



**ULTRA GENİŞ BANT SENSÖRÜ KULLANILARAK KAPALI ALANDA
NESNE KONUMLANDIRMA**

GÖKHAN KARAÇOBAN

**YÜKSEK LİSANS TEZİ
SAVUNMA TEKNOLOJİLERİ ANA BİLİM DALI**

**SİVAS BİLİM VE TEKNOLOJİ ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

KASIM 2023

GÖKHAN KARAÇOBAN tarafından hazırlanan “ULTRA GENİŞ BANT SENSÖRÜ KULLANILARAK KAPALI ALANDA NESNE KONUMLANDIRMA” adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Sivas Bilim ve Teknoloji Üniversitesi Savunma Teknolojileri Ana Bilim Dalında Yüksek Lisans Tezi olarak kabul edilmiştir.

Danışman: Prof. Dr. Murat EFE

Kontrol ve Kumanda Sistemleri Ana Bilim Dalı, Ankara Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum.

Başkan: Doç. Dr. Fatih KABURCUK

Elektromanyetik Alanlar ve Mikrodalga Tekniği Ana Bilim Dalı, Cumhuriyet Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans olduğunu onaylıyorum.

Üye: Prof. Dr. Murat EFE

Kontrol ve Kumanda Sistemleri Ana Bilim Dalı, Ankara Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans olduğunu onaylıyorum.

Üye: Dr. Öğr. Üyesi Nurbanu GÜZEY

Telekomünikasyon Ana Bilim Dalı, Sivas Bilim ve Teknoloji Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans olduğunu onaylıyorum.

Tez Savunma Tarihi: 07/11/2023

Jüri tarafından kabul edilen bu çalışmanın YÜKSEK LİSANS TEZİ olması için gerekli şartları yerine getirdiğini onaylıyorum

.....

Prof. Dr. Emre BİÇER

Lisansüstü Eğitim Enstitüsü Müdürü

ETİK BEYAN

Sivas Bilim ve Teknoloji Üniversitesi Lisansüstü Eğitim Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

.....
GÖKHAN KARAÇOBAN
07/11/2023

ULTRA GENİŞ BANT SENSÖRÜ KULLANILARAK KAPALI ALANDA NESNE KONUMLANDIRMA

(YÜKSEK LİSANS TEZİ)

GÖKHAN KARAÇOBAN

SİVAS BİLİM ve TEKNOLOJİ ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

Kasım 2023

ÖZET

Günümüzde, kapalı alan konumlandırma sistemleri hem sivil hem de askeri birçok alanda kullanılmakta olup önemli bir çalışma alanı haline gelmiştir. Kapalı alanlarda nesne konumlandırmak için küresel konumlandırma sistemleri kullanılamadığından dolayı, kapalı alanlarda konum belirleme problemlerinin çözümü için, farklı konumlandırma sistemlerine ihtiyaç duyulmaktadır. Bu tez çalışmasında, ultra geniş bant sensörü kullanılarak kapalı alanda nesne konumlandırma probleminde, genişletilmiş Kalman Filtre algoritmasının başarımlarının belirlenmesi ile farklı sensör geometrilerinin ve sensör sayısının, genişletilmiş Kalman filtre algoritmasının kestirim performansı üzerindeki etkilerinin araştırılması amaçlanmıştır. Bu amaçla farklı sensör geometrileri oluşturulup, genişletilmiş Kalman filtre algoritmasında bu geometrilerin kullanılmasıyla, nesnenin hareketi esnasında bulunduğu konumlara ait kestirimler ve kestirim hataları elde edilmiştir. Elde edilen sonuçlar incelendiğinde, sensör geometrisi değiştirildiğinde, genişletilmiş Kalman filtre algoritmasının kestirim hatalarının da değiştiği görülmüştür. Dolayısıyla, sensör geometrisinin, genişletilmiş Kalman filtre algoritmasının konum kestirim performansını etkilediği gözlemlenmiş ve bu nedenle, genişletilmiş Kalman filtre algoritması kullanılarak, düşük hatalarla konum kestirimi elde edilebilmesi için, sensör geometrisinin optimum bir şekilde tasarlanması gerektiği sonucuna ulaşılmıştır. Ayrıca, sensör sayısının, genişletilmiş Kalman filtre algoritmasının kestirim hatası üzerindeki etkilerinin incelenmesi için, tez çalışması kapsamında tasarlanan farklı sensör geometrilerinin her birisinde, kullanılan sensör sayıları artırılarak, genişletilmiş Kalman filtre algoritmasının konum kestirim hataları elde edilmiştir. Sensör sayısı artırıldığında, genişletilmiş Kalman filtre algoritmasının konum kestirim hatasının azalmasının sensör geometrisine bağlı olduğu sonucuna ulaşılmıştır.

Bilim Kodu : 90524

Anahtar Kelimeler : Genişletilmiş Kalman filtresi, ultra geniş bant sensörü, kapalı alan konum belirleme, sensör geometrisi

Sayfa Adedi : 209

Danışman : Prof. Dr. Murat EFE

INDOOR OBJECT POSITIONING USING ULTRA WIDE BAND SENSOR

(M.Sc. THESIS)

GÖKHAN KARAÇOBAN

SIVAS UNIVERSITY OF SCIENCE AND TECHNOLOGY

INSTITUTE OF GRADUATE STUDIES

November 2023

ABSTRACT

Today, indoor positioning systems are used in many areas, both civil and military and they become an important study field. Since global positioning systems cannot be used to locate objects indoors, different positioning systems are needed to solve indoor positioning problems. In this thesis, it is aimed to determine the performance limits of the EKF algorithm and to investigate the effects of different sensor geometries and sensor numbers on the estimation performance of the EKF algorithm in indoor object positioning problem using an ultra wideband sensor. Therefore, position estimation and estimation errors of the EKF algorithm were obtained by using different sensor geometries. When the results were examined, it was seen that the estimation errors have changed when sensor geometry was changed. Therefore, sensor geometry affects the estimation performance of the EKF algorithm. Consequently, sensor geometry should be designed optimally to obtain position estimation with low errors. Also, to examine the effects of the sensor numbers on position estimation performance of the EKF algorithm, estimation errors of the EKF algorithm were obtained by increasing the sensor numbers. It was concluded that when the number of sensors is increased, the decrease in the estimation error of the EKF algorithm depends on the sensor geometry. |

Science Code : 90524

Key Words : Extended Kalman filter, UWB sensor, indoor positioning, sensor geometry

Page Number : 209

Supervisor : Prof. Dr. Murat EFE

TEŞEKKÜR

Tez çalışmasının her aşamasında bilgi, tecrübe, nasihat ve yardımlarıyla hem tez çalışmasında hem de akademik ortamda bana yol gösteren, destek olan ve gelişmeme çok büyük katkılarda bulunan kıymetli danışman hocam Sayın Prof. Dr. Murat EFE'ye, en kalbi ve içten duygularıyla teşekkür ederim. Ayrıca, Aselsan'dan sanayi danışmanlığımı yürüten Sayın Dr. Murat EREN'e çok teşekkür ederim.

Tez çalışması sırasında karşılaştığım problemlerin çözümünde bana her zaman destek olan ve yardımlarını esirgemeyen Sayın Dr. Öğr. Üyesi Gökhan SOYSAL ve Öğr. Gör. Yetkin ERSOY hocalarıma en kalbi duygularıyla teşekkür ederim.

Tez çalışması sürecinde her türlü desteklerini esirgemeyen aileme çok teşekkür ederim. |

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT	v
ÇİZELGELERİN LİSTESİ.....	xi
ŞEKİLLERİN LİSTESİ.....	xiv
1. GİRİŞ.....	1
2. KAYNAK ARAŞTIRMASI	11
3. MATERYAL ve YÖNTEM.....	53
3.1. Kinematik Modeller.....	53
3.1.1. Koordineli dönüş hareket modeli.....	53
3.1.2. Yaklaşık sabit hızlı düzgün doğrusal hareket modeli	54
3.2. Ultra Geniş Bant Sensörünün Özellikleri ve Ölçüm Modeli	55
3.3. Genişletilmiş Kalman Filtre Algoritması.....	57
3.4. Farklı Sensör Geometrisi Senaryoları.....	65
3.4.1. Genişletilmiş Kalman filtresinin süreç gürültüsü varyansının ve sensörün ölçüm gürültü varyansının genişletilmiş Kalman filtre algoritmasının kestirim hatası üzerindeki etkilerinin incelenmesi için kullanılan senaryo	66
3.4.2. Tasarlanan farklı sensör geometrileri.....	67
3.5. Sensör Sayısının Genişletilmiş Kalman Filtre Algoritmasının Kestirim Hatası Üzerindeki Etkilerinin İncelenmesi İçin Kullanılan Senaryolar	77
3.5.1. Birinci sensör geometrisinde kullanılan sensör sayılarının arttırılmasıyla tasarlanan sensör geometrileri.....	77
3.5.2. İkinci sensör geometrisinde kullanılan sensör sayılarının arttırılmasıyla tasarlanan sensör geometrileri.....	84
3.5.3. Üçüncü sensör geometrisinde kullanılan sensör sayılarının arttırılmasıyla tasarlanan sensör geometrileri.....	90

3.5.4. Dördüncü sensör geometrisinde kullanılan sensör sayılarının arttırılmasıyla tasarlanan sensör geometrileri.....	96
3.5.5. Beşinci sensör geometrisinde kullanılan sensör sayılarının arttırılmasıyla tasarlanan sensör geometrileri.....	103
3.5.6. Altıncı sensör geometrisinde kullanılan sensör sayılarının arttırılmasıyla tasarlanan sensör geometrileri.....	109
3.5.3. 10 adet UGB sensör kullanılarak tasarlanan sensör geometrileri	110
4. BULGULAR ve TARTIŞMA.....	117
4.1. Genişletilmiş Kalman Filtre Algoritmasının Süreç Gürültüsü Varyansının Kestirim Hatası Üzerindeki Etkilerinin İncelenmesi.....	117
4.1.1. UGB sensörünün ölçüm gürültüsü varyansı $0,5 \text{ m}^2$ iken GKF algoritmasının farklı süreç gürültüsü varyans değerleri için elde edilen konum kestirim hataları	119
4.1.2. UGB sensörünün ölçüm gürültüsü varyansı 1 m^2 iken GKF algoritmasının farklı süreç gürültüsü varyans değerleri için elde edilen konum kestirim hataları	120
4.1.3. UGB sensörünün ölçüm gürültüsü varyansı 5 m^2 iken GKF algoritmasının farklı süreç gürültüsü varyans değerleri için elde edilen konum kestirim hataları	121
4.1.4. UGB sensörünün ölçüm gürültüsü varyansı 15 m^2 iken GKF algoritmasının farklı süreç gürültüsü varyans değerleri için elde edilen konum kestirim hataları	122
4.2. UGB Sensörünün Ölçüm Gürültüsü Varyansının Genişletilmiş Kalman Filtre Algoritmasının Konum Kestirim Hatası Üzerindeki Etkilerinin İncelenmesi	125
4.2.1. GKF algoritmasının süreç gürültüsü varyansı $1 \left(\frac{m}{sn^2}\right)^2$ iken elde edilen konum kestirim hataları.....	125
4.2.2. GKF algoritmasının süreç gürültüsü varyansı $10 \left(\frac{m}{sn^2}\right)^2$ iken elde edilen konum kestirim hataları.....	126
4.2.3. GKF algoritmasının süreç gürültüsü varyansı $50 \left(\frac{m}{sn^2}\right)^2$ iken elde edilen konum kestirim hataları.....	126
4.3. Sensör Geometrisinin Genişletilmiş Kalman Filtre Algoritmasının Konum Kestirim Hatası Üzerindeki Etkilerinin İncelenmesi.....	129
4.3.1. Birinci sensör geometrisi kullanıldığında elde edilen genişletilmiş Kalman filtre algoritmasının konum kestirim hatası sonuçları.....	133

4.3.2. İkinci sensör geometrisi kullanıldığında elde edilen genişletilmiş Kalman filtre algoritmasının konum kestirim hatası sonuçları.....	135
4.3.3. Üçüncü sensör geometrisi kullanıldığında elde edilen genişletilmiş Kalman filtre algoritmasının konum kestirim hatası sonuçları.....	137
4.3.4. Dördüncü sensör geometrisi kullanıldığında elde edilen genişletilmiş Kalman filtre algoritmasının konum kestirim hatası sonuçları.....	139
4.3.5. Beşinci sensör geometrisi kullanıldığında elde edilen genişletilmiş Kalman filtre algoritmasının konum kestirim hatası sonuçları.....	141
4.3.6. Altıncı sensör geometrisi kullanıldığında elde edilen genişletilmiş Kalman filtre algoritmasının konum kestirim hatası sonuçları.....	144
4.4. Sensör Sayısının Genişletilmiş Kalman Filtre Algoritmasının Konum Kestirim Hatası Üzerindeki Etkilerinin İncelenmesi.....	147
4.4.1. Birinci sensör geometrisinde kullanılan sensör sayıları arttırıldığında elde edilen genişletilmiş Kalman filtre algoritmasının konum kestirim hatası sonuçları.....	147
4.4.2. İkinci sensör geometrisinde kullanılan sensör sayıları arttırıldığında elde edilen genişletilmiş Kalman filtre algoritmasının konum kestirim hatası sonuçları.....	148
4.4.3. Üçüncü sensör geometrisinde kullanılan sensör sayıları arttırıldığında elde edilen genişletilmiş Kalman filtre algoritmasının konum kestirim hatası sonuçları.....	150
4.4.4. Dördüncü sensör geometrisinde kullanılan sensör sayıları arttırıldığında elde edilen genişletilmiş Kalman filtre algoritmasının kestirim hatası sonuçları.....	151
4.4.5. Beşinci sensör geometrisinde kullanılan sensör sayıları arttırıldığında elde edilen genişletilmiş Kalman filtre algoritmasının kestirim hatası sonuçları.....	153
4.4.6. Altıncı sensör geometrisinde kullanılan sensör sayıları arttırıldığında elde edilen genişletilmiş Kalman filtre algoritmasının kestirim hatası sonuçları.....	154
4.5. Farklı Ölçüm Gürültüsü Varyans Değerleri Kullanıldığında Sensör Sayısının Genişletilmiş Kalman Filtre Algoritmasının Kestirim Hatası Üzerindeki Etkilerinin İncelenmesi.....	156
4.5.1. GKF algoritmasında birinci sensör geometrisi kullanıldığında elde edilen sonuçlar.....	157
4.5.2. GKF algoritmasında ikinci sensör geometrisi kullanıldığında elde edilen sonuçlar.....	163

4.5.3. GKF algoritmasında üçüncü sensör geometrisi kullanıldığında elde edilen sonuçlar.....	169
4.5.4. GKF algoritmasında dördüncü sensör geometrisi kullanıldığında elde edilen sonuçlar.....	175
4.5.5. GKF algoritmasında beşinci sensör geometrisi kullanıldığında elde edilen sonuçlar.....	181
4.5.6. GKF algoritmasında altıncı sensör geometrisi kullanıldığında elde edilen sonuçlar.....	187
4.6. Nesne Hareketinin Genişletilmiş Kalman Filtre Algoritmasının Kestirim Hatası Üzerindeki Etkilerinin İncelenmesi.....	193
4.6.1. Nesne karesel hareket yaptığı ve 1. sensör geometrisi kullanıldığında, GKF algoritmasının konum kestirim hatası.....	193
4.6.2. Nesne karesel hareket yaptığı ve 5. sensör geometrisi kullanıldığında, GKF algoritmasının konum kestirim hatası.....	197
4.6.3. Nesne karesel hareket yaptığı ve 6. sensör geometrisi kullanıldığında, GKF algoritmasının konum kestirim hatası.....	199
5. SONUÇ VE ÖNERİLER	203
KAYNAKLAR.....	205

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 4.1. Genişletilmiş Kalman filtresinin farklı süreç gürültüsü varyansı değerleri ve UGB sensörünün farklı ölçüm gürültüsü varyans değerleri kullanıldığında, 100 Monte Carlo simülasyonu yapılarak elde edilen, ortalama KOK konum kestirim hataları	124
Çizelge 4.2. Genişletilmiş Kalman filtresinin farklı süreç gürültüsü varyans değerleri ve farklı ölçüm gürültüsü varyans değerleri için elde edilen ortalama KOK konum kestirim hataları	127
Çizelge 4.3. GKF algoritmasının, 1. sensör geometrisinde kullanılan sensör sayıları artırıldığında elde edilen KOK konum kestirim hataları	147
Çizelge 4.4. GKF algoritmasının, 2. sensör geometrisinde kullanılan sensör sayıları artırıldığında elde edilen ortalama KOK konum kestirim hataları	150
Çizelge 4.5. GKF algoritmasının, 3. sensör geometrisinde kullanılan sensör sayıları artırıldığında elde edilen ortalama KOK konum kestirim hataları	151
Çizelge 4.6. GKF algoritmasının, 4. sensör geometrisinde kullanılan sensör sayıları artırıldığında elde edilen ortalama KOK konum kestirim hataları	153
Çizelge 4.7. GKF algoritmasının, 5. sensör geometrisinde kullanılan sensör sayıları artırıldığında elde edilen ortalama KOK konum kestirim hataları	154
Çizelge 4.8. GKF algoritmasının, 6. sensör geometrisinde kullanılan sensör sayıları artırıldığında elde edilen ortalama KOK konum kestirim hataları	156
Çizelge 4.9. Ölçüm gürültüsü varyans değerleri karışık sırayla güncellendiğinde, her bir sensör için kullanılan ölçüm gürültüsü varyans değerleri	157
Çizelge 4.10. GKF algoritmasının, 1. sensör geometrisinde sensör sayısı artırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansı büyükten küçüğe doğru sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları	159
Çizelge 4.11. GKF algoritmasının, 1. sensör geometrisinde sensör sayısı artırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları karışık sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları	161
Çizelge 4.12. GKF algoritmasının, 1. sensör geometrisinde sensör sayısı artırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansı küçükten büyüğe doğru sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları	163

Çizelge**Sayfa**

Çizelge 4.13. GKF algoritmasının, 2. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansı büyükten küçüğe doğru sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları	165
Çizelge 4.14. GKF algoritmasının, 2. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansı karışık sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları	167
Çizelge 4.15. GKF algoritmasının, 2. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları	169
Çizelge 4.16. GKF algoritmasının, 3. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları	171
Çizelge 4.17. GKF algoritmasının, 3. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları karışık sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları	173
Çizelge 4.18. GKF algoritmasının, 3. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları	175
Çizelge 4.19. GKF algoritmasının, 4. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları	177
Çizelge 4.20. GKF algoritmasının, 4. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları karışık sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları	179
Çizelge 4.21. GKF algoritmasının, 4. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları	181
Çizelge 4.22. GKF algoritmasının, 5. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları	183
Çizelge 4.23. GKF algoritmasının, 5. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları karışık sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları	185
Çizelge 4.24. GKF algoritmasının, 5. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları	187

Çizelge	Sayfa
Çizelge 4.25. GKF algoritmasının, 6. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları	189
Çizelge 4.26. GKF algoritmasının, 6. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları karışık sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları	191
Çizelge 4.27. GKF algoritmasının, 6. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları	193



ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 1.1. Multilaterasyon yönteminin gösterimi.....	6
Şekil 1.2. İki sensör kullanıldığı durumda trilaterasyon yönteminin gösterimi.....	9
Şekil 1.3. 3 sensör kullanıldığı durumda trilaterasyon yönteminin gösterimi	9
Şekil 2.1. Trilaterasyon algoritmasının gösterimi.....	31
Şekil 2.2. Trilaterasyon algoritmasında kullanılan çember geometrisinde, çemberlerin tek bir noktada kesişmeme problemini gösterimi	32
Şekil 2.3. Multilaterasyon algoritmasının gösterimi.....	34
Şekil 2.4. Üçgenleme algoritmasının gösterimi	37
Şekil 2.5. Parmak izi algoritmasının gösterimi	38
Şekil 2.6. İki boyutlu Gauss dağılımına sahip rasgele değişkenin doğrusal olmayan dönüşümü sonucu ortalama ve kovaryans değerlerinin gerçek değerleri ile GKF ve kokusuz dönüşüm sonucu elde edilen değerleri	44
Şekil 3.1. Genişletilmiş Kalman filtre algoritmasının blok şeması	59
Şekil 3.2. İki boyutlu Gauss dağılımına sahip rasgele değişkenin doğrusal olmayan dönüşümü sonucu ortalama ve kovaryans değerlerinin gerçek değerleri ile GKF algoritması kullanılarak ve kokusuz dönüşüm sonucu elde edilen değerleri....	62
Şekil 3.3. Hedef nesnenin hareket modeli ve UGB sensörlerinin konumları	66
Şekil 3.4. Birinci sensör geometrisi	68
Şekil 3.5. Birinci sensör geometrisinde 1. sensörün yeri değiştirilerek elde edilen yeni sensör geometrisi	68
Şekil 3.6. Birinci sensör geometrisinde 2. sensörün yeri değiştirilerek elde edilen yeni sensör geometrisi	69
Şekil 3.7. İkinci sensör geometrisi	69
Şekil 3.8. İkinci sensör geometrisinde 1. sensörün yeri değiştirilerek elde edilen yeni sensör geometrisi	70
Şekil 3.9. İkinci sensör geometrisinde 2. sensörün yeri değiştirilerek elde edilen yeni sensör geometrisi.....	70

Şekil	Sayfa
Şekil 3.10. Üçüncü sensör geometrisi.....	71
Şekil 3.11.Üçüncü sensör geometrisinde, 1. sensörün yeri değiştirilerek elde edilen yeni sensör geometrisi.....	71
Şekil 3.12.Üçüncü sensör geometrisinde, 2. sensörün yeri değiştirilerek elde edilen yeni sensör geometrisi.....	72
Şekil 3.13. Dördüncü sensör geometrisi	72
Şekil 3.14. Dördüncü sensör geometrisinde, 2. sensörün yeri değiştirilerek elde edilen yeni sensör geometrisi.....	73
Şekil 3.15. Dördüncü sensör geometrisinde, 1. sensörün yeri değiştirilerek elde edilen yeni sensör geometrisi.....	73
Şekil 3.16. Beşinci sensör geometrisi	74
Şekil 3.17. Beşinci sensör geometrisinde, 1. ve 3. sensörlerin yerleri değiştirilerek elde edilen yeni sensör geometrisi.....	74
Şekil 3.18. Beşinci sensör geometrisinde, 4. ve 5. sensörlerin yerleri değiştirilerek elde edilen yeni sensör geometrisi.....	75
Şekil 3.19. Beşinci sensör geometrisinde, 1. sensör, 5. ve 6. sensörlerin ortasına taşındığında elde edilen yeni sensör geometrisi	75
Şekil 3.20. Altıncı sensör geometrisi	76
Şekil 3.21. Altıncı sensör geometrisinde, 1. sensör, 6. sensöre yaklaştırılarak elde edilen yeni sensör geometrisi.....	76
Şekil 3.22. Altıncı sensör geometrisinde, 3. ve 5. sensörlerin yerleri değiştirilerek elde edilen yeni sensör geometrisi.....	77
Şekil 3.23. 6 adet sensör kullanılarak tasarlanan 1. sensör geometrisi	78
Şekil 3.24. 8 adet sensör kullanılarak tasarlanan 1. sensör geometrisi	78
Şekil 3.25. 10 adet sensör kullanılarak tasarlanan 1. sensör geometrisi	79
Şekil 3.26. 12 adet sensör kullanılarak tasarlanan 1. sensör geometrisi	79
Şekil 3.27. 16 adet sensör kullanılarak tasarlanan 1. sensör geometrisi	80
Şekil 3.28. 20 adet sensör kullanılarak tasarlanan 1. sensör geometrisi	80

Şekil	Sayfa
Şekil 3.29. 25 adet sensör kullanılarak tasarlanan 1. sensör geometrisi	81
Şekil 3.30. 30 adet sensör kullanılarak tasarlanan 1. sensör geometrisi	81
Şekil 3.31. 35 adet sensör kullanılarak tasarlanan 1. sensör geometrisi	82
Şekil 3.32. 40 adet sensör kullanılarak tasarlanan 1. sensör geometrisi	82
Şekil 3.33. 45 adet sensör kullanılarak tasarlanan 1. sensör geometrisi	83
Şekil 3.34. 50 adet sensör kullanılarak tasarlanan 1. sensör geometrisi	83
Şekil 3.35. 6 adet sensör kullanılarak tasarlanan 2. sensör geometrisi	84
Şekil 3.36. 8 adet sensör kullanılarak tasarlanan 2. sensör geometrisi	85
Şekil 3.37. 10 adet sensör kullanılarak tasarlanan 2. sensör geometrisi	85
Şekil 3.38. 12 adet sensör kullanılarak tasarlanan 2. sensör geometrisi	86
Şekil 3.39. 16 adet sensör kullanılarak tasarlanan 2. sensör geometrisi	86
Şekil 3.40. 20 adet sensör kullanılarak tasarlanan 2. sensör geometrisi	87
Şekil 3.41. 25 adet sensör kullanılarak tasarlanan 2. sensör geometrisi	87
Şekil 3.42. 30 adet sensör kullanılarak tasarlanan 2. sensör geometrisi	88
Şekil 3.43. 35 adet sensör kullanılarak tasarlanan 2. sensör geometrisi	88
Şekil 3.44. 40 adet sensör kullanılarak tasarlanan 2. sensör geometrisi	89
Şekil 3.45. 45 adet sensör kullanılarak tasarlanan 2. sensör geometrisi	89
Şekil 3.46. 50 adet sensör kullanılarak tasarlanan 2. sensör geometrisi	90
Şekil 3.47. 6 adet sensör kullanılarak tasarlanan 3. sensör geometrisi	90
Şekil 3.48. 8 adet sensör kullanılarak tasarlanan 3. sensör geometrisi	91
Şekil 3.49. 10 adet sensör kullanılarak tasarlanan 3. sensör geometrisi	91
Şekil 3.50. 12 adet sensör kullanılarak tasarlanan 3. sensör geometrisi	92
Şekil 3.51. 16 adet sensör kullanılarak tasarlanan 3. sensör geometrisi	92
Şekil 3.52. 20 adet sensör kullanılarak tasarlanan 3. sensör geometrisi	93

Şekil	Sayfa
Şekil 3.53. 25 adet sensör kullanılarak tasarlanan 3. sensör geometrisi	93
Şekil 3.54. 30 adet sensör kullanılarak tasarlanan 3. sensör geometrisi	94
Şekil 3.55. 35 adet sensör kullanılarak tasarlanan 3. sensör geometrisi	94
Şekil 3.56. 40 adet sensör kullanılarak tasarlanan 3. sensör geometrisi	95
Şekil 3.57. 45 adet sensör kullanılarak tasarlanan 3. sensör geometrisi	95
Şekil 3.58. 50 adet sensör kullanılarak tasarlanan 3. sensör geometrisi	96
Şekil 3.59. 6 adet sensör kullanılarak tasarlanan 4. sensör geometrisi	97
Şekil 3.60. 8 adet sensör kullanılarak tasarlanan 4. sensör geometrisi	97
Şekil 3.61. 10 adet sensör kullanılarak tasarlanan 4. sensör geometrisi	98
Şekil 3.62. 12 adet sensör kullanılarak tasarlanan 4. sensör geometrisi	98
Şekil 3.63. 16 adet sensör kullanılarak tasarlanan 4. sensör geometrisi	99
Şekil 3.64. 20 adet sensör kullanılarak tasarlanan 4. sensör geometrisi	99
Şekil 3.65. 25 adet sensör kullanılarak tasarlanan 4. sensör geometrisi	100
Şekil 3.66. 30 adet sensör kullanılarak tasarlanan 4. sensör geometrisi	100
Şekil 3.67. 35 adet sensör kullanılarak tasarlanan 4. sensör geometrisi	101
Şekil 3.68. 40 adet sensör kullanılarak tasarlanan 4. sensör geometrisi	101
Şekil 3.69. 45 adet sensör kullanılarak tasarlanan 4. sensör geometrisi	102
Şekil 3.70. 50 adet sensör kullanılarak tasarlanan 4. sensör geometrisi	102
Şekil 3.71. 8 adet sensör kullanılarak tasarlanan 5. sensör geometrisi	103
Şekil 3.72. 10 adet sensör kullanılarak tasarlanan 5. sensör geometrisi	104
Şekil 3.73. 12 adet sensör kullanılarak tasarlanan 5. sensör geometrisi	104
Şekil 3.74. 16 adet sensör kullanılarak tasarlanan 5. sensör geometrisi	105
Şekil 3.75. 20 adet sensör kullanılarak tasarlanan 5. sensör geometrisi	105
Şekil 3.76. 26 adet sensör kullanılarak tasarlanan 5. sensör geometrisi	106

Şekil	Sayfa
Şekil 3.77. 30 adet sensör kullanılarak tasarlanan 5. sensör geometrisi	107
Şekil 3.78. 36 adet sensör kullanılarak tasarlanan 5. sensör geometrisi	107
Şekil 3.79. 40 adet sensör kullanılarak tasarlanan 5. sensör geometrisi	108
Şekil 3.80. 46 adet sensör kullanılarak tasarlanan 5. sensör geometrisi	108
Şekil 3.81. 50 adet sensör kullanılarak tasarlanan 5. sensör geometrisi	109
Şekil 3.82. 8 adet sensör kullanılarak tasarlanan 6. sensör geometrisi	110
Şekil 3.83. 10 adet sensör kullanılarak tasarlanan 6. sensör geometrisi	110
Şekil 3.84. 12 adet sensör kullanılarak tasarlanan 6. sensör geometrisi	111
Şekil 3.85. 16 adet sensör kullanılarak tasarlanan 6. sensör geometrisi	111
Şekil 3.86. 20 adet sensör kullanılarak tasarlanan 6. sensör geometrisi	112
Şekil 3.87. 26 adet sensör kullanılarak tasarlanan 6. sensör geometrisi	112
Şekil 3.88. 30 adet sensör kullanılarak tasarlanan 6. sensör geometrisi	113
Şekil 3.89. 36 adet sensör kullanılarak tasarlanan 6. sensör geometrisi	114
Şekil 3.90. 40 adet sensör kullanılarak tasarlanan 6. sensör geometrisi	114
Şekil 3.91. 46 adet sensör kullanılarak tasarlanan 6. sensör geometrisi	115
Şekil 3.92. 50 adet sensör kullanılarak tasarlanan 6. sensör geometrisi	115
Şekil 4.1. Hedef nesnenin hareket modeli ve UGB sensörlerinin konumları	119
Şekil 4.2. UGB sensörünün ölçüm gürültüsü varyansı $0,5 m^2$ iken farklı GKF süreç gürültüsü varyans değerleri için 100 Monte Carlo simülasyonu yapılarak elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları	120
Şekil 4.3. UGB sensörünün ölçüm gürültüsü varyansı $1 m^2$ iken farklı GKF süreç gürültüsü varyans değerleri için, 100 Monte Carlo simülasyonu yapılarak elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları	121
Şekil 4.4. UGB sensörünün ölçüm gürültüsü varyansı $5 m^2$ iken farklı GKF süreç gürültüsü varyans değerleri için 100 Monte Carlo simülasyonu yapılarak elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları	122

- Şekil 4.5. UGB sensörünün ölçüm gürültüsü varyansı 15 m^2 iken farklı GKF süreç gürültüsü varyans değerleri için 100 Monte Carlo simülasyonu yapılarak elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları123
- Şekil 4.6. Genişletilmiş Kalman filtresinin süreç gürültüsü varyansı $1 \left(\frac{m}{sn^2}\right)^2$ iken UGB sensörünün farklı ölçüm gürültüsü varyans değerleri için elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları125
- Şekil 4.7. Genişletilmiş Kalman filtresinin süreç gürültüsü varyansı $10 \left(\frac{m}{sn^2}\right)^2$ iken UGB sensörünün farklı ölçüm gürültüsü varyans değerleri için elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları126
- Şekil 4.8. Genişletilmiş Kalman filtresinin süreç gürültüsü varyansı $50 \left(\frac{m}{sn^2}\right)^2$ iken UGB sensörünün farklı ölçüm gürültüsü varyans değerleri için elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları127
- Şekil 4.9. Sensörler, nesne hareketinin dışına yerleştirilerek elde edilen senaryo ve UGB sensörlerinin konumları.....131
- Şekil 4.10. Sensörler, nesne hareketinin içine yerleştirilerek elde edilen senaryo ve UGB sensörlerinin konumları.....131
- Şekil 4.11. Sensörler nesne hareketinin dışına yerleştirildiğinde, GKF algoritmasının, 100 Monte Carlo simülasyonu yapılarak elde edilen x ve y eksenlerindeki KOK konum kestirim hatası132
- Şekil 4.12. Sensörler nesne hareketinin içine yerleştirildiğinde, GKF algoritmasının, 100 Monte Carlo simülasyonu yapılarak elde edilen x ve y eksenlerindeki KOK konum kestirim hatası132
- Şekil 4.13. Birinci sensör geometrisi ve birinci sensör geometrisinde bulunan sensörlerin yerleri değiştirilerek elde edilen yeni sensör geometrileri kullanıldığında, GKF algoritmasının nesnenin x ve y eksenlerindeki konum bileşenlerinin kestiriminde 3. sensör kullanıldığında elde edilen Kalman kazançları.....134
- Şekil 4.14. Birinci sensör geometrisi ve birinci sensör geometrisinde bulunan sensörlerin yerlerinin değiştirilmesiyle elde edilen yeni sensör geometrileri kullanıldığında, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları134
- Şekil 4.15. İkinci sensör geometrisi ve ikinci sensör geometrisinde bulunan sensörlerin yerleri değiştirilerek elde edilen yeni sensör geometrileri kullanıldığında, GKF algoritmasının nesnenin x ve y eksenlerindeki konum bileşenlerinin kestiriminde 3. sensör kullanıldığında elde edilen Kalman kazançları.....136

Şekil**Sayfa**

- Şekil 4.16. İkinci sensör geometrisi ve ikinci sensör geometrisinde bulunan sensörlerin yerlerinin değiştirilmesiyle elde edilen yeni sensör geometrileri kullanıldığında, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları136
- Şekil 4.17. Üçüncü sensör geometrisi ve üçüncü sensör geometrisinde bulunan sensörlerin yerleri değiştirilerek elde edilen yeni sensör geometrileri kullanıldığı durumda, GKF algoritmasının nesnenin x ve y eksenlerindeki konum bileşenlerinin kestiriminde 3. sensör kullanıldığında elde edilen Kalman kazançları.....138
- Şekil 4.18. Üçüncü sensör geometrisi ve üçüncü sensör geometrisinde bulunan sensörlerin yerlerinin değiştirilmesiyle elde edilen yeni sensör geometrileri kullanıldığında, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları138
- Şekil 4.19. Dördüncü sensör geometrisi kullanıldığı durumda, GKF algoritmasının nesnenin x ve y eksenlerindeki konum bileşenlerinin kestiriminde 3. sensör kullanıldığında elde edilen Kalman kazançları140
- Şekil 4.20. Dördüncü sensör geometrisi ve dördüncü sensör geometrisinde bulunan sensörlerin yerlerinin değiştirilmesiyle elde edilen yeni sensör geometrileri kullanıldığında, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları.....140
- Şekil 4.21. Beşinci sensör geometrisi ve beşinci sensör geometrisinde bulunan sensörlerin yerleri değiştirilerek elde edilen yeni sensör geometrileri kullanıldığı durumda, GKF algoritmasının nesnenin x ve y eksenlerindeki konum bileşenlerinin kestiriminde 6. sensör kullanıldığında elde edilen Kalman kazançları.....142
- Şekil 4.22. Beşinci sensör geometrisi ve beşinci sensör geometrisinde bulunan sensörlerin yerlerinin değiştirilmesiyle elde edilen yeni sensör geometrileri kullanıldığında, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları142
- Şekil 4.23. Altıncı sensör geometrisi ve altıncı sensör geometrisinde bulunan sensörlerin yerleri değiştirilerek elde edilen yeni sensör geometrileri kullanıldığı durumda, GKF algoritmasının nesnenin x ve y eksenlerindeki konum bileşenlerinin kestiriminde, 6. sensör kullanıldığında elde edilen Kalman kazançları.....144
- Şekil 4.24. Altıncı sensör geometrisi ve altıncı sensör geometrisinde bulunan sensörlerin yerlerinin değiştirilmesiyle elde edilen yeni sensör geometrileri kullanıldığında, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları145
- Şekil 4.25. İki boyutlu Gauss dağılımına sahip rasgele değişkenin doğrusal olmayan dönüşümü sonucu ortalama ve kovaryans değerlerinin gerçek değerleri ile GKF ve kokusuz dönüşüm sonucu elde edilen değerleri146
- Şekil 4.26. 1. sensör geometrisinde kullanılan sensör sayısının arttırılmasıyla elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları148

Şekil	Sayfa
Şekil 4.27. 2. sensör geometrisinde kullanılan sensör sayısının arttırılmasıyla elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları	149
Şekil 4.28. 3. sensör geometrisinde kullanılan sensör sayısının arttırılmasıyla elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları	150
Şekil 4.29. 4. sensör geometrisinde kullanılan sensör sayısının arttırılmasıyla elde edilen GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları	152
Şekil 4.30. 5. sensör geometrisinde kullanılan sensör sayısının arttırılmasıyla elde edilen GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları	153
Şekil 4.31. 6. sensör geometrisinde kullanılan sensör sayısının arttırılmasıyla elde edilen GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları	155
Şekil 4.32. Birinci sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla güncellendiğinde, sensör sayısı arttırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları	158
Şekil 4.33. Birinci sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları karışık sırayla güncellendiğinde, sensör sayısı arttırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları	160
Şekil 4.34. Birinci sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiğinde, sensör sayısı arttırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları	162
Şekil 4.35. İkinci sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla güncellendiğinde, sensör sayısı arttırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları	164
Şekil 4.36. İkinci sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları karışık sırayla güncellendiğinde, sensör sayısı arttırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları	166
Şekil 4.37. İkinci sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiğinde, sensör sayısı arttırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları	168

Şekil**Sayfa**

- Şekil 4.38. Üçüncü sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları170
- Şekil 4.39. Üçüncü sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları karışık sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları172
- Şekil 4.40. Üçüncü sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları174
- Şekil 4.41. Dördüncü sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları176
- Şekil 4.42. Dördüncü sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları karışık sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları178
- Şekil 4.43. Dördüncü sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları180
- Şekil 4.44. Beşinci sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları182
- Şekil 4.45. Beşinci sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları karışık sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları184
- Şekil 4.46. Beşinci sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları186

Şekil**Sayfa**

Şekil 4.47. Altıncı sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları	188
Şekil 4.48. Altıncı sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları karışık sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları	190
Şekil 4.49. Altıncı sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları	192
Şekil 4.50. Nesnenin karesel hareket modeli ve sensör konumları.....	194
Şekil 4.51. Nesne karesel şekilde hareket ettiğinde ve 1. sensör geometrisi kullanıldığında GKF algoritmasının, nesnenin x ve y eksenlerindeki konum kestirim hataları.....	194
Şekil 4.52. Nesnenin dairesel hareket modeli ve sensör konumları	195
Şekil 4.53. Nesne dairesel şekilde hareket ettiğinde ve 1. sensör geometrisi kullanıldığında GKF algoritmasının, nesnenin x ve y eksenlerindeki konum kestirim hataları.....	195
Şekil 4.54. Nesnenin hareket modeli ve sensör konumları.....	196
Şekil 4.55. Nesne dairesel şekilde hareket ettiğinde ve 1. sensör geometrisi kullanıldığında GKF algoritmasının, nesnenin x ve y eksenlerindeki konum kestirim hataları.....	197
Şekil 4.56. Nesnenin karesel hareket modeli ve sensör konumları.....	198
Şekil 4.57. Nesne karesel şekilde hareket ettiğinde ve 5. sensör geometrisi kullanıldığında GKF algoritmasının, nesnenin x ve y eksenlerindeki konum kestirim hataları.....	198
Şekil 4.58. Nesnenin dairesel hareket modeli ve sensör konumları	199
Şekil 4.59. Nesne dairesel hareket ettiğinde ve 5. sensör geometrisi kullanıldığında GKF algoritmasının, nesnenin x ve y eksenlerindeki konum kestirim hataları	199
Şekil 4.60. Nesnenin karesel hareket modeli ve sensör konumları.....	200

Şekil**Sayfa**

Şekil 4.61. Nesne karesel hareket ettiğinde ve 6. sensör geometrisi kullanıldığında GKF algoritmasının, x ve y eksenlerindeki konum kestirim hataları.....	200
Şekil 4.62. Nesnenin dairesel hareket modeli ve sensör konumları	201
Şekil 4.63. Nesne dairesel hareket ettiğinde ve 6. sensör geometrisi kullanıldığında GKF algoritmasının, x ve y eksenlerindeki konum kestirim hataları.....	201



SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler

c	Işık hızı
dB	Desibel
m	Metre
m²	Metre kare
sn²	Saniye kare
T	Örnekleme aralığı
σ^2	Ölçüm gürültü varyansı
Ω	Açısal hız
σ_v	Süreç gürültüsünün standart sapması
σ_v^2	Süreç gürültüsü varyansı

Kısaltmalar

GHz	Gigahertz
GKF	Genişletilmiş Kalman filtresi
KKS	Küresel konumlandırma sistemi
KOK	Kare ortalamaların karekökü
Mbps	Saniye başına megabit (İng. Megabits per second)
MHz	Megahertz
RF	Radyo frekans
RSS	Alınan sinyal şiddeti (İng. Received signal strength)
SNR	Sinyal gürültü oranı (İng. Signal to noise ratio)
UGB	Ultra geniş bant

1. GİRİŞ

Konum belirleme, nesne, insan veya herhangi bir kullanıcının, belirli bir koordinat sistemi üzerinde bulunduğu yerin belirlenmesidir. Günümüzde nesnelerin, insanların bulundukları yerleri tespit etmek için çeşitli konumlandırma sistemleri kullanılmaktadır. Açık alanlarda nesnelerin ve insanların bulundukları konumları belirlemek amacıyla, en yaygın olarak kullanılan araç KKS (Küresel Konumlandırma Sistemi)'dir. KKS, uydu tabanlı bir sistemdir ve konum tespiti yapabilmek için uydu sinyallerini kullanmaktadır, dolayısıyla KKS kullanarak konum tespiti yapabilmek için, uydulardan gönderilen sinyallerin alıcıya ulaşması ve bunun için de konumu tespit edilecek nesnenin görüş hattında olması gerekmektedir (Kumar ve Moore, 2002). Ancak kapalı alanlarda çatılar, duvarlar gibi yapılar uydu sinyallerinin, konumu tespit edilmek istenen nesneye ulaşmasını engellemektedir. Bu sebeple, küresel konumlandırma sistemi kapalı alanlarda kullanılamamaktadır. Bundan dolayı, kapalı alanlarda nesnelerin konum tespitinin yapılabilmesi için, uydu sinyallerine ihtiyaç duymayan çeşitli kapalı alan konumlandırma sistemlerine ihtiyaç duyulmaktadır.

Kapalı alan konumlandırma sistemleri hem sivil hem de askeri birçok alanda kullanılmaktadır. Kapalı alan konumlandırma sistemleri sivil uygulamalarda müzelerde, ziyaretçiler tarafından en çok ilgi çeken eserlerin ya da günün yoğun olduğu zaman dilimlerinin belirlenmesi gibi amaçlarla kullanılırken (Lanir vd., 2017), özellikle büyük hastanelerde hastaların tıbbi bölümleri, tedavi odalarını zaman kaybetmeden bulmalarını sağlamak amacıyla da kullanılmaktadır (Yang vd., 2015). Benzer şekilde, bu sistemler, alışveriş merkezlerinde insanların mağazaları daha kolay ve hızlı bir şekilde bulmalarına da yardımcı olmaktadır. Kapalı alan konumlandırma sistemleri ayrıca savunma sanayiinde de özellikle askeri operasyonlarda, askerlerin operasyon bölgesinde bulundukları konumları tespit etmek amacıyla da kullanılmaktadır. Askeri operasyonlarda, askerlerin operasyon bölgesinde bulundukları konumların tespit edilmesi ve böylece askerlerin birbirlerinin konumlarını bilmeleri sayesinde operasyon hakkında çeşitli kararların alınması ve gerektiğinde operasyon stratejilerinin hızlı bir şekilde değiştirilebilmesi sağlanmaktadır. Böylece, daha başarılı operasyonlar gerçekleştirilebilmektedir. Ayrıca, askeri operasyonlarda keşif, gözetleme ve istihbarat amaçlı insansız hava araçları kullanılmaktadır. İnsansız hava araçları bu görevleri yerine getirebilmek için otonom navigasyon sistemine ihtiyaç duymakta ve otonom navigasyon

sistemi için gerekli olan, hava aracının başlangıçta bulunmuş olduğu konum verileri, konumlandırma sistemleri kullanılarak elde edilmektedir. Bu nedenle, küresel konumlandırma sisteminin kullanılmadığı kapalı alanlarda keşif ve gözetleme amaçlı kullanılan insansız hava araçlarının konum verilerini elde etmek için, küresel konumlandırma sistemlerine alternatif olabilecek konumlandırma sistemlerinin geliştirilmesi oldukça önemli hale gelmektedir. Kapalı alan konumlandırma sistemlerinin hem sivil hem askeri alanlardaki kullanım amaçları göz önünde bulundurulduğunda, bu sistemlerin gerekliliği ve önemi açığa çıkmaktadır. Bu nedenle, kapalı alan konumlandırma sistemleri son yıllarda oldukça popüler bir çalışma alanı haline gelmiştir.

Kapalı alanlarda nesne konumlandırma için çeşitli sensörler kullanılmaktadır. En sık kullanılan RF (İng. *Radio Frequency – RF*) tabanlı sensörler arasında WiFi (İng. *Wireless Fidelity – WIFI*), *bluetooth*, RFID (İng. *Radio Frequency Identification – RFID*) ve UGB (Ultra Geniş Bant) sensörleri bulunmaktadır. WiFi, radyo dalgaları kullanarak cep telefonu, bilgisayar gibi farklı cihazlarla haberleşme yapabilmektedir. WiFi tabanlı kapalı alan konumlandırma sistemlerinde verici tarafından gönderilen sinyalin alıcıya ulaşma süresi, alınan sinyal şiddeti (İng. *Received Signal Strength – RSS*), varış açısı gibi ölçüm yöntemleri kullanılarak alıcı ve verici arasındaki uzaklık elde edilmekte ve bu uzaklık bilgisi trilaterasyon algoritmasında kullanılarak, kullanıcının konumu belirlenmektedir (İlçi vd., 2015). WiFi tabanlı kapalı alan konumlandırma sistemleri kolay uygulanabilir, kapsama alanı yüksektir. Fakat WiFi tabanlı sistemlerin bir dezavantajı, WiFi erişim noktalarının sayısının yeterli olmadığı durumlarda, konumlandırma için yeteri kadar RSS (İng. *Received Signal Strength – RSS*) değeri elde edilememektedir ve RSS tabanlı konumlandırmada yeteri kadar RSS değeri olmadığı durumlarda, konumlandırma doğruluğu da istenilen seviyede olmamaktadır. Ayrıca, WiFi sinyali kapalı ortamlarda bulunan nesnelerden geçerken kırılma, yansıma, çok yollu yayılım gibi problemler sebebiyle zayıfladığı için, RSS değeri düşmekte ve bu sorun konumlandırma doğruluğunu önemli ölçüde etkilemektedir. Bu durum, WiFi temelli konumlandırma sistemlerinin önemli bir dezavantajıdır. Kapalı alan konumlandırma sistemlerinde kullanılan bir başka teknoloji ise *bluetooth* teknolojisidir. *Bluetooth* tabanlı kapalı alan konumlandırma sistemlerinde genellikle *bluetooth* sinyal vericileri kullanılmaktadır. Kapalı alana yerleştirilen *bluetooth* sinyal vericileri, ortama *bluetooth* sinyali yaymakta ve bu sinyaller ortama yayılarak bir kapsama alanı

oluşturmaktadır. Üzerinde *bluetooth* sinyal alıcısı bulunan nesne, *bluetooth* sinyallerinin kapsama alanına girdiğinde, bu sinyalleri algılamaktadır. Konumu tespit edilmek istenen nesne üzerinde bulunan alıcı tarafından alınan sinyalin varış açısı, varış zamanı, alınan sinyal şiddeti gibi çeşitli ölçüm yöntemleri kullanılarak *bluetooth* sinyal vericileri ile kullanıcı arasındaki uzaklık hesaplanmakta ve bu uzaklık verisi multilaterasyon algoritmasında kullanılarak, hedef kullanıcının konumu belirlenmektedir (Duong ve Thi, 2021). WiFi ve *bluetooth* tabanlı kapalı alan konumlandırma sistemleri birbirleri ile karşılaştırıldığında, WiFi sinyallerinin ve *bluetooth* sinyallerinin her ikisinin de 2,4 GHz frekans bandını kullandıkları görülmektedir ve bunun için WiFi ve *bluetooth* sinyallerinin yayılım özellikleri birbirlerine benzemektedir (Do vd., 2021). Fakat güç tüketimi açısından karşılaştırıldığında *bluetooth*, WiFi'ya göre daha az güç tüketmekte ve *bluetooth* kullanılarak daha yüksek konumlandırma doğruluğu elde edilebilmektedir.

RFID tabanlı kapalı alan konumlandırma sistemlerinde, 869-928 MHz frekans aralığında ultra yüksek frekans bandında RF sinyali kullanılmaktadır (Tan vd., 2022). RFID sistemi okuyucu ve RFID etiketinden oluşmaktadır. RFID okuyucu ortama RF sinyali yaymakta ve bu sinyaller ortamda kapsama alanı oluşturmaktadır. Üzerinde RFID etiket bulunan nesne, okuyucunun kapsama alanına girdiğinde, etiket aktif hale gelerek sinyali almakta ve alınan sinyal şiddeti değerini RFID okuyucularına iletmektedir. Okuyuculara iletilen, alınan sinyal şiddeti değerleri mesafe bilgisine dönüştürülerek elde edilen mesafe verileri, multilaterasyon algoritmasında kullanılarak konum tespiti yapılmaktadır (Chan ve Zhang, 2017). Alınan sinyal şiddeti değerinin, mesafe bilgisine dönüştürülmesi için yol kaybı modeli (İng. *Path Loss Model*) kullanılmaktadır. Bu modelde verici tarafından gönderilen sinyalin, RSS değerindeki zayıflama miktarına bağlı olarak, nesne ile verici arasındaki uzaklık hesaplanmaktadır. Sensör vericisinin gönderdiği sinyalin, 1 metre mesafedeki referans RSS değeri, sensör üretimi aşamasında belirlenen bir sabit olup bilinmektedir. Ayrıca sensörün yerleştirildiği konum da bilinmektedir. Hedef nesne, RFID okuyucunun kapsama alanına girdiğinde, okuyucunun ilettiği sinyal, etiket tarafından alınmakta ve bu sinyalin RSS değeri elde edilmektedir. Yol kaybı modeli kullanılarak, etiketin aldığı sinyalin RSS değeri ile okuyucunun gönderdiği sinyalin 1 metre uzaklıktaki referans RSS değeri arasındaki fark hesaplanmaktadır. Böylece sinyalin, konumu bilinen referans noktadan, nesneye ulaştığı anda ne kadar zayıfladığı belirlenmektedir. Bu zayıflama miktarının, yani konumu bilinen referans noktadaki RSS değeri ile nesneye ulaşan sinyalin RSS değeri arasındaki farkın, yol kaybı modelinde

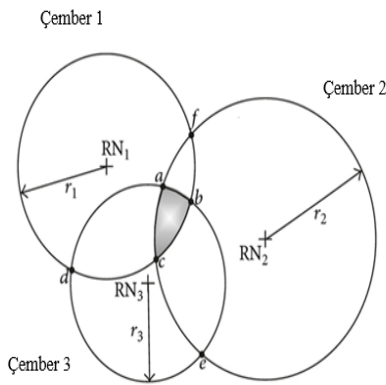
kullanılmasıyla uzaklık hesaplanmaktadır. Yol kaybı modelinde, bu farkın yol kaybı indeksine oranlanmasıyla nesne ile okuyucu arasındaki uzaklık hesaplanmaktadır. Yol kaybı indeksi, sinyalin iletiildiği ortama bağılı olarak kullanılan bir sabittir (Thaljaoui vd., 2015). RFID tabanlı kapalı alan konumlandırma sisteminin avantajı, RFID sensörünün düşük güç tüketimi ve düşük maliyete sahip olmasıdır. Ancak, RFID sensörünün dezavantajı ise, RFID sensörünün ilettiği RF sinyali, ortamda bulunan nesnelerden dolayı kırınım, yansıma ve çok yollu yayılım gibi problemlerle karşılaştığı için sinyal alıcıya ulaştığında, RSS değerinin düşmesinden kaynaklanan yüksek konumlandırma hatasına sahip olmasıdır.

UGB sensörü, geniş bant aralığında veri iletimi yapabilen ve güç tüketimi düşük olan kablosuz bir teknolojidir (Muqaibel vd., 2004). UGB sinyali geniş bant aralığına sahip olduğu için, sinyal farklı frekans bileşenlerinden oluşmakta ve bu frekans çeşitliliği ise UGB sinyali için bazı avantajlar sağlamaktadır. Farklı frekanslara sahip sinyallerden bazıları konumlandırma yapılan kapalı ortamdaki engellerden geçemese bile, farklı frekansa sahip sinyaller bu engellerden geçebilirler ve bu sinyaller konumlandırma yapabilmek için yeterlidirler. Bu özellik ise, *bluetooth* ve WiFi ile karşılaştırıldığında, konumu tespit edilmek istenen nesne ile sensör arasındaki görüş hattının iyi olmadığı durumlarda, veri iletimi açısından, geniş bant aralığından dolayı ultra geniş bant sensörünün sunmuş olduğu bir avantajdır. Ayrıca, UGB sinyallerinin, farklı frekanslara sahip bileşenler içermesinin, bir başka faydası ise sinyalin karıştırılma sorununa karşı UGB sensörünün daha dayanıklı olmasıdır. UGB vericisinden gönderilen sinyaller karıştırılmak istendiğinde, sinyaldeki bazı frekans bileşenleri karıştırılabilse bile, zarar verilemeyen farklı frekanslara sahip sinyaller, UGB alıcısına ulaşabilmekte ve bu sinyaller konumlandırma yapabilmek için yeterli olmaktadır. UGB sensörünün sahip olduğu bu avantaj, özellikle savunma sanayiinde, olası operasyon anında sinyal karıştırma sorununa karşı oldukça önemlidir. UGB sensör sinyalinin yüksek bant genişliği ve kısa darbeleri dalga formu özelliğinin olması, sinyalin çok yollu yayılma sorununa karşı daha dayanıklı olmasını sağlamaktadır. Çok yollu yayılım sorunu, kapalı alanda bulunan nesnelerden dolayı meydana gelmektedir. Verici tarafından gönderilen sinyal alıcıya ulaşırken nesnelere çarptığında yansımakta ve yansıyan sinyal ortamda bulunan başka nesne veya duvarlardan tekrar yansiyarak birden fazla yoldan alıcıya ulaşmaktadır. Bu durumda, çok yollu yayılım problemi meydana gelmektedir. UGB vericisi tarafından iletilen ve okuyucuyla görüş hattında olan sinyal direk alıcıya

ulaşmaktadır. Ancak, okuyucu ile görüş hattında olmayan sinyal ortamda bulunan nesnelerden yansiyarak okuyucuya ulaşmaktadır. Okuyucuyla görüş hattında bulunan kısa darbeli UGB sinyali, çok yollu yayılım sorunuyla karşılaşarak alıcıya ulaşan sinyallerden daha önce alıcıya ulaşmaktadır. Sinyaller farklı zamanlarda alıcıya ulaştığı için, görüş hattından alıcıya ulaşan kısa darbeli sinyal ile çok yollu yayılıma uğrayarak alıcıya ulaşan sinyaller birbirleriyle girişime uğramamakta ve karışmamaktadır (Schejbal vd., 2006). UGB verici tarafından gönderilen kısa darbeli sinyal, ortamdaki nesnelerden yansiyarak çok yollu yayılıma uğrayarak alıcıya gelen diğer sinyallerle girişime girip, o sinyallerle karışmadığı için bozulmamaktadır. Bu sebeple, kısa darbeli UGB sinyalleri kullanılarak çok yollu yayılım durumunda elde edilen ölçüm hataları düşük olduğu için, bu sinyaller çok yollu yayılıma karşı daha dayanıklı olmaktadır. Ayrıca, RFID, *bluetooth* ve WiFi gibi diğer sensörlerin kapsama alanları göz önüne alındığında, ultra geniş bant sensörünün, diğer sensörlerden daha büyük kapsama alanına sahip olduğu görülmektedir (Do vd., 2021). Sonuç olarak, UGB sensörü diğer sensörlerle karşılaştırıldığında, daha yüksek konumlandırma doğruluğu elde edilebilmesi, UGB sensör sinyalinin ortamdaki bozucu girişimlere karşı dayanıklı olması, sinyalin ortamda var olan diğer RF sinyalleriyle etkileşime girmemesi, yüksek miktarda veri iletebilmesi, enerji tüketiminin düşük olması ve sensörün yüksek kapsama alanına sahip olması gibi avantajları bulunmaktadır. Bütün bu özellik ve avantajlarından dolayı, kapalı alanda nesne konumlandırma problemi için ultra geniş bant sensörü, RFID, *bluetooth*, WiFi gibi diğer sensörler arasında öne çıkmaktadır. Bu nedenle, bu tez çalışmasında, kapalı alanda nesne konumlandırma problemi için ultra geniş bant sensörünün çalışılması ve analiz edilmesi üzerine yoğunlaşmıştır.

Kapalı alanlarda konumlandırma yapabilmek için çeşitli algoritmalar kullanılmaktadır. Bu algoritmalarından bazıları multilaterasyon tabanlı algoritmalar, parmak izi algoritması ve Kalman filtre tabanlı algoritmalar. Multilaterasyon algoritmasında çember geometrisi kullanılmaktadır. Hedef nesne üzerinde bulunan sensör etiketi tarafından, okuyuculara sinyal gönderilmektedir. Bu sinyaller konum koordinatları bilinen referans noktalara yerleştirilen 3 adet sensör okuyucusu tarafından alınmaktadır. Bu sinyaller, konumu tespit edilmek istenen hedef nesne ile sensör okuyucusu arasındaki uzaklığın hesaplanması için kullanılmaktadır. Bu uzaklığın hesaplanması için, sensör etiketi tarafından iletilen sinyalin hedef nesneye varış zamanı süresi ölçümü, varış açısı ölçüm yöntemi veya alınan sinyal şiddeti ölçüm yöntemi kullanılmaktadır. Multilaterasyon

algoritmasında her bir sensör okuyucusunun bulunduğu konum koordinatları, algoritmada kullanılan dairenin merkezi ve her bir sensör okuyucusu ile kullanıcı arasındaki uzaklık ise yarıçap kabul edilerek, her bir sensör okuyucusu için daireler çizilmekte ve bu dairelerin kesişim bölgesi bulunmaktadır. Konumu tespit edilmek istenen nesnenin konum koordinatları, çemberlerin kesişimi sonucu oluşan bölgede bulunmakta ve bu kesişim bölgesi daraltılarak, nesnenin konumu tespit edilmektedir (Sang vd., 2019). Sensör ölçümleri multilaterasyon algoritmasında kullanılan çember geometrisinin yarıçapını temsil etmektedir. Kapalı alanlarda bulunan nesnelerden dolayı sensör okuyucusunun, konumu tespit edilmek istenen nesnedeki alıcıya gönderdiği sinyal, kırınım, yansıma ve çok yollu yayılım sorunlarıyla karşılaşmakta ve bu nedenle ölçüm hataları meydana gelmektedir. Bu nedenle sensör ölçümü sonucu elde edilen uzaklık verileri hatalı olmaktadır. Bu durumda, uzaklık verileri multilaterasyon algoritmasında kullanılan çemberin yarıçapını temsil ettiği için, algoritmada kullanılan çemberlerin yarıçapları hatalı olmaktadır. Bu sebeple, elde edilen çember geometrisi hatalı olmakta ve bu durumda çemberlerin kesişimi sonucu oluşan ve nesnenin konumunu tespit etmek için kullanılan kesişim bölgesi de hatalı olmaktadır. Bu nedenle, multilaterasyon algoritmasında, sensör ölçüm hataları azaltılmadığı için önemli konumlandırma hataları meydana gelmektedir.



Şekil 1.1. Multilaterasyon yönteminin gösterimi (Brida ve Machaj, 2013)

Multilaterasyon algoritmasının çalışma prensibi Şekil 1.1.'de verilmiştir. Multilaterasyon algoritmasında çember geometrisi kullanılmakta ve her bir sensör okuyucusu için sanal çemberler çizilmektedir. Sensör okuyucularının bulundukları konum koordinatları sanal çemberin merkezi ve okuyucu ile hedef nesne arasındaki uzaklık ise sanal çember yarıçapını ifade etmektedir. Nesne hareket ederken üzerinde bulunduğu sensör etiketi ile 1. sensör

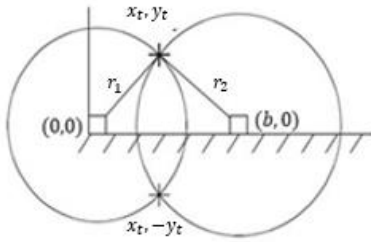
okuyucusu arasındaki uzaklık ölçülmekte ve bu ölçüm 1. sensör okuyucusu için çizilen sanal çemberin yarıçapını belirtmektedir. Şekil 1.1. incelendiğinde, RN_1 birinci sensör okuyucusunun bulunduğu konum koordinatı olup, 1. sensör okuyucusu için çizilen çemberin merkezidir. r_1 ise, 1. sensör okuyucusu ile hedef nesne arasındaki ölçülen uzaklık olup 1. çemberin yarıçapıdır. Burada 1. sensör okuyucusunun konumu bilindiği için çizilen çemberin merkezi bilinmekte ve sensörle uzaklık ölçümü yapıldığı için 1. sensör okuyucusu ile hedef nesne arasındaki uzaklık bilindiği için r_1 yarıçapı da bilinmektedir. Çünkü sanal çemberin yarıçapı, nesne ile 1. sensör okuyucusu arasındaki ölçülen uzaklığa eşittir. Dolayısıyla sensör bu uzaklığı ölçtüğü için r_1 yarıçapı bilinmektedir. Ancak hedef nesnenin yeri tam olarak bilinmemektedir. Sadece hedef nesnenin, 1. sensör okuyucusundan r_1 kadar uzaklıkta bir noktada olduğu bilinmektedir. 1. sensör okuyucusu için çizilen çember üzerindeki her bir nokta ile RN_1 noktası arasındaki uzaklık, 1. çemberin yarıçapı kadar olduğu için, hedef nesne ile okuyucu arasındaki uzaklık, çember üzerindeki her bir noktada r_1 kadardır. Bu nedenle, nesne 1. çember üzerinde bulunan herhangi bir noktada olabilir. Ancak, 1. çember üzerinde hangi noktada olduğu kesin olarak bilinmemektedir. Nesne ortamda hareket ederken, aynı zamanda, nesne üzerinde bulunan sensör etiketi ile 2. sensör okuyucusu arasındaki uzaklık ölçümü de yapılmakta ve bu uzaklık ise 2. çemberin yarıçapını belirtmekte olup, Şekil 1.1.'de r_2 ile gösterilmiştir. 2. sensör okuyucusunun bulunduğu konum koordinatı ise çemberin merkezi olup RN_2 ile gösterilmiştir. Sensör ölçümü yapıldığında, nesnenin 2. sensör okuyucusuna r_2 metre kadar uzakta herhangi bir noktada olduğu bilinmekte ancak hangi noktada olduğu kesin olarak bilinmemektedir. Bu nedenle, merkezi RN_2 ve yarıçapı r_2 olan 2. çember çizildiğinde, nesnenin bu çember üzerinde bir noktada olduğu bilinmektedir. Çünkü çember üzerindeki her bir nokta, 2. okuyucuya r_2 kadar uzaklıkta olduğu için, nesne bu çember üzerinde herhangi bir noktada bulunmaktadır. Benzer şekilde, nesne ortamda hareket ederken, nesne üzerinde bulunan sensör etiketi ile 3. sensör okuyucusu arasındaki uzaklık ölçümü de yapılmakta ve bu uzaklık, multilaterasyon algoritmasında kullanılan 3. çemberin yarıçapını belirtmekte olup, Şekil 1.1.'de r_3 ile gösterilmiştir. Ayrıca, 3. sensör okuyucusunun bulunduğu konum koordinatı ise 3. sensör için çizilen çemberin merkezi olup, Şekil 1.1.'de RN_3 ile gösterilmiştir. Sensör ölçümü yapıldığında, nesnenin 3. sensör okuyucusuna r_3 metre kadar uzakta herhangi bir noktada olduğu bilinmekte ancak kesin olarak hangi noktada olduğu bilinmemektedir. Bunun anlamı nesne, RN_3 noktasından r_3 kadar uzakta bir yerdedir. Bu nedenle, merkezi RN_3 ve yarıçapı r_3 olan 3. çember çizildiğinde, nesnenin bu çember üzerinde bir noktada olduğu

bilinmektedir. Çünkü çember üzerindeki her bir nokta, RN_3 noktasına r_3 metre kadar uzaklıkta bulunmaktadır. Bu çemberle aslında, nesnenin 3. sensör okuyucusundan r_3 metre kadar uzakta bir noktada olduğu bilinmekte ve çember üzerindeki her bir nokta 3. okuyucu ile nesne arasındaki r_3 metre kadar uzaklığa sahip noktalar olduğu için, hedef nesne bu çember üzerinde herhangi bir noktada olduğu bilinmektedir. Nesne ortamda hareket ederken, 1. sensör nesne ile arasındaki uzaklığı ölçmekte ve nesnenin 1. sensör için çizilen RN_1 merkezli ve r_1 yarıçaplı 1. çember üzerinde herhangi bir noktada olduğu bilgisi elde edilmektedir. Nesne hareket ederken aynı zamanda 2. sensör de, nesne ile arasındaki uzaklığı ölçerek nesnenin kendisinden r_2 kadar uzaklıkta herhangi bir noktada olduğu ve bu nedenle nesnenin 2. sensör için çizilen RN_2 merkezli ve r_2 yarıçaplı daire üzerinde herhangi bir noktada bulunduğu bilinmektedir. Aynı şekilde nesne hareket ederken 3. sensör ile de nesnenin kendisine olan uzaklığı ölçülmekte ve kendisinden r_3 metre kadar uzakta, RN_3 merkezli ve r_3 yarıçaplı 3. çember üzerinde bir noktada olduğu bilinmektedir. Her üç sensör de nesne ile arasındaki uzaklığın ölçümünü aynı anda yaptığı ve nesnenin her bir sensör için çizilen çemberler üzerinde bir noktada olduğu bilindiği için, nesne Şekil 1.1. ile gösterilen çemberlerin kesişim bölgesi içinde bulunmalıdır. Çünkü 3 sensörün her birisi nesneyle kendisi arasındaki uzaklığı ölçtüğü için, nesne her 3 sensör için çizilen çember üzerinde bir noktadadır. Nesne her bir sensör için çizilen çember üzerinde herhangi bir noktada olduğu için ve nesne hareket ederken her 3 sensör de aynı zamanda ölçüm yaptığı için, nesne her bir sensör için çizilen 3 çemberin kesişim bölgesinde bir noktada olmalıdır. Bu nedenle nesne, Şekil 1.1.'de gri ile gösterilen alanda bulunmaktadır. Bu kesişim bölgesi daraltılarak, nesnenin konumu tespit edilmektedir (Sang vd., 2019).

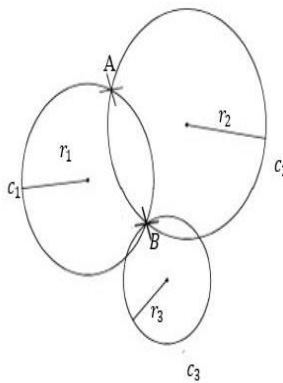
Parmak izi algoritması, kapalı alanda konum belirlemek için kullanılan bir başka algoritmadır. Parmak izi algoritmasında, alınan sinyal şiddeti değerleri kullanılmaktadır. Konumu belli olan referans noktalara yerleştirilmiş olan sensör okuyucularının RSS değerleri elde edilerek kümeleme yöntemiyle gruplandırılmakta ve veri tabanında tutulmaktadır. Konumu tespit edilmek istenen nesne üzerinde bulunan sensör etiketinin (*Ing. Tag*) RSS değerinin, veri tabanında bulunan ve sensör okuyucularına ait olan RSS değerleriyle eşleştirilmesiyle, nesnenin konumu tespit edilmektedir (Zegeye vd., 2016). Ancak, kapalı ortamlarda bulunan engellerden dolayı sensör sinyalinin ciddi anlamda zayıflaması nedeniyle RSS değerleri düşmekte ve bu nedenle yüksek konumlandırma hataları meydana gelmektedir. Bu problemler Kalman filtre algoritmaları kullanılarak

ortadan kaldırılabilir. Bu nedenle Kalman filtre tabanlı algoritmalar kullanılarak konum belirleme problemleri daha yüksek doğruluklarla çözülebilmektedir.

Tez çalışması kapsamında, ultra geniş bant sensörü kullanarak kapalı alanda nesne konumlandırma probleminde, GKF (Genişletilmiş Kalman Filtre) algoritmasının başarımlarının belirlenmesi ile farklı sensör geometrilerinin ve kullanılan sensör sayısının, GKF algoritmasının kestirim performansı üzerindeki etkilerinin araştırılması amaçlanmıştır. 2 boyutlu Kartezyen koordinat sistemi üzerinde nesne konumlandırmak için 3 adet sensöre ihtiyaç duyulmakta ve sensör ölçümleri GKF algoritmasında kullanılarak, konum kestirimi elde edilmektedir. Her bir sensör tarafından sadece 1 boyutlu ölçüm elde edilmektedir. 2 boyutlu koordinat sistemi üzerinde konum belirleyebilmek için 3 adet sensöre ihtiyaç vardır.



Şekil 1.2. İki sensör kullanıldığı durumda trilaterasyon yönteminin gösterimi (Sánchez vd., 2009)



Şekil 1.3. 3 sensör kullanıldığı durumda trilaterasyon yönteminin gösterimi (Sánchez vd., 2009)

Şekil 1.2.'de 2 adet sensör kullanıldığı durumda trilaterasyon algoritmasında kullanılan çember geometrisi gösterilmiştir. Trilaterasyon algoritmasında sensörlerin bulunduğu konum, o sensör için çizilecek çemberin merkezi ve sensör ile hedef nesne arasındaki uzaklık

ise çemberin yarıçapı kabul edilerek, her bir sensör için çember çizilmektedir. Çemberlerin kesişim noktası, nesnenin bulunduğu konumu vermektedir. Şekil 1.2.'de birinci sensör, iki boyutlu koordinat sistemi üzerinde (0,0) noktasında bulunduğu için, 1. sensör için çizilen çemberin merkezi (0,0) noktasıdır. 1. sensörle nesne arasındaki uzaklık ise çemberin yarıçapı olan r_1 kadardır. 1. sensör için merkezi (0, 0) noktası, yarıçapı ise r_1 olan çember çizilmiştir. Aynı şekilde, 2. sensörle nesne arasındaki uzaklık ise çemberin yarıçapı olan r_2 kadardır. 2. sensör için de sensörün bulunduğu konum koordinatı olan (b, 0) noktası merkez ve yarıçap r_2 kabul edilerek, 2. sensör için de çember çizilmiştir. Trilaterasyon algoritmasında, her bir sensör için çember çizilmekte ve çizilen bu çemberlerin kesişim noktası, hedef nesnenin bulunduğu konumu vermektedir. Bu nedenle, konum belirlemek için 2 adet sensör kullanıldığında, trilaterasyon algoritmasında da 2 adet çember geometrisi kullanılmaktadır. Şekil 1.2. incelendiğinde, 2 adet çember kullanıldığı için, bu çemberlerin iki farklı noktada kesiştiği görülmektedir. Bu nedenle, hedef nesne bu iki noktadan birisinde bulunmaktadır. Ancak, nesnenin kesin olarak hangi noktada bulunduğu bilinmemektedir. Bu nedenle, nesnenin bulunduğu konumu kesin olarak belirleyebilmek için 1 sensöre daha ihtiyaç duyulmaktadır. Şekil 1.3.'te, 3 sensör kullanıldığı durumda trilaterasyon algoritmasında kullanılan çember geometrisi gösterilmiştir. Şekil 1.3.'te, 3. sensörün bulunduğu konum 3. sensör için çizilecek çemberin merkezi, sensör ile hedef nesne arasındaki uzaklık ise çemberin yarıçapı kabul edilerek, 3. sensör için de çember çizilmiştir. Bu durumda, 3 çember sadece tek bir noktada kesişmiştir. Çemberlerin kesişim noktası, nesnenin konumu olduğu için, artık hedef nesnenin, Şekil 1.3. te verilen B noktasında olduğu belirlenmiştir. Şekil 1.2. de, konum belirlemek için 2 adet sensör kullanıldığı durumda, çemberler 2 farklı noktada kesişmekteydi ancak, 3. sensörün eklenmesiyle, Şekil 1.2.'de verilen çember geometrisine üçüncü bir çember eklenerek Şekil 1.3. elde edilmiştir. Şekil 1.3. incelendiğinde, 3. çemberin eklenmesiyle, çemberlerin tek bir noktada kesiştiği görülmektedir. Çemberlerin kesişim noktası, hedef nesnenin konumu olduğu için, artık 3 sensör kullanarak nesnenin kesin olarak B noktasında olduğunu belirlemiş olmaktayız. Bu nedenle, GKF algoritması kullanarak 2 boyutlu koordinat sistemi üzerinde konum belirlemek için, 3 sensöre ihtiyaç vardır. 3 adet sensör ölçümü GKF algoritmasında birlikte kullanılarak konum kestirimi elde edilmektedir. 3 adet sensörden elde edilen ölçümler, GKF algoritmasında birlikte kullanılarak konum belirlendiği için, sensör geometrisi algoritmanın kestirim performansını etkilemektedir. Bu nedenle, bu tez çalışmasında sensör geometrisinin, GKF algoritmasının kestirim performansı üzerindeki etkileri araştırılmıştır.

2. KAYNAK ARAŞTIRMASI

Kapalı alanlarda konum belirleme problemi insan, nesne veya herhangi bir kullanıcının kapalı alanda referans koordinat sistemi üzerinde bulunduğu konumunun, yerinin belirlenmesidir. Kapalı alanlarda konum belirlemek için kullanılan sistemler ise kapalı alan konumlandırma sistemleri olarak adlandırılabilir. Günümüzde açık alanlarda konum belirlemek için yaygın olarak kullanılan araç KKS'dir. KKS kullanılarak konum belirlemek için uydu sinyalleri kullanıldığı için, uydu sinyallerinin, üzerinde KKS bulunan ve konumu tespit edilmek istenen nesneye ulaşması gerekmektedir. Genellikle açık alanlarda, uydu sinyali, KKS'ye ulaşmakta zorluk yaşamadığı için açık alanlarda KKS kullanılarak konum tespiti yapılabilir. Ancak, kapalı alanlarda konum tespiti yapılmak istendiğinde, çatılar, duvarlar gibi yapılar uydu sinyallerinin kapalı alana geçmesini ve konumu tespit edilmek istenen ve üzerinde KKS bulunan nesneye ulaşmasını engellemektedir. Bu nedenle kapalı alanlarda uydu sinyalleri, KKS'ye ulaşamadığından dolayı, KKS kapalı alanlarda kullanılamamaktadır. Bu durum, kapalı alanlarda konum belirleme ihtiyacının karşılanması için, KKS'ye ihtiyaç duymayan, kapalı alan konumlandırma sistemlerinin geliştirilmesi gerekliliğini ortaya çıkarmaktadır.

