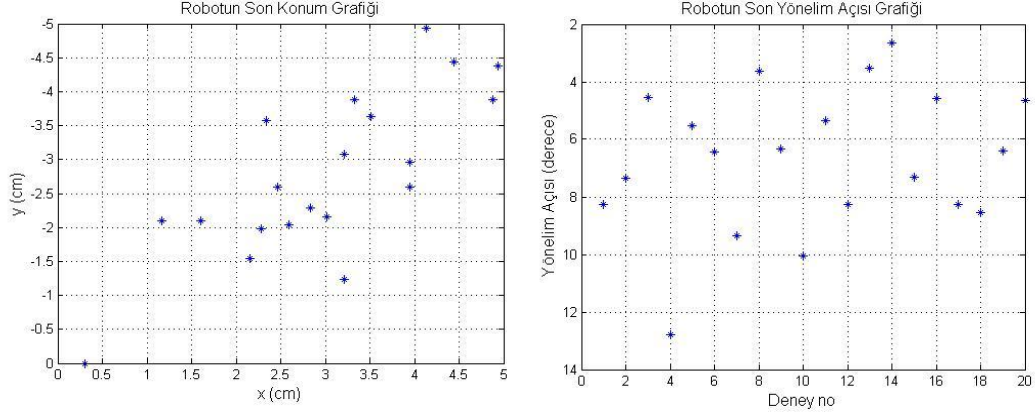
Şekil 2.15. Parçacık süzgeci uygulanarak IMU destekli kör konumlandırma ile güzergâh takibi örnekleri.

Şekil 2.15.'te ise robotun parçacık süzgeci uygulanılarak IMU destekli kör konumlandırma ile gerçekte takip ettiği güzergâh örnekleri gösterilmektedir. Şekilde görüldüğü gibi güncelleme olana kadar IMU destekli kör konumlandırma sistemi ile yaklaşık aynı performansı gösteren parçacık süzgeçli konumlandırma parçacık ağırlıklarının güncellenmesiyle güzergâhtan sapmalar göstermektedir. Sapmaların daha çok köşe noktalarda yani tahminler ve güncellemeler olduktan sonra meydana geldiği gözlemlenmektedir.

Şekil 2.16.'da ise robotun 20 deney sonunda geldiği son konum ve yönelim açıları gösterilmektedir. Şekil 2.16.'de robotun son konum grafiğinde gösterilen konumların x eksenindeki ortalaması 3.0168 cm, varyansı 1.4495, y eksenindeki ortalaması -2.77 cm, varyansı 1.4951 ve son yönelim açısı grafiğinde gösterilen açıların ortalaması ise 6.6921 derece ve varyansı ise 6.2991'dir. Daha önce de belirtildiği gibi eğer robotun



konumlandırma sisteminde hatalar oluşmasaydı robot her zaman (0, 0) konumuna sıfır derece yönelim açısıyla dönebilecekti ve böylelikle tüm eksenlerdeki ortalama ve varyanslar sıfıra eşit olacaktı.



Şekil 2.16. Parçacık süzgeci uygulanarak IMU destekli kör konumlandırma ile robotun bulunduğu son konum ve yönelim açıları.

Şekil 2.16.'de görüldüğü gibi parçacık süzgeci uygulanarak IMU destekli kör konumlandırma ile konumlandırmada oluşan sistematik hatalar az miktarda da olsa kör konumlandırma sistemine göre azalma göstermektedir. Fakat diğer yandan IMU destekli kör konumlandırma sistemine göre bozucu bir etken olarak sisteme yansıdığı açıkça anlaşılabilir. Daha detaylı analiz ve açıklamalar Bölüm 2.4.'te mevcuttur.

#### 2.4. Güzergâh Takip Sonuçlarının Karşılaştırılması

Kör konumlandırma sistemi ile güzergâh takibi sonuçları incelendiğinde konumlandırma sisteminde oluşan hataların toplanarak büyüdüğü gözlemlenmektedir. Yapılan 20 deneyin sonunda kör konumlandırma sisteminde son konumlarda ve yönelim açılarında meydana gelen hatalar (Denklem 2.11) ise Şekil 2.17.'de gösterilmektedir. Bu hata grafikleri elde edilirken güzergâh takibi için kullanılan kontrol programının hatalarının sistem dışında tutulması için robotun kör konumlandırma sistemi ile hesaplanan son konum ve yönelim açıları da hata denklemlerine dahil edilmektedir. Kör konumlandırma iyileştirmesi üzerine çalışma yapan Borenstein ve Feng [5]'in çalışmalarından yola çıkılarak ayrıca hata analizi

yapılırken robotun sadece son bulunduğu konum ve yönelim açıları dikkate alınmıştır. Deneyler sonunda elde edilen robot konumlarının ortalama mutlak hataları ise Denklem 2.12'ye göre hesaplanmıştır.

$$\begin{aligned}\varepsilon x &= x_{ger} - x_{hes} \\ \varepsilon y &= y_{ger} - y_{hes} \\ \varepsilon \theta &= \theta_{ger} - \theta_{hes}\end{aligned}\tag{2.11}$$

$\varepsilon x$  : robotun x eksenindeki hatası

$\varepsilon y$  : robotun y eksenindeki hatası

$\varepsilon \theta$  : robotun yönelim açısındaki hatası

$x_{ger}, y_{ger}, \theta_{ger}$  : robotun eksenlerdeki gerçek konumları ve yönelim açısı

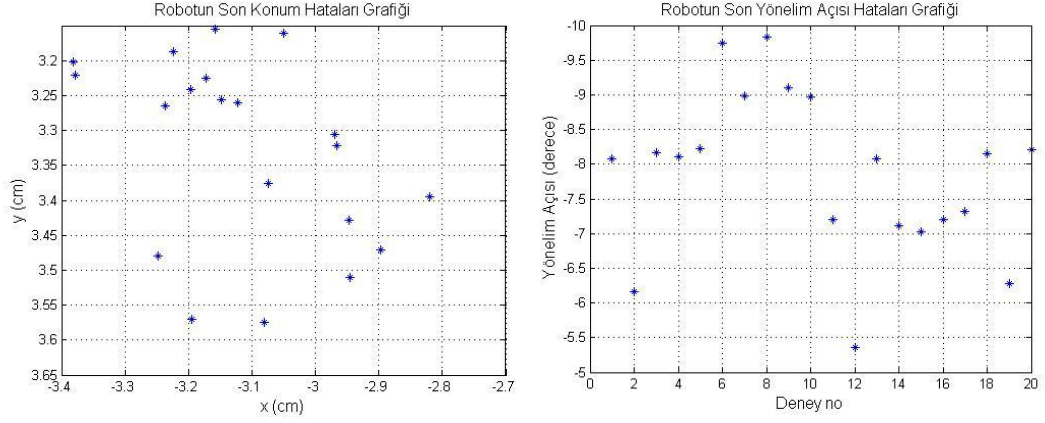
$x_{hes}, y_{hes}, \theta_{hes}$  : robotun kendi hesapladığı eksenlerdeki konumları ve yönelim açısı

$$\begin{aligned}\varepsilon x_{ort} &= \frac{1}{20} \sum_{i=1}^{20} |\varepsilon x_i| \\ \varepsilon y_{ort} &= \frac{1}{20} \sum_{i=1}^{20} |\varepsilon y_i| \\ \varepsilon \theta_{ort} &= \frac{1}{20} \sum_{i=1}^{20} |\varepsilon \theta_i|\end{aligned}\tag{2.12}$$

Hata analizleri için önemli olan diğer bir konum hata ifadesi yöntemi ise robotun son konumunun başlangıç konumuna olan uzaklığıdır. Bu yöntemde ise robotun her deney için x ve y eksenindeki son konumunun başlangıç noktasına olan uzaklıkları bulunarak, tüm deneylerin sonunda bu uzaklıkların ortalama karesel hataları hesaplanmaktadır (Denklem 2.13). Yönelim açısındaki hata ortalaması ise aynı yöntem ile yapılmaktadır.

$$\varepsilon m_{ort} = \frac{1}{20} \sum_{i=1}^{20} \sqrt{\varepsilon x_i^2 + \varepsilon y_i^2}\tag{2.13}$$

Burada  $\varepsilon m_{ort}$  başlangıç konumuna olan mesafelerin ortalamasını ifade etmektedir.



Şekil 2.17. Kör konumlandırma ile güzergâh takibinde son konum ve yönelim hataları grafiği.

Şekil 2.17.'de görüldüğü gibi 20 adet kör konumlandırma ile güzergâh takibi deneyi sonucunda robotun son konumları hataları x ekseninde -3.4 cm ile -2.8 cm arasında, y ekseninde 3.15 cm ile 3.6 cm arasında ve yönelim açısında ise -5 ile -10 derece arasında değişmektedir. Bu sonuçlardan anlaşıldığı gibi sadece kör konumlandırma sistemini kullanan bir mini robot toplamda x ekseninde sadece 100 cm, y ekseninde 100 cm yol almasına ve yönelim açısında ise 360 derece değişiklik yapmasına rağmen robotun konumundaki hata, deneyin sonunda alınan yol ve yönelim açısındaki değişime göre oldukça fazladır. Kör konumlandırma ile yapılan deneyler sonucunda elde edilen hata ortalamaları aşağıdaki gibidir.

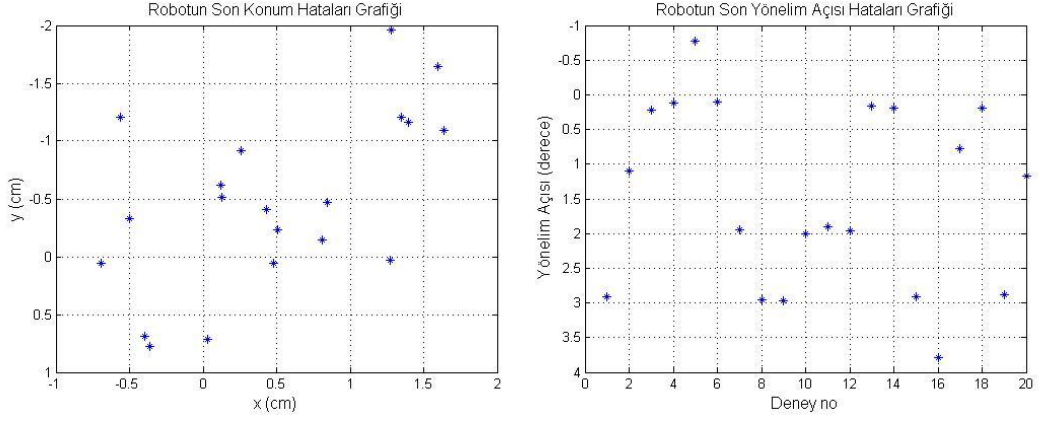
$$\varepsilon x_{ort} = 3.11 \text{ cm}$$

$$\varepsilon y_{ort} = 3.33 \text{ cm}$$

$$\varepsilon m_{ort} = 4.56 \text{ cm}$$

$$\varepsilon \theta_{ort} = 7.86^\circ$$

Kör konumlandırma ile yapılan deneyler aynı şartlar altında IMU destekli kör konumlandırma sistemi ile tekrarlandığında robotun son konumlarında ve yönelim açılarında meydana gelen hatalar Şekil 2.18.'da gösterilmektedir.



Şekil 2.18. IMU destekli kör konumlandırma ile güzergâh takibinde son konum ve yönelim hataları grafiği.

Şekil 2.18.'de görüldüğü gibi 20 adet IMU destekli kör konumlandırma ile güzergâh takibi deneyi sonucunda robotun son konumları hataları x ekseninde -0.75 cm ile 1.7 cm arasında, y ekseninde -2 cm ile 0.75 arasında ve yönelim açısında ise -1 ile 3.75 derece arasında değişmektedir. IMU destekli kör konumlandırma ile yapılan deneyler sonucunda elde edilen hata ortalamaları aşağıdaki gibidir.

$$\varepsilon x_{ort} = 0.73 \text{ cm}$$

$$\varepsilon y_{ort} = 0.71 \text{ cm}$$

$$\varepsilon m_{ort} = 1.1 \text{ cm}$$

$$\varepsilon \theta_{ort} = 1.55^\circ$$

Kör konumlandırma ile yapılan deneyler aynı şartlar altında parçacık süzgeci uygulanarak IMU destekli kör konumlandırma sistemi ile tekrarlandığında robotun son konumlarında ve yönelim açılarında meydana gelen hatalar Şekil 2.19.'da gösterilmektedir.

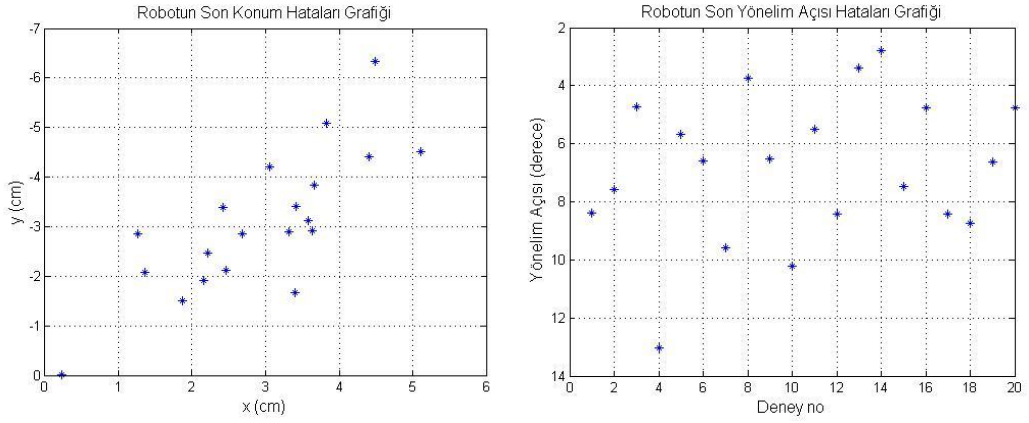
Şekil 2.19.'de görüldüğü gibi 20 adet parçacık süzgeci uygulanarak IMU destekli kör konumlandırma ile güzergâh takibi deneyi sonucunda robotun son konumları hataları x ekseninde 0.2 cm ile 5.1 cm arasında, y ekseninde -6.4 cm ile 0 cm arasında ve yönelim açısında ise 3 ile 13 derece arasında değişmektedir. Parçacık süzgeci uygulanarak IMU destekli kör konumlandırma ile yapılan deneyler sonucunda elde edilen hata ortalamaları aşağıdaki gibidir.

$$\varepsilon x_{ort} = 2.93 \text{ cm}$$

$$\varepsilon y_{ort} = 3.08 \text{ cm}$$

$$\varepsilon m_{ort} = 4.29 \text{ cm}$$

$$\varepsilon \theta_{ort} = 6.85^\circ$$



Şekil 2.19. Parçacık süzgeci uygulanarak IMU destekli kör konumlandırma ile güzergâh takibi son konum ve yönelim hataları grafiği.

	<b>Kör K.</b>	<b>IMU Kör K.</b>	<b>PS. IMU Kör K.</b>
$\varepsilon x_{ort}(cm)$	3.11	0.73	2.93
$\varepsilon y_{ort}(cm)$	3.33	0.71	3.08
$\varepsilon m_{ort}(cm)$	4.56	1.1	4.29
$\varepsilon \theta_{ort}(derece)$	7.86	1.55	6.85
<b>% İyileşme</b>			
$\varepsilon x_{ort}$	0	76.53	5.79
$\varepsilon y_{ort}$	0	78.68	7.5
$\varepsilon m_{ort}$	0	75.88	5.92
$\varepsilon \theta_{ort}$	0	80.28	12.85

Tablo 2.2. Güzergâh takibi ortalama hataları ve yüzdeler iyileşmeler tablosu.

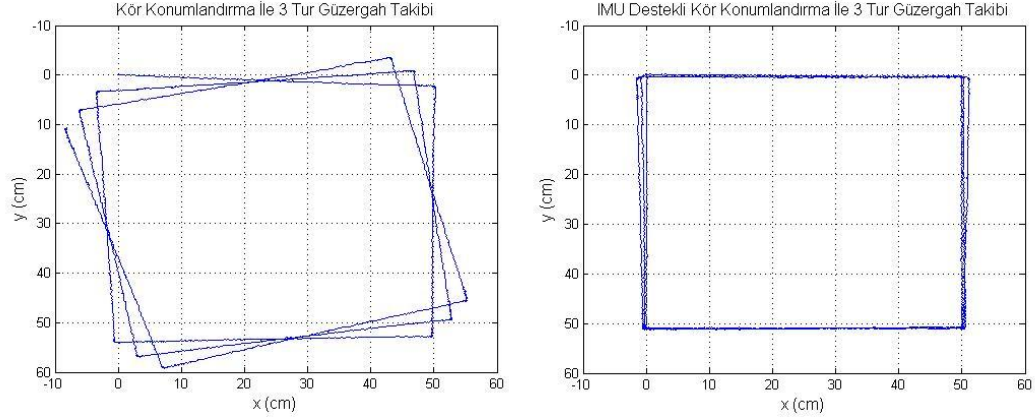
3 farklı konumlandırma yöntemiyle güzergâh takibi deney sonuçları için son konum ve yönelim açıları için yapılan hata hesapları kör konumlandırma sistemine göre yüzdelik iyileşmeler Tablo 2.2.'de gösterilmektedir.

Parçacık süzgeciyle IMU destekli kör konumlandırma sisteminde iyileştirme sağlanamamasının en büyük sebebi parçacık ağırlıklarının güncelleştirilmesi safhasında gerekli olan mutlak konumlandırma sistemi verileri yerine yine kendi çıktılarının gerçek veriler olarak kabul edilmesidir. Bu durumda parçacık ağırlıkları yanlış veriye göre güncellenip ağırlıklı ortalamanın gerçek değerden sapmasına sebep olmaktadır. Parçacık süzgecinin iyileştirme sağlayamamasının diğer bir sebebi ise parçacık süzgeci için gerekli olan standart sapmalar hesaplanırken mutlak konumlandırma için kullanılan tepe kamerası sisteminin yönelim bulmadaki hatasının yaklaşık  $\pm 3^\circ$  olmasıdır. Halbuki IMU destekli konumlandırma sisteminin yönelimdeki hatası yaklaşık  $\pm 1.5^\circ$  olmaktadır. Yönelim açısı hesabındaki artan hataysa standart sapmaların yanlış hesaplanmasına sebep olmaktadır. Daha iyi bir mutlak konumlandırma sistemi mevcut olsaydı ve başlangıçta parçacık süzgecinde kullanılan parametreler daha iyi hesaplanabilseydi parçacık süzgecinin yüzdelik iyileşmesi daha iyi olacaktı. Fakat parçacık süzgecinin ana fikrinde mutlak konumlandırma sistemi ulaşılabilir olduğunda parçacık ağırlıklarının gerçek veri ile güncellenmesi vardır. Yani gerçek veriye daha yakın olan parçacığın ağırlığı da daha fazla olmaktadır. Eğer bu ağırlık güncellemeleri doğru veriye göre değil de tahmin edilen veriye göre güncellenirse ağırlıklar yanlış güncellenecek ve sistemdeki hatalar büyümeye başlayacaktır.

Bu verilerden çıkarabilecek sonuç IMU algılayıcısının kullanılması kör konumlandırma sisteminin iyileştirilmesine önemli ölçüde katkı sağlamaktadır. Bu iyileştirme Şekil 2.20.'de açıkça görülmektedir.

Şekil 2.20.'de kör konumlandırma sistemi ve IMU destekli kör konumlandırma sistemi ile yapılan bir deney sonucu gösterilmektedir. Bu deneyde robot daha önce anlatılan güzergâh takibi deneyini durmaksızın 3 kez tekrarlamaktadır. Yani robot 3 kat daha fazla mesafe kat etmektedir ve yönelim açısını 3 kat daha fazla değiştirmektedir. Şekil 2.20.'den de anlaşılacağı gibi IMU destekli kör

konumlandırma sisteminde konumlandırma ve yönelim açısındaki hatalar kör konumlandırma sistemine göre daha yavaş bir hızla büyümekte ve çok daha tutarlı bir takip ortaya koymaktadır.



Şekil 2.20. IMU destekli kör konumlandırma ile güzergâh takibinin iyileştirilmesi grafiği.

## BÖLÜM 3

### 3. TEK-ROBOTLU HARİTALAMA SİSTEMLERİNDE KONUM TUTARLILIĞININ HARİTA ÜZERİNE ETKİLERİ

#### 3.1. Kör Konumlandırma Sistemi İle Tek-Robotlu Haritalama

Haritalama yapan bir robotun haritaya yeni bilgiler ekleyebilmesi için her şeyden önce kendisinin nerede ve yönelim açısının ne olduğunu bilmesi gerekmektedir. Birçok kapalı ortam robotu konumlarını hesaplamak için kör konumlandırma sistemlerini kullanmaktadır fakat bu sistemlerdeki sistematik ve sistematik olmayan hatalardan dolayı konumlandırmadaki hatalar zamanla toplanarak büyümektedir.

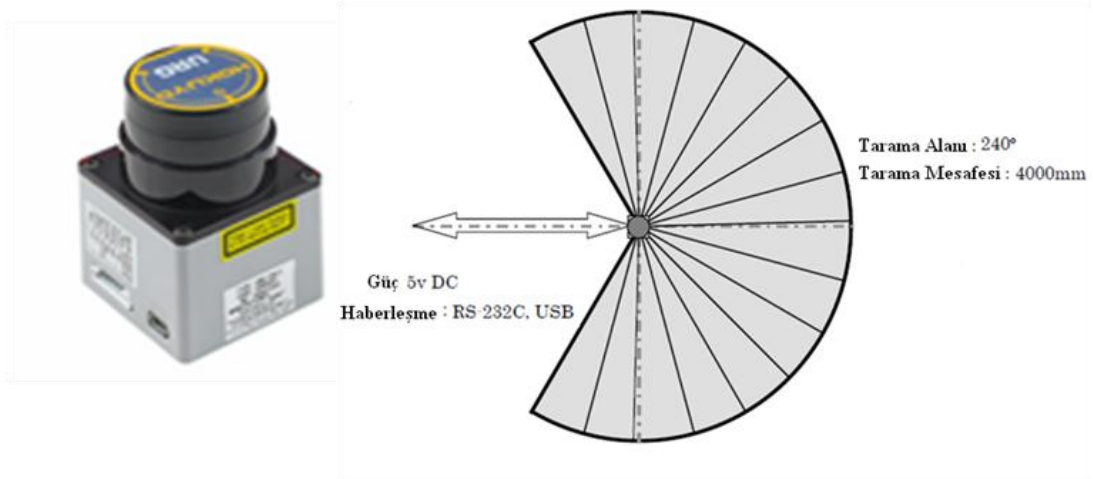
Bu çalışmada laboratuvar deney alanında tek robotlu bir harita oluşturma senaryosu gerçekleştirilmiştir. Bu senaryoda, robot konumlandırma için başlangıç noktası olarak kabul edilen bir giriş noktasından deney alanına girmekte ve sonrasında önceden belirlenmiş bir güzergâhı takip ederek kör konumlandırma ile lazer algılayıcı yardımıyla deney alanının haritasını çıkarmaktadır. Bu deneyin amacı kör konumlandırma sisteminde oluşan hataların haritalama sistemine nasıl bir etkisi olduğunun tespit edilmesidir. Bu amaçla robot konum ve lazer algılayıcı verilerini kablosuz ağ üzerinden bilgisayara aktarmaktadır. Bilgisayar üzerinde robottan alınan bu veriler MATLAB yazılım ortamında deney alanının iki boyutlu haritasının çıkarılması için kullanılmaktadır.

Uygulamada K-Team Firması tarafından üretilen KheperaIII mini gezgin robotu kullanılmaktadır. KheperaIII robotu alandaki engellerden kaçınmasını sağlamak amacı ile robotun üzerinde kısa mesafeler için dokuz adet kızıl ötesi ve uzun mesafeler için ise beş adet ses ötesi algılayıcı bulunmaktadır. Robot hareketleri iki adet kodlayıcı bağlı DC servo motor ile sağlanmaktadır. Her motor, ayrı bir PIC16F4431 mikrodenetleyicisinin kontrol ettiği kendi PID denetleyicisine sahiptir ve ayrıca bu mikrodenetleyiciler kör konumlandırma yöntemi ile konumun belirlenmesi için kodlayıcı ölçümlerini yapmak için kullanılmaktadırlar. Motor kontrol blokları I2C haberleşme hattında bağımlı elemanlar olarak çalışmaktadırlar.



Algılayıcılardan gerekli ölçümlerin alınması için ise 60 MHz’de çalışan bir adet DSPIC30F5011 mikrodnetleyicisi bulunmaktadır. Ana işlemci ile mikrodnetleyiciler arasındaki iletişim I2C haberleşme hattı üzerinden gerçekleşmektedir.

KheperaIII robotu merkezi işlemci birimi olarak 400 MHz hızında çalışan XScale PXA255 işlemcisini bünyesinde barındıran güçlü bir gömülü işletim sistemine sahip KoreBot LE kartı ile donatılmıştır. KoreBot LE gömülü işletim sistemi kartı, 64 Mbayt bellek sığası ve 32 Mbayt kalıcı belleğe sahiptir. Kart üzerinde Linux kernel 2.6 işletim sistemi çalışmaktadır. Kartın üzerinde ayrıca diğer bilgisayar veya robotlarla haberleşmeyi sağlayan LAN IEEE 802.11b kablosuz Ethernet kartı bulunmaktadır. KB-250 iletişim hattını kullanılarak USB kamera, kablosuz kamera ve diğer Korebot eklentileri robota kolayca bağlanabilmektedir.



Şekil 3.1. Hokuyo URG-04LX lazer mesafe ölçer.

Haritalama işleminde kullanılmak üzere, KheperaIII robotuna USB çoklayıcı bağlanarak (detaylı bilgi için bakınız Ek D.1.) robota Hokuyo URG-04LX (Şekil 3.1.) lazer mesafe algılayıcısının eklenmiştir. Lazer algılayıcının güç gereksinimiye daha önce bahsedilen ara haberleşme kartından sağlanmıştır. Algılayıcıdan USB haberleşme yöntemi ile Linux işletim sisteminde veri okunabilmesi için gerekli Linux sürücüsü ise C programlama ile geliştirilmiştir (sürücü hakkında daha detaylı bilgi Ek D.2.’de bulunabilir). Lazer algılayıcısı bağlı bir KheperaIII robotu Şekil

3.2.'de gösterilmektedir. Lazer algılayıcıdan kısaca bahsetmek gerekirse, algılama mesafesi 4 m, tarama açısı 0.35 derece aralıklarla 240 derece, hassasiyeti ise 1 mm ve doğruluğu ise  $\pm 10$  mm olarak verilmektedir.

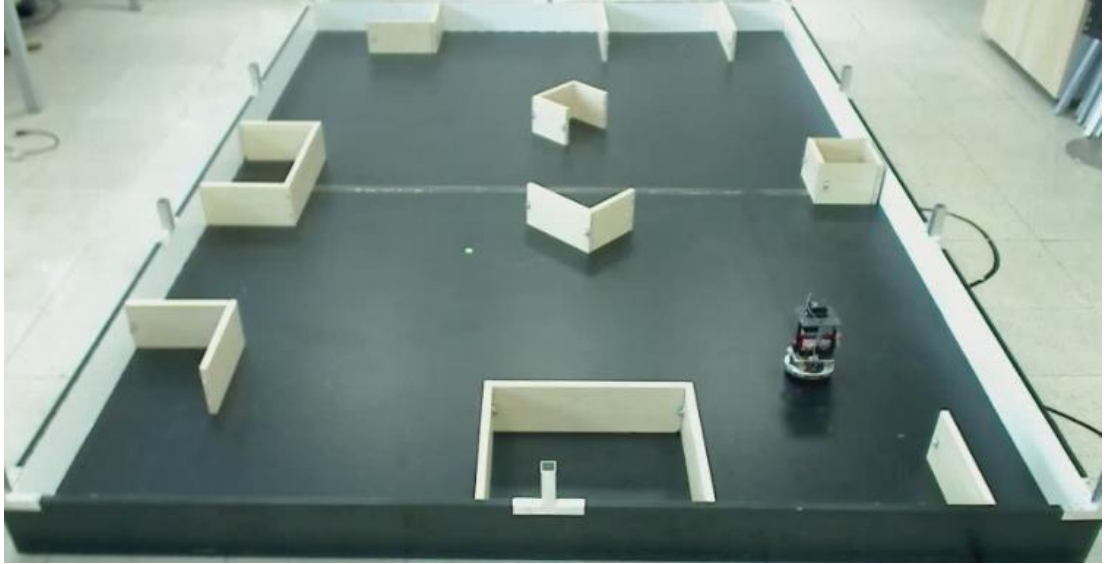
Robotun kendi pili, otonom hareket etme gibi işlemlerde 30 dakikaya yakın beslemeye yeterli olmakta iken Hokuyo lazer algılayıcı robota güç desteği olmadan bağlandığında bu süre 15 dakikaya kadar düşmektedir. Bu sebepten dolayı robotun çalışma sürelerini artırmak için robotun kendi piline ek olarak ara haberleşme kartı üzerinden robota ikincil bir lityum polimer pil bağlantısı yapılmıştır. Sonradan eklenen bu pil sayesinde lazer algılayıcı ile robotun çalışma süresi 45 dakikanın üzerine çıkmıştır. Böylelikle robot için harita oluşturma işleminde yeterli çalışma süresi elde edilmiştir.



Şekil 3.2. Lazer algılayıcı bağlı KheperaIII robotu.

Uygulama laboratuvar ortamında özel olarak hazırlanmış bir deney ortamında gerçekleştirilmiştir. Bu düzenek 240x340 cm olan bir robot alanından (Şekil 3.3.), lazer algılayıcı bağlı bir adet KheperaIII robotundan ve bir adet bilgisayardan oluşmaktadır. Robot lazerle tarama için kullanılırken bilgisayar alanın haritasının oluşturulması için kullanılmaktadır. Deney düzeneğinin gerçek bir alanı temsil etmesi için robot alanı içine 20 cm yüksekliğinde bazı duvarlar ve engeller yerleştirilmiştir.

Kullanıcı ara yüzü ve haritanın oluşturulduğu yer olarak bir masaüstü bilgisayar kullanılmaktadır. Bu masaüstü bilgisayar aynı zamanda yukarıda adı geçen kumanda merkezini temsil etmektedir. Robottan kablosuz ağ üzerinden alınan lazer algılayıcı ve konum verileri, bilgisayar üzerinde Matlab yazılımı ile görüntü verilerine dönüştürülerek alanın haritası oluşturulmaktadır. Kumanda merkezinde haritalama sistemi için yazılan bir C programı robotlardan kablosuz ağ üzerinden gelen verileri robota ait bir “.txt” dosyasına kaydetmektedir. Matlab programı ise bu “.txt” dosyasından robota ait konum ve lazer algılayıcı verilerini okumaktadır. Bu sayede robottan alınan veriler Matlab ortamında işlenmeye hazır hale gelmektedir. Matlab'ta ilk işlem olarak 600 x 600 'lük boş bir görüntü dosyası oluşturulmaktadır. Bu görüntünün (300, 500) noktası robot koordinat sisteminin başlangıç noktası kabul edilerek robotun çevrimiçi konumu görüntü üzerinde işaretlenmektedir. Görüntü üzerinde her bir pixel bir santimetreye karşılık gelecek şekilde tanımlanmaktadır. Robottan alınan lazer algılayıcı verileri ise, robotun etrafında algılanan her bir noktanın robota olan mesafe ve açı bilgilerinden oluştuğundan dolayı, bu veriler görüntü üzerindeki robot konumu referans alınarak işaretlenmektedir.



Şekil 3.3. Robot deney alanı.

Bu deney düzeneğinde robotun başlangıç noktası (0,0) konumu olarak kabul edilmiş ve haritalama işlemi için robotun alan içerisinde dolaşıp başlangıç noktasına geri

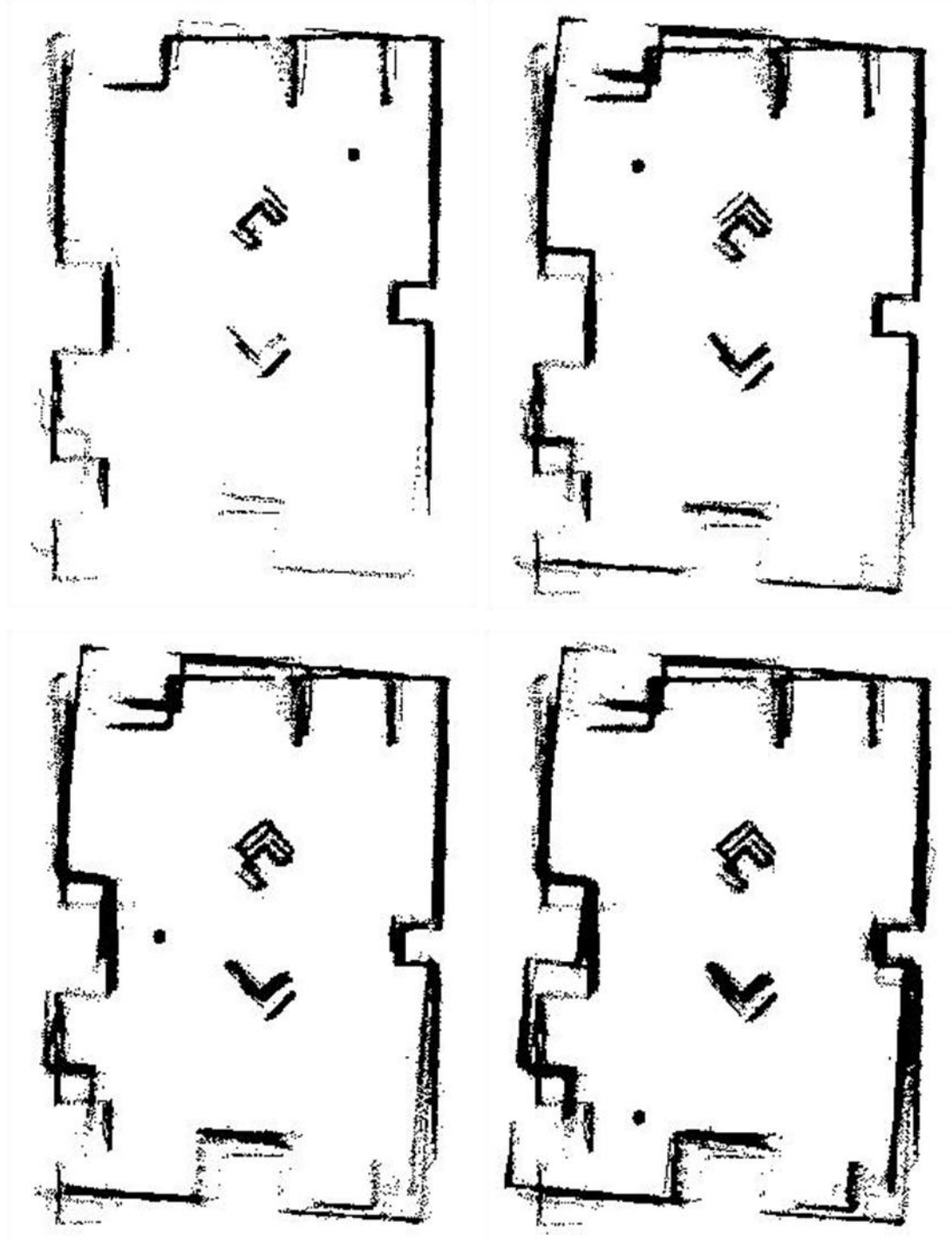
dönmesi programlanmıştır. Bu amaçla robot alan içinde köşe noktaları [(1.8,0), (1.8,-1.5), (0, -1.5), (0,0)] olan dikdörtgen bir güzergâhı takip etmektedir. Ayrıca dikdörtgenin köşe noktalarına ulaştığında  $180^\circ$  dönerek geldiği yolun farklı açıdan lazer algılayıcısı ile görüntüsünü almakta ve bir sonraki noktaya ilerlemektedir. Robot güzergâhı takip ederken yaklaşık 0.02 m/sn hızla hareket ederken yönelim değiştirme hareketlerinde ise 25 derece/sn açısal hız ile dönüş yapmaktadır. Robot bu hızlarla alanı yaklaşık olarak 7.5 dakikada dolaşmaktadır. Robot alanı dolaşırken 5 saniyede bir duraklamakta ve daha sonra konum ve lazer algılayıcı verilerini UDP (User Datagram Protocol) haberleşme yöntemiyle (detaylı bilgi için bakınız Ek E.) alanın haritasının çizilmesi için bilgisayara göndermektedir. Robot hareket halindeyken lazer algılayıcıdan alınan ölçümler robotun o anda bulunduğu konuma ait lazer verileri olmayabilmektedir. Bu amaçla robot durakladıktan sonra harita için kullanılacak lazer ölçümlerini yapıp bilgisayara göndermektedir.

Bilgisayara tarama yapan robot tarafından gönderilen veri paketleri robotun anlık konumu ve lazer algılayıcısından aldığı ölçümlerden oluşmaktadır. Burada bahsedilen anlık konum bilgileri robotun haritalamaya başladığı noktayı orijin olarak kabul ettiği yerel konumdan ve yönelim açısından, lazer algılayıcı ölçümleri ise 240 derecelik bir alan için 682 adet 0.36 derece çözünürlüklü ölçümlerden oluşmaktadır. Bu ölçümlerin çözünürlüğü ihtiyaca göre değiştirilebilmektedir.

Kablosuz ağ üzerinden bilgisayara gelen veriler MATLAB yazılım ortamında alanın 2 boyutlu haritasının çıkarılması için kullanılmaktadır. Oluşan harita üzerinde robotun alan içerisinde ilerlemesi ve alanın harita oluşumu adım adım görülebilmektedir (Şekil 3.4.). Burada lazer algılayıcı verilerinin tümü yani 4 metreye kadar olan ölçümlerin hepsi kullanıldığından dolayı haritalama tam görüş mesafesinde haritama olarak adlandırılmaktadır. Lazer verilerinin kısıtlandığı yani sadece belirli mesafeden küçük lazer ölçümlerinin kullanıldığı haritalama sistemleri ise kısa görüş mesafesinde haritalama olarak adlandırılmaktadır.

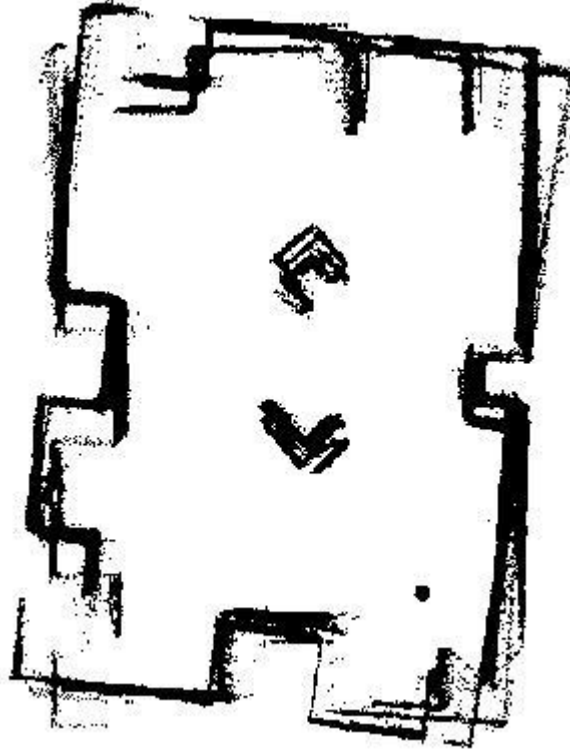
Şekil 3.4.'te robotun güzergâhı takibi esnasında tam görüş mesafesinde oluşturulan harita örnekleri gösterilmektedir. Burada alan içindeki daire robotun o anda alan içinde bulunduğu konumu ifade etmektedir. Bu haritalar üst soldaki haritadan alt

sağdaki haritaya doğru 2., 3.5., 4.5. ve 5.5. dakikalarda bilgisayar üzerinde zamanla oluşturulan haritalardır.



Şekil 3.4. Kör konumlandırma sistemi ile tek robotlu haritalamada tam görüş mesafesinde zamanla büyüyen haritalar.

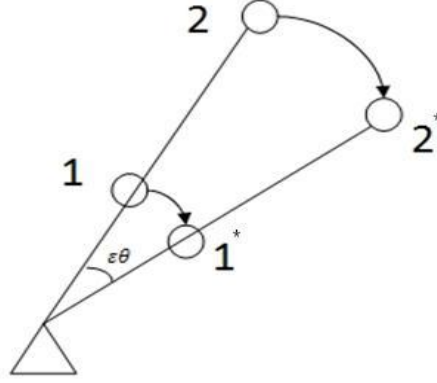
Şekil 3.4.'teki haritalar incelendiğinde kör konumlandırma sisteminde hataların zamanla artmaya başladığı harita üzerindeki şekillerin yerlerinin zamanla kaymasından anlaşılabilir. Haritanın gerçek alanla tutarlılığı zaman içinde önemli derecede azalmaktadır. Duvarlardaki ve nesneleredeki kalınlaşmanın yönelim açısında meydana gelen hatalardan başka nedeni ise lazer algılayıcısının  $\pm 10$  mm'lik bir hata payının bulunmasıdır. Şekil 3.5.'te ise güzergâh takibi tamamlandıktan sonra oluşan bir harita örneği gösterilmektedir.



Şekil 3.5. Kör konumlandırma sistemi ile tek robotlu haritalamada tam görüş mesafesinde oluşturulan son harita.

Şekil 3.5 incelendiğinde robotun konumlandırma sistemindeki hatalardan dolayı haritadaki bozulma açıkça görülmektedir. Alanın ortasında bulunan nesnelere ve duvarların zamanla harita üzerinde farklı yerler ve duruş açıları ile işaretlenmelerinin temel sebebi kör konumlandırma sisteminin yönelim açısında yaptığı hatadır. Yönelim açısındaki hataların haritaya etkisi haritada işaretlenen nesnenin robota olan uzaklığı ile doğru orantılı olarak artmaktadır. Lazer mesafe ölçümlerinin 0.36 derece

aralıklarla, yarıçapı 4 metre olan 240 derecelik bir daire parçası alanını kapsayacak biçimde alındığı düşünülürken robotun yönelim açısındaki küçük bir hata bile 4 metre uzaklıkta bulunan bir noktanın harita üzerinde çok farklı bir yerde işaretlenmesine sebep olmaktadır. Bu hatalar yakında olan noktalar için daha az olmaktadır çünkü yakındaki bir nokta harita üzerinde daha az yer değiştirecektir.