Günümüzde sivil ve askeri birçok farklı alanlarda, kapalı alan konumlandırma sistemlerine ihtiyaç duyulmaktadır. Kapalı alan konumlandırma sistemlerinin önemini anlamak için, bu sistemlerin kullanım alanları ve sağlamış olduğu kolaylıklar göz önünde bulundurulmalıdır. Kapalı alan konumlandırma sistemleri sivil alanlarda hava alanları, tren istasyonları, alışveriş merkezleri, hastane, ofis ve müzelerde yer, yön tayin etmek ve insanları yönlendirmek için kullanılırken, büyük üretim ve depo sistemlerinde kontrolü sağlamak, ürün yerlerini zaman kaybetmeden hızlı bir şekilde belirlemek için kullanılmaktadır. Büyük üretim merkezleri ve fabrikalarda, kapalı alan konumlandırma sistemlerinin getirmiş olduğu kolaylıkla, ürün yerleri zaman kaybedilmeden, hızlı bir şekilde belirlenebilmekte ve zaman tasarrufunun getirmiş olduğu avantaj sebebiyle üretim hızı da artmaktadır. Kapalı alan konumlandırma sistemleri, büyük alışveriş merkezlerinde insanların ulaşmak istedikleri mağazaları zaman kaybetmeden, hızlı ve kolay bir şekilde bulmalarını sağlarken, bu sistemler, hastanelerde hastaların ve ziyaretçilerin doktor odalarını, test odalarını hızlı bir şekilde bulmalarını sağlayarak hastalara büyük kolaylıklar sağlamaktadır. Ayrıca kapalı alan konumlandırma sistemleri, büyük hastanelerde tıbbi ekipmanların yerlerini belirlemek ve çalışanların acil bir ihtiyaç durumunda zaman kaybetmeden gerekli tıbbi ekipmanları kolay bir şekilde bulabilmeleri için de yardımcı olmaktadır. Özellikle acil hastalar için gereken bu

ekipmanların zaman kaybedilmeden yerlerinin tespit edilerek hızlı bir şekilde kullanılması, hastalar için hayati önem taşımaktadır. Ayrıca büyük hastanelerde, hasta odalarının, kapalı alan konumlandırma sistemleri ile belirlenmesi sayesinde, hayati risk taşıyan hastalara acil müdahale gerektiği durumlarda, sağlık personellerinin zaman kaybetmeden hızlı bir şekilde hasta odasına ulaşmaları da kapalı alan konumlandırma sistemlerinin sağlamış olduğu bir avantajdır. Kapalı alan konumlandırma sistemleri havaalanları, tren istasyonları gibi özellikle insanlar için zamanın çok önemli olduğu yerlerde, herhangi bir ihtiyaç halinde en yakın danışma ve görevliye kolay bir şekilde ulaşma ve ihtiyaçlarını zaman kaybetmeden, hızlı bir şekilde karşılama imkânı sunmaktadır. Kapalı alan konumlandırma sistemlerinin bir başka kullanım alanı olan müzeler içerisinde sergilenen eserlerin yanına yaklaşınca, bu nesneler hakkında otomatik olarak bilgi verilmesi gibi ihtiyaçları karşılayacak altyapıların temelinde de kapalı alan konum tespiti yer almaktadır. Günümüzde özellikle gelişmiş ülkelerde ofisler, akıllı ofislere dönüştürülmektedir. Normal ofisleri, akıllı ofis yapan şey aslında, o ofislerde farklı ve insan hayatını kolaylaştıran teknolojilerin kullanılmasıdır. Bu teknolojilerden birisi de kapalı alan konumlandırma sistemleridir. Gelişmiş ülkelerde büyük ofislerde artık, kapalı alan konumlandırma sistemlerinin kullanılmasıyla, çalışanlar ve ziyaretçiler gitmek istedikleri birimleri, yerleri bu sistemler sayesinde kolaylıkla bulabilmektedirler. Öte yandan, büyük ofislerde, kullanılan araç ve ekipmanların kolay ve hızlı bir şekilde bulunabilmesi de, o ekipmanların konumlarının belirlenmesi için geliştirilen kapalı alan konumlandırma sistemleriyle mümkün olmaktadır. Günümüzde artık gelişmiş büyük fabrika ve firmalar, gerekli ekipman, ürün ve stok kontrollerinin yapılabilmesi için kapalı alan konumlandırma sistemlerini kullanmaktadırlar. Böylece hem üretim maliyeti hem de zaman tasarrufunun getirmiş olduğu üretim verimliliği arttığı için, bu fabrikaların kar payları da artmaktadır. Kapalı alan konumlandırma sistemlerinin kullanıldığı bir başka yer ise çok katlı kapalı otoparklardır. Bu otoparklarda, insanların araçlarını bıraktıkları yeri unutabilme ihtimaline karşı geliştirilen kapalı alan konumlandırma sistemleri sayesinde, insanlar araçlarını hızlı ve kolay bir şekilde bulabilmektedirler. Ayrıca, müze, hastane ve alışveriş merkezlerinde görme engelli insanları yönlendirmek için de kapalı alan konumlandırma sistemleri kullanılmakta ve bu sistemler görme engelli insanlara büyük kolaylıklar sağlayarak, onların da sosyal hayata katılmalarına ve ihtiyaçlarını karşılamalarına sebep olmaktadır (Kirbaş ve Arslan, 2020). Kapalı alan konumlandırma sistemlerinin sivil alanlardaki kullanım alanları göz önüne alındığında, günümüzde bu sistemlerin artık insan hayatının hemen her alanında kullanılmaya başlandığı ve insanların işlerini ne kadar kolaylaştırdığı açık bir şekilde görülmektedir.

Kapalı alan konumlandırma sistemleri sivil hayatın yanı sıra, savunma sanayiinde de çok yaygın olarak kullanılmaktadır. Savunma sanayi, bir toplumun, dışarıdan gelebilecek ve ülke güvenliğini tehdit eden her türlü saldırılara karşı kendini savunması ve bu tehditleri püskürtmesi için gerekli olan sistemlerin, teknolojilerin geliştirilmesini sağlayan bir sektördür (Baran, 2018). Bu nedenle savunma sanayi, olası bir tehdit anında ülkenin güvenlik ve huzurunu koruma anlamında hayati önem taşımaktadır. Kapalı alan konumlandırma sistemleri savunma sanayinde birçok yerde kullanılmakta ve çok kritik roller üstlenmektedir. Kapalı alanlarda gerçekleştirilen askeri operasyonlarda, uydulardan gönderilen sinyaller, çatılar, duvarlar ve görüş hattını kapatan diğer engeller sebebiyle küresel konumlandırma sistemine ulaşamadığı için kapalı alanlarda KKS kullanılamamakta ve bu nedenle, kapalı alanlarda askerlerin bulundukları konumların belirlenebilmesi için kapalı alan konumlandırma sistemlerine ihtiyaç duyulmaktadır. Askeri operasyonlarda, kapalı alanlarda keşif ve gözetleme amaçlı otonom insansız hava araçları kullanılmaktadır. İnsansız hava araçları, bu alanlarda görevlerini yerine getirmek için otonom navigasyon sistemlerini kullanmaktadır. Ancak otonom navigasyon sisteminin kullanılabilmesi için insansız hava araçlarının başlangıç konum bilgisi gerekmektedir. Bu nedenle, başlangıç konum bilgisi elde edilemediğinde, otonom navigasyon sistemi de çalışmamakta ve bu durumda insansız hava aracı için tanımlanan keşif, gözetleme ve istihbarat gibi çok önemli görevler yerine getirilememektedir. Bu nedenle, insansız hava araçlarının otonom navigasyonu için konumlandırma sistemlerine ihtiyaç duyulmaktadır. Ayrıca, küresel konumlandırma sisteminin kullanılamadığı kapalı alanlarda gerçekleştirilen askeri operasyonlarda kullanılan, konumlandırma sistemleri sayesinde, askerlerin operasyon bölgesinde bulundukları konumların tespit edilmesi ve böylece birbirlerinin konumlarını bilmeleri sayesinde operasyon hakkında çeşitli kararların alınması ve gerektiğinde operasyon stratejilerinin hızlı bir şekilde değiştirilebilmesi sağlanmaktadır. Böylece, daha başarılı operasyonlar gerçekleştirilebilmektedir. Ayrıca uydu sinyallerinin kolay bir şekilde KKS'ye ulaştığı ve bu nedenle KKS'nin kullanılabildiği açık alanlarda gerçekleştirilen askeri operasyonlarda dahi, askerlerin konum bilgisinin KKS kullanılarak elde edilmesi bazı tehlike ve dezavantajı da beraberinde getirmektedir. Küresel konumlandırma sistemlerinde, uydu sinyalleri kullanılarak konum belirlendiği için, olası bir savaş anında, uydu sinyallerinin kesilmesi veya karıştırılması halinde, KKS ile konum bulunamayacağından, askeri operasyon esnasında askerlerin ihtiyaç duyduğu konum bilgisi elde edilemeyecektir. Bu durum, operasyon başarısını ciddi şekilde azaltmakta ve ayrıca ciddi risk ve tehditler oluşturmaktadır. Bu nedenle, uydu sinyallerine ve küresel konumlandırma sistemlerine

ihtiyaç duymayan konumlandırma sistemlerinin kullanımı, tüm bu risk ve tehditlerin önlenerek daha başarılı askeri operasyonların gerçekleştirilebilmesi için hayati önem taşımaktadır.

Kapalı alanlarda konum belirleme sistemleri, temelde sensörler ve konum belirlemek için kullanılan algoritmalar olmak üzere iki kısımdan oluşmaktadır. Bu nedenle, bu bölümde, kapalı alanlarda konumlandırma yapmak için kullanılan sensörler ve algoritmalar özelinde literatür taraması yapılarak, kullanılan yöntemlerin birbirlerine göre avantaj ve dezavantajları incelenmiştir.

Kapalı alan konumlandırma sistemlerinde çeşitli sensörler kullanılmaktadır. Bu sensörler, çeşitli kısıtlara göre sınıflandırılabilir. Hayward ve diğerleri kapalı alanda konum bulma problemi için kullanılan sensörleri temelde kablosuz RF sensörler, akustik sensörler ve görüntü tabanlı sistemler (İng. *Vision Systems*) olarak gruplandırmışlardır. Kablosuz RF sensörleri RFID, WiFi, *bluetooth*, *zigbee* ve UWB sensörü olarak sınıflandırmışlardır. Akustik sensörleri, ultrasonik sensörler ve ses sensörleri olmak üzere iki gruba ayırırken, görüntü tabanlı sistemleri ise kamera, infrared sensörler ve lazerler olmak üzere 3 grupta toplamışlardır (Hayward vd., 2022). Bu tez çalışması kapsamında, kapalı alanda konum belirleme problemlerinde kullanılan sensörler, kablosuz RF sensörler ve akustik sensörler olmak üzere 2 ana grupta incelenmiştir. Kablosuz RF sensörler, RFID, WiFi, *bluetooth* sensörü ve UWB sensörü olarak gruplandırılabilir.

RFID tabanlı kapalı alan konumlandırma sistemlerinde, radyo dalgası kullanılmaktadır. RFID sensörü, RFID okuyucu ve RFID etiketinden oluşmaktadır. RFID okuyucular, konumlandırma yapılacak iç mekânda, koordinatları bilinen referans noktalara yerleştirilmekte ve RFID etiket ise konumu tespit edilmek istenen nesne üzerinde bulunmaktadır. Üzerinde RFID etiket bulunan nesne, okuyucuların kapsama alanına girdiğinde, RFID etiket aktif hale gelmekte ve okuyuculara radyo dalgası iletmektedir. Hedef nesne üzerinde bulunan RFID etiket tarafından gönderilen RF sinyali, RFID okuyucular tarafından algılandığında, alınan sinyal şiddeti değeri elde edilmektedir (Bouet ve Dos Santos, 2008). Elde edilen RSS değeri, mesafe verisine dönüştürülüp, ileride anlatılacak olan multilaterasyon algoritmasında kullanılarak, hedef nesnenin konumu tespit edilmektedir (Costin ve Teizer, 2015). Ayrıca, RSS verisi literatürdeki bir başka yöntem olan parmak izi algoritmasında kullanılarak da konum tespiti yapılabilmektedir (Belhadi ve Fergani, 2014). RFID tabanlı konumlandırma sistemlerinde, az sayıda cihaz kullanılarak yüksek kapsama alanı elde edilebilmektedir. Kapalı alanlarda RFID sensörü kullanılarak

tasarlanan konumlandırma sistemlerinin avantajları olarak, kullanılan cihazların masraflarının düşük olması, cihaz bakımlarının kolaylığı, enerji tüketiminin düşük olması gibi özellikler gösterilebilir. Öte yandan, ortamda bulunan nesnelerden dolayı RFID sinyalinin yansıma, kırınım gibi problemlerle karşılaşmasıyla, sinyalin ciddi anlamda zayıflamasının bir sonucu olarak, alınan sinyal şiddeti değerlerinin kararsızlık göstermesi ve sinyalin zayıflaması sonucu konumlandırma doğruluğunun azalması, RFID tabanlı konumlandırma sistemlerinin önemli bir dezavantajıdır (Sanpechuda ve Kovavisaruch, 2008). Montaser ve arkadaşları kapalı bir alanda insanın konumunu bulmak için RFID sensörü ve konumlandırma algoritması olarak ise üçgenleme (İng. *Trilateration*) ve yakınlık (İng. *Proximity*) algoritması kullanmışlardır. Çalışma ortamında önceden konumu bilinen referans noktalara 3 adet RFID okuyucu yerleştirilmiştir. RFID etiket ise, insanın üzerinde bulunmakta ve insan, RFID okuyucuların kapsama alanına girdiğinde, RFID etiket aktif hale gelerek okuyucuya RF sinyali göndermekte ve alınan sinyal şiddeti değeri elde edilmektedir. Bu çalışmada, her 3 RFID sensörden elde edilen, alınan sinyal şiddeti değeri, yol kaybı modeli (İng. *Path Loss Model*) kullanılarak, uzaklık verisine dönüştürülerek üçgenleme algoritmasında kullanılmış ve insanın konumu 1 metre ortalama hata ile bulunmuştur. Alınan sinyal şiddeti değeri yakınlık algoritmasında kullanıldığında ise, 1,9 metre ortalama hata ile insanın konumu tespit edilmiştir. Sonuç olarak, üçgenleme algoritması kullanılarak, yakınlık algoritmasına göre daha yüksek doğrulukla konum tespiti yapılmıştır. Yakınlık algoritmasında, her bir referans RFID okuyucu tarafından iletilen sinyalin alınan sinyal şiddeti değeri toplanmakta ve bu değere göre, insanın hangi okuyucuya daha yakın olduğu bilinmektedir. Bütün okuyuculardan, çeşitli ölçümler sonucu elde edilen, RSS değerleri toplanmakta ve ortalamaları alınarak bir ağırlık çarpanı elde edilmektedir. Ağırlık çarpanı değerine göre insanın hangi okuyucuya daha yakın olduğu bilgisi elde edilmekte ve bu değer büyük olduğunda, insanın o alınan sinyal şiddeti değerinin elde edildiği okuyucuya daha yakın olduğu anlaşılmaktadır. Konum tespiti için ise, her 3 okuyucu için elde edilen ağırlık çarpanı ile okuyucuların konum bileşenlerinin çarpımlarının ortalaması alınarak, insanın bulunduğu konum koordinatları elde edilmektedir (Montaser ve Moselhi, 2014). Başka bir çalışmada Papapostolou ve diğerleri RFID tabanlı kapalı alan konumlandırma sisteminde, RFID sinyalinin ortamda bulunan diğer sinyallerle girişim sorununun, konumlandırma performansı üzerindeki etkisini incelemişlerdir. Gerçekleştirilen çalışmada, multilaterasyon algoritmasını kullanmışlardır. Multilaterasyon algoritmasında çember geometrisi kullanılmaktadır. Konum koordinatları bilinen referans noktalara yerleştirilmiş RFID sensörlerinin bulunduğu konum koordinatları, çizilecek çemberin merkezi, RFID sensör ile

hedef nesne arasındaki uzaklık ise çemberin yarıçapı olarak kabul edilerek, her bir RFID sensör için sanal çemberler çizilmektedir. Çizilen sanal çemberlerin kesişimi sonucu bir kesişim bölgesi oluşmakta ve hedef nesnenin bulunduğu konum koordinatı, bu kesişim bölgesi içerisinde bulunmaktadır. Hedef nesnenin konumunun tespit edilmesi için, çemberlerin kesişimiyle oluşan bu kesişim bölgesi daraltılarak, konum belirlenmektedir. Gerçekleştirilen simülasyonda, sensör ölçümünü, çevresel etkilerden kaynaklanan ve RFID sinyaline bozucu olarak etki eden girişimleri de hesaba katarak modellemişlerdir. Papapostolou ve arkadaşları, sinyale etki eden girişimleri 4 ana başlık altında değerlendirmişlerdir. Bunlar, birden fazla etiketin tek bir okuyucu ile haberleşmesi sonucu oluşan sinyal etkileşimleri, birden fazla okuyucu ile tek bir etiket haberleşmesi sonucu oluşan sinyal etkileşimleri ve birden fazla okuyucu kullanıldığı durumda okuyucuların ilettikleri sinyallerin birbirleriyle olan girişimleridir. Sensör ölçümlerini ise, her bir girişim için farklı şekilde modelleyerek elde edilen ölçümleri, multilaterasyon algoritmasında kullanarak konum tespiti yapmışlardır. Çevresel etkilerden kaynaklanan sinyal girişimlerinin konumlandırma doğruluğunu ciddi seviyede azalttığı ve özellikle birden fazla RFID etiketinin tek bir okuyucu ile haberleşmesi durumunda, konumlandırma doğruluğunun daha fazla azaldığı sonucuna ulaşmışlardır. Bunun sebebi, her bir etiket tarafından iletilen sinyallerin, okuyucuda birbirleriyle girişime uğraması sebebiyle, okuyucu tarafında her bir sensör için elde edilen, alınan sinyal şiddeti değerlerinin hatalı ölçülmesidir. Alınan sinyal şiddeti değeri, uzaklık verisine dönüştürülmekte ve elde edilen uzaklık verileri ise multilaterasyon algoritmasında kullanılarak konum belirlenmektedir. Bu nedenle, alınan sinyal şiddeti değerleri hatalı olduğunda, her bir sensörden elde edilen nesne ile sensör arasındaki uzaklık verisi de hatalı bir şekilde ölçülmektedir. Ölçüm hatası yüksek olan bu uzaklık verileri ise multilaterasyon algoritmasında kullanılarak konum belirlendiğinden, sinyal girişimlerinin etkisiyle, konumlandırma hatası da ciddi şekilde azalmaktadır (Papapostolou ve Chaouchi, 2011).

Kapalı alanda nesne konumlandırmak için kullanılan bir başka yöntem ise WiFi'dır. Cep telefonları ve bilgisayarlar WiFi özellikli oldukları için, WiFi kapalı alanlarda nesne konumlandırmak için sık kullanılan teknolojilerden birisidir. WiFi tabanlı kapalı alan konumlandırma sistemlerinde hedef nesne ile WiFi vericisi arasındaki uzaklık, multilaterasyon algoritmasında kullanılarak konum belirlendiğinden, hedef nesne ile WiFi vericisi arasındaki uzaklık ölçümünün yapılması gerekmektedir. Bu uzaklık ölçümü için ise verici tarafından gönderilen sinyalin alıcıya ulaşma süresi, alınan sinyal şiddeti, varış açısı

gibi farklı ölçüm yöntemleri kullanılmaktadır. Elde edilen uzaklık ölçümlerinin multilaterasyon algoritmasında kullanılmasıyla, hedef nesnenin konumu belirlenmektedir. Ayrıca, WiFi sinyalleri, hedef nesneye ulaştığında elde edilen, alınan sinyal şiddeti değerleri, parmak izi algoritmasında kullanılarak ta konum tespiti yapılabilmektedir. Ancak parmak izi yönteminin bir dezavantajı, referans noktalara yerleştirilen birden fazla WiFi tarafından elde edilen alınan sinyal şiddeti verilerinin, veri tabanında saklanması gerekliliğidir. Bu nedenle, konumlandırma yapmadan önce ortam keşfi yapılarak her bir WiFi modülü için, alınan sinyal şiddeti verilerinin toplanması ve veri tabanında saklanması ek bir maliyet ve zaman gerektirdiğinden, bu durum parmak izi algoritmasının kullanımı için bir dezavantaj oluşturmaktadır. WiFi tabanlı konumlandırma sistemlerinin diğer dezavantajı ise, WiFi sinyalinin kapalı alanda bulunan nesnelerden dolayı kırınım, yansıma ve çok yollu yayılım problemleriyle karşılaştığında, alınan sinyal şiddeti değerinin ciddi şekilde azalmasına bağlı olarak, konumlandırma hatasının önemli ölçüde artmasıdır. WiFi tabanlı sistemlerin bir başka dezavantajı ise, WiFi erişim noktalarının sayısının yeterli olmadığı durumlarda, konumlandırma yapabilmek için yeteri kadar RSS değeri elde edilememektedir. RSS tabanlı konumlandırma yöntemlerinde, yeteri kadar RSS parmak izleri olmadığı durumda, konumlandırma doğruluğu da istenilen seviyede olmamaktadır. Öte yandan, WiFi tabanlı konumlandırma sistemlerinin avantajları ise, WiFi sinyalinin kapsama alanının yüksek olması, birçok yerde WiFi olduğu için kolay erişilebilir olması ve veri aktarım hızının yüksek olmasıdır (Boonsriwai ve Apavatjirut, 2013).

Korogodir ve arkadaşları tarafından WiFi sinyalleri kullanılarak, üçgenleme yöntemiyle kapalı alanda konumlandırma yapılmıştır. Bu yöntemde, WiFi sinyalinin alıcıya varış açısı (İng. *Angle of Arrival*) kullanılmaktadır. Varış açısının kestirimi için de maksimum olabilirlik algoritması (İng. *Maximum Likelihood Algorithm*) kullanmışlardır. Bu algoritma ile WiFi sinyalinin vericiden iletildiği açı ve alıcıya ulaşma açısı belirlenmiştir. Geliştirilen algoritma, WiFi anteninin matematiksel modelini kullanmaktadır. Matematiksel model, antenler için sinyalin faz farkı ile varış açısı arasında bir ilişki oluşturmaktadır. Konumlandırma algoritmasının düşük hata ile konum kestirimi yapabilmesi için, uygun matematiksel model gerekmektedir. Korogodir ve arkadaşları, bu amaçla, antenin matematiksel modeli için iki farklı model karşılaştırmışlardır. Bu modellerden birisi, bağımsız sıfır boyutlu anten modeli ve bir diğeri ise karşılıklı birbirine bağlı antenlerin sonlu boyutlu modelidir. Bağımsız sıfır boyutlu anten modeli, trigonometrik formüllerle oluşturulmuştur. İkinci modelde ise faz-açı ilişkilerinin elde edilmesi için elektromanyetik

simülasyon programı kullanmışlardır. Elektromanyetik simülasyon programı yardımıyla, basit bağımsız anten modeline kıyasla, varış açısında 5-10 dereceye kadar düzeltmelerin tahmin edilebildiği belirtilmiştir. Simülasyon sonuçlarıyla karşılaştırmak için, WiFi alıcı-verici prototipi tasarlanmış ve yapılan deneylerde 3-10 derece arasında değişen varış açısı hataları elde edilmiştir. Bu hatalara sahip ölçümlerin, üçgenleme algoritmasında kullanılması sonucunda, 1 metre hatayla konum belirlenmiştir (Korogodir vd., 2019). Elde edilen sonuç, 3-10 derece gibi küçük açı ölçüm hatalarının bile, konumlandırma doğruluğunu ciddi anlamda düşürdüğünü göstermektedir. Düşük ölçüm hatalarının dahi, üçgenleme algoritmasında yüksek konumlandırma hatalarına yol açması, üçgenleme algoritmasının önemli bir dezavantajıdır. Bu nedenle daha yüksek doğrulukla konumlandırma yapabilmek için, sensörün ölçüm hatasının azaltılması gerektiği ve bunun için de Kalman filtre tabanlı farklı algoritmaların kullanılması gerektiği sonucu ortaya çıkmaktadır.

Bluetooth sensörü tabanlı kapalı alan konumlandırma sistemlerinde, *bluetooth* teknolojisi kullanılmaktadır. *Bluetooth*, cep telefonları, dizüstü bilgisayarlar, tablet gibi çeşitli cihazlarda kullanılan bir teknolojidir. *Bluetooth*'un en son sürümü olan *bluetooth* düşük enerji (İng. *Bluetooth Low Energy*), eski sürümlere kıyasla daha yüksek enerji verimliliği ile 24 Mbps'lik gelişmiş bir veri hızı sağlayabilmektedir (Shorey ve Miller, 2000). *Bluetooth*, RF sinyalleri vasıtasıyla kablosuz haberleşme imkânı sağlamaktadır. *Bluetooth* sensörü RF sinyali yaymaktadır. *Bluetooth* sensörünün kapsama alanındaki mobil cihaz, yayılan RF sinyallerini algılamakta ve sinyalin varış açısı, varış zamanı, alınan sinyal şiddeti gibi çeşitli ölçüm yöntemleri kullanılarak, *bluetooth* sensörü ile kullanıcı arasındaki uzaklık hesaplanmaktadır. Bu uzaklık verisi ise, multilaterasyon algoritmasında kullanılmakta ve nesnenin konumu belirlenmektedir. Diğer sensörlerde olduğu gibi, *bluetooth* sensörü kullanılarak gerçekleştirilen kapalı alan konumlandırma sistemlerinin de bazı avantaj ve dezavantajları bulunmaktadır. *Bluetooth* sensörünün düşük güç tüketimine sahip olması, maliyetinin düşük olması ve WiFi ile karşılaştırıldığında, *bluetooth* sensörü kullanılarak daha yüksek doğruluklarla konum belirlenebilmesi gibi avantajları bulunmaktadır. Ancak, *bluetooth* sensörü tarafından gönderilen sinyalin, konumu tespit edilmek istenen nesneye ulaşırken, kapalı ortamda bulunan nesnelerden dolayı kırınımı, yansıması, çok yollu yayılımı veya diğer RF sinyalleriyle olan girişimi sonucunda, alınan sinyal şiddeti düşmektedir. Alınan sinyal şiddetinin düşmesi sebebiyle, konumlandırma doğruluğunun önemli bir şekilde azalması *bluetooth*, WiFi ve RFID sensörlerinin, konum bulma uygulamalarında

karşılaştıkları çok önemli bir dezavantajdır. Handojo ve arkadaşları, *bluetooth* sensörü kullanarak müzeyi ziyaret eden insanların konumlarını izleyen bir sistem tasarlamışlardır. Müzeye girişte ziyaretçilerden cep telefonlarına müze bilgilerini içeren uygulama indirmeleri istenmiştir. Bu uygulamada, müzede sergilenen eserler hakkında bilgiler bulunmaktadır. Ayrıca uygulama, müzenin haritasını ve ziyaretçilerin bulunmuş oldukları konum bilgilerini de içermektedir. Uygulama ziyaretçilerin yerini bulmak için, müzenin belirli yerlerine yerleştirilmiş olan *bluetooth* sensörlerinin yerlerini tespit etmektedir. Bunun için cep telefonundaki *bluetooth* özelliğini kullanmaktadır. *Bluetooth* sensörler, önceden koordinatları bilinen noktalara yerleştirilmiştir. *Bluetooth* sensörü tarafından gönderilen sinyaller, ziyaretçilerin cep telefonları tarafından algılanmakta ve kullanıcının konumunu belirlemek için referans olarak kullanılmaktadır. *Bluetooth* sensörü tarafından gönderilen sinyal, ziyaretçilerin cep telefonlarındaki *bluetooth* tarafından algılanmaktadır. Cep telefonuna indirilmiş olan uygulama, *bluetooth* sensörünün konumunu belirlemek için sunucuya, *bluetooth* sensörü kullanıcı numara bilgisini göndermekte ve sunucudan *bluetooth* sensörlerin konum bilgileri elde edilmektedir. *Bluetooth* sensörü tarafından gönderilen sinyaller, ziyaretçilerin cep telefonları tarafından algılandığında, alınan sinyalin şiddeti değerleri elde edilmekte ve bu değerler, yol kaybı modeli kullanılarak, *bluetooth* sensörü ile kullanıcı arasındaki uzaklık bilgisine dönüştürülmektedir. Elde edilen uzaklık verilerinin ise, üçgenleme algoritmasında kullanılmasıyla, ziyaretçilerin konumları belirlenmektedir. Bu yöntemde, alınan sinyal şiddeti değeri uzaklık verisine dönüştürülerek trilaterasyon algoritmasında kullanıldığı için, konumlandırma doğruluğunu, alınan sinyal şiddeti değeri oldukça etkilemektedir. Sonuç olarak yapılan testlerde ortalama konumlandırma hatasının 1,14 metre olduğu görülmüştür (Handojo vd., 2019). Başka bir çalışmada Thaljaoui ve arkadaşları, RSS değerlerindeki kararsızlığın sebep olduğu konumlandırma hatasını azaltmak için yeni bir yöntem önermişlerdir. Gerçekleştirilen çalışmada, 3 adet BLE *bluetooth* sensörü ve iRingLA adını verdikleri bir algoritma kullanarak, kapalı ortam konumlandırma sistemi tasarlamışlardır. Bu çalışmada, RSS'in herhangi bir anlık değeri yerine, birden fazla RSS ölçüm değerleri kullanılmıştır. Bunun sebebi, RSS değerlerinin anlık olarak ortam koşullarından etkilenmesidir. Çünkü RF sinyali yayan cihazdan alınan RSS değerleri cihazın statik olmasından, cihazın etrafındaki hareketli nesnelerin varlığından etkilenmektedir. Bu sebepten dolayı, yapılan çalışmada, tek bir anlık RSS değeri yerine, birden fazla RSS ölçümü olan ölçüm setleri kullanılarak ortalama RSS değerinin hesaplanmasıyla, RSS değerlerindeki kararsızlığın azaltılması amaçlanmıştır. Bu sayede, RSS değerlerindeki kararsızlığın konumlandırma performansı üzerindeki negatif etkisinin

giderilmesi amaçlanmıştır. Bu amaçla, en yüksek değere sahip olan 10 adet RSS ölçümlerinin ortalaması alınarak elde edilen RSS değeri kullanılarak, mesafe ölçümü elde edilmiş ve bu mesafe verisi ise iRingLA algoritmasında kullanılarak konumlandırma yapılmıştır. Trilaterasyon yönteminde çember geometrisi kullanılmaktadır. 3 adet çemberin kesişim noktası kullanılarak konum tahmini yapılmaktadır. Bu çalışmada, araştırmacılar alıcıyla *bluetooth* sensörü arasındaki mesafenin kesin olarak doğru bir şekilde hesaplanamayacağını ve bu sebeple trilaterasyon yönteminde kullanılan çemberlerin tek bir noktada kesişmeme veya herhangi bir kesişim noktasının olmaması ihtimali olduğunu savunmuşlar ve trilaterasyon algoritmasına alternatif olarak iRingLA adını verdikleri bir algoritma geliştirmişlerdir. Bu algoritmada çember yerine halka geometrisi kullanılmaktadır. Bu halkaların yarıçapları, ilk aşamada hâlihazırda gerçekleştirilen RSS ölçümlerine göre hesaplanmaktadır. RSS değerlerinin, uzaklık bilgisine dönüştürülmesiyle elde edilen her bir sensör ile nesne arasındaki uzaklıklar, kullanılan halkanın yarıçaplarını ifade etmektedir. Alıcı ilk olarak birbirine en yakın 2 adet halkanın kesişim noktalarını kullanarak, hedef nesnenin bulunduğu konuma dair arama alanı oluşturmaktadır. 2 halkanın kesişim noktaları kullanılarak bir matris oluşturulmakta ve bu matris içerisindeki her bir nokta başlangıçta seçilen iki halkaya göre test edilerek, üçüncü halka için kullanılacak olan noktalar seçilmektedir. Böylece, küçük bir nokta grubu elde edilmektedir. Aslında bu aşamada, iki halkanın kesişim bölgesi daraltılarak, kullanıcının bulunduğu konuma yaklaşılmaya çalışılmaktadır. Daha sonra üçüncü *bluetooth* sensörü için, üçüncü halka çizilerek seçilen noktalar ve üçüncü halkadaki noktaların ağırlık merkezleri bulunmuş ve bu ağırlık merkezinin koordinatları, kullanıcının konumu olarak belirlenmiştir. Yapılan testlerde ortalama 40 cm'lik hatayla konum tespit edildiği belirtilmiştir (Thaljaoui ve Val,2016).

Kapalı alanlarda konum belirlemek için kullanılan bir başka sensör ise UWB sensörüdür. UWB sensörü, veri iletimi yapmak için geniş bant aralığını kullanmaktadır. Geniş bant aralığı kullandığı için, ultra geniş bant sensör sinyali, frekans çeşitliliği açısından oldukça zengindir. Frekans çeşitliliği, UWB sensörüne, WiFi, RFID ve *bluetooth* sensörü sensörlerinde olmayan bazı avantajlar sağlamaktadır. UWB sinyalinin farklı frekans bileşenlerinden oluşması, özellikle kapalı alan konum bulma sistemlerinde bir problem olarak sıkça karşımıza çıkan ve konum belirleme hatasını ciddi şekilde artıran sinyalin çok yönlü yayılımı, yansıması ve sinyal girişimleri gibi problemlerde, UWB sensörüne ciddi avantajlar sağlamaktadır. Frekans çeşitliliğinin UWB sensörüne sağladığı bir başka avantaj

ise, kapalı alanlarda bulunan nesnelerden dolayı, sensörün bazı frekans bileşenlerine sahip sinyalleri, bu nesnelerden geçemese veya nesneler sebebiyle kırınım ve yansıma gibi engellerle karşılaşsa bile, farklı frekanslardan oluşan sensör sinyalleri bu engellerden geçerek konumu tespit edilmek istenen nesnedeki alıcıya ulaşabilmektedir. Alıcıya ulaşan bu sinyaller konum belirlemek için yeterli olmaktadır. Bu nedenle, ortamda bulunan nesnelerin sebep olduğu, sinyal üzerindeki kırınım ve yansıma gibi negatif etkiler, UWB sensörü kullanıldığında daha az hissedilmekte ve bu negatif etkiler konumlandırma doğruluğunu fazla etkilememektedir. Bu nedenle, UWB sensörü, WiFi, RFID ve *bluetooth* sensörü gibi diğer sensörlerin ihtiyaç duyduğu, hedef nesne ile sensör okuyucusu arasında görüş hattına ihtiyaç duymamaktadır. UWB sensörünün görüş hattı ihtiyacı olmaması özellikle kapalı alanlarda konum belirleme uygulamalarında, UWB sensörüne büyük avantaj sağlamaktadır. UWB sinyalindeki frekans çeşitliliğinin başka bir avantajı, UWB sinyallerinin ortamda bulunan diğer RF sinyalleriyle oluşabilecek girişimlere karşı dayanıklılığının bir sonucu olarak konumlandırma doğruluğunun azalmamasıdır. Bazı frekanslara sahip UWB sensör sinyalleri, kapalı alanda bulunan diğer RF sinyallerle etkileşime girse bile, farklı frekanslara sahip UWB sensör sinyalleri, kapalı alanda bulunan diğer RF sinyallerle etkileşime girmemektedir. Diğer RF sinyalleriyle etkileşime giren UWB sensör sinyalleri, sensörde ölçüm hatalarına sebep olmaktadır. Bu ölçüm hataları ise yüksek doğrulukla konumlandırma yapmaya bir dezavantaj oluşturmaktadır. Ancak, ortamda bulunan RF sinyallerinin frekanslarından farklı frekanslara sahip UWB sinyalleri, diğer RF sinyalleriyle etkileşime girmeden konumu tespit edilmek istenen nesneye ulaşmaktadır. Hedef nesneye ulaşan bu UWB sinyalleri, ortamda bulunan diğer RF sinyalleriyle etkileşime girmediği için, bu UWB sinyalleri kullanılarak yapılan ölçümlerde, ölçüm hataları düşük olmaktadır. Ölçüm hataları düşük olduğu için, kapalı ortamdaki diğer RF sinyallerinin varlığı sonucu meydana gelen sinyal girişimleri, ultra geniş bant sensör tabanlı konumlandırma sistemlerinin doğruluğunu azaltmamaktadır. Bu nedenle, diğer RF sinyalleriyle olan etkileşimler, UWB tabanlı konumlandırma sistemleri için herhangi bir dezavantaj oluşturmamaktadır. Benzer şekilde, kapalı ortamda bulunan nesnelerden dolayı sinyalde meydana gelen kırınım, yansıma veya çok yollu yayılım gibi negatif etkilere karşı, UWB tabanlı konumlandırma sistemlerinin diğer WiFi, RFID ve *bluetooth* tabanlı konumlandırma sistemlerine kıyasla çok daha dayanıklı ve avantajlı olmasının nedeni de yine UWB sensör sinyallerinin farklı frekans bileşenlerinden oluşmasıdır.

Bütün bu avantajlarının yanı sıra, UGB sensörü kullanılarak gerçekleştirilen kapalı alan konumlandırma sistemlerinin bazı dezavantajları da bulunmaktadır. Bu dezavantajlar, UGB sensör ekipmanlarının, diğer WiFi, *bluetooth* sensörü ve RFID sensörleriyle karşılaştırıldığında daha pahalı olmasıdır. Ayrıca UGB sensörü, diğer sensörlerle karşılaştırıldığında, her ne kadar kapalı alanlarda bulunan nesnelerin sebep olduğu kırımın gibi sorunlara karşı daha dayanıklı olsa da, ortamda bulunan metal malzemelerin sebep olduğu kırımın problemleriyle, UGB sensör sinyalleri de karşılaşmaktadır. Ancak, sinyal girişimi, kırımını ve çok yollu yayılım gibi problemler düşünüldüğünde, UGB sensörü, diğer sensörlerle karşılaştırıldığında çok daha avantajlı olduğu görülebilmekte ve UGB sensörünün kullanılmasıyla, kapalı alanlarda daha yüksek doğruluklarla konum belirlenebilmektedir (Liu vd., 2007). Gharat ve arkadaşları, kapalı alanda konum belirleme problemindeki, çok yollu yayılım ve görüş hattının olmama durumunda, UGB ve RFID sensörlerinin performanslarını karşılaştırmışlardır (Gharat vd., 2017). Bu amaçla, sensör okuyucuları kapalı alanda önceden konum koordinatları bilinen referans noktalara yerleştirilmiştir. RFID sensörü kullanılarak konum tahmini yapmak için, RSS ölçümleri, uzaklık verisine dönüştürülüp, trilaterasyon algoritmasında kullanılarak hedef nesnenin konumu belirlenmiştir. RSS ölçümleriyle, hedef nesne ve okuyucu arasındaki uzaklığı hesaplamak için, RFID verici tarafından gönderilen sinyalin RSS değeri ve sinyalin RFID alıcıya ulaştığında elde edilen RSS değerleri arasındaki fark kullanılarak, sinyalin ne kadar zayıfladığı bilgisi elde edilmiştir. Elde edilen bu verinin yol kaybı modelinde kullanılmasıyla, RFID okuyucu ile hedef nesne arasındaki uzaklık elde edilmiştir. Ultra geniş bant sensör okuyucuları ile hedef nesne arasındaki uzaklık ölçümü için ise iki yönlü ölçme (İng. *Two Way Ranging*) yöntemi kullanılmıştır. İki yönlü ölçme yönteminde, UGB okuyucularının kapsama alanına giren hedef nesne üzerinde bulunan UGB etiketi aktif hale gelerek, okuyuculara sinyal göndermekte ve etikette bulunan kronometre çalışmaya başlamaktadır. Okuyucular sinyali aldıktan sonra belli bir işlem zamanı (İng. *Processing Delay*) sonrası bu sinyali tekrar, UGB etikete göndermekte ve etiket, sinyali aldığı anda kronometreyi durdurmaktadır. Böylece UGB etiket tarafından okuyucuya gönderilen sinyalin, tekrar etikete dönme süresi belirlenmektedir. Bu süreden, okuyucudaki işlem süresinin çıkarılıp, elde edilen sürenin yarısı alınarak sinyalin uçuş süresi (İng. *Time of Flight*) elde edilmektedir. Sinyal ışık hızıyla hareket ettiği için, elde edilen uçuş süresinin ışık hızıyla çarpımı sonucu, UGB etiket ile hedef nesne arasındaki uzaklık elde edilmektedir (W. Zhang vd., 2020). UGB sensörü kullanılarak elde edilen uzaklık ölçümleri de trilaterasyon algoritmasında kullanılarak konum tespiti yapılmıştır. Elde edilen sonuçlarda,

UGB sensörü ve trilaterasyon algoritması kullanılarak yapılan konum belirlemede ortalama 0,58 metre, RFID sensörü ve trilaterasyon algoritması kullanıldığında ise ortalama 3,78 metre hata ile konum belirlendiği görülmüştür (Gharat vd., 2017). Ulaşılan sonuçlar, görüş hattının olmadığı ve çok yollu yayılım sorunlarına sebep olan kapalı alanlarda, UGB sensörünün kullanılmasıyla, RFID sensörüne kıyasla çok daha yüksek doğrulukla konum belirlenebildiğini göstermektedir. Bunun nedeni, UGB sensörünün geniş bant aralığına sahip olması sebebiyle, farklı frekanslara sahip sinyal bileşenleri içermesidir. Başka bir çalışmada, Retscher ve arkadaşları laboratuvar ortamında konum belirleme probleminde, WiFi ve UGB sensörlerinin performanslarını karşılaştırmışlardır. Çalışmada, WiFi ile konumlandırma yapmak için 3 adet WiFi modülü ve UGB sensörü ile konumlandırma yapmak için ise 5 adet UGB sensörü kullanmışlardır. WiFi ile konum belirlemek için alınan sinyal şiddeti ölçümleri yapmışlar ve her bir WiFi modülü için elde edilen alınan sinyal şiddeti değerlerini, yol kaybı modeli kullanarak uzaklık bilgisine dönüştürerek elde edilen uzaklık verilerini trilaterasyon algoritmasında kullanarak konum tespiti yapmışlardır. UGB sensörü kullanarak konum tespiti yapmak için de genişletilmiş Kalman filtre algoritması kullanmışlardır. Elde edilen sonuçlar incelendiğinde, WiFi ve trilaterasyon algoritması kullanılarak 2,65 metre ortalama hata ile konum tespiti yapılırken, UGB sensörü ve genişletilmiş Kalman filtre algoritması kullanılarak ise 0,76 metre ortalama hatayla konum belirlendiği görülmektedir. Retscher ve arkadaşlarının elde ettikleri sonuçlar, UGB sensörünün kullanılmasıyla, WiFi tabanlı konum belirleme sistemine kıyasla, çok daha yüksek doğrulukla konum belirlenebildiğini göstermekte ve ayrıca Kalman filtre algoritmasının kullanılması da avantajını ortaya koymaktadır (Retscher vd., 2019). Bir başka çalışmada Botler ve arkadaşları, UGB sensörü ile *bluetooth* sensörünün, kapalı alanda görüş hattının olduğu, ancak çok yollu yayılım sorununun olduğu durumda ve görüş hattının azaldığı durumlarda, her iki sensörün ölçüm hatalarını, her bir durum için ayrı ayrı karşılaştırmışlardır. Her iki sensör için de, ölçüm metodu olarak varış açısı (İng. *Angle of Arrival*) yöntemi kullanılmıştır. Varış açısı yönteminde, sensör vericisi tarafından gönderilen sinyal, alıcıda bulunan anten yardımıyla algılanmaktadır. Hem vericinin alıcıya gönderdiği sinyalin fazı, hem de alıcının aldığı sinyalin fazı verici ve alıcı tarafından belirlenmektedir. Böylece, alıcının aldığı sinyalin fazı ile verici tarafından gönderilen sinyalin fazı belli olduğu için bu sinyaller arasındaki faz farkı belirlenebilmektedir. Elde edilen faz farkı bilgisiyle, sinyalin dalga boyu ve sensör vericisi ile alıcısı arasındaki uzaklık verileri, varış açısı hesabı için kullanılan matematiksel denklemde kullanılarak, sinyalin varış açısı bulunabilmektedir (Botler vd., 2020). Görüş hattının olduğu ve çok yollu yayılım sorununun olduğu durumda, UGB vericiden gönderilen

sinyalin, UGB alıcıya varış açısının, ortalama kare hatanın karekökü (İng. *Root Mean Square*) cinsinden ölçüm hatası $36,8^\circ$ olarak elde edilirken, *bluetooth* sensörü için ise $41,9^\circ$ olarak elde edilmiştir. Sensörlerin alıcı ve vericileri arasında engel olarak insan bulunduğu ve bu sebeple görüş hattının azaldığı durumda, UGB vericiden gönderilen sinyalin, UGB alıcıya varış açısının, ortalama kare hatanın karekökü cinsinden ölçüm hatası $8,7^\circ$ olarak elde edilirken, *bluetooth* sensörü için ise bu hata $18,9^\circ$ olarak elde edilmiştir. Elde edilen sonuçlar, görüş hattı olmama ve çok yollu yayılım sorunun olduğu durumlarda, UGB sensörünün ölçüm hatasının, *bluetooth* sensörü sensörünün ölçüm hatasından daha az olduğunu göstermektedir. Bunun sebebi, UGB sinyalinin, farklı frekans bileşenlerinden oluşmasıdır. Çünkü görüş hattı olmadığı durumlarda, UGB sensör sinyallerinin bazıları engellerden geçemese bile, farklı frekanslara sahip sinyaller engellerden geçerek UGB alıcıya ulaşmakta ve böylece sensör ölçümü gerçekleşmektedir. Bu nedenle UGB sensörünün ölçüm hatası, *bluetooth* sensörü ile karşılaştırıldığında daha az olmaktadır. Aynı şekilde, çok yollu yayılım sorunu olduğunda da yine, UGB sensörüne geniş bant aralığından dolayı sinyal frekans çeşitliliğinin sağlamış olduğu avantajdan dolayı UGB sensörünün ölçüm hatası, *bluetooth* sensörü ölçüm hatasından daha küçüktür. UGB sensöründeki farklı frekanslara sahip sinyallerden bazıları çok yollu yayılım sorunu sebebiyle kullanılamasa ve yüksek ölçüm hatalarına sebep olsa bile, farklı frekanslara sahip sinyaller çok yollu yayılım sorunuyla karşılaşmamaktadır. Bu sinyaller alıcıya ulaşarak sensör ölçümü gerçekleştiği ve bu sensörler çok yollu yayılım sorunuyla karşılaşmadığı için UGB sensörünün ölçüm hatası, *bluetooth* sensörü sensörünün ölçüm hatasından daha küçük olmaktadır (Botler vd., 2020).

Kapalı alanda konum belirlemek için kullanılan bir başka sensör ise akustik sensörlerdir. Akustik sensörler, verici ve alıcıdan oluşmaktadır. Verici (İng. *Transmitter*), konumu tespit edilmek istenen hedef nesne üzerinde bulunmakta ve sensör alıcısı (İng. *Anchor*) ise kapalı alanda daha önceden konum koordinatları bilinen referans noktalara yerleştirilmektedir. Sensör vericisi tarafından iletilen sinyalin, üzerinde sensör alıcısı bulunan hedef nesneye varma süresi kullanılarak, akustik sensörle hedef nesne arasındaki uzaklık hesaplanmaktadır. 2 boyutlu koordinat sistemi üzerinde konum belirlemek için üç adet sensöre ihtiyaç duyulmakta ve her bir sensör için elde edilen uzaklık verileri, trilaterasyon algoritmasında kullanılarak hedef nesnenin konumu tespit edilmektedir. Akustik sensörler, ultrasonik sensörler olarak ta adlandırılmaktadır. Ultrasonik sensörler ile ses dalgaları kullanılarak hedef nesne ve ultrasonik sensör alıcısı arasındaki uzaklık ölçümü yapılmakta ve elde edilen uzaklık ölçümleri, trilaterasyon algoritmasında kullanılarak konum tespit edilmektedir.

Ultrasonik sensörler ile uzaklık ölçümü yapmak için genellikle, iki yönlü ölçme, alınan sinyal şiddeti, varış zamanı farkı ve varış zamanı gibi ölçüm yöntemleri kullanılmaktadır. 2 boyutlu koordinatta konum belirlemek için, 3 adet ultrasonik sensöre ihtiyaç duyulmakta ve her bir sensör ile hedef nesne arasındaki uzaklık ölçümleri trilaterasyon algoritmasında kullanılarak, hedef nesnenin konumu tespit edilmektedir. Ultrasonik sensörlerin bant aralığı dar olduğundan, bu sensörlerin bazı dezavantajları vardır. Dar bant aralığının bir özelliği olarak, ultrasonik sensörlerin veri iletim hızları düşüktür. Ayrıca, ses dalgaları, sinyal girişimlerine karşı RF dalgalarla karşılaştırıldıklarında çok daha zayıftırlar. Ultrasonik sensörler, kapalı alanda bulunan nesnelerden dolayı meydana gelen, çok yollu yayılım, sinyalin kırınım ve yansıması gibi problemlere karşı da çok hassas olup, bu problemleri tolere edememeleri sonucunda, ciddi ölçüm hataları meydana gelmektedir. Bu nedenle, RF tabanlı konumlandırma sistemleriyle karşılaştırıldığında, ultrasonik sensörler kullanılarak gerçekleştirilen kapalı alan konumlandırma sistemlerinin hataları, çok daha yüksektir. Ultrasonik sensörlerin avantajlarına bakıldığında ise, ucuz oldukları ve düşük güç tüketimine sahip oldukları görülmektedir (Hazas ve Hopper, 2006).

Kapalı alan konumlandırma sistemlerinde karşılaşılan bazı zorluklar bulunmaktadır. Bu zorluklar sinyal girişimleri, sinyalin çok yollu yayılımı ve sensörle konumu tespit edilmek nesne arasında görüş hattı olmama durumlarıdır. Kapalı alanlarda konum belirleme problemlerinde karşılaşılan bu zorluklar sebebiyle, konumlandırma doğruluğu düşmektedir.

Kapalı alanlarda çeşitli sinyal kaynakları bulunmakta ve bu sinyaller, konum belirlemek için kullanılan sensör sinyalleriyle girişime sebep olabilmektedir. Kapalı alanlarda bulunabilecek sinyal kaynaklarına örnek olarak WiFi, *bluetooth* ve hücreli ağlar verilebilir. Konum belirlemek için kullanılan sensör sinyalinin frekansı, kapalı ortamda bulunan diğer sinyallerin frekansı ile aynı olduğunda, sinyal girişimleri meydana gelmektedir. Bu durumda, konum belirlemek için kullanılan sensör sinyalinin karşılaştığı girişim, sensör ölçümüne gürültü olarak etki ettiği için, sensör ölçümünde hatalar meydana gelmektedir. Bu ölçümler, konumlandırma algoritmalarında kullanıldığı için de sensör ölçümlerinde meydana gelen bu hatalar, konum belirleme hatalarına sebep olmaktadır.

Kapalı alanlarda konum belirleme problemlerinde karşılaşılan bir diğer zorluk ise konum belirlemek için kullanılan sensör sinyalinin, çok yollu yayılım problemiyle karşılaşmasıdır. Çok yollu yayılım problemi, sensör sinyalinin kapalı alanda bulunan nesnelerden dolayı yansıması sonucu meydana gelmektedir. Sensör sinyali kapalı alanda bulunan nesnelerden yansıdığı zaman, sinyal birden fazla yoldan alıcıya ulaşmakta ve bu durumda ölçüm hataları

meydana gelmektedir. Örnek olarak, kullanılan sensörün varış zamanı yöntemiyle hedef nesne ve sensör okuyucusu arasındaki uzaklığı ölçtüğü düşünülürse, sinyal çok yollu yayılıma uğradığında, vericiden iletilen sinyalin, alıcıya ulaşma süresi değişeceğinden, ölçüm hataları meydana gelecektir. Bir başka örnek olarak, vericiden gönderilen sinyal birden fazla yoldan alıcıya ulaştığında, sinyalde faz kayması meydana gelmekte ve özellikle varış açısı ölçümlerinde, faz farkı kayması ciddi ölçüm hatalarına sebep olmaktadır. Ayrıca, sinyal çok yollu yayılıma uğradığında, alınan sinyal şiddeti de değişeceği ve hatalı olacağı için, RSS ölçümü kullanılarak gerçekleştirilen konum bulma problemlerinde, yüksek hatalar meydana gelmektedir.

İç mekânlarda konum belirleme sürecinde karşılaşılan bir başka sorun, sensör vericisi ile konumu tespit edilmek istenen hedef nesne arasında görüş hattının zayıflaması veya tamamen kaybolmasıdır. Sensör ile konumu tespit edilmek istenen nesne arasında görüş hattı olmaması aslında, sensör ile nesne arasında başka objelerin olmasıdır. Bu durumda, sensör tarafından nesneye gönderilen sinyal, ortamda bulunan objelere çarptığında, direk olarak nesneye ulaşamamakta ve sinyalin bir kısmı yansımakta, bir kısmı kırılmaktadır. Bu durumda, sensör vericisi tarafından gönderilen sinyalin, üzerinde sensör alıcısı bulunan nesneye varış açısı, varış zamanı veya alınan sinyal şiddeti değerleri değiştiği için, ölçüm hataları meydana gelmekte ve hedef nesne ile sensör arasındaki uzaklık ölçümleri hatalı olmaktadır. Bu durumda, elde edilen uzaklık ölçümleri veya RSS ölçümleri, multilaterasyon veya parmak izi algoritması gibi konumlandırma algoritmalarında kullanılarak konum belirlendiği için, konumlandırma hataları meydana gelmektedir.

Kapalı alanlarda yaşanan bu zorluklar göz önüne alındığında, yüksek doğrulukla konumlandırma yapabilmek için, sensör ve algoritmaların doğru bir şekilde seçilmesi önem taşımaktadır. Bu nedenle literatürde, sensör seçimi için bazı metrikler tanımlanmıştır (Alarifi vd., 2016). Bu metrikler doğruluk, kapsama alanı, maliyet ve güç tüketimidir. Doğruluk, ölçüm hatasının ne kadar düşük olduğunun, kapsama alanı sensör sinyalinin ne kadar mesafeye kadar yayılabildiğinin ve güç tüketimi ise sensörün ne kadar enerji tükettiğinin bir göstergesidir.

Kapalı alanda konum belirlemek için kullanılan RFID, *bluetooth*, WiFi, ultrasonik sensör ve UGB sensörleri doğruluk açısından karşılaştırıldığında, doğruluğu en yüksek olan sensör UGB ve ultrasonik sensördür. Ancak ultrasonik sensör çok kısa kapsama alanı içerisinde ve küçük alanlarda yüksek konumlandırma doğruluğuna sahip olup, konumlandırma alanı genişledikçe doğruluk ciddi şekilde azalmaktadır. Bunun sebebi, ultrasonik sensörün dar

bant aralığına sahip olmasıdır. Ancak, UGB sensörünün kapsama alanı yaklaşık 150 metreye kadar çıkmakta ve geniş bant aralığının sunmuş olduğu frekans çeşitliliğinin bir avantajı olan kapalı alanlardaki engeller sebebiyle sinyalin kırınım, yansıma ve çok yollu yayılım sorunlarıyla karşılaşmadığı için, UGB sensörü geniş alanlarda da yüksek doğruluğa sahiptir (Hazas ve Hopper, 2006).

Sensörler kapsama alanı açısından karşılaştırıldığında, en geniş kapsama alanına sahip olan sensörler UGB ve WiFi'dır. Ancak WiFi kullanılarak, RSS ölçümleri kullanılarak konum belirlenmektedir. Kapalı alanlarda vericiden iletilen sinyal kırınım, çok yollu yayılım gibi sorunlarla karşılaştığı için zayıflamakta ve bu nedenle RSS ölçümleri hatalı olmaktadır. Bunun sonucunda, WiFi tabanlı kapalı alan konumlandırma sistemlerinin hatası, UGB tabanlı konumlandırma sistemlerine kıyasla daha fazla olmaktadır (Deak vd., 2012).

Sensörler güç tüketimi açısından karşılaştırıldığında, RFID, *bluetooth* sensörü, UGB ve ultrasonik sensörlerin güç tüketimi, WiFi'ya göre daha düşüktür (Deak vd., 2012).

Maliyet açısından karşılaştırıldığında ise UGB, WiFi ve ultrasonik sensörlerin maliyeti orta, RFID ve *bluetooth* sensörü ise daha düşük maliyete sahiptir (Deak vd., 2012).

Kapalı alanda nesne konumlandırma problemlerinin çözümü için, bazı algoritmalar kullanılmaktadır. Konumlandırma algoritmaları, bu literatür taraması kapsamında, parmak izi algoritması, yakınlık algoritması, multilaterasyon tabanlı algoritmalar ve Kalman filtre tabanlı algoritmalar olarak sınıflandırılmıştır. Multilaterasyon tabanlı algoritmalar ise sensör ölçüm yöntemlerinin nasıl kullanıldığına bağlı olarak, 3 farklı grupta incelenebilmektedir. Bunlar, üçgenleme, trilaterasyon ve multilaterasyon algoritmalarıdır (Papapostolou ve Chaouchi, 2011). Sensör ölçüm yöntemleri ise varış zamanı kullanarak uzaklık ölçümü, iki yönlü ölçme yöntemi ile uzaklık ölçümü, varış açısı yöntemi ve alınan sinyal şiddeti ölçümleridir. Sensör ölçümleri, ölçümler sonucu elde edilen verilerin türlerine bağlı olarak farklı algoritmalarda kullanılarak konum belirlenmektedir. Örneğin, sensör ölçümleri sonucu sadece uzaklık verisi elde edildiğinde, bu uzaklık verileri multilaterasyon veya trilaterasyon algoritmalarında kullanılarak konum belirlenmektedir. Öte yandan, sensör ölçümleri ile sinyalin varış açısı ölçümleri elde edildiğinde, bu defa üçgenleme algoritması kullanılarak konum belirlenmektedir. Eğer sensör ölçümleri sonucu alınan sinyal şiddeti ölçümleri kullanılırsa, parmak izi algoritması veya yakınlık algoritması kullanılarak konum belirlenmektedir. Sonuç olarak, sensör ölçümü sonucu elde edilen veriye ve konum belirlemek için bu verilerin işleme yöntemlerine göre farklı konumlandırma algoritmaları

kullanılmaktadır. Bu nedenle, bu kısımda, sensör ölçüm yöntemleri açıklandıktan sonra konumlandırma algoritmaları incelenmiştir.

Sensör ölçüm yöntemleri, varış zamanı (İng. *Time of Arrival*) yöntemi, iki yönlü ölçme (İng. *Two Way Ranging*) yöntemi, varış açısı yöntemi, varış zamanı farkı (İng. *Time Difference of Arrival*) yöntemi ve alınan sinyal şiddeti yöntemi olarak sınıflandırılabilir.

Varış zamanı yönteminde sensör etiketi tarafından, sensör okuyucusuna gönderilen sinyalin varış zamanı ölçülmekte ve bu varış zamanının kullanılmasıyla sensör okuyucusu ile üzerinde sensör etiketi bulunan hedef nesne arasındaki uzaklık elde edilmektedir. Sensör sinyalinin yayılım hızı ile sinyalin alıcıya varış zamanının çarpımı sonucu, sensör vericisi ile alıcısı arasındaki uzaklık hesaplanmaktadır. İki boyutlu koordinat düzleminde konum belirlemek için üç adet sensör ölçümüne ihtiyaç duyulmaktadır. Her üç sensör, hedef nesnenin kendisine olan uzaklığını ölçmekte ve her üç sensör ölçümü sonucu elde edilen üç adet uzaklık verisi, trilaterasyon algoritmasında kullanılarak konum belirlenebilmektedir. Varış zamanı yöntemi kullanılarak, her üç sensörün uzaklık ölçümü yaptığı durumlarda, her üç sensör vericisi ile sensör alıcısının senkronize olması gerekir ki, sinyalin varış zamanı doğru bir şekilde ölçülebilsin. Bundan dolayı, sensörler arasında senkronizasyon olmadığı durumlarda, varış zamanı yöntemiyle uzaklık ölçümleri hatalı olmaktadır. Bu durum varış zamanı yöntemiyle uzaklık ölçümünün bir dezavantajıdır. Çünkü senkronizasyon problemi, kapalı alan konumlandırma sistemleri için ciddi bir problemdir ve sensör alıcısı ile vericisi arasındaki saatlerdeki (kronometre) 1 nano saniyelik zaman farkı bile, kapalı alan konumlandırma sistemleri için ciddi ölçüm hatasına sebep olmaktadır. Varış zamanı yöntemi ile elde edilen uzaklık ölçümleri trilaterasyon veya multilaterasyon algoritmalarında kullanılarak hedef nesnenin konumu tespit edilmektedir.

Bir başka uzaklık ölçüm yöntemi, iki yönlü ölçme yöntemidir. İki yönlü ölçme yönteminde, üzerinde sensör vericisi bulunan nesne, sensör alıcısının kapsama alanına girdiğinde aktif hale gelerek alıcıya sinyal göndermekte ve kronometresini başlatmaktadır. Sensör alıcısı, sinyali aldığı anda belli bir işleme süresi sonunda, sinyali tekrar vericiye iletmekte ve verici kendi sinyali geldiğinde kronometresini durdurmaktadır. Böylece sinyalin alıcıya gidip tekrar vericiye dönene kadar geçen süre elde edilmektedir. Ancak sinyal, alıcıda işleme zamanından dolayı bir gecikmeye uğradığı için, bu işleme zamanı, sinyalin vericiden çıkıp, tekrar vericiye dönene kadar geçen süreden çıkarılmaktadır. Daha sonra, sinyal vericiden çıkarak alıcıya ulaştığı ve tekrar vericiye geldiği için aynı yolu iki kez almış olmaktadır. Bu nedenle, sinyalin vericiden çıkıp tekrar kendisine dönene kadar geçen süre ile alıcıdaki

işleme süresi arasındaki fark ikiye bölünerek, sinyalin uçuş süresi (İng. *Time of Flight*) elde edilmektedir. Sinyal ışık hızıyla yayıldığı için, elde edilen uçuş süresi ile ışık hızının çarpımıyla da üzerinde sensör vericisi bulunan hedef nesne ile sensör okuyucusu arasındaki uzaklık hesaplanmaktadır. İki yönlü ölçme yönteminde, sensör alıcısı ile vericisinde bulunan saatler (kronometre) arasında senkronizasyon ihtiyacı olmadığı için, varış zamanı, varış zamanı farkı yöntemleriyle karşılaştırıldığında, iki yönlü ölçme yöntemi ile daha doğru uzaklık ölçümleri elde edilmektedir. İki yönlü ölçme yöntemiyle elde edilen uzaklık ölçümleri, trilaterasyon veya multilaterasyon algoritmalarında kullanılarak, hedef nesnenin konumu belirlenmektedir. İki yönlü ölçme yönteminin, sensör vericisi ve alıcısındaki saatlerin senkronize olmasına ihtiyaç duymaması bir avantajdır. Ancak, kapalı alanda bulunan nesnelerden dolayı sinyal kırınım, yansıma ve çok yollu yayılım sorunlarıyla karşılaştığı için, iki yönlü ölçme yönteminde, sinyalin vericiden çıkıp alıcıya ulaşma ve tekrar vericiye dönme süre ölçümlerinin hatalı olması, iki yönlü ölçme yönteminin dezavantajıdır.

Bir diğer ölçüm yöntemi varış açısı yöntemidir. Varış açısı yönteminde sensör vericisi tarafından gönderilen sinyalin, alıcıya varma açısı kullanılmaktadır. Sensör vericisi tarafından gönderilen sinyal, alıcıda bulunan anten yardımıyla algılanmaktadır. Hem vericinin alıcıya gönderdiği sinyalin fazı, hem de alıcının aldığı sinyalin fazı, verici ve alıcı tarafından belirlenmektedir. Böylece, alıcının aldığı sinyalin fazı ile verici tarafından gönderilen sinyalin fazı belli olduğu için, bu sinyaller arasındaki faz farkı belirlenebilmektedir. Bu faz farkı bilgisi, sinyalin dalga boyu ve sensör vericisi ile alıcısı arasındaki uzaklık verileri, varış açısı hesabı için kullanılan matematiksel denklemde kullanılarak, sinyalin hedef nesneye varış açısı bulunabilmektedir (Botler vd., 2020). Varış açısı ölçümleri üçgenleme algoritmasında kullanılarak, hedef nesnenin konumu belirlenmektedir. Varış açısı ölçüm yönteminin bazı dezavantajları bulunmaktadır. Bu yöntemin kullanılması için ek antenlere ihtiyaç duyulmakta ve bu nedenle sistem maliyeti artmaktadır. Ayrıca, kapalı alanda bulunan nesnelerden dolayı sensör vericisi tarafından gönderilen sinyal kırınım, yansıma ve çok yollu yayılım problemleri sebebiyle, alıcıya farklı açıda ulaşmakta ve bu nedenle varış açısı hatalı ölçülmektedir. Elde edilen açı ölçümleri üçgenleme algoritmasında kullanılarak konum belirlendiği için, varış açısı ölçüm yöntemi ve üçgenleme algoritması kullanılarak elde edilen konum belirleme problemlerinde hata yüksek olmaktadır.

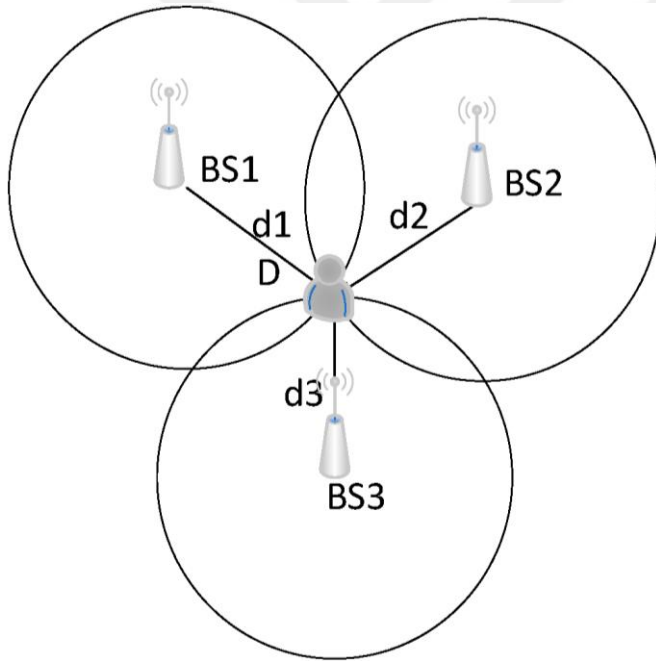
Bir diğerk ölçüm yöntemi varış zamanı farkı yöntemidir. Varış zamanı farkı yönteminde, hedef nesne üzerinde bulunan sensör vericisi tarafından, her bir sensör alıcısına gönderilen sinyalin, her bir alıcıya varış süresi arasındaki fark kullanılmaktadır. Elde edilen varış zamanı farkından yola çıkılarak, nesnenin her bir okuyucuya olan uzaklığı arasındaki fark hesaplanmaktadır. Okuyucuların yerleştirildikleri konumlar belli olduğu için, her bir okuyucu çevresinde, varış zamanı farkından yola çıkılarak elde edilmiş olan, nesnenin her bir okuyucuya olan uzaklık farkları göz önüne alınarak hiperboller çizilmektedir. Bu hiperbollerin kesişim noktası, hedef nesnenin konumu olmaktadır. Bu yöntemin dezavantajları, sensör vericisi ve alıcısındaki saatlerin birbirleriyle senkronize olması gerekliliğidir. Ayrıca, sensör sinyalinin kapalı alanda bulunan nesnelerden dolayı kırımım, çok yollu yayılım ve yansıma gibi problemlerle karşılaşmasından dolayı, varış zamanı ölçümlerinde ve dolayısıyla varış zamanı farkı ölçümlerinde yüksek ölçüm hataları meydana gelmektedir. Çünkü sinyalin yansıma ve çok yollu yayılım sorunlarıyla karşılaşmasından dolayı varış zamanı değışmekte ve hatalı ölçülmektedir.

Bir başka ölçüm yöntemi alınan sinyal şiddeti yöntemidir. Alınan sinyal şiddeti yönteminde, üzerinde sensör etiketi olan hedef nesne, sensör okuyucusunun kapsama alanına girdiğinde aktif hale gelerek okuyucuya sinyal göndermektedir. Okuyucu sinyali aldığıında, alınan sinyal şiddeti değeri ölçülmekte ve bu ölçüm yol kaybı modeli adı verilen matematiksel modelde kullanılarak hedef nesne ile sensör okuyucusu arasındaki uzaklık elde edilmektedir. Elde edilen uzaklık ise, trilaterasyon veya multilaterasyon algoritmalarında kullanılarak, hedef nesnenin konumu tespit edilmektedir. Ayrıca, RSS tabanlı konumlandırma için kullanılan bir başka yöntem ise parmak izi algoritması kullanılmaktadır. Elde edilen RSS ölçümleri, parmak izi algoritmasında kullanılarak ta konum tespiti yapılabilmektedir. Ancak, RSS ölçümleriyle konum tespiti yapmanın bazı dezavantajları bulunmaktadır. Kapalı alanda bulunan nesnelerden dolayı, sinyal kırımım, yansıma ve çok yollu yayılım sorunlarıyla karşılaştığında, sinyal ciddi anlamda zayıfladığı için alınan sinyal şiddeti ölçümleri de yüksek hata içermektedir. Ayrıca, RSS değerklerindeki kararsızlık ta ciddi ölçüm hatalarına sebep olduğundan, elde edilen uzaklık ölçümleri de hatalı olmaktadır. Bu uzaklık ölçümleri multilaterasyon veya trilaterasyon algoritmasında kullanıldığı için, RSS ölçümleri hatalı olduğunda, konum belirleme hatası da yüksek olmaktadır.

Kapalı alan konumlandırma problemlerinde, sensör ölçümleri konumlandırma algoritmalarında kullanılarak konum belirlenmektedir. Kapalı alanda konumlandırma yapmak için multilaterasyon tabanlı algoritmalar, parmak izi algoritması, yakınlık

algoritması ve Kalman tabanlı algoritmalar kullanılabilmektedir. Multilaterasyon tabanlı konumlandırma algoritmaları olarak ise trilaterasyon, multilaterasyon ve üçgenleme algoritmaları kullanılmaktadır.

Trilaterasyon algoritmasında çember geometrisi kullanılmakta ve her bir sensör okuyucusu için çember çizilmektedir. 2 boyutlu koordinat sistemi üzerinde konum belirlemek için, üç tane sensöre ihtiyaç duyulmaktadır. Konumu tespit edilmek istenen nesne ile her bir sensör okuyucusu arasındaki uzaklık varış zamanı yöntemi, alınan sinyal şiddeti yöntemi veya iki yönlü ölçme yöntemi kullanılarak ölçülmektedir. Nesne ile her bir okuyucu arasındaki uzaklık, ilgili okuyucu için çizilecek çemberin yarıçapı ve sensör okuyucusunun bulunduğu konum koordinatı ise o okuyucu için çizilecek çemberin merkezi olacak şekilde her bir okuyucu için çemberler çizilmektedir. Bu çemberlerin kesişim noktası, hedef nesnenin konumunu vermektedir.

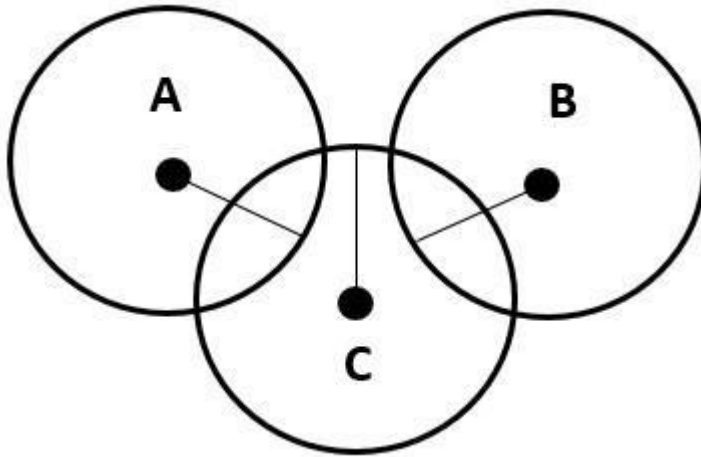


Şekil 2.1. Trilaterasyon algoritmasının gösterimi (Goldoni vd., 2010)

Şekil 2.1’de, trilaterasyon algoritmasının geometrik gösterimi verilmiştir. 3 adet sensör okuyucuları önceden konumları bilinen referans noktalara yerleştirilmekte ve her bir sensör okuyucusunun bulunduğu konum koordinatları, o sensör okuyucuları için çizilen sanal çemberlerin merkezi olmaktadır. Buna göre Şekil 2.1’de, 1. sensör okuyucusunun bulunduğu konum koordinatı BS1 ile gösterilmiş olup, 1. sensör okuyucusu için çizilen sanal çemberin merkezidir. Şekil 2.1’de gösterilen d_1 uzaklığı ise, 1. sensör okuyucusu ile D noktasında bulunan konumu tespit edilmek istenen nesne arasındaki uzaklık olup, 1. sensör okuyucusu için çizilen sanal çemberin yarıçapıdır. d_2 uzaklığı, 2. sensör okuyucusu ile D noktasında

bulunan konumu tespit edilmek istenen nesne arasındaki uzaklık olup, 2. sensör okuyucusu için çizilen sanal çemberin yarıçapıdır. 2. sensör okuyucusunun bulunduğu konum koordinatı ise, 2. sensör okuyucusu için çizilen sanal çemberin merkezi olup, Şekil 2.1’de BS2 ile gösterilmiştir. Son olarak d_3 uzaklığı ise, 3. sensör okuyucusu ile D noktasında bulunan konumu tespit edilmek istenen nesne arasındaki uzaklık olup, 3. sensör okuyucusu için çizilen sanal çemberin yarıçapıdır. 3. sensör okuyucusunun bulunduğu konum koordinatı ise, 3. sensör okuyucusu için çizilen sanal çemberin merkezidir ve Şekil 2.1’de BS3 ile gösterilmiştir. Şekil 2.1’de, her bir sensör okuyucusu için çizilen sanal çemberlerin kesişim noktası D noktası olup, konumu tespit edilmek istenen hedef nesnenin bulunduğu konumu ifade etmektedir.

Trilaterasyon algoritmasının bazı dezavantajları bulunmaktadır. Her bir sensör okuyucusu için çizilen çemberler çoğu zaman hiçbir noktada kesişmemekte veya bazı durumlarda ise bu çemberlerin kesişimi sonucu tek bir kesişim noktası ortaya çıkmayıp, bir kesişim bölgesi oluşabilmekte veya çemberlerin birden fazla kesişim noktası oluşabilmektedir. Bu nedenle, yüksek konum belirleme hataları meydana gelmektedir (Asmaa vd., 2014).