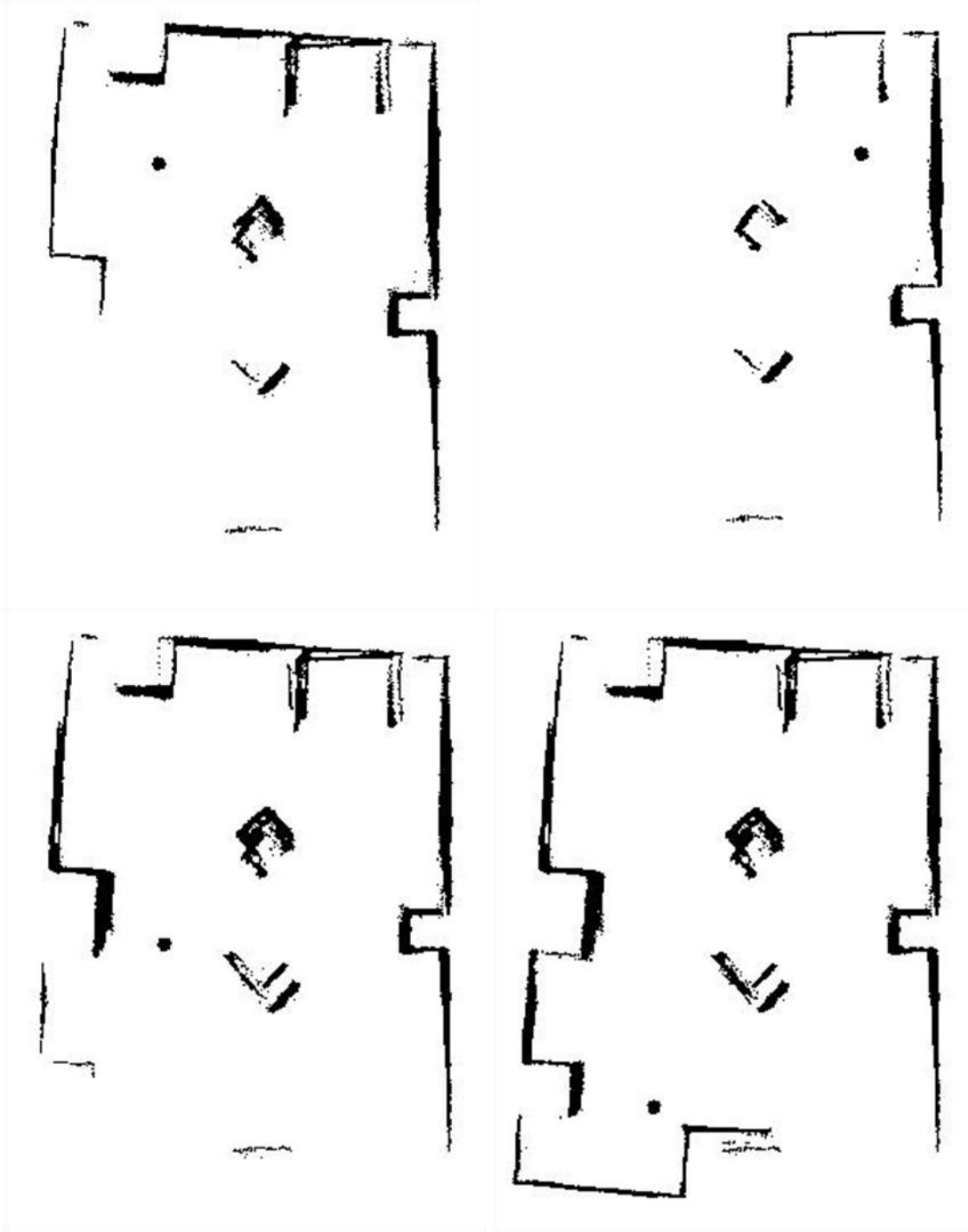


Şekil 3.6. Robot yönelim açısındaki hataların harita üzerine etkisi.

Şekil 3.6.'da üçgen şeklinin robot, doğru üzerindeki iki çemberin ise farklı iki nesne olduğu kabul edilirse robotun yönelim açısında meydana gelen  $\epsilon\theta$  kadar bir hata ile harita üzerinde şekilde görüldüğü gibi yakında olan bir nesne için uzakta olan bir nesneye göre daha az yer değiştirmekte ve dolayısıyla daha az hatalı olarak işaretlenmektedir.

Kör konumlandırma sistemi ile haritalamada robotun yönelim açısındaki hatanın harita üzerindeki etkisini azaltmak için önerilebilecek bir yöntem sadece belirlenen bir eşik değerinden küçük lazer ölçümlerinin harita oluşturulmada kullanılmasıdır.

Bu yöntem uygulandığında harita üzerinde daha az nokta işaretlenmesine rağmen sadece yakındaki noktalar işaretlendiğinden, yani işaretlenen noktaların konumlarında daha az hata olduğundan dolayı oluşan haritanın gerçek alan ile tutarlılığı da artmaktadır. Ayrıca robotun görüş mesafesinin azaltılması deney alanı ve robotun boyutu göz önüne alındığında deneylerin hangar gibi büyük bir alanın küçük bir temsili olduğu düşünülebilir. Görüş mesafesini azaltma yöntemi kullanılarak aynı deney lazer mesafesi eşik değeri 1 metre için tekrarlanmıştır. Bu durumda zamanla büyüyen harita Şekil 3.7.'de gösterilmektedir.

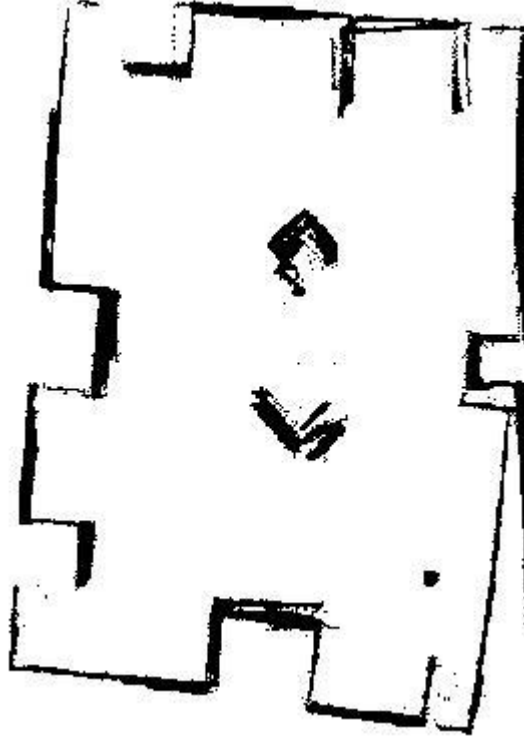


Şekil 3.7. Kör konumlandırma sistemi ile tek robotlu haritalamada kısa görüş mesafesinde zamanla büyüyen haritalar.

Şekil 3.7.'de robotun güzergâhı takibi esnasında kısa görüş mesafesinde oluşturulan harita örnekleri gösterilmektedir. Burada alan içindeki daire robotun o anda alan içinde bulunduğu konumu ifade etmektedir. Bu haritalar üst soldaki haritadan alt



sağdaki haritaya doğru 2., 3.5., 4.5. ve 5.5. dakikalarda bilgisayar üzerinde zamanla oluşturulan haritalardır. Haritalar incelendiğinde kör konumlandırma sisteminde hataların zamanla artmaya başladığı harita üzerindeki şekillerin yerlerinin zamanla kaymasından anlaşılabilir. Fakat bu kaymaların tam görüş mesafesinde yapılan haritalamaya göre nispeten daha az olduğu gözlemlenmektedir. Şekil 3.8.'de ise kısa görüş mesafesinde güzergâh takibi tamamlandıktan sonra oluşan bir harita örneği gösterilmektedir.



Şekil 3.8. Kör konumlandırma sistemi ile tek robotlu haritalamada kısa görüş mesafesinde oluşturulan son harita.

Şekil 3.8.'de görüldüğü gibi kör konumlandırma ile kısa görüş mesafesinde oluşturulan son haritada kaymalar devam etmektedir. Haritalama işlemi tamamlandıktan sonra alanın sağ duvarının harita üzerindeki başlangıç yöneline göre  $5.5^\circ$  döndüğü ölçülmüştür. Fakat tam görüş mesafesinde haritalamaya göre gerçek alan ile daha tutarlı bir harita oluşturulmuştur. Kısa görüş mesafesinde haritalamada harita üzerinde daha az nokta işaretlenmektedir fakat işaretlenen

noktaların hata oranı daha düşüktür. Bu sebepten dolayı kısa görüş mesafesinde haritalama ile tam görüş mesafesinde haritalamaya göre daha sade ve tutarlı haritalar oluşturulmaktadır.

### **3.2. IMU Destekli Kör Konumlandırma İle Tek-Robotlu Haritalama**

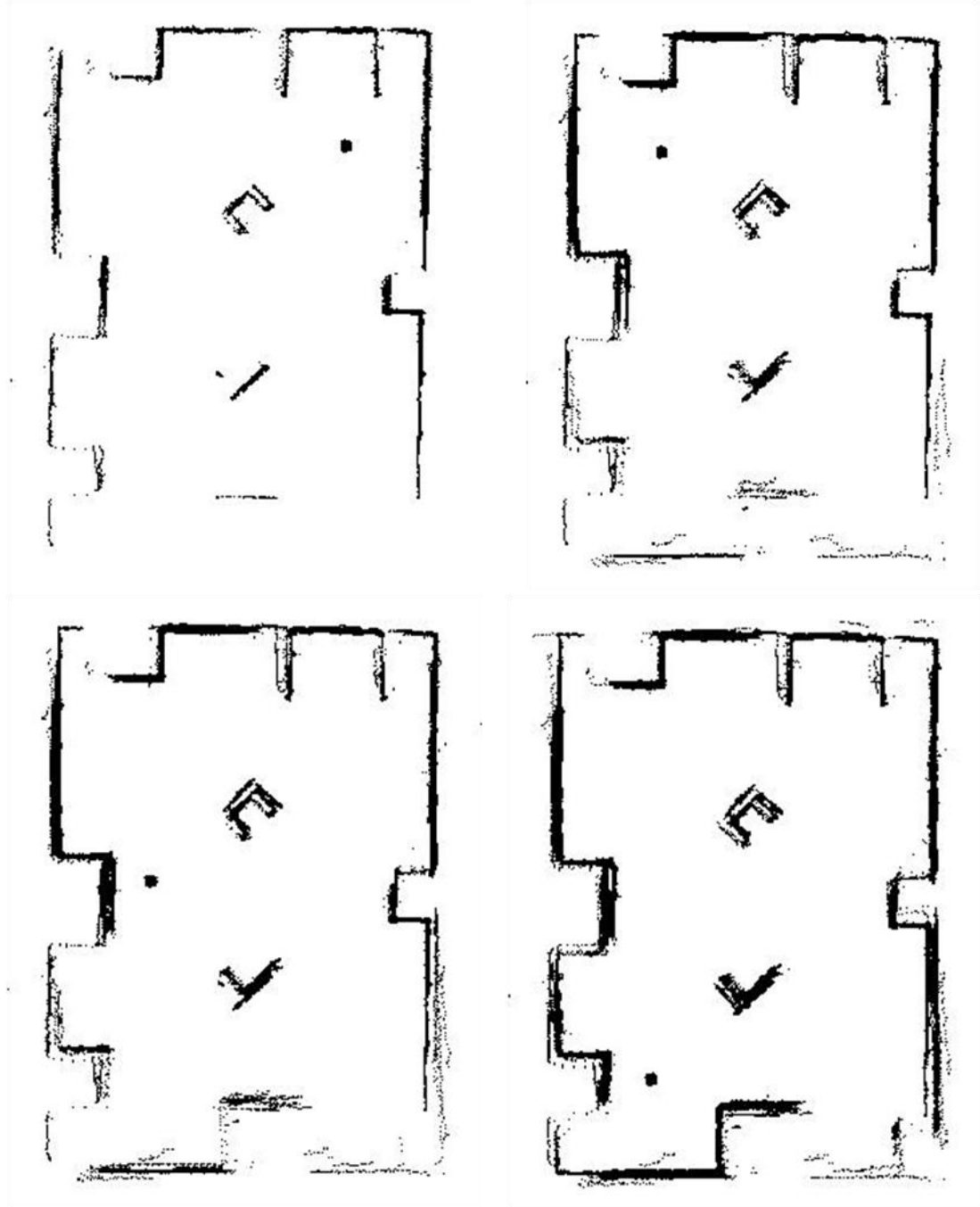
Kör konumlandırma işlemlerinde yönelim açısında oluşan hataların IMU desteği ile azaltılabileceği daha önceki bölümlerde gerekli deney ve çalışmalar yapılarak gösterilmişti. Bu bölümde ise kör konumlandırma ile haritalama işlemlerinde yönelim açısındaki hataların sebep olduğu kaymaların IMU destekli kör konumlandırma sistemi ile azaltılıp azaltılamayacağı üzerine bir çalışma yapılmıştır.

IMU destekli kör konumlandırma sisteminin haritalama üzerindeki etkisinin incelenmesi amacıyla Bölüm 3.1.'de yapılan kör konumlandırma ile tek-robotlu haritalama deneyleri aynı şartlar altında IMU destekli kör konumlandırma sistemiyle tekrarlanmıştır. Deney esnasında tam görüş mesafesinde çevrimiçi oluşan ve zamanla büyüyen haritalar Şekil 3.9.'da gösterilmektedir.

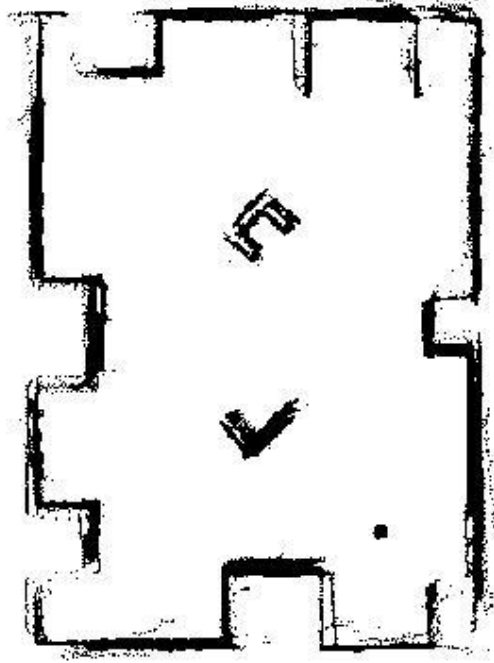
IMU destekli kör konumlandırma ile robotun yönelim açısındaki iyileşme sayesinde Şekil 3.9.'da görüldüğü gibi haritadaki kaymalar kör konumlandırma sistemi ile haritalamaya göre daha yavaş bir hızla büyümekte ve harita daha uzun süre geçerliliğini korumaktadır. Şekil 3.9.'daki haritalar tek tek incelendiğinde yönelim açısındaki iyileşmeden dolayı tam görüş mesafesinde uzakta işaretlenen nesnelere konumlarında daha az hatalar meydana gelmektedir.

Şekil 3.10.'da ise IMU destekli kör konumlandırma ile tam görüş mesafesinde oluşturan son harita gösterilmektedir. Şekil 3.10. incelendiğinde robotun konumlandırma sistemindeki iyileşmeyle haritadaki bozunmaların, duvar ve alan içindeki nesnelere yönelim ve konumlarındaki hataların da önemli ölçüde azaldığı yani haritanın gerçek alan ile daha tutarlı hale geldiği açıkça görülmektedir. Fakat haritalara dikkat edildiğinde kaymaların tam olarak giderilemediği duvarların kalınlaşmasından ve şekillerde az da olsa meydana gelen kaymalardan anlaşılabilir. Özellikle robota uzak olan noktalarda bu kaymalar daha çok görülmektedir. Bu çeşit kaymaların azaltılabilmesi için lazer verilerini lazer

verilerinin eşiklenmesi IMU destekli kör konumlandırma sistemiyle haritalamaya uygulanmıştır.

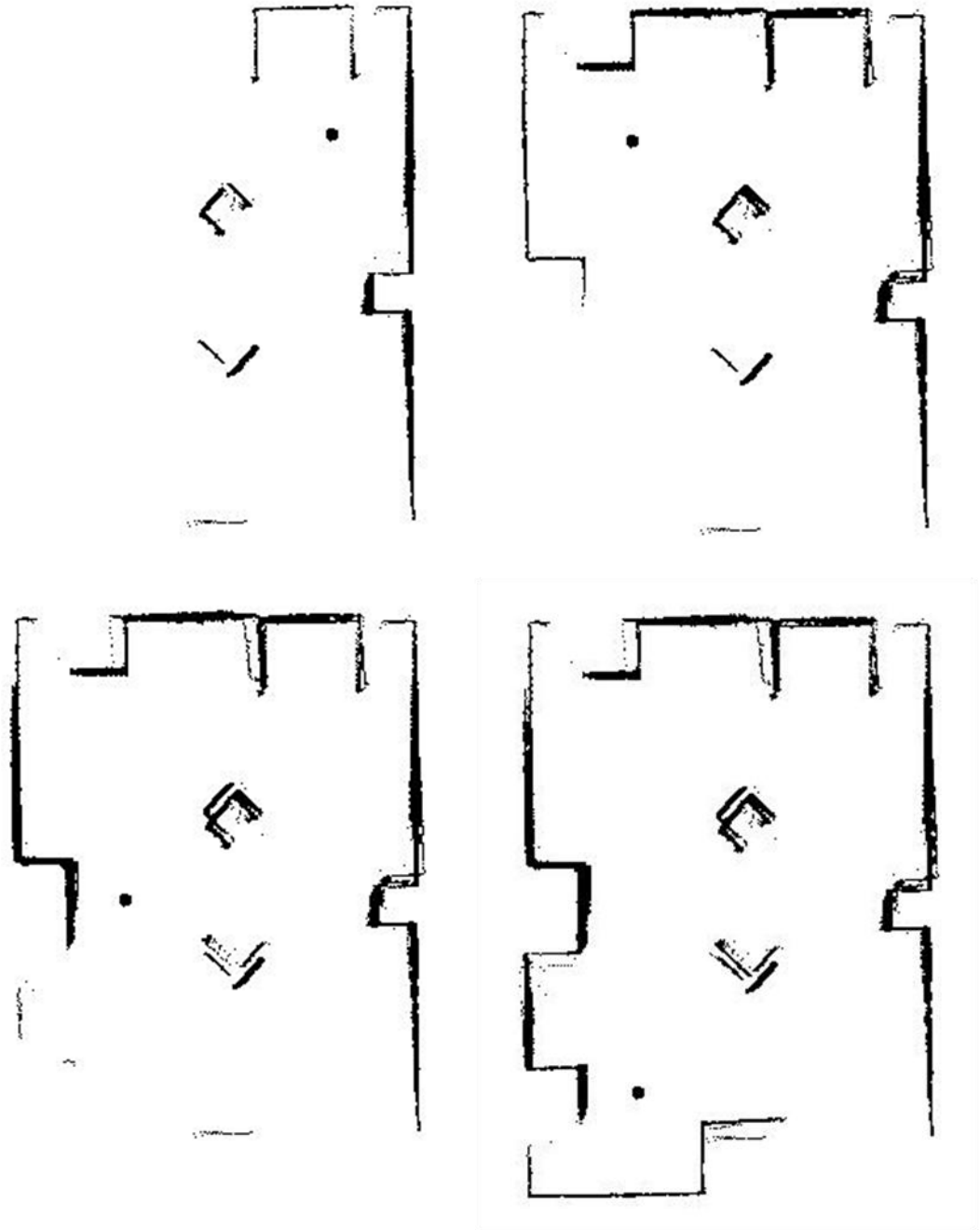


Şekil 3.9. IMU destekli kör konumlandırma sistemi ile tek robotlu haritalamada tam görüş mesafesinde zamanla büyüyen haritalar.



Şekil 3.10. IMU destekli kör konumlandırma sistemi ile tek robotlu haritalamada tam görüş mesafesinde oluşturulan son harita.

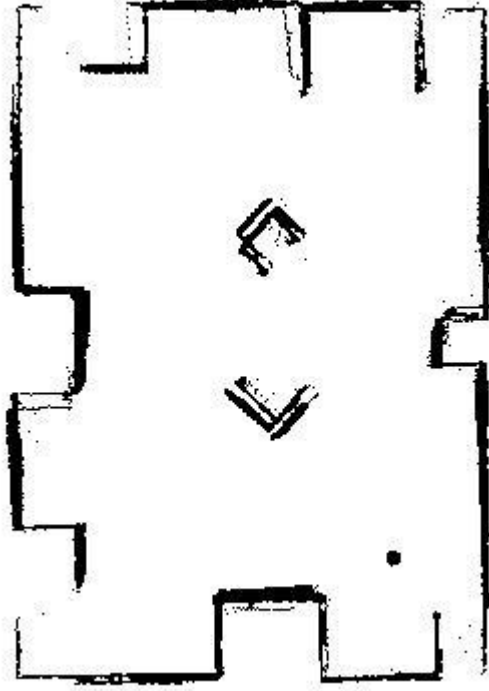
Şekil 3.11.'de IMU destekli kör konumlandırma ile haritalamada eşikleme yani kısa görüş mesafesinde haritalama ile oluşturulan ve zamanla büyüyen çevrimiçi harita örnekleri gösterilmektedir. Şekil 3.11.'de robotun güzergâhı takibi esnasında kısa görüş mesafesinde oluşturulan harita örnekleri gösterilmektedir. Bu haritalar üst soldaki haritadan alt sağdaki haritaya doğru 2., 3.5., 4.5. ve 5.5. dakikalarda bilgisayar üzerinde zamanla oluşturulan haritalardır. Haritalardan da anlaşılacağı üzere kısa görüş mesafesinde haritalardaki bozunmalar daha da yavaş artmaktadır. Alan içindeki nesnelerin ve duvarların kalınlaşması ve harita üzerinde duruş açılarının değişmesi tam görüş mesafesinde haritalamaya göre nispeten daha az olmaktadır. Şekil 3.12.'de ise kısa görüş mesafesinde güzergâh takibi tamamlandıktan sonra oluşan bir harita örneği gösterilmektedir.



Şekil 3.11. IMU destekli kör konumlandırma sistemi ile tek robotlu haritalamada kısa görüş mesafesinde zamanla büyüyen haritalar.

Şekil 3.12.'de görüldüğü gibi IMU destekli kör konumlandırma ile kısa görüş mesafesinde oluşturulan son haritada kaymalar oldukça azalmış ve gerçek alan ile oldukça tutarlı bir harita oluşturulmuştur. Özellikle IMU destekli kör konumlandırma sistemi ile kısa görüş mesafesinde haritalamada harita üzerinde daha az nokta

işaretlenmektedir fakat işaretlenen noktaların hata oranı daha düşüktür. Bu nedenle kısa görüş mesafesinde haritalama ile tam görüş mesafesinde haritalamaya göre daha sade ve tutarlı haritalar oluşturulmaktadır. Haritalama işlemi tamamlandıktan sonra alanın sağ duvarının harita üzerindeki başlangıç yönelimine göre  $0.5^\circ$  döndüğü ölçülmüştür. Bu dönme kör konumlandırma sistemi ile haritalamada  $5.5^\circ$  derece olarak ölçülmüştür. Bu ölçümlere dayanarak IMU destekli kör konumlandırma sistemi ile haritalamada kör konumlandırma sistemi ile haritalamaya göre % 90'a varan bir iyileşme olduğu görülmüştür. Robot konumlandırmasındaki iyileşme oranı ve dolayısıyla haritalamadaki iyileşme oranı robotun işlem sırasında yaptığı hareketlere ve işlem süresine bağlıdır. Bu nedenle burada dikkat edilmesi gereken bu iyileşme oranının her zaman garanti edilememesidir.