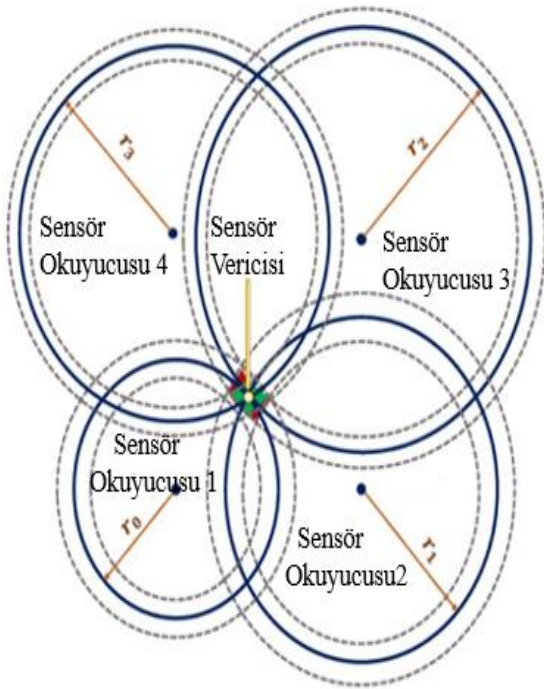
Şekil 2.2. Trilaterasyon algoritmasında kullanılan çember geometrisinde, çemberlerin tek bir noktada kesişmeme problemini gösterimi (Rukaiya vd., 2015)

Şekil 2.2 ‘de trilaterasyon algoritmasında kullanılan çember geometrisinde, çemberlerin tek bir noktada kesişmeme problemi gösterilmiştir. Çemberlerin tek bir noktada kesişmemesinin sebebi, sensörlerin ölçüm hatalarıdır. Sensör ölçümü sonucu elde edilen hedef nesne ile sensör okuyucusu arasındaki uzaklık, trilaterasyon algoritmasında kullanılan çemberlerin yarıçapını ifade etmektedir. Sensörün ölçüm hatasından dolayı bu uzaklık verisi hatalı elde edildiği için, trilaterasyon algoritmasında kullanılan çemberlerin yarıçapları da hatalı olmaktadır. Yarıçap hatalı olduğu için de, algorithmada kullanılan çemberler tek bir noktada

kesişmemektedir. Trilaterasyon algoritmasında nesne konumu, çemberlerin kesişim noktasının koordinatı olduğu için ve sensör ölçüm hatalarından dolayı çemberlerin tek bir noktada kesişmeme sorunları olduğu için, bu algoritma ile konum belirleme hatası yüksek olmaktadır. Bu durum trilaterasyon algoritmasının dezavantajıdır. Trilaterasyon algoritması kullanılarak daha yüksek doğrulukla konum belirleyebilmek için, sensörlerin ölçüm hatalarının azaltılması gerekmektedir. Bu noktada, Kalman filtre tabanlı algoritmalar kullanılabilmekte ve trilaterasyon algoritmasına göre çok daha yüksek doğruluklarla konum belirlenebilmektedir (Asmaa vd., 2014). Goldoni ve arkadaşları, kapalı alanda konum belirlemek için RSS ölçümlerini ve trilaterasyon algoritmasını kullanmışlardır. RSS ölçümleri sonucu elde edilen RSS değerlerini yol kaybı modelinde kullanarak hedef nesne ile her bir sensör okuyucusu arasındaki uzaklık verisini elde ederek trilaterasyon algoritmasında kullanmışlardır. 3 farklı hedefin konumunu tespit etmişlerdir. Hedeflerden iki tanesi sensör okuyucularına daha yakın konumda bulunurken, diğer hedef okuyuculardan uzak olacak şekilde yerleştirilmiştir. Trilaterasyon algoritması kullanılarak konumlandırılan ve okuyuculara yakın olan iki hedef için ortalama hatalar sırasıyla 1,44 metre ve 3,83 metre olarak elde edilmiştir. Okuyuculara en uzak olan hedef için trilaterasyon algoritmasının konumlandırma ortalama hatası ise, 9,12 metre olarak elde edilmiştir. Hedeflerden en yüksek konumlandırma hatasına sahip olan ve 9,12 metre ortalama hatayla konumu tespit edilen hedef, çalışma ortamında sensörlerle arasındaki görüş hattının olmayacağı ve çok yollu yayılıma uğrayacağı şekilde yerleştirilmiştir. Böylece, çok yollu yayılım ve görüş hattı olmama durumlarında trilaterasyon algoritmasının performansı da test edilmiştir. Bu durumda konumlandırma hatasının oldukça yüksek olduğu ve 9 metrelere kadar çıktığı görülmektedir. Ancak, görüş hattı olmama sorununu fazla yaşamayan diğer iki hedef için konumlandırma hatası görece daha düşük çıkmıştır. Açıkçası, görüş hattı olmadığı durumda bu kadar yüksek konumlandırma hatasının çıkması mantıklı ve anlaşılabilir. Çünkü bu çalışmada, trilaterasyon algoritmasında kullanılan çember yarıçapını elde etmek yani hedef nesne ile sensör okuyucuları arasındaki uzaklığı tespit etmek için RSS ölçümleri kullanılmıştır. Ancak, RSS ölçümleri kapalı ortamlarda bulunan nesnelerden dolayı sinyalin uğradığı çok yollu yayılım ve kırınım gibi sorunlara karşı zayıf ve hassas olduğundan yüksek ölçüm hatası içermektedir. Bu nedenle, trilaterasyon algoritmasında kullanılan çember geometrisinin yarıçap ölçümü hatalı olduğu için, bu hedef için konumlandırma hatası yüksek çıkmış ve 9 metrelere kadar ulaşmıştır. Bu nedenle, Goldoni ve arkadaşlarının yapmış oldukları çalışma aslında Kalman filtre tabanlı algoritmaların kapalı alan konumlama içi özellikle kullanılmasının gerekliliğini göstermektedir (Goldoni vd., 2010). Çünkü Kalman

filtre tabanlı algoritmalar ile sensör ölçüm hataları önemli ölçüde azaltılmakta ve yüksek konumlandırma doğruluklarına ulaşılabilmektedir. Ayrıca, Kalman filtre algoritması sadece sensör ölçüm gürültülerini azaltmakla kalmayıp, en optimal konum kestirimleri elde etmektedir. Bunun ayrıntıları ileriki kısımda anlatılacağı için burada kısaca bahsedilmiştir.

Bir diğer konumlandırma algoritması multilaterasyon algoritmasıdır. Multilaterasyon algoritmasının da çalışma mantığı, trilaterasyon algoritmasıyla aynıdır. Ancak tek farkları, trilaterasyon algoritmasında 3 tane sensör kullanılmakta ve bu nedenle 3 adet çember geometrisi kullanılmaktadır. Multilaterasyon algoritmasında ise üçten fazla sensör kullanılmakta ve bu nedenle üçten fazla çember geometrisi kullanılmaktadır. Konum belirlemek için kullanılan sensörler, önceden kapalı alanda referans noktalara yerleştirilmektedir. Multilaterasyon algoritmasında, sensörlerin bulundukları konum çember merkezi ve hedef nesne ile sensör arasındaki uzaklık ise yarıçap kabul edilerek, her bir sensör için çemberler çizilmekte ve bu çemberlerin kesişim bölgesi elde edilmektedir. Konumu tespit edilmek istenen hedef nesnenin bulunduğu konum koordinatı, elde edilen kesişim bölgesi içerisinde bulunmaktadır. Hedef nesnenin konumunun tespit edilmesi için, çemberlerin kesişimi sonucu oluşan bölge küçültülerek, hedef nesnenin bulunduğu konuma doğru yaklaşılmaktadır. Bu kesişim bölgesi küçültülmeye devam edilerek, hedef nesnenin konumu elde edilmektedir.



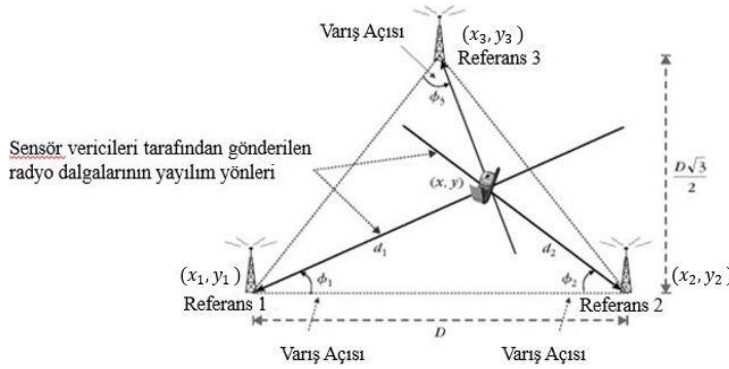
Şekil 2.3. Multilaterasyon algoritmasının gösterimi (Sang vd., 2019)

Şekil 2.3'te, multilaterasyon algoritmasında kullanılan çember geometrisi gösterilmiştir. 4 adet sensör okuyucuları, konumları bilinen referans noktalara yerleştirilmektedir. Sensör vericisi ise konumu tespit edilmek istenen hedef nesne üzerinde bulunmaktadır. Her bir sensör okuyucusu için çizilecek sanal çemberlerin yarıçapları, her bir sensör okuyucusu ile konumu tespit edilmek istenen hedef nesne arasındaki uzaklığa eşittir. Buna göre, Şekil 2.3 incelendiğinde, 1. sensör okuyucusu ile sensör vericisi arasındaki uzaklık r_0 kadar olup, 1. sensör okuyucusu için çizilen sanal çemberin yarıçapını ifade etmektedir. 2. sensör okuyucusu ile sensör vericisi arasındaki uzaklık r_1 kadar olup, 2. sensör okuyucusu için çizilen sanal çemberin yarıçapını temsil etmektedir. Benzer şekilde, 3. sensör okuyucusu ile sensör vericisi arasındaki uzaklık r_2 kadar olup, 3. sensör okuyucusu için çizilen sanal çemberin yarıçapıdır. 4. sensör okuyucusu ile sensör vericisi arasındaki uzaklık ise r_3 kadar olup, bu r_3 uzaklığı ise 4. sensör okuyucusu için sanal çemberin yarıçapını ifade etmektedir. Her bir sensör okuyucusu, konumlandırma yapılmak istenen kapalı alanda, konumları bilinen referans noktalara yerleştirilmektedir. Bu nedenle, her bir sensör okuyucularının bulundukları konum koordinatları bellidir ve bu konum koordinatları her bir sensör okuyucusu için çizilen sanal çemberlerin merkezini ifade etmektedir. Şekil 2.3'te, her bir sensör okuyucusu için yarıçap ve çember merkezleri kullanılarak sanal çemberler çizilmiştir. Üzerinde sensör vericisi bulunan hedef nesnenin konumu ise bu çemberlerin kesişim bölgesi içerisinde bulunmaktadır. Multilaterasyon algoritması da trilaterasyon algoritmasında olan dezavantajlarla karşılaşmaktadır. Multilaterasyon algoritmasında kullanılan çember yarıçapları, konumu tespit edilmek istenen nesne ile sensör okuyucusu arasındaki uzaklık ölçümüdür. Uzaklık ölçümleri, kapalı alanda bulunan nesnelerden dolayı sensör sinyallerinin kırınım, yansıma gibi sorunlarla karşılaşmalarından dolayı hatalı olmaktadır. Bu durumda uzaklık ölçümleri hatalı olduğu için, algoritmada kullanılan çemberlerin yarıçap değerleri hatalı olmaktadır. Bu nedenle, çemberler tek bir noktada kesişmemekte veya bazı durumlarda ölçüm hatasından dolayı çemberlerin hiçbir kesişim bölgesi oluşmayabilmektedir. Bu durumda, multilaterasyon algoritması kullanılarak elde edilen konum tespiti, yüksek hatalara sahip olmaktadır. Asmaa ve arkadaşları konum belirleme probleminde, trilaterasyon ve multilaterasyon algoritmalarının, kullanılan sensör sayılarına bağlı olarak performanslarının nasıl değiştiğini karşılaştırmışlardır. Sensör sayısının artırılarak multilaterasyon algoritmasında kullanılmasıyla, konum belirleme ortalama hatanın azaldığını gözlemlemişlerdir (Asmaa vd., 2014). Başka bir çalışmada Wessels ve arkadaşları, multilaterasyon algoritması kullanarak konum belirleme problemi üzerine çalışmışlardır. Multilaterasyon algoritmasında kullanılan çemberlerin yarıçaplarını temsil

eden hedef nesne ve sensör okuyucu arasındaki uzaklıkları belirlemek için, RSS ölçümlerini kullanarak uzaklık hesaplamışlardır. Ancak bu çalışmada, RSS ölçümlerini direkt kullanmadan önce, ölçüm hatasını azaltmak için ölçümleri filtrelemiş ve ardından algoritmada kullanmışlardır. Böylece 0,8 metre hatayla konum tespiti yapmışlardır (Wessels vd., 2010). Elde edilen sonuçlar Goldoni ve arkadaşlarının yaptıkları çalışmayla karşılaştırılacak olursa, Goldoni ve arkadaşları RSS ölçümlerini filtrelemeden multilaterasyon algoritmasında kullanmış ve hedefin okuyuculara en yakın olduğu durumda bile, konum belirleme hatası 1,44 metre olarak elde edilmiştir. Ancak RSS ölçümleri filtrelenerek ölçüm hatalarının azaltıldığı durumda, konumlandırma hatasının 0,8 metreye kadar azaldığı görülmektedir. Bu beklenen bir sonuçtur. Çünkü RSS ölçümleri yol kaybı modelinde kullanılarak sensör ile hedef nesne arasındaki uzaklık bilgisi elde edilmekte ve bu uzaklık bilgisi multilaterasyon algoritmasında kullanılan çemberlerin yarıçaplarını ifade etmektedir. RSS değerleri filtrelenerek ölçüm hataları azaltıldığında, daha doğru RSS verileri elde edildiği için, yol kaybı modeli ile hesaplanan sensör ve nesne arasındaki uzaklık değeri de daha doğru olmaktadır. Bu uzaklık değeri multilaterasyon algoritmasında kullanılan çemberlerin yarıçapını ifade ettiği için, çemberlerin kesişimi sonucu elde edilen bölge de daha doğru bir şekilde elde edilmektedir. Çünkü RSS ölçümlerinin filtrelenerek ölçüm hatalarının azaltılmasıyla, sensör nesne arasındaki uzaklık verisinin doğruluğu arttığı için, algoritmada kullanılan çember yarıçapları da daha doğru şekilde elde edilmiştir. Bu durumda, kullanılan çember geometrileri daha yüksek doğrulukla elde edilmektedir. Bu nedenle, nesnenin konumunun da içerisinde bulunduğu çember kesişim bölgesi daha yüksek doğrulukla elde edildiği için, nesnenin konumu da daha doğru bir şekilde bulunmaktadır.

Üçgenleme algoritması kapalı alanda konum bulmak için kullanılan bir başka algoritmadır. Üçgenleme algoritması kullanılarak gerçekleştirilen konumlandırma sistemlerinde, varış açısı yöntemi kullanılmaktadır. Varış açısı bilgisi, yöne duyarlı antenler kullanılarak elde edilmektedir. Varış açısı yönteminde, sensör vericisi tarafından gönderilen sinyal, alıcıda bulunan anten yardımıyla algılanmaktadır. Verici ve alıcı tarafından, gönderilen ve alınan sinyallerin fazları belirlenmekte ve böylece faz farkı hesaplanabilmektedir. Elde edilen faz farkı bilgisi, sinyalin dalga boyu ve sensör vericisi ile alıcısı arasındaki uzaklık verileri varış açısı hesabı için kullanılan matematiksel denklemde kullanılarak, vericiden gönderilen sinyalin alıcıya varış açısı bulunabilmektedir (Botler vd., 2020). Varış açısı ölçümü, sensör vericisi tarafından iletilen ve sensörün alıcısı tarafındaki anten ile algılanan radyo dalgasının yayılım yönünü belirlemektedir. Sensör vericisi, konumu tespit edilmek istenen nesne

üzerinde bulunmakta ve sensör alıcıları ise konumu bilinen referans noktalara yerleştirilmektedir. Bu yöntemde nesnenin konumu, iletilen radyo dalgalarının yönleri kullanılarak belirlenmektedir. Hedef nesne üzerinde bulunan sensör vericisi, referans noktalara yerleştirilen sensör alıcılarının her birisine radyo dalgası iletmektedir. Radyo dalgaları, sensör alıcısında bulunan anten vasıtasıyla alınmaktadır. Ancak, sensör alıcıları farklı konumlarda bulunduklarından dolayı, gönderilen her bir radyo dalgası alıcılara farklı açılarla gelmektedir. Her bir sensör alıcılarına gönderilen radyo dalgalarının, alıcı antenine varış açıları farklı olduğu için dalga yayılım yönleri de farklı olmakta ve yayılan dalgalar birbirleriyle kesişmekte ve bu kesişim noktasının koordinatı, kullanıcının konumunu ifade etmektedir.



Şekil 2.4. Üçgenleme algoritmasının gösterimi (Munoz vd., 2009)

Üçgenleme algoritması, kapalı alanda konum belirlemek için iyi bir seçim değildir. Bunun nedeni, kapalı alanda bulunan nesnelerden dolayı sensör tarafından, hedef nesneye gönderilen sinyal yansıma ve çok yollu yayılıma uğradığında, sinyalin varış açısı değiştiği için, üçgenleme algoritmasıyla konum belirleme hatası çok yüksek olmaktadır. Bu nedenle, üçgenleme algoritması tek başına kullanıldığında yüksek konumlandırma hatalarına sahip olduğundan, daha yüksek doğruluklarla konum belirleyebilmek için, sensör ölçüm hatasını azaltacak farklı algoritmalarla beraber kullanılmalıdır. Kim ve arkadaşları, WiFi sensörü ve üçgenleme algoritması kullanarak, kapalı bir alanda konum belirleme problemiyle uğraşmışlardır. Üçgenleme algoritmasında, iki boyutta konumlandırma yapmak için 2 adet açı ve bir adet uzaklık bilgisine ihtiyaç duyulmaktadır. Gerçekleştirilen çalışmada, uzaklık verisini elde etmek için alınan sinyal şiddeti ölçümü yapılarak elde edilen RSS ölçümleri yol kaybı modeli kullanılarak uzaklık verisi elde edilmiştir. Açı verisi elde etmek için ise, hedef nesneden alıcıya gönderilen sinyalin varış açıları elde edilerek, en küçük kareler kestirici algoritmasında kullanılmıştır. En küçük kareler kestiricisi algoritmasıyla kestirilen açı

verileri için gradyen filtre kullanılarak, kapalı alanda bulunan nesnelerden dolayı sinyalin kırım ve yansımasının neden olduğu RSS ölçümündeki gürültülerin azaltılmasıyla, ölçüm hataları azaltılmış ve bu RSS ölçümleri veri tabanında saklanmıştır. Daha sonra yol kaybı modeli kullanılarak filtrelenmiş RSS ölçümleri, uzaklık verilerine dönüştürülmüş ve trilaterasyon algoritmasında kullanılarak ortalama 2,67 metre hatayla konum belirlenmiştir. RSS ölçümlerinin filtrelenmesi sonucunda ortalama hata 5,87 metreden 2,67 metreye düşmüştür. Aslında Subhan ve arkadaşlarının yaptıkları çalışma incelendiğinde, konum bulmak için kullanılan yöntem için parmak izi ve trilaterasyon yöntemi adını verme sebepleri trilaterasyon algoritmasında kullanılan nesne ve sensör arasındaki uzaklığı hesaplamak için RSS ölçümlerini kullanmış olmalarıdır. Aslında RSS ölçümünün elde edilmesi, bu çalışmada sadece sensör ölçüm yöntemidir, konumlandırma yapmak için kullanılan yöntem değildir. Bu nedenle, bu çalışmada kullanılan yöntem parmak izi yöntemi denilmesi aslında çok ta doğru değildir. Çünkü parmak izi yönteminde, sensör okuyucularından elde edilen RSS ölçümlerinin, hedef nesnede bulunan sensör ölçümüyle elde edilen RSS ölçümleriyle eşleştirilmesi ve bu eşleştirme sonucu parmak izinde yani veri setinde bulunan RSS değerlerinden, hangi değere daha yakın olduğu göz önüne alınarak ve kümeleme algoritmaları kullanılarak konum belirlenmektedir. Ancak Subhan ve arkadaşlarının kullandığı yöntemde RSS ölçümleri sadece, hedef nesne ve sensör okuyucuları arasındaki uzaklığı elde etmek için bir ölçüm metodu olarak kullanılmıştır. Subhan ve arkadaşlarının elde ettikleri sonuç, RSS ölçümlerinin direkt olarak trilaterasyon algoritmasında kullanılmadan önce filtrelenerek ölçüm hatasının azaltılması gerektiği düşüncemizi doğrulamaktadır. Çünkü kapalı alanda bulunan nesneler, sensör sinyalinin kırım, yansıma gibi sorunlarla karşılaşmasına yol açtığı için ölçüm hataları artmaktadır. Bu nedenle RSS ölçüm hataları da artmaktadır. Elde edilen RSS ölçümleri yol kaybı modeli kullanılarak uzaklık verisine dönüştürülüp trilaterasyon algoritmasında kullanıldığı için, bu algoritmaların kullanımıyla elde edilen konum belirleme hatası da doğal olarak yüksek çıkmaktadır. Çünkü RSS ölçümü hatalı olduğunda, nesneyle sensör okuyucusu arasındaki uzaklık ta hatalı olmaktadır. Bu uzaklık trilaterasyon algoritmasında kullanılan çemberin yarıçapını ifade ettiği için, algoritmada kullanılan çemberin yarıçapı yanlış olmakta ve çizilen çemberler de hatalı olmaktadır. Bunun sonucunda da çemberlerin kesişim bölgeleri hatalı olduğundan, konumlandırma hataları da yüksek olmaktadır. Bu nedenle, sensör ölçümleri filtrelenerek gürültüler ve ölçüm hataları azaltılmalı ki, multilaterasyon tabanlı algoritmaların kullanılması sonucu elde edilen konum belirleme hataları da azaltılmış olsun.

Bu nedenle Subhan ve arkadaşlarının elde ettikleri sonuç, bizim bu düşüncemizi destekler niteliktedir (Subhan vd., 2011).

Kapalı alanlarda konum belirlemek için kullanılan bir başka algoritma ise yakınlık (İng. *Proximity*) algoritmasıdır. Yakınlık algoritmasında alınan sinyal şiddeti ölçümleri kullanılmaktadır. Hedef nesnede bulunan sensör etiketi, sensör okuyucularına sinyal iletmektedir. Her bir sensör okuyucusu bu sinyali aldığı anda o okuyucular için alınan sinyal şiddeti değerleri elde edilmektedir. Sensör etiketinin gönderdiği sinyal, okuyucuda algılandığında elde edilen RSS değeri yüksekse, hedef nesnenin o okuyucuya uzak olduğu, RSS değeri düşükse de yakın olduğu anlaşılmaktadır. Bunun sebebi, uzaklık arttıkça alınan sinyal şiddeti değerinin de düşüyor olmasıdır. Sensör etiketi her bir okuyucuya sinyal göndermekte ve okuyucular tarafından alınan bu sinyallere ait alınan sinyal şiddeti değerlerinin birbirleriyle karşılaştırılmasıyla, hedef nesnenin hangi okuyucuya daha yakın olduğu anlaşılabilmektedir. Bu şekilde, bütün okuyuculardan, farklı ölçümler sonucu elde edilen, RSS değerleri toplanmakta ve ortalamaları alınarak bir ağırlık çarpanı elde edilmektedir. Hedef nesnenin konumunun tespit edilebilmesi için, her 3 okuyucu için elde edilen ağırlık çarpanı ile okuyucuların konum bileşenlerinin çarpımlarının ortalaması alınarak hedef nesnenin bulunduğu konum koordinatları elde edilmektedir (Montaser ve Moselhi, 2014). Yakınlık algoritmasında temel olarak hedef nesnenin, sensör okuyucularına olan yakınlığı ölçülmektedir. Bunun için RSS ölçümleri kullanıldığı gibi ayrıca sensör sinyalinin varış zamanı ölçümü de kullanılabilmektedir. Konumu tespit edilmek istenen nesne üzerinde bulunan sensör etiketi, sensör okuyucularına sinyal göndermekte ve bu sinyallerin okuyuculara varış zamanları ölçülmektedir. Bu ölçüme göre nesnenin hangi okuyucuya yakın olduğu bilgisi elde edilebilmektedir. Nesne tarafından gönderilen sinyalin, okuyucuya varış zamanı küçük ise nesnenin o okuyucuya yakın olduğu, varış zamanı büyükse nesnenin o okuyucuya uzak olduğu anlaşılabilmektedir. Üzerinde sensör etiketi bulunan nesne, her bir okuyucuya sinyal gönderdiği için bu sinyallerin her bir okuyucuya varış zamanı ölçümleri kullanılarak, nesnenin hangi okuyucuya daha yakın olduğu belirlenmekte ve sensör okuyucularının bulundukları konum koordinatları ile yakınlık bilgileri kullanılarak hedef nesnenin konumu tespit edilmektedir. Yakınlık algoritmasının dezavantajı, kapalı alanda sensör ölçümlerinin gürültülü olması sebebiyle yüksek ölçüm hataları meydana gelmektedir. Yakınlık algoritmasında RSS ve varış zamanı ölçümleri yapılmakta ve kapalı alanda bulunan nesnelerden dolayı sensör sinyalinin kırımım, yansıma ve çok yönlü yayılımına uğraması sebebiyle zayıflaması sonucu RSS ölçümleri yüksek hatalar

içermektedir. Ayrıca okuyucuya gönderilen sinyalin varış zamanı da sinyalin kırınım, çok yollu yayılım ve yansıma gibi sorunlardan dolayı hatalı ölçüldüğünden, yakınlık algoritmasında kullanılan, nesnenin okuyuculara ne kadar yakın olduğu bilgisi de hatalı olmaktadır. Dolayısıyla, yakınlık algoritmasıyla konum belirleme problemlerinde hata yüksek olmaktadır (Li ve Becerik-Gerber, 2011). Bu nedenle yakınlık algoritması kullanılarak konum belirlemek için öncelikle sensör ölçüm hatalarının azaltılması gerekmektedir. Bu amaçla Kalman filtre tabanlı algoritmalar gibi çeşitli yöntemler kullanılabilir. Ölçüm hataları azaltıldığında daha doğru RSS ve varış zamanı ölçümleri elde edileceği için, konumu tespit edilmek istenen nesnenin hangi okuyucuya daha yakın olduğu bilgisi daha yüksek doğrulukla öğrenilebilmektedir. Bu yakınlık verileriyle, okuyucu konumlarının birlikte yakınlık algoritmasında kullanılması sonucu konumlandırma hatası da azalmış olacaktır (Li ve Becerik-Gerber, 2011).

Kapalı alanda konum belirlemek için kullanılan bir diğer algoritma ise Kalman filtre tabanlı algoritmalar. Doğrusal Kalman filtresi (İng. *Linear Kalman Filter*), genişletilmiş Kalman filtresi ve kokusuz Kalman filtresi en sık kullanılan Kalman filtre tabanlı algoritmalar. Kalman filtre algoritmaları, konumlandırma ve hedef takip performansı yüksek olduğu için, uzun yıllardır kullanılan bir algoritmadır (Zhang vd., 2012). Kalman filtre tabanlı algoritmalar, hedef ve ölçüm modeli için, süreç ve ölçüm gürültüsünün beyaz Gauss gürültüsü olduğu durum için en uygun kestirimi üretmektedir. Bu nedenle, multilaterasyon, trilaterasyon, parmak izi algoritması ve yakınlık algoritması gibi diğer konumlandırma algoritmalarına göre daha yüksek doğruluklarla konum tahmini yapılabilmektedir. Bundan dolayı, konum belirleme problemleri için, Kalman filtre algoritması, diğer konumlandırma algoritmalarından daha gelişmiş algoritmalar.

Doğrusal Kalman filtresi, hedef ve sensör ölçüm modelinin doğrusal olduğu durumlarda kullanılmaktadır. Kalman filtresi, hedefin başlangıç durum, başlangıç kovaryansı ve sensör ölçümlerini kullanarak, o hedefe ait en optimum durum kestirimini elde etmektedir. Kalman filtre algoritmasında öngörüm ve güncelleme (İng. *Update*) aşamaları bulunmaktadır. Öngörüm aşamasında, algoritma kendi modeli ile hedef nesnenin durum öngörülerini ve bu durum öngörümündeki belirsizliği ifade eden durum öngörüm kovaryansını elde etmektedir. Ayrıca algoritmanın öngörüm aşamasında, sensörün yapacağı ölçüme dair ölçüm öngörümü elde edilmektedir. Sensör ölçümü ile algoritmanın yapmış olduğu bu ölçüme ait ölçüm öngörümü arasındaki fark inovasyonu vermektedir. Bu nedenle inovasyon aslında, Kalman filtre algoritmasının yapmış olduğu ölçüm öngörümünün, sensörün

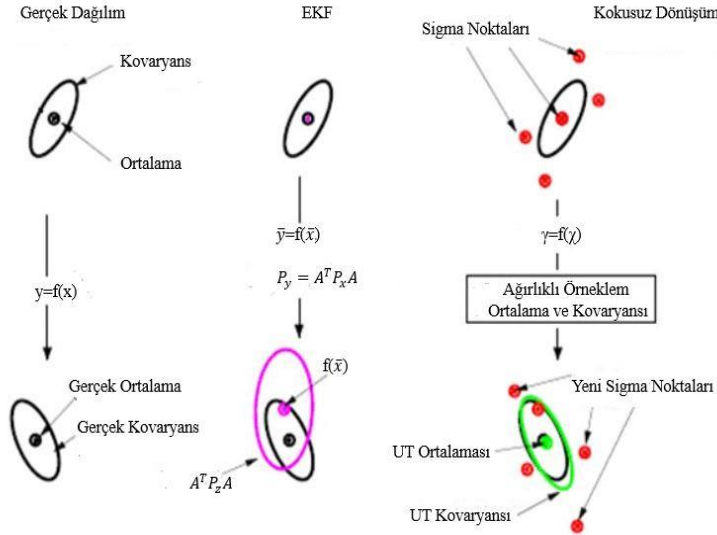
gerçekte yaptığı ölçümünden ne kadar farklı olduğunu belirtmektedir. Ölçüm öngörümü, algoritmanın yapmış olduğu bir öngörüm olduğu için bu öngörüm bir belirsizlik içermekte ve bu belirsizlik sebebiyle elde edilen inovasyon da bir belirsizlik içermektedir. Bu belirsizliğe inovasyon kovaryansı adı verilmektedir. Doğrusal Kalman filtre algoritmasında durum öngörüm kovaryansı ile inovasyon kovaryansı kullanılarak Kalman kazancı hesaplanmakta ve bu kazanç, Kalman filtre algoritmasının durum kestirimini üretmesi için kullanılmaktadır. Kalman kazancı, durum öngörüm kovaryansı ile inovasyon kovaryansının birbirlerine oranı ile doğru orantılıdır. Bu nedenle, durum öngörüm kovaryansı arttığında Kalman kazancı artmakta, inovasyon kovaryansı arttığında ise Kalman kazancı azalmaktadır. Kalman filtre algoritmasının kestirim üretmek için elinde, kendi modeli ile öngörmüş olduğu durum öngörümü ve ayrıca o sistem durumlarına ait sensör ölçümleri bulunmaktadır. Algoritma ise, kestirim üretmek için elinde bulunan bu durum öngörümünü ve sensör ölçümünden gelen bilgiyi kullanmaktadır. Ancak, Kalman filtre algoritması durum öngörümüne mi, yoksa sensör ölçümünden gelen bilgiye mi daha çok güvenerek kestirim üreteceğine ise Kalman kazancını kullanarak karar vermektedir. Kalman kazancı büyük olduğunda sensör ölçümünden gelen bilgiye daha çok güvenip, daha çok ağırlık vererek kestirim üretirken, Kalman kazancı küçük olduğunda, kendi durum öngörümüne daha çok güvenerek kestirim üretmektedir. Kalman kazancının büyük veya küçük olması ise durum öngörüm kovaryansı ile inovasyon kovaryansı arasındaki ilişkiye bağlıdır. İnovasyon kovaryansı arttığında Kalman kazancı küçülmekte ve bu durumda algoritma kestirim üretmek için kendi modeline yani durum öngörümüne daha çok güvenip ağırlık vermektedir. Durum öngörüm kovaryansı arttığında ise kazanç artmakta ve bu durumda algoritma kendi durum öngörümüne güvenmeyip, sensör ölçümünden gelen bilgiye daha çok güvenerek kestirim üretmektedir. Bu nedenle, Kalman filtre algoritmaları ile en uygun, en optimal durum kestirimleri elde edilmektedir. Bu nedenle, Kalman filtre tabanlı algoritmalar kullanılarak, multilaterasyon, trilaterasyon ve parmak izi algoritması gibi diğer algoritmalarda, sensörlerin ölçüm hatalarından kaynaklanan yüksek konumlandırma hataları olmamakta ve daha yüksek doğruluklarla konum tahmini yapılabilmektedir. Bu nedenle, kapalı alanda konum bulmak için, Kalman filtre tabanlı algoritmalar multilaterasyon, parmak izi algoritmaları gibi diğer algoritmalara kıyasla çok daha başarılıdır. Ancak, Kalman filtre algoritmasının, en uygun kestirimi üretebilmesi için Kalman kazancı önemli rol oynamaktadır. Çünkü algoritmanın sensör ölçümüne mi yoksa durum öngörümüne mi ağırlık vererek kestirim üreteceğini ve dolayısıyla algoritmanın kestirim performansını Kalman kazancı etkilemektedir. Bu nedenle, Kalman filtre algoritmalarında, kazancın

değerini etkileyen, filtrenin süreç gürültü varyansının seçimi oldukça önemli olduğundan, filtrenin süreç gürültü varyansı algoritmanın tasarım parametresidir (Huang vd., 2015).

Doğrusal Kalman filtre algoritması durum denklemi ve sensörün ölçüm modelinin doğrusal olduğu durumlarda kullanılmaktadır. Ancak, durum denklemi veya sensörün ölçüm modeli veya her ikisi de doğrusal olmadığı zaman, genişletilmiş Kalman filtresi veya kokusuz Kalman filtresi kullanılmaktadır.

Genişletilmiş Kalman filtresi için de doğrusal Kalman filtresinde olduğu gibi başlangıç durum ve başlangıç kovaryans değerlerinin biliniyor olması ve ölçüm ve süreç gürültülerinin beyaz Gauss gürültüsü olup aralarında bağımsız olması gerekmektedir. Ayrıca hedef nesnenin durum değişkenleri ile bu durum değişkenleri arasındaki ilişkiyi ifade eden durum denklemi bir Markov sürecidir. Tüm bunlar genişletilmiş Kalman filtre algoritmasının temel varsayımları olup, algoritma bu varsayımları kullanarak kestirim üretmektedir (Huang ve Wang, 2006). Genişletilmiş Kalman filtre algoritması, durum denkleminin veya sensör ölçüm modelinin doğrusal olmadığı durumlar için kullanılmaktadır. GKF algoritmasında, doğrusal olmayan bu fonksiyonlar Taylor serisi açılımı yapılarak doğrusallaştırılmaktadır. Taylor serisi açılımı ile Jakobyen matris hesaplanmaktadır. Taylor serisi açılımı sonsuz terime sahiptir. Ancak, doğrusal olmayan fonksiyon, Taylor serisi açılımı yardımıyla doğrusallaştırılırken, sonsuz terime sahip bu açılım, sadece ilk iki terimle ifade edilerek Jakobyen matris hesaplanmaktadır. Bu durumda, sonsuz terime sahip olan bir açılımı, sadece ilk iki terimiyle ifade etmekten kaynaklanan doğrusallaştırma hataları meydana gelmektedir. Bu durum GKF algoritmasının performansını düşürmekte ve doğrusalsızlığın (İng. *Nonlinearity*) yüksek olduğu durumlarda ise filtrenin ıraksamasına neden olmaktadır. Bu nedenle, bu durum GKF algoritmasının önemli bir dezavantajıdır. GKF algoritmasında meydana gelen doğrusallaştırmadan kaynaklı bu sorunların çözümü olarak kokusuz Kalman filtre algoritması kullanılmaktadır (Huang ve Wang, 2006). Kokusuz Kalman filtre algoritmasında, hedef nesneye ait başlangıç durumları ve bu durumlara ait başlangıç kovaryans kullanılarak, hedef nesneye ait bir sonraki an için durum kestirimleri elde edilmektedir. Konum belirleme problemlerinde başlangıç durumları, nesnenin başlangıçta bulunduğu konum ve başlangıç hız bileşenlerinden oluşmaktadır. Konumu tespit edilmek istenen nesneye ait başlangıç durumları rastgele bir değişken olup, bu rastgele değişken ortalama ve kovaryans ile ifade edilmektedir. Rastgele değişkene ait bu başlangıç durum ve kovaryans değerleri, kokusuz Kalman filtre algoritmasında başlangıç değerler olarak kullanılıp, bu algoritmadaki formüllere göre zamanda ilerletilerek, rastgele değişkene ait bir

sonraki an için ortalama ve kovaryans değerleri kestirilmektedir. Böylece başlangıç değerleri kullanılarak, kokusuz Kalman filtre algoritmasının bir sonraki an için durum kestirimleri elde edilmektedir. Kokusuz Kalman filtre algoritmasında doğrusallaştırma için kokusuz dönüşüm yöntemi (İng. *Unscented Transform*) kullanılmaktadır (Huang ve Wang, 2006).



Şekil 2.6. İki boyutlu Gauss dağılımına sahip rasgele değişkenin doğrusal olmayan dönüşümü sonucu ortalama ve kovaryans değerlerinin gerçek değerleri ile GKF ve kokusuz dönüşüm sonucu elde edilen değerleri (Lima, 2017).

Şekil 2.6’da, iki boyutlu Gauss dağılımına sahip rasgele değişkenin doğrusal olmayan dönüşümü sonucu ortalama ve kovaryans değerlerinin gerçek değerleri ile GKF ve kokusuz dönüşüm sonucu elde edilen değerleri gösterilmektedir (Lima, 2017). Kokusuz dönüşümde, kestirmek istenen rastgele değişkene ait olasılık yoğunluk fonksiyonundan sigma noktaları seçilip, bu noktalar ortalama ve kovaryans değerleriyle ifade edilmektedir. Çünkü rastgele bir değişken ilk iki momentle yani ortalama ve varyans ile ifade edilmektedir. Seçilen sigma noktaları ile bir örnek uzay oluşturulmaktadır. Bu sigma noktalarının durum geçiş fonksiyonundan geçirilip zamanda ilerletilmesiyle, kokusuz Kalman filtre algoritmasındaki denklemler işletilmekte ve böylece rastgele değişkene ait ortalama ve kovaryans değerlerinin kestirimleri elde edilmektedir. Ortalama ve kovaryans değerlerinin kestirimleri elde edilen bu rastgele değişken, kokusuz Kalman filtre algoritmasıyla kestirmek istenen, nesnenin konum ve hız bileşenlerinden oluşan durum değişkenidir. Bu nedenle, bu durum değişkenini ifade eden rastgele değişken, kokusuz Kalman filtre algoritmasıyla kestirildiğinde aslında nesnenin konum ve hız bileşenlerinden oluşan durum değişkeni kestirilmiş ve böylece nesnenin konumu kestirilmiş olmaktadır. Rastgele bir değişken de ortalama ve kovaryans ile

ifade edilmektedir. Konumu belirlenmek istenen nesnenin konumu da istatistiksel anlamda bir rasgele değişken olduğu için, kokusuz Kalman filtre algoritması sonucu elde edilen rasgele değişkene ait ortalama değer aslında konumu tespit edilmek istenen nesnenin konum kestirimi, kovaryans değeri ise kestirim kovaryansı olmaktadır.

Genişletilmiş Kalman filtresi ile kokusuz Kalman filtresi arasındaki fark, rasgele değişkene ait ortalama ve kovaryans değerlerinin, algoritma içinde zamanda farklı şekilde ilerletiliyor olmalarından kaynaklanmaktadır. GKF algoritmasında doğrusal olmayan fonksiyonların doğrusallaştırılması için Taylor serisi açılımı yapılarak Jakobyan matris hesaplanmaktadır. Ancak, kokusuz Kalman filtre algoritmasında, nesneye ait başlangıç durum değişkenleri ve bu durum değişkenlerine ait başlangıç kovaryans değerleri, zamanda ilerletilirken, herhangi bir Taylor serisi açılımı ve Jakobyan matris hesabı yapılmamaktadır. Kokusuz Kalman filtre algoritmasında, başlangıç durum değişkenleri ve bu durum değişkenlerine ait başlangıç kovaryans değerleri, kokusuz dönüşüm yöntemi kullanılarak zamanda ilerletilerek kestirim elde edilmektedir. Bu nedenle, kokusuz Kalman filtre algoritmasında, GKF algoritmasının bir dezavantajı olan, doğrusallaştırmadan kaynaklı hatalar ve bu nedenle filtrenin performansında bir azalma olmamaktadır. Bu nedenle, Şekil 2.6 incelendiğinde, kokusuz Kalman filtre algoritmasında, kokusuz dönüşüm sonucunda elde edilen rasgele değişkene ait ortalama değer ve kovaryans dağılımının, o rasgele değişkene ait gerçek ortalama değer ve kovaryans dağılımıyla daha iyi örtüştüğü ve rasgele değişkene ait ortalama değer ve kovaryans dağılımının daha iyi korunduğu görülmektedir. Ancak Şekil 2.6 incelendiğinde, GKF algoritması kullanılarak kestirilen rasgele değişkene ait ortalama değer ve kovaryans dağılımının, o rasgele değişkene ait gerçek ortalama değer ve kovaryans dağılımından farklı olduğu ve birbirleriyle örtüşmediği görülmektedir. Bunun nedeni, GKF algoritmasında doğrusal olmayan durum ve ölçüm fonksiyonlarının, Taylor serisi açılımı yapılarak doğrusallaştırılırken, doğrusallaştırmadan kaynaklanan hatalar, GKF algoritmasının kestirim hatasını artırmaktadır. Özellikle GKF algoritmasında, doğrusalsızlığın yüksek olduğu durumlarda, doğrusal olmayan fonksiyonların Taylor serisi açılımı yapıp, aslında sonsuz terime sahip olan Taylor serisi açılımının, sadece ilk iki teriminin kullanılmasıyla elde edilen Jakobyan matris hesabından kaynaklanan, yani doğrusallaştırma probleminden kaynaklanan sorunlar, GKF filtresinin ıraksamasına dahi neden olmaktadır. Bu durum, GKF algoritmasının önemli bir dezavantajıdır. Zafari ve arkadaşları kapalı alan konumlandırma probleminin çözümü için *bluetooth* sensörü kullanarak RSS ölçümleri elde etmişler ve konumlandırma algoritması olarak ise yakınlık algoritması kullanmışlardır. Kapalı

alanlardaki nesnelerden dolayı, sensör sinyali zayıfladığı ve kırıma uğradığı için RSS değerleri zayıflamaktadır. Bu nedenle Zafari ve arkadaşları RSS değerlerini daha kararlı hale getirmek için Kalman filtre algoritması kullanmışlardır. Kalman filtre algoritması sonucu elde edilen RSS kestirimlerini yakınlık algoritmasında kullanarak konum belirlemişler ve konumlandırma hatasının %32 oranında azaldığını belirtmişlerdir. Ayrıca konum belirlemek için, Kalman filtre ve parçacık filtre algoritmalarını beraber kullandıkları ikinci bir kaskat algoritma geliştirmişlerdir. Bu kaskat algortmada, Kalman filtre algoritmasını RSS ölçümlerinin hatalarını azaltmak için kullanmışlardır. Kalman filtre algoritmasının ölçümle güncelleme aşamasında, *bluetooth* sensörünün kullanılmasıyla elde edilen RSS ölçümlerini kullanmışlardır. Ayrıca, Kalman filtre algoritmasının durum öngörümü aşamasında, RSS değerlerinin zamanla nasıl değiştiği ve RSS değerlerinin nasıl dalgalandığı arasında bir ilişki kuran dinamik model oluşturmuşlardır. Sistem durum değişken matrisinde ise Kalman filtrenin kestirimi sonucu elde edilmek istenen RSS değeri ile RSS değerlerinin ortamda nasıl dalgalandığını ifade eden durum değişkeni kullanmışlardır. Böylece, algoritmanın durum öngörümü aşamasında elde edilen RSS değerinin öngörümünü, sensörden gelen RSS ölçümleri ile güncelleyerek, RSS kestirimleri elde edilmiştir. Elde edilen RSS kestirimleri yol kaybı modelinde kullanılarak, nesne ile *bluetooth* sensörü okuyucular arasındaki uzaklık elde edilerek yakınlık algoritmasında kullanılarak konum tespiti yapılmıştır. Kullanılan bu yöntemle yakınlık algoritmasının konum belirleme doğruluğunun %32 oranında arttığı belirtilmiştir. Bunun sebebi, Kalman filtre algoritması kullanılarak, RSS ölçümlerinin hatalarının azaltılması ve RSS değerleri için en iyi, en optimum kestirimin elde edilmiş olmasıdır. Kalman filtre algoritması en iyi RSS değerinin kestirimi için öncelikle durum öngörüm aşamasında kendi modeli ile RSS değerinin öngörümünü yapmaktadır ve bu bir öngörüm olduğu için belirsizlik içerdiği için durum öngörüm kovaryansını da hesaplamaktadır. Daha sonra sensör ölçümü için de öngörüm yapmakta ve gerçek sensör ölçümü ile algoritmanın öngördüğü ölçüm öngörümü arasındaki farkı hesaplayarak inovasyonu elde etmektedir. Ancak, yine inovasyon için de bir belirsizlik vardır. Bu belirsizlik ise inovasyon kovaryansı olarak adlandırılmaktadır. Kalman kazancı hesaplanırken, durum öngörüm kovaryansı ile inovasyon kovaryansı arasındaki oran elde edildiği için, durum öngörüm kovaryansı yüksek olduğunda kazanç yüksek olmakta ve filtre kestirim üretmek için sensör ölçümünden gelen bilgiyi daha çok kullanmaktadır. Kalman filtre algoritması kestirim üretmek için sensör ölçümünden gelen bilgi ve filtrenin kendi öngörüm değerini kullanmaktadır. Sensör ölçümünden gelen bilgi yani inovasyona ait belirsizlik olan inovasyon kovaryansı yüksek olduğunda, Kalman kazancı düşük olmaktadır.

Bu durumda GKF algoritması, durum öngörüm değerine ağırlık vererek kestirim üretmektedir. Durum öngörüm belirsizliği yüksek olduğunda ise Kalman kazancı azalmakta ve bu kez, algoritma sensör ölçümünden gelen bilgiye daha çok ağırlık vererek kestirim üretmektedir. Dolayısıyla, GKF algoritması, belirsizliği daha düşük olan veriyi kestirim üretmek için daha ağırlıklı olarak kullandığından dolayı, elindeki verileri, yani algoritmanın öngörüm aşamasında kendi modeliyle öngördüğü durum öngörüm verisi ile sensör ölçümünden gelen veriyi kullanarak en iyi kestirimi üretmektedir. Bu nedenle, Kalman filtre tabanlı algoritmalar kullanılarak, sensör ölçüm hatalarından daha düşük ve en iyi kestirimler elde edilebilmektedir. Zafari ve arkadaşlarının yapmış oldukları çalışmada da bunu görmekteyiz. *Bluetooth* sensör ölçümü ile elde edilen RSS ölçümleri, Kalman filtre algoritmasında kullanılmış ve algoritmanın öngörüm aşamasında geliştirilen matematiksel modelle RSS değerlerinin öngörümü yapılmıştır. Ardından, bu öngörüme ait belirsizliği ifade eden durum öngörüm kovaryansı hesaplanmıştır. Daha sonra *bluetooth* sensörü ile RSS ölçümleri elde edilmiş ve elde edilen RSS ölçümlerine dair de Kalman filtre algoritması bir öngörümde bulunarak bu ölçümlere ait, ölçüm öngörümeleri elde edilmiş ve bu öngörümlerdeki belirsizlik hesaplanmıştır. Daha sonra durum öngörüm kovaryansı, inovasyon kovaryansının oranı alınarak Kalman kazancı elde edilmiş ve filtre kestirim üretirken bu kazancı kullanmıştır. Bu nedenle, inovasyon kovaryansı büyük olduğunda kazanç küçülerek, filtre kestirim üretirken daha çok kendi modeliyle öngördüğü durum öngörümelerini kullanmış çünkü kazancın küçük olması demek aslında sensörden gelen bilginin belirsizliğinin yüksek olması demektir. Bu nedenle kestirim üretilirken sensör ölçümünden gelen bilgi fazla kullanılmamış ve daha doğru olan ve belirsizliği daha düşük olan durum öngörümeleri kullanıldığı için Kalman filtresi en iyi RSS kestirimlerini elde etmiştir. Bu nedenle Zafari ve arkadaşlarının yaptıkları çalışmada, sensör ölçümünden gelen bilgi olan RSS ölçümlerinden, daha yüksek doğrulukta kestirimler elde edilmiştir. Ardından, RSS kestirimleri yol kaybı modelinde kullanılarak nesne ile *bluetooth* sensörü okuyucular arasındaki uzaklıklar hesaplanmış ve yakınlık algoritmasında kullanılmıştır. Sonuçta yakınlık algoritmasının Kalman filtre kestirimi sonucu elde edilen RSS verilerini kullanınca, konumlandırma hatasının %32 oranında azalmasının sebebi, Kalman filtre algoritması kullanılarak, sensör ölçümünden daha az hataya sahip ve en iyi RSS kestirimlerinin elde edilmiş olmasıdır. Zafari ve arkadaşları ayrıca konum belirlemek için, ikinci bir kaskat algoritma geliştirmişler ve Kalman filtre ile parçacık filtre algoritmalarını beraber kullanmışlardır. Bu kaskat algortmada, Kalman filtre algoritmasını, RSS ölçümlerinin hatalarını azaltmak için kullanmışlar ve Kalman filtre kestirimi sonucu elde edilen RSS

verilerini parçacık algoritmasında konum belirlemek için kullanmışlardır. Gerçekleştirdikleri deneyde, geliştirilen kaskat algoritmayla elde edilen konum kestirim hatasının, sadece parçacık algoritma kullanarak elde edilen konum kestirim hatasından, 2 ve 3 boyutlu koordinat sistemi üzerinde belirlenen konumlar için sırasıyla, %28,16 ve %25,59 oranında daha az olduğunu gözlemlemişlerdir. Elde edilen sonuçlar, Kalman filtre algoritmasının konum belirleme hatalarını ciddi şekilde azalttığını ve konum bulma uygulamalarında ne kadar başarılı olduğunu göstermektedir (Zafari vd., 2017). Başka bir çalışmada Cholz ve arkadaşları, UGB sensörü kullanarak geliştirilen kapalı alan hedef takip sisteminde trilaterasyon, GKF, parçacık filtre, çok boyutlu ölçekleme ile ağırlıklandırılmış en küçük kareler (İng. *Weighted Least Square with Multidimensional Scaling*) ve mesafe küçültme ile en küçük kareler kestiricisi (İng. *Least Square with Distance Contraction*) algoritmalarının performanslarını karşılaştırmışlardır. Mesafe küçültme ile en küçük kareler kestiricisi algoritmasında, mesafe küçültme yöntemi sensör ölçüm hatalarını azaltmak için kullanılmakta olup, ölçüm hatalarının azaltılması ile en küçük kareler kestiricisi algoritmasının, kestirim hatasının azaltılması amaçlanmaktadır. Gerçekleştirilen simülasyon çalışmasında, 10 adet hedef takip edilmiş ve sensör sayısı 3'ten 10'a kadar arttırılarak, sensör sayısının algoritmaların performansları üzerindeki etkisi incelenmiştir. Sensörler arası mesafe 10 metre ve 12,5 metre olduğu durum için 2 farklı senaryo oluşturulmuştur. Her iki senaryo için, algoritmaların hedef takip performansları sensör sayısına bağlı olarak karşılaştırılmıştır. Sensörler arası mesafe 10 metre olduğu durumda, en fazla konum belirleme hatası trilaterasyon algoritması kullanıldığında elde edilmiştir. Trilaterasyon algoritmasında sensör sayısının arttırılması, algoritmanın konum belirleme hatasını değiştirmemiştir. Bunun nedeni, algoritmanın 3 adet sensör kullanması ve sensör sayısı artsa dahi algoritma yine 3 adet sensör kullanarak konum belirlediği için, trilaterasyon algoritması için konum belirleme hatası sensör sayısından bağımsızdır. Konum belirleme hatası en düşük olan algoritmanın parçacık filtre algoritması olduğu görülmüş ve sensör sayısı arttıkça, algoritmanın konum belirleme hatasının da azaldığı görülmüştür. 5 adet sensör kullanıldığında GKF, çok boyutlu ölçekleme ile ağırlıklandırılmış en küçük kareler (İng. *Weighted Least Square with Multidimensional Scaling*) ve mesafe küçültme ile en küçük kareler kestiricisi (İng. *Least Square with Distance Contraction*) algoritmalarının kestirim hatalarının birbirlerine çok yakın olduğu görülmüş ve bu algoritmalar için optimum sensör sayısının 5 adet kullanıldığı durumda elde edildiği belirtilmiştir. GKF algoritması için 5 sensörden fazla sensör kullanıldığında, kullanılan fazla sensörlerin hedefle görüş hattında olmamasından dolayı, GKF algoritmasının kestirim hatasının arttığı belirtilmiştir. 2.

senaryoda, sensörler arası mesafe 10 metreden 12,5 metreye çıkarılmıştır. Bu durumda sensörler, hedef nesneden daha uzakta olduğu için her bir algoritmanın konum belirleme hatası artmıştır. Aslında bu beklenen sonuçtur. Çünkü, sensör ile hedef nesne arasındaki uzaklık arttıkça, sensörle hedef nesne arasında görüş hattı olmama veya azalması durumunda sinyalin daha çok kırınım ve yansımaya uğraması sonucu ölçüm hataları artacağı için, algoritmaların konum belirleme hataları da artmaktadır. Yapılan çalışmada da bazı sensörlerin nesneden daha uzakta kalması sebebiyle algoritmaların hatalarının arttığı belirtilmiştir. Bu durumda en iyi performansa sahip algoritmanın parçacık filtre algoritması ve en çok hataya sahip algoritmaların ise trilaterasyon ve GKF algoritması olduğu belirtilmiştir. Sensör sayısı 3, 4, 5 ve devam ederek 10'a kadar arttırılmıştır. Sensör sayısı 3'ten 4 ve 5'e çıkarıldığında her iki durum için de GKF ve diğer algoritmaların hatalarının azaldığı ancak 6 sensörden fazla sensör kullanıldığı durumda hatanın daha fazla azalmadığı görülmektedir. Cholíiz ve arkadaşları bunun sebebinin ise hedefin 5 sensörden fazla sensör kullanıldığı durumda, bu sensörlerin kapsama alanı dışına çıktığı ve bu nedenle sensörlerden, hedefe artık ölçüm gelmemesi olduğunu belirtmişlerdir (Cholíiz vd., 2011). Bu mantıklı bir sonuçtur. Çünkü, GKF algoritmasına sensör ölçümü gelmediğinde, bir önceki anda yapılan kestirim güncellenemediği için, artık bir önceki anda elde edilen kestirim sonucu bir sonraki anda elde edilen ve diğer sonraki anlarda elde edilen kestirim sonuçlarına eşittir. Bu nedenle hata sabit kalmıştır. Başka bir çalışmada, Oliveira ve arkadaşları *bluetooth* sensörü kullanarak kapalı alan konumlandırma probleminde, multilaterasyon algoritması ile Kalman filtre algoritmasını beraber kullanarak konum tespiti yapmışlardır. *Bluetooth* sensörü kullanarak her bir sensör okuyucusu için elde edilen RSS ölçümleri yol kaybı modeli (İng. *Path Loss Model*) kullanılarak, okuyucular ile konumu tespit edilmek istenen nesne arasındaki uzaklık verisine dönüştürülmüştür. Bu uzaklık verileri ise, multilaterasyon algoritmasında kullanılarak konum belirlenmiştir. Daha sonra Kalman filtre algoritması kullanılmış ve multilaterasyon algoritması sonucu elde edilen konum verileri, Kalman filtre algoritmasında durum öngörümü aşamasında, algoritmanın öngördüğü konum olarak kullanılmıştır. Ardından, bu öngörüm değeri, sensör ölçümleri ile güncellenerek algoritma kestirim üretmiştir. Kalman filtre algoritmasında kazanç kullanılarak filtrenin öngördüğü konum verileri, sensör ölçümleriyle güncellenmekte ve sensör ölçümlerinden ve durum öngörümü sonucu elde edilen verilerden daha yüksek doğrulukla kestirim üretilmektedir. Bu nedenle, multilaterasyon algoritması sonucu elde edilen konum verilerindeki hataların da bu şekilde azaltılması amaçlanmıştır. Multilaterasyon algoritması tek başına kullanıldığında, konum belirleme hatasının 3,27 metreye kadar çıktığı ve ortalama

hatanın ise 2,57 metre olduğu belirtilmiştir. Multilaterasyon algoritması, Kalman filtre algoritmasıyla beraber kullanıldığında ise ortalama hatanın 2,33 metreye düştüğü ve maksimum hatanın ise 2,97 metreye kadar çıktığı gözlemlenmiştir. Kalman filtre algoritması kullanılarak elde edilen kestirim sonucu, multilaterasyon algoritmasıyla elde edilen konum belirleme hatasının azalmıştır. Bunun sebebi, Kalman filtre algoritmasının durum öngörüm aşamasında, multilaterasyon algoritmasıyla elde edilen konum verilerinin kullanılması ve Kalman filtre algoritmasının ölçümle güncelleme aşamasında sensör ölçümleri ve Kalman kazancının kullanılmasıyla, durum öngörümü aşamasında elde edilen ve multilaterasyon algoritmasıyla hesaplanmış konum verilerinin düzeltilerek, Kalman filtre algoritmasının daha doğru kestirim üretmesini sağlamaktır. Burada, multilaterasyon algoritması sonucu elde edilen verilerin hatalarının Kalman filtre algoritmasıyla azaltılmasında, Kalman kazancı önemli rol oynamaktadır. Kalman kazancı, Kalman filtre algoritmasının durum öngörümü aşamasında elde edilen durum öngörümüne yani bu çalışma için multilaterasyon algoritmasıyla bulunan konum verilerine mi daha çok ağırlık vererek kestirim üreteceğine, yoksa sensör ölçümüne mi ağırlık vererek kestirim üreteceğine karar vermekte ve filtre buna göre kestirim üretmektedir. Eğer durum öngörüm kovaryansı yani multilaterasyon algoritması sonucu elde edilen konum verilerinin belirsizliğini ifade eden durum öngörüm kovaryansı yüksek ise Kalman kazancı artmakta ve böylece algoritma kestirim üretmek için multilaterasyon algoritması sonucu elde edilen konum verilerine güvenmediği için, sensör ölçümüne ağırlık vererek kestirim üretmektedir. Bu nedenle, Kalman filtre algoritması sonucu, multilaterasyon algoritmasıyla bulunan konum verilerinden daha doğru konum verileri elde edilmektedir. Bu nedenle, iyi bir kestirim elde edebilmek için Kalman kazancının doğru bir şekilde ayarlanması gerekmektedir. Kalman kazancının ayarlanması için de Kalman filtre algoritmasının tasarım parametresi olan filtrenin süreç gürültü varyansı değiştirilmektedir (De Oliveira vd., 2021). Başka bir çalışmada Kong ve diğerleri, kapalı alanda konumlandırma problemi için *bluetooth* sensörü sensörü kullanmışlar ve GKF ile üçgenleme algoritmalarının performanslarını karşılaştırmışlardır. Aynı zamanda, sensör sayısındaki artışın algoritmaların konum belirleme hataları üzerindeki etkilerini de araştırmışlardır. Başlangıçta 177 tane sensör kullanmışlar ve sensör sayısını 140, 100, 80, 60, 40, 30 ve 20 olacak şekilde sırayla azaltarak, algoritmaların hatalarındaki değişiklikleri incelemişlerdir. GKF algoritması için sensör sayısının 177'den 60'a kadar düşürmenin algoritmanın kestirim hatası üzerinde çok fazla değişiklik oluşturmadığı, sadece kestirim hatasında çok küçük azalışlar olduğu görülmüştür. GKF algoritması için, sensör sayısı sırasıyla 177'den 140, 100, 80 ve 60'a düşürüldüğünde, ortalama kestirim hatasının 2 metre

ile 2,5 metre arasında olduğu ve birbirlerine çok yakın olduğu görülmüştür. Sensör sayısı 177'den 60'a düşürüldüğünde ortalama hata 2 metreden, 2,5 metreye çıkarken, sensör sayısı 177'den 140'a düşürüldüğünde ortalama hatanın 2 metreden 2,2 metreye çıktığı, sensör sayısı 100'e düşürüldüğünde hatanın 2,3 metreye çıktığı görülmüştür. Ancak, sensör sayısı 20'ye düşürüldüğünde GKF hatasının ortalama kestirim hatası daha çok artmış ve 4,3 metre civarına kadar çıkmıştır. Sensör sayısı 177 iken, GKF algoritmasının kestirim hatası 2 metre iken, sensör sayısı 20'ye düşürüldüğünde hata 4,3 metre olmuş ve ciddi oranda artarak yaklaşık 2 katı olmuştur. Üçgenleme algoritmasının hatası ise, 177 sensör kullanıldığı durumda 2,30 metre iken 40 sensör kullanıldığında 3,0144 metreye kadar çıktığı görülmektedir. Üçgenleme algoritması ile GKF algoritmasının konum belirleme hataları karşılaştırıldığında, GKF algoritmasının daha yüksek doğruluklarla konum belirlenebildiği görülmektedir. Bunun sebebi, sensör ölçüm hatalarının üçgenleme algoritmasında azaltılamaması ve bu hataların üçgenleme algoritmasında ciddi konum belirleme hatalarına sebep olmasıdır. Çünkü sensör ölçümleri üçgenleme algoritmasında hedef nesne ve sensör arasındaki uzaklık ile açı verileri olarak kullanıldığından, sensör ölçümleri hatalı olduğunda algoritmada kullanılan açı ve uzaklık verileri de hatalı olduğu için, konum belirleme hataları da yüksek olmaktadır. Ancak GKF algoritmasıyla, sensörlerin ölçüm hatalarından daha az hataya sahip en optimum konum kestirimleri yapıldığı için, GKF algoritması kullanılarak üçgenleme, multilaterasyon, parmak izi ve trilaterasyon gibi algoritmalarından daha yüksek doğruluklarla konum belirlenebilmektedir (Kong vd., 2023).

Literatür taraması sonucu elde edilen bilgiler, GKF algoritması kullanılarak, kapalı alanda nesne konumlandırma problemleri için kullanılan diğer algoritmalar olan trilaterasyon, üçgenleme, multilaterasyon ve parmak izi gibi algoritmalarından, daha yüksek doğruluklarla konum belirlenebildiğini göstermektedir. Multilaterasyon tabanlı algoritmalar ve parmak izi algoritmasıyla konum belirleme hatasının yüksek olmasının sebebi, sensör ölçüm hatalarının azaltılmadan, sensör ölçümlerinin bu algoritmalarda direkt olarak kullanılmasıdır. Kapalı alanda sensör ölçüm hataları yüksek olmaktadır. Bunun nedeni, sensör sinyallerinin kapalı alanlarda bulunan nesnelerden dolayı kırınım, yansıma ve çok yollu yayılım gibi problemlerle karşılaşmaları sonucu sinyalin sensör alıcısına ulaştığı anda elde edilen RSS, varış zamanı, varış açısının değişmesidir. Multilaterasyon, parmak izi, üçgenleme gibi algoritmalarda ise bu ölçüm yöntemleriyle elde edilen sensör ölçümleri kullanılarak konum belirlendiği için, ölçüm hataları yüksek olduğunda algoritmalarda kullanılan ölçüm verileri hatalı olmaktadır. Bu nedenle, bu algoritmalar kapalı alanlarda konum belirleme

problemlerinde yüksek konumlandırma hatalarına sahip olmaktadır. Bundan dolayı, multilaterasyon tabanlı algoritmalar ile konum belirlenmeden önce, sensörlerin ölçüm hatalarının azaltılması için ek yöntemlere ihtiyaç duyulmaktadır. Ancak, kapalı alanda konum belirleme problemlerinde, genişletilmiş Kalman filtre algoritması kullanılarak en optimum kestirimler elde edilebildiği için çok daha az hatalarla konum belirlenebilmektedir. Bu nedenle, Kalman filtre tabanlı algoritmalar konum belirleme ve navigasyon problemlerinde günümüzde en sık kullanılan ve en gelişmiş algoritmalar arasındadır.

Bu tez çalışmasında, nesne hareketinin, sensör geometrisinin ve kullanılan sensör sayısının GKF algoritmasının kestirim performansı üzerindeki etkileri incelenmiştir. Sensör geometrisi, GKF algoritmasında Kalman kazancını etkilemekte ve Kalman kazancının etkilenmesi sebebiyle de, kestirim hatası da sensör geometrisinden etkilenmektedir. Bu nedenle, sensör geometrisi GKF algoritmasının kestirim hatası üzerinde etkili olduğundan, algoritmanın konum kestirim hatasının ve performansının incelenmesi için, kullanılan sensör geometrisinin algoritma üzerindeki etkilerinin araştırılması önemli bir konudur.

3. MATERYAL ve YÖNTEM

Bu bölümde, tez çalışmasında kullanılan yöntemlerin teorik açıklamalarına yer verilmiştir. Bu bölüm, 4 alt bölümden oluşmaktadır. Birinci alt bölümde, tezde kullanılan kinematik hareket modelleri anlatılmıştır. İkinci alt bölümde, UGB sensörünün özellikleri ve ölçüm modeline yer verilmiştir. Üçüncü alt bölümde, genişletilmiş Kalman filtre algoritması anlatılmıştır. Dördüncü ve son alt bölümde ise tez çalışması kapsamında kullanılan farklı sensör geometrilerinden oluşan senaryolar tanıtılmıştır.

3.1. Kinematik Modeller

Tez çalışması kapsamında, senaryoların oluşturulması için çeşitli hareket modelleri kullanılmıştır. Konumu tespit edilmek istenen hedef nesnenin hareketinin modellenmesi için koordineli dönüş hareket modeli kullanılmıştır. Genişletilmiş Kalman filtre algoritmasının tasarımıda ise, algoritmanın durum öngörüm modeli için, yaklaşık sabit hızlı düzgün doğrusal hareket modeli kullanılmıştır. Durum öngörüm aşamasında, nesne hareketi yaklaşık sabit hızlı düzgün doğrusal hareket modeli kullanılarak öngörülmüştür. Kinematik modeller, 2 boyutlu kartezyen koordinat sistemi üzerinde modellenmiştir.

3.1.1. Koordineli dönüş hareket modeli

Tez çalışması kapsamında, konumu tespit edilmek istenen hedef nesne, 2 boyutlu kartezyen koordinat sisteminde koordineli dönüş hareketi yapmaktadır. Koordineli dönüş hareketi, herhangi bir nesnenin sabit açısal hızla yaptığı dönüş hareketidir (Rong Li ve Jilkov, 2003). 2 boyutlu kartezyen koordinat sisteminde, koordineli dönüş hareket modeline ait eşitlikler aşağıda verilmektedir. Koordineli dönüş hareket modeli oluşturulurken, eksenler arasında bir ilişki olmadığı x ve y eksenlerindeki hareketlerin birbirlerinden bağımsız olduğu varsayılmaktadır.

$$\mathbf{X}(k+1) = \mathbf{F} * \mathbf{X}(k) + \mathbf{B} * \mathbf{U}(k) \quad (3.1)$$

$$\mathbf{X} = \begin{bmatrix} x_{\text{pos}} \\ x_{\text{hız}} \\ y_{\text{pos}} \\ y_{\text{hız}} \end{bmatrix} \quad (3.2)$$

$$\mathbf{F} = \begin{bmatrix} 1 & \frac{\sin \Omega T}{\Omega} & 0 & -\frac{1 - \cos \Omega T}{\Omega} \\ 0 & \cos \Omega T & 0 & -\sin \Omega T \\ 0 & \frac{1 - \cos \Omega T}{\Omega} & 1 & \frac{\sin \Omega T}{\Omega} \\ 0 & \sin \Omega T & 0 & \cos \Omega T \end{bmatrix} \quad (3.3)$$

$$B = \begin{bmatrix} 0.5T^2 & 0 \\ T & 0 \\ 0 & 0.5T^2 \\ 0 & T \end{bmatrix} \quad (3.4)$$

$$U(k) \sim N(0, \sigma_v) \quad (3.5)$$

Eş. 3.1’de koordineli dönüş hareket modeli için durum denklemi verilmiştir. Eş. 3.1’de, X nesnenin konum ve hızının x ve y eksenlerindeki bileşenlerinden oluşan durum değişken vektörüdür. F matrisi, k anından $k + 1$ anına geçmemizi sağlayan durum geçiş matrisidir. $U(k)$ vektörü, harekete etki eden süreç gürültü matrisi ve B ise süreç gürültü katsayı matrisidir. Süreç gürültüsü, harekete bozucu olarak etki etmekte olup, 0 ortalamalı beyaz *Gauss* gürültüsüdür. Verilen eşitliklerdeki T ifadesi, örnekleme aralığını ve Ω ise açısal hızı belirtmektedir. Süreç gürültüsü, nesne hareketine ivme olarak etki etmektedir. İvme ise, nesnenin pozisyonuna $0.5 T^2$ ve hızına ise T ağırlığında etki etmektedir.

3.1.2. Yaklaşık sabit hızlı düzgün doğrusal hareket modeli

Yaklaşık sabit hızlı düzgün doğrusal hareket modelinde, hedef nesnenin pozisyon ve hız durumları kullanılmaktadır. Bu modelde, nesnenin yapmış olduğu doğrusal hareket modellenmektedir. Yaklaşık sabit hızlı düzgün doğrusal hareket modelinde, nesne sabit hız ile hareket etmekte ve nesne hareketine süreç gürültüsü etki etmektedir. Süreç gürültüsü, 0 ortalamalı beyaz *Gauss* gürültüsüdür (Rong Li ve Jilkov, 2003). Bu nedenle, nesnenin hızı 0 ortalamalı beyaz *Gauss* gürültüsü ile bozulmaktadır. Tez çalışması kapsamında geliştirilen simülasyonlarda, genişletilmiş Kalman filtre algoritmasının öngörüm aşamasında, yaklaşık sabit hızlı düzgün doğrusal hareket modeli kullanılmıştır. 2 boyutlu kartezyen koordinat sisteminde, yaklaşık sabit hızlı düzgün doğrusal hareket modeline ait eşitlikler aşağıda verilmiştir. Eksenler arasında bir ilişki olmadığı x ve y eksenlerindeki hareketlerin birbirlerinden bağımsız olduğu varsayılmaktadır.

$$X(k + 1) = F * X(k) + B * U(k) \quad (3.6)$$

$$X = \begin{bmatrix} x_{\text{pos}} \\ x_{\text{hız}} \\ y_{\text{pos}} \\ y_{\text{hız}} \end{bmatrix} \quad (3.7)$$

$$F = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.8)$$

$$B = \begin{bmatrix} 0.5T^2 & 0 \\ T & 0 \\ 0 & 0.5T^2 \\ 0 & T \end{bmatrix} \quad (3.9)$$

$$U(k) \sim N(0, \sigma_v) \quad (3.10)$$

Eş. 3.6’da, X nesnenin konum ve hızının x ve y eksenlerindeki bileşenlerinden oluşan durum değişken matrisidir. F matrisi, k anından $k + 1$ anına geçmemizi sağlayan durum geçiş matrisidir. $U(k)$, harekete etki eden süreç gürültü matrisi ve B ise süreç gürültü katsayı matrisidir. Süreç gürültüsü, harekete bozucu olarak etki etmekte olup, 0 ortalamalı beyaz Gauss gürültüsü olarak modellenmektedir. Süreç gürültüsü, nesnenin hareketine ivme olarak etki etmekte ve ivme ise, nesnenin pozisyonuna $0,5 T^2$ ve hızına ise T ağırlığında etki etmektedir. Verilen eşitliklerdeki T ifadesi, örnekleme aralığını belirtmektedir.

3.2. Ultra Geniş Bant Sensörünün Özellikleri ve Ölçüm Modeli

Ultra geniş bant sensörü, geniş bant aralığında veri iletimi yapan bir sensördür. UGB sensör sinyali geniş bant aralığına sahip olması sebebiyle, farklı frekans bileşenlerinden oluşmaktadır. UGB sensör sinyallerinin farklı frekans bileşenlerinden oluşması, kapalı alanda nesne konumlandırma problemlerinde, UGB sensörüne bazı avantajlar sağlamaktadır (Liu vd., 2007). Kapalı alanda hedef nesne ile UGB okuyucular arasında başka nesneler olduğunda, UGB okuyucu ile konumu tespit edilmek istenen hedef nesne arasındaki görüş hattı tamamen ortadan kalkmakta veya azalmaktadır. Bu durumda, bazı frekansa sahip UGB sensör sinyalleri kapalı alanda bulunan nesnelere çarptığında, kırınım ve yansıma gibi sorunlarla karşılaşsa bile, farklı frekansa sahip sinyaller bu engellerden geçerek hedef nesneye ulaşabilmektedir. Bu sinyaller, hedef nesne ile UGB okuyucu arasındaki uzaklığı ölçmek için yeterli olduğu için, nesne ile UGB okuyucu arasında görüş hattı olmadığı durumlarda da UGB sensörünün ölçüm hatası düşük olmaktadır (Liu vd., 2007). Bu ise, kapalı alanda nesne konumlandırma problemleri için, UGB sensörünün sunmuş olduğu bir avantajdır. Kapalı alanlarda konum belirlemek için UGB sensörünün farklı frekans bileşenlerinden oluşan sinyaller içermesinin bir başka avantajı ise UGB sensör sinyallerinin, ortamda bulunan diğer RF sinyallerle girişime girmemeleridir. Bazı frekansa sahip UGB sensör sinyalleri, kapalı alanda bulunan diğer RF sinyallerle girişime uğrasa bile, farklı frekansa sahip UGB sensör sinyalleri girişime uğramadan hedef nesneye ulaşmakta ve

ölçüm gerçekleştirilmektedir. Hedef nesneye ulaşan UGB sensör sinyalleri, ortamdaki diğer RF sinyalleriyle girişime uğramadığından dolayı bozulmadığı için, ölçüm hatası düşük olmaktadır (Liu vd., 2007). Ölçüm hatasının düşük olması, ortamda bulunan ve UGB sensör ölçümüne gürültü olarak etki eden diğer RF sinyallerinin olması durumunda, UGB sensörünün sağlamış olduğu bir avantajdır.