Şekil 3.12. IMU destekli kör konumlandırma sistemi ile tek robotlu haritalamada kısa görüş mesafesinde oluşturulan son harita.

## BÖLÜM 4

### 4. YARDIMLAŞMALI ÇOK-ROBOTLU HARİTALAMA

Bu bölümde laboratuvar deney alanında çoklu robotlar ile iki farklı yardımlaşmalı harita oluşturma yöntemi karşılaştırılmıştır. İncelenen durumda, bütün robotlar sıra ile konumlandırma için başlangıç noktası olarak kabul edilen ortak bir giriş noktasından deney alanına girmektedirler. Robotlardan ikisi, bir robot alanın sağında kalan ve diğer bir robot ise alanın solunda kalan duvarları takip edecek şekilde dışsal alanların haritasını çıkarmak için iş bölümü yapmaktadırlar. Aynı anda diğer bir robot ise içsel bölgenin haritasını çıkarmak için alanın içinde rastgele dolaşma görevini üstlenmektedirler. Rastgele dolaşma görevini üstlenen robot iki farklı hareket yöntemi kullanarak diğer duvar takibi yapan robotların geçtiği yerlerden geçmemektedir. Bu yöntemlerden birincisi boşluklara yönelerek seyrüsefer diğeri ise yapay potansiyel fonksiyonlar kullanarak diğer robotların geçtiği güzergâhlardan itilme biçiminde seyrüseferdir. Boşluklara yönelerek seyrüsefer ile rastgele dolaşma görevini üstlenen robot duvar takibi yapan robotların geçtikleri bir yere geldiğinde durup lazer tarayıcısını kullanarak etrafındaki en uzun nokta ne yönde ise o tarafa doğru yönelimini değiştirerek hareketine devam etmektedir. Bu sayede diğer robotların geçtikleri yerlerden geçmemektedir. Yapay potansiyel fonksiyonlar kullanarak hareket yönteminde ise rastgele dolaşma görevini üstlenen robot duvar takibi yapan robotların geçtiği yerlere geldiğinde geçilen noktalara yapay potansiyel kuvvetler (yapay potansiyel fonksiyonlar hakkında daha geniş bilgi Bölüm 4.1.'de verilecektir) atayarak bu noktalardan kaçınmaktadır. Robotlar konumlarını ve lazer algılayıcı verilerini kaydederlerken aynı zamanda bu bilgileri kendi aralarında paylaşmaktalar ve kablosuz ağ üzerinden kumanda merkezine bu verileri aktarmaktadırlar. Kumanda merkezinde robotlardan alınan bu veriler deney alanının iki boyutlu haritasının çıkarılması için kullanılmaktadır. Robotların birbirleri ile çarpışmalarını engellemek ve hareketlerini sağlamak amacı ile uygulamada yapay potansiyel fonksiyonlardan faydalanılmıştır.

IMU destekli kör konumlandırma sistemi daha önceki bölümlerde bahsedildiği üzere tek robotlu haritalama sistemlerinde oldukça kullanışlı bir konumlandırma iyileştirme yöntemi olmakla birlikte robot için fazladan işlem yükü getirmektedir. Tek robotlu haritalamada sadece bilgisayara tek yönlü veri gönderildiğinden dolayı KheperaIII robotunun işlem gücü yeterli olabilmekte iken çok robotlu haritalama sistemlerinde yeterli olamamaktadır. Çünkü çok robotlu haritalamada robot sadece kontrol merkezi ile değil aynı zamanda bilgisayarla haberleşmeye ek olarak alanda bulunan tüm robotlarla iki yönlü haberleşmekte yani verileri hem alıp hem de göndermektedir. IMU algılayıcısından veri okunma sürecine ek olarak diğer robotlardan gelen verilerin işleme süreci de eklenmekte ve IMU destekli konumlandırma için örnekleme zamanı da artmaktadır. Bu sebepten dolayı robotun konumu verimli bir şekilde takip edilememektedir. Yapılan deneyler sonucunda KheperaIII robotlarının işlem gücünün yetersiz olmasından dolayı kör konumlandırma sisteminin IMU destekli kör konumlandırma sistemine göre çok robotlu haritalama sistemlerinde daha iyi sonuç verdiği anlaşılmıştır. Bu sebepten dolayı çok robotlu haritalamada robotların konum takipleri için sadece kör konumlandırma sistemi kullanılmıştır.

#### 4.1. Robot Modeli ve Denetimi

Yardımlaşmalı çok-robotlu haritalama yöntemlerinde iki boyutta hareket eden ve sürekli zaman hız kısıtlı hareket dinamikleri aşağıda verilen  $N$  adet gezgin robot üzerinde çalışılmıştır.

$$\begin{aligned}
 \dot{x}_i(t) &= \bar{v}_i(t) \cos(\theta_i(t)) \\
 \dot{y}_i(t) &= \bar{v}_i(t) \sin(\theta_i(t)) \\
 \dot{\theta}_i(t) &= w_i(t)
 \end{aligned} \tag{4.1}$$

Burada  $x_i(t)$  ve  $y_i(t)$ ,  $i$ . robotun  $t$  zamanındaki dikgen koordinatlarda konumunu ve  $\theta_i(t)$  ise yönelim açısını ifade etmektedirler.  $i$ . robotun kontrol girdileri doğrusal hız  $\bar{v}_i(t)$  ve açısal hız  $w_i(t)$ 'dir.



Bu uygulamalarda robotlar, içinde engeller ve duvarlar bulunan bilinmeyen kapalı bir alanın haritasını çıkarmaktadırlar. Kapalı alanın sınırlarını ve farklı mesafelerdeki nesnelerin uzaklıklarını tespit etmek için robotlar lazer uzaklık ölçerleriyle donatılmışlardır. Bunun yanı sıra, robotlar kendi konumlarını kör konumlandırma ile hesaplayabilmektedirler. Lazer uzaklık ölçerden alınan ölçümler ile konum bilgileri birleştirilerek ortamın haritasının oluşturulmasında kullanılmaktadırlar. Bilinmeyen bir alan hakkında robotun sahip olduğu bilgi, ölçüm yapılan uzaklıklar ve bu uzaklıkların robota göre ait oldukları bağıl açıların bulunduğu veri dizisinden oluşmaktadır. Etrafta engellerden kaçınarak dolaşma amacı ile bu dizi altı kümeye ayrılmıştır (Şekil 4.1.) ve her küme birden fazla mesafe ölçümünü barındırmaktadır.  $d_{pn}^i(t) = [d_{xpn}^i(t), d_{ypn}^i(t)]$  denkleminde,  $d_{pn}^i(t)$ 'nin  $t$  zamanında  $i$ . robotun  $p$ . kümesinin  $n$ . ölçümünün uzaklık vektörü olduğunu kabul edilmektedir. Burada  $d_{xpn}^i(t)$ ,  $d_{xpn}^i(t) = \|d_{pn}^i(t)\| \cos(\theta_{pn}^i(t))$  denkleminde  $\|d_{pn}^i(t)\|$ 'nin  $t$  zamanında her bir ölçüme ait mesafe ve  $\theta_{pn}^i(t)$  'nin ise açı bilgisini temsil ettiği denklem ile verilen, mesafe vektörünün x bileşenini ifade etmektedir. Aynı şekilde  $d_{ypn}^i(t)$ ,  $d_{ypn}^i(t) = \|d_{pn}^i(t)\| \sin(\theta_{pn}^i(t))$  denklemi ile verilen, mesafe vektörünün y bileşenini ifade etmektedir. Ek bilgi olarak, lazer mesafe ölçerden hem mesafe bilgisi  $\|d_{pn}^i(t)\|$  hem de açı bilgisi  $\theta_{pn}^i(t)$  çok küçük hatalarla elde edilebilmektedir. Alan içinde engellere çarpmadan dolaşma amacı ile yapay potansiyel kuvvetlerinin oluşturulmasında sadece kümeler içindeki en kısa mesafeler kullanılmaktadır. Bu işlem, sistemin verimini istenilen seviyede korurken, işlem yükünü büyük oranda azaltmaktadır.

Robotlar ortamdaki diğer robotlarla ve kumanda merkezi ile haberleşmektedirler ve lazer mesafe ölçümlerini ve konumlarını kurulan ağ üzerinde paylaşmaktadırlar. Böylece bilinmeyen bir ortamın haritasının oluşturulma işlemi yardımlaşmalı olarak gerçekleştirilmektedir. Robotlardan biri alanın sol tarafındaki diğeryse sağ tarafındaki duvarı takip ederek bilinmeyen bir alanın dışsal haritasını oluştururlarken, sistemdeki diğer robotlar ise bunların arasındaki alanda engellere çarpmadan rastgele dolaşarak bilinmeyen alanın içsel haritasını yardımlaşmalı olarak oluşturmaktadırlar. Yukarıda bahsedilen robotların serbest dolaşma işlemsel süreçlerini gerçekleştirmek için, alt seviye kontrollerinin işlevsel süreçlerinde yapay potansiyel fonksiyonlar

kullanılmaktadır. Potansiyel fonksiyonlar sürü sistemlerin özellikleri olan, bir araya toplanma, beslenme ve dizilim oluşturma denetimi gibi birçok sistemde başarı ile uygulanmaktadır [39][40]. Özellikle duvar takibi için, duvara olan uzaklığın istenilen uzaklıktan fazla olması durumunda yapay bir çekim kuvveti, az olması durumunda ise yapay itim kuvveti oluşturan bir potansiyel alan kullanılmaktadır. Robotlar da bu potansiyel alanın negatif eğimi yönünde hareket etmektedirler. Duvara olan mesafe istenilen mesafeye eşit olduğunda itim ve çekim kuvvetleri eşitlenir ve robotun üzerine etkiyen yapay kuvvet sıfıra eşit olur. Böylece robot doğrusal hızını  $\bar{v}_i(t)$  ve açısal hızını  $w_i(t)$  değiştirmeksizin hareketine devam eder. İşlem süreçlerinde kullanılan potansiyel fonksiyonun grafiği Şekil 4.2.'de verilmiştir. Şekilden de anlaşılacağı üzere uygulama esnasında istenilen mesafe 20 cm olarak ayarlanmıştır. Duvar takibi işleminde, soldan duvar takibi için en soldaki, sağdan duvar takibi için ise en sağdaki veri kümesi kullanılmaktadır. Diğer veri kümeleri ise aynı anda etraftaki engellerden kaçınmak için kullanılmışlardır. Bunun yanı sıra, etraftaki nesnelere çarpmadan alan içinde rastgele dolaşma işleminde ise potansiyel fonksiyonun sadece itim kuvveti oluşturma kısmı kullanılmıştır. Yani ölçülen mesafe daha önceden belirlenen bir mesafeden daha küçük olduğunda, robotun üzerinde yapay bir itim kuvveti oluşturulmakta ve robot bu sayede doğrusal ve açısal hızını değiştirerek etrafındaki engellerden kaçabilmektedir.  $t$  zamanında  $i$ . robotun özel bir  $p$ . veri kümesi için potansiyel fonksiyonun eğimini  $G_{pn}^i(t)$  olarak kabul edilmektedir. Bu çalışmada bu değer aşağıdaki gibi seçilmiştir.

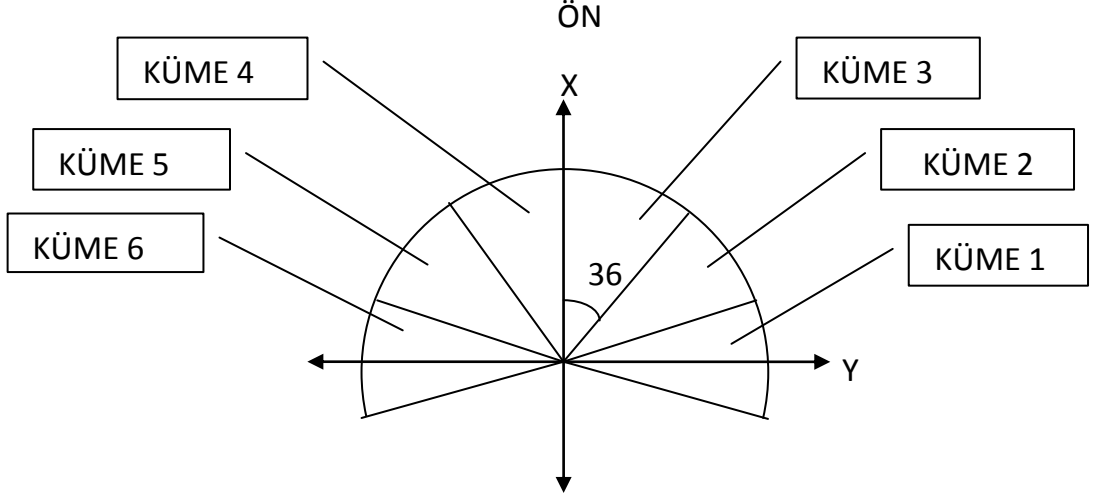
$$G_p^i(t) = d_p^i(t) \left( \frac{1}{\|d_p^i(t)\|^2} - \frac{k}{\|d_p^i(t)\|^3} \right) \quad (4.2)$$

Burada  $k$ , duvar takip ve engellerden kaçma mesafesini ifade etmektedir ve bu deneyler sırasında duvar takibi için bu mesafe  $k=0.2$  olarak seçilmiştir,  $\|d_p^i(t)\| = \min_{j=1,2,\dots,n} \{\|d_{pj}^i(t)\|\}$  ve  $d_p^i(t)$  ise sadece ilgili ölçüme ait olan vektördür. Diğer bir deyişle, yukarıda bahsedildiği gibi, veri kümelerinden potansiyel kuvvetlerin hesaplanmasında sadece o veri kümesindeki en küçük ölçüm için işlem yapılmaktadır.

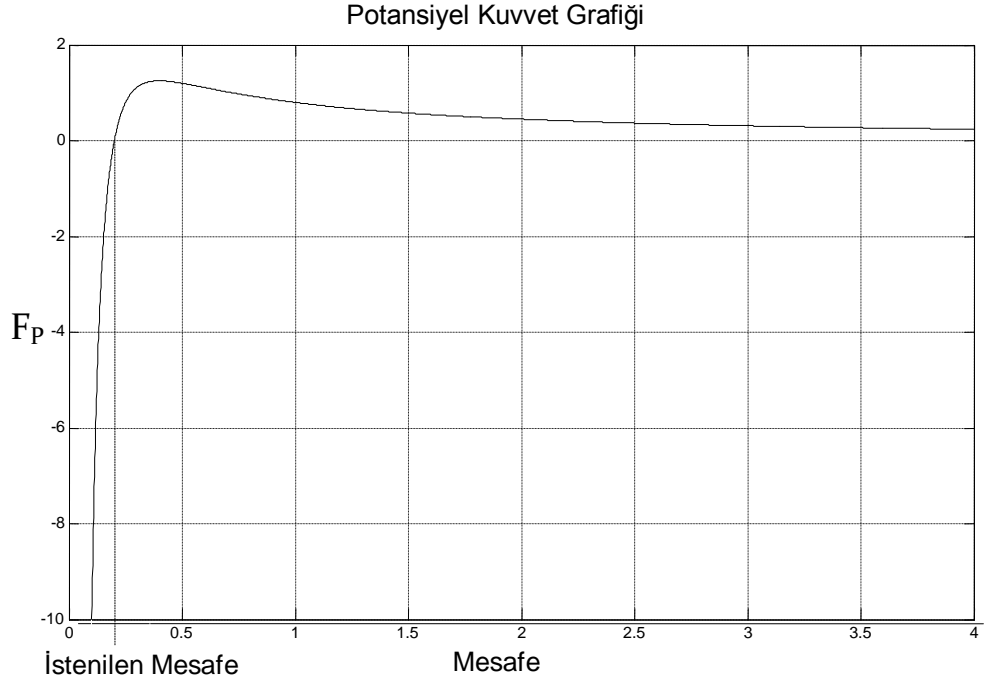
$t$  zamanında  $i$ . robot için yukarıda verilen toplam potansiyel alan  $G^i(t)$  denklemi aşağıdaki gibi belirlenmiştir.

$$G^i(t) = \sum_{j=1}^L G_j^i(t) \quad (4.3)$$

Burada  $L$ , veri dizilerindeki küme sayısını ifade etmektedir ve bu deneyler sırasında  $L=6$  olarak seçilmiştir. Ayrıca robotun üzerine etki eden toplam vektörel kuvvetin hesaplanması için yerel koordinat sistemi kullanılmaktadır.



Şekil 4.1. Lazer verileri küme şeması.



Şekil 4.2. Potansiyel kuvvet grafiği.

Uygulamada kullanılan robotlar hız – kısıtlı olduklarından yani sadece iki tekerleğini kesen doğruya dik yönde hareket edebildiklerinden dolayı ve yönelimleri başlangıçta potansiyel vektör yönünde olmayabileceğinden, dönülmek istenen yönelim açısı aşağıdaki gibi tanımlanmaktadır.

$$\theta_{id}(t) = \text{atan2}(G_{yi}(t), G_{xi}(t)) \quad (4.4)$$

Burada  $G_{xi}(t)$  ve  $G_{yi}(t)$ , Denklem 4.3'te verilen potansiyel alanın  $x$  ve  $y$  bileşenlerini ifade etmektedir. Böylece robot  $i$ 'nin yönelim dinamikleri basit bir oransal denetleyici kullanılarak aşağıdaki denklem ile denetlenmektedir.

$$w_i(t) = -\alpha(\theta_i(t) - \theta_{id}(t)) \quad (4.5)$$

Burada  $\theta_i(t)$ ,  $t$  zamanında robotun güncel yönelimini ve  $\alpha$  ise denetleyicinin oransal kazancını ifade etmektedir. Doğrusal hız denetimi aşağıdaki gibi tanımlanmaktadır.

$$\bar{v}_i(t) = \min\{v_{nom} + \|G_i(t)\|, v_{max}\} \quad (4.6)$$

Burada  $v_{nom}$  itibari hızı ve  $v_{max}$  ise hızı üzerinde tavan sınırını ifade etmektedir. Robotun üzerine etki eden sanal potansiyel kuvvetin büyüklüğü sıfıra eşit olduğunda robot, itibari hız  $v_{nom}$  ile bulunduğu yönelimde hareketine devam etmektedir.



Şekil 4.3. Deney uygulama alanı.

Uygulamalar laboratuvar ortamında özel olarak hazırlanmış bir deney ortamında gerçekleştirilmiştir. Bu düzenek daha bölümde de anlatıldığı gibi boyutları 240x340 cm olan bir robot alanından (Şekil 4.3), üç adet KheperaIII robotundan ve bir adet bilgisayardan oluşmaktadır. Üç adet robot lazerle tarama için kullanılırken bilgisayar ise kumanda merkezi olarak birleşik haritanın oluşturulması için kullanılmaktadır. Daha önce de belirtildiği gibi deney düzeneğinin gerçek bir alanı temsil etmesi için robot alanı içine 20 santimetre yüksekliğinde bazı duvarlar ve engeller yerleştirilmiştir. Önceki bölümde de anlatıldığı gibi bilgisayar üzerinde harita oluşturulmaktadır. Bu amaçla çok robotlu haritalamada her robota özel konum ve lazer verilerinin kaydedildiği ayrı bir “.txt” dosyası oluşturulmaktadır. Bu dosyalar sayesinde her robotun verileri haritalama için işlenebilecek hale gelmektedir. Robotların konum ve özellikle yönelim açılarındaki hatalar nedeniyle robotlara uzak mesafelerdeki noktaların işaretlenmesinde zamanla büyük hatalar oluşmaktadır. Bu nedenle işaretleme esnasında 100 cm’den büyük mesafe bilgileri görüntü üzerinde işaretlenmemektedir ve böylece haritadaki hata oranı azaltılmaktadır. Fakat haritanın tamamlanma süresi artmaktadır. Bu sayede deney alanının çevrimiçi harita çıkarma işlemi, deney esnasında ortalama 2 cm/sn hızla hareket eden KheperaIII mini gezgin robotları ile yaklaşık olarak 3 dakikada tamamlanabilmektedir.

Ek olarak, harita çıkarma işleminde kullanılan robotlar ile haritanın oluşturulduğu bilgisayar arasındaki mesafe uzak olduğunda iletişim ağı zayıf olabilir. Bu gibi durumlarda bir başka robot, bilgisayar ile robotlar arasında iletişim köprüsü olacak şekilde yerleştirilebilir. Çok büyük boyutlarda bilinmeyen bir alanın haritasının çıkarılmasının gerekli olduğu durumlarda bu şekilde bir iletişim köprüsü kurularak iletişim ağı güçlendirilebilmektedir.