Bu tez çalışmasında, UGB sensörü, konumu tespit edilmek istenen hedef nesne ile UGB okuyucu arasındaki uzaklık ölçümü için kullanılmıştır. UGB sensörü, etiket ve okuyucudan oluşmaktadır. UGB sensör okuyucuları, kapalı alanda konumu bilinen referans noktalara yerleştirilmektedir. UGB sensör etiketi ise konumu tespit edilmek istenen hedef nesne üzerinde bulunmaktadır. UGB sensörünün ölçümü, Eş. 3.11 ile verilen eşitlik kullanılarak modellenmiştir.

$$h = R_{\text{gerçek}} + n \quad (3.11)$$

$$R_{\text{gerçek}} = \sqrt{(x_{\text{pos}} - R_{\text{ox}})^2 + (y_{\text{pos}} - R_{\text{oy}})^2} \quad (3.12)$$

$$n \sim N(0, \sigma^2) \quad (3.13)$$

Eş 3.12’de R_{ox} ve R_{oy} ifadeleri UGB sensör okuyucusunun bulunduğu konumun sırasıyla x ve y eksenlerindeki bileşenlerini belirtmektedir. Ayrıca x_{pos} ve y_{pos} ifadeleri ise üzerinde UGB etiketi bulunan nesnenin, hareketi esnasında bulunmuş olduğu konumların sırasıyla x ve y eksenlerindeki bileşenlerini ifade etmektedir. Her bir UGB okuyucu için öklid uzaklık denklemi, ayrı ayrı kullanılarak, her bir okuyucu ile nesne arasındaki gerçek uzaklık değerleri elde edilmiştir. $R_{\text{gerçek}}$, nesne ile UGB sensör okuyucusu arasındaki ölçülmek istenen gerçek uzaklık değerini ve h ise UGB sensörü tarafından, UGB sensör okuyucusu ile konumu tespit edilmek istenen hedef nesne arasındaki ölçülen uzaklığı belirtmektedir. UGB sensörü, ölçülmek istenen bu gerçek uzaklık değerini, eklenebilir ölçüm gürültüsü varlığında ölçmektedir. Bu nedenle sensör ölçümü elde edilirken, gerçek uzaklık değerine, sensörün ölçüm gürültüsü eklenmiştir. Kapalı alanda UGB sensörü ile ölçüm yaparken, UGB etiket tarafından, UGB sensör okuyucusuna gönderilen sinyal, ortamda bulunan nesnelere çarptığında kırınım, yansıma ve çok yollu yayılım gibi problemlerle karşılaşmakta ve bu problemler sensör ölçümüne gürültü olarak etki etmektedir. Sensör ölçümü modellenirken, bu ölçüm gürültüleri 0 ortalamalı, σ^2 varyanslı beyaz Gauss gürültüsü olarak modellenmiş ve Eş. 3.11’de n ile gösterilmiştir.

3.3. Genişletilmiş Kalman Filtre Algoritması

Genişletilmiş Kalman filtre algoritması, dinamik bir sistemin durum denklemi veya sensör ölçüm modelinin veya her ikisinin de doğrusal olmadığı durumlarda kullanılan optimal durum kestiricisidir (B. Huang ve Wang, 2006). Genişletilmiş Kalman filtre algoritması kullanılarak, dinamik bir sistemin durum kestirimleri elde edilmektedir (B. Huang ve Wang, 2006). Dinamik bir sistemin durum denklemi, Eş. 3.13 ile verilmektedir.

$$\mathbf{X}(k + 1) = \mathbf{F} * \mathbf{X}(k) + \mathbf{B} * \mathbf{U}(k) \quad (3.14)$$

Eş. 3.13'te $\mathbf{X}(k + 1)$, sistemin $k + 1$ anındaki durum vektörü, $\mathbf{X}(k)$, sistemin k anındaki durum vektörü, $\mathbf{U}(k)$, hedef hareketindeki öngörülemez bozucuları modellemek için kullanılan süreç gürültüsü ve $\mathbf{B}$ ise süreç gürültüsü katsayısıdır. $\mathbf{F}$ ise k anından $k + 1$ anına geçmemizi sağlayan durum geçiş matrisidir. Süreç gürültüsü, 0 ortalamalı beyaz Gauss gürültüsü olarak modellenmektedir. Genişletilmiş Kalman filtre algoritmasında kullanılan sensör ölçüm modeli ise Eş. 3.14'te verilmiştir.

$$\mathbf{Z}(k) = \mathbf{H} * \mathbf{X}(k) + \mathbf{W}(k) \quad (3.15)$$

Eş. 3.14'te $\mathbf{Z}(k)$, k anındaki sensör ölçümü, $\mathbf{H}$ ölçüm matrisi, $\mathbf{X}(k)$, k anındaki durum vektörü ve $\mathbf{W}(k)$ ise ölçüm gürültüsü olup 0 ortalamalı beyaz Gauss gürültüsü olarak modellenmektedir. Ölçüm ve süreç gürültüleri birbirlerinden bağımsızdır. Genişletilmiş Kalman filtre algoritmasında doğrusal olmayan durum veya ölçüm fonksiyonu Taylor serisi açılımı yapılarak doğrusallaştırılmaktadır (B. Huang ve Wang, 2006).

$$\mathbf{F} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \bigg|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k}} \quad (3.16)$$

$$\mathbf{H} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \bigg|_{\mathbf{x}=\hat{\mathbf{x}}_{k+1|k}} \quad (3.17)$$

Eş. 3.16'da doğrusal olmayan durum geçiş matrisi için Taylor serisi açılımı, durum kestiriminin o anki değeri etrafında yapılmaktadır. Eş. 3.17'de ise $\mathbf{h}$ doğrusal olmayan ölçüm fonksiyonu için Taylor serisi açılımı, durum öngörüm değeri etrafında yapılmaktadır. Taylor serisi açılımı sonsuz terime sahiptir. Ancak, doğrusal olmayan ölçüm ve durum fonksiyonları, Taylor serisi açılımı yapılarak doğrusallaştırılırken, sonsuz terime sahip bu açılım, bu açılımın sadece ilk iki terimiyle ifade edilerek Jakobyan matris hesaplanmaktadır. Bu durumda sonsuz terime sahip olan Taylor serisi açılımı, sadece ilk iki terimiyle ifade edildiğinden dolayı doğrusallaştırma hataları oluşmaktadır. Doğrusallaştırma hatalarından

dolayı da GKF algoritmasının kestirim hatası artmaktadır. Doğrusallaştırmadan kaynaklanan hatalar, doğrusalsızlığın yüksek olduğu durumlarda, filtrenin ıraksamasına dahi sebep olabilmektedir (B. Huang ve Wang, 2006). Bu nedenle, doğrusal olmayan ölçüm ve durum geçiş fonksiyonun doğrusallaştırılmasından kaynaklanan hataların, GKF algoritmasının performansını düşürmesi ve doğrusalsızlığın yüksek olduğu durumda filtrenin ıraksamasına sebep olması, GKF algoritmasının önemli bir dezavantajıdır.

Tez çalışması kapsamında doğrusal olmayan ölçüm fonksiyonunun doğrusallaştırılması için H Jakobyian matrisi, Eş. 3.20’de gösterildiği şekilde hesaplanmıştır. Sensörün ölçüm fonksiyonunun, X durum vektörüne göre kısmi türevi alınarak hesaplanmıştır. Durum vektörü, nesnenin hareketi esnasındaki konum ve hızının, 2 boyutlu kartezyen koordinat sistemi üzerindeki x ve y bileşenlerinden oluşmaktadır. Eş 3.19’da gösterilen $\hat{X}$ durum öngörüm vektörü, GKF algoritmasının öngörüm aşamasında elde edilen, X durum vektörünün öngörümüdür.

$$X = \begin{bmatrix} x_{pos} \\ x_{hız} \\ y_{pos} \\ y_{hız} \end{bmatrix} \quad (3.18)$$

$$\hat{X} = \begin{bmatrix} x_{pos}' \\ x_{hız}' \\ y_{pos}' \\ y_{hız}' \end{bmatrix} \quad (3.19)$$

$$H = \frac{\partial h}{\partial x} \big|_{x=\hat{X}} = [H_{11} \ H_{12} \ H_{13} \ H_{14}] \quad (3.20)$$

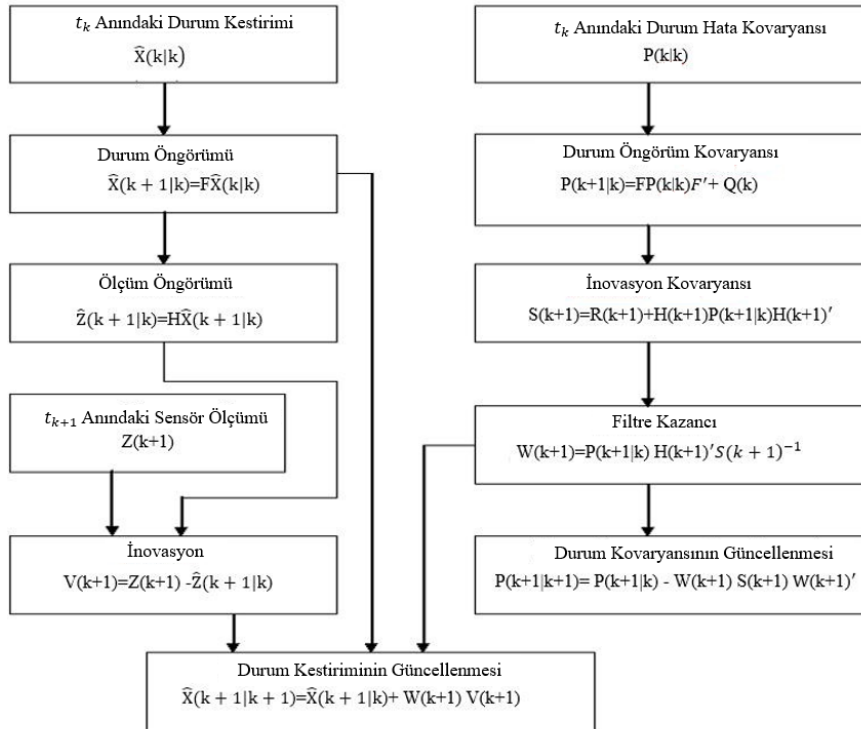
$$H_{11} = \frac{\partial h}{\partial x_{pos}} \big|_{x_{pos}=x_{pos}'} = \frac{x_{pos}' - R_{ox}}{\sqrt{(x_{pos}' - R_{ox})^2 + (y_{pos} - R_{oy})^2}} \quad (3.21)$$

$$H_{12} = \frac{\partial h}{\partial x_{hız}} \big|_{x_{hız}=x_{hız}'} = 0 \quad (3.22)$$

$$H_{13} = \frac{\partial h}{\partial y_{pos}} \big|_{y_{pos}=y_{pos}'} = \frac{y_{pos}' - R_{oy}}{\sqrt{(x_{pos} - R_{ox})^2 + (y_{pos}' - R_{oy})^2}} \quad (3.23)$$

$$H_{14} = \frac{\partial h}{\partial y_{hız}} \big|_{y_{hız}=y_{hız}'} = 0 \quad (3.24)$$

Yukarıda verilen eşitliklerdeki x_{pos} ve y_{pos} ifadeleri sırasıyla, nesnenin hareketi esnasında bulunduğu konumun, 2 boyutlu kartezyen koordinat sistemi üzerindeki x ve y eksenlerindeki bileşenleridir. $x_{hız}$ ve $y_{hız}$ ifadeleri ise sırasıyla, nesne hızının, 2 boyutlu kartezyen koordinat sistemi üzerindeki x ve y eksenindeki bileşenleridir. Yukarıda verilen eşitliklerde x_{pos}' ve y_{pos}' ifadeleri ise sırasıyla, GKF algoritmasının, nesnenin hareketi esnasında bulunduğu konumun x ve y eksenlerindeki bileşenlerinin öngörüm değerleri olup bu değerlerin hesaplanması için kullanılan eşitlik Eş. 3.25'te verilmiştir. Yukarıda verilen eşitliklerde bulunan R_{ox} ve R_{oy} ifadeleri ise sensör konumunun, 2 boyutlu kartezyen koordinat sistem üzerinde x ve y eksenlerindeki bileşenleridir. Tez çalışması kapsamında, sensör geometrisinin, GKF algoritmasının kestirim hatası üzerindeki etkileri incelenmiştir. Farklı sensör geometrileri elde etmek için, sensörlerin bulundukları yerler değiştirildiğinde R_{ox} ve R_{oy} ile gösterilen sensör konumunun x ve y eksenlerindeki bileşenleri değiştiği için, Jakobyian matris değişmektedir. Jakobyian matris değiştiğinde ise, Eş. 3.29'da verilen Kalman kazancı değişmekte ve kazanç değiştiği için de Eş. 3.31'de verilen, kestirim değeri değişmektedir. Dolayısıyla, sensör geometrisinin değişmesine bağlı olarak, GKF algoritmasının kestirim değeri ve buna bağlı olarak kestirim hatası değiştiği için, sensör geometrisi, GKF algoritmasının kestirim hatasını etkilemektedir.



Şekil 3.1. Genişletilmiş Kalman filtre algoritmasının blok şeması (Bar-Shalom vd., 2001).

Şekil 3.1’de, GKF algoritmasının blok şeması gösterilmiştir. GKF algoritmasının eşitlikleri ise aşağıda verilmiştir.

$$\hat{\mathbf{X}}(k+1|k) = \mathbf{F}\hat{\mathbf{X}}(k|k) \quad (3.25)$$

$$\mathbf{P}(k+1|k) = \mathbf{F}\mathbf{P}(k|k)\mathbf{F}' + \mathbf{Q}(k) \quad (3.26)$$

$$\hat{\mathbf{Z}}(k+1|k) = \mathbf{H}\hat{\mathbf{X}}(k+1|k) \quad (3.27)$$

$$\mathbf{S}(k+1) = \mathbf{R}(k+1) + \mathbf{H}(k+1)\mathbf{P}(k+1|k)\mathbf{H}(k+1)' \quad (3.28)$$

$$\mathbf{W}(k+1) = \mathbf{P}(k+1|k)\mathbf{H}(k+1)'\mathbf{S}(k+1)^{-1} \quad (3.29)$$

$$\mathbf{V}(k+1) = \mathbf{Z}(k+1) - \hat{\mathbf{Z}}(k+1|k) \quad (3.30)$$

$$\hat{\mathbf{X}}(k+1|k+1) = \hat{\mathbf{X}}(k+1|k) + \mathbf{W}(k+1)\mathbf{V}(k+1) \quad (3.31)$$

$$\mathbf{P}(k+1|k+1) = \mathbf{P}(k+1|k) - \mathbf{W}(k+1)\mathbf{S}(k+1)\mathbf{W}(k+1)' \quad (3.32)$$

Eş. 3.25’de verilen $\hat{\mathbf{X}}(k+1|k)$ ifadesi, GKF algoritmasının durum öngörüm değeridir. Eş. 3.26’da verilen $\mathbf{P}(k+1|k)$ ifadesi ise, durum öngörüm kovaryans değeri olup, durum öngörümündeki belirsizliği ifade etmektedir. Eş. 3.27’de $\hat{\mathbf{Z}}(k+1|k)$ ifadesi, ölçüm öngörümü olup, sensörün gerçekte yaptığı ölçümün, GKF algoritması tarafından öngörülmüş değeridir. Eş. 3.28’de, $\mathbf{S}(k+1)$ ifadesi, inovasyon kovaryansı olup inovasyondaki belirsizliği ifade etmektedir. İnovasyon, sensörün gerçekte yaptığı ölçümle, ölçüm öngörümü arasındaki fark olduğundan, ölçüm öngörümündeki belirsizlikten dolayı, bir belirsizlik içermektedir. Bu belirsizlik inovasyon kovaryansı olarak adlandırılmaktadır. Eş. 3.29’da verilen $\mathbf{W}(k+1)$ ifadesi Kalman kazancı ve Eş. 3.30’da gösterilen $\mathbf{V}(k+1)$ ise inovasyon olup, sensörün gerçekte yaptığı ölçüm ile ölçüm öngörümü arasındaki farkı ifade etmektedir. Eş. 3.31’de verilen $\hat{\mathbf{X}}(k+1|k+1)$ ifadesi, GKF algoritmasının kestirim değeri ve Eş. 3.32’de verilen $\mathbf{P}(k+1|k+1)$ ise güncellenmiş durum kovaryans değeridir. Eş. 3.26’da kullanılan, genişletilmiş Kalman filtresinin süreç gürültüsü kovaryansı $\mathbf{Q}$ ile ifade edilmekte olup, aşağıda verilen Eş. 3.33 numaralı eşitlikle hesaplanmıştır.

$$\mathbf{Q} = \begin{bmatrix} \frac{T^4}{4} & \frac{T^3}{2} & 0 & 0 \\ \frac{T^3}{2} & T^2 & 0 & 0 \\ 0 & 0 & \frac{T^4}{4} & \frac{T^3}{2} \\ 0 & 0 & \frac{T^3}{2} & T^2 \end{bmatrix} \sigma_v^2 \quad (3.33)$$

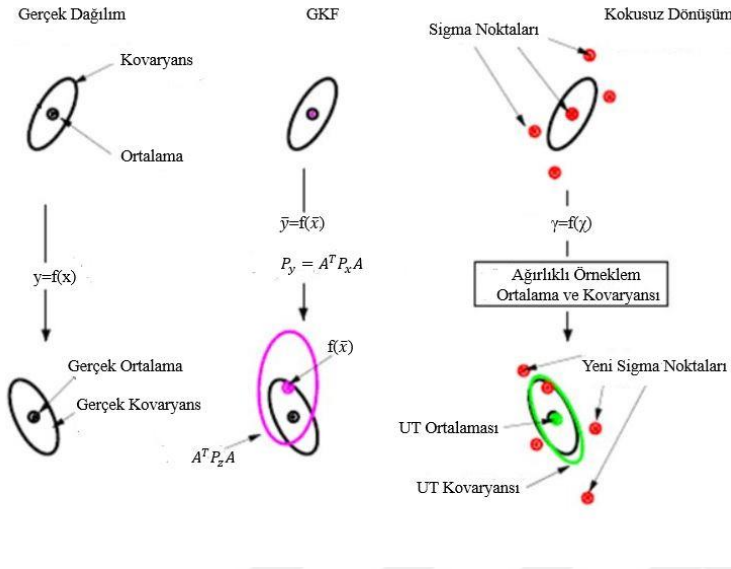
Eş. 3.33'te, σ_v^2 ifadesi genişletilmiş Kalman filtresinin süreç gürültüsü varyansı ve T ifadesi ise, örnekleme aralığını belirtmektedir. T ifadesi sabit bir değer olduğu için, filtrenin süreç gürültüsü kovaryansını değiştirmek için süreç gürültüsü varyansını değiştirmek gerekmektedir. Filtrenin süreç gürültüsü varyansı değiştirildiğinde, filtrenin süreç gürültüsü kovaryansı değişmektedir. Eş. 3.26 ile gösterilen durum öngörüm kovaryans eşitliği göz önüne alındığında, filtrenin süreç gürültüsü kovaryansı değiştiğinde, durum öngörüm kovaryansı değişmektedir. Durum öngörüm kovaryansı değiştiğinde ise Kalman kazancı değişmektedir. Eş. 3.31 incelendiğinde, Kalman kazancı değiştiğinde, GKF algoritması kestirim üretmek için, sensör ölçümünden gelen bilgiye verdiği ağırlık değiştiği için kestirim değeri ve buna bağlı olarak kestirim hatası değişmektedir. Bu nedenle, filtrenin süreç gürültüsü varyansı, genişletilmiş Kalman filtre algoritmasının tasarım parametresi olup, algoritmanın kestirim yeteneğini belirlemektedir. Genişletilmiş Kalman filtresinin süreç gürültüsü varyansının seçimi için, konumu tespit edilmek istenen hedef nesnenin hareketindeki maksimum ivme miktarı kullanılmaktadır. Maksimum ivme miktarı, Eş. 3.34 ile gösterilen eşitsizlikte kullanılarak, filtrenin süreç gürültüsünün standart sapma değeri seçilebilmektedir. Standart sapma değerinin karesi alınarak ise filtrenin süreç gürültüsü varyans değeri seçilebilmektedir (Bar-Shalom vd., 2001).

$$\frac{a_m}{2} \leq \sigma_v \leq a_m \quad (3.34)$$

Eş. 3.34'te σ_v değeri, genişletilmiş Kalman filtresinin süreç gürültüsünün standart sapmasını ve a_m ise hedef nesnenin hareketindeki maksimum ivme miktarını ifade etmektedir.

Bu tez çalışmasında, GKF algoritması, üzerinde UGB sensör etiketi bulunan hedef nesnenin hareketi esnasında bulunduğu konumların kestirimlerinin elde edilmesi için kullanılmıştır. İstatistiksel anlamda düşünüldüğünde, nesnenin konumu aslında rasgele bir değişkendir. Rasgele bir değişken ilk iki momentle ifade edilebilmektedir. O nedenle, rasgele bir değişken ortalama ve kovaryans ile ifade edilebilmekte ve GKF algoritması ile de o rasgele değişkene ait ortalama ve kovaryans değerlerinin kestirimleri elde edilmektedir. Rasgele değişkene ait başlangıç ortalama ve başlangıç kovaryans değerleri bilinmektedir. Rasgele değişkene ait başlangıç ortalama değer, GKF algoritmasında, başlangıç durum kestirim ve başlangıç kovaryans ise başlangıç durum kovaryans kestirim değeri olarak kullanılmaktadır. Bu değerler, GKF algoritmasındaki denklemlerde başlangıç değerler olarak kullanılmaktadır. GKF algoritmasındaki denklemlerin kullanılmasıyla, rasgele değişkene ait ortalama ve kovaryans değerleri zamanda ilerletilerek rasgele değişkene ait ortalama ve kovaryans

kestirimleri elde edilmektedir. Rasgele değişkene ait ortalama değerin kestirimi, durum kestirim değeri ve kovaryans değerinin kestirimi ise durum kovaryans kestirim değeridir. Böylece GKF algoritması kullanılarak, rasgele değişkene ait durum ve kovaryans değerlerinin kestirimleri elde edilmektedir.



Şekil 3.2. İki boyutlu Gauss dağılımına sahip rasgele değişkenin doğrusal olmayan dönüşümü sonucu ortalama ve kovaryans değerlerinin gerçek değerleri ile GKF algoritması kullanılarak ve kokusuz dönüşüm sonucu elde edilen değerleri (Lima, 2017).

Şekil 3.2’de, rasgele bir değişkene ait gerçek ortalama ve kovaryans dağılımları verilmiştir. Ayrıca, GKF algoritması kullanılarak gerçek ortalama ve kovaryans dağılımlarının kestirilmesiyle elde edilen dağılımlar ile kokusuz dönüşüm yöntemi kullanılarak elde edilen dağılımları gösterilmektedir. Kokusuz dönüşüm sonucu elde edilen, rasgele değişkene ait ortalama değeri ve kovaryans dağılımının, rasgele değişkene ait gerçek ortalama ve kovaryans değerleriyle daha iyi örtüştüğü görülmektedir. Ancak, GKF algoritmasının kestirimi sonucunda elde edilen, rasgele değişkene ait ortalama ve kovaryans dağılımlarının, bu rasgele değişkene ait başlangıçtaki gerçek kovaryans ve ortalama dağılımlarından farklı olduğu görülmektedir. Bu nedenle, Şekil 3.2’de gösterilen, rasgele değişkene ait gerçek kovaryans dağılımıyla, GKF algoritması kullanılarak kestirilen kovaryans dağılımının üst üste gelmediği, birbirleriyle örtüşmediği görülmektedir. Bunun sebebi, sensörün ölçüm modelinin veya durum denkleminin doğrusal olmamasıdır. Çünkü, GKF algoritmasında doğrusal olmayan ölçüm veya durum denkleminin doğrusallaştırılması için Taylor serisi açılımı kullanılmakta ve bu açılımın ilk iki terimi kullanılarak ta Jakobyen matris hesaplanmaktadır. Taylor serisi açılımı aslında sonsuz terime sahiptir. Ancak, Taylor serisi açılımında sadece ilk iki terim kullanılarak Jakobyen matris hesaplandığında, sonsuz terime

sahip açılımı, sadece iki terimle ifade etmekten kaynaklanan doğrusallaştırma hataları meydana gelmektedir (B. Huang ve Wang, 2006). Bu hatalardan dolayı, rasgele değişkene ait gerçek ortalama ve kovaryans dağılımları, GKF algoritması tarafından kestirilen, rasgele değişkene ait ortalama ve kovaryans dağılımlarıyla örtüşmediğinden, GKF algoritmasının kestirim hatası artmaktadır.

GKF algoritması, öngörüm ve güncelleme aşamalarından oluşmaktadır. Öngörüm aşamasında, başlangıç durum kovaryansı ve başlangıç durum kestirim değerlerinin, bir sonraki an için öngörümleri yapılmaktadır. Tez çalışması kapsamında, hedef nesne gerçekte 2 boyutlu kartezyen koordinat sistemi üzerinde koordineli dönüş hareketi yapmaktadır. GKF algoritmasının durum öngörümü aşamasında ise hedef nesnenin hareketi, yaklaşık sabit hızlı düzgün doğrusal hareket modeli kullanılarak öngörülmüştür. GKF algoritması düşünüldüğünde, hedef nesnenin gerçekte yaptığı hareket aslında, GKF algoritmasından bağımsız olarak gerçekleşmekte ve GKF algoritması açısından düşünüldüğünde bu olay dış dünyada gerçekleşmektedir. Bu nedenle, GKF algoritması, nesnenin gerçekte nasıl hareket ettiğini kesin olarak bilememektedir. Ancak bu hareketi, algoritmanın durum öngörümü aşamasında kendi modeliyle öngörebilmektedir. GKF algoritmasının öngörüm aşamasında, bir model geliştirilmekte ve bu modelle, konumu tespit edilmek istenen nesneye ait durum öngörümü yapılmaktadır. Bu nedenle, nesne gerçekte koordineli dönüş hareketi yapsa dahi, tasarlanan GKF algoritması ile nesnenin hareketi, yaklaşık sabit hızlı düzgün doğrusal hareket modeli ile de öngörülebilir. Ancak, gerçekte koordineli dönüş hareketi yapan nesnenin hareketi, durum öngörüm aşamasında yaklaşık sabit hızlı düzgün doğrusal hareket modeliyle öngörüldüğünden, kestirim hatasının düşük olması için filtrenin durum öngörüm belirsizliğinin yüksek olması gerekmektedir. Durum öngörüm belirsizliğinin yüksek olması için, tasarlanan GKF algoritmasında, durum öngörüm kovaryansının büyük olması gerekmektedir. Durum öngörüm kovaryansının büyük olması için de, filtrenin süreç gürültü varyansı büyük seçilmelidir. Filtrenin süreç gürültü varyansı arttırıldığında, durum öngörüm kovaryansı arttırılmış yani durum öngörümündeki belirsizlik arttırılmış olmaktadır. Böylece, GKF algoritması hedef nesnenin yapmış olduğu manevraları, dönüş hareketlerini takip edebilmiş olacaktır. Ancak, eğer nesne gerçekte, düzgün doğrusal hareket yapıyor ise ve GKF algoritmasıyla bu hareket düzgün doğrusal hareket modeliyle öngörüldüğünde, filtrenin süreç gürültü varyansının büyük alınması sebebiyle, manevralı hareketin olmadığı yerlerde, GKF algoritmasının kestirim hatası artacaktır.

GKF algoritmasının öngörüm aşamasında yapılan durum öngörümü ile nesnenin yaptığı hareket kesin olarak bilinemediği için durum öngörümü bir belirsizlik içermektedir. Bu belirsizlik, durum öngörüm kovaryansı olarak adlandırılmaktadır. GKF algoritması ile durum öngörümü haricinde, bir de ölçüm öngörümü yapılmaktadır. Ölçüm öngörümü, eğer nesne, GKF algoritmasının durum öngörümü sonucu elde edilen konumda bulunursa, sensörün, nesnenin öngörülen konumu ile UGB sensör okuyucusu arasındaki uzaklığı nasıl ölçeceğini belirtmektedir. Böylece, eğer nesne GKF algoritmasının öngörüm aşamasındaki öngörülen konumda bulunursa, o konumla sensör okuyucusu arasındaki uzaklığın, sensör tarafından nasıl ölçüleceğine dair ölçüm öngörümü yapılmaktadır. Ancak, GKF algoritması öngörüm sonucunda, hedef nesne üzerindeki sensörün gerçekte yaptığı ölçümü kesin olarak bilemediği için, ölçüm öngörümü sonucu bir belirsizlik meydana gelmektedir. Inovasyon, sensörün gerçekte yaptığı ölçüm ile ölçüm öngörümü arasındaki fark olduğu için, ölçüm öngörümündeki belirsizlikten dolayı, inovasyonda da bir belirsizlik olmaktadır. Bu belirsizlik ise, inovasyon kovaryansı olarak adlandırılmaktadır (B. Huang ve Wang, 2006). Kalman kazancı ise, durum öngörüm kovaryansı ile Jakobyian matrisin transpozunun çarpımının inovasyon kovaryansına oranıdır. Bu nedenle, Kalman kazancı, durum öngörüm kovaryansı ile inovasyon kovaryansı arasındaki oranla doğru orantılı olduğundan, inovasyon kovaryansı arttığında Kalman kazancı küçülmekte, durum öngörüm kovaryansı arttığında ise Kalman kazancı artmaktadır (B. Huang ve Wang, 2006). Kalman kazancı kullanılarak durum kestirimi elde edilmekte ve durum kovaryansı güncellenmektedir. Daha sonra, güncellenen durum kovaryansı ve durum kestirim değeri, bir sonraki ana ait durum öngörümü ve durum öngörüm kovaryansının hesaplanması için kullanılmakta ve GKF algoritması Şekil 3.1’de gösterildiği şekilde, bir sonraki an için tekrarlanmaktadır. GKF algoritması durum kestirim değerini üretmek için, Kalman kazancı ve inovasyonun çarpımı ile durum öngörümünü toplamaktadır. Bu nedenle, Kalman kazancı arttığında, kestirim değerini üretmek için sensör ölçümünden gelen bilgiye daha çok ağırlık vermektedir. Kalman kazancı küçük olduğunda ise, inovasyon kovaryansı, durum öngörüm kovaryansından daha büyük olmaktadır. Bu durumda, sensör ölçümünden gelen bilginin belirsizliği, durum öngörümü sonucu elde edilen bilginin belirsizliğinden daha büyük olmakta ve GKF algoritması, durum öngörümüne daha çok ağırlık vererek kestirim üretmektedir. Kalman kazancı, GKF algoritmasının durum öngörüm değerine mi yoksa sensör ölçümünden gelen bilgiye mi daha çok ağırlık vererek kestirim üretmesi gerektiğini belirlemektedir. Bu nedenle, Kalman kazancı, GKF algoritmasının kestirim yeteneğini belirlemektedir. Kalman kazancı, durum öngörüm kovaryansı ve inovasyon kovaryansına

bağlıdır. Kalman kazancını arttırmak veya azaltmak için durum öngörüm kovaryansı veya inovasyon kovaryansı değerlerini değiştirmek gerekmektedir. GKF algoritması tasarımı düşünüldüğünde, kullanılan sensör hazır alındığı için sensörün ölçüm gürültü kovaryansı belli ve sabit olduğu için onu değiştiremeyiz. Bu nedenle, Kalman kazancının değişmesi için, durum öngörüm kovaryansı değişmelidir. Durum öngörüm kovaryansı, durum geçiş matrisi ve filtrenin süreç gürültü kovaryansına bağlıdır. Durum geçiş matrisi sabit olduğu için, durum öngörüm kovaryansını değiştirmenin yolu, filtrenin süreç gürültüsü kovaryansını değiştirmektir. Filtrenin süreç gürültü kovaryansı ise genişletilmiş Kalman filtresinin süreç gürültü varyansına bağlı olarak değişmektedir. Bu nedenle, Kalman kazancını artırmak veya azaltmanın yolu, filtrenin süreç gürültüsü varyansını değiştirmektir. Bu sebeple, filtrenin süreç gürültüsü varyansı, GKF algoritmasının tasarım parametresidir.

GKF algoritmasının kestirim hatası, 100 Monte Carlo simülasyonu yapılarak elde edilen, kare ortalamanın karekökü türünden (İng. *Root Mean Square – RMS*) hesaplanmış ve Eş. 3.35'te verilmiştir.

$$KOK = \sqrt{\frac{\sum_{i=1}^N (\hat{X}_i(k+1|k+1) - X(k+1))^2}{N}} \quad (3.35)$$

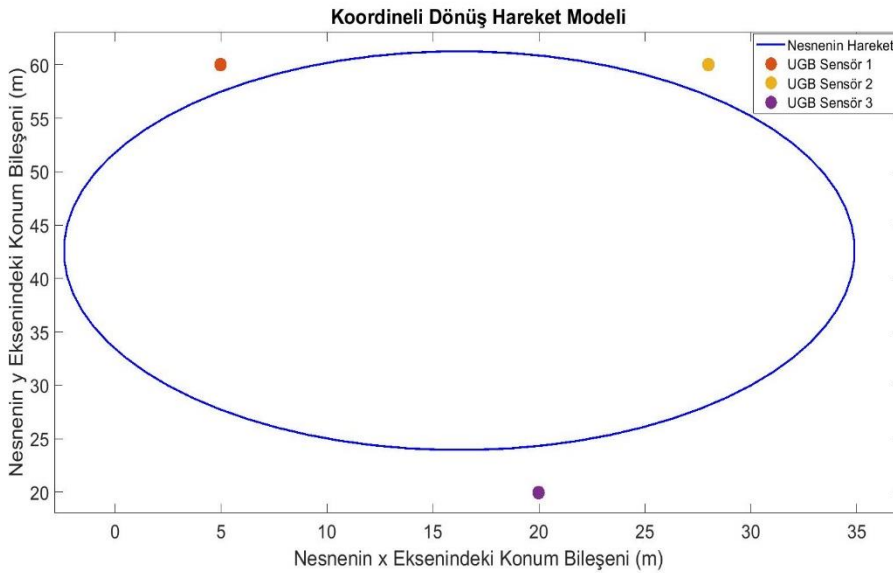
Eş. 3.35'te, N Monte Carlo simülasyonu sayısını, $\hat{X}_i(k+1|k+1)$ ifadesi, i . Monte Carlo simülasyonunda, $k+1$ anı için, GKF algoritmasının kestirim değerini ve $X(k+1)$ ise $k+1$ anında, nesnenin konum ve hızının gerçek değerlerinden oluşan durum değişken vektörünü belirtmektedir.

3.4. Farklı Sensör Geometrisi Senaryoları

Genişletilmiş Kalman filtresinin süreç gürültü varyansının ve sensörün ölçüm gürültüsü varyansının, GKF algoritmasının kestirim hatası üzerindeki etkilerinin incelenmesi için, Şekil 3.3'te gösterilen senaryo kullanılmıştır. Ayrıca, tez çalışması kapsamında sensör geometrisinin ve kullanılan sensör sayılarının, GKF algoritmasının kestirim performansı üzerindeki etkileri de incelenmiştir. Bu amaçla, farklı sensör geometrilerinden oluşan senaryolar tasarlanmıştır. Tasarlanan senaryolarda, nesne hareketine, 2 boyutlu kartezyen koordinat sistemi üzerinde (30, 30) metre konumundan başlamakta ve 360 saniye boyunca $0,0873 \text{ radyan} / \text{saniye}$ açısal hız ile koordineli dönüş hareketi yapmaktadır.

3.4.1. Geniřletilmiř Kalman filtresinin sre grlts varyansının ve sensrn lm grlt varyansının geniřletilmiř Kalman filtre algoritmasının kestirim hatası zerindeki etkilerinin incelenmesi iin kullanılan senaryo

lm grlt varyansının, GKF algoritmasının kestirim hatası zerindeki etkilerinin incelenmesi iin, geniřletilmiř Kalman filtresinin sre grlt varyansı sabit tutulup, lm grlt varyansı deėiřtirilerek, GKF algoritmasının konum kestirim hataları elde edilmiřtir. GKF algoritmasının sre grlt varyansı sırasıyla $1 \left(\frac{m}{sn^2}\right)^2$, $10 \left(\frac{m}{sn^2}\right)^2$ ve $50 \left(\frac{m}{sn^2}\right)^2$ olarak alındığında ve her bir sre grlt varyans deėeri iin sensrn lm grlt varyansı $0,5 m^2$, $1 m^2$, $5 m^2$ ve $15 m^2$ olduėu durumlar iin, konum kestirim hataları elde edilmiřtir. lm grlt varyansı sabit tutulup, geniřletilmiř Kalman filtresinin sre grlts varyansının deėiřtirilmesiyle, filtrenin sre grlts varyansının, GKF algoritmasının kestirim hatası zerindeki etkileri incelenmiřtir. Bu amala, lm grlts varyansı sabit tutularak, geniřletilmiř Kalman filtresinin sre grlt varyansı sırasıyla $1 \left(\frac{m}{sn^2}\right)^2$, $10 \left(\frac{m}{sn^2}\right)^2$ ve $50 \left(\frac{m}{sn^2}\right)^2$ alınarak konum kestirim hataları elde edilmiřtir. Gerekleřtirilen senaryoda, UGB sensrler sırasıyla, (5,60), (28,60) ve (20,20) metre konumlarında bulunmakta ve nesne ise hareketine (30, 30) metre konumundan bařlayıp, (1,1;1,1) m/sn sabit hız ve 0,0873 $radyan/sn$ aısal hız ile 360 saniye boyunca koordineli dnř hareketi yapmaktadır.



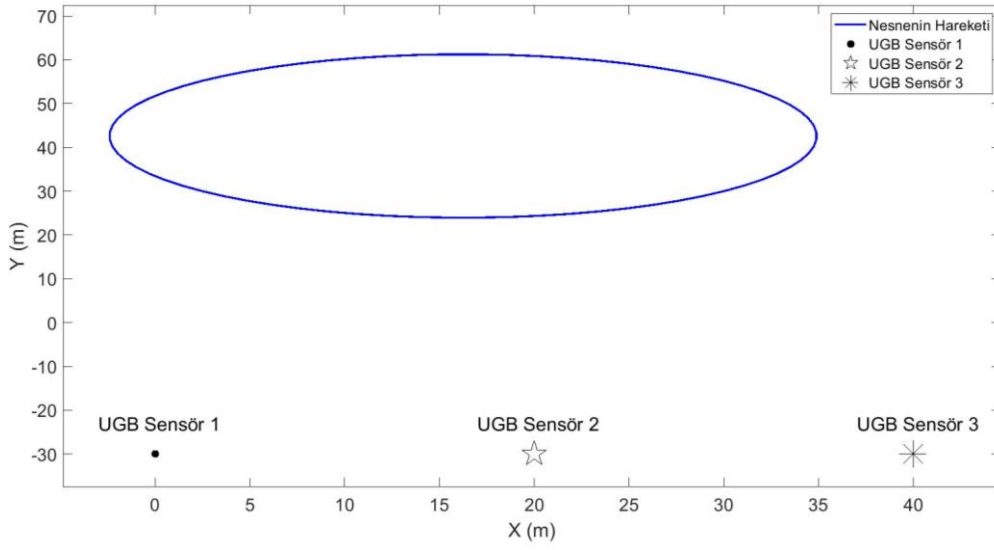
řekil 3.3. Hedef nesnenin hareket modeli ve UGB sensrlerinin konumları

3.4.2. Tasarlanan farklı sensör geometrileri

Tez çalışması kapsamında sensör geometrisinin GKF algoritmasının kestirim performansı üzerindeki etkilerini incelemek için farklı sensör geometrileri oluşturulmuştur. GKF algoritmasının ölçümle güncelleme aşamasında, sıralı güncelleme (*İng. Sequential Updating*) yöntemi kullanılmıştır. Ölçümler sıralı güncelleme yöntemi ile güncellenirken ve sensör geometrisinde 3 adet sensör kullanıldığı 1., 2., 3. ve 4. sensör geometrilerinin her birisi için, ilk olarak 1. sensör ölçümleri, daha sonra 2. sensör ölçümleri ve en son ise 3. sensör ölçümleri, GKF algoritmasının güncelleme aşamasında kullanılarak ölçümler güncellenmekte ve 3. sensör ölçümleri kullanılarak elde edilen kestirim değeri, algoritmanın kestirim değeri olmaktadır. Bu nedenle, 1., 2., 3. ve 4. sensör geometrilerinde, ölçümle güncelleme aşamasında, 3. sensör ölçümleri kullanılarak güncelleme yapıldığı durumda, Kalman kazancı ve kestirim değerinin, farklı sensör geometrilerine göre değişimi incelenmiştir. Gerçekleştirilen her bir farklı sensör geometrisi senaryolarında, konumu tespit edilmek istenen hedef nesne başlangıçta, kartezyen koordinat sistemi üzerinde (30, 30) metre konumunda bulunmakta ve (1,1;1,1) m/sn sabit hız ve 0,0873 $radyan/saniye$ açısal hız ile 360 saniye boyunca koordineli dönüş hareketi yapmaktadır. Hedef nesne, her bir farklı sensör geometrilerinden oluşan senaryoların her birisinde aynı hareketi yapmaktadır. Ancak, bazı şekillerde hedef nesnenin yaptığı hareketi temsil eden ve mavi daire ile gösterilen şekil, sensör konumlarından kaynaklanan grafiğin eksen farklarından dolayı farklı gözükmemektedir. Bunun sebebi, sensör konumlarından kaynaklanan grafiğin eksenlerinin farklı olmasıdır.

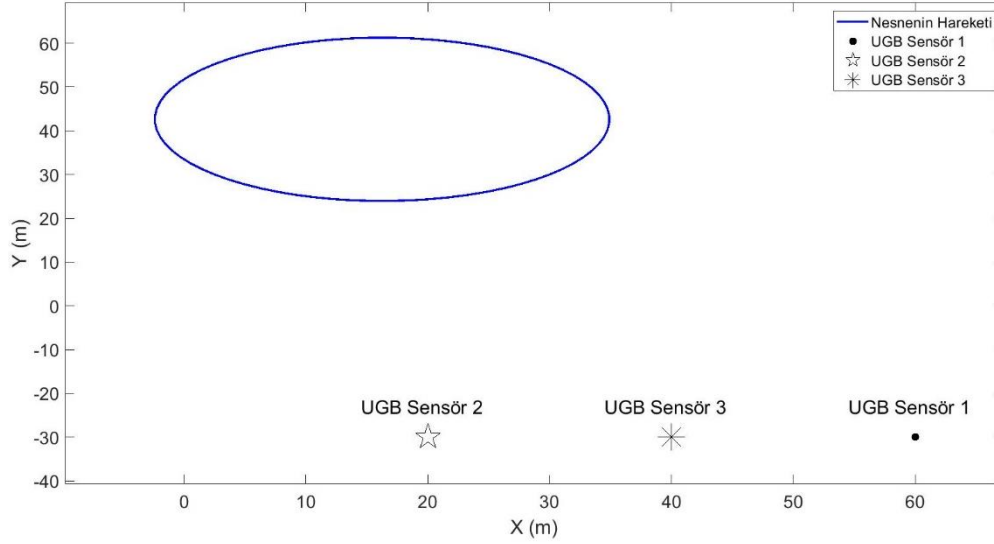
Birinci sensör geometrisi

Birinci sensör geometrisinde 3 adet UGB sensörü kullanılmıştır. 1. sensör (0, -30) metre, 2. sensör (20, -30) metre ve 3. sensör ise (40, -30) metre konumuna yerleştirilmiştir.



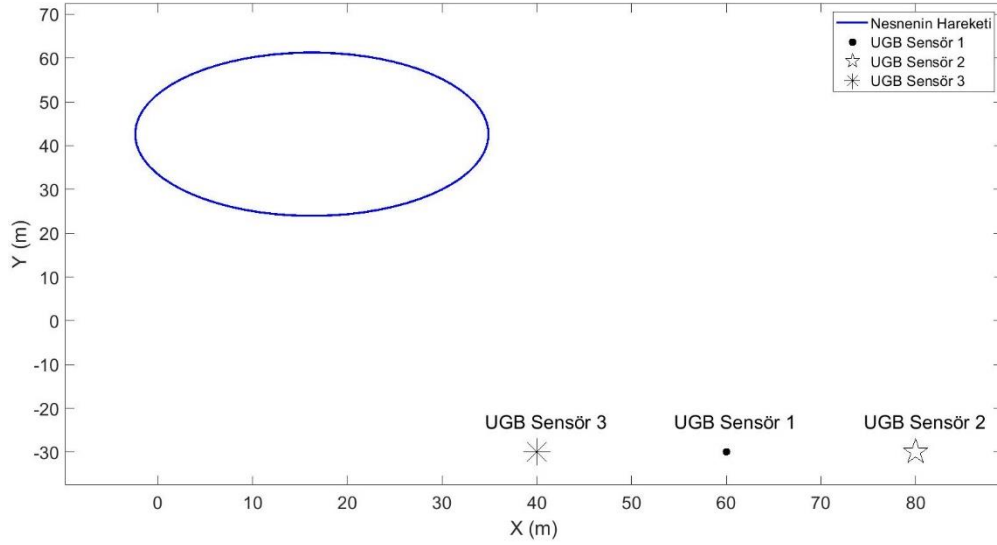
Şekil 3.4. Birinci sensör geometrisi

Sensör geometrilerinin, GKF algoritmasının kestirim hatası üzerindeki etkilerinin incelenebilmesi için, birinci sensör geometrisinde, 1. sensörün yeri değiştirilerek, 3. sensörün yanına, (60, -30) metre konumuna alınarak yeni sensör geometrisi elde edilmiştir. Elde edilen yeni sensör geometrisi ise Şekil 3.5'te gösterilmiştir.



Şekil 3.5. Birinci sensör geometrisinde, 1. sensörün yeri değiştirilerek elde edilen yeni sensör geometrisi

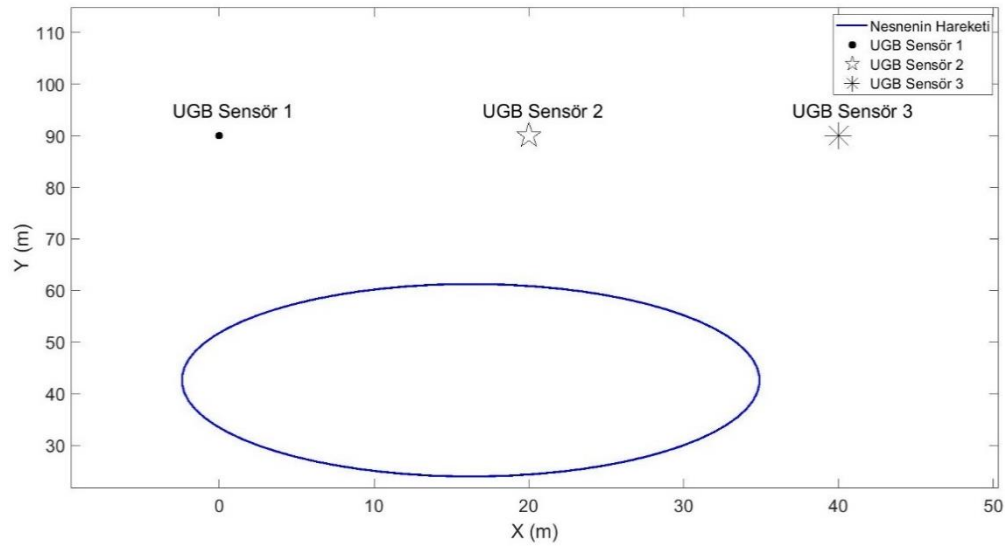
Şekil 3.5'te verilen geometride 2. sensör, 1. sensörün yanına, (80, -30) metre konumuna alındığında elde edilen yeni sensör geometrisi ise Şekil 3.6'da gösterilmiştir.



Şekil 3.6. Birinci sensör geometrisinde, 2. sensörün yeri değiştirilerek elde edilen yeni sensör geometrisi

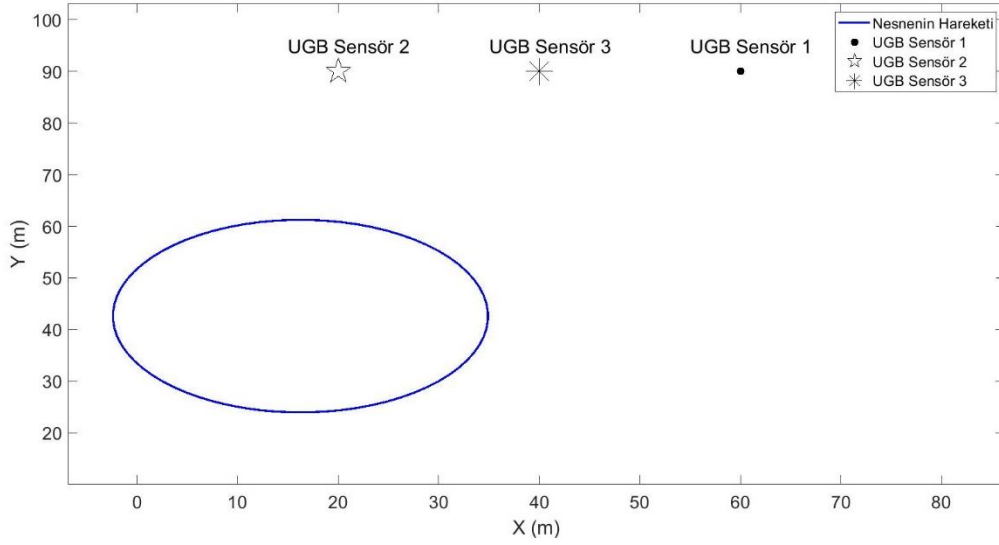
İkinci sensör geometrisi

Tez çalışması kapsamında kullanılan, ikinci sensör geometrisi Şekil 3.7’de gösterilmiştir. İkinci sensör geometrisinde 3 adet UGB sensör kullanılmıştır. 1. sensör (0, 90) metre, 2. sensör (20, 90) metre ve 3. sensör ise (40, 90) metre konumunda bulunmaktadır.



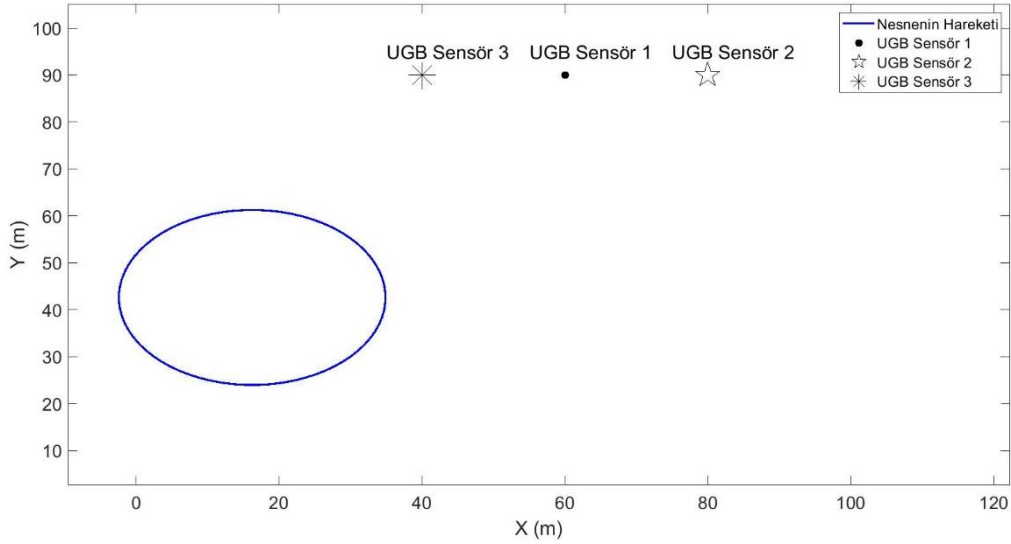
Şekil 3.7. İkinci sensör geometrisi

İkinci sensör geometrisinde, 1. sensörün yeri değiştirilerek, 3. sensörün yanına alındığında elde edilen yeni sensör geometrisi ise Şekil 3.8’de gösterilmiştir.



Şekil 3.8. İkinci sensör geometrisinde, 1. sensörün yeri değiştirilerek elde edilen yeni sensör geometrisi

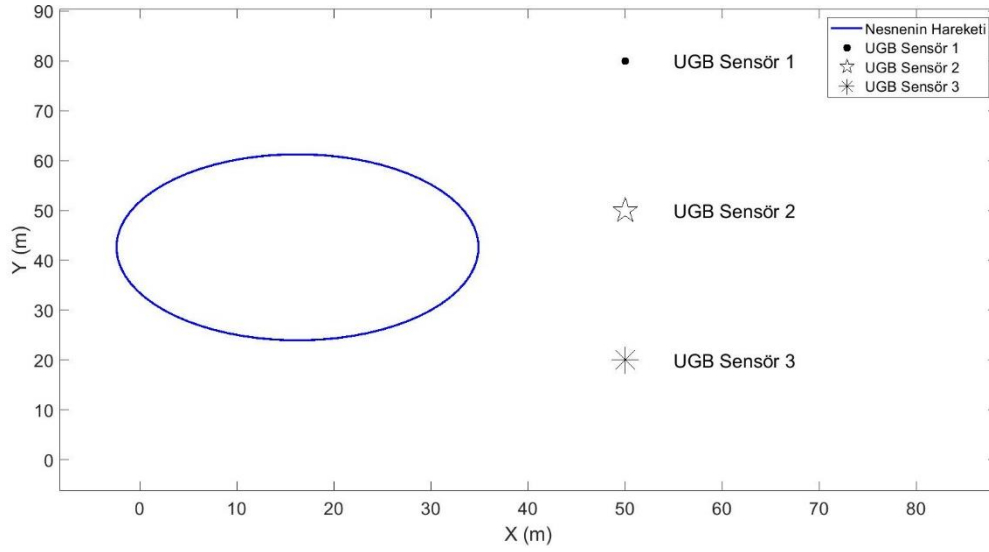
İkinci sensör geometrisinde, 1. sensörün yeri değiştirilerek, 3. sensörün yanına alınmıştır. 1. sensör (60, 90) metre konumuna alınmıştır. 2. sensörün Şekil 3.8’de gösterilen geometride, 1. sensörün yanına, (80, 90) metre konumuna alındığında elde edilen yeni sensör geometrisi ise Şekil 3.9’da gösterilmiştir.



Şekil 3.9. İkinci sensör geometrisinde, 2. sensörün yeri değiştirilerek elde edilen yeni sensör geometrisi

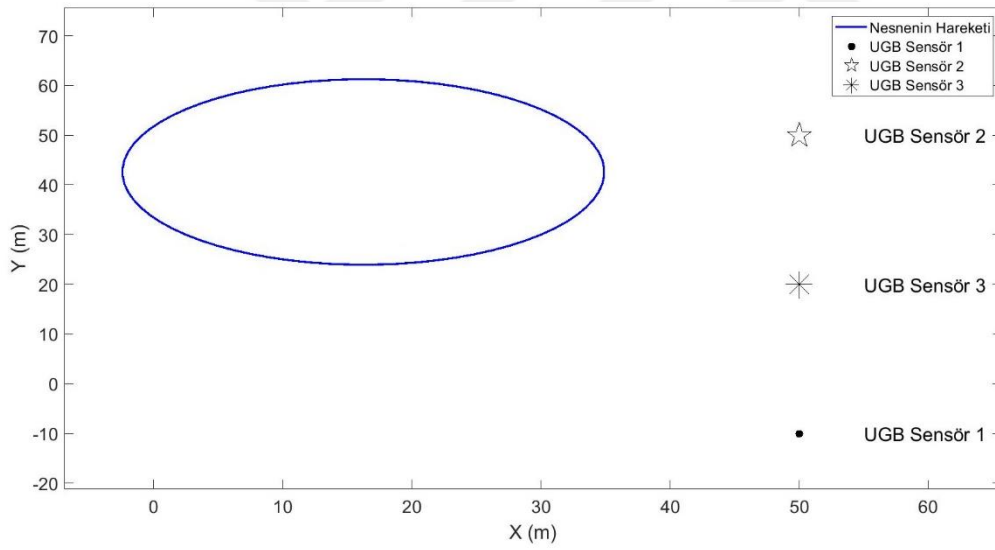
Üçüncü sensör geometrisi

Tez çalışmasında kullanılan üçüncü sensör geometrisi, Şekil 3.10’da gösterilmiştir.



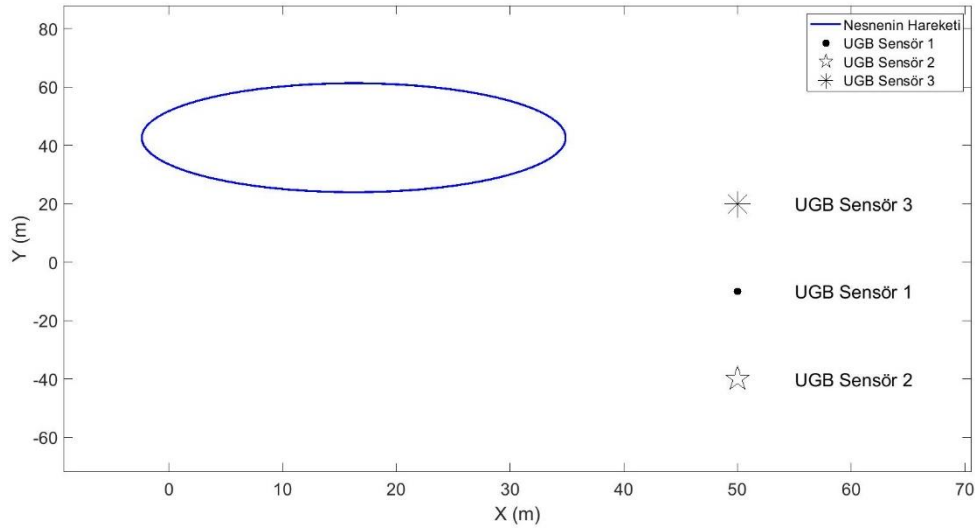
Şekil 3.10. Üçüncü sensör geometrisi

Üçüncü sensör geometrisinde, 1. sensör (50, 80) metre, 2. sensör (50, 50) metre ve 3. sensör ise (50, 20) metre konumlarına yerleştirilmiştir.



Şekil 3.11. Üçüncü sensör geometrisinde, 1. sensörün yeri değiştirilerek elde edilen yeni sensör geometrisi

Üçüncü sensör geometrisinde 1. sensör, 3. sensörün altına yerleştirildiğinde elde edilen yeni geometri Şekil 3.11’de gösterilmiştir. 1. sensör (50, -10) metre konumuna alınmıştır. 1. sensörün yerinin değiştirilmesinin, üçüncü sensör geometrisinde, GKF algoritmasının, hedef nesnenin konum kestirimi üzerindeki etkilerinin incelenmesi amaçlamıştır.

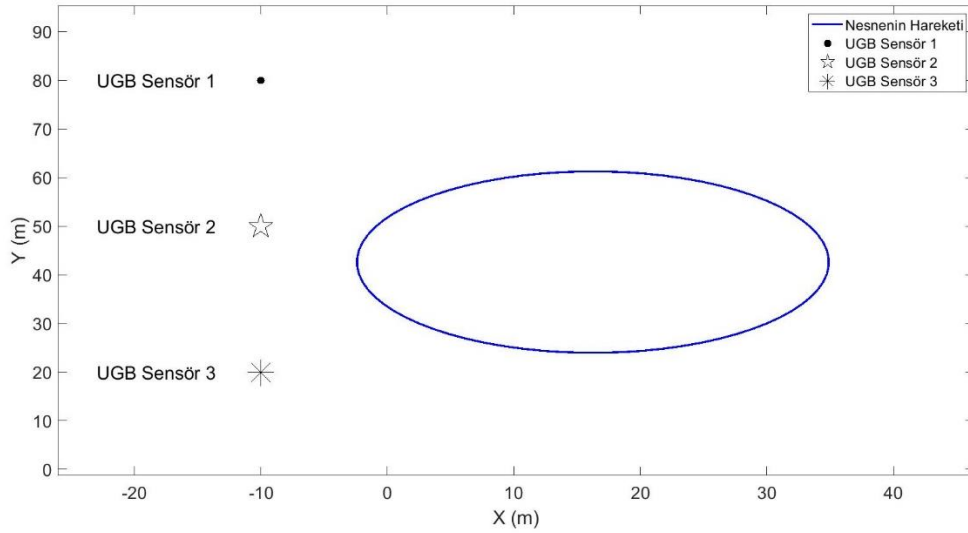


Şekil 3.12.Üçüncü sensör geometrisinde, 2. sensörün yeri değiştirilerek elde edilen yeni sensör geometrisi

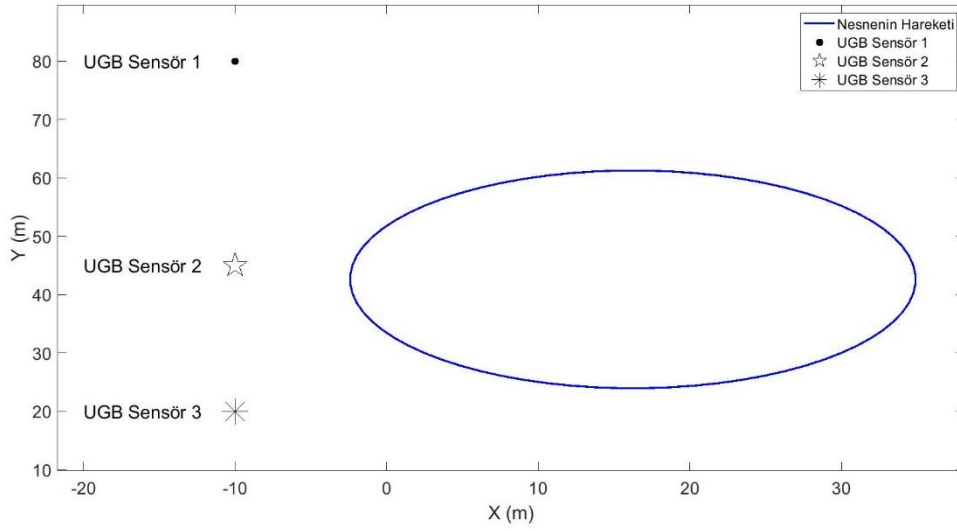
Üçüncü sensör geometrisinde, 2. sensör, Şekil 3.11’de gösterilen 1. sensörün altına yerleştirildiğinde elde edilen yeni geometri, Şekil 3.12’de gösterilmiştir. Şekil 3.12’de verilen sensör geometrisinde, 2. sensör (50, -40) metre konumuna alınmıştır.

Dördüncü sensör geometrisi

Dördüncü sensör geometrisinde 3 adet UGB sensör kullanılmıştır. 1. sensör (-10, 80) metre, 2. sensör (-10, 50) metre ve 3. sensör ise (-10, 20) metre konumlarına yerleştirilmiştir.

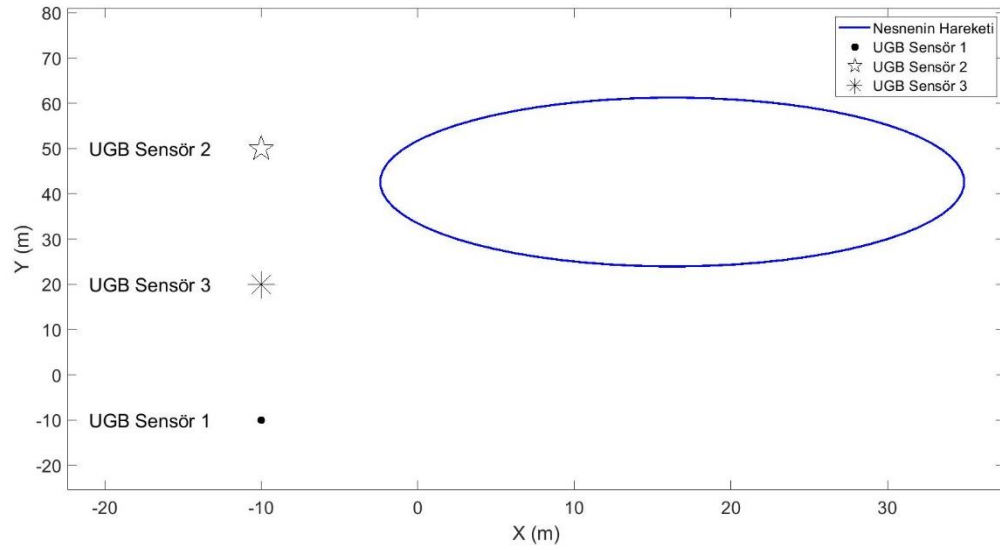


Şekil 3.13. Dördüncü sensör geometrisi



Şekil 3.14. Dördüncü sensör geometrisinde, 2. sensörün yeri değiştirilerek elde edilen yeni sensör geometrisi

Dördüncü sensör geometrisinde, 2. sensörün yeri değiştirilerek (-10, 45) metre konumuna alındığında elde edilen yeni geometri, Şekil 3.14'te gösterilmiştir.

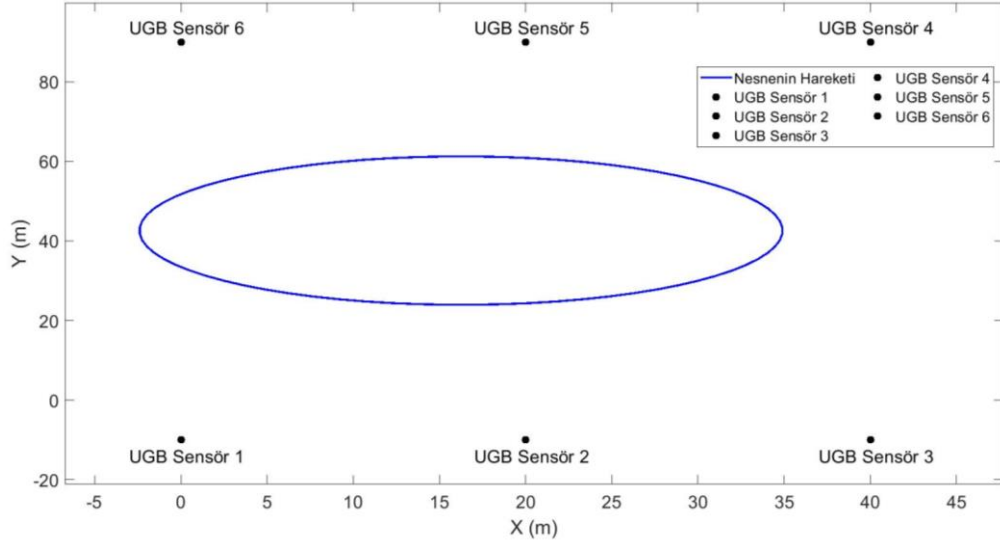


Şekil 3.15. Dördüncü sensör geometrisinde, 1. sensörün yeri değiştirilerek elde edilen yeni sensör geometrisi

Dördüncü sensör geometrisinde, 1. sensörün konumu değiştirilerek, (-10, -10) metre konumuna alındığında elde edilen yeni sensör geometrisi, Şekil 3.15'te gösterilmiştir.

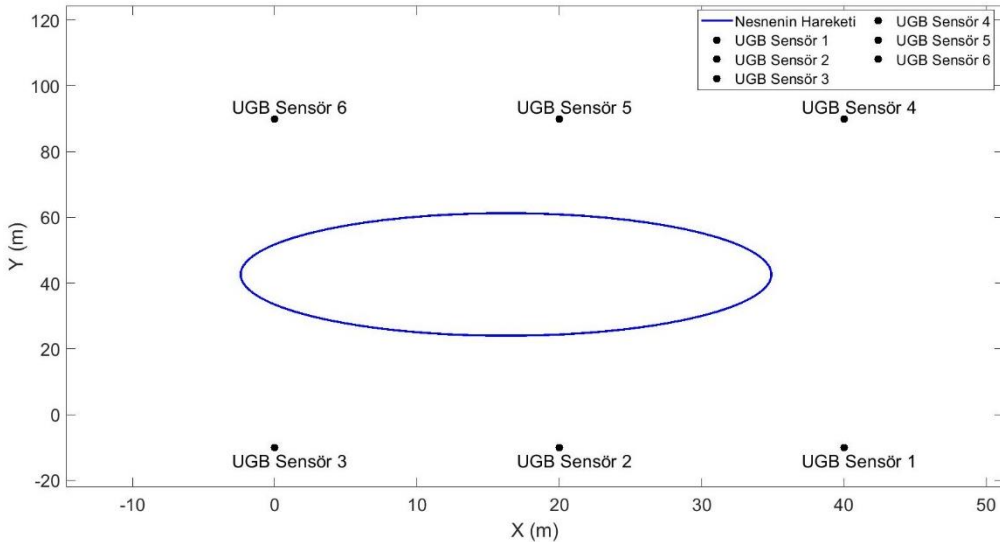
Beşinci sensör geometrisi

Beşinci sensör geometrisinde, 1. sensör (0, -10) metre, 2. sensör (20, -10) metre, 3. sensör (40, -10) metre, 4. sensör (40, 90) metre, 5. sensör (20, 90) metre ve 6. sensör ise (0, 90) metre konumuna yerleştirilmiştir. Beşinci sensör geometrisi, Şekil 3.16’da gösterilmiştir.



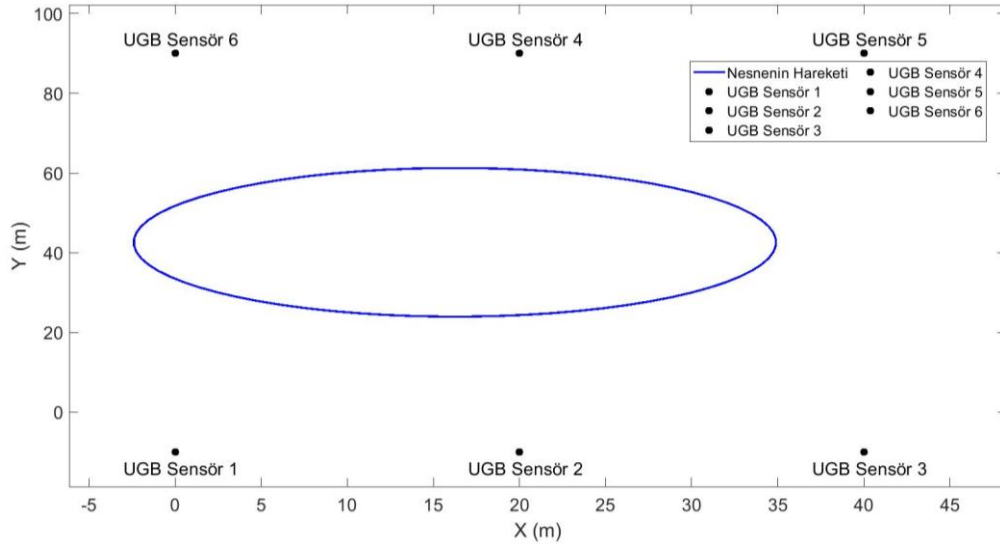
Şekil 3.16. Beşinci sensör geometrisi

Beşinci sensör geometrisinde 1. ve 3. sensörlerin yerleri değiştirilerek, 1. sensör (40, -10) metre ve 3. sensör ise (0, -10) metre konumuna alınmıştır. Bu durumda elde edilen yeni sensör geometrisi, Şekil 3.17’de gösterilmiştir.



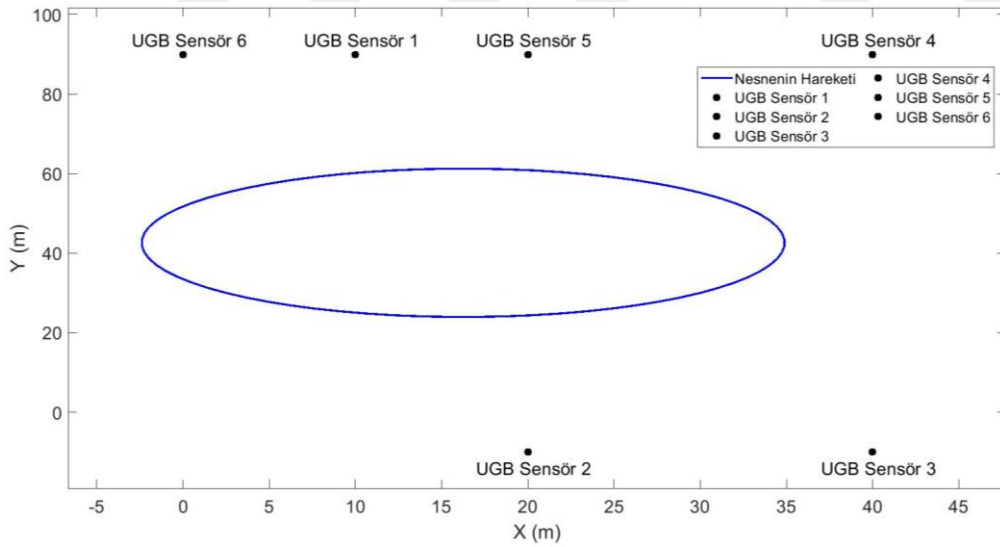
Şekil 3.17. Beşinci sensör geometrisinde 1. ve 3. sensörlerin yerleri değiştirilerek elde edilen yeni sensör geometrisi

Beşinci sensör geometrisinde, yeni sensör geometrisi elde etmek için, 4. ve 5. sensörler yer değiştirildiğinde elde edilen yeni geometri, Şekil 3.18’de gösterilmiştir.



Şekil 3.18. Beşinci sensör geometrisinde, 4. ve 5. sensörlerin yerleri değiştirilerek elde edilen yeni sensör geometrisi

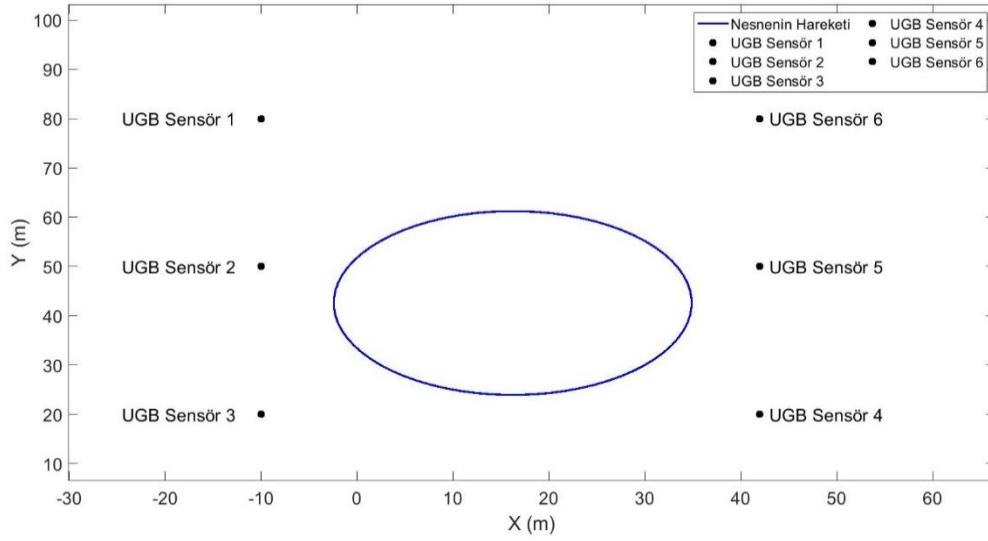
Beşinci sensör geometrisinde, 4. ve 5. sensörlerin yerleri değiştirildiğinde, 4. sensör (20, 90) metre, 5. sensör ise (40, 90) metre konumuna alınmıştır. Ayrıca, beşinci sensör geometrisinde 1. sensör, 5. ve 6. sensörlerin ortasına, (10, 90) metre konumuna alınarak elde edilen yeni sensör geometrisi, Şekil 3.19’da gösterilmiştir.