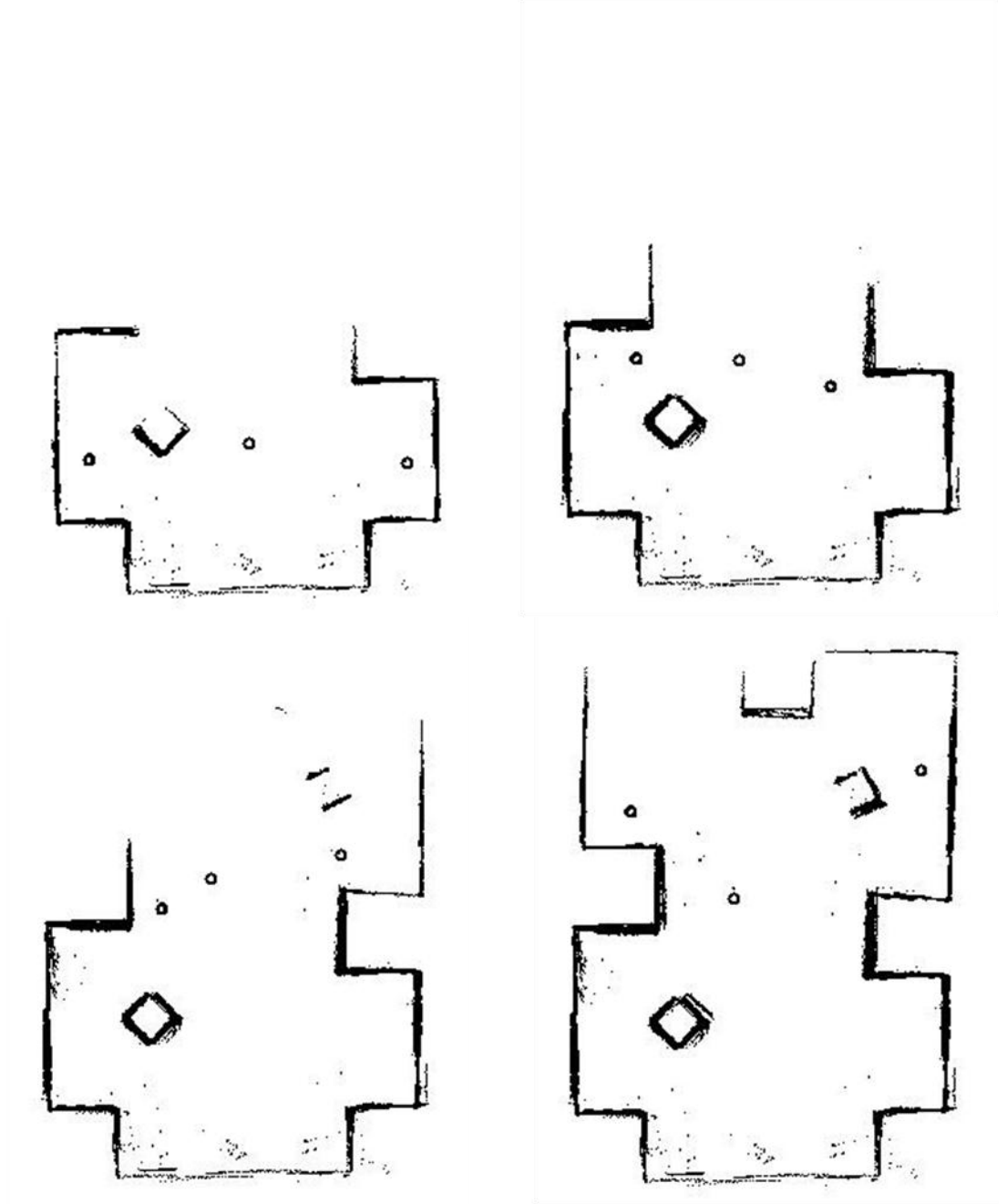
#### **4.2. Boşluklara Yönelerek Yardımlaşmalı Haritalama**

Bu uygulamada robotlar temel olarak üç farklı otonom davranışa sahiptirler. Bu otonom davranışlardan ilki robotların ortamlardaki engellerden kaçarak rastgele etrafta dolaşmaları, ikinci alanın sol tarafında kalan duvarların takip edilmesi ve üçüncüsü de alanın sağ tarafında kalan duvarların takip edilmesidir. Haritalama

işlemi başlamadan önce soldaki robot sol duvarı ve sağdaki robot ise sağ duvarı takip edecek şekilde haritalama işlemine başlamaktadırlar. Bu işlem tamamlandıktan sonra bu iki robotun arasında kalan robotlar ise yukarıda anlatılan rastgele dolaşma otonom davranışını yapmaktadırlar. Bu uygulamada harita çıkarma işleminde üç adet robot kullanıldığından iki robot duvar takibi yaparken geriye kalan robot ise rastgele alanın iç kısımlarında dolaşmaktadır. Bu sayede duvar takibi yapan robotlar bilinmeyen alanın dışsal haritasının çıkarılması için lazer algılayıcı ölçümlerini ve kendi konumlarını bilgisayara gönderirken, iç kısımlarda rastgele dolaşan robot ise alanın içsel haritasının oluşturulması için lazer algılayıcı ölçümlerini ve kendi konum bilgilerini bilgisayara göndermektedir. Büyük ölçekli alanların haritalarının çıkarılması işlemlerinde ise bu ölçümler doğrudan bilgisayara gönderilmeyip daha önceden köprü olarak belirlenen robot üzerinden gönderilebilecektir. Bilgisayara gönderilen veriler tek bir harita üzerinde birleştirilerek alanın çevrimiçi haritasının çıkarılması için kullanılmaktadırlar.

Bilgisayar üzerinde harita oluşturma işlemi yapılırken aynı zamanda duvar takibi yapan robotların geçtikleri noktalar ayrı bir “.txt” dosyasına kaydedilerek rastgele dolaşma işlemi yapan robotun bu noktalar üzerinde olup olmadığı kontrol edilmektedir. Rastgele dolaşma işlemi yapan robot bu noktalardan herhangi birine 20 cm yaklaştığında bu robota daha önceden taranmış bir alanda olduğu bilgisayar tarafından bildirilmektedir. Burada dikkat edilmesi gereken robotlar birbirleri arasında konumlarını paylaşarak bu bilgileri bir nesne ile karşılaştığında karşısındakinin robot olup olmadığına karar vermek için kullanılmaktadırlar. Rastgele dolaşma işlemi yapan robotun duvar takibi yapan robotların geçtikleri yerlerden geçmemesi işlemi ise robotlarda diğer işlemlerin yavaşlamasına sebep olduğundan dolayı bilgisayar üzerinde yapılarak sonuç rastgele dolaşma işlemi yapan robota gönderilmektedir. Daha önce taranmış bir alanda olduğu bilgisini alan robot ise o noktada durup lazer tarayıcısı ile bir ölçüm yapmaktadır ve sonrasında ise bu ölçümlerden etrafındaki en uzak noktayı bulmaktadır. Bir sonraki adım olarak robot yönelimini bu noktaya doğru değiştirip rastgele dolaşma işlemine bu yönde devam etmektedir. Böylelikle rastgele dolaşma işlemi yapan robot duvar takibi yapan robotların geçtikleri yerlerden geçmediğinden daha verimli ve çabuk bir harita

çıkarma işlemi gerçekleştirilmektedir. Bu yöntemle çıkarılan ve zamanla büyüyen harita örnekleri Şekil 4.4.'te gösterilmektedir.



Şekil 4.4. Boşluklara yönelerek yardımlaşmalı zamanla büyüyen haritalar.

Şekil 4.4.'te alan içindeki daireler robotların o anda alan içinde buldukları konumları ifade etmektedir. Bu haritalar üst soldaki haritadan alt sağdaki haritaya

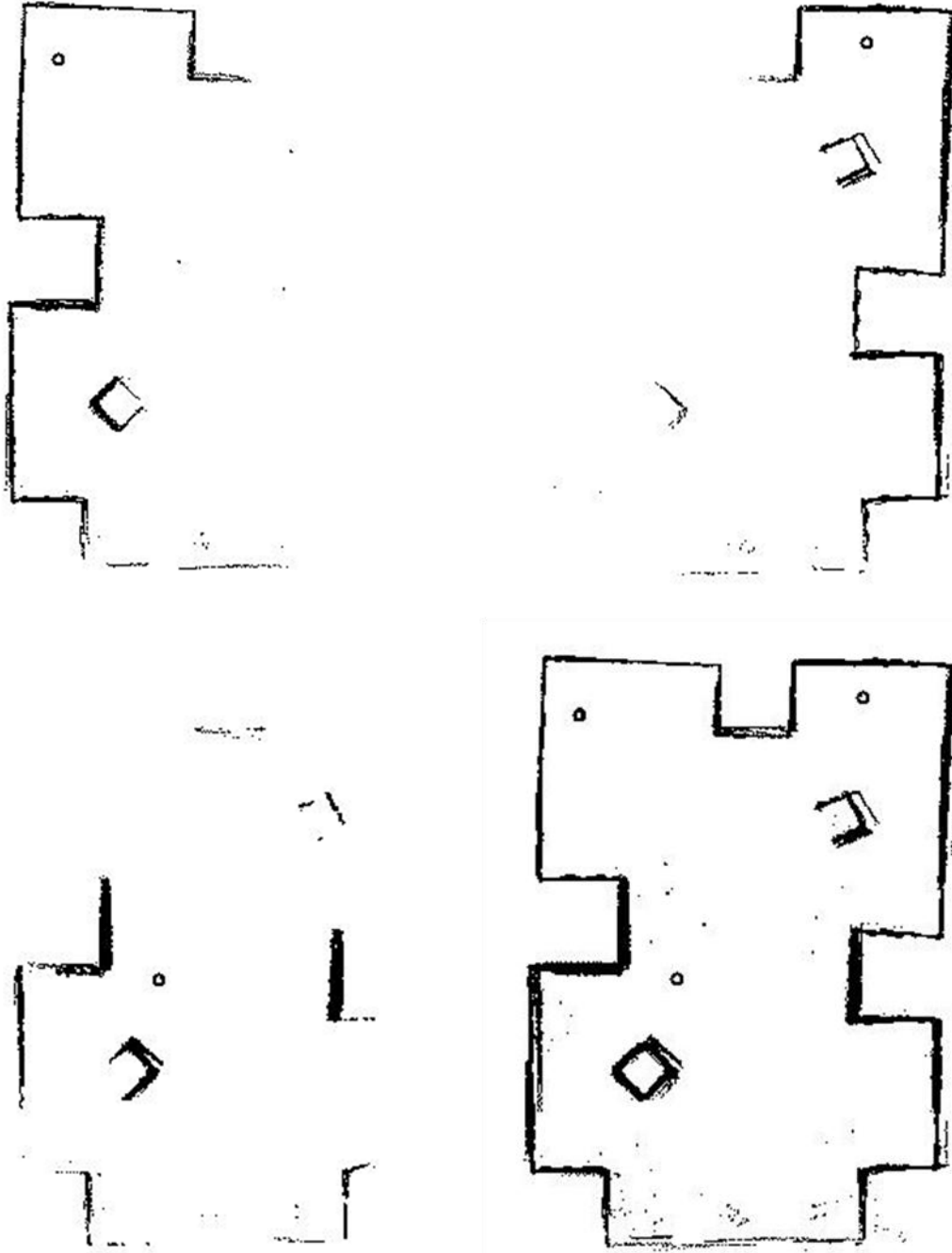
dođru 40., 80., 120. ve 160. saniyelerde bilgisayar üzerinde zamanla oluşturulan haritalardır. Haritalardan da anlaşılacağı üzere bilgisayar üzerinde oluşturulan harita zamanla büyüyerek alan hakkında daha fazla bilgi içermeye başlamaktadır. Haritalar incelendiğinde alanın duvarları ve alan içindeki nesnelere zamanla belirginleşerek tamamlanmaktadır.

Lazer mesafe algılayıcı tek bir tarama ile robota en yakın duvar veya engelin robota olan uzaklığını algıladığından tarama alanı dışında kalan veya belirlenen engellerin arkasında bulunan diğer nesnelere ölçüm alınmamaktadır. Bu nedenle bilinmeyen bir alanın haritasının çıkarılması işlemi tek bir robot ile çok fazla zaman alabilmektedir ve çıkarılan haritanın tutarlılığı azalabilmektedir. Bu uygulamada gezgin robotlar ile harita çıkarma işlemlerinde robotların konumlarının hesaplanması için kör konumlandırma sistemlerinden faydalanılmaktadır. Bölüm 2.'de de açıkça görüldüğü gibi kör konumlandırma sistemleri zaman içinde toplamsal olarak hatanın büyüdüğü sistemlerdir. Yani zamanla beraber sistemin hatası birikerek artmaktadır. Bu nedenle uygulamamız için zaman çok önemli bir etken haline gelmektedir. Bu konumlandırma hatalarına ek olarak, robotların başlangıç konumlarının doğru bir şekilde tespit edilememesi haritalama işlemlerinde başka bir problemdir. Robotların başlangıç konumlarının doğru bir şekilde tespit edilememesinin temel nedeni robotların uygulama esnasında alana tam olarak istenilen konum ve yönelim açılarında yerleştirilememeleridir. Bu küçük hatalar, Şekil 4.5.'de boşluklara yönelme yöntemi kullanılarak tamamlanan bir harita örneğinden de görüldüğü gibi haritaların kaymasına ve şekillerin kalınlaşmasına sebep olmaktadır. Çoklu-robotlarla oluşturulan bu haritada farklı robotların, engellerin ve duvarların farklı yüzlerini görerek en kısa zamanda bu bilgilerin tek bir haritada birleştirilmesi sağlanmaktadır.

Şekil 4.5.'te bilgisayar üzerinde robotların bireysel olarak boşluklara yönelme yöntemiyle oluşturdukları son haritalar ve bunların birleştirilmesiyle oluşturulan birleşik harita gösterilmektedir. Alanın son haritası yaklaşık olarak 3 dakikada oluşturulmuştur. Sol üst köşedeki harita soldan duvar takibi yapan robota, sağ üst köşedeki harita sağdan duvar takibi yapan robota ve sol alt köşedeki harita ise rastgele dolaşma görevini üstlenen robota ait bireysel son haritalardır. Bu bireysel



haritaların birleştirilmesiyle ise sağ alt köşede gösterilen alanın son haritası oluşturulmaktadır.



Şekil 4.5. Boşluklara yönelerek yardımlaşma yöntemiyle olarak oluşturulan bireysel haritalar ve birleştirilen son harita.

Şekil 4.5. dikkatle incelendiğinde rastgele dolaşma görevini üstlenen robot duvar takibi yapan robotların geçtikleri noktalardan geçmeden boşluklara yönelerek alan içinde duvar takibi yapan robotların henüz geçmedikleri yerleri aramakta ve haritalama işlemimin verimini artırmaktadır. Haritalama işlemlerinde ziyaret edilen alanın tekrar ziyaret edilmesi haritalama işlem süresini uzatmakta ve harita üzerindeki hataları artırmaktadır. Bu sebepten dolayı taranan bir alanın daha az tekrar ziyaret edilmesiyle daha tutarlı haritalar oluşturulmaktadır.

Boşluklara yönelerek yardımlaşma yöntemiyle oluşturulan harita tek robotlu haritalamaya göre yaklaşık olarak üç kat daha az zamanda tamamlanmıştır. Kör konumlandırma sistemlerindeki zamanla toplanarak büyüyen hatalardan dolayı da yardımlaşmalı haritalamada daha az hata oluşmuş ve daha gerçek alan ile daha tutarlı bir harita oluşturulmuştur.

### **4.3. Yapay Potansiyel Fonksiyonlar Kullanarak Yardımlaşmalı Haritalama**

Teorik incelemesi bir önceki bölümde anlatılan yapay potansiyel fonksiyonlar robot gezinmelerinde sıkça başvurulan bir yöntemdir. Bu bölümde ise yardımlaşmalı harita çıkarma işlemi yapan robotlar üzerinde yapılan bir çalışma anlatılmaktadır. Bu çalışmada bir önceki bölümde yapılan haritalama deneyi aynı şartlar altında rastgele dolaşma işlemi yapan robotun diğer robotların geçtikleri noktalardan kaçınmak için kullandığı boşluklara yönelme yöntemi yerine yapay potansiyel fonksiyonlar yöntemini kullanmasıyla tekrarlanmaktadır. Bu yöntemde rastgele dolaşma görevini üstlenen robot duvar takibi yapan robotların geçtiği noktalarda yapay itim kuvvetleri sayesinde kaçınmaktadır.

Daha önceki bölümlerde anlatıldığı gibi alan içinde rastgele dolaşma yapan robottan veri geldiğinde diğer robotların geçtikleri tüm konumların kayıtlı olduğu “.txt” dosyasından diğer robotların konumları okunarak rastgele dolaşma yapan robotun duvar takibi yapan robotların geçtikleri noktalara ne kadar yakın olduğu bilgisayar üzerinde hesaplanmaktadır. Eğer hesaplanan değer belirli bir mesafeden küçük ise bu mesafe için Şekil 4.2.’de grafiği verilen potansiyel fonksiyonun itim kısmı için yapay itim kuvvet vektörü hesaplanmakta ve rastgele dolaşma yapan robota daha önce

ziyaret edilen bir bölgede olduğu işareti ve bu itim kuvveti vektörünün bileşenleri gönderilmektedir. Burada bahsedilen itim mesafesi bu deneyler için 20 cm olarak seçilmiştir. Yani rastgele dolaşma işlemi yapan robot duvar takibi yapan robotların geçtiği noktalara 20 cm kadar yaklaştığında bilgisayar tarafından robota daha önceden ziyaret edilmiş bir alana girdiği işareti gönderilmektedir. Bu sayede alan içinde rastgele dolaşma işlemi yapan robot bu alandan çıkması gerektiğini anlamakta ve çıkmak için ise bilgisayar tarafından gönderilen bu itim vektörünü kullanmaktadır. Robot bulunduğu alandan çıkmak için bu itim kuvvet vektörü yönünde ve bu kuvvetin büyüklüğüne göre hareketine devam etmektedir. Bu yöntemle çıkarılan ve zamanla büyüyen harita örnekleri Şekil 4.6.'te gösterilmektedir.

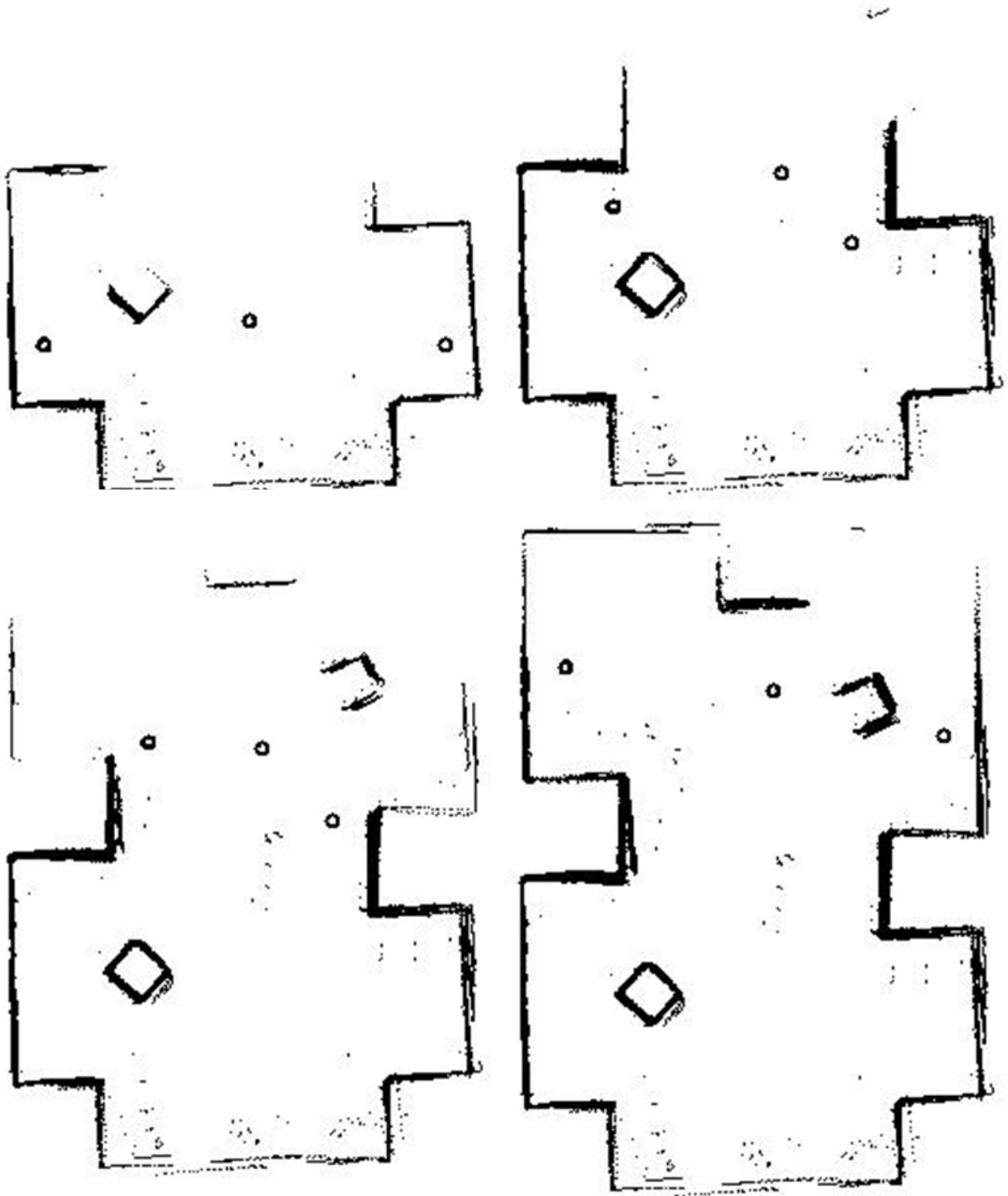
Şekil 4.6.'da alan içindeki daireler robotların o anda alan içinde buldukları konumları ifade etmektedir. Bu haritalar üst soldaki haritadan alt sağdaki haritaya doğru 40., 80., 120. ve 160. saniyelerde bilgisayar üzerinde zamanla oluşturulan haritalardır. Şekil 4.7.'de ise bilgisayar üzerinde robotların bireysel olarak potansiyel fonksiyonlar yöntemiyle oluşturdukları son haritalar ve bunların birleştirilmesiyle oluşturulan son harita gösterilmektedir.

Şekil 4.7.'de alanın son haritası yaklaşık olarak 3 dakikada oluşturulmuştur. Sol üst köşedeki harita soldan duvar takibi yapan robota, sağ üst köşedeki harita sağdan duvar takibi yapan robota ve sol alt köşedeki harita ise rastgele dolaşma görevini üstlenen robota ait bireysel son haritalardır. Bu bireysel haritaların birleştirilmesiyle ise sağ alt köşede gösterilen alanın son haritası oluşturulmaktadır.