Şekil 3.19. Beşinci sensör geometrisinde, 1. sensör, 5. ve 6. sensörlerin ortasına taşındığında elde edilen yeni sensör geometrisi

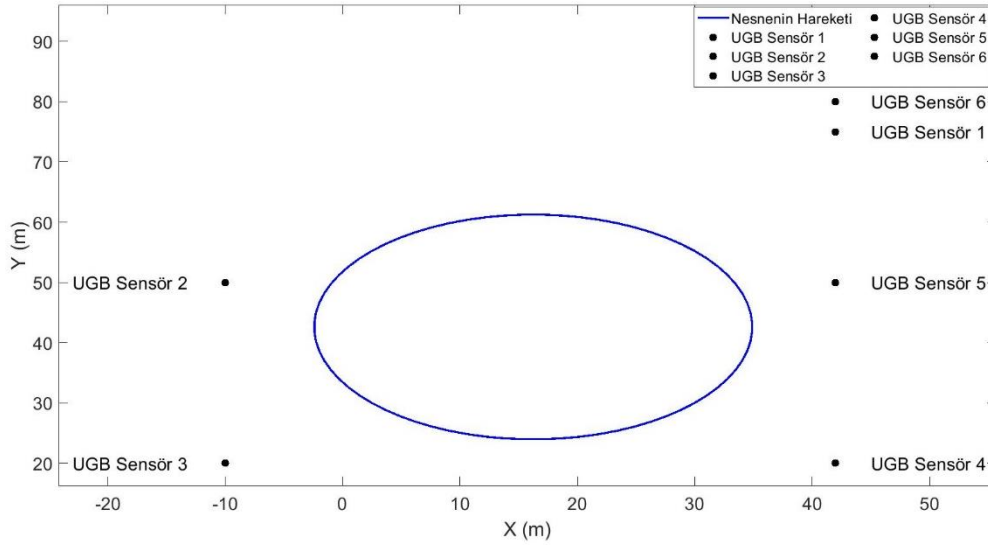
Altıncı sensör geometrisi

1. sensör (-10, 80) metre, 2. sensör (-10, 50) metre, 3. sensör (-10, 20) metre, 4. sensör (42, 20) metre, 5. sensör (42, 50) metre ve 6. sensör ise (42, 80) metre konumuna yerleştirilmiştir. Altıncı sensör geometrisi, Şekil 3.20’de gösterilmiştir.



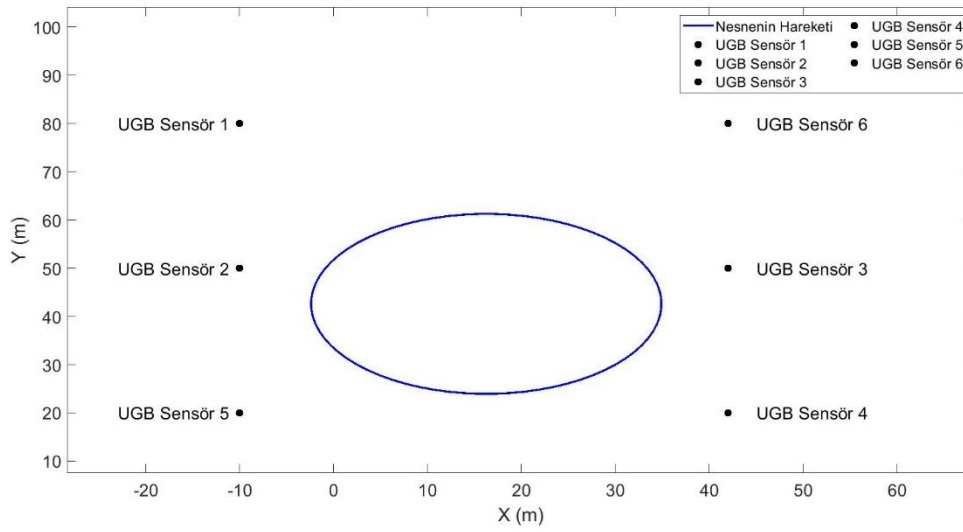
Şekil 3.20. Altıncı sensör geometrisi

Altıncı sensör geometrisinde, sensörlerin yerleri değiştirilerek farklı sensör geometrileri oluşturulmuştur. Altıncı sensör geometrisinde, 1. sensör, 6. sensöre yaklaştırılarak elde edilen yeni sensör geometrisi, Şekil 3.21’de gösterilmiştir.



Şekil 3.21. Altıncı sensör geometrisinde, 1. sensör, 6. sensöre yaklaştırılarak elde edilen yeni sensör geometrisi

Şekil 3.21’de gösterilen sensör geometrisinde, 1. sensör, 6. sensöre yaklaştırılarak (42, 75) metre konumuna yerleştirilmiş ve 1. sensörün bulunduğu konumun değişiminin, 6. sensör geometrisi için, GKF algoritmasının kestirim hatası üzerindeki etkileri incelenmiştir. Altıncı sensör geometrisinde, 3. ve 5. sensörlerin yerleri değiştirilerek elde edilen yeni sensör geometrisi ise Şekil 3.22’de gösterilmiştir.



Şekil 3.22. Altıncı sensör geometrisinde, 3. ve 5. sensörlerin yerleri değiştirilerek elde edilen yeni sensör geometrisi

3. ve 5. sensörlerin yerleri değiştirildiğinde, 5. sensör (-10, 20) metre ve 3. sensör (42, 50) metre konumuna alınmıştır.

3.5. Sensör Sayısının Genişletilmiş Kalman Filtre Algoritmasının Kestirim Hatası Üzerindeki Etkilerinin İncelenmesi İçin Kullanılan Senaryolar

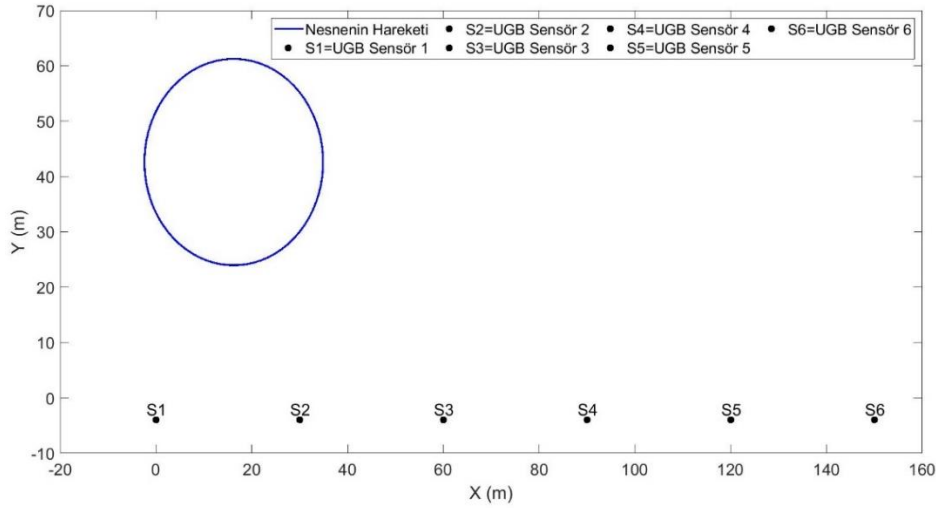
Tez çalışması kapsamında, sensör sayısının GKF algoritmasının kestirim hatası üzerindeki etkileri incelenmiştir. Bölüm 3.4.2.'de tanıtılan ve farklı sensör geometrilerinden oluşan her bir senaryoda kullanılan sensör sayıları 50'ye kadar arttırılarak, sensör sayılarının GKF algoritmasının kestirim hatasına olan etkileri incelenmiştir.

3.5.1. Birinci sensör geometrisinde kullanılan sensör sayılarının arttırılmasıyla tasarlanan sensör geometrileri

Birinci sensör geometrisinde kullanılan sensör sayılarının arttırılmasıyla elde edilen yeni sensör geometrileri, 3.5.1. numaralı bölümün alt bölümlerinde sırasıyla tanıtılmıştır.

6 adet UGB sensör kullanılarak tasarlanan 1. sensör geometrisi

6 adet sensör kullanılarak tasarlanan 1. sensör geometrisinde, 1. sensör (0, -4) metre, 2. sensör (30, -4) metre, 3. sensör (60, -4) metre, 4. sensör (90, -4) metre, 5. sensör (120, -4) metre ve 6. sensör ise (150, -4) metre konumunda bulunmaktadır. 6 adet UGB sensör kullanılarak elde edilen 1. sensör geometrisi, Şekil 3.23'te gösterilmiştir.

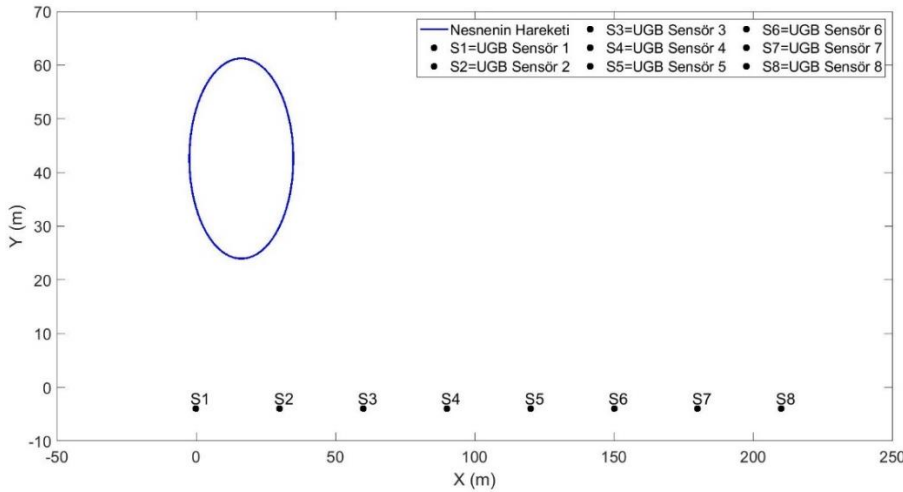


Şekil 3.23. 6 adet sensör kullanılarak tasarlanan 1. sensör geometrisi

8 adet UGB sensör kullanılarak tasarlanan 1. sensör geometrisi

8 adet UGB sensör kullanılarak elde edilen 1. sensör geometrisi, Şekil 3.24'te gösterilmiştir.

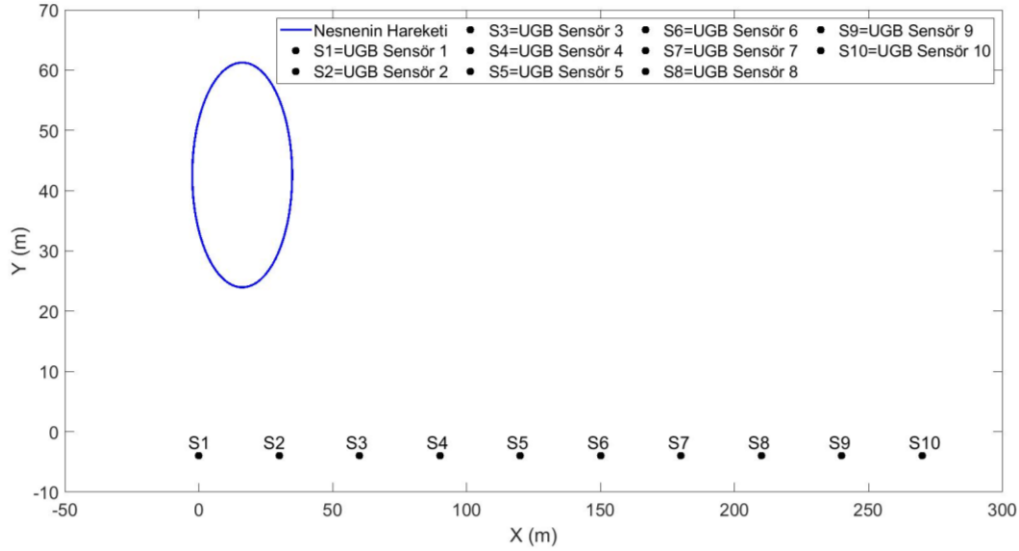
1. sensör (0, -4) metre, 2. sensör (30, -4) metre, 3. sensör (60, -4) metre, 4. sensör (90, -4) metre, 5. sensör (120, -4) metre, 6. sensör (150, -4) metre, 7. sensör (180, -4) metre ve 8. sensör ise (210, -4) metre konumunda bulunmaktadır.



Şekil 3.24. 8 adet sensör kullanılarak tasarlanan 1. sensör geometrisi

10 adet UGB sensör kullanılarak tasarlanan 1. sensör geometrisi

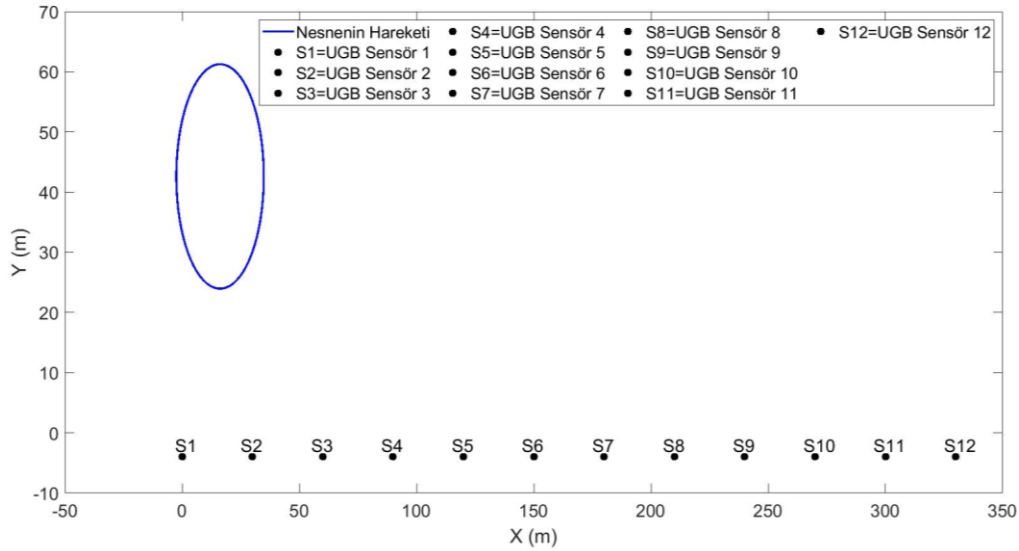
10 adet UGB sensör kullanılarak elde edilen 2. sensör geometrisinde, sensörler 1. sensörden, 10. sensöre doğru sırayla, (0, -4) metre konumundan, (270, -4) metre konumuna kadar 30 metre aralıklarla yerleştirilmiş olup, sensör geometrisi Şekil 3.25'te gösterilmiştir.



Şekil 3.25. 10 adet sensör kullanılarak tasarlanan 1. sensör geometrisi

12 adet UGB sensör kullanılarak tasarlanan 1. sensör geometrisi

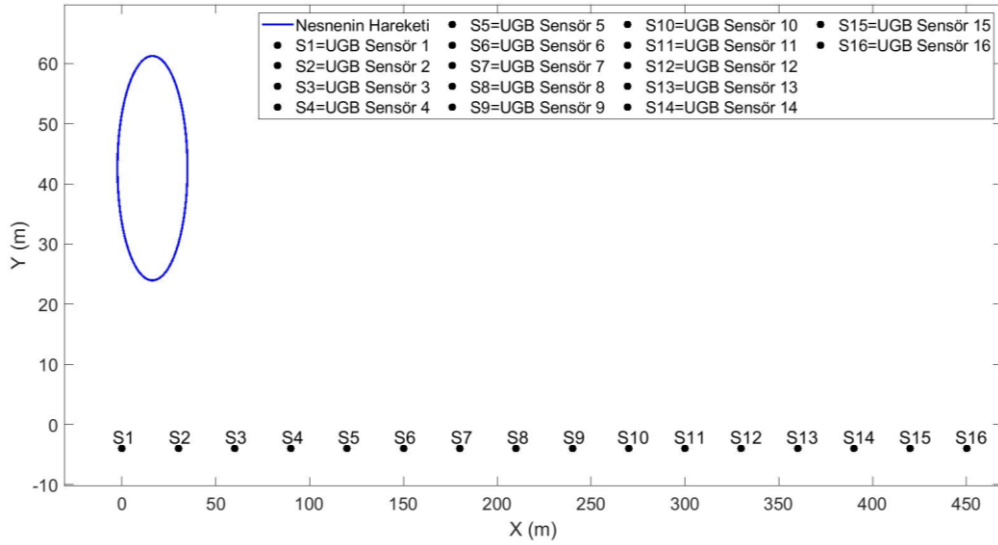
12 adet UGB sensör kullanılarak elde edilen 1. sensör geometrisinde, sensörler 1. sensörden, 12. sensöre doğru sırayla, (0, -4) metre konumundan, (330, -4) metre konumuna kadar 30 metre aralıklarla yerleştirilmiş olup, sensör geometrisi Şekil 3.26'da gösterilmiştir.



Şekil 3.26. 12 adet sensör kullanılarak tasarlanan 1. sensör geometrisi

16 adet UGB sensör kullanılarak tasarlanan 1. sensör geometrisi

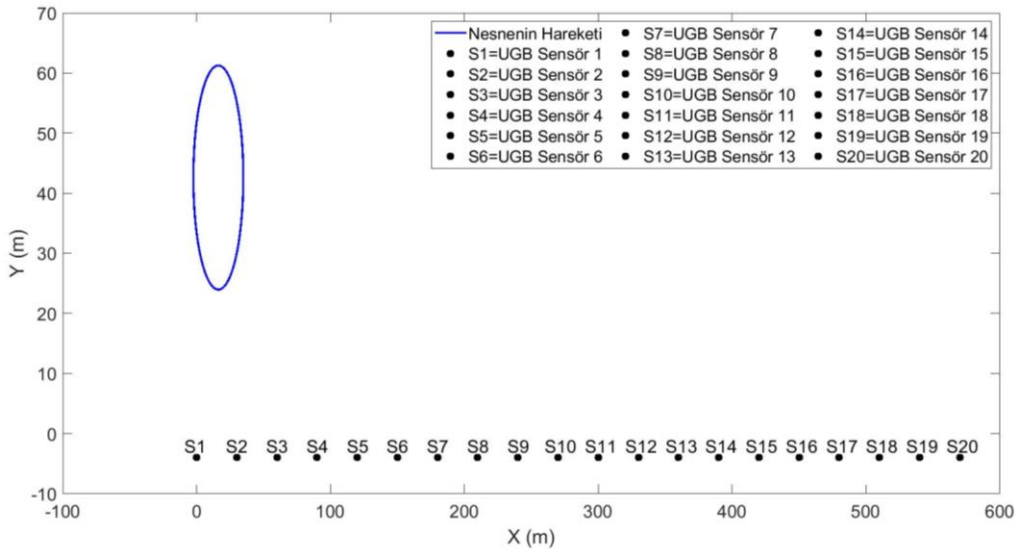
16 adet UGB sensör kullanılarak elde edilen 1. sensör geometrisinde, sensörler 1. sensörden, 16. sensöre doğru sırayla, (0, -4) metre konumundan, (450, -4) metre konumuna kadar 30 metre aralıklarla yerleştirilmiş olup, sensör geometrisi Şekil 3.27’de gösterilmiştir.



Şekil 3.27. 16 adet sensör kullanılarak tasarlanan 1. sensör geometrisi

20 adet UGB sensör kullanılarak tasarlanan 1. sensör geometrisi

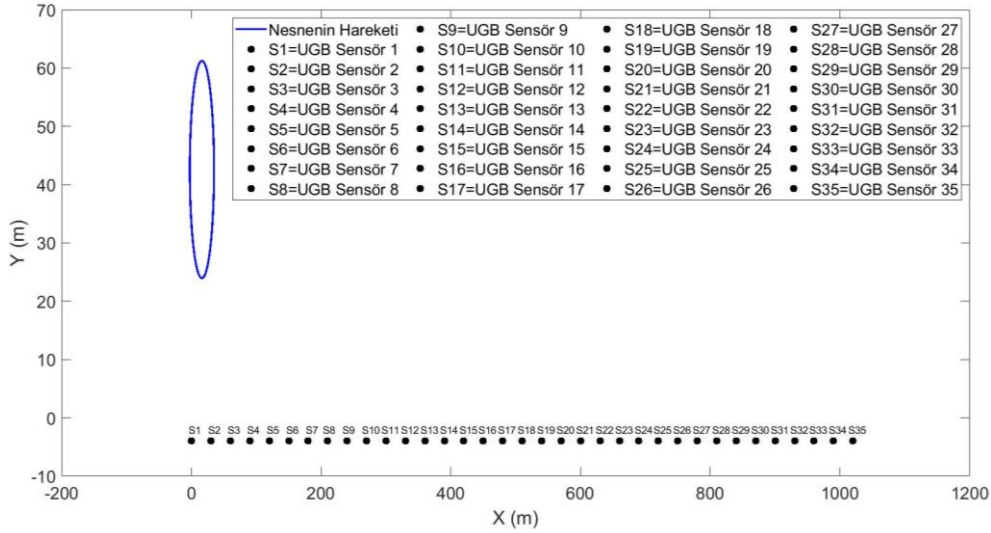
20 adet UGB sensör kullanılarak elde edilen 2. sensör geometrisinde, sensörler 1. sensörden, 20. sensöre doğru sırayla, (0, -4) metre konumundan, (570, -4) metre konumuna kadar, 30 metre aralıklarla yerleştirilmiş ve sensör geometrisi, Şekil 3.28’de gösterilmiştir.



Şekil 3.28. 20 adet sensör kullanılarak tasarlanan 1. sensör geometrisi

35 adet UGB sensör kullanılarak tasarlanan 1. sensör geometrisi

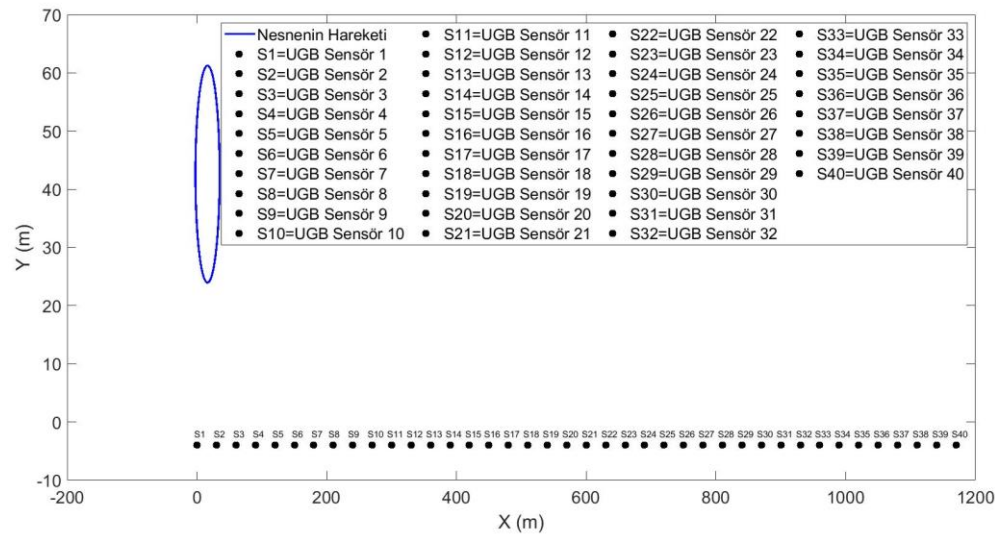
35 adet UGB sensör kullanılarak elde edilen 1. sensör geometrisinde, sensörler 1. sensörden, 35. sensöre doğru sırayla, (0, -4) metre konumundan, (1020, -4) metre konumuna kadar, 30 metre aralıklarla yerleştirilmiş olup, 2. sensör geometrisi Şekil 3.31’de gösterilmiştir.



Şekil 3.31. 35 adet sensör kullanılarak tasarlanan 1. sensör geometrisi

40 adet UGB sensör kullanılarak tasarlanan 1. sensör geometrisi

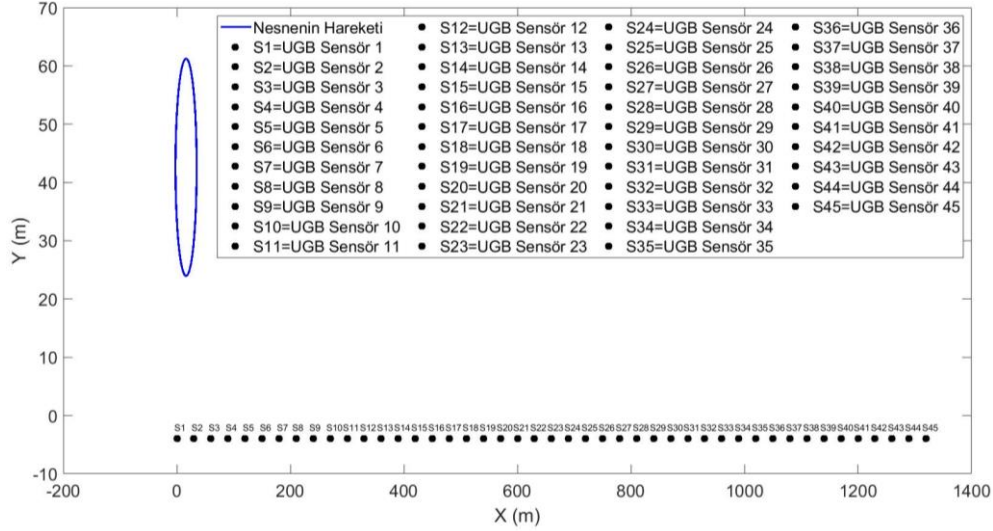
40 adet UGB sensör kullanılarak elde edilen 1. sensör geometrisinde, sensörler 1. sensörden, 40. sensöre doğru sırayla, (0, -4) metre konumundan, (1170, -4) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 40 adet UGB sensör kullanılarak elde edilen 1. sensör geometrisi, Şekil 3.32’de gösterilmiştir.



Şekil 3.32. 40 adet sensör kullanılarak tasarlanan 1. sensör geometrisi

45 adet UGB sensör kullanılarak tasarlanan 1. sensör geometrisi

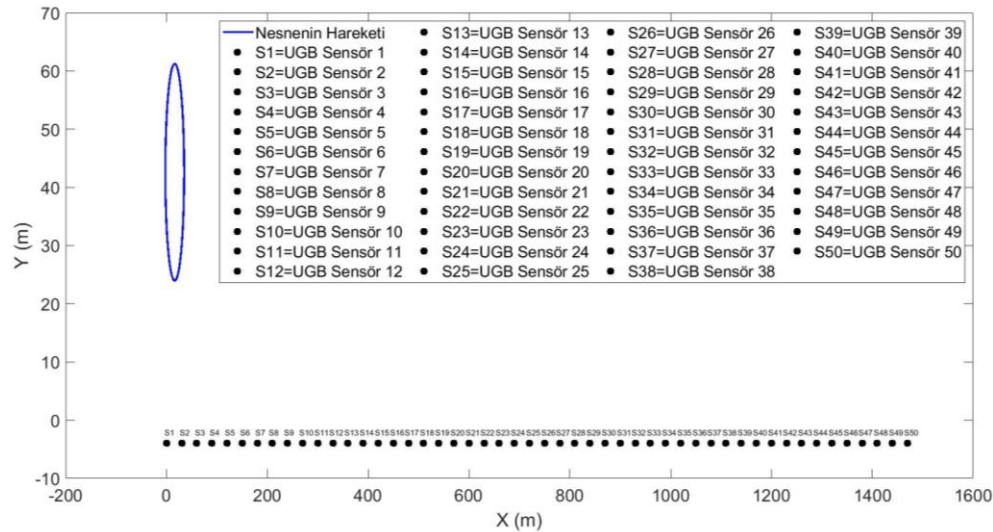
45 adet UGB sensör kullanılarak elde edilen 1. sensör geometrisinde, sensörler 1. sensörden, 45. sensöre doğru sırayla, (0, -4) metre konumundan, (1320, -4) metre konumuna kadar 30 metre aralıklarla yerleştirilmiş olup, 1. sensör geometrisi Şekil 3.33'te gösterilmiştir.



Şekil 3.33. 45 adet sensör kullanılarak tasarlanan 1. sensör geometrisi

50 adet UGB sensör kullanılarak tasarlanan 1. sensör geometrisi

50 adet UGB sensör kullanılarak elde edilen 1. sensör geometrisinde, sensörler 1. sensörden, 50. sensöre doğru sırayla, (0, -4) metre konumundan, (1470, -4) metre konumuna kadar 30 metre aralıklarla yerleştirilmiş olup, sensör geometrisi Şekil 3.34'te gösterilmiştir.



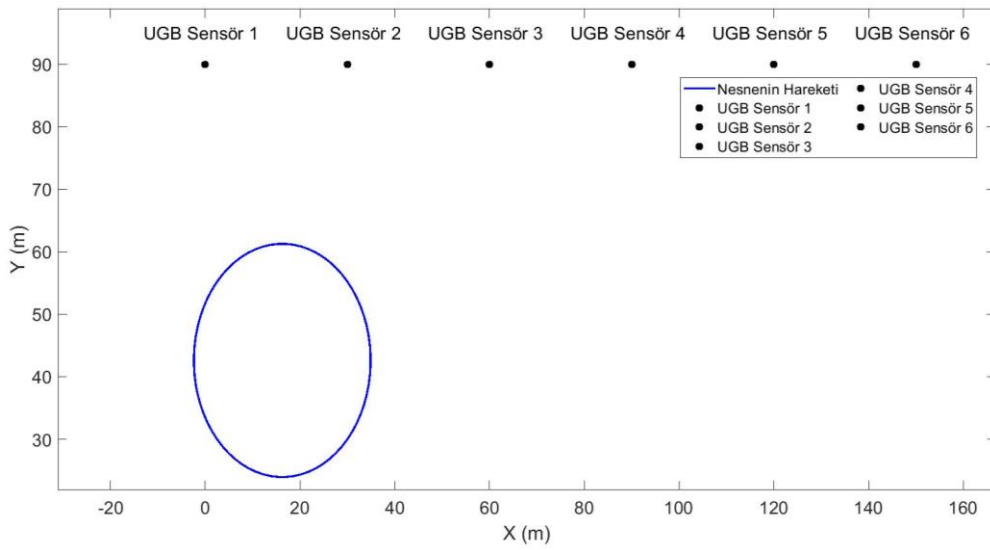
Şekil 3.34. 50 adet sensör kullanılarak tasarlanan 1. sensör geometrisi

3.5.2. İkinci sensör geometrisinde kullanılan sensör sayılarının arttırılmasıyla tasarlanan sensör geometrileri

İkinci sensör geometrisinde kullanılan sensör sayılarının arttırılmasıyla elde edilen yeni sensör geometrileri, 3.5.2. numaralı bölümün alt bölümlerinde sırasıyla tanıtılmıştır.

6 adet UGB sensör kullanılarak tasarlanan 2. sensör geometrisi

6 adet UGB sensör kullanılarak elde edilen 2. sensör geometrisinde, 1. sensör (0, 90) metre, 2. sensör (30, 90) metre, 3.sensör (60, 90) metre, 4. sensör (90, 90) metre, 5. sensör (120, 90) metre ve 6. sensör ise (150, 90) metre konumunda bulunmaktadır.

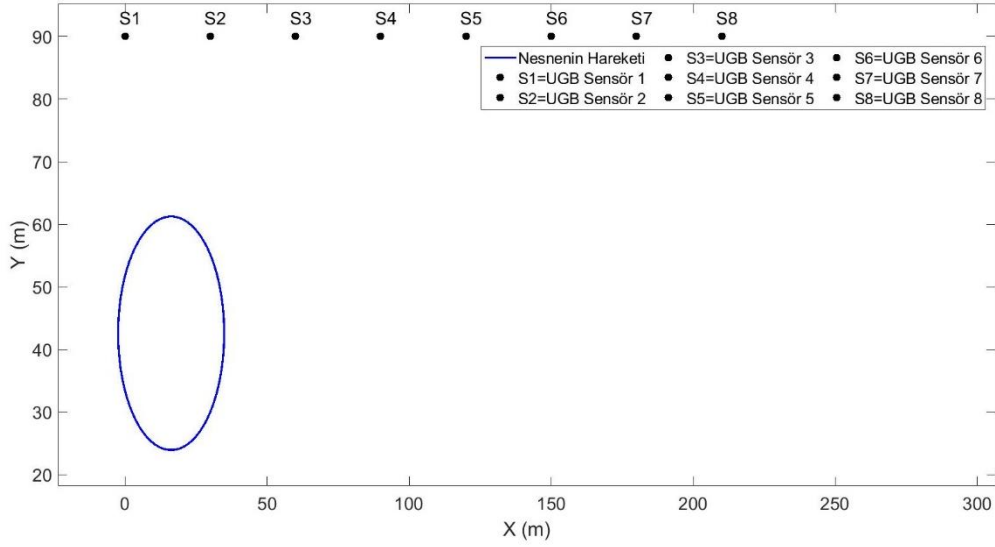


Şekil 3.35. 6 adet sensör kullanılarak tasarlanan 2. sensör geometrisi

8 adet UGB sensör kullanılarak tasarlanan 2. sensör geometrisi

8 adet UGB sensör kullanılarak elde edilen 2. sensör geometrisi, Şekil 3.36'da gösterilmiştir.

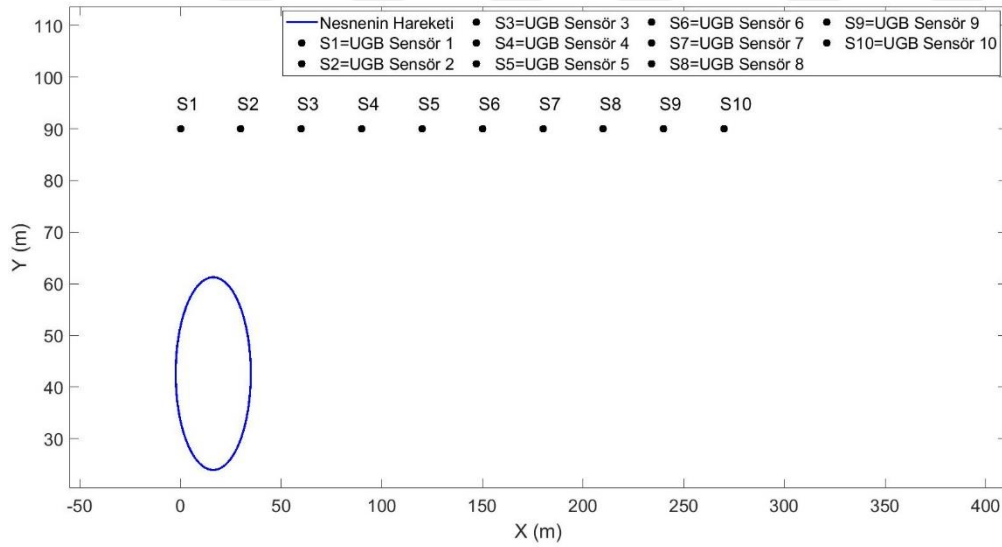
1. sensör (0, 90) metre, 2. sensör (30, 90) metre, 3.sensör (60, 90) metre, 4. sensör (90, 90) metre, 5. sensör (120, 90) metre, 6. sensör (150, 90) metre, 7.sensör (180, 90) metre ve 8. sensör ise (210, 90) metre konumunda bulunmaktadır.



Şekil 3.36. 8 adet sensör kullanılarak tasarlanan 2. sensör geometrisi

10 adet UGB sensör kullanılarak tasarlanan 2. sensör geometrisi

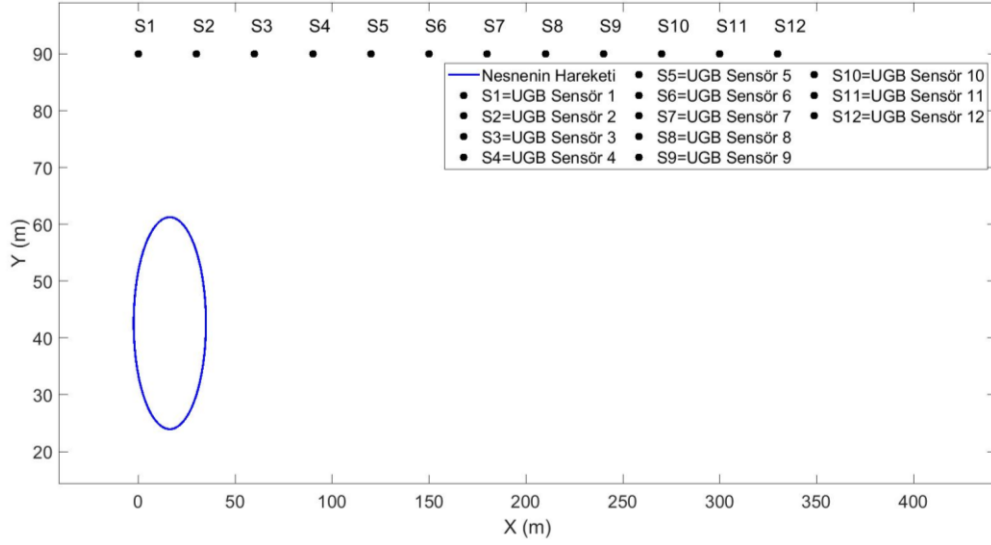
10 adet UGB sensör kullanılarak elde edilen 2. sensör geometrisinde, sensörler 1. sensörden, 10. sensöre doğru sırayla, (0, 90) metre konumundan, (270, 90) metre konumuna kadar 30 metre aralıklarla yerleştirilmiş olup, sensör geometrisi Şekil 3.37’de gösterilmiştir.



Şekil 3.37. 10 adet sensör kullanılarak tasarlanan 2. sensör geometrisi

12 adet UGB sensör kullanılarak tasarlanan 2. sensör geometrisi

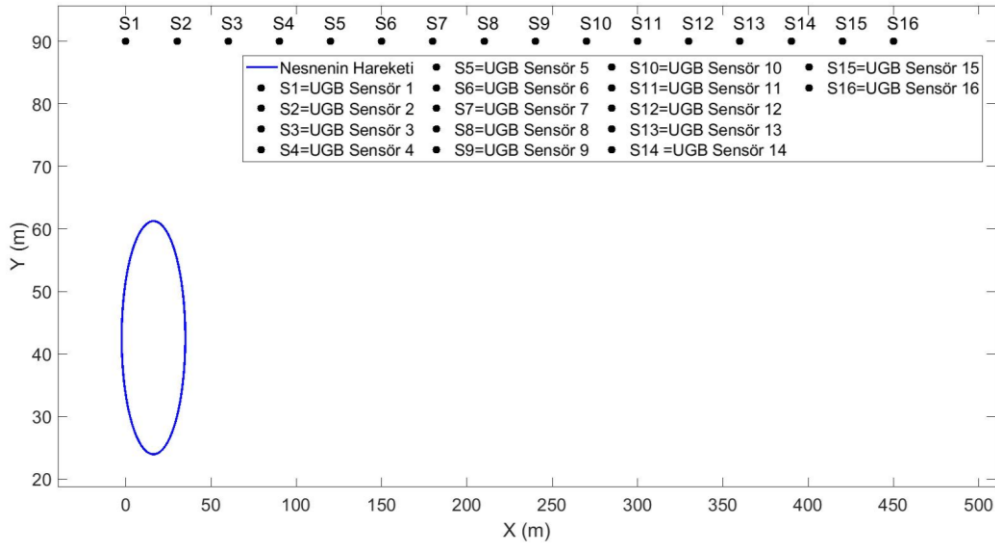
12 adet UGB sensör kullanılarak elde edilen 2. sensör geometrisinde, sensörler 1. sensörden, 12. sensöre doğru sırayla, (0, 90) metre konumundan, (330, 90) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir.



Şekil 3.38. 12 adet sensör kullanılarak tasarlanan 2. sensör geometrisi

16 adet UGB sensör kullanılarak tasarlanan 2. sensör geometrisi

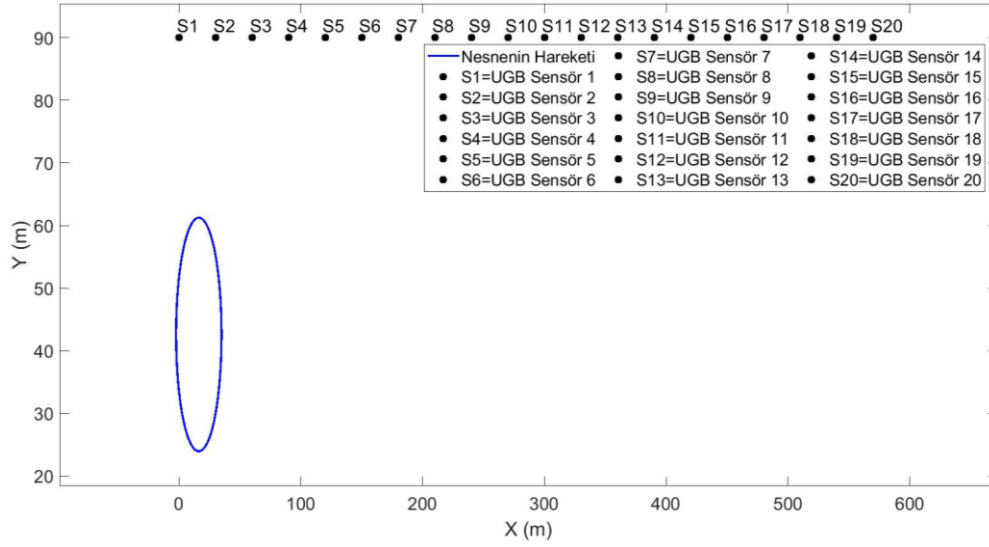
16 adet UGB sensör kullanılarak elde edilen 2. sensör geometrisinde, sensörler 1. sensörden, 16. sensöre doğru sırayla, (0, 90) metre konumundan, (450, 90) metre konumuna kadar 30 metre aralıklarla yerleştirilmiş olup, sensör geometrisi Şekil 3.39’da gösterilmiştir.



Şekil 3.39. 16 adet sensör kullanılarak tasarlanan 2. sensör geometrisi

20 adet UGB sensör kullanılarak tasarlanan 2. sensör geometrisi

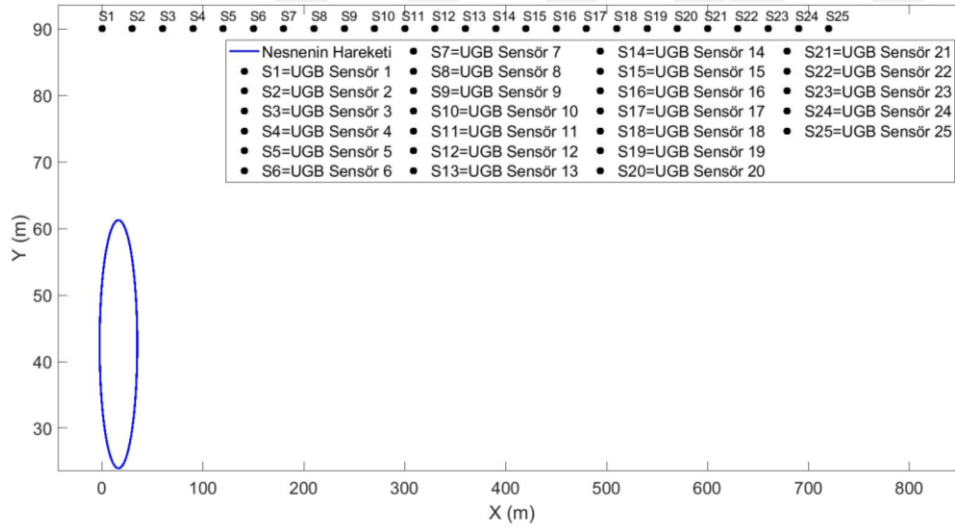
20 adet UGB sensör kullanılarak elde edilen 2. sensör geometrisinde, sensörler 1. sensörden, 20. sensöre doğru sırayla, (0, 90) metre konumundan, (570, 90) metre konumuna kadar 30 metre aralıklarla yerleştirilmiş olup, sensör geometrisi Şekil 3.40’ta gösterilmiştir.



Şekil 3.40. 20 adet sensör kullanılarak tasarlanan 2. sensör geometrisi

25 adet UGB sensör kullanılarak tasarlanan 2. sensör geometrisi

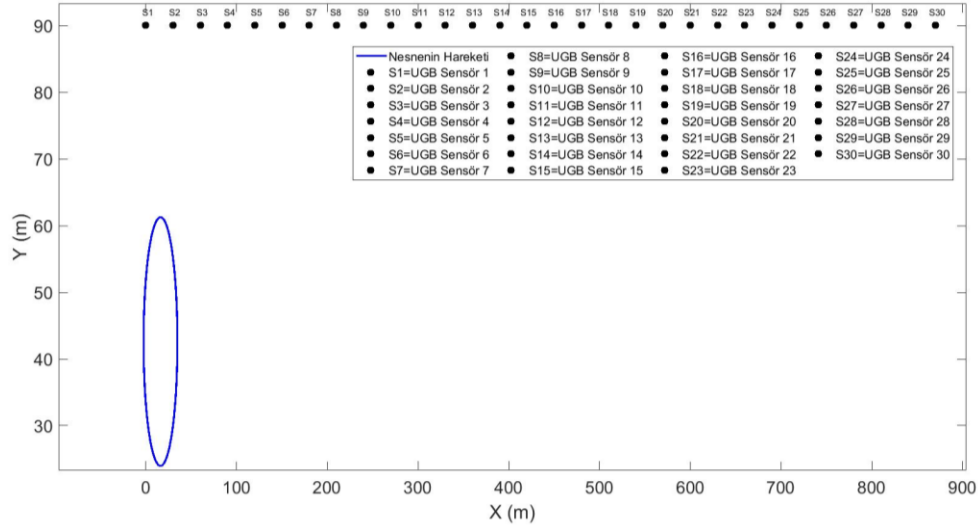
2. sensör geometrisinde, sensörler 1. sensörden, 25. sensöre doğru sırayla, (0, 90) metre konumundan, (720, 90) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir.



Şekil 3.41. 25 adet sensör kullanılarak tasarlanan 2. sensör geometrisi

30 adet UGB sensör kullanılarak tasarlanan 2. sensör geometrisi

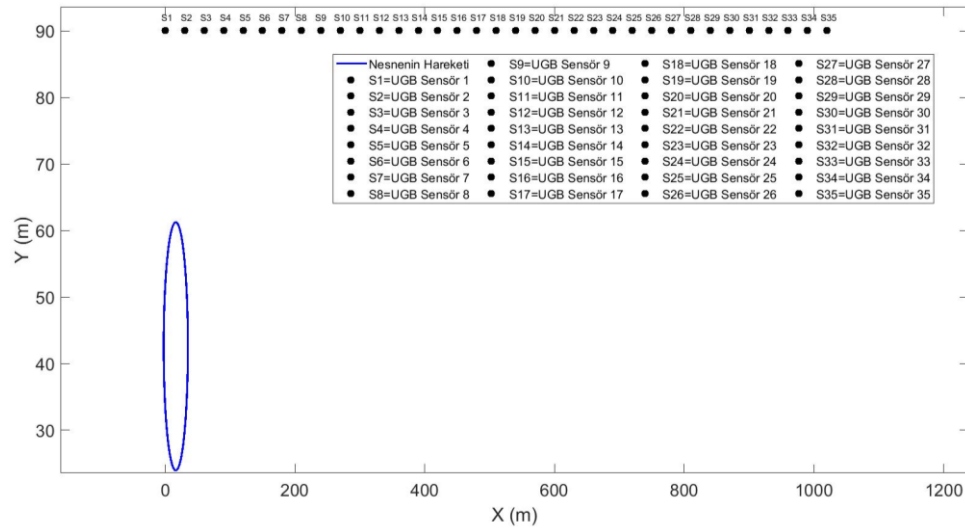
30 adet UGB sensör kullanılarak elde edilen 2. sensör geometrisinde, sensörler 1. sensörden, 30. sensöre doğru sırayla, (0, 90) metre konumundan, (870, 90) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 30 adet UGB sensör kullanılarak elde edilen 2. sensör geometrisi, Şekil 3.42’de gösterilmiştir.



Şekil 3.42. 30 adet sensör kullanılarak tasarlanan 2. sensör geometrisi

35 adet UGB sensör kullanılarak tasarlanan 2. sensör geometrisi

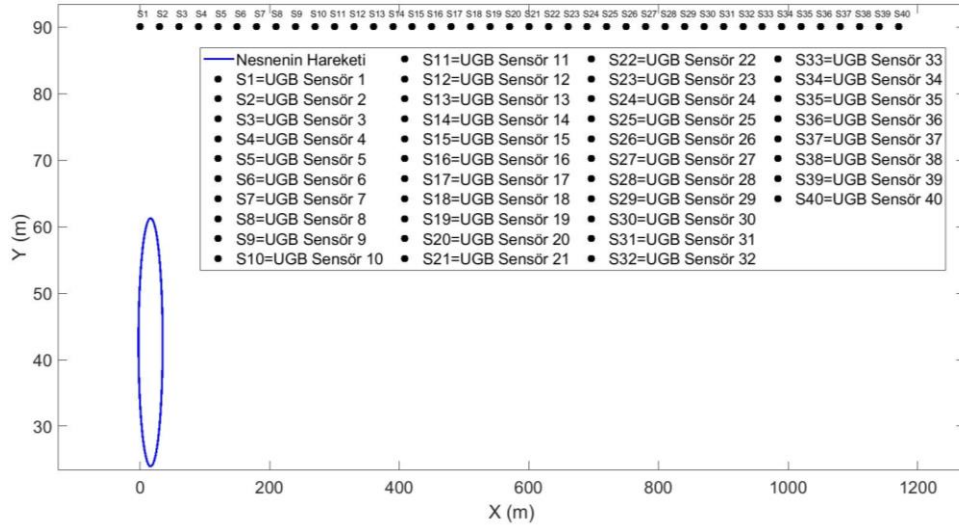
35 adet UGB sensör kullanılarak elde edilen 2. sensör geometrisinde, sensörler 1. sensörden, 35. sensöre doğru sırayla, (0, 90) metre konumundan, (1020, 90) metre konumuna kadar 30 metre aralıklarla yerleştirilmiş olup, 2. sensör geometrisi Şekil 3.43'te gösterilmiştir.



Şekil 3.43. 35 adet sensör kullanılarak tasarlanan 2. sensör geometrisi

40 adet UGB sensör kullanılarak tasarlanan 2. sensör geometrisi

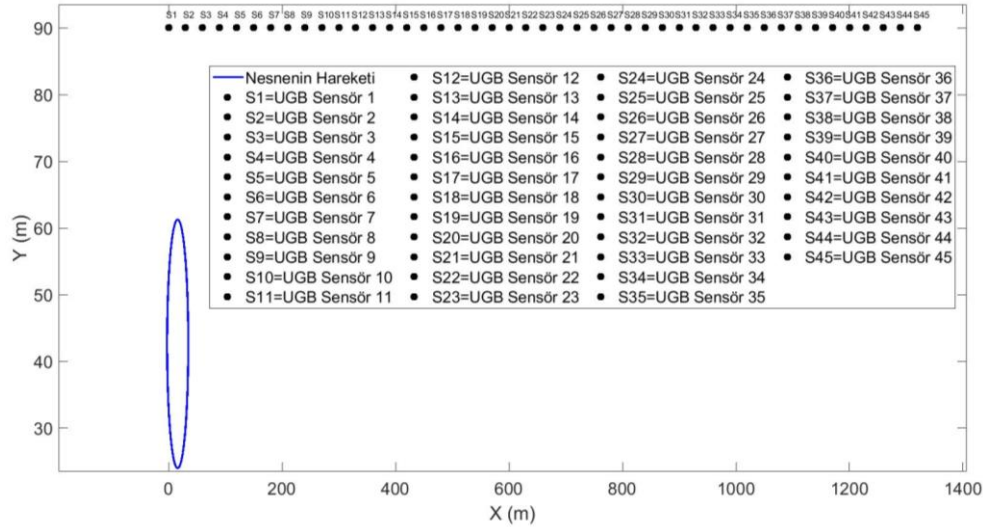
2. sensör geometrisinde, sensörler 1. sensörden, 40. sensöre doğru sırayla, (0, 90) metre konumundan, (1170, 90) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 40 adet UGB sensör kullanılarak elde edilen 2. sensör geometrisi, Şekil 3.44'te gösterilmiştir.



Şekil 3.44. 40 adet sensör kullanılarak tasarlanan 2. sensör geometrisi

45 adet UGB sensör kullanılarak tasarlanan 2. sensör geometrisi

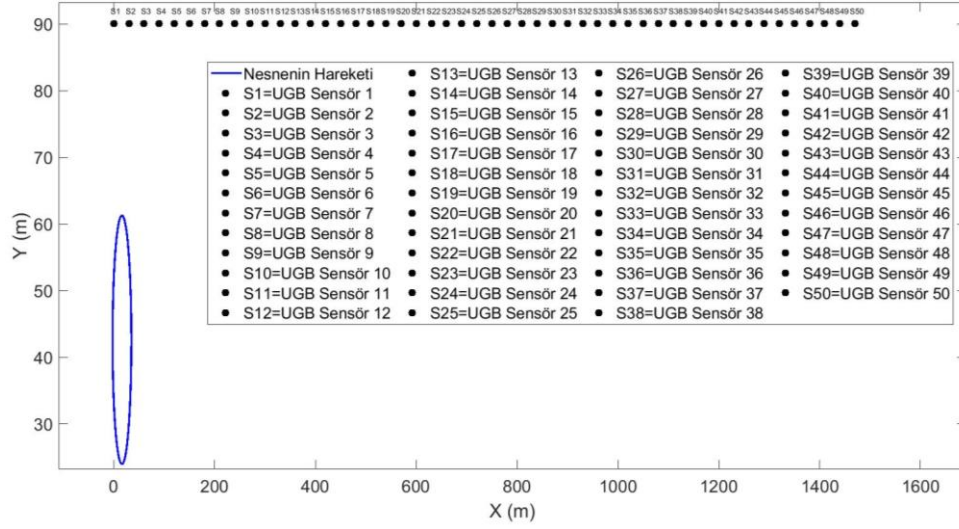
2. sensör geometrisinde, sensörler 1. sensörden, 45. sensöre doğru sırayla, (0, 90) metre konumundan, (1320, 90) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir.



Şekil 3.45. 45 adet sensör kullanılarak tasarlanan 2. sensör geometrisi

50 adet UGB sensör kullanılarak tasarlanan 2. sensör geometrisi

50 adet UGB sensör kullanılarak elde edilen 2. sensör geometrisinde, sensörler 1. sensörden, 50. sensöre doğru sırayla, (0, 90) metre konumundan, (1470, 90) metre konumuna kadar 30 metre aralıklarla yerleştirilmiş olup, Şekil 3.46'da gösterilmiştir.



Şekil 3.46. 50 adet sensör kullanılarak tasarlanan 2. sensör geometrisi

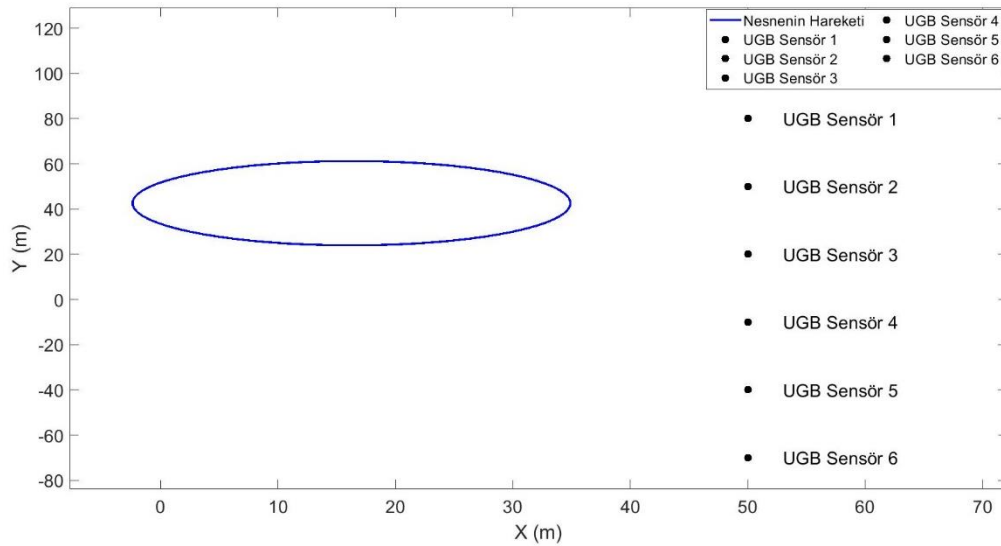
3.5.3. Üçüncü sensör geometrisinde kullanılan sensör sayılarının arttırılmasıyla tasarlanan sensör geometrileri

Üçüncü sensör geometrisinde kullanılan sensör sayılarının arttırılmasıyla elde edilen yeni sensör geometrileri 3.5.3. numaralı bölümün alt bölümlerinde sırasıyla tanıtılmıştır.

6 adet UGB sensör kullanılarak tasarlanan 3. sensör geometrisi

6 adet UGB sensör kullanılarak tasarlanan 3. sensör geometrisi Şekil 3.47’de gösterilmiştir.

1. sensör (50, 80) metre, 2. sensör (50, 50) metre, 3. sensör (50, 20) metre, 4. sensör (50, -10) metre, 5. sensör (50, -40) metre ve 6. sensör ise (50, -70) metre konumundadır.

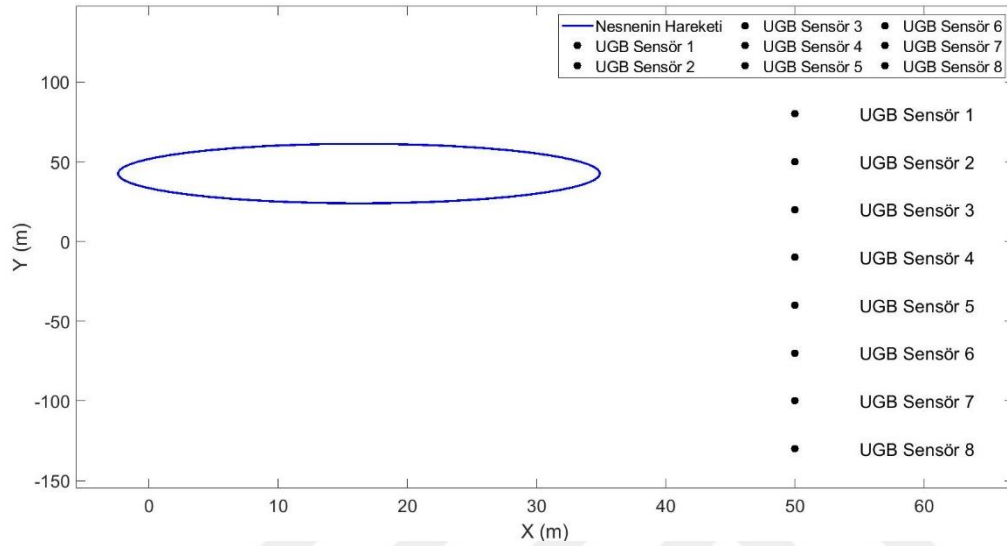


Şekil 3.47. 6 adet sensör kullanılarak tasarlanan 3. sensör geometrisi

8 adet UGB sensör kullanılarak tasarlanan 3. sensör geometrisi

8 adet UGB sensör kullanılarak elde edilen 3. sensör geometrisi, Şekil 3.48’de gösterilmiştir.

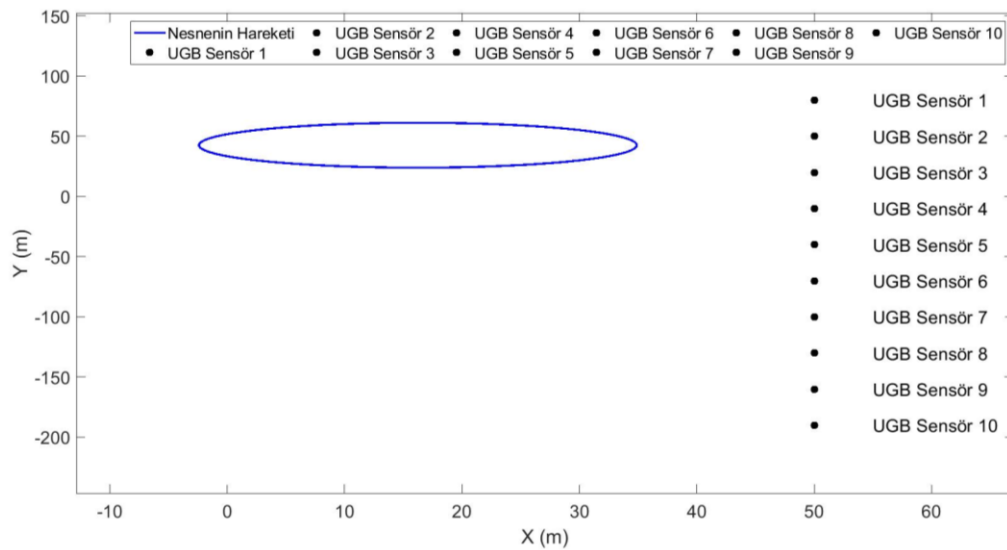
1. sensör (50, 80) metre, 2. sensör (50, 50) metre, 3. sensör (50, 20) metre, 4. sensör (50, -10) metre, 5. sensör (50, -40) metre, 6. sensör (50, -70) metre, 7. sensör (50, -100) metre ve 8. sensör ise (50, -130) metre konumlarında bulunmaktadır.



Şekil 3.48. 8 adet sensör kullanılarak tasarlanan 3. sensör geometrisi

10 adet UGB sensör kullanılarak tasarlanan 3. sensör geometrisi

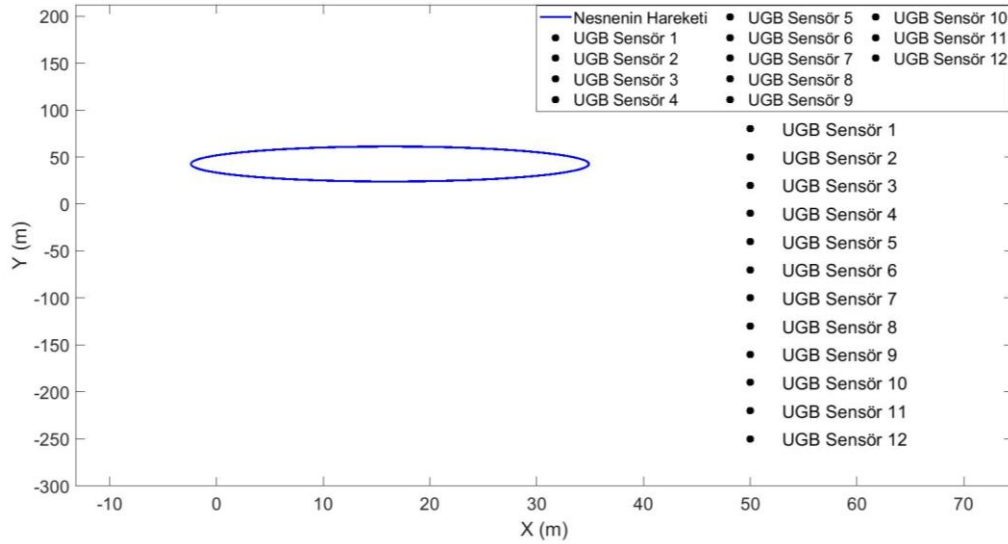
10 adet UGB sensör kullanılarak elde edilen 3. sensör geometrisinde, sensörler 1. sensörden, 10. sensöre doğru sırayla, (50, 80) metre konumundan, (50, -190) metre konumuna kadar 30 metre aralıklarla yerleştirilmiş olup, Şekil 3.49’da gösterilmiştir.



Şekil 3.49. 10 adet sensör kullanılarak tasarlanan 3. sensör geometrisi

12 adet UGB sensör kullanılarak tasarlanan 3. sensör geometrisi

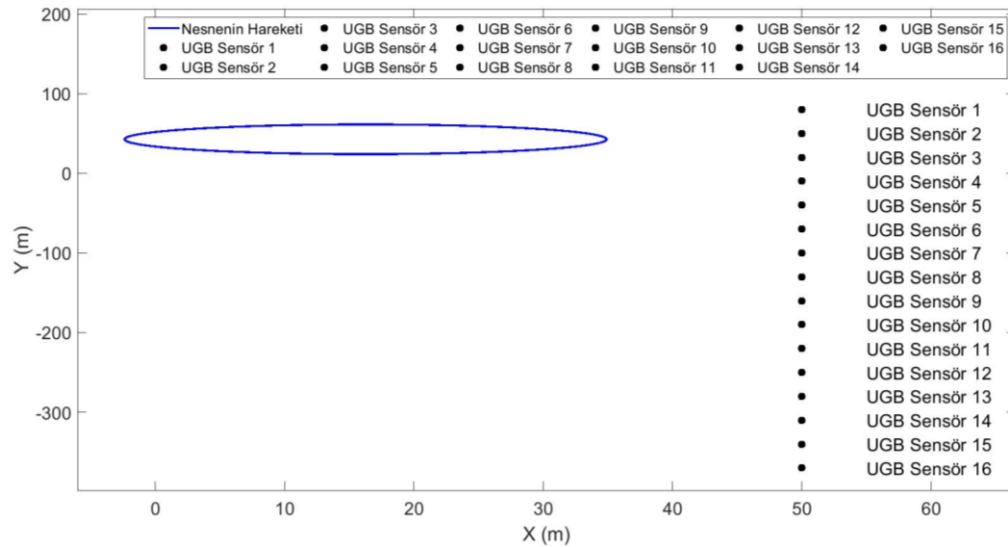
12 adet UGB sensör kullanılarak elde edilen 3. sensör geometrisinde, sensörler 1. sensörden, 12. sensöre doğru sırayla, (50, 80) metre konumundan, (50, -250) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir.



Şekil 3.50. 12 adet sensör kullanılarak tasarlanan 3. sensör geometrisi

16 adet UGB sensör kullanılarak tasarlanan 3. sensör geometrisi

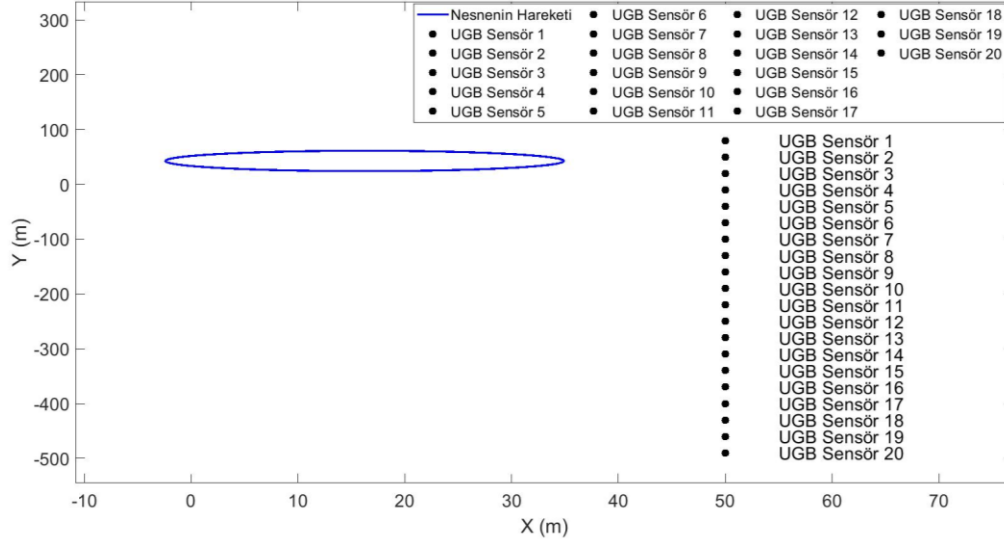
16 adet UGB sensör kullanılarak elde edilen 3. sensör geometrisinde, sensörler 1. sensörden, 16. sensöre doğru sırayla, (50, 80) metre konumundan, (50, -370) metre konumuna kadar 30 metre aralıklarla yerleştirilmiş olup, Şekil 3.51’de gösterilmiştir.



Şekil 3.51. 16 adet sensör kullanılarak tasarlanan 3. sensör geometrisi

20 adet UGB sensör kullanılarak tasarlanan 3. sensör geometrisi

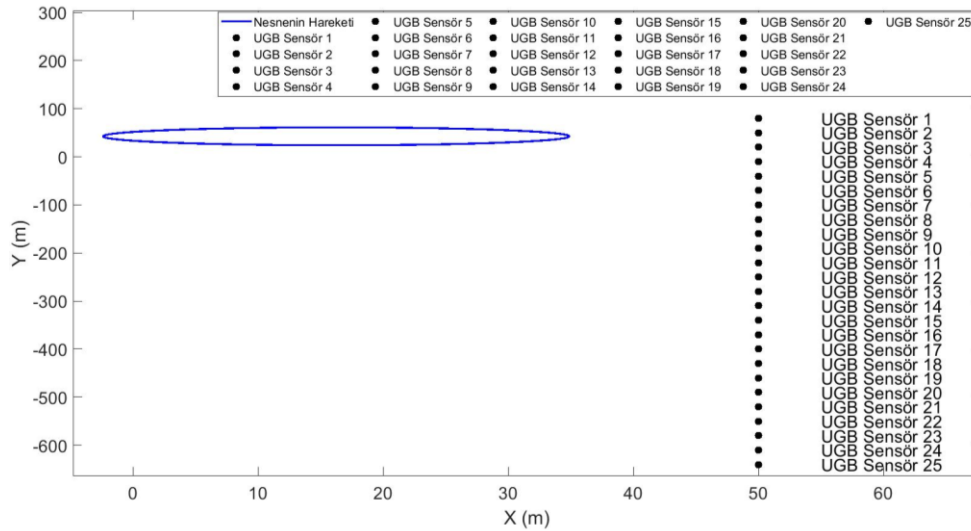
20 adet UGB sensör kullanılarak elde edilen 3. sensör geometrisinde, sensörler 1. sensörden, 20. sensöre doğru sırayla, (50, 80) metre konumundan, (50, -490) metre konumuna kadar 30 metre aralıklarla yerleştirilmiş olup, Şekil 3.52’de gösterilmiştir.



Şekil 3.52. 20 adet sensör kullanılarak tasarlanan 3. sensör geometrisi

25 adet UGB sensör kullanılarak tasarlanan 3. sensör geometrisi

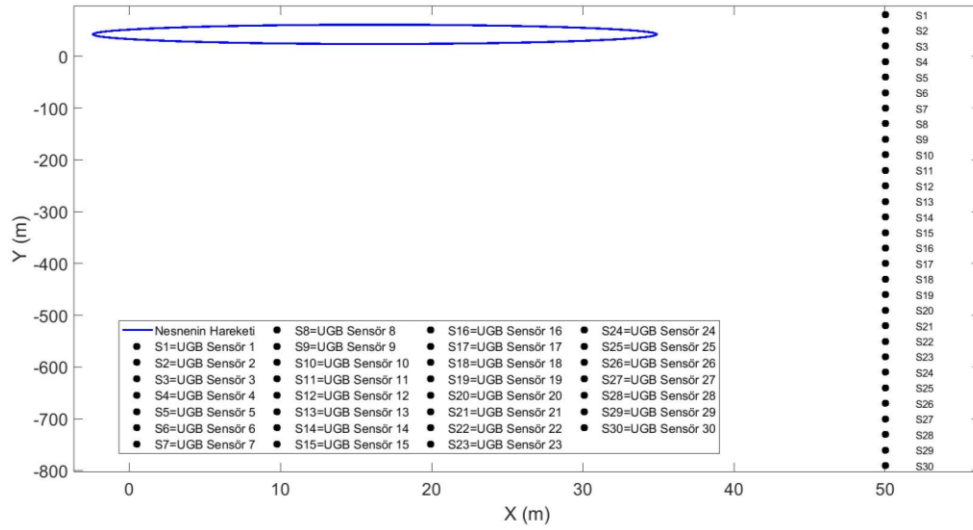
25 adet UGB sensör kullanılarak elde edilen 3. sensör geometrisinde, sensörler 1. sensörden, 25. sensöre doğru sırayla, (50, 80) metre konumundan, (50, -640) metre konumuna kadar 30 metre aralıklarla yerleştirilmiş olup, 3. sensör geometrisi Şekil 3.53’te gösterilmiştir.



Şekil 3.53. 25 adet sensör kullanılarak tasarlanan 3. sensör geometrisi

30 adet UGB sensör kullanılarak tasarlanan 3. sensör geometrisi

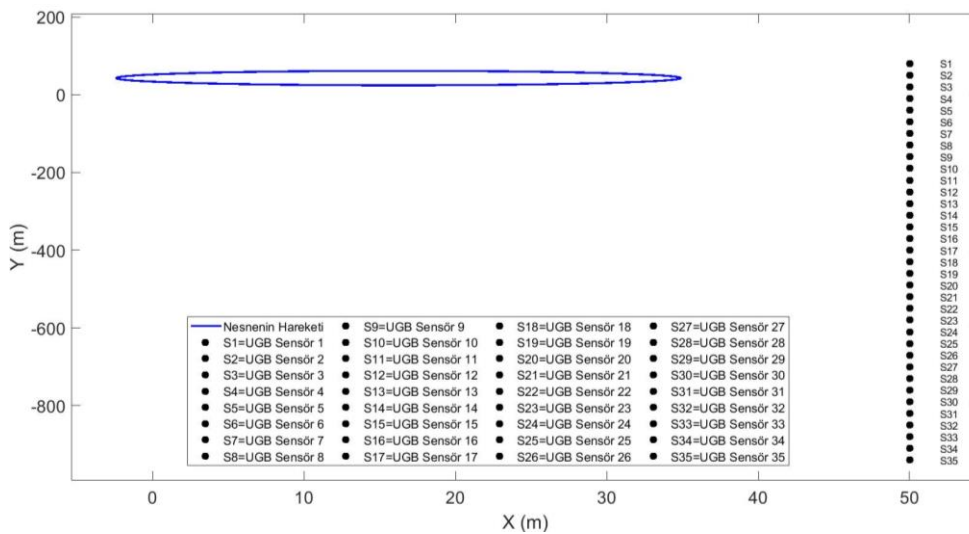
30 adet UGB sensör kullanılarak elde edilen 3. sensör geometrisinde, sensörler 1. sensörden, 30. sensöre doğru sırayla, (-10, 50) metre konumundan, (-10, 50) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 30 adet UGB sensör kullanılarak elde edilen 3. sensör geometrisi, Şekil 3.54'te gösterilmiştir.



Şekil 3.54. 30 adet sensör kullanılarak tasarlanan 3. sensör geometrisi

35 adet UGB sensör kullanılarak tasarlanan 3. sensör geometrisi

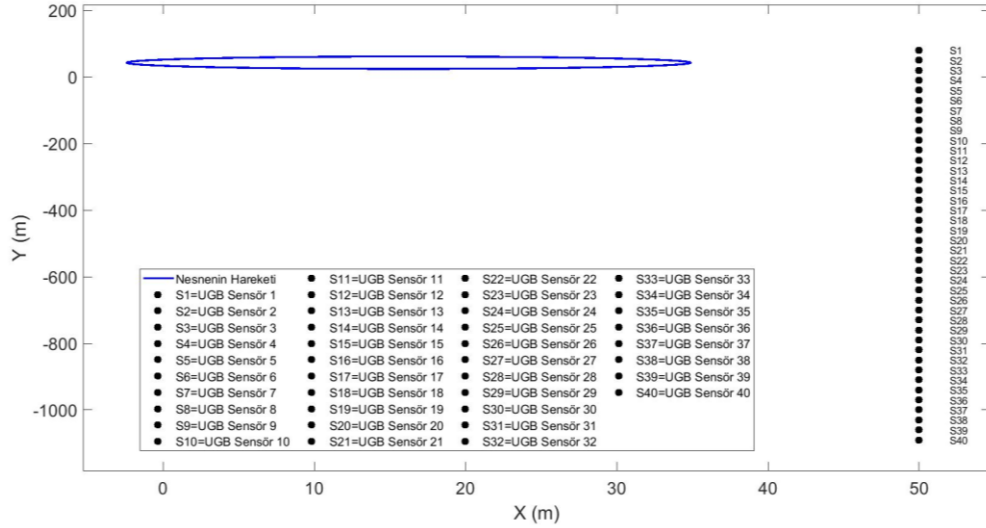
35 adet UGB sensör kullanılarak elde edilen 3. sensör geometrisinde, sensörler 1. sensörden, 35. sensöre doğru sırayla, (50, 80) metre konumundan, (50, -940) metre konumuna kadar 30 metre aralıklarla yerleştirilmiş olup, 3. sensör geometrisi Şekil 3.55'te gösterilmiştir.



Şekil 3.55. 35 adet sensör kullanılarak tasarlanan 3. sensör geometrisi

40 adet UGB sensör kullanılarak tasarlanan 3. sensör geometrisi

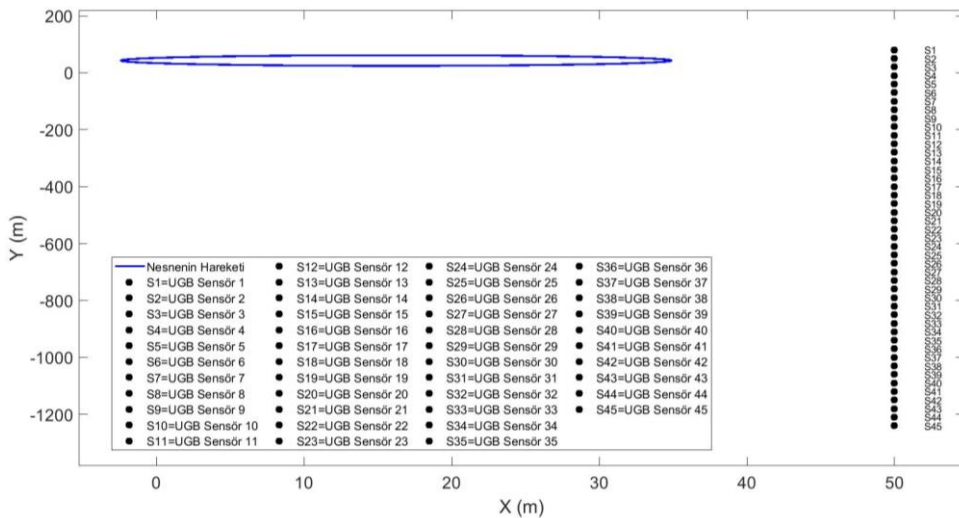
40 adet UGB sensör kullanılarak elde edilen 3. sensör geometrisinde, sensörler 1. sensörden, 40. sensöre doğru sırayla, (50, 80) metre konumundan, (50, -1090) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 40 adet UGB sensör kullanılarak elde edilen 3. sensör geometrisi Şekil 3.56’da gösterilmiştir.



Şekil 3.56. 40 adet sensör kullanılarak tasarlanan 3. sensör geometrisi

45 adet UGB sensör kullanılarak tasarlanan 3. sensör geometrisi

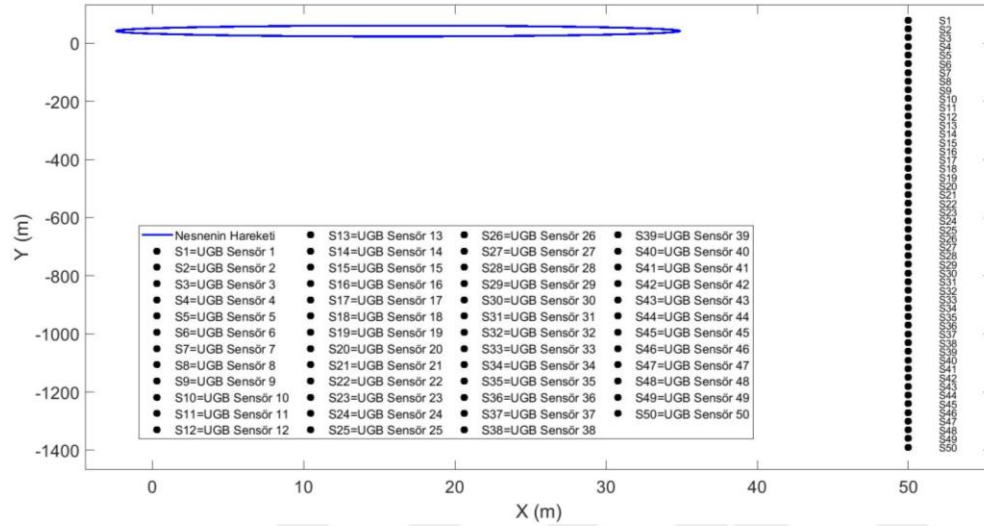
45 adet UGB sensör kullanılarak elde edilen 3. sensör geometrisinde, sensörler y eksenı boyunca, 1. sensörden, 45. sensöre doğru sırasıyla (50, 80) metre konumundan, (50, -1240) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 45 adet UGB sensör kullanılarak elde edilen 3. sensör geometrisi Şekil 3.57’de gösterilmiştir.



Şekil 3.57. 45 adet sensör kullanılarak tasarlanan 3. sensör geometrisi

50 adet UGB sensör kullanılarak tasarlanan 3. sensör geometrisi

3. sensör geometrisinde, sensörler y eksenini boyunca hareketli nesnenin sağ tarafında, 1. sensörden, 50. sensöre doğru sırasıyla (-10, 80) metre konumundan, (50, -1390) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 50 adet sensörle oluşturulan 3. sensör geometrisi Şekil 3.58’de gösterilmiştir.



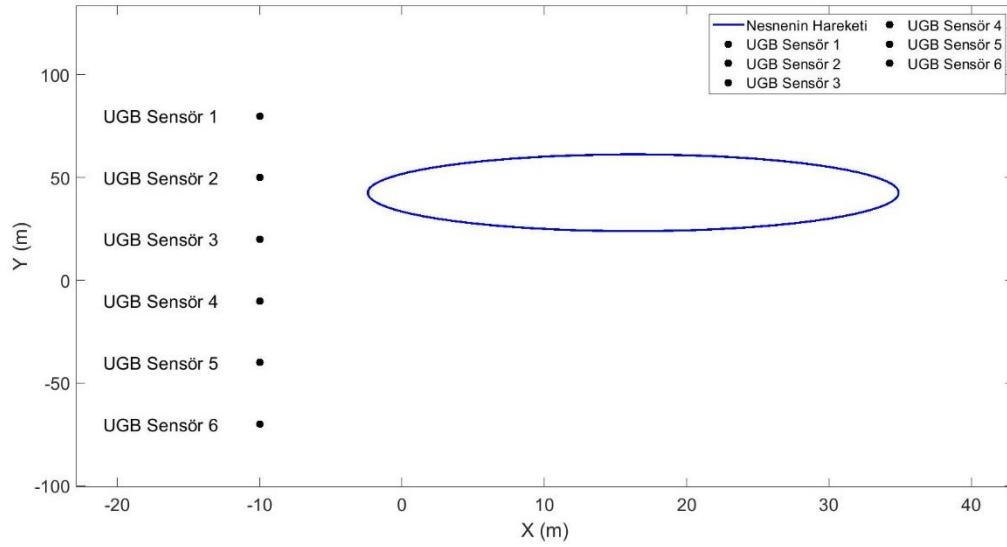
Şekil 3.58. 50 adet sensör kullanılarak tasarlanan 3. sensör geometrisi

3.5.4. Dördüncü sensör geometrisinde kullanılan sensör sayılarının arttırılmasıyla tasarlanan sensör geometrileri

Dördüncü sensör geometrisinde kullanılan sensör sayılarının arttırılmasıyla elde edilen yeni sensör geometrileri 3.5.4. numaralı bölümün alt bölümlerinde sırasıyla tanıtılmıştır.

6 adet UGB sensör kullanılarak tasarlanan 4. sensör geometrisi

6 adet UGB sensör kullanılarak elde edilen 4. sensör Şekil 3.59’da gösterilmiştir. 1. sensör (-10, 80) metre, 2. sensör (-10, 50) metre, 3.sensör (-10, 20) metre, 4. sensör (-10, -10) metre, 5. sensör (-10, -40) metre ve 6. sensör ise (-10, -70) metre konumunda bulunmaktadır.

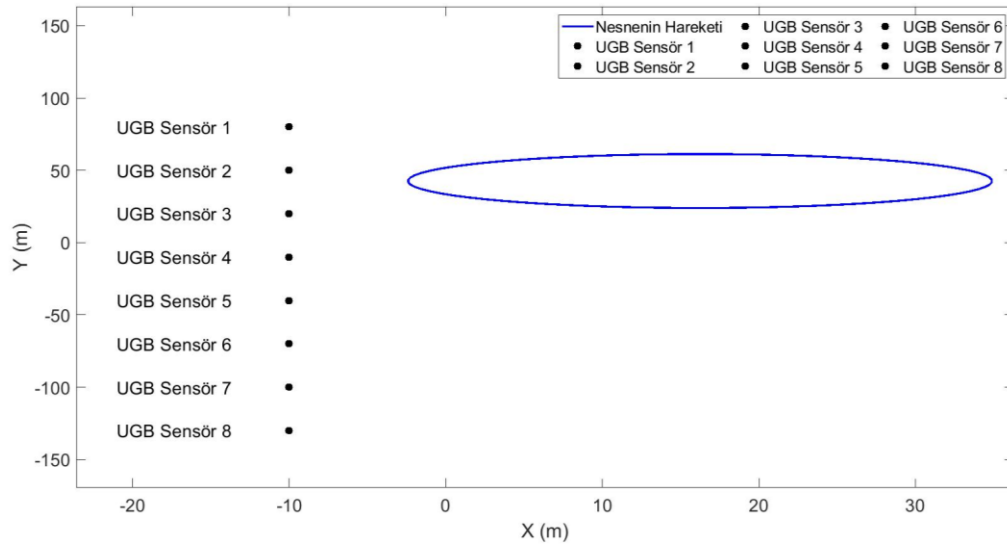


Şekil 3.59. 6 adet sensör kullanılarak tasarlanan 4. sensör geometrisi

8 adet UGB sensör kullanılarak tasarlanan 4. sensör geometrisi

8 adet UGB sensör kullanılarak elde edilen 4. sensör geometrisi Şekil 3.60'ta gösterilmiştir.

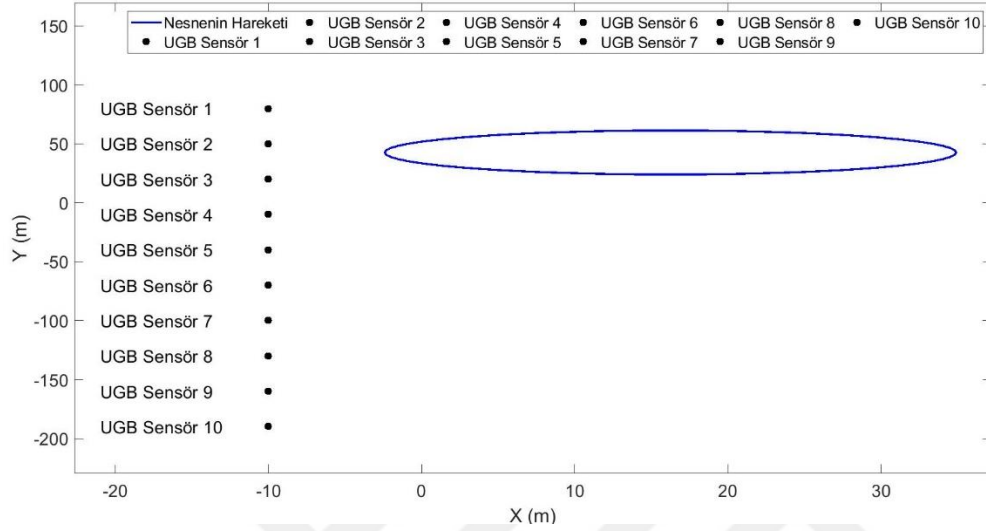
4. sensör geometrisinde, 1. sensör (-10, 80) metre, 2. sensör (-10, 50) metre, 3.sensör (-10, 20) metre, 4. sensör (-10, -10) metre, 5. sensör (80, -40) metre, 6. sensör (-10, -70) metre, 7.sensör (-10, -100) metre ve 8. sensör ise (-10, -130) metre konumunda bulunmaktadır.



Şekil 3.60. 8 adet sensör kullanılarak tasarlanan 4. sensör geometrisi

10 adet UGB sensör kullanılarak tasarlanan 4. sensör geometrisi

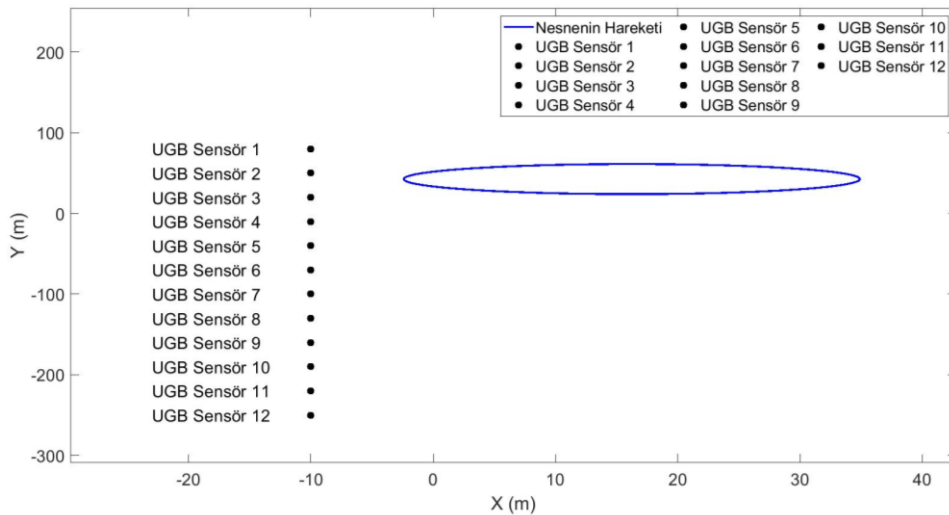
10 adet UGB sensör kullanılarak elde edilen 4. sensör geometrisinde, sensörler 1. sensörden, 10. sensöre doğru sırayla, (-10, 80) metre konumundan, (-10, -190) metre konumuna kadar 30 metre aralıklarla yerleştirilmiş olup, Şekil 3.61’de gösterilmiştir.



Şekil 3.61. 10 adet sensör kullanılarak tasarlanan 4. sensör geometrisi

12 adet UGB sensör kullanılarak tasarlanan 4. sensör geometrisi

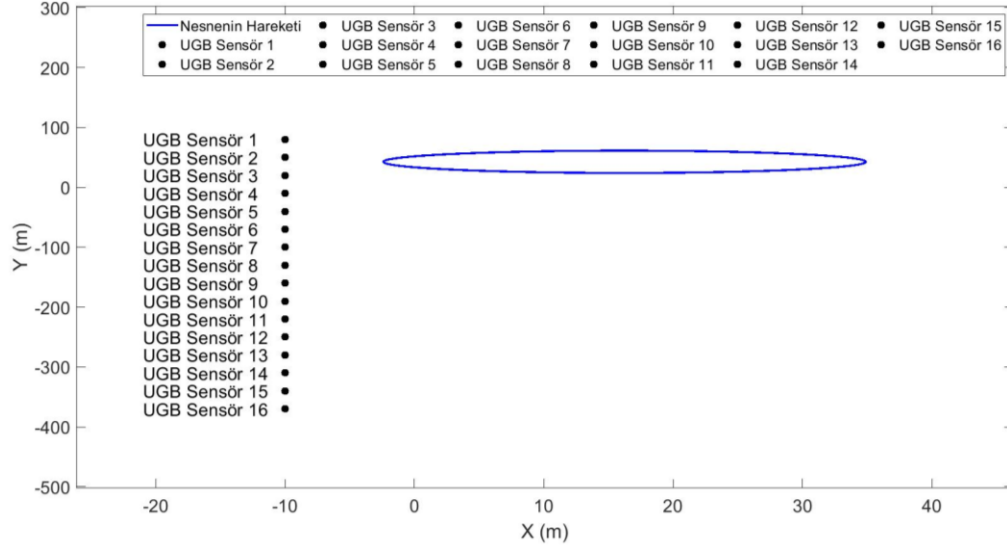
12 adet UGB sensör kullanılarak elde edilen 4. sensör geometrisinde, sensörler 1. sensörden, 12. sensöre doğru sırayla, (-10, 80) metre konumundan, (-10, -250) metre konumuna kadar 30 metre aralıklarla yerleştirilmiş olup, Şekil 3.62’de gösterilmiştir.



Şekil 3.62. 12 adet sensör kullanılarak tasarlanan 4. sensör geometrisi

16 adet UGB sensör kullanılarak tasarlanan 4. sensör geometrisi

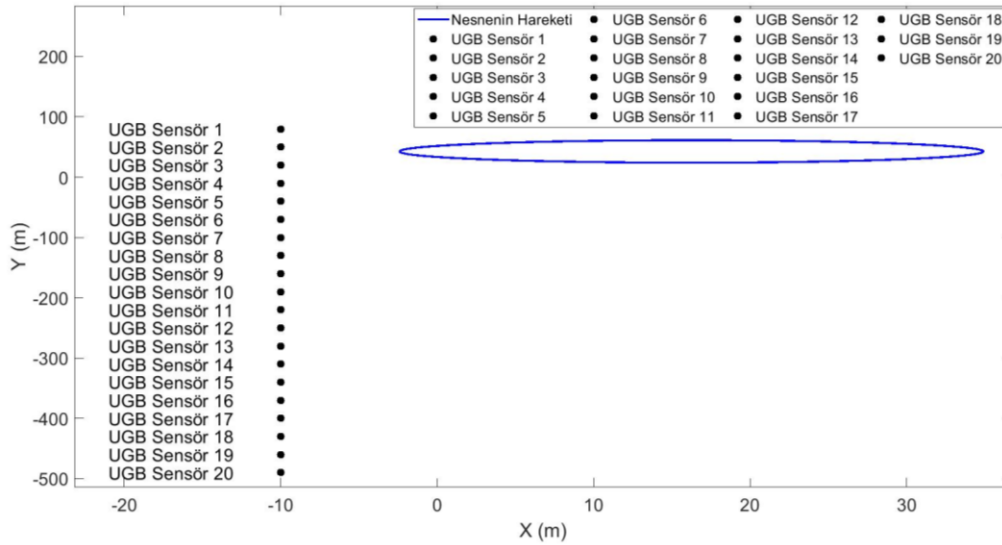
16 adet UGB sensör kullanılarak elde edilen 4. sensör geometrisinde, sensörler 1. sensörden, 16. sensöre doğru sırayla, (-10, 80) metre konumundan, (-10, 370) metre konumuna kadar 30 metre aralıklarla yerleştirilmiş olup Şekil 3.63'te gösterilmiştir.



Şekil 3.63. 16 adet sensör kullanılarak tasarlanan 4. sensör geometrisi

20 adet UGB sensör kullanılarak tasarlanan 4. sensör geometrisi

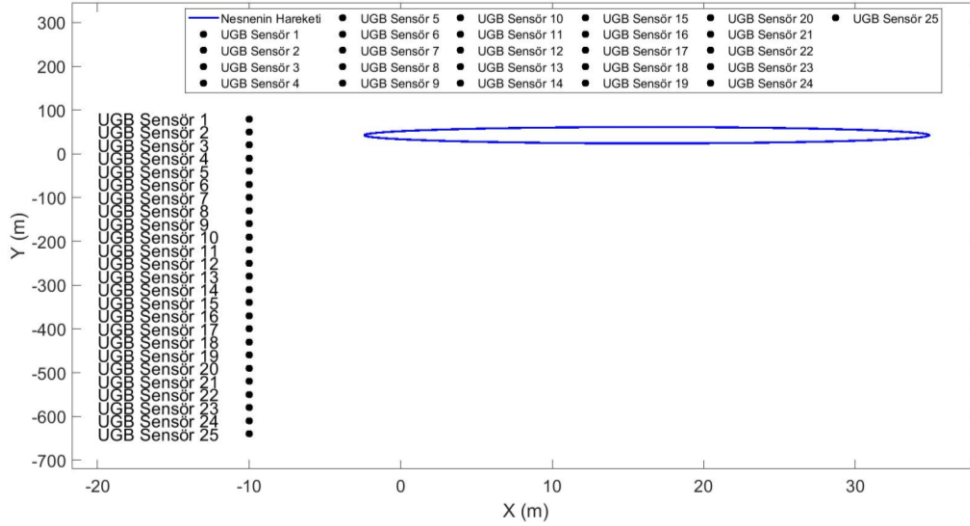
20 adet UGB sensör kullanılarak elde edilen 4. sensör geometrisinde, sensörler 1. sensörden, 20. sensöre doğru sırayla, (-10, 80) metre konumundan, (-10, -490) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir.



Şekil 3.64. 20 adet sensör kullanılarak tasarlanan 4. sensör geometrisi

25 adet UGB sensör kullanılarak tasarlanan 4. sensör geometrisi

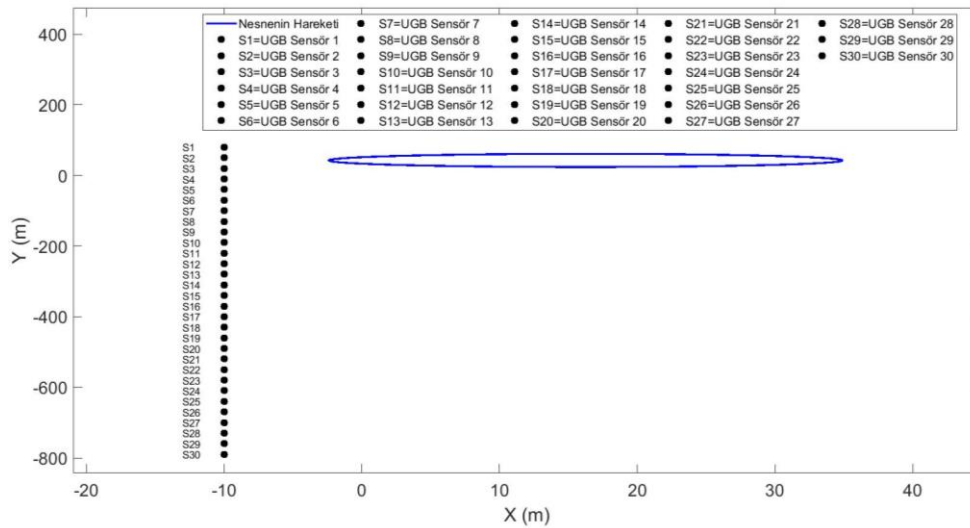
25 adet UGB sensör kullanılarak elde edilen 4. sensör geometrisinde, sensörler 1. sensörden, 25. sensöre doğru sırayla, (-10, 80) metre konumundan, (-10, -640) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 25 adet UGB sensör kullanılarak elde edilen 4. sensör geometrisi Şekil 3.65'te gösterilmiştir.



Şekil 3.65. 25 adet sensör kullanılarak tasarlanan 4. sensör geometrisi

30 adet UGB sensör kullanılarak tasarlanan 4. sensör geometrisi

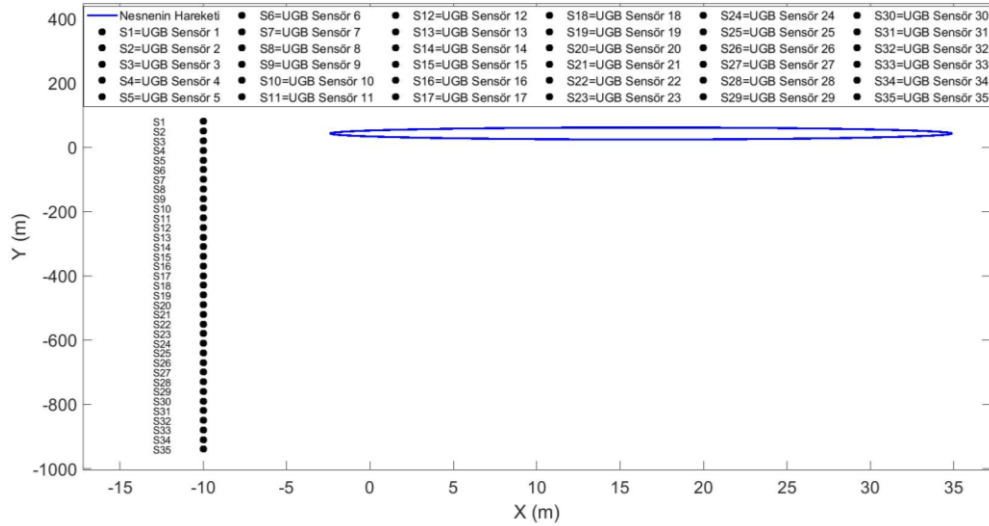
30 adet UGB sensör kullanılarak elde edilen 4. sensör geometrisinde, sensörler 1. sensörden, 30. sensöre doğru sırayla, (-10, 80) metre konumundan, (-10, -790) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 30 adet UGB sensör kullanılarak elde edilen 4. sensör geometrisi Şekil 3.66'da gösterilmiştir.



Şekil 3.66. 30 adet sensör kullanılarak tasarlanan 4. sensör geometrisi

35 adet UGB sensör kullanılarak tasarlanan 4. sensör geometrisi

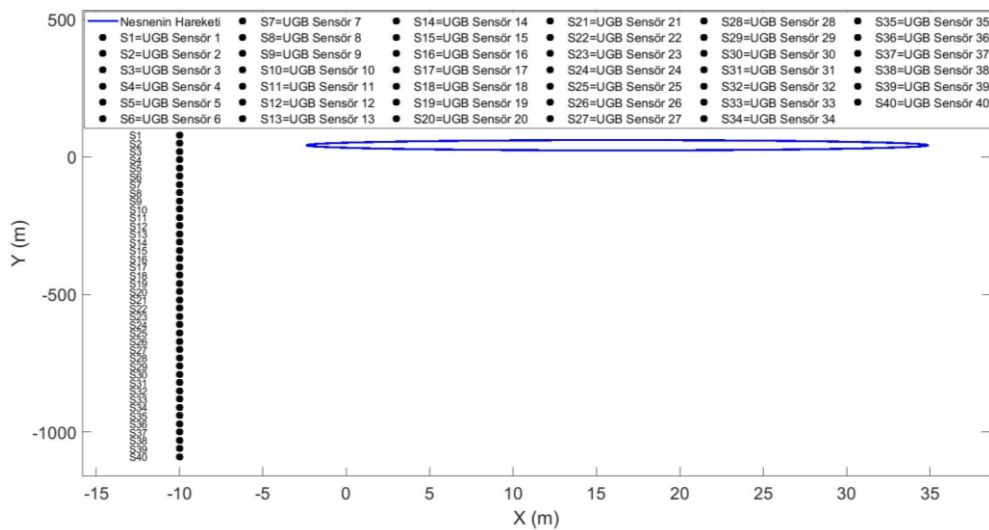
35 adet UGB sensör kullanılarak elde edilen 4. sensör geometrisinde, sensörler 1. sensörden, 35. sensöre doğru sırayla, (-10, 80) metre konumundan, (-10, -940) metre konumuna kadar 30 metre aralıklarla yerleştirilmiş olup, 4. sensör geometrisi Şekil 3.67’de gösterilmiştir.



Şekil 3.67. 35 adet sensör kullanılarak tasarlanan 4. sensör geometrisi

40 adet UGB sensör kullanılarak tasarlanan 4. sensör geometrisi

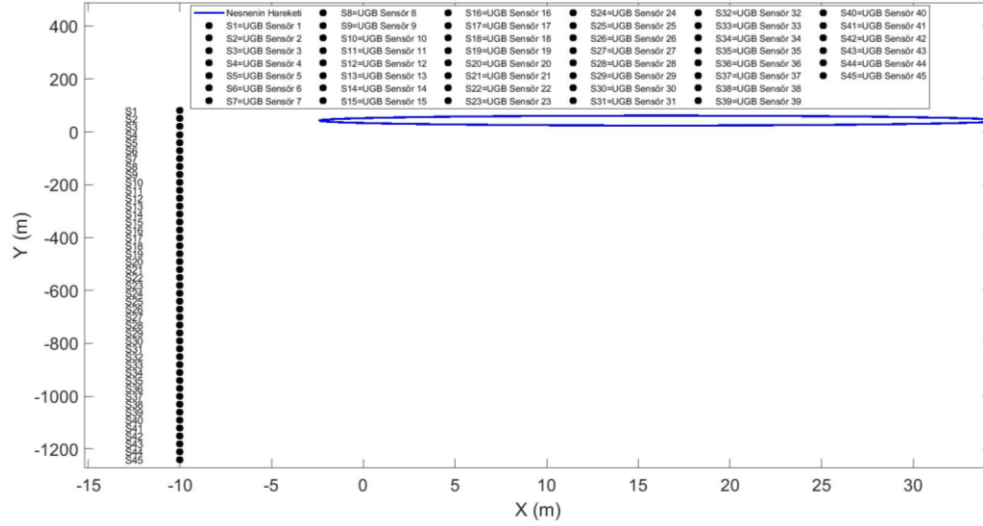
40 adet UGB sensör kullanılarak elde edilen 4. sensör geometrisinde, sensörler 1. sensörden, 40. sensöre doğru sırayla, (-10, 80) metre konumundan, (-10, -1090) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 40 adet UGB sensör kullanılarak elde edilen 4. sensör geometrisi Şekil 3.68’de gösterilmiştir.



Şekil 3.68. 40 adet sensör kullanılarak tasarlanan 4. sensör geometrisi

45 adet UGB sensör kullanılarak tasarlanan 4. sensör geometrisi

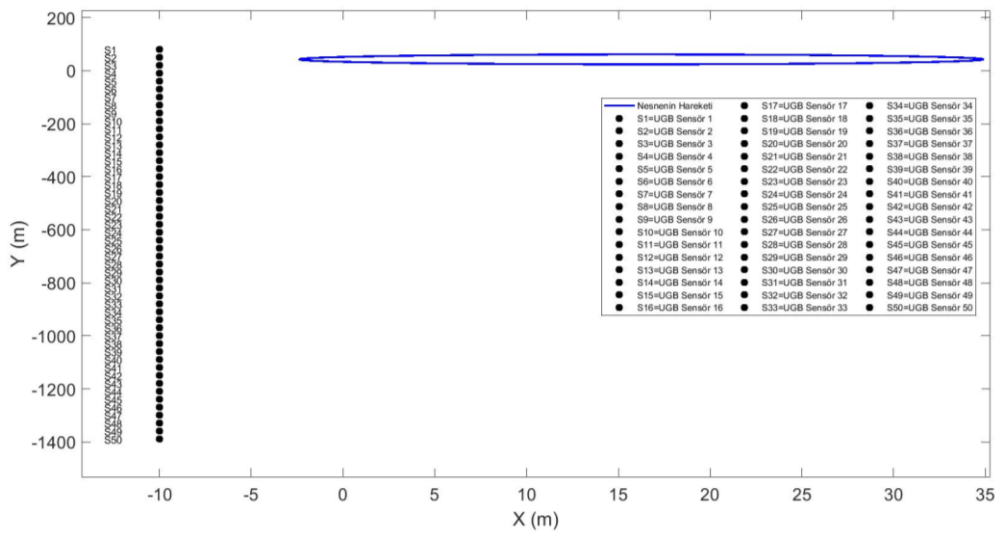
45 adet UGB sensör kullanılarak elde edilen 4. sensör geometrisinde, sensörler y eksenı boyunca, 1. sensörden, 45. sensöre doğru sırasıyla (-10, 80) metre konumundan, (-10, -1240) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 45 adet sensör kullanılarak elde edilen 4. sensör geometrisi Şekil 3.69’da gösterilmiştir.



Şekil 3.69. 45 adet sensör kullanılarak tasarlanan 4. sensör geometrisi

50 adet UGB sensör kullanılarak tasarlanan 4. sensör geometrisi

50 adet UGB sensör kullanılarak elde edilen 4. sensör geometrisinde, sensörler y eksenı boyunca, 1. sensörden, 50. sensöre doğru sırasıyla (-10, 80) metre konumundan, (-10, -1390) metre konumuna kadar 30 metre aralıklarla yerleştirilmiş ve Şekil 3.70’de gösterilmiştir.



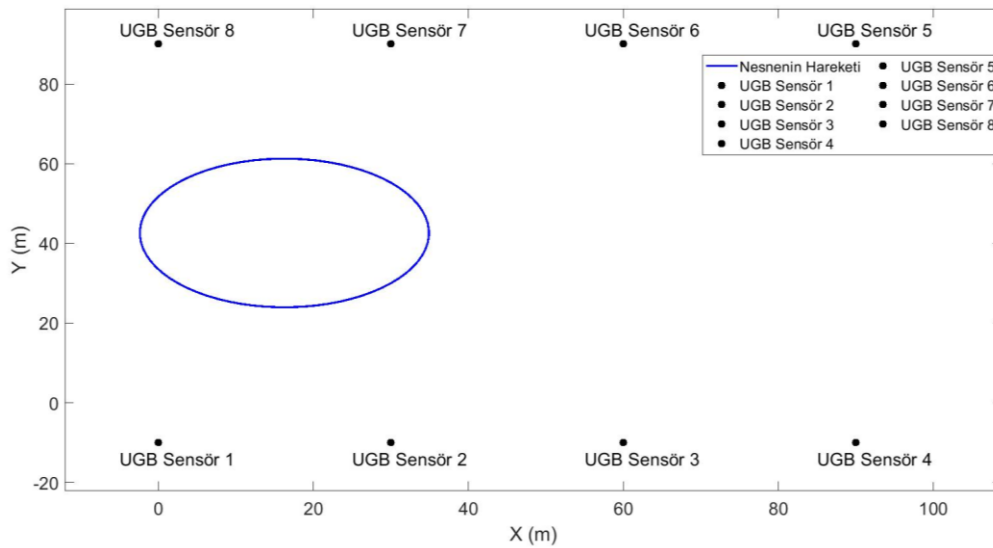
Şekil 3.70. 50 adet sensör kullanılarak tasarlanan 4. sensör geometrisi

3.5.5. Beşinci sensör geometrisinde kullanılan sensör sayılarının arttırılmasıyla tasarlanan sensör geometrileri

Beşinci sensör geometrisinde kullanılan sensör sayılarının arttırılmasıyla elde edilen yeni sensör geometrileri 3.5.5. numaralı bölümün alt bölümlerinde sırasıyla tanıtılmıştır.

8 adet UGB sensör kullanılarak tasarlanan 5. sensör geometrisi

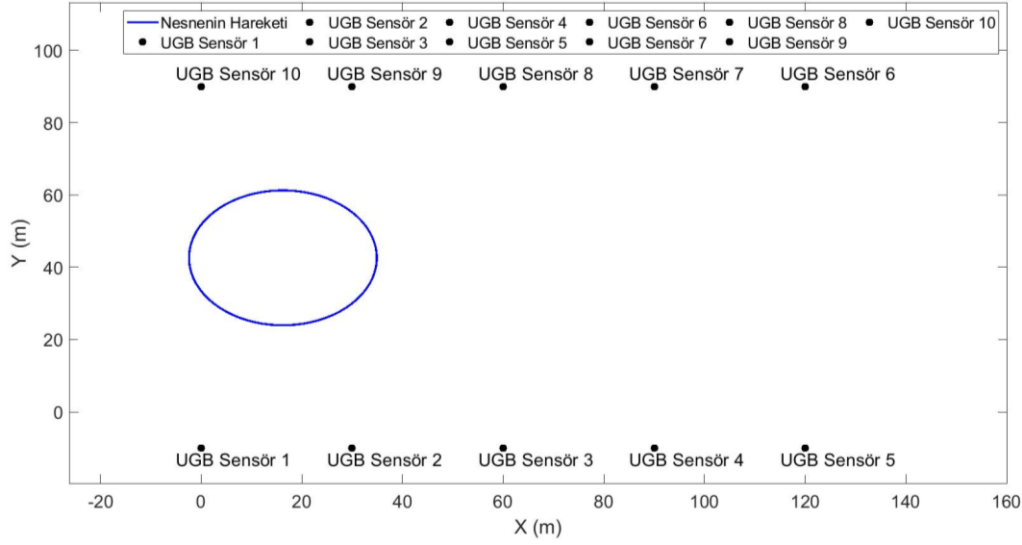
8 adet UGB sensör kullanılarak elde edilen 5. sensör geometrisinde, sensörler 1. sensörden, 4. sensöre doğru sırayla, (0, -10) metre konumundan, (90, -10) metreye ve 5. sensörden, 8. sensöre ise (90, -10) metreden, (0, 90) metreye kadar 30 metre aralıklarla yerleştirilmiş olup, Şekil 3.71’de gösterilmiştir.



Şekil 3.71. 8 adet sensör kullanılarak tasarlanan 5. sensör geometrisi

10 adet UGB sensör kullanılarak tasarlanan 5. sensör geometrisi

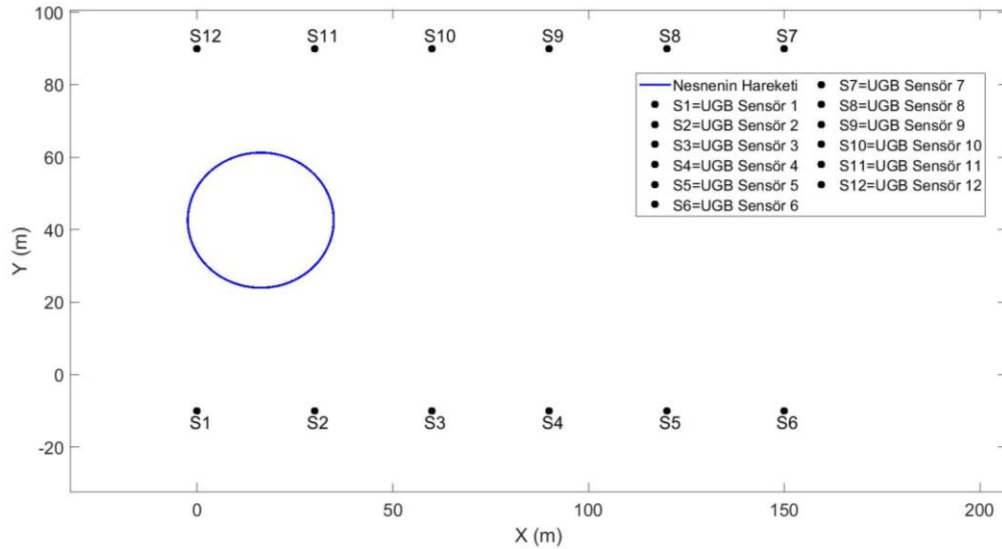
10 adet UGB sensör kullanılarak elde edilen 5. sensör geometrisinde, sensörler 1. sensörden, 5. sensöre doğru sırayla, (0, -10) metre konumundan, (120, -10) ve 6. sensörden, 10’a kadar olan sensörler ise sırayla (120, 90) metreden, (0, 90) metreye kadar 30 metre aralıklarla yerleştirilmiş olup, Şekil 3.72’de gösterilmiştir.



Şekil 3.72. 10 adet sensör kullanılarak tasarlanan 5. sensör geometrisi

12 adet UGB sensör kullanılarak tasarlanan 5. sensör geometrisi

12 adet UGB sensör kullanılarak elde edilen 5. sensör geometrisinde, sensörler 1. sensörden, 6. sensöre doğru sırayla, (0, -10) metre konumundan, (150, -10) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 7. sensörden, 12. sensöre kadar olan sensörler ise sırayla (150, 90) metreden, (0, 90) metreye kadar 30 metre aralıklarla yerleştirilmiş olup, Şekil 3.73'te gösterilmiştir.



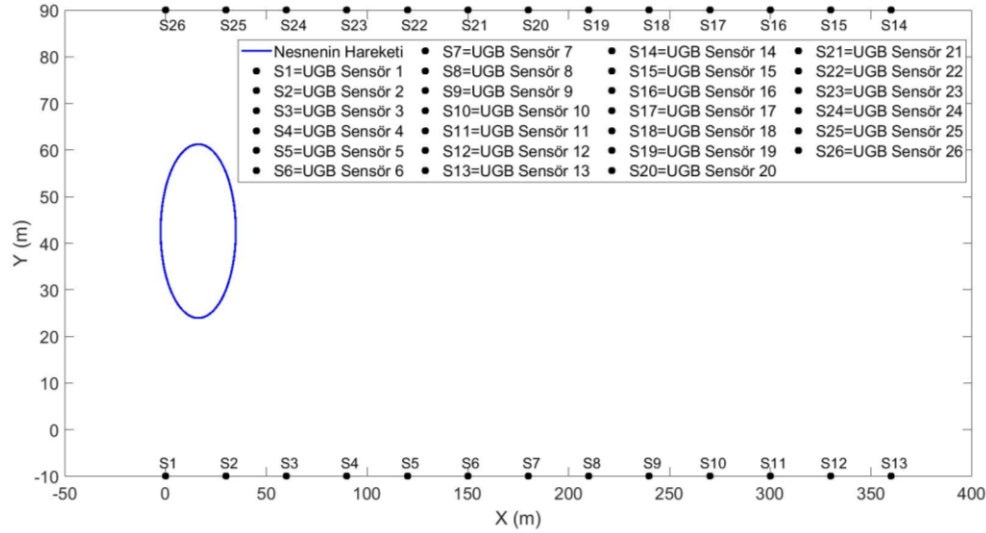
Şekil 3.73. 12 adet sensör kullanılarak tasarlanan 5. sensör geometrisi

16 adet UGB sensör kullanılarak tasarlanan 5. sensör geometrisi

5. sensör geometrisinde, sensörler 1. sensörden, 8. sensöre doğru sırayla, (0, -10) metre konumundan, (210, -10) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 9.

26 adet UGB sensör kullanılarak tasarlanan 5. sensör geometrisi

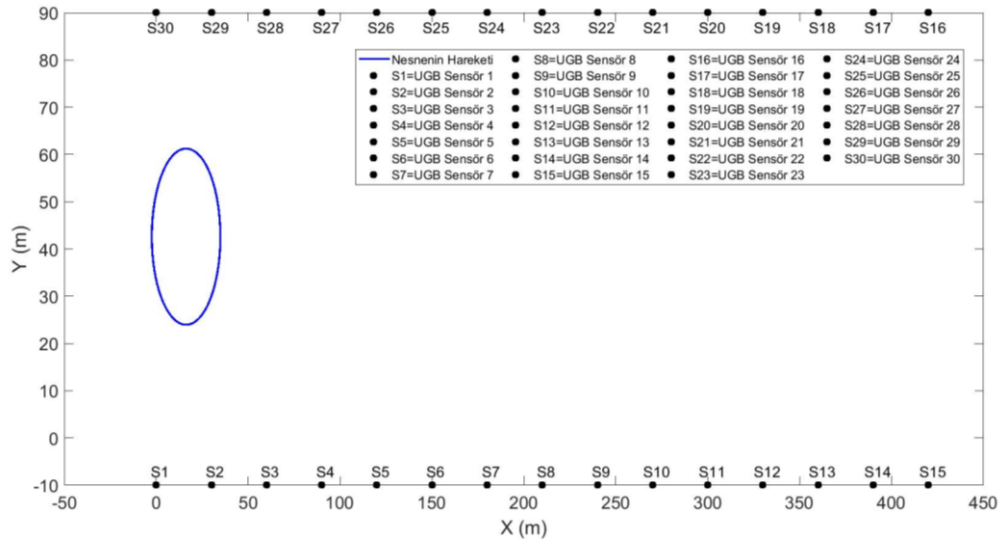
26 adet UGB sensör kullanılarak elde edilen 5. sensör geometrisinde, sensörler 1. sensörden, 13. sensöre doğru sırayla, (0, -10) metre konumundan, (360, -10) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 14. sensörden, 26. sensöre kadar olan sensörler ise sırayla (360, 90) metreden, (0, 90) metreye kadar 30 metre aralıklarla yerleştirilmiştir.



Şekil 3.76. 26 adet sensör kullanılarak tasarlanan 5. sensör geometrisi

30 adet UGB sensör kullanılarak tasarlanan 5. sensör geometrisi

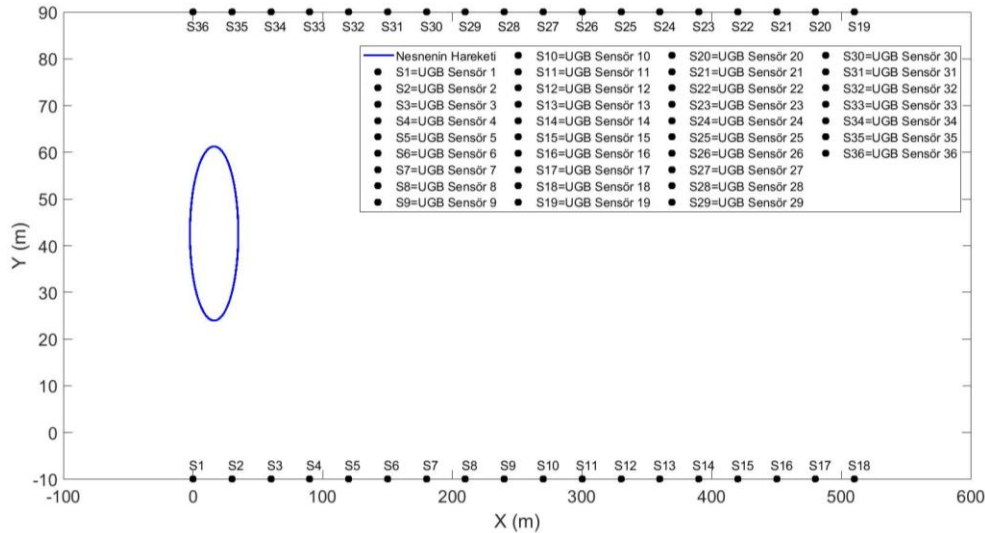
30 adet UGB sensör kullanılarak elde edilen 5. sensör geometrisinde, sensörler 1. sensörden, 15. sensöre doğru sırayla, (0, -10) metre konumundan, (420, -10) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 16. sensörden, 30. sensöre kadar olan sensörler ise sırayla (420, 90) metreden, (0, 90) metreye kadar 30 metre aralıklarla yerleştirilmiştir. 30 adet UGB sensör kullanılarak elde edilen 5. sensör geometrisi, Şekil 3.77'de gösterilmiştir.



Şekil 3.77. 30 adet sensör kullanılarak tasarlanan 5. sensör geometrisi

36 adet UGB sensör kullanılarak tasarlanan 5. sensör geometrisi

36 adet UGB sensör kullanılarak elde edilen 5. sensör geometrisinde, sensörler 1. sensörden, 18. sensöre doğru sırayla, (0, -10) metre konumundan, (510, -10) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 19. sensörden, 36. sensöre kadar olan sensörler ise sırayla (510, 90) metreden, (0, 90) metreye kadar 30 metre aralıklarla yerleştirilmiştir. 36 adet UGB sensör kullanılarak elde edilen 5. sensör geometrisi, Şekil 3.78’te gösterilmiştir.

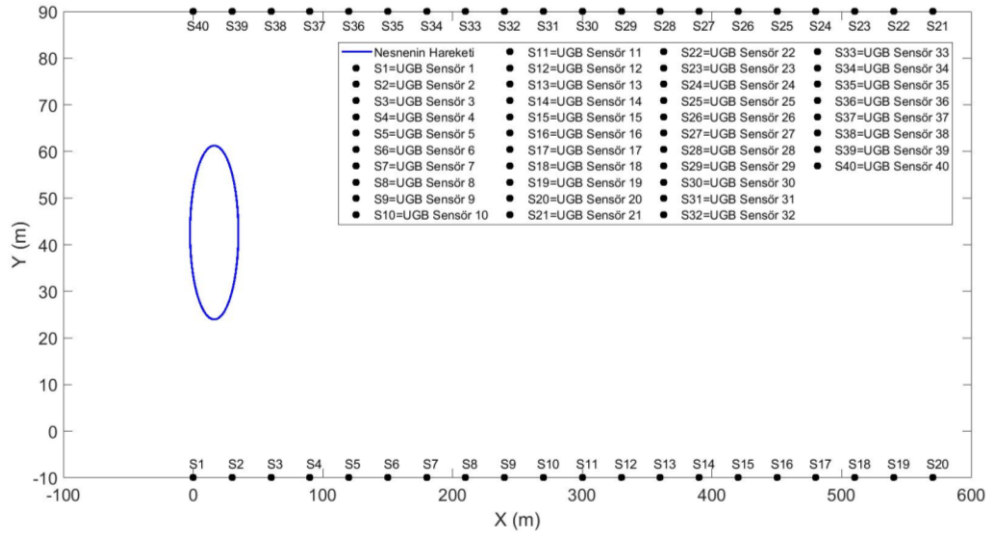


Şekil 3.78. 36 adet sensör kullanılarak tasarlanan 5. sensör geometrisi

40 adet UGB sensör kullanılarak tasarlanan 5. sensör geometrisi

Sensörler 1. sensörden, 20. sensöre doğru sırayla, (0, -10) metre konumundan, (570, -10) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 21. sensörden, 40. sensöre kadar

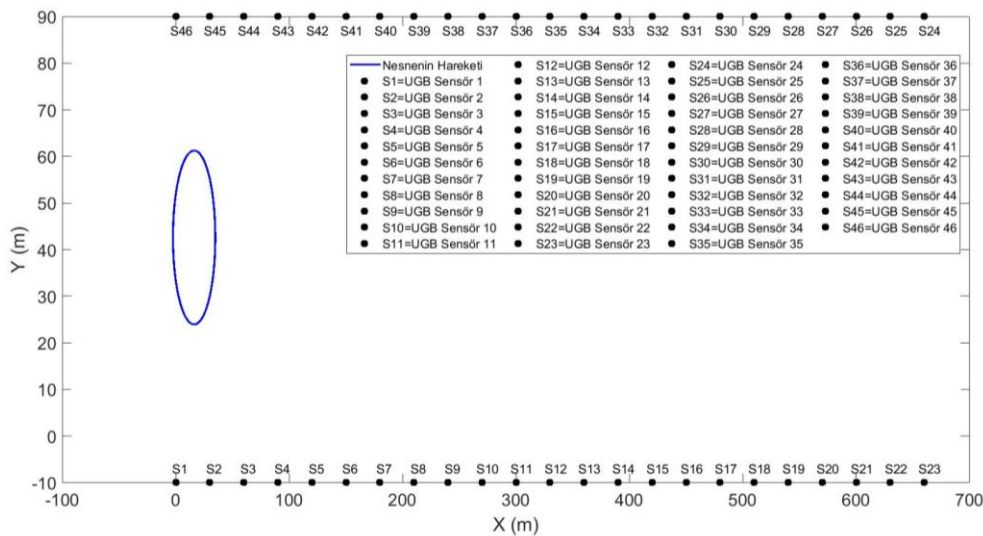
olan sensörler ise (570, 90) metreden, (0, 90) metreye kadar 30 metre aralıklarla yerleştirilmiş olup, 5. sensör geometrisi Şekil 3.79’da gösterilmiştir.



Şekil 3.79. 40 adet sensör kullanılarak tasarlanan 5. sensör geometrisi

46 adet UGB sensör kullanılarak tasarlanan 5. sensör geometrisi

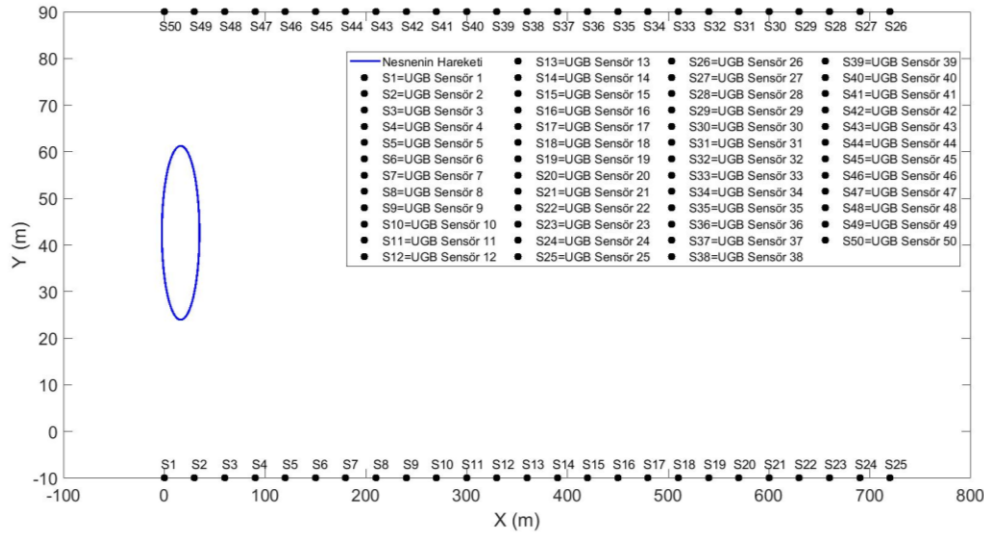
46 adet sensör kullanılarak elde edilen 5. sensör geometrisinde, sensörler, 1. sensörden, 23. sensöre kadar (0, -10) metre konumundan, (660, -10) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 24. sensörden, 46. sensöre kadar olan sensörler ise sırasıyla (660, 90) metre konumundan, (0, 90) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 46 adet UGB sensör kullanılarak elde edilen 5. sensör geometrisi, Şekil 3.80’de gösterilmiştir.



Şekil 3.80. 46 adet sensör kullanılarak tasarlanan 5. sensör geometrisi

50 adet UGB sensör kullanılarak tasarlanan 5. sensör geometrisi

50 adet UGB sensör kullanılarak elde edilen 5. sensör geometrisi, Şekil 3.81’de gösterilmiştir. Sensörler, 1’den, 25’e kadar sırasıyla (0, -10) metre konumundan, (720, -10) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 26. sensörden, 50. sensöre kadar olan sensörler ise sırasıyla (720, 90) metre konumundan, (0, 90) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir.



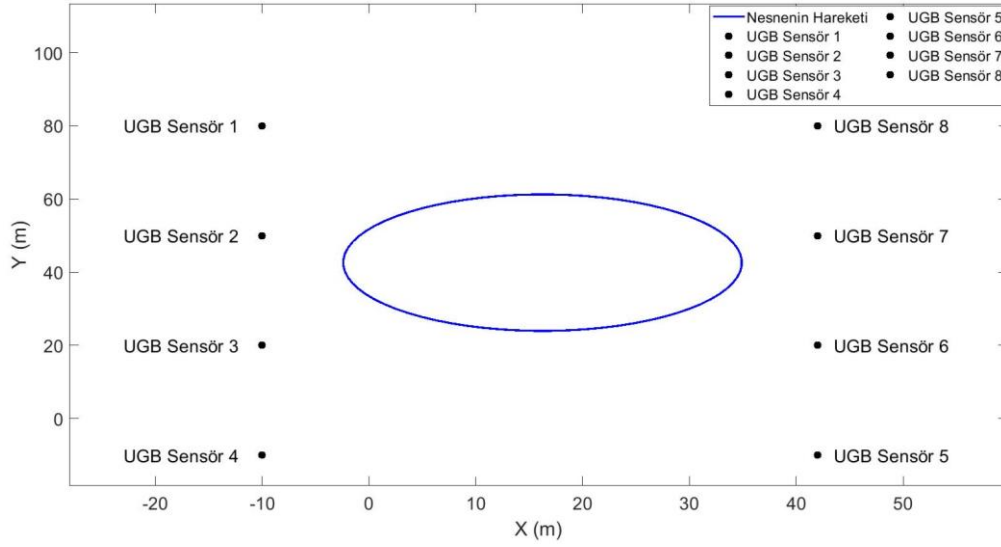
Şekil 3.81. 50 adet sensör kullanılarak tasarlanan 5. sensör geometrisi

3.5.6. Altıncı sensör geometrisinde kullanılan sensör sayılarının arttırılmasıyla tasarlanan sensör geometrileri

Altıncı sensör geometrisinde kullanılan sensör sayılarının arttırılmasıyla elde edilen yeni sensör geometrileri 3.5.6. numaralı bölümün alt bölümlerinde sırasıyla tanıtılmıştır.

8 adet UGB sensör kullanılarak tasarlanan 6. sensör geometrisi

6. sensör geometrisinde, sensörler 1. sensörden, 4. sensöre doğru sırayla, (-10, 80) metre konumundan, (-10, -10) metreye ve 5. sensörden, 8. sensöre kadar ise (42, -10) metreden, (42, 80) metreye kadar 30 metre aralıklarla yerleştirilmiştir.



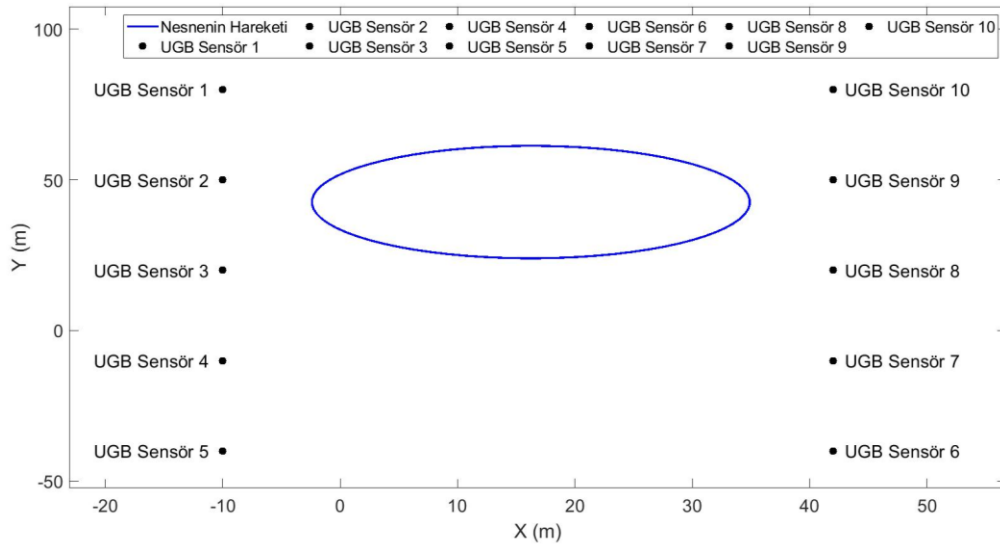
Şekil 3.82. 8 adet sensör kullanılarak tasarlanan 6. sensör geometrisi

3.5.3. 10 adet UGB sensör kullanılarak tasarlanan sensör geometrileri

Bölüm 3.4.2.'de anlatılan, her bir sensör geometrilerinde, sensör sayıları artırılarak 10 adet UGB sensör kullanılarak oluşturulan sensör geometrileri bölüm 3.5.3'te anlatılmıştır.

10 adet UGB sensör kullanılarak tasarlanan 6. sensör geometrisi

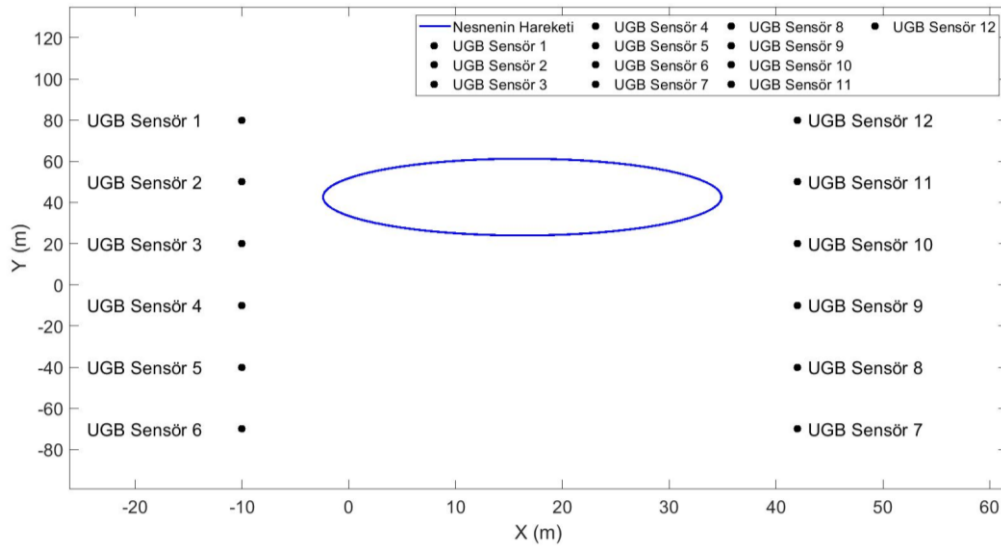
10 adet UGB sensör kullanılarak elde edilen 6. sensör geometrisinde, sensörler 1. sensörden, 5. sensöre doğru sırayla, (-10, 80) metre konumundan, (-10, -40) ve 6. sensörden, 10'a kadar olan sensörler ise sırayla (42, -40) metreden, (42, 80) metreye kadar 30 metre aralıklarla yerleştirilmiş olup, Şekil 3.83'te gösterilmiştir.



Şekil 3.83. 10 adet sensör kullanılarak tasarlanan 6. sensör geometrisi

12 adet UGB sensör kullanılarak tasarlanan 6. sensör geometrisi

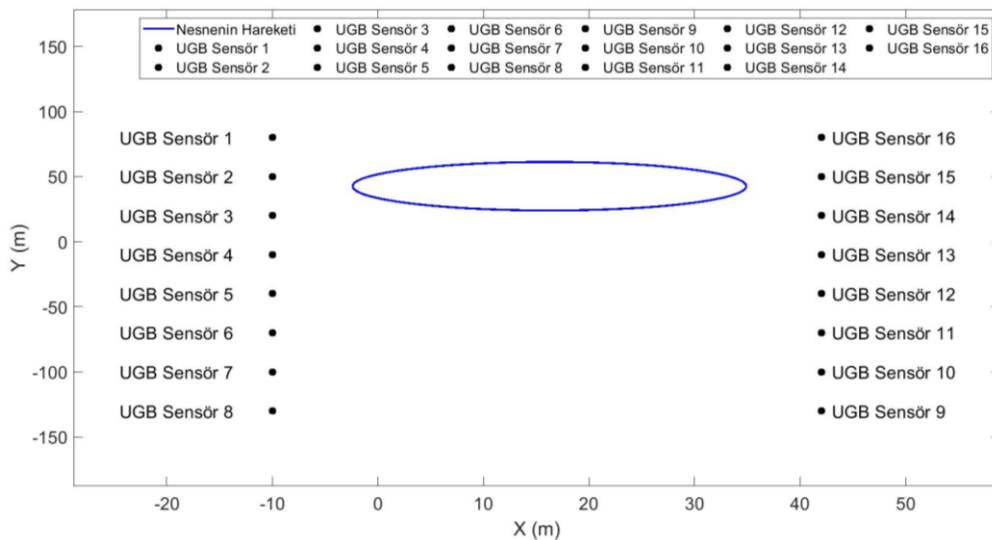
12 adet UGB sensör kullanılarak elde edilen 6. sensör geometrisinde, sensörler 1. sensörden, 6. sensöre doğru sırayla, (-10, 80) metre konumundan, (-10, -70) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 7. sensörden, 12. sensöre kadar olan sensörler ise sırayla (42, -70) metreden, (42, 80) metreye kadar 30 metre aralıklarla yerleştirilmiştir.



Şekil 3.84. 12 adet sensör kullanılarak tasarlanan 6. sensör geometrisi

16 adet UGB sensör kullanılarak tasarlanan 6. sensör geometrisi

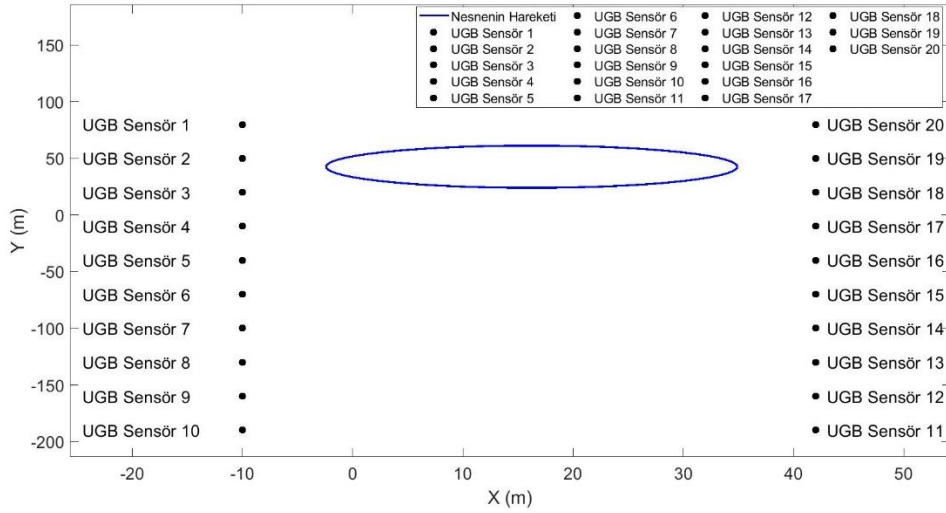
16 adet UGB sensör kullanılarak elde edilen 6. sensör geometrisinde, sensörler 1. sensörden, 8. sensöre doğru sırayla, (-10, 80) metre konumundan, (-10, -130) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 9. sensörden, 16. sensöre kadar olan sensörler ise sırayla (42, -130) metreden, (42, 80) metreye kadar 30 metre aralıklarla yerleştirilmiştir.



Şekil 3.85. 16 adet sensör kullanılarak tasarlanan 6. sensör geometrisi

20 adet UGB sensör kullanılarak tasarlanan 6. sensör geometrisi

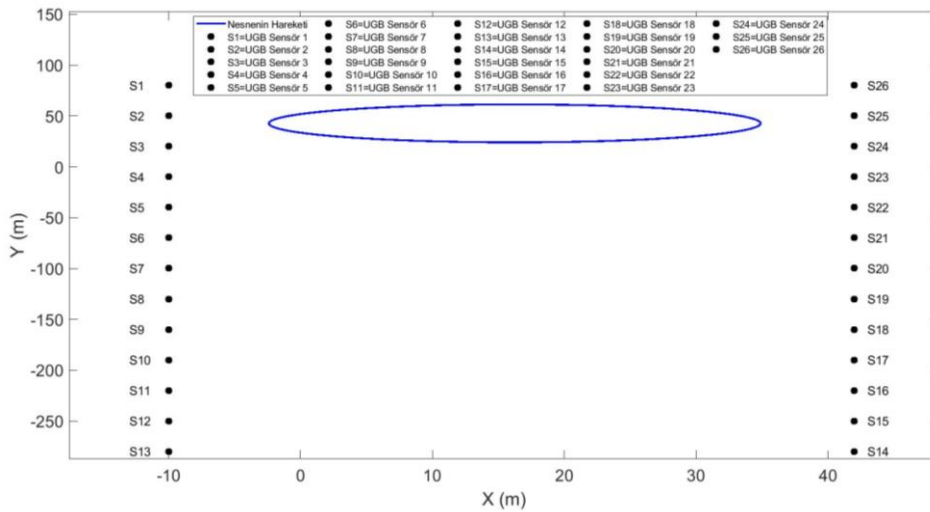
20 adet UGB sensör kullanılarak elde edilen 6. sensör geometrisinde, sensörler 1. sensörden, 10. sensöre doğru sırayla, (-10, 80) metre konumundan, (-10, -190) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 11. sensörden, 20. sensöre kadar olan sensörler ise sırayla (42, -190) metreden, (42, 80) metreye kadar 30 metre aralıklarla yerleştirilmiş olup, Şekil 3.86’da gösterilmiştir.



Şekil 3.86. 20 adet sensör kullanılarak tasarlanan 6. sensör geometrisi

26 adet UGB sensör kullanılarak tasarlanan 6. sensör geometrisi

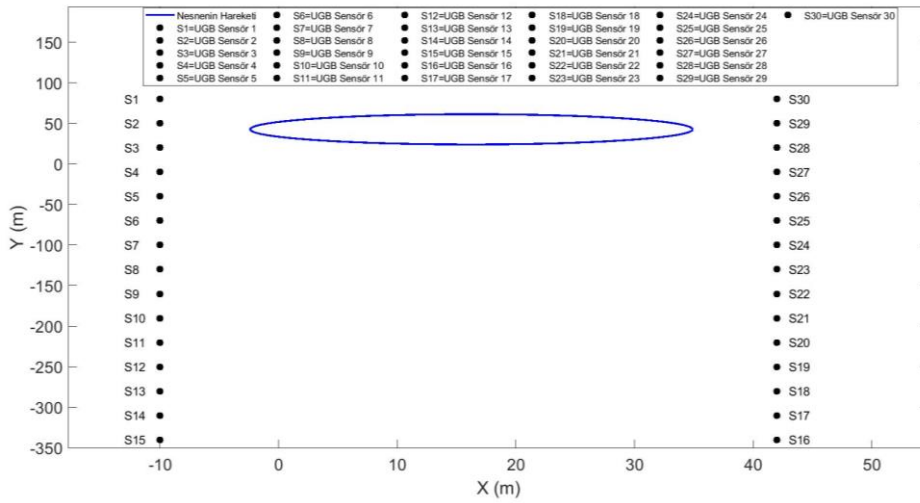
26 adet UGB sensör kullanılarak elde edilen 6. sensör geometrisinde, sensörler 1. sensörden, 13. sensöre doğru sırayla, (-10, 80) metre konumundan, (-10, -280) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 14. sensörden, 26. sensöre kadar olan sensörler ise sırayla (42, -280) metreden, (42, 80) metreye kadar 30 metre aralıklarla yerleştirilmiştir.



Şekil 3.87. 26 adet sensör kullanılarak tasarlanan 6. sensör geometrisi

30 adet UGB sensör kullanılarak tasarlanan 6. sensör geometrisi

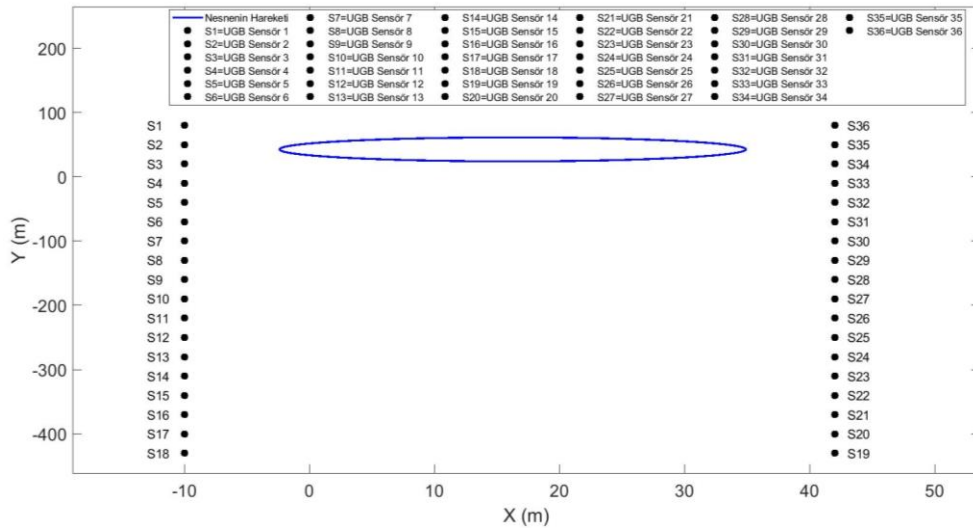
30 adet UGB sensör kullanılarak elde edilen 6. sensör geometrisinde, sensörler 1. sensörden, 15. sensöre doğru sırayla, (-10, 80) metre konumundan, (-10, -340) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 16. sensörden, 30. sensöre kadar olan sensörler ise sırayla (42, -340) metreden, (42, 80) metreye kadar 30 metre aralıklarla yerleştirilmiştir. 30 adet UGB sensör kullanılarak elde edilen 6. sensör geometrisi Şekil 3.88’de gösterilmiştir.



Şekil 3.88. 30 adet sensör kullanılarak tasarlanan 6. sensör geometrisi

36 adet UGB sensör kullanılarak tasarlanan 6. sensör geometrisi

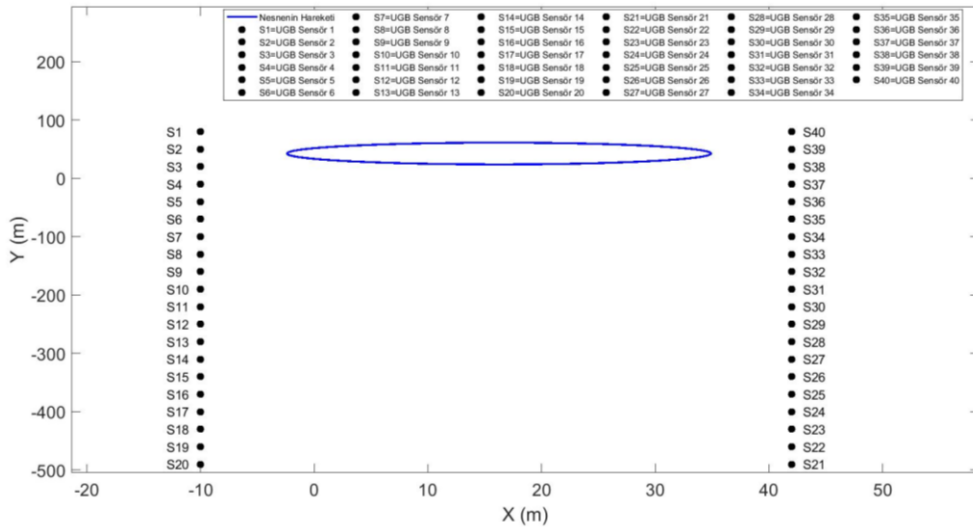
36 adet UGB sensör kullanılarak elde edilen 6. sensör geometrisinde, sensörler 1. sensörden, 18. sensöre doğru sırayla, (-10, 80) metre konumundan, (-10, -430) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 19. sensörden, 36. sensöre kadar olan sensörler ise sırayla (42, -430) metreden, (42, 80) metreye kadar 30 metre aralıklarla yerleştirilmiştir. 36 adet UGB sensör kullanılarak elde edilen 6. sensör geometrisi Şekil 3.89’da gösterilmiştir.



Şekil 3.89. 36 adet sensör kullanılarak tasarlanan 6. sensör geometrisi

40 adet UGB sensör kullanılarak tasarlanan 6. sensör geometrisi

Sensörler 1. sensörden, 20. sensöre doğru sırayla, (-10, 80) metre konumundan, (-10, -490) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 21. sensörden, 40. sensöre kadar olan sensörler ise sırayla (42, -490) metreden, (42, 80) metreye kadar 30 metre aralıklarla yerleştirilmiş olup, 6. sensör geometrisi Şekil 3.90'da gösterilmiştir.