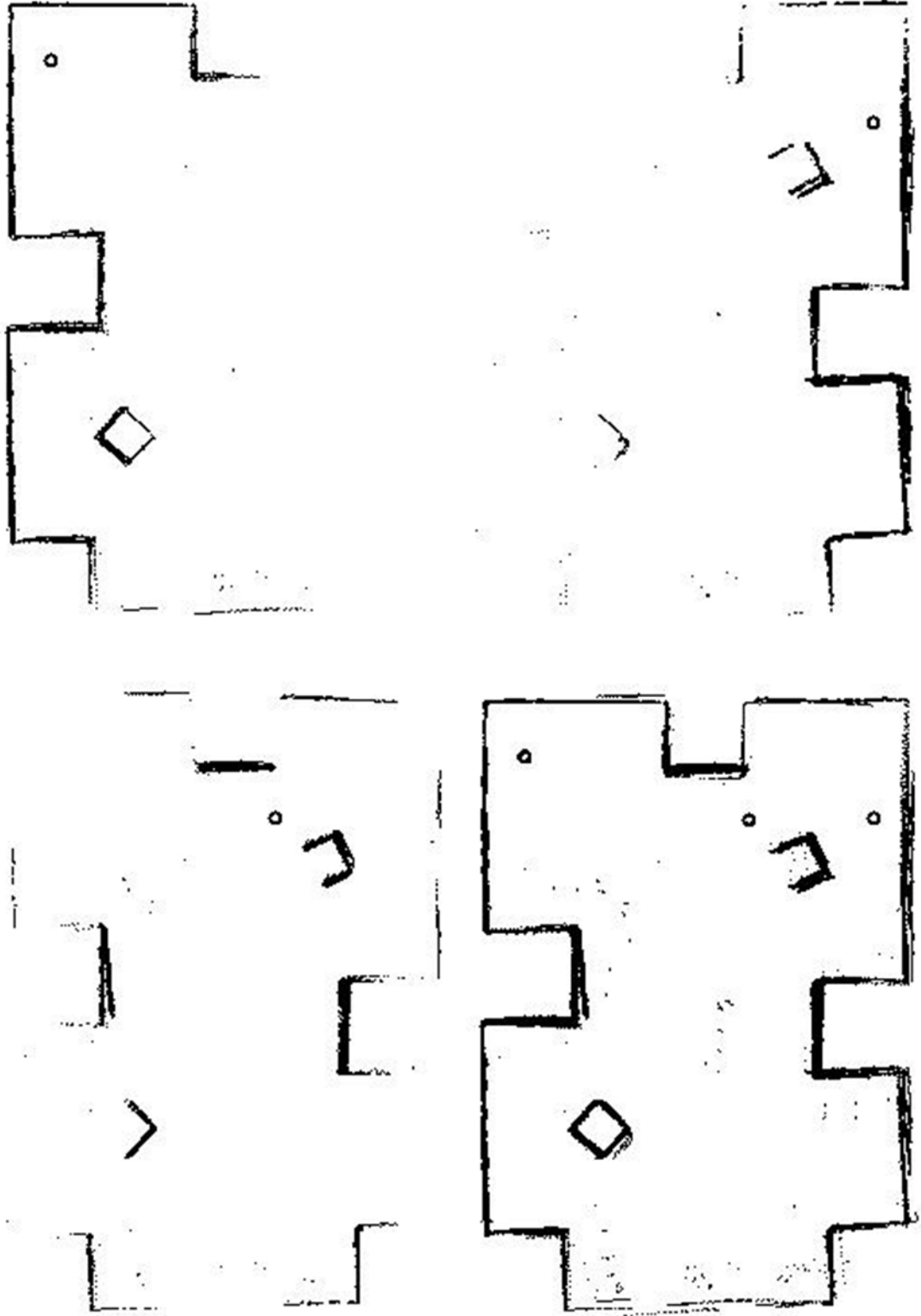
Şekil 4.7. dikkatle incelendiğinde rastgele dolaşma görevini üstlenen robot potansiyel fonksiyonlar kullanarak duvar takibi yapan robotların geçtikleri noktalardan geçmeden alan içinde duvar takibi yapan robotların henüz geçmedikleri yerlere doğru yönelmekte ve böylelikle yardımlaşarak haritalama işleminin verimini artırmaktadır.

Yapılan bu deneylerde boşluklara yönelerek yardımlaşmalı harita çıkarma yöntemi ile potansiyel fonksiyonlar kullanarak yardımlaşmalı harita çıkarma yöntemi arasında haritalar üzerinde görsel bir fark olmamasına rağmen potansiyel fonksiyonlar çalışma yöntemiyle diğer yöntemlere göre daha akılcı bir yöntemdir. Çünkü potansiyel itim kuvvetleri sayesinde alan içinde rastgele dolaşan robot diğer robotların henüz

geçmedikleri yönlerde doğru yönelip rastgele dolaşma işlemini bu yönde devam ettirmektedir.



Şekil 4.6. Potansiyel fonksiyonlar kullanarak yardımlaşma yöntemiyle zamanla büyüyen haritalar.



Şekil 4.7. Potansiyel fonksiyonlar kullanarak yardımlaşma yöntemiyle oluşturulan bireysel haritalar ve birleştirilen son harita.

Boşluklara yönelerek yardımlaşmalı haritalamada ise diğer robotların geçtikleri noktaları hesaba katmadan sadece etrafında hareket edebileceği noktalara bakmaktadır. Diğer robotlar alan içinde rastgele dolaşan robotun hareket edebileceği bu boşluk yönünden daha önce geçtilerse robot bu noktalara kadar hareketini devam ettirecek ve daha sonra bilgisayardan taranmış alan bilgisi geldiğinde bu alandan ayrılmaya karar verecektir. Potansiyel fonksiyonlar yönteminde ise bu boşluklara hiç yönelmeden doğrudan diğer robotların geçmediği alanlara doğru yönelecektir. Bu sayede robot zaman kaybetmeden taranmamış alanları taramaya devam edecektir. Deney alanı yeterince büyük olmadığından bu iki yöntem arasındaki fark tam olarak gözlemlenememektedir. Deney alanının daha büyük olması ve mutlak konumlandırma sistemlerinin ulaşılabilir olduğu durumlarda potansiyel fonksiyonlar kullanılarak yardımlaşmalı haritalamanın çok daha iyi sonuçlar vereceği tahmin edilmektedir.

## BÖLÜM 5

### 5. SONUÇ

#### 5.1. Yorumlar

Bu tez çalışmasında robot sistemleri için kör konumlandırma iyileştirmesi, bu iyileşmenin tek robotlu haritalama sistemlerine etkisi ve iki farklı seyrüsefer yöntemi ile çok robotlu yardımlaşmalı haritalama sistemi üzerine incelemeler yapılmıştır. Burada kör konumlandırma sistemlerinin hata oranlarının düşürülerek robot çalışma sistemi maliyetlerinin düşürülmesi ve gerçek alan ile daha tutarlı haritalar çıkarılması hedeflenmiştir. Bu amaçla tüm deneyler gerçek robotlar üzerinde uygulanmış ve deney sonuçları ayrıntılı bir şekilde incelenmiştir.

Yapılan ilk çalışmada tepe kamerasıyla mutlak konumlandırmanın yapıldığı deney alanında kör konumlandırma, IMU desteli kör konumlandırma ve parçacık süzgeci uygulanarak IMU destekli kör konumlandırma ile bir robotun önceden belirlenmiş kare şeklinde bir güzergâhı takibi deneyleri yapılmıştır. Bu deneylerde her bir konumlandırma sistemi ile robotun güzergâhı ne derecede takip edebildiği, son konumlarındaki hata oranları ve son yönelim açılarındaki hata oranları incelenmiştir. Kör konumlandırma ile güzergâh takibi deneyinde konumlandırma sistemindeki hatalardan dolayı robotun hiçbir zaman başlangıç noktasına gerçekte dönemediği, konum ve yönelim açısından hataların ise oldukça fazla olduğu gözlemlenmiştir. IMU destekli kör konumlandırma sistemi ile güzergâh takibi deneyleri sonucunda robot birçok kez başlangıç noktasına dönebilmiştir. Bu yöntem ile robotun son konum ve yönelimindeki hatası kör konumlandırma sistemi ile güzergâh takibine göre ortalama %80'e daha düşmüştür. Parçacık süzgeci uygulanarak IMU destekli kör konumlandırma sistemi ile güzergâh takibinde ise hata azalma oranları ortalama olarak yaklaşık % 7 olarak hesaplanmıştır. Bu oran IMU destekli kör konumlandırma sistemine göre oldukça düşüktür ve gerçekte IMU destekli kör konumlandırma sistemine bozucu bir etkide bulunmuştur. Bu bozucu etkinin temel sebepleri parçacık süzgecinin çalışma prensibinde mutlak konumlandırma sistemleri ile robot

konumunun aralıklarla düzeltilmesinin gerekmesi ve parçacık süzgecinin parametreleri hesaplanırken standart sapmaların çok iyi hesaplanmasının gerekliliğidir. Bu çalışmada ise mutlak konumlandırma sistemi olmadan IMU destekli kör konumlandırma sistemi çıktılarının gerçek konumlandırma olarak kabul edildiği durumda parçacık süzgecinin sisteme olan etkisi incelenmiştir. Parçacık süzgecinin sisteme bozucu bir etki yaptığı gözlemlenmiştir. Yapılan incelemelerde standart sapmaların hesaplanması için kullanılan tepe kamerası ile konumlandırma sisteminin hata oranının IMU destekli konumlandırma sisteminin hata oranından daha fazla olması nedeni ile standart sapmalar doğru bir şekilde hesaplanamamıştır. Parçacık süzgecindeki parçacık ağırlıklarının güncellenmesi için gerekli olan mutlak konumlandırmanın yerine parçacık süzgeci kullanılarak IMU destekli kör konumlandırma sistemini çıktılarının kullanılması yani hatalı veriye göre ağırlıkların güncellenmesi yukarıda bahsedilen bozucu etkilerin sebebi olarak tespit edilmiştir.

Yapılan diğer bir çalışma ise IMU algılayıcısı ile kör konumlandırma sisteminde gerçekleştirilen iyileşmenin tek robotlu haritalama sistemlerine etkisinin ve lazer görüş mesafesinin haritalama üzerindeki etkisinin incelenmesidir. Yapılan deneyler sonunda kör konumlandırma sistemindeki iyileşme sayesinde gerçek alan ile daha tutarlı haritalar oluşturulabildiği gözlemlenmiştir. Robot konumlandırmasında oluşan hatalar çizilen nesnelerin harita üzerindeki yerlerinin kaymasına ve şekillerin kalınlaşmasına neden olmaktadır. Bu nedenle haritalama sistemlerinde robot konumunun tam olarak bilinmesi çok önemlidir. Haritalama sistemlerinde oluşan hataların temel sebebi robotun yönelim açısındaki hatalardır. Robotun yönelim açısındaki hata robota uzak mesafede olan bir nesnenin harita üzerindeki konumunun yakın mesafede olan bir nesneye göre daha fazla kaymasına sebep olmaktadır. Bu yönelim açısındaki hataların haritaya etkisini en aza indirebilmek için sadece belirli bir lazer görüş mesafesi için harita çizimi yapılmasının gerçek alan ile daha tutarlı bir harita oluşturulmasına yardımcı olduğu yapılan deneylerle gözlemlenmiştir.

Bir sonraki çalışmada ise kör konumlandırma sistemi kullanılarak iki farklı seyrüsefer yöntemi kullanılarak yardımlaşmalı çok robotlu haritalama sistemleri üzerine çalışılmıştır. Bu hareket yöntemlerinden ilki boşluklara yönelerek diğer robotların güzergâhlarından uzaklaşma, diğeri ise potansiyel fonksiyonlar kullanarak



diğer robotların güzergâhlarından uzaklaşmadır. Boşluklara yönelerek haritalamada rastgele dolaşma işlemini üstlenen robot duvar takibi yapan robotların geçtikleri yerlerden geçmemek için o anda lazer tarayıcısı ile tespit ettiği en uzak ölçüm yönüne dönerek rastgele dolaşma hareketine bu yönde devam ettiği gözlemlenmiştir. Bu yöntemde rast gele dolaşan robot yöneldiği tarafta duvar takibi yapan diğer robotların daha önce geçtiği başka bir güzergâh ile karşılaşarak yönünü tekrar değiştirmek zorunda kalabilmektedir. Bu ise haritanın tamamlanma süresinin artmasına sebep olabilmektedir. Potansiyel fonksiyonlar kullanarak haritalama işleminde ise alan içinde rastgele dolaşan robot duvar takibi yapan robotların daha önce geçtikleri bir alan ile karşılaştığında belirli bir alan içinde robotların geçtikleri noktalar için hesaplanan potansiyel itim kuvveti sayesinde duvar takibi yapan robotların henüz geçmedikleri yönde hareketine devam ettiği gözlemlenmiştir. Bu sayede haritalama işlem süresi kısalabilmekte ve gerçek alan ile daha tutarlı haritalar çıkarılabilmektedir. Yukarıda bahsedilen iki yöntem karşılaştırıldığında deney alanının küçük olması sebebiyle aralarında gözlemlenebilir bir fark olmadığı görülmüştür. Fakat daha büyük deney alanları için potansiyel fonksiyonlar kullanarak haritalama ile çok daha kısa sürede gerçek alan ile daha tutarlı haritalar çıkarılabileceği düşünülmektedir.

## **5.2. Gelecek Çalışmalar**

Yapılan çalışmalarda kullanılan tepe kameralı konumlandırma sisteminin hata oranın IMU destekli kör konumlandırma sisteminin hatasından daha fazla olması nedeniyle parçacık süzgecinin parametreleri doğru olarak hesaplanamamıştır. Gelecekteki çalışmalarda mutlak konumlandırma sistemlerinden faydalanarak ve IMU destekli kör konumlandırma sisteminin hata oranından daha az hata oranına sahip bir konumlandırma sistemi ile parçacık süzgecinin IMU destekli kör konumlandırma sistemine uygulanması yararlı olacaktır. Böylelikle parçacık süzgeci ile kör konumlandırma sistemindeki hatalar daha da azaltılabilecek ve mutlak konumlandırma sayesinde düşük hata oranlarıyla güzergâh takibi yapılabilecektir.

Daha sonraki aşamada ise parçacık süzgeci ile elde edilen iyileşme ve mutlak konumlandırma sisteminin tek robotlu haritalama üzerindeki etkisinin incelenmesi

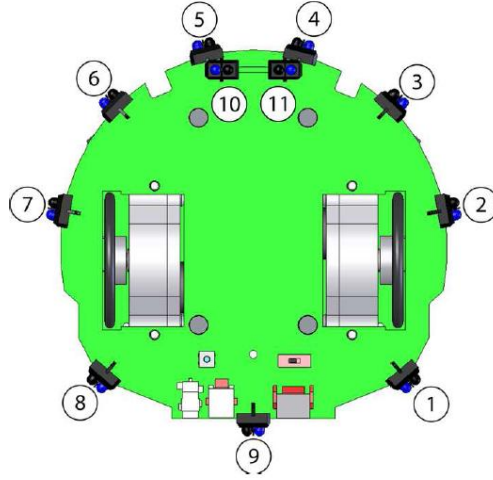
düşünülmektedir. Bu amaçla kör konumlandırma ve IMU destekli kör konumlandırma ile tek robotlu haritalama deneylerine ek olarak parçacık süzgeci uygulanarak tek robotlu haritalama deneylerinin tekrarlanması ve aradaki farkların incelenerek haritalamada bir iyileşme olup olmadığının belirlenmesi gerekmektedir.

Bir sonraki adım ise parçacık süzgeci uygulanarak IMU destekli kör konumlandırma sistemi ile yardımlaşmalı çok robotlu haritalamadır. Yapılan deneylerde KheperaIII robotunun işlem gücünün IMU destekli kör konumlandırma sistemi ile yardımlaşmalı çok robotlu haritalama için yetersiz olduğu gözlemlenmiştir. Bu sorunun aşılabilmesi için IMU ve lazer algılayıcılarından verileri okuyup işleyen ve istenen sonucu sürekli olarak robota gönderen bir mikrodenetleyici devresi tasarlanabilir. Böylelikle robot, IMU algılayıcı verilerinin okunması ve işlenmesi için zaman ayırmak yerine işlenmiş hazır verileri kullanacaktır. Ana programın işletilmesi için daha çok işlem gücü ayrılabilir ve çok-robotlu haritalamada IMU ve lazerin bir arada çalıştırılması ile ilgili bir sorun yaşanmayacaktır. Bu tez çalışmasında karşılaşılan diğer bir sorun ise haritalama deneylerinin yapıldığı alanın küçük olmasıdır. Bu nedenle boşluklara yönelerek ve potansiyel fonksiyonlar kullanarak yardımlaşmalı çok-robotlu haritalama yöntemlerinin harita üzerindeki etkilerinin tam olarak karşılaştırılamamıştır. İki farklı haritalama yönteminin haritalama üzerindeki etkilerinin tam olarak incelenebilmesi için gelecek çalışmalarda aynı haritalama deneyleri kör konumlandırma ve parçacık süzgeci uygulanarak IMU destekli kör konumlandırma ile yardımlaşmalı çok robotlu haritalama deneyleri daha büyük bir deney alanında tekrarlanabilir.

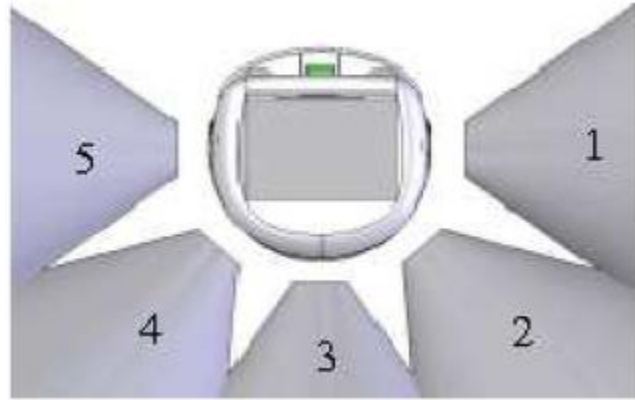
## **EKLER**

### **A. KHEPERAIII ROBOTU**

KheperaIII mini gezgin robotları (Şekil 2.1.) İsviçre’de K-Team firması tarafından üretilmektedirler. Bu robotlarda işlemci olarak 400 Mhz’de çalışan Intel PXA255 işlemcisi bulunmaktadır. Robotların hareketi 2 adet fırçasız DC servo motor ile sağlanmaktadır. Motorlar iki adet PIC18F4432 mikrodenetleyici tarafından PID denetleyicisi ile sürülmekte ve bu mikrodenetleyiciler motorların mesafe sayaç değerlerinin elde edilmesi için de kullanılmaktadırlar. Denetleyiciler motorların hız veya konumu denetimini gerçek hız ve sayaçlardan okunan konum bilgisi kullanarak işaret genişlik modülasyonu değerini ayarlayarak yapmaktadırlar. Motor denetim blokları i2c veri yolunda yönetilen donanım olmakta ve yönetici işlemci ile i2c veri yolu ile haberleşmektedir. Robotların ön ve yan taraflarında laboratuvar ortamında 20 cm’ye kadar ölçüm yapabilen 9 adet, alt tarafında 2 adet olmak üzere toplam 11 adet Vishay Telefunken TCRT5000 kızıl ötesi (Şekil A.1.) ve ön ve yan taraflarında ise 20 cm ile 270 cm arasında ölçüm yapabilen 5 adet ses ötesi algılayıcı (Şekil A.2.) (verici Midas 400ST100, alıcı Midas 400SR100) bulunmaktadır. Kızıl ötesi ve ses ötesi algılayıcılarda alıcı ve verici üniteleri bir arada bulunmaktadır. Algılayıcı okumalarının/ölçümlerinin elde edilmesi için 60Mhz’de çalışan DSPIC30F5011 mikrodenetleyicisi kullanılmaktadır. Algılayıcı değerlerinin elde edildiği DSPIC30F5011 ve motor değerlerinin elde edildiği iki adet PIC18F4432 mikrodenetleyicisi yamak işlemciler olarak i2c veri yolu usta işlemci olan INTEL PXA255 işlemcisi ile haberleşmektedir.



Şekil A.1. KheperaIII robotu kızıl ötesi algılayıcıları dizilimi.

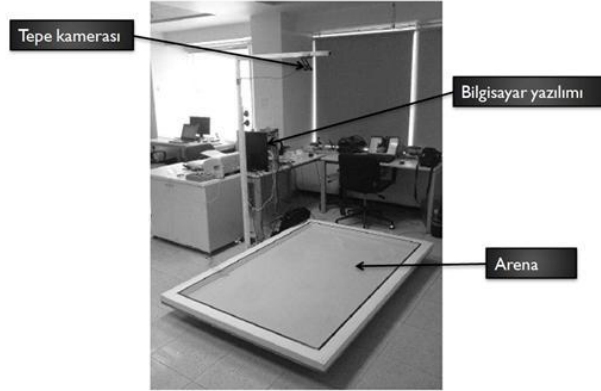


Şekil A.2. KheperaIII robotu ses ötesi algılayıcıları dizilimi.

## B. TEPE KAMERALI DENEY DÜZENEĞİ

### B.1. Deney alanı

Bu deney düzeneğinin açıklanmasında Şamiloğlu A. T.'nin çalışmasından [48] faydalanılmıştır. Tepe kameralı deney düzeneği 120x180 cm boyutlarında dört tarafı kapalı pürüzsüz bir zemine sahip robot alanından, tepe kamerası olarak sekiz mega pixel web kamerasından ve kameradan alınan görüntünün işlenmesi için Matlab yüklü bir bilgisayardan oluşmaktadır (Şekil B.1.).



Şekil B.1. Robot alanı, tepe kamerası ve masaüstü bilgisayardan oluşan deney düzeneği.