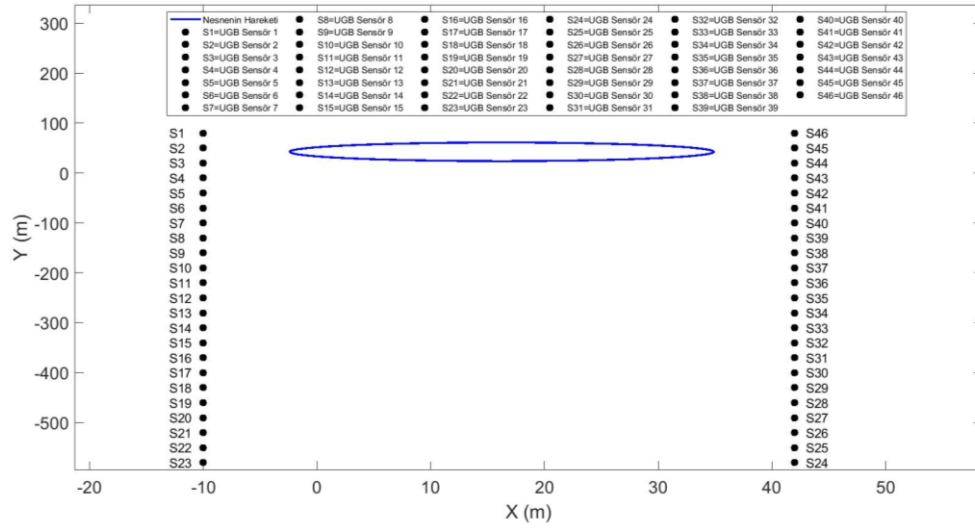


Şekil 3.90. 40 adet sensör kullanılarak tasarlanan 6. sensör geometrisi

46 adet UGB sensör kullanılarak tasarlanan 6. sensör geometrisi

46 adet sensör kullanılarak elde edilen 6. sensör geometrisinde, sensörler, 1. sensörden, 23. sensöre kadar (-10, 80) metre konumundan, (-10, -580) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 24. sensörden, 46. sensöre kadar olan sensörler ise sırasıyla (42, -580) metre konumundan, (42, 80) metre konumuna kadar 30 metre aralıklarla

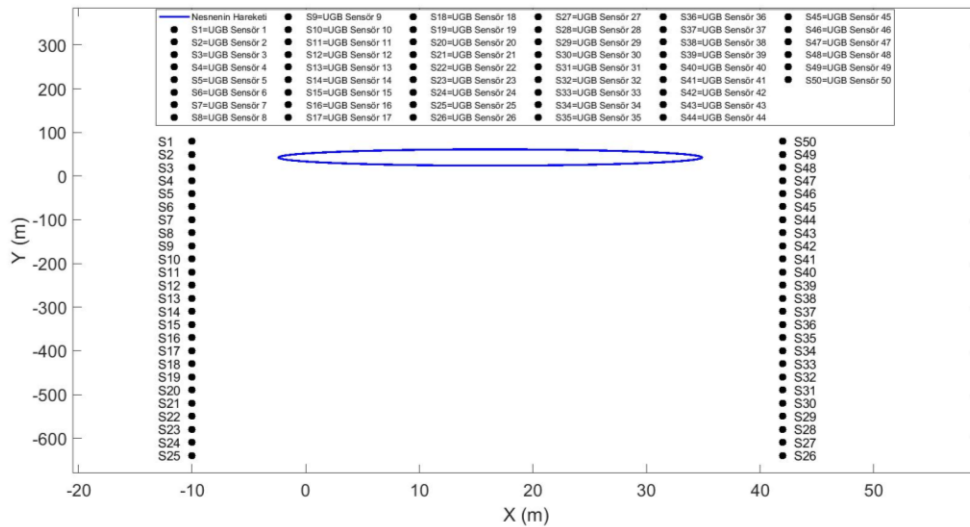
yerleştirilmiştir. 46 adet UGB sensör kullanılarak oluşturulan 6. sensör geometrisi Şekil 3.91’de gösterilmiştir.



Şekil 3.91. 46 adet sensör kullanılarak tasarlanan 6. sensör geometrisi

50 adet UGB sensör kullanılarak tasarlanan 6. sensör geometrisi

50 adet UGB sensör kullanılarak elde edilen 6. sensör geometrisi Şekil 3.92’de gösterilmiştir. Sensörler, 1. sensörden, 25. sensöre kadar (-10, 80) metre konumundan, (-10, -640) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir. 26. sensörden, 50. sensöre kadar olan sensörler ise sırasıyla (42, -640) metre konumundan, (42, 80) metre konumuna kadar 30 metre aralıklarla yerleştirilmiştir.



Şekil 3.92. 50 adet sensör kullanılarak tasarlanan 6. sensör geometrisi



4. BULGULAR ve TARTIŞMA

Tez çalışması kapsamında, GKF algoritmasının süreç gürültü varyansının ve sensörün ölçüm gürültüsü varyansının, GKF algoritmasının kestirim hatası üzerindeki etkileri incelenmiştir. Ayrıca, sensör geometrisinin ve kullanılan sensör sayısının GKF algoritmasının kestirim hatası üzerindeki etkileri incelenmiştir. Sensörlerin ölçüm gürültü varyans değerlerinin aynı ve farklı olduğu durumlar için sensör sayısının kestirim hatası üzerindeki etkileri ayrı ayrı incelenmiştir. Ayrıca nesnenin yaptığı dairesel ve karesel hareketlerin, GKF algoritmasının kestirim hatası üzerindeki etkileri incelenmiştir. Elde edilen sonuçlar, bulgular ve tartışma bölümünde 6 ayrı alt başlık altında incelenmiştir.

4.1. Genişletilmiş Kalman Filtre Algoritmasının Süreç Gürültüsü Varyansının Kestirim Hatası Üzerindeki Etkilerinin İncelenmesi

Genişletilmiş Kalman filtresinin süreç gürültüsü varyansının ve UGB sensörünün ölçüm gürültüsü varyansının, kestirim hatası üzerindeki etkilerinin incelenmesi için Şekil 4.1’de gösterilen senaryo kullanılmıştır. Ölçüm gürültüsü varyansı sabit tutulup, genişletilmiş Kalman filtresinin süreç gürültüsü varyansının değiştirilmesiyle, genişletilmiş Kalman filtresinin süreç gürültüsü varyansının, konum kestirim hatası üzerindeki etkileri incelenmiştir. Genişletilmiş Kalman filtresinin süreç gürültüsü varyansı sırasıyla $1 \left(\frac{m}{sn^2}\right)^2$, $10 \left(\frac{m}{sn^2}\right)^2$ ve $50 \left(\frac{m}{sn^2}\right)^2$ olarak alınmış ve 100 Monte Carlo simülasyonu yapılarak, GKF algoritmasının KOK konum kestirim hataları elde edilmiştir. UGB sensörünün ölçüm gürültüsü varyansı sırasıyla $0,5 m^2$, $1 m^2$, $5 m^2$ ve $15 m^2$ olarak alınmış ve her bir ölçüm gürültüsü varyansı için, filtrenin süreç gürültüsü varyansı artırılarak GKF algoritmasının KOK konum kestirim hataları elde edilmiştir.

Genişletilmiş Kalman filtresinin süreç gürültüsü varyansının seçimi için, konumu tespit edilmek istenen hedef nesnenin hareketindeki maksimum ivme miktarı kullanılmaktadır. Maksimum ivme miktarı hesaplanıp, Eş. 4.1 ile gösterilen aralık göz önüne alınarak, filtrenin süreç gürültüsünün standart sapma değeri seçilebilmektedir. Standart sapma değerinin karesi alınarak ise filtrenin süreç gürültüsü varyans değeri seçilebilmektedir (Bar-Shalom vd., 2001).

$$\frac{a_m}{2} \leq \sigma_v \leq a_m \quad (4.1)$$

Eş 4.1’de, σ_v değeri genişletilmiş Kalman filtresinin süreç gürültüsünün standart sapmasını ve a_m ise, hedef nesnenin hareketindeki maksimum ivmeyi ifade etmektedir. Nesnenin

hareketi esnasındaki maksimum ivme büyüklüğü (a_m) 0,142014 m/sn^2 olarak hesaplanmıştır. Elde edilen maksimum ivme değeri Eş. 4.1'de yerine koyularak, genişletilmiş Kalman filtresinin süreç gürültüsü standart sapma ve varyans aralıkları Eş. 4.2 ve Eş. 4.3'te verildiği gibi elde edilmiştir.

$$0,071 \leq \sigma_v \leq 0,142014 \quad (4.2)$$

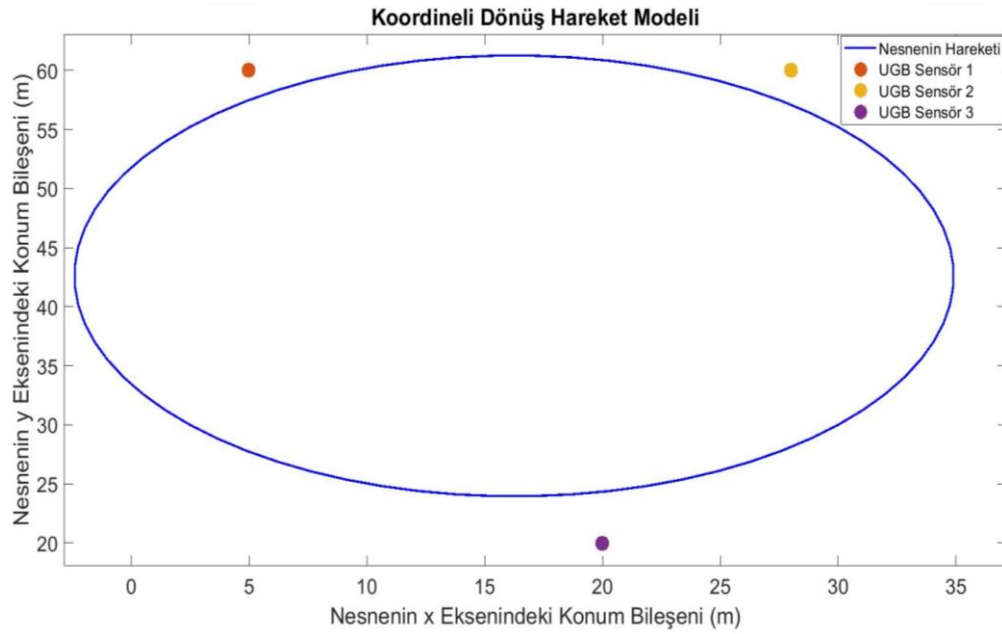
$$0,005 \leq \sigma_v^2 \leq 0,0202 \quad (4.3)$$

Filtrenin süreç gürültü varyansı değiştiğinde, filtrenin süreç gürültü kovaryansı değişmekte ve buna bağlı olarak ta durum öngörüm kovaryansı değişmektedir. Durum öngörüm kovaryansının değişmesiyle Kalman kazancı değiştiğinden kestirim değeri ve dolayısıyla kestirim hatası değişmektedir. Bu nedenle, süreç gürültü varyansı, kestirim hatasının değişmesinde rol oynamaktadır. 4.3 numaralı eşitsizlik incelendiğinde, filtrenin süreç gürültü varyansının aralığının ve bu aralıktan belirlenecek değerin çok küçük olduğu görülmektedir. Bu nedenle, bu aralıktan seçilecek varyans değerleri birbirlerine yakın olacakları için, filtrenin süreç gürültü kovaryansında fazla bir değişiklik yapmayacağından dolayı, durum öngörüm kovaryansı ve Kalman kazancında da fazla değişiklik olamayacaktır. Kalman kazancında fazla değişiklik olmadığı için kestirim değerleri ve kestirim hatalarında fazla değişiklik olmayacak ve filtrenin süreç gürültü varyansının kestirim hatası üzerindeki etkisi fazla görülemeyebilecektir. Bu nedenle, süreç gürültü varyans aralığını daha da büyütmek ve süreç gürültü varyansının filtre kestirimi üzerindeki etkisini daha iyi görebilmek için, filtrenin süreç gürültü varyansı $1 (\frac{m}{sn^2})^2$, $10 (\frac{m}{sn^2})^2$ ve $50 (\frac{m}{sn^2})^2$ olarak alınmıştır. Gerçekleştirilen simülasyonda, sensörler ile hedef nesne arasındaki uzaklık 40 metreye kadar çıkmaktadır. Genişletilmiş Kalman filtresinin süreç gürültüsü varyansının kestirim hatasına olan etkisini incelemek için GKF algoritması, 40 metrelik alanda, en fazla $7 \frac{m}{sn^2}$ standart sapmaya sahip süreç gürültüsü varlığında kestirim üretecek şekilde, filtrenin süreç gürültüsü varyansı $50 (\frac{m}{sn^2})^2$ 'ye kadar arttırılmıştır.

UGB sensörünün ölçüm gürültüsü varyans aralığı seçilirken, Eş. 4.4'te gösterilen, ölçüm gürültüsü standart sapma formülü kullanılmış ve ölçüm gürültüsü standart sapma değerleri bu formüle göre belirlenmiştir.

$$\sigma = \frac{c}{2B\sqrt{2SNR}} \quad (4.4)$$

Eş. 4.4'te, C ışık hızı, B UGB sinyalinin bant genişliği, SNR sinyal gürültü oranı ve σ ise ölçüm gürültü standart sapma değeridir. Ölçüm gürültüsünün standart sapma değeri hesaplanırken, ışık hızı 3.10^8 m/sn , sinyalin bant genişliği, UGB sensörün veri setinden 500 MHz ve SNR değeri ise 21 dB olarak alınmıştır. Eş. 4.4 kullanılarak yapılan hesaplamada, ölçüm gürültüsü standart sapma değeri 0,0463 metre olarak elde edilmiştir. Ancak, sensörler ile 40 metreye kadar olan uzaklıkta ölçüm alındığı için, bu hesaplamada elde edilen ölçüm gürültü standart sapma değeri çok küçük çıkmıştır. Bu nedenle, farklı ölçüm gürültü varyanslarının, GKF algoritmasının konum kestirim hatası üzerindeki etkileri incelenirken, ölçüm gürültüsü standart sapma değerleri daha yüksek olacak şekilde, ölçüm gürültüsü varyansları $0,5 \text{ m}^2$, 1 m^2 , 5 m^2 ve 15 m^2 olarak modellenmiştir.

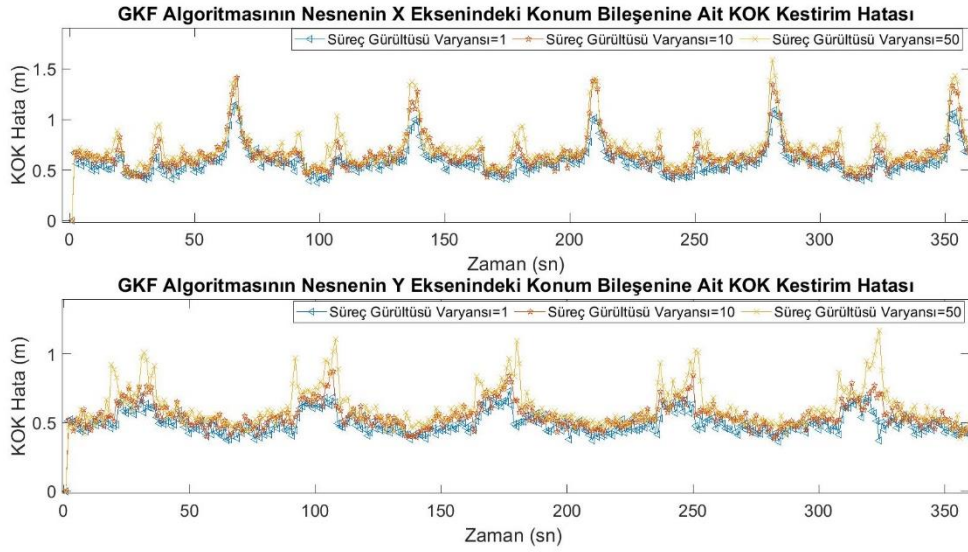


Şekil 4.1. Hedef nesnenin hareket modeli ve UGB sensörlerinin konumları

Şekil 4.1'de, 1. sensör (5, 60) metre, 2. sensör (28, 60) metre ve 3. sensör ise (20, 20) metre konumlarında bulunmaktadır. Gerçekleştirilen senaryoda, nesnenin başlangıçta bulunduğu konum koordinatı ise (30, 30) metredir. Nesne, $(1,1;1,1) \text{ m/sn}$ sabit hız ve $0,0873 \text{ radyan / saniye}$ açısal hız ile 360 saniye boyunca koordineli dönüş hareketi yapmaktadır.

4.1.1. UGB sensörünün ölçüm gürültüsü varyansı $0,5 \text{ m}^2$ iken GKF algoritmasının farklı süreç gürültüsü varyans değerleri için elde edilen konum kestirim hataları

Ölçüm gürültüsü varyansı $0,5 \text{ m}^2$ olduğunda ve genişletilmiş Kalman filtresinin süreç gürültüsü varyans değerleri $1 \left(\frac{\text{m}}{\text{sn}^2}\right)^2$, $10 \left(\frac{\text{m}}{\text{sn}^2}\right)^2$ ve $50 \left(\frac{\text{m}}{\text{sn}^2}\right)^2$ olarak alındığında, 100 Monte Carlo simülasyonu yapılarak elde edilen x ve y eksenlerindeki konum kestirim hataları, Şekil 4.2'de gösterilmiştir.

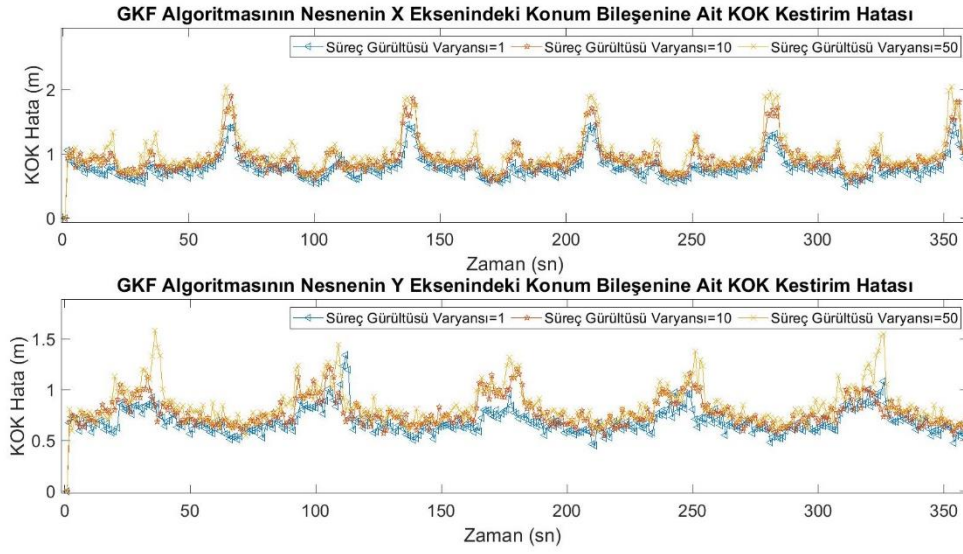


Şekil 4.2. UGB sensörünün ölçüm gürültüsü varyansı $0,5 \text{ m}^2$ iken farklı GKF süreç gürültüsü varyans değerleri için 100 Monte Carlo simülasyonu yapılarak elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

Ölçüm gürültüsü varyansı $0,5 \text{ m}^2$ ve genişletilmiş Kalman filtresinin süreç gürültüsü varyansları $1 \left(\frac{\text{m}}{\text{sn}^2}\right)^2$, $10 \left(\frac{\text{m}}{\text{sn}^2}\right)^2$ ve $50 \left(\frac{\text{m}}{\text{sn}^2}\right)^2$ iken, GKF algoritmasının, x eksenindeki ortalama KOK konum kestirim hataları sırasıyla 0,5838 metre, 0,6661 metre ve 0,7031 metre olarak elde edilmiştir. Ölçüm gürültüsü varyansı $0,5 \text{ m}^2$ ve genişletilmiş Kalman filtresinin süreç gürültüsü varyansları $1 \left(\frac{\text{m}}{\text{sn}^2}\right)^2$, $10 \left(\frac{\text{m}}{\text{sn}^2}\right)^2$ ve $50 \left(\frac{\text{m}}{\text{sn}^2}\right)^2$ iken, GKF algoritmasının, y eksenindeki ortalama KOK konum kestirim hataları sırasıyla 0,4962 metre, 0,5588 metre ve 0,6002 metre olarak elde edilmiştir.

4.1.2. UGB sensörünün ölçüm gürültüsü varyansı 1 m^2 iken GKF algoritmasının farklı süreç gürültüsü varyans değerleri için elde edilen konum kestirim hataları

Şekil 4.3'te, UGB sensörünün ölçüm gürültüsü varyansı 1 m^2 iken, genişletilmiş Kalman filtresinin farklı süreç gürültüsü varyans değerleri için, 100 Monte Carlo simülasyonu yapılarak elde edilen x ve y eksenlerindeki KOK konum kestirim hataları gösterilmiştir.

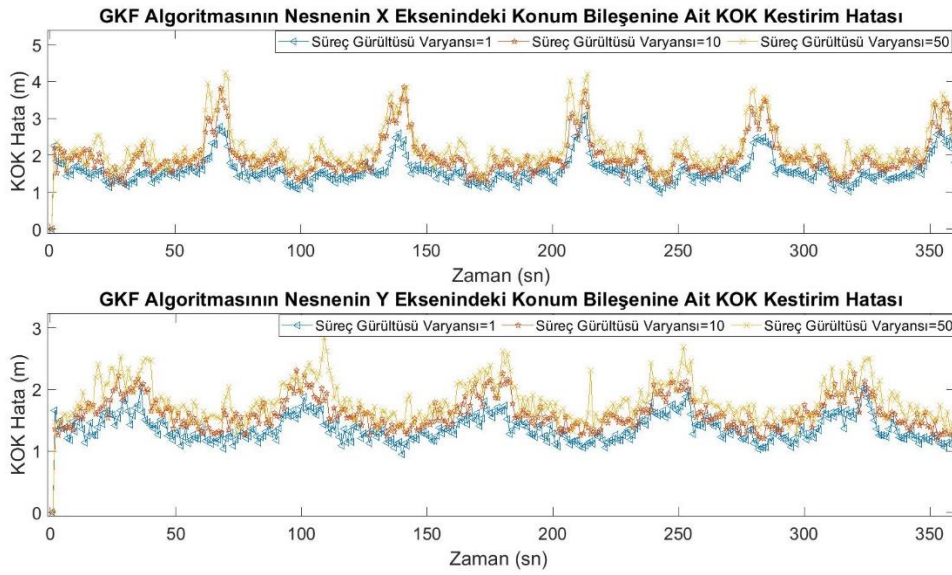


Şekil 4.3. UGB sensörünün ölçüm gürültüsü varyansı 1 m^2 iken farklı GKF süreç gürültüsü varyans değerleri için, 100 Monte Carlo simülasyonu yapılarak elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

Ölçüm gürültüsü varyansı 1 m^2 ve genişletilmiş Kalman filtresinin süreç gürültüsü varyansı $1 \left(\frac{\text{m}}{\text{sn}^2}\right)^2$ iken, GKF algoritmasının, x eksenindeki ortalama KOK konum kestirim hatası 0,7970 metre olarak elde edilmiştir. Genişletilmiş Kalman filtresinin süreç gürültüsü varyansı $10 \left(\frac{\text{m}}{\text{sn}^2}\right)^2$ iken, x eksenindeki ortalama KOK konum kestirim hatası 0,9144 metre ve $50 \left(\frac{\text{m}}{\text{sn}^2}\right)^2$ iken ise 0,9980 metre olarak elde edilmiştir. UGB sensörünün ölçüm gürültüsü varyansı 1 m^2 ve genişletilmiş Kalman filtresinin süreç gürültüsü varyansı $1 \left(\frac{\text{m}}{\text{sn}^2}\right)^2$ iken, y eksenindeki ortalama KOK konum kestirim hatası 0,6759 metre olarak elde edilmiştir. Genişletilmiş Kalman filtresinin süreç gürültüsü varyansı $10 \left(\frac{\text{m}}{\text{sn}^2}\right)^2$ iken, y eksenindeki ortalama KOK konum kestirim hatası 0,7734 metre ve $50 \left(\frac{\text{m}}{\text{sn}^2}\right)^2$ iken ise 0,8404 metre olarak elde edilmiştir.

4.1.3. UGB sensörünün ölçüm gürültüsü varyansı 5 m^2 iken GKF algoritmasının farklı süreç gürültüsü varyans değerleri için elde edilen konum kestirim hataları

UGB sensörünün ölçüm gürültüsü varyansı 5 m^2 iken ve genişletilmiş Kalman filtresinin süreç gürültüsü varyansı $1 \left(\frac{\text{m}}{\text{sn}^2}\right)^2$, $10 \left(\frac{\text{m}}{\text{sn}^2}\right)^2$ ve $50 \left(\frac{\text{m}}{\text{sn}^2}\right)^2$ olduğunda, 100 Monte Carlo simülasyonu yapılarak elde edilen, GKF algoritmasının x ve y eksenlerindeki bileşenlerine ait konum kestirim hataları Şekil 4.4'te gösterilmiştir.

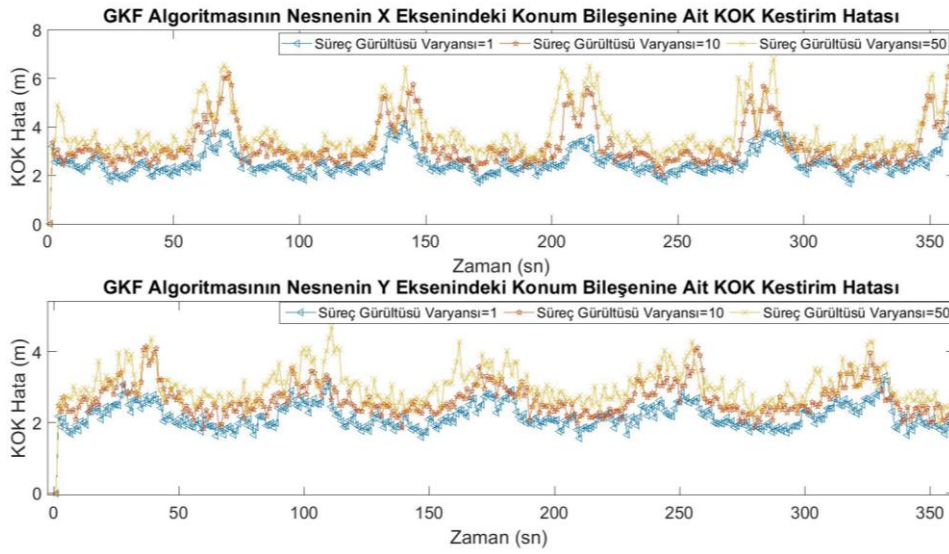


Şekil 4.4. UGB sensörünün ölçüm gürültüsü varyansı 5 m^2 iken farklı GKF süreç gürültüsü varyans değerleri için 100 Monte Carlo simülasyonu yapılarak elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

UGB sensörünün ölçüm gürültüsü varyansı 5 m^2 ve genişletilmiş Kalman filtresinin süreç gürültüsü varyansı $1 \left(\frac{\text{m}}{\text{sn}^2}\right)^2$ iken, x eksenindeki ortalama KOK konum kestirim hatası 1,5930 metre olarak elde edilmiştir. Genişletilmiş Kalman filtresinin süreç gürültüsü varyansı 10 $\left(\frac{\text{m}}{\text{sn}^2}\right)^2$ iken 1,9328 metre ve 50 $\left(\frac{\text{m}}{\text{sn}^2}\right)^2$ iken ise 2,2140 metre olarak elde edilmiştir. UGB sensörünün ölçüm gürültüsü varyansı 5 m^2 ve genişletilmiş Kalman filtresinin süreç gürültüsü varyansları sırasıyla 1 $\left(\frac{\text{m}}{\text{sn}^2}\right)^2$ iken, 10 $\left(\frac{\text{m}}{\text{sn}^2}\right)^2$ ve 50 $\left(\frac{\text{m}}{\text{sn}^2}\right)^2$ iken, y eksenindeki ortalama KOK konum kestirim hataları sırasıyla 1,3711 metre, 1,6156 metre ve 1,8029 metre olarak elde edilmiştir.

4.1.4. UGB sensörünün ölçüm gürültüsü varyansı 15 m^2 iken GKF algoritmasının farklı süreç gürültüsü varyans değerleri için elde edilen konum kestirim hataları

UGB sensörünün ölçüm gürültüsü varyansı 15 m^2 iken ve genişletilmiş Kalman filtresinin farklı süreç gürültüsü varyans değerleri için elde edilen, GKF algoritmasının x ve y eksenlerindeki konum kestirim hataları, Şekil 4.5'te gösterilmiştir.



Şekil 4.5. UGB sensörünün ölçüm gürültüsü varyansı 15 m^2 iken farklı GKF süreç gürültüsü varyans değerleri için 100 Monte Carlo simülasyonu yapılarak elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

Ölçüm gürültüsü varyansı 15 m^2 ve genişletilmiş Kalman filtresinin süreç gürültüsü varyansı $1 \left(\frac{\text{m}}{\text{sn}^2}\right)^2$ iken, GKF algoritmasının, x eksenindeki ortalama KOK konum kestirim hatası 2,5394 metre olarak elde edilmiştir. Genişletilmiş Kalman filtresinin süreç gürültüsü varyansı $10 \left(\frac{\text{m}}{\text{sn}^2}\right)^2$ iken, x eksenindeki ortalama KOK konum kestirim hatası 3,2545 metre ve filtrenin süreç gürültüsü varyansı $50 \left(\frac{\text{m}}{\text{sn}^2}\right)^2$ iken ise 3,7327 metre olarak elde edilmiştir. UGB sensörünün ölçüm gürültüsü varyansı 15 m^2 ve genişletilmiş Kalman filtresinin süreç gürültüsü varyansı $1 \left(\frac{\text{m}}{\text{sn}^2}\right)^2$ iken, y eksenindeki ortalama KOK konum kestirim hatası 2,1734 metre, filtrenin süreç gürültüsü varyansı $10 \left(\frac{\text{m}}{\text{sn}^2}\right)^2$ iken ise 2,6456 metre olarak elde edilmiştir. Filtrenin süreç gürültüsü varyansı $50 \left(\frac{\text{m}}{\text{sn}^2}\right)^2$ iken ise y eksenindeki ortalama KOK konum kestirim hatası 3,0179 metre olarak elde edilmiştir.

Çizelge 4.1. Genişletilmiş Kalman filtresinin farklı süreç gürültüsü varyansı değerleri ve UGB sensörünün farklı ölçüm gürültüsü varyans değerleri kullanıldığında, 100 Monte Carlo simülasyonu yapılarak elde edilen, ortalama KOK konum kestirim hataları

Genişletilmiş Kalman Filtresinin Süreç Gürültüsü Varyansı $((\frac{m}{sn^2})^2)$	Ölçüm Gürültüsü Varyansı (m^2)	GKF Algoritmasının X Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Y Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)
1	0,5	0,5838	0,4962
10	0,5	0,6661	0,5588
50	0,5	0,7031	0,6002
1	1	0,7970	0,6759
10	1	0,9144	0,7734
50	1	0,9980	0,8404
1	5	1,5930	1,3711
10	5	1,9328	1,6156
50	5	2,2140	1,8029
1	15	2,5394	2,1734
10	15	3,2545	2,6456
50	15	3,7327	3,0179

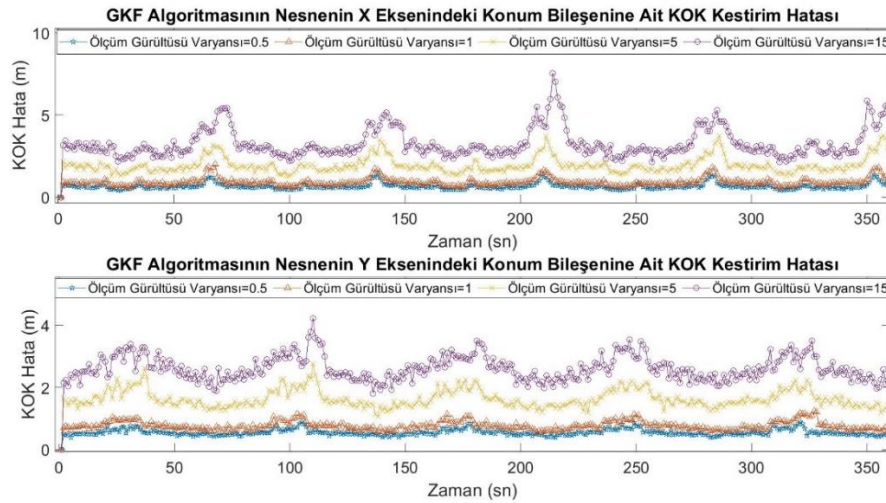
UGB sensörünün ölçüm gürültü varyansı sabit tutulup, genişletilmiş Kalman filtresinin süreç gürültüsü varyansının değiştirilmesiyle, filtrenin süreç gürültüsü varyansının, GKF algoritmasının konum kestirim hatası üzerindeki etkileri incelenmiştir. Çizelge 4.1 incelendiğinde, ölçüm gürültüsü varyansı sabit tutularak, filtrenin süreç gürültüsü varyansı artırıldığında, GKF algoritmasının ortalama KOK konum kestirim hatalarının arttığı görülmektedir. Filtrenin süreç gürültüsü varyansı arttıkça, filtrenin süreç gürültüsü kovaryansı artmakta ve bu nedenle durum öngörüm kovaryansı artmaktadır. Durum öngörüm kovaryansı arttığı için Kalman kazancı artmakta ve GKF algoritması sensör ölçümüne daha çok ağırlık vererek kestirim üretmektedir. Ancak, sensör ölçümü, ölçüm gürültüsü içermekte ve bu gürültü ise ölçüm hatasına sebep olmaktadır. Bu nedenle, kazanç arttığında, ölçüm gürültüsü GKF algoritmasının kestirimine daha çok etki ettiğinden, ölçüm gürültüsünün neden olduğu ölçüm hatasından dolayı, kestirim hatası da artmaktadır.

4.2. UGB Sensörünün Ölçüm Gürültüsü Varyansının Genişletilmiş Kalman Filtre Algoritmasının Konum Kestirim Hatası Üzerindeki Etkilerinin İncelenmesi

Ölçüm gürültüsü varyansının, konum kestirim hatası üzerindeki etkilerinin incelenmesi için Şekil 4.1’de gösterilen senaryo kullanılmış ve genişletilmiş Kalman filtresinin süreç gürültüsü varyansı sabit tutulup, ölçüm gürültü varyansı değiştirilerek, GKF algoritmasının KOK konum kestirim hataları elde edilmiştir. GKF süreç gürültü varyansı sırasıyla $1 \left(\frac{m}{sn^2}\right)^2$, $10 \left(\frac{m}{sn^2}\right)^2$ ve $50 \left(\frac{m}{sn^2}\right)^2$ alındığında ve genişletilmiş Kalman filtresinin her bir süreç gürültüsü varyans değerleri için, ölçüm gürültüsü varyansı $0,5 m^2$, $1 m^2$, $5 m^2$ ve $15 m^2$ olduğu durumlar için, KOK konum kestirim hataları, 100 Monte Carlo simülasyonu yapılarak elde edilmiştir.

4.2.1. GKF algoritmasının süreç gürültüsü varyansı $1 \left(\frac{m}{sn^2}\right)^2$ iken elde edilen konum kestirim hataları

Genişletilmiş Kalman filtresinin süreç gürültü varyansı $1 \left(\frac{m}{sn^2}\right)^2$ olduğunda ve ölçüm gürültüsü varyansı $0,5 m^2$, $1 m^2$, $5 m^2$ ve $15 m^2$ olarak alındığında elde edilen x ve y eksenlerindeki KOK konum kestirim hataları, Şekil 4.6’da gösterilmiştir.



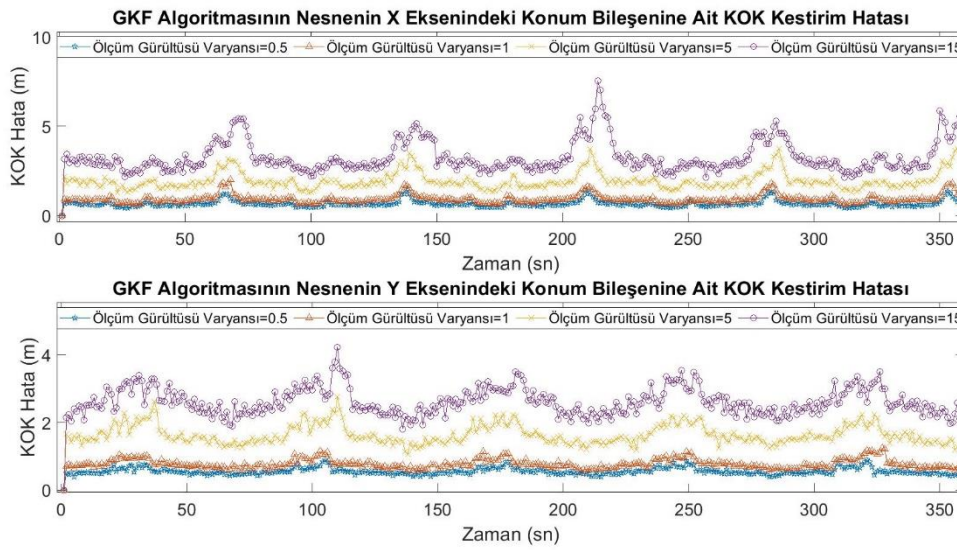
Şekil 4.6. Genişletilmiş Kalman filtresinin süreç gürültüsü varyansı $1 \left(\frac{m}{sn^2}\right)^2$ iken UGB sensörünün farklı ölçüm gürültüsü varyans değerleri için elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

Ölçüm gürültüsü varyansı $0,5 m^2$, $1 m^2$, $5 m^2$ ve $15 m^2$ iken, x eksenindeki ortalama KOK konum kestirim hataları sırasıyla 0,5837 metre, 0,7949 metre, 1,5987 metre ve 2,5853 metre olarak elde edilmiştir. Ölçüm gürültüsü varyansı $0,5 m^2$, $1 m^2$, $5 m^2$ ve $15 m^2$ iken y

eksenindeki ortalama KOK konum kestirim hataları ise sırasıyla 0,4958 metre, 0,6752 metre, 1,3853 metre ve 2,1627 metre olarak elde edilmiştir.

4.2.2. GKF algoritmasının süreç gürültüsü varyansı $10 \left(\frac{m}{sn^2}\right)^2$ iken elde edilen konum kestirim hataları

Genişletilmiş Kalman filtresinin süreç gürültü varyansı $10 \left(\frac{m}{sn^2}\right)^2$ olduğunda ve ölçüm gürültüsü varyansı $0,5 m^2$, $1 m^2$, $5 m^2$ ve $15 m^2$ olarak alındığında elde edilen x ve y eksenlerindeki konum kestirim hataları Şekil 4.7’de gösterilmiştir.

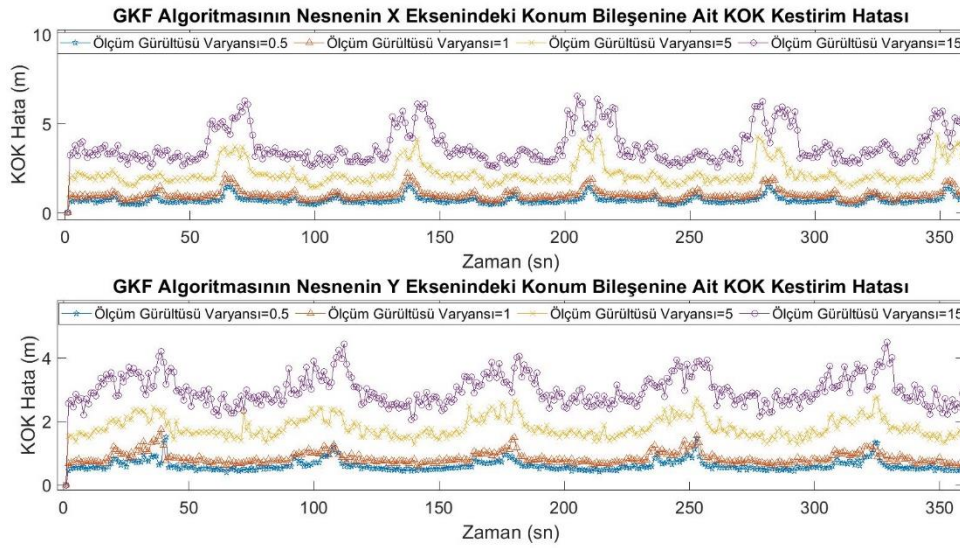


Şekil 4.7. Genişletilmiş Kalman filtresinin süreç gürültüsü varyansı $10 \left(\frac{m}{sn^2}\right)^2$ iken UGB sensörünün farklı ölçüm gürültüsü varyans değerleri için elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

Ölçüm gürültüsü varyansı $0,5 m^2$, $1 m^2$, $5 m^2$ ve $15 m^2$ iken GKF algoritmasının x eksenindeki ortalama KOK konum kestirim hataları sırasıyla 0,6590 metre, 0,9110 metre, 1,9405 metre ve 3,2763 metre olarak elde edilmiştir. Ölçüm gürültüsü varyansı $0,5 m^2$, $1 m^2$, $5 m^2$ ve $15 m^2$ iken y eksenindeki ortalama KOK konum kestirim hataları sırasıyla 0,5568 metre, 0,7661 metre, 1,6060 metre ve 2,6381 metre olarak elde edilmiştir.

4.2.3. GKF algoritmasının süreç gürültüsü varyansı $50 \left(\frac{m}{sn^2}\right)^2$ iken elde edilen konum kestirim hataları

Genişletilmiş Kalman filtresinin süreç gürültü varyansı $50 \left(\frac{m}{sn^2}\right)^2$ olduğunda ve ölçüm gürültüsü varyansı $0,5 m^2$, $1 m^2$, $5 m^2$ ve $15 m^2$ olarak alındığında elde edilen, GKF algoritmasının x ve y eksenlerindeki konum kestirim hataları Şekil 4.8’de gösterilmiştir.



Şekil 4.8. Genişletilmiş Kalman filtresinin süreç gürültüsü varyansı $50 \left(\frac{m}{sn^2}\right)^2$ iken UGB sensörünün farklı ölçüm gürültüsü varyans değerleri için elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

Ölçüm gürültüsü varyansı $0,5 m^2$, $1 m^2$, $5 m^2$ ve $15 m^2$ iken, x eksenindeki ortalama KOK konum kestirim hataları sırasıyla 0,7043 metre, 0,9915 metre, 2,2082 metre ve 3,7374 metre ve y eksenindeyse 0,6107 metre, 0,8467 metre, 1,8032 metre ve 2,9966 metredir.

Çizelge 4.2. Genişletilmiş Kalman filtresinin farklı süreç gürültüsü varyans değerleri ve farklı ölçüm gürültüsü varyans değerleri için elde edilen ortalama KOK konum kestirim hataları

Genişletilmiş Kalman Filtresinin Süreç Gürültüsü Varyansı $\left(\frac{m}{sn^2}\right)^2$	Ölçüm Gürültüsü Varyansı (m^2)	GKF Algoritmasının X Eksenindeki Ortalama KOK Kestirim Hatası (m)	GKF Algoritmasının Y Eksenindeki Ortalama KOK Kestirim Hatası (m)
1	0,5	0,5837	0,4958
1	1	0,7949	0,6752
1	5	1,5987	1,3853
1	15	2,5853	2,1627
10	0,5	0,6590	0,5568
10	1	0,9110	0,7661
10	5	1,9405	1,6060
10	15	3,2763	2,6381
50	0,5	0,7043	0,6107
50	1	0,9915	0,8467
50	5	2,2082	1,8032
50	15	3,7374	2,9966

Çizelge 4.2 incelendiğinde, filtrenin süreç gürültüsü varyansı sabit tutularak, ölçüm gürültüsü varyansı artırıldığında, ortalama KOK konum kestirim hatasının arttığı görülmektedir. Bunun nedeni, filtrenin süreç gürültü varyansı arttıkça, durum öngörüm kovaryansı artmakta ve bu nedenle de Kalman kazancı arttığından, sensör ölçümüne daha çok ağırlık verilerek kestirim üretilmektedir. Sensör ölçümü ise, ölçüm gürültüsü içermekte ve bu gürültü ölçüm hatasına sebep olmaktadır. Bu nedenle, kazanç arttığında, sensörün ölçüm hatası, kestirime daha çok etki etmiş ve ölçüm hatasından kaynaklanan kestirim hatası artmıştır.

Genişletilmiş Kalman filtresinin, süreç gürültüsü varyansı artırıldığında, ölçüm gürültüsü varyans değerlerinin $0,5 m^2$, $1 m^2$, $5 m^2$ ve $15 m^2$ olduğu durumlar için elde edilen konum kestirim hata değerleri birbirlerine yaklaşmaktadır. Bunun sebebi, GKF algoritmasının, bazı eşitliklerinin kullanılmasıyla açıklanabilir.

$$Q = \begin{bmatrix} \frac{T^4}{4} & \frac{T^3}{2} & 0 & 0 \\ \frac{T^3}{2} & T^2 & 0 & 0 \\ 0 & 0 & \frac{T^4}{4} & \frac{T^3}{2} \\ 0 & 0 & \frac{T^3}{2} & T^2 \end{bmatrix} \sigma_v^2 \quad (4.5)$$

$$\mathbf{P}(k+1|k) = \mathbf{F}\mathbf{P}(k|k)\mathbf{F}' + \mathbf{Q}(k) \quad (4.6)$$

$$\mathbf{W}(k+1) = \mathbf{P}(k+1|k)\mathbf{H}(k+1)'\mathbf{S}(k+1)^{-1} \quad (4.7)$$

$$\mathbf{V}(k+1) = \mathbf{Z}(k+1) - \hat{\mathbf{Z}}(k+1|k) \quad (4.8)$$

$$\hat{\mathbf{X}}(k+1|k+1) = \hat{\mathbf{X}}(k+1|k) + \mathbf{W}(k+1)\mathbf{V}(k+1) \quad (4.9)$$

Eş 4.5 ile genişletilmiş Kalman filtresinin süreç gürültüsü kovaryans eşitliği verilmiştir. Eş 4.5'te σ_v^2 ifadesi, genişletilmiş Kalman filtresinin süreç gürültüsü varyansı ve T ise örnekleme aralığını belirtmektedir. T ifadesi sabit bir değer olduğundan, genişletilmiş Kalman filtresinin süreç gürültüsü kovaryansını değiştirmek için süreç gürültüsü varyansını değiştirmek gerekmektedir. Eş 4.5 incelendiğinde, genişletilmiş Kalman filtresinin süreç gürültüsü varyansı artırıldığında, süreç gürültüsü kovaryansı artmaktadır. Eş 4.6'da $\mathbf{P}(k+1|k)$ ifadesi, durum öngörüm kovaryansı ve $\mathbf{Q}$ ise genişletilmiş Kalman filtresinin süreç gürültüsü kovaryansıdır. Eş 4.6 incelendiğinde, genişletilmiş Kalman filtresinin süreç

gürültüsü kovaryansı arttığında ise, durum öngörüm kovaryansı artmakta ve durum öngörüm kovaryansı arttığında, Kalman kazancı artmaktadır. Eş 4.8’de ise V ifadesi inovasyonu belirtmekte olup, sensör ölçümü ile GKF algoritmasının o ölçüme ait ölçüm öngörümü arasındaki farka eşittir. Eş 4.9 incelendiğinde, Kalman kazancı arttığında, GKF algoritması kestirim üretmek için, sensör ölçümünden gelen bilgiye daha çok ağırlık vererek kestirim ürettiğinden kestirim değerleri, sensörün ölçüm değerlerine yakın çıkmaktadır. Bu nedenle, kestirim hata değerleri de ölçüm hatalarına yaklaşmaktadır.

4.3. Sensör Geometrisinin Genişletilmiş Kalman Filtre Algoritmasının Konum Kestirim Hatası Üzerindeki Etkilerinin İncelenmesi

Sensör geometrisinin, GKF algoritmasının kestirim hatası üzerindeki etkilerini incelemek için farklı sensör geometrileri oluşturulmuştur. Her bir sensör geometrisi için, sensörlerin ölçüm gürültüsü varyans değeri $0,5 \text{ m}^2$ ve tasarlanan GKF algoritmasının süreç gürültüsü varyans değeri ise $1 \left(\frac{m}{sn^2}\right)^2$ olarak alınmıştır. Sensör geometrisinin, GKF algoritmasının, kestirim hatasını neden etkilediği aşağıda verilen GKF algoritmasının eşitlikleri kullanılarak açıklanmıştır.

Tez çalışması kapsamında, doğrusal olmayan ölçüm fonksiyonunun doğrusallaştırılması için H Jakobyen matrisi kullanılmıştır. Jakobyen matrisi sensörün ölçüm fonksiyonunun, sistemin durum değişkeni olan X durum vektörüne göre kısmi türevi alınarak hesaplanmıştır. X durum vektörü, nesnenin hareketi esnasında bulunduğu konum ve hızının, 2 boyutlu kartezyen koordinat sistemi üzerinde x ve y eksenlerindeki bileşenlerinden oluşmaktadır.

$$X = \begin{bmatrix} x_{\text{pos}} \\ x_{\text{hız}} \\ y_{\text{pos}} \\ y_{\text{hız}} \end{bmatrix} \quad (4.13)$$

$$\hat{X} = \begin{bmatrix} x_{\text{pos}}' \\ x_{\text{hız}}' \\ y_{\text{pos}}' \\ y_{\text{hız}}' \end{bmatrix} \quad (4.14)$$

$$H = \frac{\partial h}{\partial x} \big|_{x=\hat{x}} = [H_{11} \ H_{12} \ H_{13} \ H_{14}] \quad (4.15)$$

$$H_{11} = \frac{\partial h}{\partial x_{\text{pos}}} \big|_{x_{\text{pos}}=x_{\text{pos}}'} = \frac{x_{\text{pos}}' - R_{0x}}{\sqrt{(x_{\text{pos}}' - R_{0x})^2 + (y_{\text{pos}} - R_{0y})^2}} \quad (4.16)$$

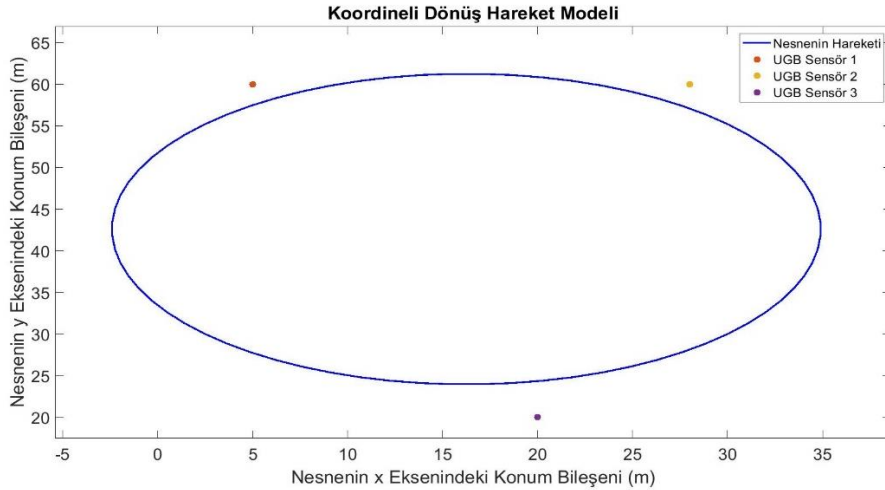
$$H_{12} = \frac{\partial h}{\partial x_{hiz}} \big|_{x_{hiz}=x_{hiz}'} = 0 \quad (4.17)$$

$$H_{13} = \frac{\partial h}{\partial y_{pos}} \big|_{y_{pos}=y_{pos}'} = \frac{y_{pos}' - R_{oy}}{\sqrt{(x_{pos} - R_{ox})^2 + (y_{pos}' - R_{oy})^2}} \quad (4.18)$$

$$H_{14} = \frac{\partial h}{\partial y_{hiz}} \big|_{y_{hiz}=y_{hiz}'} = 0 \quad (4.19)$$

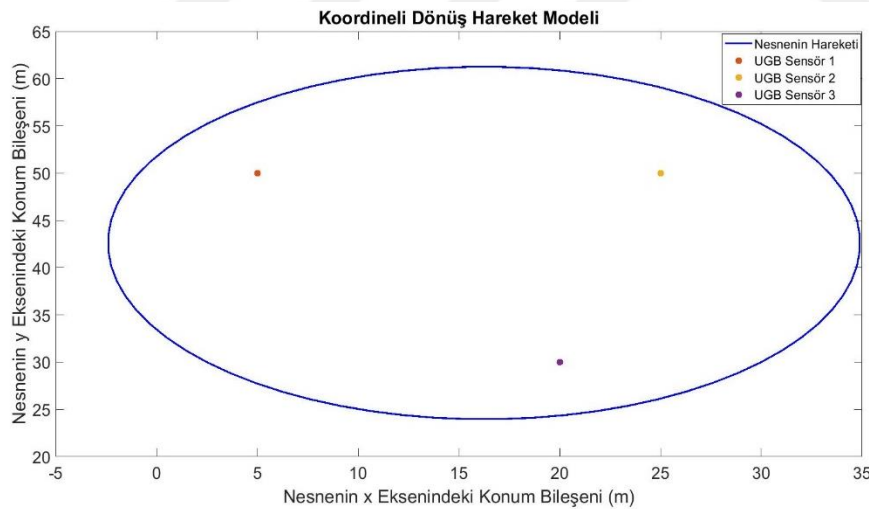
Yukarıda verilen eşitliklerdeki x_{pos} ve y_{pos} ifadeleri sırasıyla, konumu tespit edilmek istenen nesnenin konumunun, 2 boyutlu kartezyen koordinat sistemi üzerindeki x ve y eksenlerindeki bileşenlerini ifade ederken x_{hiz} ve y_{hiz} ifadeleri ise sırasıyla x ve y eksenlerindeki hız bileşenleridir. Yukarıda verilen eşitliklerde x_{pos}' ve y_{pos}' ifadeleri ise sırasıyla, nesnenin hareketi esnasında bulunduğu konumun x ve y eksenlerindeki bileşenlerinin öngörüm değerleridir. Yukarıda verilen eşitliklerde bulunan R_{ox} ve R_{oy} ifadeleri ise sensör konumunun, 2 boyutlu kartezyen koordinat sistemi üzerinde sırasıyla x ve y eksenlerindeki bileşenleridir. Eş. 4.15'te gösterilen H ise Jakobyen matrisi ifade etmektedir.

Tez çalışması kapsamında, sensör geometrisinin, GKF algoritmasının kestirim hatası üzerindeki etkilerinin incelenmesi için farklı sensör geometrileri kullanılmıştır. Bu sensör geometrilerini elde etmek için, sensör geometrilerinde bulunan sensörlerin yerleri değiştirilmiştir. Sensörlerin yerleri değiştirildiğinde, yukarıda Eş. 4.16 ve Eş. 4.18 numaralı eşitliklerde verilen R_{ox} ve R_{oy} değerleri değiştiği için, Jakobyen matrisi değişmektedir. Jakobyen matrisi değiştiğinde ise, Eş. 4.10'da verilen Kalman kazancı değişmekte ve kazanç değiştiği için de Eş. 4.7'de verilen, GKF algoritmasının kestirim değeri değişmektedir. Bu nedenle, sensör geometrisi, GKF algoritmasının kestirim hatasını etkilemektedir. Sensör geometrisinin GKF algoritmasının kestirim hatası üzerindeki etkisinin incelenmesi için tasarlanan ve bölüm 4.3.1'de verilen sensör geometrileri haricinde, sensörlerin nesne hareketinin içinde ve dışında kaldığı farklı sensör geometrileri kullanılarak da sensör geometrisinin, kestirim hatası üzerindeki etkisi incelenmiştir. Bu amaçla oluşturulan sensör geometrileri Şekil 4.9 ve Şekil 4.10'da gösterilmiştir.



Şekil 4.9. Sensörler, nesne hareketinin dışına yerleştirilerek elde edilen senaryo ve UGB sensörlerinin konumları

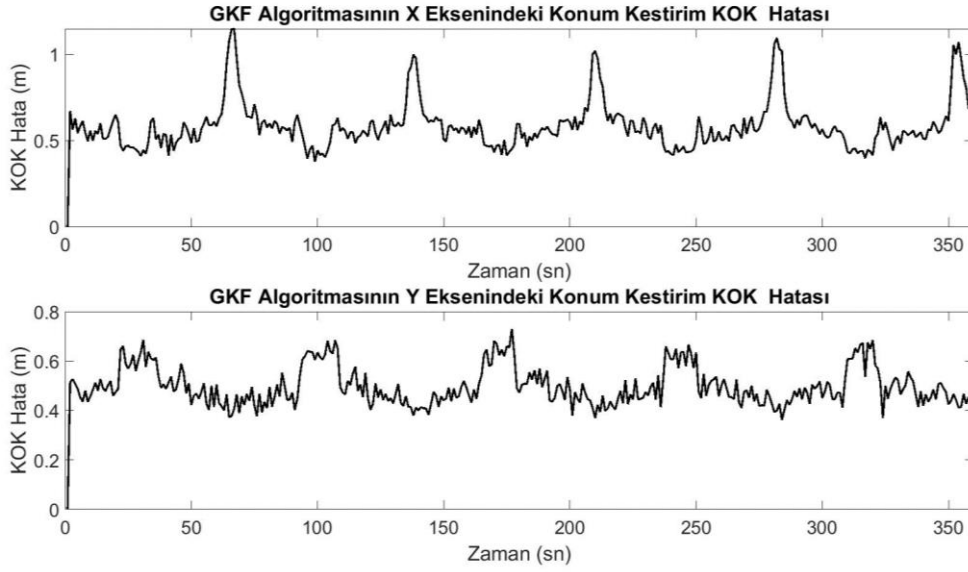
Şekil 4.9’da, 1. sensör (5, 60) metre, 2. sensör (28, 60) metre ve 3. sensör ise (20, 20) metre konumunda bulunmaktadır. Gerçekleştirilen senaryoda, nesne (30, 30) metre konumundan harekete başlayıp, $(1,1; 1,1)$ m/sn sabit hız ve $0,0873$ $radyan/saniye$ açısal hız ile 360 saniye boyunca koordineli dönüş hareketi yapmakta ve sensör geometrisinin içinde hareket etmektedir.



Şekil 4.10. Sensörler, nesne hareketinin içine yerleştirilerek elde edilen senaryo ve UGB sensörlerinin konumları

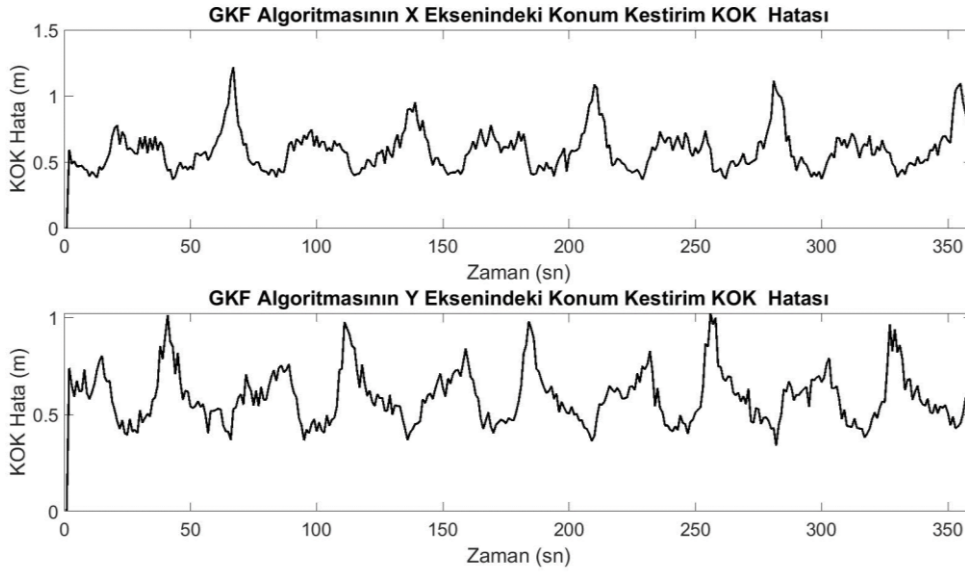
Şekil 4.10’da 1. sensör (5, 50) metre, 2. sensör (25, 50) metre ve 3. sensör ise (20, 30) metre konumunda bulunmakta ve sensörler, elips şeklindeki hareket eden nesnenin oluşturduğu hareket rotasının, içinde kalacak şekilde yerleştirilmektedir. Nesnenin başlangıçta, (30, 30) metre konumundan harekete başlayıp, $(1,1; 1,1)$ m/sn sabit hız ve $0,0873$ $radyan/saniye$ açısal hız ile 360 saniye boyunca, koordineli dönüş hareket modeline göre sensör ağının

etrafında dönen bir yol üzerinde hareket etmektedir. Sensörlerin, nesne hareketinin içine yerleştirildiğinde elde edilen, konum kestirim hata grafiği Şekil 4.12’de gösterilmiştir.



Şekil 4.11. Sensörler nesne hareketinin dışına yerleştirildiğinde, GKF algoritmasının, 100 Monte Carlo simülasyonu yapılarak elde edilen x ve y eksenlerindeki KOK konum kestirim hatası

Sensörler nesne hareketinin dışında kalacak şekilde yerleştirildiğinde, GKF algoritmasının, 100 Monte Carlo simülasyonu yapılarak elde edilen x ve y eksenlerindeki ortalama KOK konum kestirim hataları sırasıyla 0,5824 metre ve 0,4940 metre olarak elde edilmiştir.



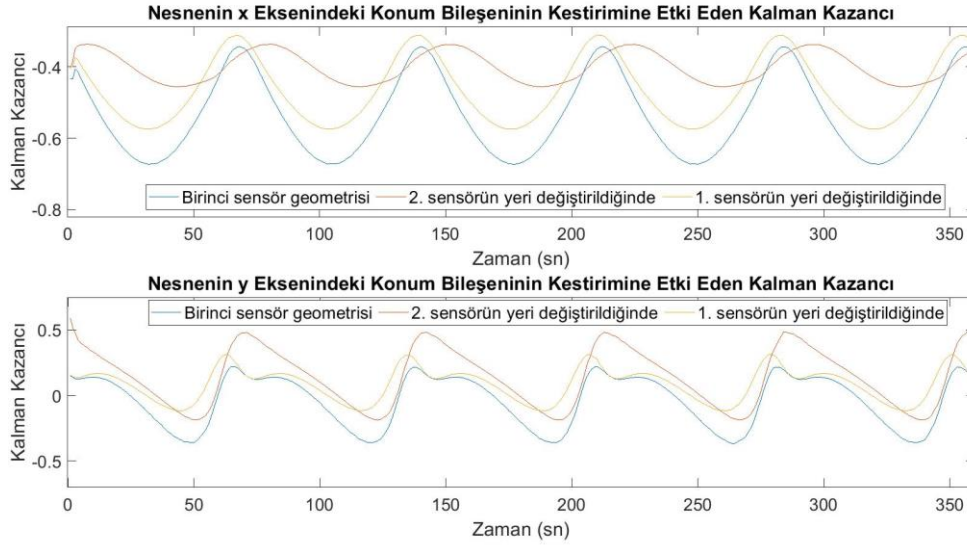
Şekil 4.12. Sensörler nesne hareketinin içine yerleştirildiğinde, GKF algoritmasının, 100 Monte Carlo simülasyonu yapılarak elde edilen x ve y eksenlerindeki KOK konum kestirim hatası

Sensörler nesne hareketinin içinde kalacak şekilde yerleştirildiğinde, GKF algoritmasının, 100 Monte Carlo simülasyonu yapılarak elde edilen x ve y eksenlerindeki ortalama KOK konum kestirim hataları sırasıyla 0,5890 metre ve 0,5873 metredir. Elde edilen sonuçlar incelendiğinde, sensörler nesne hareketinin içinde kalacak şekilde yerleştirildiğinde, nesneye daha yakın olduğundan, GKF algoritmasıyla daha küçük hatayla konum kestirimi üretilmiştir. Ayrıca sensör geometrisinin, GKF algoritmasının kestirim hatası üzerindeki etkilerinin incelenmesi için oluşturulan farklı sensör geometrilerinin kullanılmasıyla elde edilen, GKF algoritmasının konum kestirim hatası sonuçları, bölüm 4.3.1.'de verilmiştir.

4.3.1. Birinci sensör geometrisi kullanıldığında elde edilen genişletilmiş Kalman filtre algoritmasının konum kestirim hatası sonuçları

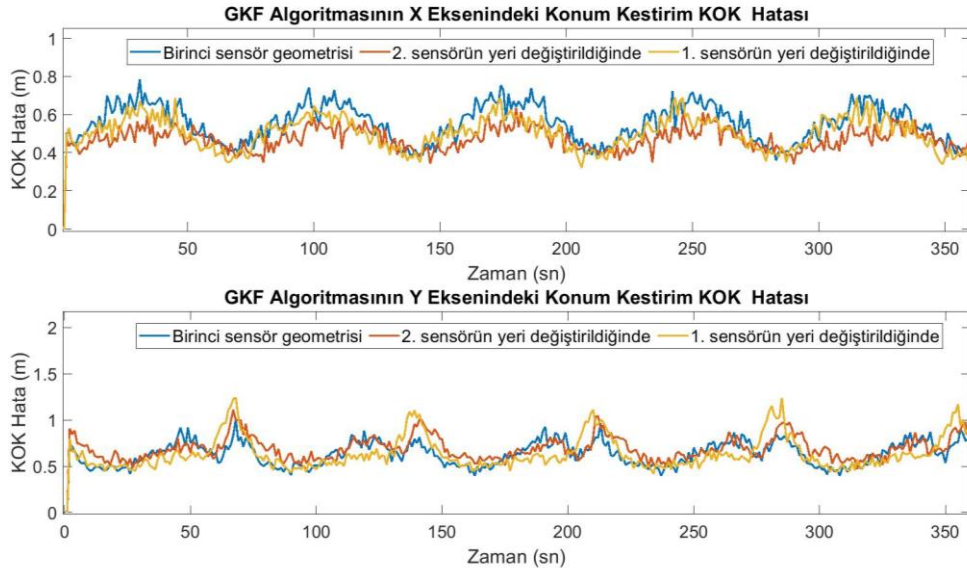
GKF algoritmasında sensör ölçümlerinin güncellenmesi için sıralı güncelleme yöntemi kullanılmıştır. Bunun için ilk olarak 1. sensör ölçümleri GKF algoritmasında kullanılarak kestirim değerleri elde edilmektedir. Elde edilen bu kestirim değeri, sıralı güncelleme aşamasında, 2. sensör ölçümleri kullanıldığında, algoritmanın durum öngörüm değeri olmaktadır. Bu durum öngörüm değeri, 2. sensör ölçümleriyle güncellenerek kestirim üretilmektedir. Ardından, bu kestirim değeri ise 3. sensör ölçümü kullanıldığında, durum öngörüm değeri olmakta ve 3. sensör ölçümleriyle güncellenerek kestirim üretilmektedir. Böylece, GKF algoritması, 3. sensör ölçümleri kullanılarak elde edilen kestirim değerini, nesnenin konum kestirim değeri olarak üretmektedir. Bu nedenle, 1., 2., 3. ve 4. sensör geometrilerinde, GKF algoritmasında sıralı güncelleme yönteminde, 3. sensör kullanıldığı durumdaki, Kalman kazancı ve kestirim değerinin, farklı sensör geometrilerine göre değişimi incelenmiştir. Sensörlerin ölçüm gürültüsü varyans değerleri $0,5 \text{ m}^2$ ve tasarlanan GKF algoritmasının süreç gürültüsü varyans değeri ise $1 \left(\frac{m}{sn^2}\right)^2$ olarak alınmıştır.

Şekil 3.4'te gösterilen birinci sensör geometrisi ve bu geometrideki 1. ve 2. sensörlerin yerleri değiştirilerek tasarlanan ve sırasıyla Şekil 3.5 ve Şekil 3.6'da gösterilen sensör geometrilerinde 3. sensör ölçümü kullanıldığında, x ve y eksenlerindeki konum kestirimine etki eden Kalman kazançları Şekil 4.13'te gösterilmiştir.



Şekil 4.13. Birinci sensör geometrisi ve birinci sensör geometrisinde bulunan sensörlerin yerleri değiştirilerek elde edilen yeni sensör geometrileri kullanıldığında, GKF algoritmasının nesnenin x ve y eksenlerindeki konum bileşenlerinin kestiriminde 3. sensör kullanıldığında elde edilen Kalman kazançları

Şekil 4.13 incelendiğinde, 1. ve 2. sensörün yeri değiştirildiğinde x ve y eksenlerindeki konum kestirimine etki eden Kalman kazançlarının çok az arttığı görülmektedir. Kazanç arttığı için, GKF algoritması, sensör ölçümüne daha çok ağırlık vererek kestirim üretmiştir.



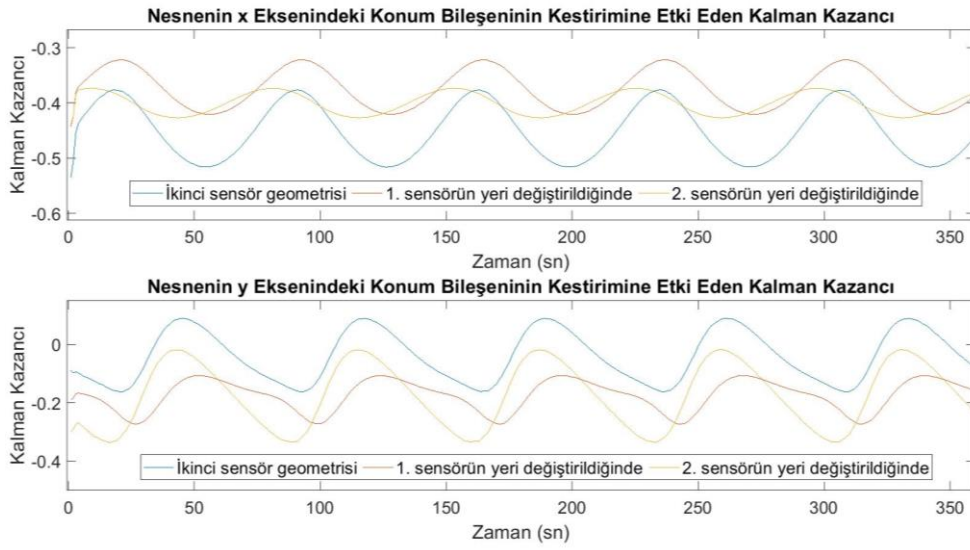
Şekil 4.14. Birinci sensör geometrisi ve birinci sensör geometrisinde bulunan sensörlerin yerlerinin değiştirilmesiyle elde edilen yeni sensör geometrileri kullanıldığında, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

Şekil 4.14'te, GKF algoritmasında 1. sensör geometrisi ve 1. sensör geometrisinde bulunan, 1. ve 2. sensörlerin yerleri değiştirilerek elde edilen yeni sensör geometrileri kullanıldığında,

100 Monte Carlo simülasyonu yapılarak elde edilen x ve y eksenlerindeki KOK konum kestirim hataları gösterilmiştir. GKF algoritmasında 1. sensör geometrisi kullanıldığında, GKF algoritmasının x ve y eksenlerindeki ortalama KOK konum kestirim hataları sırasıyla 0,5474 metre ve 0,6346 metre olarak elde edilmiştir. 1. sensör geometrisinde bulunan 1. sensörün yeri değiştirilerek elde edilen ve tezin 3. bölümünde, Şekil 3.5'te gösterilen sensör geometrisi kullanıldığında x ve y eksenlerindeki ortalama KOK konum kestirim hataları sırasıyla 0,5597 metre ve 0,6419 metre olarak elde edilmiştir. Ayrıca, 2. sensörün yeri değiştirilerek elde edilen ve Şekil 3.6'da gösterilen yeni sensör geometrisi kullanıldığında x ve y eksenlerindeki ortalama KOK konum kestirim hataları sırasıyla 0,4553 metre ve 0,6078 metre olarak elde edilmiş ve hata azalmıştır. Kalman kazancı, 2. sensörün yeri değiştirilerek elde edilen yeni sensör geometrisi kullanıldığında, 1. sensör yer değiştirildiğindeki Kalman kazancına kıyasla daha çok artmıştır. Bu nedenle, 2. sensörün yeri değiştirildiğinde, GKF algoritması kestirim üretmek için sensör ölçümünden gelen bilgiyi daha çok kullanmıştır. Nesne, gerçekte koordineli dönüş hareketi yapmakta ve GKF algoritmasının durum öngörümü aşamasında bu hareket, yaklaşık sabit hızlı hareket modeli kullanılarak öngörülmektedir. Bu nedenle, durum öngörümü aşamasında modelleme hatası yapılmakta ve bu hatalar sebebiyle, GKF algoritmasında kestirim hataları meydana gelmektedir. 2. sensörün yeri değiştirildiğinde Kalman kazancının artmasıyla, sensör ölçümünden gelen bilgi daha çok kullanıldığından, durum öngörümü aşamasındaki modellemeden kaynaklı hataların sebep olduğu kestirim hatası, sensör ölçümü ile azaltılabilmektedir. Bu nedenle, GKF algoritmasının kestirim hatası azalmıştır. Ancak, 1. sensör yer değiştirdiğinde, Kalman kazancı fazla değişmediğinden, kestirim hatasını fazla etkilememiştir.

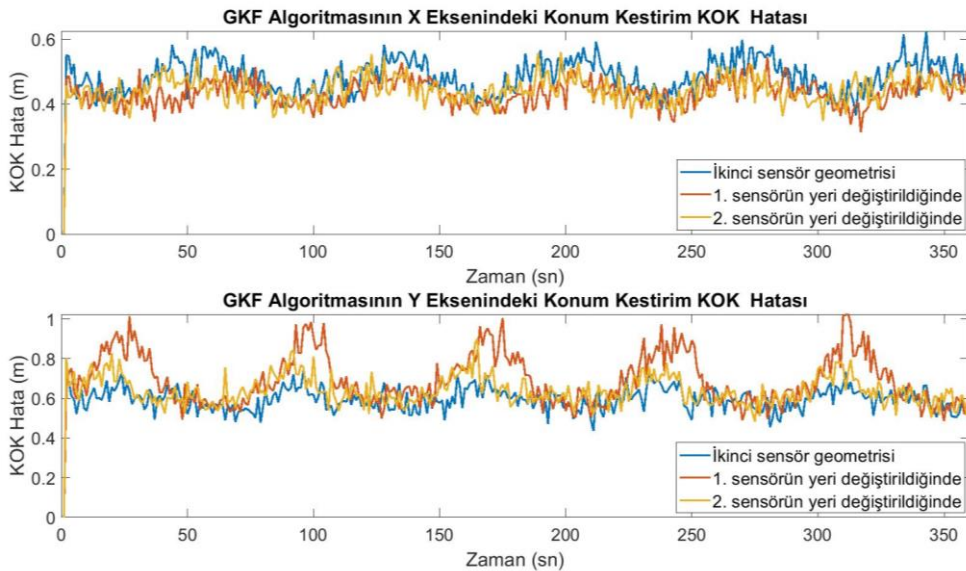
4.3.2. İkinci sensör geometrisi kullanıldığında elde edilen genişletilmiş Kalman filtre algoritmasının konum kestirim hatası sonuçları

GKF algoritmasında, Şekil 3.7'de gösterilen ikinci sensör geometrisi ve bu geometrideki 1. ve 2. sensörlerin yerleri değiştirilerek tasarlanan, Şekil 3.8 ve Şekil 3.9'da gösterilen sensör geometrileri kullanıldığında elde edilen Kalman kazançları, Şekil 4.15'te gösterilmiştir. Sensörlerin ölçüm gürültüsü varyans değeri $0,5 \text{ m}^2$ ve tasarlanan GKF algoritmasının süreç gürültüsü varyans değeri ise $1 \left(\frac{m}{sn^2}\right)^2$ olarak alınmıştır.



Şekil 4.15. İkinci sensör geometrisi ve ikinci sensör geometrisinde bulunan sensörlerin yerleri değiştirilerek elde edilen yeni sensör geometrileri kullanıldığında, GKF algoritmasının nesnenin x ve y eksenlerindeki konum bileşenlerinin kestiriminde 3. sensör kullanıldığında elde edilen Kalman kazançları

Şekil 4.15 incelendiğinde, GKF algoritmasında, 1. sensörün ve 2. sensörün yerleri değiştirilerek elde edilen yeni sensör geometrileri kullanıldığında, x eksenindeki konum kestirimine etki eden Kalman kazançlarının arttığı, y eksenindeki konum kestirimine etki eden Kalman kazançlarının ise azaldığı görülmektedir.