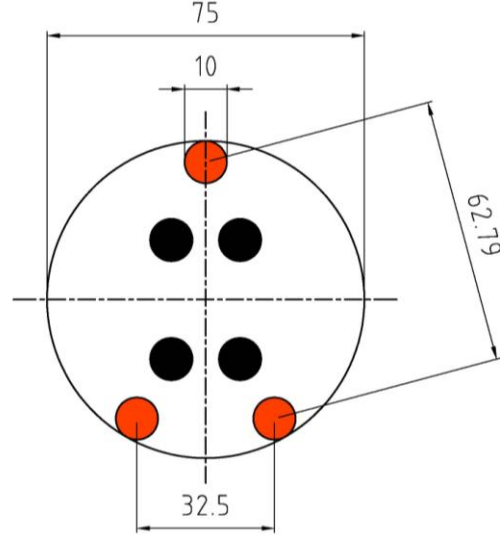
Görüntü işlemenin hızlanması açısından kameradan görüntüler 640x480 çözünürlüğünde alınarak işleme konulmaktadır. Görüntü işleme hızı saniyede 4-5 kareyi aşmadığından dolayı buradaki sistemde yüksek yakalama hızına sahip kameraya ihtiyaç duyulmamıştır. Bu sebeple sıradan bir web kamerası yeterli olmaktadır. Fakat görüntü kalitesi sistemin performansını çok fazla etkilediği için kaliteli bir lense sahip, düşük ışık koşullarında bile net görüntü verebilen CCD (Charge Coupled Device) algılayıcıya sahip Logitech QuickCam Pro9000 adlı web kamera bu çalışmada kullanılmak üzere seçilmiştir.

Görüntü işleme Matlab yazılımı aracılığıyla gerçekleştirilmiştir. Matlab yazılımının tercih edilmesinin bir diğer nedeni ise içerisinde bulunan görüntüye erişim ve görüntü işleme kütüphaneleridir.

## B.2. Kullanılan Görüntü İşleme Yöntemleri

Gezgin robot uygulamalarının büyük bir kısmında robotlar çalışabilmek için bütünsel ya da göreceli konumlara ihtiyaç duymaktadırlar. Tepe kameralı deney düzeneğinde web kamerası kullanılarak robotların konum, yönelim ve gerektiğinde kimlik bilgileri görüntü işleme ile bulunmaktadır. Şekil B.2.'de robotların algılanmasını sağlayan kodlama sisteminin örnek bir görüntüsü verilmiştir. Bu şekiller robotların üzerine yerleştirilen 75 mm çapındaki dairesel şapkalara çizilmiştir. Burada robotların konumları ve yönelim açıları kenarlardaki dışsal alandaki noktalar ile

robotların kimlikleri ise merkezdeki noktalardan oluşan ikili bir kodlama sistemi ile belirlenmektedir.



Şekil B.2. Robotların konum ve yönelimlerinin bulunmasını sağlayan kodlama sisteminin boyutları mm cinsinden olan örnek görüntüsü.

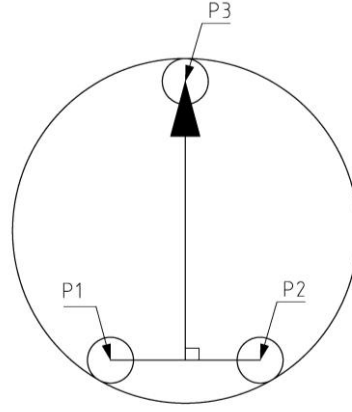
### B.2.1. Robot Konumlarının Bulunması

Robot konumlarının bulunması için görüntüdeki dışsal daireler kullanılmaktadır. Bu amaçla kameradan alınan görüntü dışsal dairelerin belirlenmesi için bir filtreden geçirilmektedir. Bu sayede sadece algılanması istenilen daireler ortaya çıkmaktadır. Ayrıca yapılan çalışmalarda renkli nesnelere ayırt etmekte ve istenilen renklerin bulunmasında daha başarılı sonuçlar verdiği için HSV (Renk özü, Doymuluk, Değer) renk uzayının kullanılması uygun görülmüştür.

Görüntü üzerinde dışsal daireler bulunduktan sonra bulunan bu daireler birleştirilerek etiketlenmekte ve etiketlenen bu nesnelere merkezleri bulunmaktadır. Bir sonraki aşamada ise birden fazla robot kullanılması durumunda bulunan bu merkez noktalarının hangi üçünün aynı robota ait olduğunun bulunması gerekmektedir. Basit bir çözüm olarak birbirine en yakın üç noktanın aynı robota ait olduğu varsayılmıştır. Birden fazla robotun kullanılması gerektiği durumlarda bu yöntem oldukça hızlı sonuç vermesine rağmen bunun bir dezavantajı da robotlar birbirine çok yaklaştıklarında noktalar birleşerek hatalara sebep olmaktadır. Bu sebeple robotların birbirlerine yaklaşabilecekleri maksimum bir mesafe değeri belirlenmiştir.

### B.2.2. Robotların Yönelimlerinin Bulunması

Kenarlarda bulunan dışsal daireler bir ikizkenar üçgen oluşturmaktadır. Eşit uzunluktaki kenarların birleştiği nokta  $P_3$  ve geriye kalan iki nokta  $P_1$  ve  $P_2$  olarak işaretlenirse (bakınız Şekil B.3.). Bu noktalardan  $P_1$  ve  $P_2$ 'nin orta noktalarını  $P_3$  ile birleştiren vektör robotun yönünü göstermektedir.



Şekil B.3. Robotların konum ve yönelimlerinin bulunmasını için kullanılan renkli noktalar.

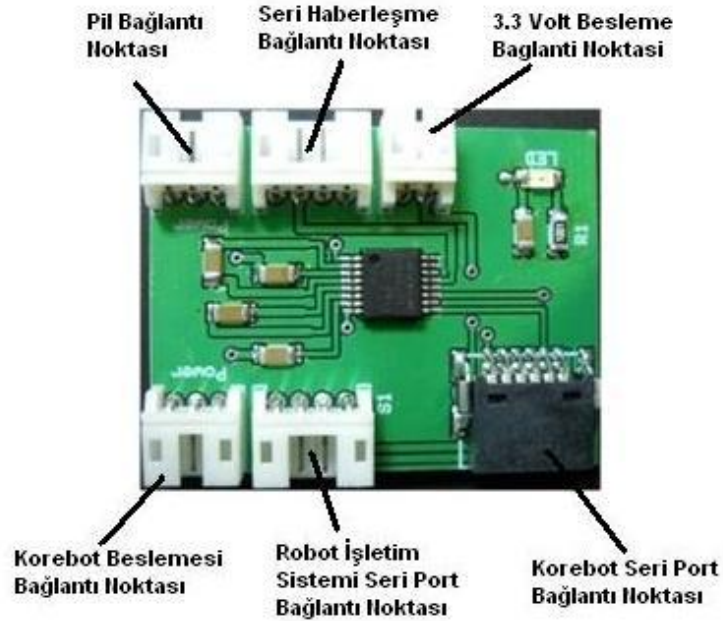
## C. IMU ALGILAYICI

Bu tez çalışmasında kullanılan ve Microstrain firması tarafından üretilen 3DM-GX2 IMU algılayıcısından RS232 haberleşme yöntemini ile veri okunmaktadır. KheperaIII robotunun üzerinde bulunan Korebot LE kartı seri haberleşme için TTL 3.3 V seviyesini kullanmasına rağmen IMU algılayıcı RS232 voltaj seviyesini kullanmaktadır. Bu sebepten dolayı IMU algılayıcı ve Korebot LE kartının haberleşmesinin sağlanması amacı ile bu tez çalışması kapsamında ara haberleşme kartı tasarlanmıştır. Bu ara haberleşme kartı, MAX3221 RS232 seviye çevirici ve ek olarak diğer devre elemanlarından oluşmaktadır. MAX3221 entegresi RS232 TTL seviyesini +2.5 V ile +5.5 V arası TTL seviyesine dönüşümünü yapan bir entegredir. Bu entegre sayesinde Korebot LE kartı ile IMU algılayıcısı arasında iletişim sağlanmaktadır. IMU algılayıcısından veri okunabilmesi için yapılan diğer bir çalışma ise 3DM-GX2 algılayıcısının Linux ortamında C tabanlı sürücüsünün

yazılmasıdır. Bu sürücüde IMU algılayıcısından tüm veriler C ortamında okunabilmektedir.

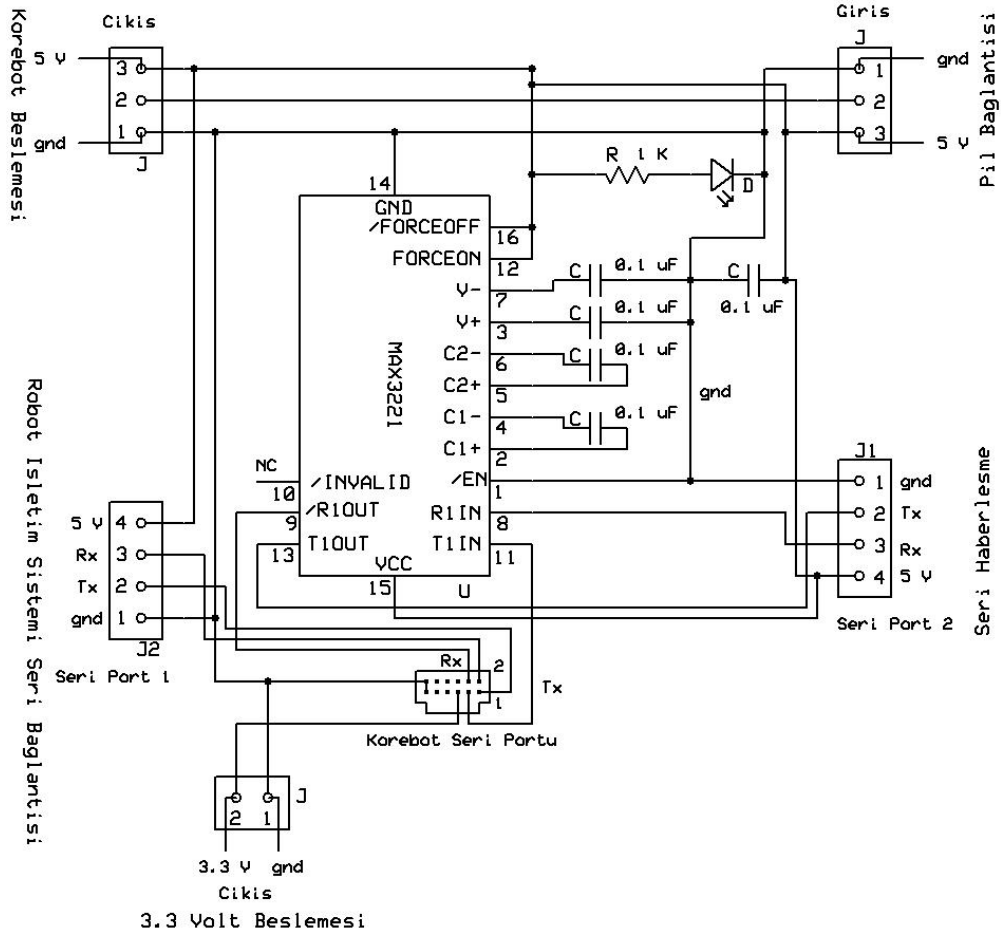
### C.1.1. Ara Haberleşme Kartı

IMU algılayıcısının KheperaIII robotuna bağlanıp veri okunabilmesi için özel olarak geliştirilen bu kart IMU algılayıcısından veri okunabilmesini sağlamanın yanında IMU, Lazer ve Korebot kartı için gerekli olan gücü sağlamaktadır. Kart üzerinde bulunan pil bağlantı noktası, 5 Volt, ve 3.3 Volt beslemeleri ve biri seri port haberleşmesi diğeri ise robot işletim sistemine seri port ile bağlanmak için 2 adet seri port bağlantısı olmak üzere çeşitli bağlantı noktaları mevcuttur. KheperaIII mini gezgin robotuna özel kart üzerinde SMD türünde devre elamanları kullanılarak 3 x 4 cm boyutlarında ve çift taraflı bir ara haberleşme kartı üretilmiştir (Şekil C.1.). Şekil C.2. 'de ise devrenin şematik gösterimi verilmektedir.



Şekil C.1. KheperaIII haberleşme kartı.





Şekil C.2. KheperaIII haberleşme kartı şematik gösterimi.

### C.1.2. IMU Algılayıcısı Linux Sürücüsü

3DM-GX2 IMU algılayıcısı verilerinin robot üzerinde çalışan kontrol programları tarafından okunabilmesi için IMU algılayıcısının Linux sürücüsüne ihtiyaç duyulmaktadır. Fakat bu sürücünün üretici firma tarafından sağlanmamasından dolayı IMU algılayıcısının robot gezinmelerinde kullanılabilmesi için bu tez çalışması kapsamında TOBB Ekonomi ve Teknoloji Üniversitesi Sürü Sistemler Araştırma Laboratuvarı'nda bir Linux sürücüsü geliştirilmiştir. Bu sürücü aracılığı ile robot üzerinde IMU algılayıcısından veri okumak için gerekli fonksiyonlar ve robota özel kütüphaneler aşağıda verilmiştir.

---

Fonksiyon : int OpenIMUPort ()

Açıklama:

Bu fonksiyon Microstrain 3DM-GX2 IMU algılayıcısından 115200 baudrate hızında veri okunabilmesi için robotun seri haberleşme portunu açmaktadır.

Çalışma Ortamı: KheperaIII Robotu

Gereksinimler : Korebot üzerinde “/dev/ttyS2” seri portuna bağlı Microstrain 3DM-GX2 IMU algılayıcısı ve khepera3toolbox

Header Dosyaları: IMU\_3DM\_GX2.h

Fonksiyon Girdileri : Yok

Fonksiyon Çıktıları:

- int flag : Başarı durumunda flag=1 diğer durumlarda flag=0 dönmektedir.

---

Fonksiyon : void CloseIMUPort ()

Açıklama:

Bu fonksiyon Microstrain 3DM-GX2 IMU algılayıcısından veri okunabilmesi için daha önceden açılmış olan robotun seri haberleşme portunu kapatmaktadır.

Çalışma Ortamı: KheperaIII Robotu

Gereksinimler : Korebot üzerinde “/dev/ttyS2” seri portuna bağlı Microstrain 3DM-GX2 IMU algılayıcısı ve khepera3toolbox

Header Dosyaları: IMU\_3DM\_GX2.h

Fonksiyon Girdileri : Yok

Fonksiyon Çıktıları: Yok

---

Fonksiyon : int SetContinuousMode(char \*commandname)

Açıklama:

Bu fonksiyon Microstrain 3DM-GX2 IMU algılayıcısını sürekli veri yollama moduna geçirmektedir. Bu fonksiyon ile kullanıcı tarafından gönderilen komuta ait olan ölçüm sonuçları kullanıcıdan tekrar komut beklenmeden sürekli olarak robota gönderilmektedir.

Çalışma Ortamı: KheperaIII Robotu

Gereksinimler : Korebot üzerinde “/dev/ttyS2” seri portuna bağlı Microstrain 3DM-GX2 IMU algılayıcısı ve khepera3toolbox

Header Dosyaları: IMU\_3DM\_GX2.h

Fonksiyon Girdileri:

- char \*commandname : Komut isminin bulunduğu diziyi işaret eden pointer. Komut isimlerinin tam listesi IMU\_3DM\_GX2.h dosyasının içinde verilmektedir.

Fonksiyon Çıktıları:

- int flag: Başarı durumunda flag=1 diğer durumlarda flag=-1 dönmektedir.

Referanslar : [42].

---

Fonksiyon : void StopContinuousMode()

Açıklama:

Bu fonksiyon sürekli veri gönderme modunda olan Microstrain 3DM-GX2 IMU algılayıcısını bu moddan çıkarıp tek tek veri okuma geçirmektedir.

Çalışma Ortamı: KheperaIII Robotu

Gereksinimler : Korebot üzerinde “/dev/ttyS2” seri portuna bağlı Microstrain 3DM-GX2 IMU algılayıcısı ve khepera3toolbox

Header Dosyaları: IMU\_3DM\_GX2.h

Fonksiyon Girdileri:

- char \*commandname : Komut isminin bulunduğu diziyi işaret eden pointer. Komut isimlerinin tam listesi IMU\_3DM\_GX2.h dosyasının içinde verilmektedir.

### Fonksiyon Çıktıları:

- int flag : Başarı durumunda flag=1 diğer durumlarda flag=0 dönmektedir.

Referanslar : [42].

---

Fonksiyon : int IMUSendCommandByte(char\* commandname)

### Açıklama:

Bu fonksiyon Microstrain 3DM-GX2 IMU algılayıcısına komut göndermek için kullanılmaktadır.

Çalışma Ortamı: KheperaIII Robotu

Gereksinimler : Korebot üzerinde “/dev/ttyS2” seri portuna bağlı Microstrain 3DM-GX2 IMU algılayıcısı ve khepera3toolbox

Header Dosyaları: IMU\_3DM\_GX2.h

### Fonksiyon Girdileri:

- char \*commandname : Komut isminin bulunduğu diziyi işaret eden pointer. Komut isimlerinin tam listesi IMU\_3DM\_GX2.h dosyasının içinde verilmektedir.

### Fonksiyon Çıktıları:

- int flag : Başarı durumunda flag=1 diğer durumlarda flag=0 dönmektedir.

Referanslar : [42].

---

Fonksiyon : int IMUReadData(char \*command\_name, float \*buffer, float \*imutime)

### Açıklama:

Bu fonksiyon Microstrain 3DM-GX2 IMU algılayıcısından istenen komuta ait verinin okuması için kullanılmaktadır. Bu fonksiyon kullanılmadan önce algılayıcının sürekli veri okuma modunda olması ya da “IMUSendCommandByte” komutunun daha önceden gönderilmesi gerekmektedir.

Çalışma Ortamı: KheperaIII Robotu

Gereksinimler : Korebot üzerinde “/dev/ttyS2” seri portuna baęlı Microstrain 3DM-GX2 IMU algılayıcısı ve khepera3toolbox

Header Dosyaları: IMU\_3DM\_GX2.h

Fonksiyon Girdileri:

- char \*commandname : Komut isminin bulunduęu diziyi iřaret eden pointer. Komut isimlerinin tam listesi IMU\_3DM\_GX2.h dosyasının iinde verilmektedir.
- float \*buffer : Verinin yazılacaęı adresi iřaret eden float pointer.
- float \*imutime : 3DM-GX2 IMU algılayıcısının ölçümü yaptıktan sonra bu ölçüme ait yerel zamanın yazıldıęı adresi iřaret eden float pointer.

Fonksiyon Çıktıları:

- int flag : Bařarı durumunda flag=1 dięer durumlarda flag=0 dönmektedir.

---

Fonksiyon : WriteAccelerometerBias(float AccelBiasx, float AccelBiasy, float AccelBiasz)

Aıklama:

Bu fonksiyon Microstrain 3DM-GX2 IMU algılayıcısına ivmeöler bařlangı deęerlerini yazmak iin kullanılmaktadır.

alıřma Ortamı: KheperaIII Robotu

Gereksinimler : Korebot üzerinde “/dev/ttyS2” seri portuna baęlı Microstrain 3DM-GX2 IMU algılayıcısı ve khepera3toolbox

Header Dosyaları: IMU\_3DM\_GX2.h

Fonksiyon Girdileri:

- float AccelBiasx : x ekseninde yazılacak ivmeöler bařlangı deęeri
- float AccelBiasy : y ekseninde yazılacak ivmeöler bařlangı deęeri
- float AccelBiasz : z ekseninde yazılacak ivmeöler bařlangı deęeri

Fonksiyon Çıktıları:

- int flag : Bařarı durumunda flag=1 dięer durumlarda flag=-1

dönmektedir.

Referanslar : [42].

---

Fonksiyon : int WriteGyroBias(float GyroBiasx, float GyroBiasy, float GyroBiasz)

Açıklama:

Bu fonksiyon Microstrain 3DM-GX2 IMU algılayıcısına cayro başlangıç değerlerini yazmak için kullanılmaktadır.

Çalışma Ortamı: KheperaIII Robotu

Gereksinimler : Korebot üzerinde “/dev/ttyS2” seri portuna bağlı Microstrain 3DM-GX2 IMU algılayıcısı ve khepera3toolbox

Header Dosyaları: IMU\_3DM\_GX2.h

Fonksiyon Girdileri:

- float AccelBiasx : x ekseninde yazılacak cayro başlangıç değeri
- float AccelBiasy : y ekseninde yazılacak cayro başlangıç değeri
- float AccelBiasz : z ekseninde yazılacak cayro başlangıç değeri

Fonksiyon Çıktıları:

- int flag: Başarı durumunda flag=1 diğer durumlarda flag=-1 dönmektedir.

Referanslar : [42].

---

Fonksiyon : int CaptureGyroBias(float \*buffer, unsigned int samplingtime)

Açıklama:

Bu fonksiyon Microstrain 3DM-GX2 IMU algılayıcısının cayro başlangıç değerlerini okumak için kullanılmaktadır.

Çalışma Ortamı: KheperaIII Robotu

Gereksinimler : Korebot üzerinde “/dev/ttyS2” seri portuna bağlı Microstrain 3DM-GX2 IMU algılayıcısı ve khepera3toolbox

Header Dosyaları: IMU\_3DM\_GX2.h

#### Fonksiyon Girdileri:

- float \*buffer : 3DM-GX2 IMU algılayıcısından cayro başlangıç değerleri okuma işlemi bittikten sonra okunan verilerin yazılacağı adresi işaret eden float pointer.
- unsigned int samplingtime : samplingtime girdi parametresi ölçüm ve ortalama alınacak olan algılayıcının ne süre ile okunacağını belirler.

#### Fonksiyon Çıktıları:

- int flag: Başarı durumunda flag=1 diğer durumlarda flag=-1 dönmektedir.

Referanslar : [42].

---

Fonksiyon : int TransferQuantity(int quantity)

#### Açıklama:

Bu fonksiyon ile ivmeölçer ve cayro başlangıç düzeltme değerleri non-volatile olan bir hafızaya yazılmaktadır. Sisteme açılışta bu non-volatile hafızadan bu değerleri okuyarak ilgili RAM hafızasına kaydetmektedir.

Çalışma Ortamı: KheperaIII Robotu

Gereksinimler: Korebot üzerinde “/dev/ttyS2” seri portuna bağlı Microstrain 3DM-GX2 IMU algılayıcısı ve khepera3toolbox

Header Dosyaları: IMU\_3DM\_GX2.h

#### Fonksiyon Girdileri:

- int quantity : Bu değişkeninin içine yazılan sayı ivmeölçer veya cayro başlangıç değerlerinden hangisinin non-volatile hafızaya aktarılacağını belirlemektedir.

#### Fonksiyon Çıktıları:

- int flag: Başarı durumunda flag=1 diğer durumlarda flag=-1 dönmektedir.

Referanslar : [42].

---

Fonksiyon : int WriteEEPROM(char\* address, char\* data)

#### Açıklama:

Bu fonksiyon ile istenen değerler Microstrain 3DM-GX2 IMU algılayıcısının belirlenen EEPROM adresine yazılabilmektedir.

Çalışma Ortamı: KheperaIII Robotu

Gereksinimler: Korebot üzerinde “/dev/ttyS2” seri portuna bağlı Microstrain 3DM-GX2 IMU algılayıcısı ve khepera3toolbox

Header Dosyaları: IMU\_3DM\_GX2.h

#### Fonksiyon Girdileri:

- char \*address : Verinin yazılacağı EEPROM adresini işaret eden char pointer.
- char \*data : EEPROM adresine yazılacak olan veri değerini işaret eden char pointer.

#### Fonksiyon Çıktıları:

- int flag: Başarı durumunda flag=1 diğer durumlarda flag=-1 dönmektedir.

---

Fonksiyon : int ReadEEPROM(char \*address, char \*data)

#### Açıklama:

Bu fonksiyon ile Microstrain 3DM-GX2 IMU algılayıcısı belirtilen EEPROM adresindeki veri değerini vermektedir.

Çalışma Ortamı: KheperaIII Robotu

Gereksinimler: Korebot üzerinde “/dev/ttyS2” seri portuna bağlı Microstrain 3DM-GX2 IMU algılayıcısı ve khepera3toolbox

Header Dosyaları: IMU\_3DM\_GX2.h

#### Fonksiyon Girdileri:

- char \*address : Verinin yazılacağı EEPROM adresini işaret eden char pointer.
- char \*data : EEPROM adresine yazılacak olan veri değerini işaret eden char pointer.



### Fonksiyon Çıktıları:

- int flag: Başarı durumunda flag=1 diğer durumlarda flag=-1 dönmektedir.

## **D. LAZER MESAFE ALGIYACI**

Bu tez çalışmasında kullanılan Hokuyo URG-04LX lazer mesafe algılayıcısı robotik uygulamalar için uygun, küçük, hafif, yüksek performanslı ve diğer lazer mesafe algılayıcılarına göre daha düşük güç tüketimi olan bir algılayıcıdır. KheperaIII mini gezgin robotlarına bağlanabilecek kadar küçük 50x50x70 mm boyutları vardır. Algılayıcı 4 metre yarıçaplı, 240 derecelik bir alanı 0.36 derecelik çözünürlükle,  $\pm 10$  mm kesinlik ve 1 mm hassasiyet ile tarama kapasitesine sahiptir. Güç tüketiminin 5V – 500 mA olması bu algılayıcıyı pil kullanılan uygulamalar için uygun bir seçim haline getirmektedir. Lazer mesafe ölçerin güç gereksinimi Ek C.'de bahsedilen ara haberleşme kartından sağlanmaktadır.

### **D.1. Lazer Mesafe Algılayıcısı USB Bağlantısı**

KheperaIII robotu üzerinde sadece bir adet seri port olduğundan ve bu porta IMU algılayıcısı bağlı olduğundan lazer mesafe ölçer USB veri yolu ile robota bağlanmıştır. Bu amaçla robot için tasarlanmış bir web kamerası kartı USB Hub olarak değiştirilmiştir (Şekil D.1.). Bu şekilde robota birden fazla USB ile haberleşen cihazlar bağlanabilmektedir.



Şekil D.1. USB Hub bağlı KheperaIII robotu.

## D.2. Lazer Mesafe Algılayıcısı Linux Sürücüsü

Hokuyo URG-04LX algılayıcısı verilerinin robot üzerinde çalışan kontrol programları tarafından okunabilmesi için Hokuyo URG-04LX lazer algılayıcısının Linux sürücüsüne ihtiyaç duyulmaktadır. Fakat bu sürücü üretici firma tarafından sağlanmamasından dolayı Hokuyo URG-04LX lazer algılayıcısının robot gezinmelerinde kullanılabilmesi için bu tez çalışması kapsamında TOBB Ekonomi ve Teknoloji Üniversitesi Sürü Sistemler Araştırma Laboratuvarı'nda Linux sürücüsü geliştirilmiştir. Bu sürücü aracılığı ile robot üzerinde lazer algılayıcısından veri okumak için gerekli fonksiyonlar ve robota özel kütüphaneler aşağıda verilmiştir.

---

Fonksiyon : int khepera3\_urglaser\_init(char \*portname, int start\_step, int stop\_step, int cluster)

### Açıklama:

Bu fonksiyon robota bağlı bir URG-04 LX lazer mesafe algılayıcısının bağlı olduğu usb/seri portu açmak için ve lazer algılayıcısından hangi özelliklerle veri okunacağını belirtmek için kullanılmaktadır.

Çalışma Ortamı: KheperaIII Robotu

Gereksinimler: Korebot üzerinde “/dev/ttyUSB0” yada “/dev/ttyS2” portuna bağlı URG-04LX lazer algılayıcısı ve khepera3toolbox

Header Dosyaları: urglaser.h

### Fonksiyon Girdileri:

- char \*portname : Lazer algılayıcısının bağlı olduğu portu belirten cahr pointer.
- int startstep : Okunacak lazer verisinin başlangıç adımı.
- int endstep : Okunacak lazer verisinin bitiş adımı.
- int cluster : Okunacak lazer verisinin kümelenme sayısı.

### Fonksiyon Çıktıları:

- int flag: Başarı durumunda flag=0 diğer durumlarda flag=-1 dönmektedir.

---

Fonksiyon : int khepera3\_urglaser\_read(float \*distance)

### Açıklama:

Bu fonksiyon URG-04 LX lazer mesafe algılayıcından veri okumak için kullanılmaktadır.

Çalışma Ortamı: KheperaIII Robotu

Gereksinimler: Korebot üzerinde “/dev/ttyUSB0” yada “/dev/ttyS2” portuna bağlı URG-04LX lazer algılayıcısı ve khepera3toolbox

Header Dosyaları: urglaser.h

### Fonksiyon Girdileri:

- float \*distance : Lazer verisi okunadıktan sonra yazılacak olan adresi işaret eden float pointer.

### Fonksiyon Çıktıları:

- int length: Başarı durumunda length=okunan veri sayısı diğer durumlarda flag=-1 dönmektedir.

Referanslar : [43].

## **E. UDP HABERLEŞME PROTOKOLÜ**

UDP (User Datagram Protocol) haberleşme yöntemi, TCP/IP protokol takımının iki aktarım katmanı protokolünden birisidir. UDP protokolü, bilgisayar ağları arasında paketlerin değişimine imkân sağlamak için tasarlanmıştır ve verilerin gönderilmesi için karşılıklı bağlantıya ihtiyaç duyulmaz. UDP protokolü gönderilen paketin teslim edileceğini garantilemez ve bunun için genellikle paket teslim garantisi istenmeyen uygulamalarda tercih edilir. UDP bağlantı kurulum işlemlerini, akış kontrolü ve tekrar iletim işlemlerini yapmayarak veri iletim süresini en aza indirir. Geniş alan ağlarında (WAN) ses ve görüntü aktarımı gibi gerçek zamanlı veri aktarımlarında UDP kullanılır. UDP protokolünün en büyük avantajı bağlantı gerektirmediğinden

bir veri paketini aynı anda tüm ağa yayımlanabilmesidir. UDP toplu yayın-grup mesajları için son derece kullanışlıdır. UDP protokolü içerisinde sadece isteğe bağlı olarak hata kontrol mekanizması yürütülür.

UDP protokolünün toplu yayın mesajları için son derece kullanışlı olmasından dolayı bu tez çalışmasında robotlar arası haberleşme için UDP kablosuz haberleşme yöntemi kullanılmıştır. Yardımlaşmalı çok robotlu haritalamada robotlar kendi konum ve lazer verileri tüm robotlara ve ayrıca kumanda merkezini temsil eden bilgisayara göndermektedirler. UDP protokolü sayesinde robot veri paketini aynı anda ağdaki tüm üyelere gönderebilmektedir. Aynı anda tüm üyelere veri yayını yapabilmek robotları büyük bir işlem yükünden kurtarmaktadır. Eğer UDP değil de TCP/IP protokolü kullanılmış olsaydı, veri göndermek isteyen bir robot ağdaki tüm üyelere ayrı ayrı bağlanıp tekrar aynı veriyi gönderecekti. Örneğin sistemde ağa bağlı dokuz üye varsa, robot aynı veriyi dokuz defa gönderecekti. UDP protokolünde ise göndermek istediği veriyi bir defa yayınlaması yeterlidir. Bir defa yayınlanan veri ağdaki tüm üyelere aynı anda gönderilebilmektedir. Bu amaçla bu tez çalışmasında geliştirilen haberleşme fonksiyonları aşağıdaki gibidir.

---

Fonksiyon : int communication\_start()

Açıklama:

Bu fonksiyon UDP haberleşme ile robotun hem veri alma hem de veri gönderebilmesi için gerekli socket ve port ayarlarını yapmaktadır.

Çalışma Ortamı: KheperaIII Robotu

Gereksinimler: Korebot kartına bağlı kablosuz haberleşme kartı.

Header Dosyaları: khepera3\_comm.h

Fonksiyon Girdileri: Yok.

Fonksiyon Çıktıları:

- int flag: Başarı durumunda flag=0 diğer durumlarda flag=-1 dönmektedir.

---

Fonksiyon : int receivedata(void\* buffer, int len)

### Açıklama:

Bu fonksiyon robot üzerinde UDP haberleşme yöntemi ile veri alınmasını sağlamaktadır

Çalışma Ortamı: KheperaIII Robotu

Gereksinimler: Korebot kartına bağlı kablosuz haberleşme kartı.

Header Dosyaları: khepera3\_comm.h

### Fonksiyon Girdileri:

- void\* buffer: Gelen veri paketinin yazılacağı adresi gösteren void pointer. Genellikle bir struct yapısı olarak kullanılmaktadır.
- int len: Kaç byte veri okunacağını gösteren değişken.

### Fonksiyon Çıktıları:

int bytes\_read: Başarı durumunda bytes\_read = okunan byte sayısı diğer durumlarda flag=-1 dönmektedir.

---

Fonksiyon : int receivedata(void\* buffer, int len)

### Açıklama:

Bu fonksiyon robot üzerinde UDP haberleşme yöntemi ile veri alınmasını sağlamaktadır

Çalışma Ortamı: KheperaIII Robotu

Gereksinimler: Korebot kartına bağlı kablosuz haberleşme kartı.

Header Dosyaları: khepera3\_comm.h

### Fonksiyon Girdileri:

- void\* buffer: Gelen veri paketinin yazılacağı adresi gösteren void pointer. Genellikle bir struct yapısı olarak kullanılmaktadır.
- int len: Kaç byte veri okunacağını gösteren değişken.

### Fonksiyon Çıktıları:

int bytes\_read: Başarı durumunda bytes\_read=okunan byte sayısı diğer durumlarda bytes\_read =-1 dönmektedir.

## KAYNAKLAR

- [1] Byrne, R.H., Klärer, P.R., Pletta, J.B., Techniques for autonomous navigation, Sandia Report SAND92-0457, Sandia National Laboratories, Albuquerque, NM, Mart 1992.
- [2] Chénavier, F., Crowley, J., Position estimation for a mobile robot using vision and odometry, Proceedings of IEEE International Conference on Robotics and Automation, 2588-2593, Nice, Fransa, 12-14 Mayıs 1992.
- [3] Evans, J. M., HelpMate: An autonomous mobile robot courier for hospitals, International Conference on Intelligent Robots and Systems (IROS '94), 1695-1700, Münih, Almanya, 12-16 Eylül 1994.
- [4] Borenstein, J., Feng. L., Correction of systematic dead-reckoning errors in mobile robots, International Conference on Intelligent Robots and Systems (IROS '95), 569-574, Pittsburgh, Pennsylvania, 5-9 Ağustos 1995.
- [5] Borenstein, J., Feng. L., UMBmark: A benchmark test for measuring dead-reckoning errors in mobile robots, SPIE Conference on Mobile Robots, Philadelphia, 22-26 Ekim 1995.
- [6] King, S., Weiman, C., Helpmate autonomous mobile robot navigation system, SPIE Conference on Mobile Robots, 1388, 190-198, Boston, MA, ABD, Kasım 1990.w
- [7] Simmons, R., Where in the world is xavier, the robot?, Machine Perception, 5, 1, 1996.
- [8] Burgard, W., Cremers, A.B., Fox, D., Hahnel, D., Lakemeyer, G., Schulz, D., Steiner, W., Thrun, S., Experiences with an interactive museum tour-guide robot, Artificial Intelligence, 114(1-2), 3-55, Ekim 1999.
- [9] Horswill, I., Specialization of perceptual processes, Technical Report AI TR-1511, MIT, AI LAB, Cambridge, MA, Eylül 1994.
- [10] Nourbakhsh, I., Boenage, J., Grange, S., Lutz, R., Meyer, R., Soto, A. An affective mobile robot with a full-time job, Artificial Intelligence, 114(1-2), 95-124, 1999.

- [11] Thrun, S., Beetz, M., Bennewitz, M., Burgard, W., Cremers, A.B., Dellaert, F., Fox, D., Hahnel, D., Rosenberg, C., Roy, N., Schulte, J., Schulz, D., Probabilistic algorithms and the interactive museum tour-guide robot minerve, *International Journal of Robotics Research*, 19(11), 972-999, 2000.
- [12] Chatila, R., Laumond, J. P., Position referencing and consistent world modeling for mobile robots, *IEEE International Conference on Robotics and Automation (ICRA)*, 2, 138-145, Mart 1985.
- [13] Leonard, J. J., Durrant-Whyte, H. F., Cox, I. J., Dynamic map building for an autonomous mobile robot, *International Journal of Robotic Search*, 11(4), 89-96, 1992.
- [14] Rencken, W. D., Concurrent localization and map building for mobile robots using ultrasonic sensors, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2129-2197, Yokohama, Japonya, Temmuz 1993.
- [15] Thrun, S., Fox, D., Burgard, W., A probabilistic approach to concurrent mapping and localization for mobile robots, *Machine Learning*, 31, 29-53, 1998.
- [16] Elfes, A., Occupancy grids: A probabilistic framework for perception and navigation, *Doktora Tezi*, Carnegie Mellon Üniversitesi, Elektrik ve Bilgisayar Mühendisliği, 1989.
- [17] Lu, F., Milios, E., Globally consistent range scan alignment for environment mapping, *Autonomous Robots*, 4, 333-349, 1997.
- [18] Moravec, H. P., Sensor fusion in certainty grids for mobile robots, *AI Magazine*, 61-74, 1988.
- [19] Fenwick, J. W., Newman, P. M., Leonard, J. J., Cooperative concurrent mapping and localization, *IEEE International Conference Robotics Automation (ICRA)*, 2, 1810-1817, 2002.
- [20] Konolige, K., Fox, D., Ortiz, C., Agno, A., Eriksen, M., Limketkai, B., Ko, J., Morisset, B., Schulz, D., Steward, B., Vincent, R., Centibots: Very large scale distributed robotic teams, *Experimental Robotics IX*, 21, 131-140, Mart 2006.
- [21] Thrun, S., Probabilistic online mapping algorithm for teams of mobile robots, *International Journal of Robotic Research*, 20(5), 335-363, 2001.

- [22] Burgard, W., Fox, D., Moors, M., Simmons, R., Thrun, S., Collaborative multi-robot exploration, IEEE International Conference on Robotics and Automation (ICRA), 1, 476-481, San Francisco, CA, ABD, 2000.
- [23] Simmons, R., Apfelbaum, D., Burgard, W., Fox, M., an Moors, D., Thrun, S., Younes, H., Coordination for multi-robot exploration and mapping, AAAI National Conference on Artificial Intelligence, 852-858, Austin, TX, ABD 2000.
- [24] Klarer, P.R., Simple 2-D navigation for wheeled vehicles, Sandia Report SAND88- 0540, Sandia National Laboratories, Albuquerque, NM, Nisan 1988.
- [25] Crowley, J.L., Reignier, P., Asynchronous control of rotation and translation for a robot vehicle, Robotics and Autonomous Systems, 10, 243-251, 1992.
- [26] Feng, L, Koren, Y., Borenstein, J., A cross-coupling motion controller for mobile robots, IEEE Journal of Control Systems, 25-43, Aralık 1993.
- [27] Barshan, B., Durrant-Whyte, H.F., Inertial Navigation Systems for Mobile Robots, IEEE Transactions on Robotics and Automation, 11, 328-342, Haziran 1995.
- [28] Murata Erie North America, 2200 Lake Park Drive, Smyrna, GA 30080.
- [29] Avionics, Kent, U.K.
- [30] MicroStrain, Inc. 459 Hurricane Lane, Suite 102 Williston, VT 05495 USA.
- [31] Fox, D., diğerleri, Monte-carlo localization: Efficient position estimation for mobile robots, National Conference on Artificial Intelligence (IAAA), Orlando, ABD, 1999.
- [32] Thrun, S., diğerleri, Monte-carlo localization with mixture proposal distribution, AAAI National Conference on Artificial Intelligence (AAAI), 2000.
- [33] Fox, D., Burgard, W., Thrun, S. Markov localization for mobile robots in dynamic environments, Journal of Artificial Intelligence Research (JAIR), 11, 391- 457, 1999.
- [34] Choset, H., diğerleri, Principles of robot motion: Theory, AI Algorithms and Implementation, *MIT Press*, 55 Hayward Street, Cambridge, 2005.
- [35] Borenstein, J., Everett, H. R., Feng, L. Navigating mobile robots: Sytems and techniques, Number ISBN 1-56881-058-X. A K Peters, Wellesley, MA, 1996.



- [36] Johann, Borenstein, Liqiang Feng, Measurement and correction of systematic odometry errors in mobile robots, *IEEE Transactions on Robotics & Automation*, 12(6), 869-880, Aralık 1996.
- [37] Rekleitis, I. M., Cooperative localization and multi - robot exploration, *Doktora Tezi*, McGill Üniversitesi, Bilgisayar Bilimleri, Montreal, Quebec, Kanada, Şubat 2003.
- [38] Liu, J. S., Chen, R., Logvinenko, T., A theoretical framework for sequential importance sampling and resampling, In A. Doucet, N. de Freitas, and N. J. Gordon, editörleri, *Sequential monte-carlo in practice*, Springer-Verlag, Ocak 2001.
- [39] Gazi, V., Passino, K.M., A class of attractions/repulsion functions for stable swarm aggregations, *International Journal of Control*, 77(18), 1567-1579, 2004.
- [40] Gazi, V., Fidan, B., Hanay, Y. S., Köksal, M. I., Aggregation, foraging, and formation control of swarms with non-holonomic agents using potential functions and sliding mode techniques, *Turkish Journal of Electric*, 15(2), 149-168, 2007.
- [41] Çayırpunar, Ö., Çoklu robot sistemlerinde robotlar arası haberleşme ve iş birliği kullanılarak arama verimliliğinin artırılması, *Yüksek Lisans Tezi*, TOBB Ekonomi ve Teknoloji Üniversitesi, Bilgisayar Mühendisliği, Eylül 2009, Ankara, Türkiye.
- [42] "Inertia Link 3DM-GX2 data communications protocol" erişim adresi: <http://www.microstrain.com/pdf/dcp/Inertia-Link-3DM-GX2-data-communications-protocol.pdf>, erişim tarihi: 08 Mart 2010.
- [43] "URG-04LX Lazer Range Finder Specifications" erişim adresi: <http://www.robotstorehk.com/sensors/doc/HOKUYO-LASER1s.pdf>, erişim tarihi: 08 Mart 2010.
- [44] Choset, H., Sensor based motion planning: The hierarchical generalized voronoi graph, *Doktora Tezi*, California Institute of Technology, 1996.
- [45] Borenstein, J., Everett, B., Feng, L., *Navigation mobile robots: Systems and Techniques*, A. K. Peters Ltd., Wellesley, MA, 1996.

- [46] Mataric, M. J., Interaction and intelligent behavior, Technical report AI-TR-1495, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, Cambridge, MA, 1994.
- [47] Kuipers, B., Byun, Y. T., A robot exploration and mapping strategy based on a semantic hierarch of spatial representations, Journal of Robotics and Autonomous Systems, 8, 47-63, 1991.
- [48] Şamilođlu, A., Çayırpunar, Ö., Gazi, V. ve Koku, A. B., Çoklu robot uygulamaları için bir deney düzeneđi, Türkiye Otomatik Kontrol UlusalToplantısı, Kasım 2008.

## ÖZGEÇMİŞ

### Kişisel Bilgiler

Soyadı, adı : ATAŞ, Yunus  
Uyruğu : T.C.  
Doğum tarihi ve yeri : 20.10.1984, Ankara  
Medeni hali : Bekâr  
Telefon : 0 (312) 292 42 91  
Faks : 0 (312) 292 42 80  
e-mail : yatas@etu.edu.tr

### Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Lisans	Anadolu Üniversitesi İşletme Bölümü	2008
Lisans	Eskişehir Osmangazi Üniversitesi Elektrik ve Elektronik Mühendisliği	2007

### İş Deneyimi

Yıl	Yer	Görev
09/2007 - 01/2010	TOBB ETÜ	Araştırma Görevlisi
02/2007 - 08/2007	Robotel Ltd. Şti.	Ar-Ge Mühendisi

### Yabancı Dil

İngilizce

### Yayınlar

- Ataş, Y., Çayırpunar, Ö., Akat, S. B., Gazi, V. ve Alboul, L., Kapalı Alanlar İçin Lazer Algılayıcı Tabanlı Yardımlaşmalı Çok- Robotlu Haritalama, Türkiye Otomatik Kontrol Ulusal Toplantısı, İstanbul, Türkiye, Ekim 2009.

- Ataş, Y., Çayırpunar, Ö., Akat, S. B., Gazi, V. ve Alboul, L., Laser Based Cooperative Multi-Robot Map Building for Indoor Environments, EURON/IARP International Workshop on Robotics for Risky Interventions and Surveillance of the Environment, Brüksel, Belçika, Ocak 2009.