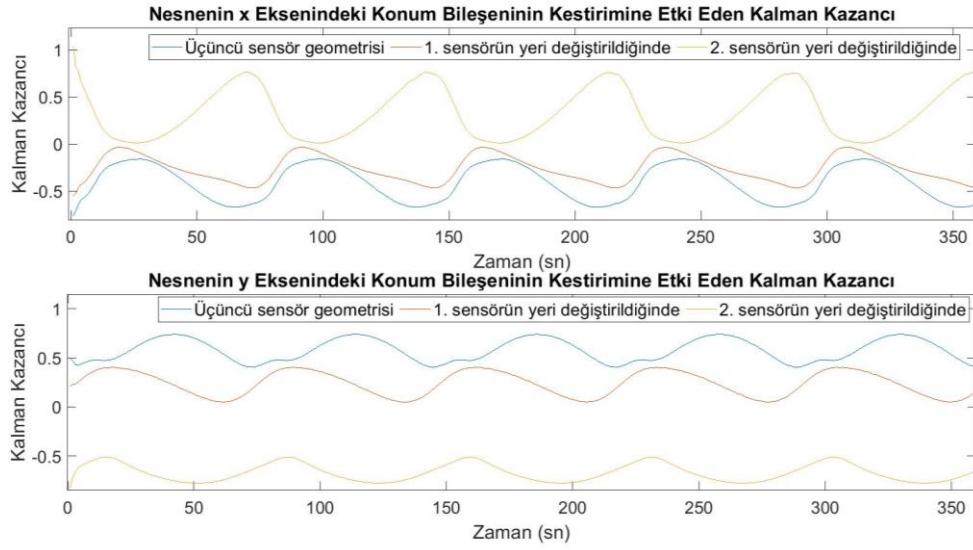
Şekil 4.16. İkinci sensör geometrisi ve ikinci sensör geometrisinde bulunan sensörlerin yerlerinin değiştirilmesiyle elde edilen yeni sensör geometrileri kullanıldığında, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

Şekil 4.16'da ikinci sensör geometrisi ve ikinci sensör geometrisinde bulunan 1. ve 2. sensörlerin yerleri değiştirilerek elde edilen yeni sensör geometrileri kullanıldığında, 100

Monte Carlo simülasyonu yapılarak elde edilen konum kestirim hataları gösterilmiştir. İkinci sensör geometrisi kullanıldığında, GKF algoritmasının x ve y eksenlerindeki ortalama KOK konum kestirim hataları sırasıyla 0,4806 metre ve 0,5954 metre olarak elde edilmiştir. İkinci sensör geometrisinde, 1. sensörün yeri değiştirilerek elde edilen yeni sensör geometrisi kullanıldığında ise x ve y eksenlerindeki ortalama KOK konum kestirim hataları sırasıyla 0,4333 metre ve 0,69 metre olarak elde edilmiştir. 2. sensörün yeri değiştirildiğinde ise x ve y eksenlerindeki ortalama KOK konum kestirim hataları sırasıyla 0,4435 metre ve 0,635 metre olarak elde edilmiştir. GKF algoritmasında, 1. ve 2. sensörün yeri değiştirildiğinde elde edilen geometriler kullanıldığında, x eksenindeki konum kestirim hataları azalmıştır. Bunun nedeni, 1. sensör ve 2. sensörün ayrı ayrı yerleri değiştirilerek elde edilen yeni sensör geometrileri kullanıldığında, Kalman kazancının artmış olmasıdır. Kalman kazancı arttığı için GKF algoritması, sensör ölçümüne daha çok ağırlık vererek kestirim ürettiğinden, durum öngörümü aşamasında modelleme hatalarından kaynaklı kestirim hataları, sensör ölçümüyle azaltılabilmektedir. Bu nedenle, kestirim hatası azalmıştır. 1. ve 2. sensörlerin yerleri değiştirilerek elde edilen yeni sensör geometrileri kullanıldığında ise y eksenindeki konum kestirim hataları, ikinci sensör geometrisi kullanıldığında elde edilen kestirim hatasından yüksektir. Nesnenin konumu, durum öngörümü aşamasında, yaklaşık sabit hızlı hareket modeli kullanılarak öngörülmüştür. Ancak, nesne gerçekte koordineli dönüş hareketi yaptığı için, durum öngörümü aşamasında modellemekten kaynaklı hatalar meydana gelmektedir. 1. ve 2. sensörlerin yerleri değiştirildiğinde, y eksenindeki konum bileşeninin kestirimine etki eden Kalman kazançları azaldığından, sensör ölçümünden gelen bilgi daha az kullanılarak kestirim üretilmiştir. Bu nedenle, durum öngörümü aşamasında modellemekten kaynaklanan hataların sebep olduğu kestirim hatası, sensör ölçümleriyle daha az azaltıldığı için kestirim hatası artmıştır.

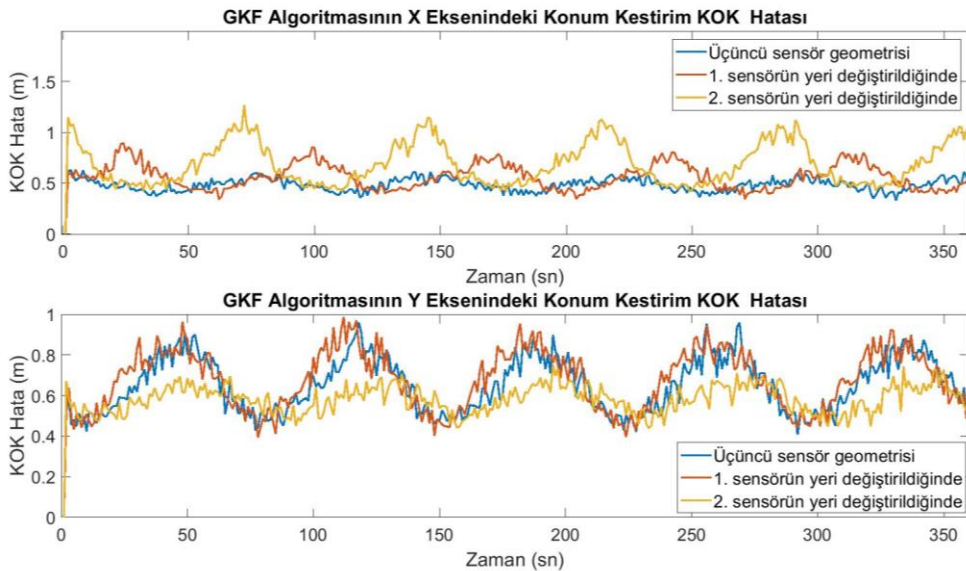
4.3.3. Üçüncü sensör geometrisi kullanıldığında elde edilen genişletilmiş Kalman filtre algoritmasının konum kestirim hatası sonuçları

GKF algoritmasında, Şekil 3.10'da verilen 3. sensör geometrisi ve bu geometrideki sensör yerlerinin değiştirilerek Şekil 3.11 ve Şekil 3.12'de gösterilen sensör geometrileri kullanıldığında elde edilen, Kalman kazanç grafikleri Şekil 4.17'de gösterilmiştir.



Şekil 4.17. Üçüncü sensör geometrisi ve üçüncü sensör geometrisinde bulunan sensörlerin yerleri değiştirilerek elde edilen yeni sensör geometrileri kullanıldığı durumda, GKF algoritmasının nesnenin x ve y eksenlerindeki konum bileşenlerinin kestiriminde 3. sensör kullanıldığında elde edilen Kalman kazançları

Şekil 4.17 incelendiğinde, 3. sensör geometrisinde bulunan 1. ve 2. sensörlerin yerleri değiştirilerek elde edilen yeni sensör geometrileri kullanıldığında x eksenindeki konum kestirimlerine etki eden Kalman kazançlarının arttığı, y eksenindeki konum kestirimlerine etki eden Kalman kazançlarının ise azaldığı görülmektedir.



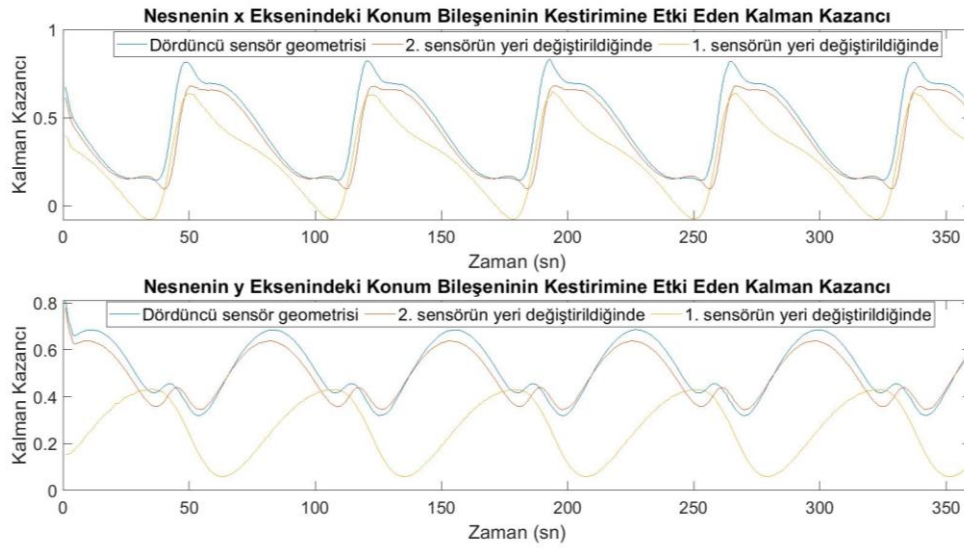
Şekil 4.18. Üçüncü sensör geometrisi ve üçüncü sensör geometrisinde bulunan sensörlerin yerlerinin değiştirilmesiyle elde edilen yeni sensör geometrileri kullanıldığında, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

Şekil 4.18'de, üçüncü sensör geometrisi ve bu geometride bulunan sensörlerin yerleri değiştirildiğinde, GKF algoritmasının, 100 Monte Carlo simülasyonu yapılarak elde edilen

KOK konum kestirim hataları gösterilmiştir. 3. sensör geometrisi kullanıldığında, GKF algoritmasının x ve y eksenlerindeki ortalama KOK kestirim hataları sırasıyla 0,4815 metre ve 0,6510 metre olarak elde edilmiştir. 2. sensör yer değiştirildiğinde ise, bu hatalar sırasıyla 0,6866 metre ve 0,5808 metre olarak elde edilmiştir. 1. sensörün yeri değiştirildiğinde ise, GKF algoritmasının x ve y eksenlerindeki ortalama KOK kestirim hataları sırasıyla 0,5594 metre ve 0,6758 metre olarak elde edilmiştir. 1. ve 2. sensörler yer değiştirildiğinde, x eksenindeki ortalama KOK konum kestirim hatası artmıştır. Bunun sebebi, Kalman kazancının artmasıdır. Kalman kazancı arttığında, GKF algoritması x eksenindeki konum kestirimini üretmek için 3. sensörün ölçümünden gelen bilgiye daha fazla ağırlık vermiştir. Sensör ölçümü, ölçüm gürültüsü içermekte ve bu gürültü ise ölçüm hatasına sebep olmaktadır. Bu nedenle, kazanç arttığında, sensörün ölçüm hatası, kestirime daha çok etki etmiş ve ölçüm hatasından kaynaklanan kestirim hatası artmıştır. GKF algoritmasında, 1. sensörün ve 2. sensörün yeri değiştirildiğinde elde edilen yeni sensör geometrileri kullanıldığında, y eksenindeki konum bileşeninin kestirimine etki eden Kalman kazançları azaldığı için, GKF algoritması sensör ölçümünden gelen bilgiyi daha az kullanarak kestirim üretmiştir. Şekil 4.44 incelendiğinde, 1. sensör yer değiştiğinde elde edilen sensör geometrisi kullanıldığında, y eksenindeki konum kestirimine etki eden Kalman kazancı fazla azalmadığından, kestirim üretmek için sensör ölçümünden gelen bilgiye verilen ağırlık ta azalmamıştır. Bu nedenle, 1. sensör yer değiştirildiğinde, sensörün ölçüm hatasından kaynaklanan, y eksenindeki konum kestirim hatası azaltılamamıştır. Ancak, 2. sensör yer değiştiğinde, y eksenindeki konum kestirimine etki eden Kalman kazancı daha çok azaldığından, sensör ölçümünden gelen bilgiye daha az ağırlık verilerek kestirim üretilmiştir. Bu durumda sensörün ölçüm hatası, kestirime daha az etki ettiğinden, ölçüm hatasından kaynaklanan kestirim hatası azalmıştır.

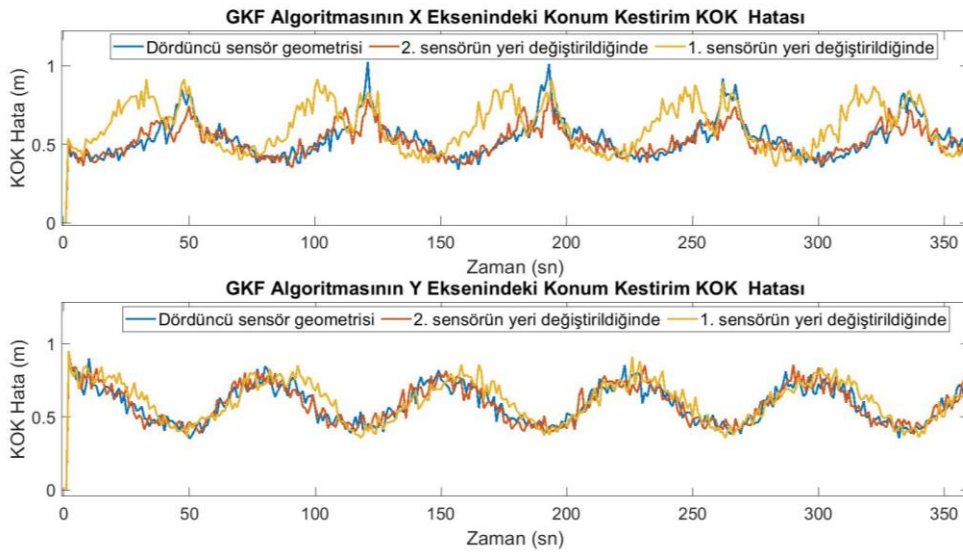
4.3.4. Dördüncü sensör geometrisi kullanıldığında elde edilen genişletilmiş Kalman filtre algoritmasının konum kestirim hatası sonuçları

Şekil 3.13'te verilen dördüncü sensör geometrisi ve bu geometrideki 1. ve 2. sensörler yeri değiştirildiğinde elde edilen ve sırasıyla Şekil 3.15 ve Şekil 3.14'te gösterilen, sensör geometrileri kullanıldığında, 3. sensörle güncelleme yapıldığı durumdaki Kalman kazanç grafikleri Şekil 4.19'da verilmiştir.



Şekil 4.19. Dördüncü sensör geometrisi kullanıldığı durumda, GKF algoritmasının nesnenin x ve y eksenlerindeki konum bileşenlerinin kestiriminde 3. sensör kullanıldığında elde edilen Kalman kazançları

1. sensörün yer değiştirilmesiyle elde edilen yeni sensör geometrisi kullanıldığında Kalman kazancı azalırken, 2. sensör yer değiştirmede fazla değişmemiştir.



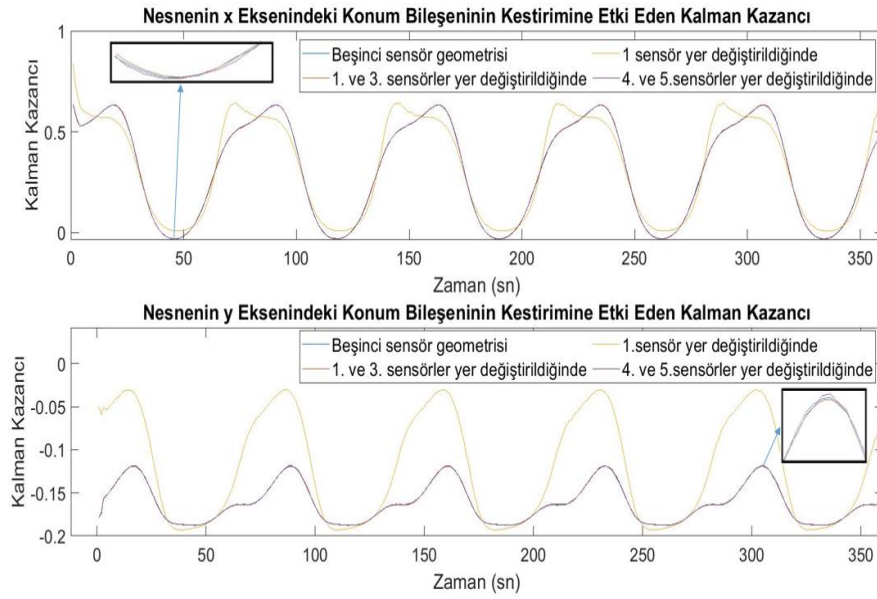
Şekil 4.20. Dördüncü sensör geometrisi ve dördüncü sensör geometrisinde bulunan sensörlerin yerlerinin değiştirilmesiyle elde edilen yeni sensör geometrileri kullanıldığında, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

Şekil 4.20’de, dördüncü sensör geometrisi ve bu geometride bulunan 1. ve 2. sensörlerin yerlerinin değiştirilmesiyle elde edilen yeni sensör geometrileri kullanıldığında, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hata grafikleri gösterilmiştir. Dördüncü sensör geometrisi kullanıldığında, GKF algoritmasının x ve y eksenlerindeki

ortalama KOK konum kestirim hataları sırasıyla 0,5313 metre ve 0,5847 metre olarak elde edilmiştir. 1. sensörün yeri değiştirilerek elde edilen yeni sensör geometrisi kullanıldığında ise, GKF algoritmasının x ve y eksenlerindeki ortalama KOK kestirim hataları sırasıyla 0,6148 metre ve 0,6112 metre, 2. sensörün yeri değiştirildiğinde ise 0,5267 metre ve 0,5862 metre olarak elde edilmiştir. 2. sensörün yeri değiştirildiğinde, kestirim hatasının fazla değişmemiştir. Bunun sebebi, 2. sensörün yeri değiştirildiğinde, Kalman kazancının fazla değişmemesidir. Kalman kazancı fazla değişmediği için, GKF algoritması, dördüncü sensör geometrisi kullanıldığı durumda, kestirim üretmek için, sensör ölçümüne verdiği ağırlıkla yaklaşık olarak aynı oranda ağırlık vererek kestirim üretmiştir. Bu nedenle, 2. sensör yer değiştirildiğinde kestirim hataları fazla değişmemiştir. 1. sensör yer değiştirilerek elde edilen yeni sensör geometrisi kullanıldığında ise x ve y eksenlerindeki konum kestirim hatalarının, dördüncü sensör geometrisi kullanıldığında kestirim hatalarına göre arttığı görülmektedir. Bunun sebebi 1. sensör yer değiştirildiğinde, Kalman kazancının azalmasıdır. Kalman kazancı azaldığından, sensör ölçümünden gelen bilgi daha az kullanıldığı için, durum öngörümü aşamasında modellemeden kaynaklı kestirim hataları, sensör ölçümüyle azaltılamamış ve kestirim hatası artmıştır.

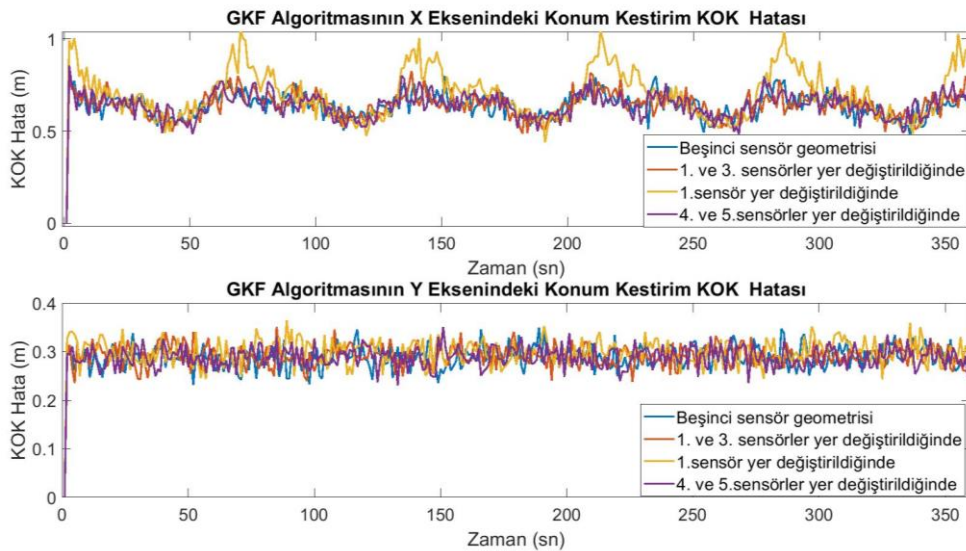
4.3.5. Beşinci sensör geometrisi kullanıldığında elde edilen genişletilmiş Kalman filtre algoritmasının konum kestirim hatası sonuçları

Şekil 3.16'da gösterilen beşinci sensör geometrisi ve bu geometrideki 1 ve 3. sensörlerin, 4 ve 5. sensörlerin ve 1. sensörün yer değiştirilmesiyle elde edilen ve sırasıyla Şekil 3.17, Şekil 3.18 ve Şekil 3.19'da gösterilen sensör geometrileri kullanıldığı durumdaki, Kalman kazanç grafikleri, Şekil 4.21'de gösterilmiştir. GKF algoritmasında sensör ölçümlerinin güncellenmesi için sıralı güncelleme yöntemi kullanılmıştır. Sensör geometrisinde 6 adet sensör kullanıldığında, sıralı güncelleme yönteminde, ilk olarak 1. sensör, daha sonra 2. sensör, şeklinde devam edilerek, en son 6. sensör ölçümleri kullanılarak ölçümler güncellenmektedir. 6. sensör ölçümleri kullanılarak elde edilen kestirim değeri de konum kestirim değeri olmaktadır. Bu sebeple, sensör geometrisi değiştirildiğinde, GKF algoritmasında sıralı güncelleme yönteminde, ölçümle güncelleme aşamasında, 6. sensör kullanıldığı durumdaki Kalman kazancı ve kestirim değerinin değişimi incelenmiştir. GKF algoritması tasarlanırken, sensörlerin ölçüm gürültüsü varyans değerleri $0,5 \text{ m}^2$ ve tasarlanan GKF algoritmasının süreç gürültüsü varyans değeri ise $1 \left(\frac{m}{sn^2}\right)^2$ olarak alınmıştır.



Şekil 4.21. Beşinci sensör geometrisi ve beşinci sensör geometrisinde bulunan sensörlerin yerleri değiştirilerek elde edilen yeni sensör geometrileri kullanıldığı durumda, GKF algoritmasının nesnenin x ve y eksenlerindeki konum bileşenlerinin kestiriminde 6. sensör kullanıldığında elde edilen Kalman kazançları

Beşinci sensör geometrisindeki, 1. sensörün yeri değiştirildiğinde x ve y eksenindeki konum kestirimine etki eden Kalman kazançları artmıştır. Ancak, 1. ve 3. sensörler ve 4. ve 5. sensörler kendi aralarında yer değiştirildiğinde x ve y eksenlerindeki konum kestirimine etki eden Kalman kazançları değişmemiştir.

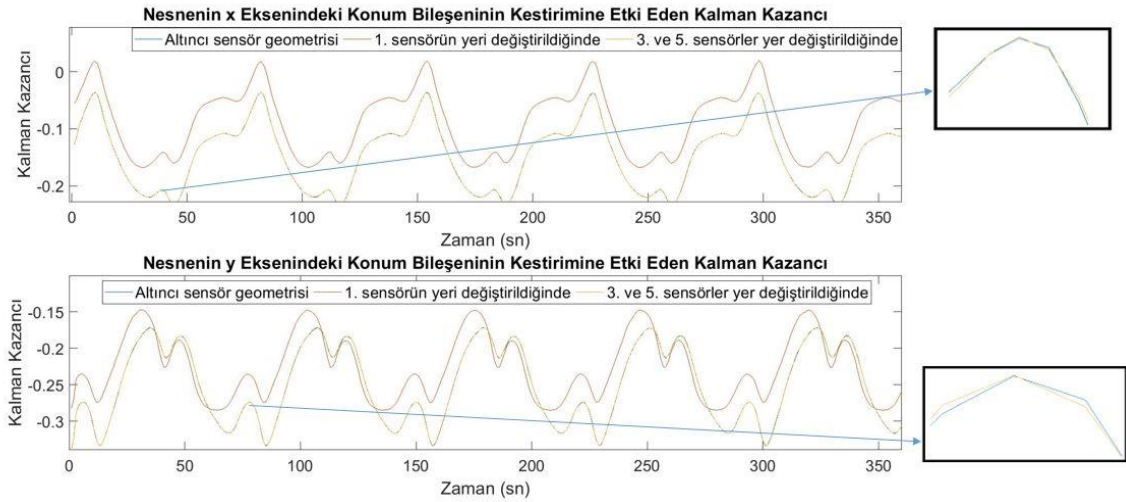


Şekil 4.22. Beşinci sensör geometrisi ve beşinci sensör geometrisinde bulunan sensörlerin yerlerinin değiştirilmesiyle elde edilen yeni sensör geometrileri kullanıldığında, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

Şekil 4.22’de, 5. sensör geometrisi ve beşinci sensör geometrisinde bulunan sensörlerin yerlerinin değiştirilmesiyle elde edilen yeni sensör geometrileri kullanıldığında, 100 Monte Carlo simülasyonu yapılarak elde edilen, x ve y eksenlerindeki KOK konum kestirim hata grafikleri gösterilmiştir. 5. sensör geometrisi kullanıldığında, GKF algoritmasının x ve y eksenlerindeki, ortalama KOK konum kestirim hataları sırasıyla 0,6424 metre ve 0,2882 metre olarak elde edilmiştir. 1. ve 3. sensörlerin kendi aralarında yer değiştirilerek elde edilen yeni sensör geometrisi kullanıldığında ise x ve y eksenlerindeki, ortalama KOK konum kestirim hataları sırasıyla 0,6430 metre ve 0,2869 metre olarak elde edilmiştir. Elde edilen sonuçlar incelendiğinde, 1. ve 3. sensörler kendi aralarında yer değiştirdiğinde, GKF algoritmasının kestirim hatasının yaklaşık olarak aynı kaldığı görülmektedir. Bunun nedeni, 1. ve 3. sensörler kendi aralarında yer değiştirdiğinde, Kalman kazancının değişmemesidir. Kalman kazancı değişmediği için, GKF algoritması kestirim üretirken, sensör ölçümünden gelen bilgiye aynı oranda ağırlık verdiğinden kestirim hatası değişmemiştir. Beşinci sensör geometrisinde, 1. sensörün yeri değiştirildiğinde, GKF algoritmasının x ve y eksenlerindeki ortalama KOK kestirim hataları sırasıyla 0,7039 metre ve 0,3015 metre olarak elde edilmiş ve kestirim hatası artmıştır. Bunun sebebi, 1. sensör yer değiştirildiğinde x ve y eksenindeki konum kestirimine etki eden Kalman kazançlarının artmasıdır. Kalman kazancı arttığı için, GKF algoritması 6. sensör ölçümüne daha çok ağırlık vererek kestirim üretmiştir. Sensör ölçümü, ölçüm gürültüsü içermekte ve bu gürültü ise ölçüm hatasına sebep olmaktadır. Kazanç arttığında, kestirim üretilirken sensör ölçümünden gelen bilgiye daha fazla ağırlık verildiğinden, sensörün ölçüm hatası, kestirime daha çok etki etmiş ve ölçüm hatasından kaynaklanan kestirim hatası artmıştır. 5. sensör geometrisinde, 4. ve 5. sensörler yer değiştirildiğinde ise algoritmanın x ve y eksenlerindeki ortalama KOK kestirim hataları sırasıyla 0,6373 metre ve 0,2887 metre olarak elde edilmiş ve GKF algoritmasının kestirim hatası yaklaşık olarak sabit kalmıştır. Bunun sebebi ise 4. ve 5. sensörler yer değiştirildiğinde, Kalman kazancı değişmediği için, 4. ve 5. sensörler yer değiştirdiğinde, GKF algoritması kestirim üretirken 6. sensörün ölçümüne aynı oranda ağırlık vererek kestirim üretmiştir. Bu sonuç, sensör geometrisi değiştiğinde, GKF algoritmasının kestirim hatasının da değişmesi için, sensör geometrisinin değişimine bağlı olarak Kalman kazancının da değişmesi gerektiğini göstermektedir. Çünkü kazanç değiştiğinde, GKF algoritmasının kestirim üretmek için, sensör ölçümünden gelen bilgiye verdiği ağırlık ta değiştiği için, kestirim değeri ve kestirim hatası değişmektedir.

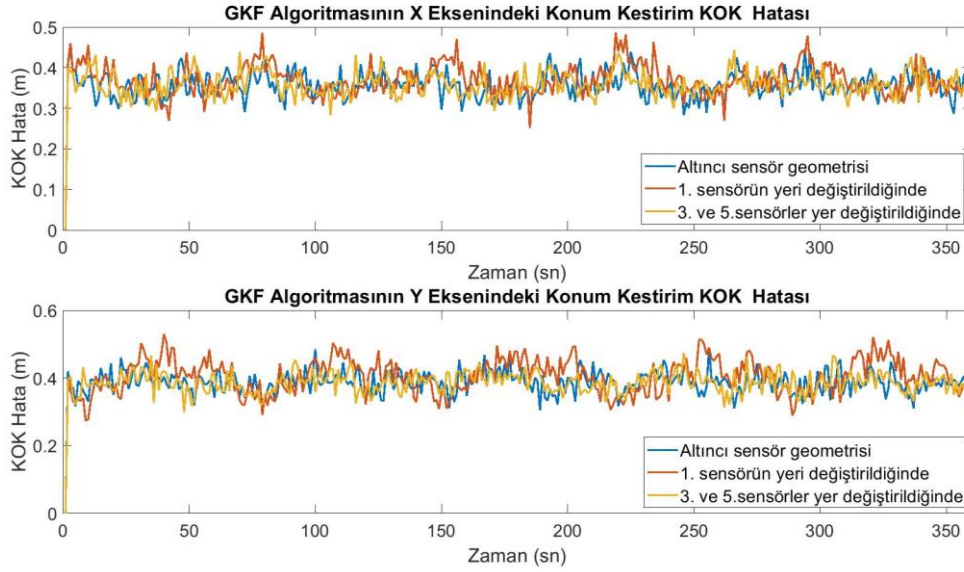
4.3.6. Altıncı sensör geometrisi kullanıldığında elde edilen genişletilmiş Kalman filtre algoritmasının konum kestirim hatası sonuçları

Şekil 4.23'te, altıncı sensör geometrisi kullanıldığında ve bu geometrideki 1. sensörün yeri değiştirilerek elde edilen ve 3. sensörle 5. sensörün kendi arasında yer değiştirilmesiyle elde edilen sensör geometrileri kullanıldığında x ve y eksenlerindeki konum kestirimindeki Kalman kazanç grafikleri gösterilmiştir. 6. sensör geometrisinde bulunan 1. sensörün yer değiştirilmesiyle elde edilen sensör geometrisi Şekil 3.21'de gösterilmiştir. 3. ve 5. sensörlerin kendi aralarında yer değiştirilmesiyle elde edilen sensör geometrisi ise Şekil 3.22'de gösterilmiştir. GKF algoritması tasarlanırken, sensörlerin ölçüm gürültüsü varyans değeri $0,5 \text{ m}^2$ ve tasarlanan GKF algoritmasının süreç gürültüsü varyans değeri ise $1 \left(\frac{m}{sn^2}\right)^2$ olarak alınmıştır.



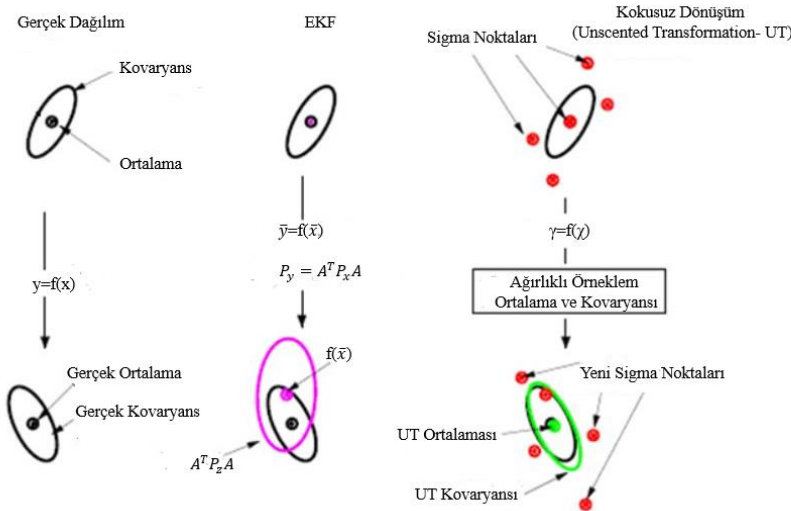
Şekil 4.23. Altıncı sensör geometrisi ve altıncı sensör geometrisinde bulunan sensörlerin yerleri değiştirilerek elde edilen yeni sensör geometrileri kullanıldığı durumda, GKF algoritmasının nesnenin x ve y eksenlerindeki konum bileşenlerinin kestiriminde, 6. sensör kullanıldığında elde edilen Kalman kazançları

Altıncı sensör geometrisinde bulunan 1. sensörün yeri değiştiğinde, GKF algoritmasının x ve y eksenlerindeki konum kestirimine etki eden Kalman kazançları artmıştır. O nedenle, GKF algoritması, 6. sensörün ölçümlerine daha fazla ağırlık vererek kestirim üretmiştir. 3. ve 5. sensörler kendi aralarında yer değiştirildiğinde ise x ve y eksenlerindeki konum kestirimine etki eden Kalman kazançlarının değişmediği görülmektedir.



Şekil 4.24. Altıncı sensör geometrisi ve altıncı sensör geometrisinde bulunan sensörlerin yerlerinin değiştirilmesiyle elde edilen yeni sensör geometrileri kullanıldığında, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

Şekil 4.24'te, GKF algoritmasında, 6. sensör geometrisi ve 6. sensör geometrisinde bulunan sensörlerin yerlerinin değiştirilmesiyle elde edilen yeni sensör geometrileri kullanıldığında, elde edilen x ve y eksenlerindeki KOK konum kestirim hataları gösterilmiştir. 6. sensör geometrisi kullanıldığında, GKF algoritmasının x ve y eksenlerindeki ortalama KOK kestirim hataları sırasıyla 0,3551 metre ve 0,3868 metre olarak elde edilmiştir. 1. sensörün yeri değiştirilerek elde edilen yeni sensör geometrisi kullanıldığında ise x ve y eksenlerindeki ortalama KOK kestirim hataları sırasıyla 0,3659 metre ve 0,4028 metre olarak elde edilmiş ve kestirim hatası artmıştır. Bunun sebebi, 1. sensörün yeri değiştirildiğinde, Kalman kazancı arttığı için sensör ölçümünden gelen bilgi, GKF algoritmasının kestirimine daha çok etki etmiştir. Sensör ölçümü, ölçüm gürültüsü içermekte ve bu gürültü ise ölçüm hatasına sebep olmaktadır. Bu nedenle, kazanç arttığında, ölçüm hatası, kestirime daha çok etki ettiği için kestirim hatası artmıştır. 3. ve 5. sensörlerin kendi aralarında yer değiştirildiğinde ise GKF algoritmasının x ve y eksenlerindeki ortalama KOK kestirim hataları sırasıyla 0,3575 metre ve 0,3831 metre olarak elde edilmiş ve kestirim hatasının, 6. sensör geometrisi kullanıldığı durumda elde edilen kestirim hatasıyla yaklaşık olarak aynı kaldığı görülmüştür. Bunun sebebi, 3. ve 5. sensörlerin kendi aralarında yer değiştirildiğinde Kalman kazancı değişmediğinden, GKF algoritması sensör ölçümüne yine aynı oranda ağırlık vererek kestirim ürettiği için, kestirim hatası değişmemiştir.



Şekil 4.25. İki boyutlu Gauss dağılımına sahip rasgele değişkenin doğrusal olmayan dönüşümü sonucu ortalama ve kovaryans değerlerinin gerçek değerleri ile GKF ve kokusuz dönüşüm sonucu elde edilen değerleri (Lima, 2017).

GKF algoritmasında, farklı sensör geometrileri kullanılarak, konum kestirim değerleri elde edilirken, bazı sensör geometrileri kullanıldığında Kalman kazancı negatif çıkmıştır. Bunun sebebi Şekil 4.25 kullanılarak açıklanmıştır. Şekil 4.25'te iki boyutlu Gauss dağılımına sahip rasgele bir değişkene ait gerçek ortalama değer ve gerçek kovaryans dağılımı gösterilmiştir. GKF algoritması kullanılarak, rasgele değişkene ait bu gerçek ortalama değer ve kovaryans dağılımının kestirimi elde edilmektedir. Bu kestirimler elde edilirken, GKF algoritmasında, doğrusal olmayan durum veya ölçüm fonksiyonlarının doğrusallaştırılması için Taylor serisi açılımı kullanılmaktadır. Doğrusal olmayan ölçüm fonksiyonunun doğrusallaştırılması için Jakobyian matris hesaplanmıştır. Jakobyian matris hesaplanırken, Taylor serisi açılımının ilk iki terimi kullanılmaktadır. Ancak, Taylor serisi açılımı sonsuz terime sahiptir. Sonsuz terime sahip bu açılım, sadece ilk iki terimle ifade edildiğinden, doğrusallaştırmadan kaynaklı hatalar meydana gelmektedir. Bu hatalar sebebiyle, GKF algoritması tarafından hesaplanan kovaryans ile rasgele değişkene ait gerçek kovaryans birbiri ile örtüşmemektedir. Şekil 4.25'te, rasgele değişkene ait gerçek kovaryans dağılımıyla, bu rasgele değişkene ait kovaryansın, GKF algoritması tarafından hesaplanmış hali olan ve mor renkli elips ile gösterilen kovaryans dağılımı, birbirlerinden farklıdır ve birbirleriyle örtüşmemektedir. Bu nedenle, Kalman kazancı negatif çıkabilmektedir.

4.4. Sensör Sayısının Genişletilmiş Kalman Filtre Algoritmasının Konum Kestirim Hatası Üzerindeki Etkilerinin İncelenmesi

Sensör sayılarının GKF algoritmasının kestirim hatası üzerindeki etkileri 4.4 numaralı bölümün alt bölümlerinde incelenmiştir. Sensörlerin ölçüm gürültü varyansları $0,5 \text{ m}^2$ ve filtrenin süreç gürültüsü varyansı $1 \left(\frac{m}{sn^2}\right)^2$ alınarak elde edilen sonuçlar gösterilmiştir.

4.4.1. Birinci sensör geometrisinde kullanılan sensör sayıları arttırıldığında elde edilen genişletilmiş Kalman filtre algoritmasının konum kestirim hatası sonuçları

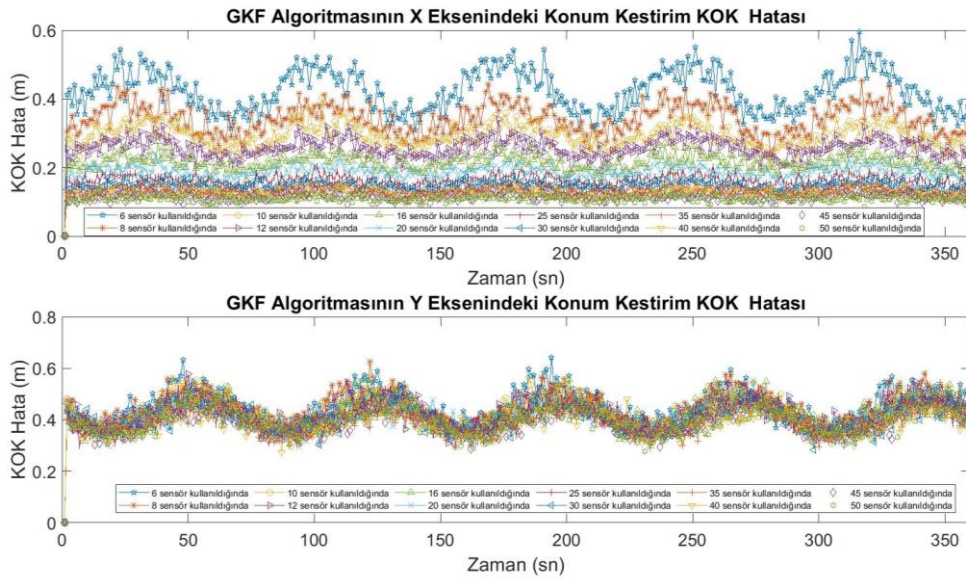
Birinci sensör geometrisinde kullanılan sensör sayıları arttırıldığında, GKF algoritmasının x ve y eksenlerindeki KOK kestirim hata grafikleri Şekil 4.26'da gösterilmiştir. Çizelge 4.3'te ise GKF algoritmasının x ve y eksenlerindeki ortalama konum kestirim hata değerleri verilmiştir. Çizelgelerde verilen, GKF algoritmasının, net ortalama KOK konum kestirim hatası, Eş. 4.19 ile verilen eşitlik kullanılarak hesaplanmıştır.

$$KOK_{net} = \sqrt{KOK_x^2 + KOK_y^2} \quad (4.19)$$

Eş. 4.19'da KOK_{net} ifadesi, net ortalama KOK kestirim hatasını KOK_x ve KOK_y ifadeleri ise sırasıyla x ve y eksenlerindeki ortalama KOK konum kestirim hatasını belirtmektedir. 1. sensör geometrisinde kullanılan sensör sayıları arttırıldığında elde edilen yeni sensör geometrileri Şekil 3.23 ile Şekil 3.34 arasında sırayla verilmiştir.

Çizelge 4.3. GKF algoritmasının, 1. sensör geometrisinde kullanılan sensör sayıları arttırıldığında elde edilen KOK konum kestirim hataları

Sensör Sayısı	X Eksenindeki Ortalama KOK Kestirim Hatası (m)	Y Eksenindeki Ortalama KOK Kestirim Hatası (m)	Net Ortalama KOK Kestirim Hatası (m)
6	0,4226	0,4488	0,6165
8	0,3392	0,4396	0,5553
10	0,2909	0,4327	0,5214
12	0,2559	0,4290	0,4995
16	0,2155	0,4229	0,4746
20	0,1853	0,4107	0,4505
25	0,1623	0,4078	0,4389
30	0,1466	0,4025	0,4283
35	0,1337	0,3989	0,4207
40	0,1239	0,3985	0,4173
45	0,1160	0,3953	0,4119
50	0,1094	0,3925	0,4075

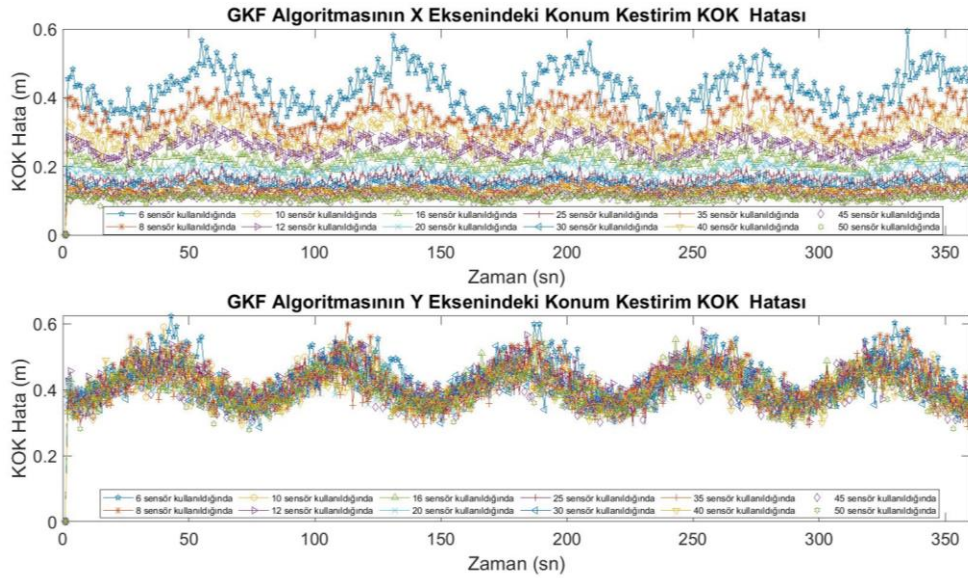


Şekil 4.26. 1. sensör geometrisinde kullanılan sensör sayısının arttırılmasıyla elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

Şekil 4.26, incelendiğinde, sensör sayısı belli miktar arttırıldığında, GKF algoritmasının x eksenindeki konum kestirim hatasının azaldığı görülmektedir. Bunun sebebi, sensör sayısı arttığında nesneden alınan bilgi miktarı arttığı için, x eksenindeki konum kestirim hatası azalmaktadır. Ancak, sensör sayısı arttırılmaya devam edildiğinde, eklenen yeni sensörler, nesneden uzaklaştığı için, sensörlerden algoritmaya ekstra bilgi gelmediğinden kestirim hatası daha fazla azalmamaktadır. 1. sensör geometrisindeki sensörlerin, y eksenlerindeki bileşenleri aynı olduğundan, her bir sensörden, GKF algoritmasına aynı bilgi geldiği için, sensör sayısı artsa da y eksenindeki kestirim hatası azalmamıştır.

4.4.2. İkinci sensör geometrisinde kullanılan sensör sayıları arttırıldığında elde edilen genişletilmiş Kalman filtre algoritmasının konum kestirim hatası sonuçları

İkinci sensör geometrisinde kullanılan sensör sayıları 6'dan 50'ye sırayla arttırıldığında ve Şekil 3.35 ile Şekil 3.46 arasında sırayla verilen sensör geometrileri kullanıldığında, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hata grafikleri, Şekil 4.27'de gösterilmiştir.



Şekil 4.27. 2. sensör geometrisinde kullanılan sensör sayısının artırılmasıyla elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

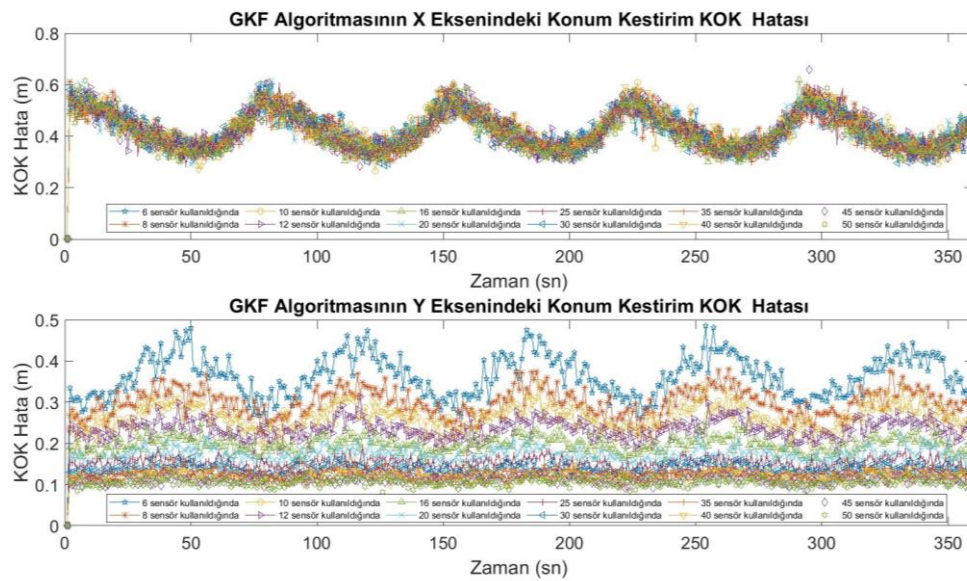
GKF algoritmasının x eksenindeki konum kestirim hatası, sensör sayısı belli miktar arttığında, azalmıştır. Bunun sebebi, sensör sayısı belli miktar arttırıldığında hedef nesneden toplanan bilgi miktarı arttığı için, kestirim hatası azalmaktadır. Ancak, sensör sayısı arttırılmaya devam edildiğinde, sensörler, hedef nesneden uzaklaştığı için, sensörlerden GKF algoritmasına ekstra bilgi gelmediğinden, kestirim hatası daha fazla azalmamaktadır. GKF algoritmasının y eksenindeki kestirim hatasının ise, sensör sayısı artsa da sabit kaldığı görülmektedir. 2. sensör geometrisinde, sensörlerin bulundukları konumların y eksenindeki bileşenleri aynı olduğundan, GKF algoritmasına her bir sensörden aynı bilgi geldiği için, sensör sayısının arttırılması, y eksenindeki konum kestirim hatasını etkilememiştir.

Çizelge 4.4. GKF algoritmasının, 2. sensör geometrisinde kullanılan sensör sayıları artırıldığında elde edilen ortalama KOK konum kestirim hataları

Sensör Sayısı	X Eksenindeki Ortalama KOK Kestirim Hatası (m)	Y Eksenindeki Ortalama KOK Kestirim Hatası (m)	Net Ortalama KOK Kestirim Hatası (m)
6	0,4229	0,4445	0,6135
8	0,3408	0,4411	0,5574
10	0,2910	0,4314	0,5203
12	0,2589	0,4268	0,4992
16	0,2146	0,4140	0,4663
20	0,1869	0,4101	0,4507
25	0,1629	0,4089	0,4401
30	0,1469	0,4015	0,4275
35	0,1333	0,3961	0,4179
40	0,1247	0,3949	0,4141
45	0,1162	0,3949	0,4116
50	0,1088	0,3914	0,4062

4.4.3. Üçüncü sensör geometrisinde kullanılan sensör sayıları artırıldığında elde edilen genişletilmiş Kalman filtre algoritmasının konum kestirim hatası sonuçları

Sensör sayıları 6'dan 50'ye sırayla artırıldığında ve Şekil 3.47 ile Şekil 3.58 arasında sırayla verilen sensör geometrileri kullanıldığında, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları Şekil 4.28'de gösterilmiştir.



Şekil 4.28. 3. sensör geometrisinde kullanılan sensör sayısının artırılmasıyla elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

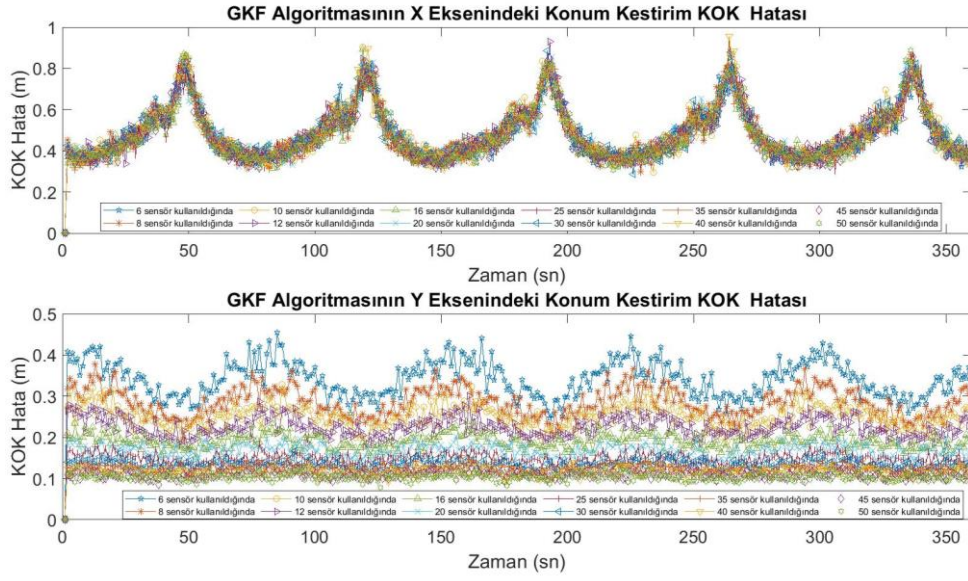
Grafik incelendiğinde sensör sayısı artışının, GKF algoritmasının x eksenindeki konum kestirimine etki etmediği görülmektedir. Bunun sebebi, 3. sensör geometrisinde, her bir sensörün x eksenindeki konum bileşeni aynı olduğundan, her bir sensörden, GKF algoritmasına aynı bilgi gelmektedir. GKF algoritmasının y eksenindeki konum kestirim hatası incelendiğinde ise, sensör sayısı belli miktar arttırıldığında kestirim hatasının azaldığı ancak arttırılmaya devam edilerek 35 sensörden, 50 sensöre kadar arttırıldığında, kestirim hatasının daha fazla azalmadığı görülmektedir. Bunun sebebi, nesne eliptik şekilde hareket ettiğinde ve 3. sensör geometrisindeki sensör sayısı arttırıldığında, eklenen yeni sensörler, nesneden uzaklaştığı için, algoritmaya sensörlerden daha fazla bilgi gelmemektedir. Bu nedenle, kestirim hatası da daha fazla azalmamaktadır. GKF algoritmasının, 3. sensör geometrisinde sensör sayısının arttırılmasına bağlı olarak elde edilen ortalama KOK kestirim hata değerleri ise Çizelge 4.5’te verilmiştir.

Çizelge 4.5. GKF algoritmasının, 3. sensör geometrisinde kullanılan sensör sayıları arttırıldığında elde edilen ortalama KOK konum kestirim hataları

Sensör Sayısı	X Eksenindeki Ortalama KOK Kestirim Hatası (m)	Y Eksenindeki Ortalama KOK Kestirim Hatası (m)	Net Ortalama KOK Kestirim Hatası (m)
6	0,4329	0,3633	0,5651
8	0,4328	0,2988	0,5259
10	0,4311	0,2619	0,5044
12	0,4293	0,2343	0,4891
16	0,4261	0,1958	0,4689
20	0,4205	0,1720	0,4543
25	0,4200	0,1526	0,4468
30	0,4190	0,1370	0,4408
35	0,4155	0,1275	0,4346
40	0,4153	0,1186	0,4319
45	0,4151	0,1113	0,4297
50	0,4127	0,1047	0,4257

4.4.4. Dördüncü sensör geometrisinde kullanılan sensör sayıları arttırıldığında elde edilen genişletilmiş Kalman filtre algoritmasının kestirim hatası sonuçları

Dördüncü sensör geometrisinde kullanılan sensör sayıları 6’dan 50’ye sırayla arttırıldığında ve Şekil 3.59 ile Şekil 3.70 arasında sırayla verilen sensör geometrileri kullanıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hata grafikleri Şekil 4.29’da gösterilmiştir.



Şekil 4.29. 4. sensör geometrisinde kullanılan sensör sayısının artırılmasıyla elde edilen GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

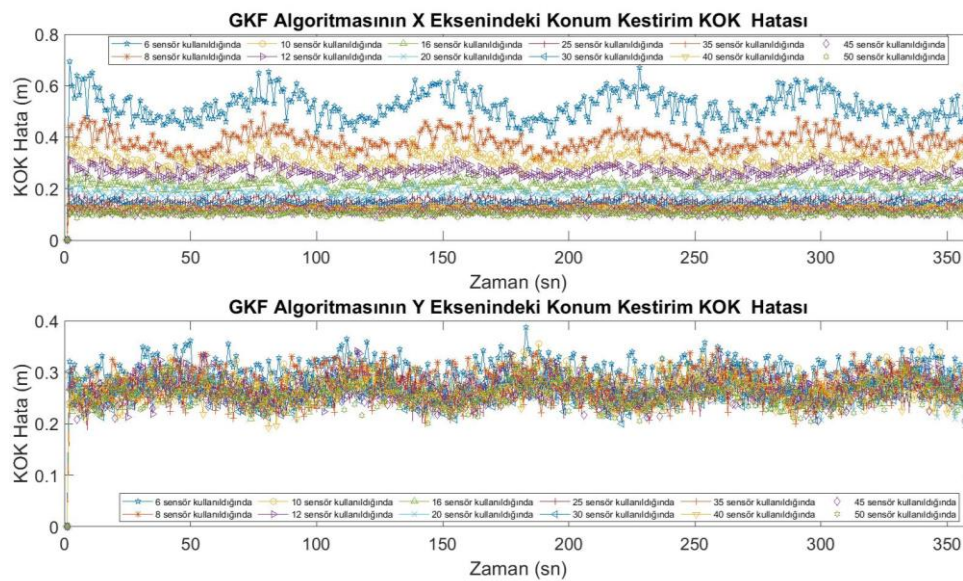
4. sensör geometrisinde, sensör sayısı artırılrsa da x eksenindeki konum kestirim hatası değişmemiştir. Bunun sebebi, sensörler x eksenine dik şekilde yerleştirildiğinden, her bir sensörün bulunduğu konumun x eksenindeki bileşenleri aynıdır. Bu nedenle, sensörlerden GKF algoritmasına sürekli aynı bilgi geldiği için sensör sayısı artsa da kestirim hatası azalmamaktadır. GKF algoritmasının, y eksenindeki konum kestirim hatası ise, sensör sayısı arttığında hedef nesneden daha fazla bilgi elde edildiği için azalmaktadır. Fakat sensör sayısı daha çok artırıldığında eklenen yeni sensörler, nesneden uzaklaştığından sensörlerden GKF algoritmasına daha fazla bilgi gelmediği için kestirim hatası daha fazla azalmamaktadır. GKF algoritmasının sensör sayısı artırıldığında elde edilen ortalama KOK kestirim hata değerleri Çizelge 4.6’da verilmiştir.

Çizelge 4.6. GKF algoritmasının, 4. sensör geometrisinde kullanılan sensör sayıları artırıldığında elde edilen ortalama KOK konum kestirim hataları

Sensör Sayısı	X Eksenindeki Ortalama KOK Kestirim Hatası (m)	Y Eksenindeki Ortalama KOK Kestirim Hatası (m)	Net Ortalama KOK Kestirim Hatası (m)
6	0,4909	0,3404	0,5973
8	0,4901	0,2840	0,5664
10	0,4886	0,2499	0,5487
12	0,4871	0,2240	0,5361
16	0,4836	0,1898	0,5195
20	0,4825	0,1676	0,5107
25	0,4819	0,1489	0,5043
30	0,4803	0,1341	0,4986
35	0,4785	0,1250	0,4945
40	0,4765	0,1163	0,4904
45	0,4765	0,1093	0,4887
50	0,4734	0,1029	0,4844

4.4.5. Beşinci sensör geometrisinde kullanılan sensör sayıları artırıldığında elde edilen genişletilmiş Kalman filtre algoritmasının kestirim hatası sonuçları

Beşinci sensör geometrisinde kullanılan sensör sayıları 6'dan 50'ye sırayla artırıldığında ve Şekil 3.71 ile Şekil 3.81 arasında sırayla verilen sensör geometrileri kullanıldığında elde edilen x ve y eksenlerindeki KOK konum kestirim hata grafikleri Şekil 4.30'da gösterilmiştir.



Şekil 4.30. 5. sensör geometrisinde kullanılan sensör sayısının artırılmasıyla elde edilen GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

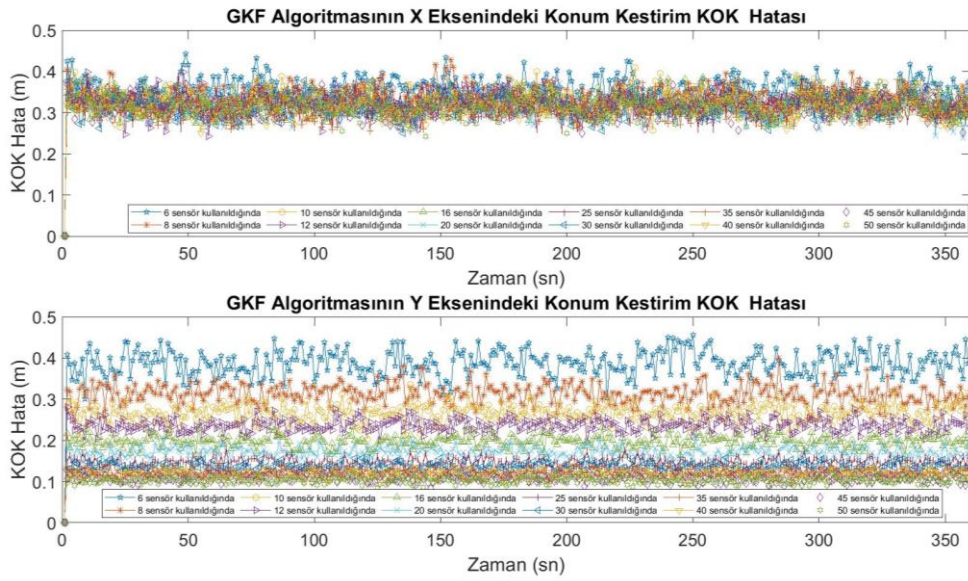
Şekil 4.30 incelendiğinde, sensör sayısı belli miktar arttırıldığında, GKF algoritmasının x eksenindeki konum kestirim hatasının azaldığı, ancak sensör sayısı arttırılmaya devam edildiğinde kestirim hatasının daha fazla azalmadığı görülmektedir. 5. sensör geometrisi incelendiğinde, sensör sayısı arttıkça sensörler ile hedef nesne arasındaki uzaklık arttığından, sensör sayısı arttırılsa bile, kestirim hatası daha fazla azalmamaktadır. Şekil 4.30 incelendiğinde, GKF algoritmasının y eksenindeki konum kestirim hatasının ise, sensör sayısı artsa da sabit kaldığı görülmektedir. 5. sensör geometrisinde, sensörler hedef nesnenin altında ve üstünde olacak şekilde ve y eksenine dik şekilde yerleştirilmiştir. Bu nedenle, sensör konumlarının y eksenindeki bileşenleri aynı olduğu için, GKF algoritmasına her bir sensörden aynı bilgi gelmektedir. O nedenle, sensör sayısının arttırılması, GKF algoritmasının y eksenindeki konum kestirim hatasını etkilememiştir. GKF algoritmasının, 5. sensör geometrisinde sensör sayısının arttırılmasına bağlı olarak elde edilen ortalama KOK kestirim hata değerleri ise Çizelge 4.7’de verilmiştir.

Çizelge 4.7. GKF algoritmasının, 5. sensör geometrisinde kullanılan sensör sayıları arttırıldığında elde edilen ortalama KOK konum kestirim hataları

Sensör Sayısı	X Eksenindeki Ortalama KOK Kestirim Hatası (m)	Y Eksenindeki Ortalama KOK Kestirim Hatası (m)	Net Ortalama KOK Kestirim Hatası (m)
6	0,5197	0,3041	0,6021
8	0,3819	0,2855	0,4768
10	0,3106	0,2789	0,4174
12	0,2650	0,2716	0,3794
16	0,2128	0,2655	0,3403
20	0,1828	0,2616	0,3191
25	0,1544	0,2604	0,3027
30	0,1401	0,2584	0,2939
35	0,1271	0,2583	0,2878
40	0,1201	0,2565	0,2832
45	0,1104	0,2558	0,2786
50	0,1046	0,2537	0,2744

4.4.6. Altıncı sensör geometrisinde kullanılan sensör sayıları arttırıldığında elde edilen genişletilmiş Kalman filtre algoritmasının kestirim hatası sonuçları

Altıncı sensör geometrisinde kullanılan sensör sayıları 6’dan 50’ye sırayla arttırıldığında ve Şekil 3.82 ile Şekil 3.92 arasında sırayla verilen sensör geometrileri kullanıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hata grafikleri Şekil 4.31’de gösterilmiştir.



Şekil 4.31. 6. sensör geometrisinde kullanılan sensör sayısının arttırılmasıyla elde edilen GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

Şekil 4.31 incelendiğinde, sensör sayısı artsa da x eksenindeki konum kestirim hatasının değişmediği görülmektedir. Bunun sebebi, 6. sensör geometrisinde sensörlerin, elips şeklindeki hedef hareketinin sağında ve solunda olmak üzere, x eksenine dik şekilde yerleştirilmiş olmasıdır. Sensörler x eksenine dik şekilde yerleştirildiğinden, x eksenindeki konum bileşenleri aynı olduğu için, GKF algoritmasına her bir sensörden aynı bilgi gelmektedir. Bu sebeple, sensör sayısı artsa da x eksenindeki konum kestirim hatası azalmamaktadır. GKF algoritmasının y eksenindeki konum kestirim hatası incelendiğinde ise, sensör sayısı belli miktar arttırıldığında, kestirim hatasının azaldığı, ancak sensör sayısı arttırılmaya devam edildiğinde kestirim hatasının daha fazla azalmadığı görülmektedir. Bunun sebebi, sensör sayısı arttırıldığında, sensörler hedef nesneden uzaklaştığı için, sensörlerden daha fazla ekstra bilgi gelmediğinden dolayı kestirim hatası daha fazla azalmamaktadır. GKF algoritmasının, 6. sensör geometrisinde sensör sayısının arttırılmasına bağlı olarak elde edilen ortalama KOK kestirim hata değerleri ise Çizelge 4.8’de verilmiştir.

Çizelge 4.8. GKF algoritmasının, 6. sensör geometrisinde kullanılan sensör sayıları artırıldığında elde edilen ortalama KOK konum kestirim hataları

Sensör Sayısı	X Eksenindeki Ortalama KOK Kestirim Hatası (m)	Y Eksenindeki Ortalama KOK Kestirim Hatası (m)	Net Ortalama KOK Kestirim Hatası (m)
6	0,3588	0,3865	0,5273
8	0,3359	0,3136	0,4595
10	0,3252	0,2662	0,4202
12	0,3243	0,2364	0,4013
16	0,3215	0,1971	0,3771
20	0,3181	0,1706	0,3609
25	0,3161	0,1474	0,3487
30	0,3156	0,1349	0,3432
35	0,3141	0,1230	0,3373
40	0,3138	0,1169	0,3348
45	0,3126	0,1076	0,3306
50	0,3116	0,1024	0,3279

4.5. Farklı Ölçüm Gürültüsü Varyans Değerleri Kullanıldığında Sensör Sayısının Genişletilmiş Kalman Filtre Algoritmasının Kestirim Hatası Üzerindeki Etkilerinin İncelenmesi

Farklı ölçüm gürültüsü varyans değerleri kullanıldığında, sensör sayılarının GKF algoritmasının kestirim hatası üzerindeki etkileri, 4.5 numaralı bölümün alt bölümünde incelenmiştir. Ölçüm gürültü varyans değeri $0,5 \text{ m}^2$ 'den başlayarak, $0,5 \text{ m}^2$ aralıklarla, 25 m^2 'ye kadar arttırılmıştır. Sensör ölçümleri, ölçüm gürültü varyansı küçükten büyüğe doğru sırayla güncellendiğinde, 1. sensörün ölçüm gürültüsü varyansı $0,5 \text{ m}^2$, 2. sensörün ölçüm gürültüsü varyansı 1 m^2 , şeklinde devam edilmiş ve en son 50. sensörün ölçüm gürültü varyansı 25 m^2 olarak alınmıştır. Sensör ölçümleri, ölçüm gürültü varyansı büyükten küçüğe doğru sırayla güncellendiğinde ise 1. sensörün ölçüm gürültüsü varyansı 25 m^2 , 2. sensörün ölçüm gürültüsü varyansı $24,5 \text{ m}^2$ şeklinde devam edilmiş ve en son 50. sensörün ölçüm gürültü varyansı $0,5 \text{ m}^2$ olarak alınmıştır. GKF algoritmasının ölçümle güncelleme aşamasında ise sıralı güncelleme yöntemi kullanılmış ve ilk olarak 1. sensör ölçümleri, daha sonra 2. sensör ölçümleri şeklinde devam edilerek en son 50. sensör ölçümleri kullanılarak elde edilen kestirim değeri, GKF algoritmasının kestirim değeri olarak üretilmiştir. GKF algoritmasının ölçümle güncelleme aşamasında, sensör ölçümleri, ölçüm gürültü varyansı büyükten küçüğe doğru sırayla, karışık sırayla ve küçükten büyüğe doğru sırayla olacak

şekilde 3 farklı şekilde güncellenmiştir. Elde edilen sonuçlar 4.5. numaralı bölümün alt bölümlerinde incelenmiştir. Sensör ölçümleri, ölçüm gürültü varyans değerleri karışık sırayla olacak şekilde güncellendiği durumda kullanılan, sensörlerin ölçüm gürültü varyans değerleri, Çizelge 4.9’da gösterilmiştir.

Çizelge 4.9. Ölçüm gürültüsü varyans değerleri karışık sırayla güncellendiğinde, her bir sensör için kullanılan ölçüm gürültüsü varyans değerleri

Sensörler	1	2	3	4	5	6	7	8	9	10	11	12
Ölçüm Gürültüsü Varyans Değerleri (m^2)	7	0,5	1,5	5	1	21	5,5	25	3	8	15	2,5
Sensörler	13	14	15	16	17	18	19	20	21	22	23	24
Ölçüm Gürültüsü Varyans Değerleri (m^2)	19,5	6,5	2	24	3,5	9	17,5	4,5	16,5	4	23,5	24,5
Sensörler	25	26	27	28	29	30	31	32	33	34	35	36
Ölçüm Gürültüsü Varyans Değerleri (m^2)	20	23	6	14	19	12,5	11,5	8,5	13,5	9,5	20,5	10,5
Sensörler	37	38	39	40	41	42	43	44	45	46	47	48
Ölçüm Gürültüsü Varyans Değerleri (m^2)	13	16	11	21,5	7,5	14,5	12	10	18,5	22,5	15,5	22
Sensörler	49	50										
Ölçüm Gürültüsü Varyans Değerleri (m^2)	17	18										

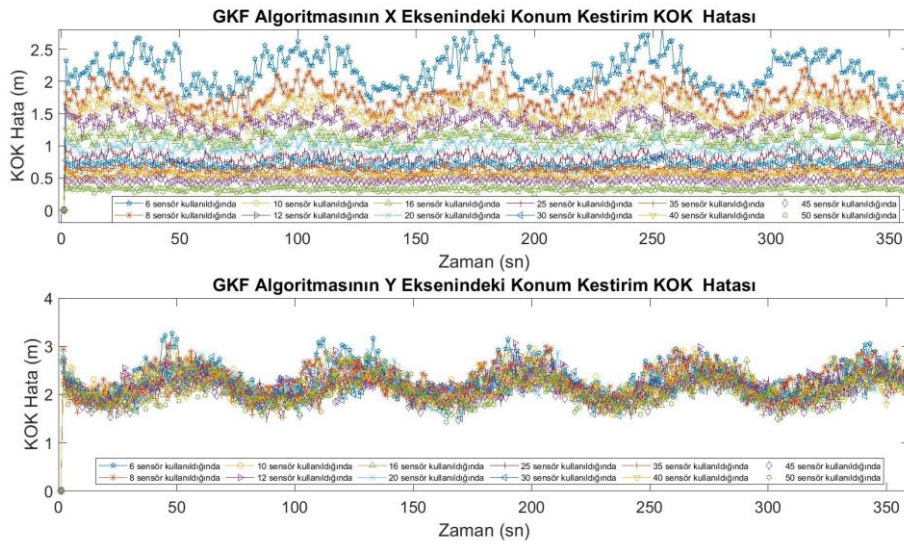
4.5.1. GKF algoritmasında birinci sensör geometrisi kullanıldığında elde edilen sonuçlar

GKF algoritmasında 1. sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültü varyansları büyükten küçüğe doğru sırayla, karışık sırayla ve küçükten büyüğe doğru sırayla

güncellendiğinde elde edilen sonuçlar, 4.5.1 numaralı bölümün alt bölümlerinde sırayla incelenmiştir.

GKF algoritmasında, sensör ölçümleri, ölçüm gürültü varyansları büyükten küçüğe doğru sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının konum kestirim hatası sonuçları

Birinci sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla güncellendiğinde ve sensör sayısı artırıldığında elde edilen, GKF algoritmasının kestirim hatası grafiği, Şekil 4.32’de gösterilmiştir.



Şekil 4.32. Birinci sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

Şekil 4.32’de, sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla güncellendiğinde x ve y eksenlerindeki konum kestirim hataları başlangıçta yüksektir. Bunun sebebi, ölçüm gürültü varyansı büyük olduğu için kestirim hatasının da yüksek olmasıdır. Sensör sayısı artırıldığında ve sensörlerin ölçüm gürültü varyansları azaldıkça, daha doğru ölçümlerle güncelleme yapılarak kestirim üretildiğinden, x eksenindeki kestirim hatası oldukça azalmıştır. Bu durumda, kestirim hatasının azalış miktarı, ölçüm gürültü varyansı küçükten büyüğe doğru sırayla güncellendiği durumdaki kestirim hatasının azalma miktarından daha fazladır. Bunun sebebi, ölçümler, ölçüm gürültü varyansı büyükten küçüğe doğru sırayla güncellendiğinde, sensör sayısı arttığında, ölçüm gürültü varyansı daha küçük olan sensörlerle güncelleme yapıldığından, GKF algoritması ölçüm gürültü varyansı küçük olan bu ölçümleri daha fazla kullanmaktadır. Bu nedenle, daha doğru ölçümlerle durum

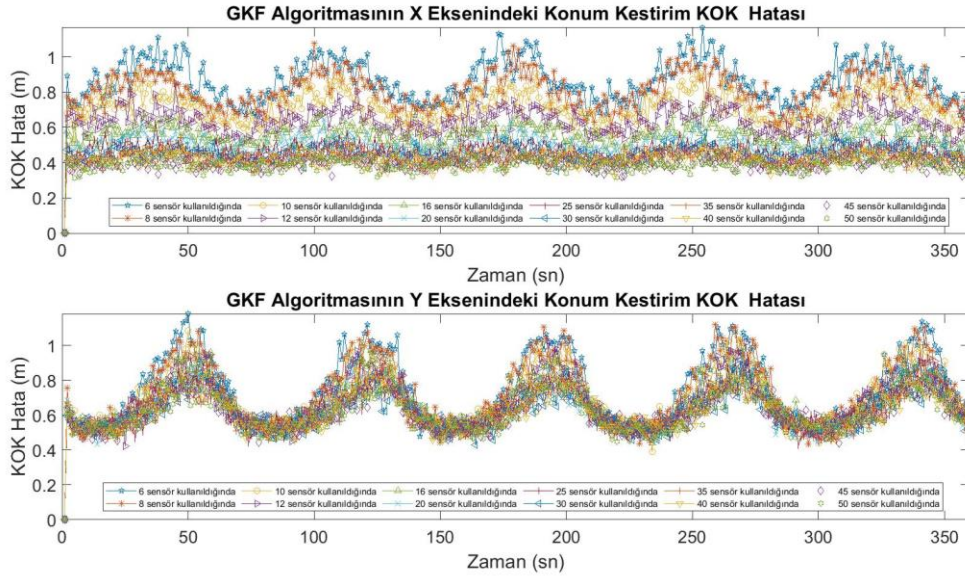
güncellenmesi yapıldığı için kestirim hatası da daha fazla azalmaktadır. Ancak, sensör sayısı arttırılmaya devam edildiğinde, sensörler nesneden uzaklaştığı için, sensörlerden ekstra bilgi elde edilemediğinden, x eksenindeki kestirim hatası daha fazla azalmamıştır. Birinci sensör geometrisinde, her bir sensörün y eksenindeki bileşeni aynı olduğu için, y eksenindeki konum kestirimi için, her bir sensörden, GKF algoritmasına aynı bilgi geldiğinden, y eksenindeki kestirim hatası azalmamıştır.

Çizelge 4.10. GKF algoritmasının, 1. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansı büyükten küçüğe doğru sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları

Sensör Geometrisinde Kullanılan Sensör Sayısı	GKF Algoritmasının X Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Y Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Net Ortalama KOK Konum Kestirim Hatası (m)
6	2,1567	2,3784	3,2106
8	1,7540	2,3068	2,8979
10	1,5240	2,2930	2,7532
12	1,3445	2,2528	2,6235
16	1,1055	2,2003	2,4624
20	0,9524	2,1743	2,3737
25	0,8150	2,1308	2,2813
30	0,7014	2,1257	2,2384
35	0,6203	2,0877	2,1779
40	0,5398	2,0846	2,1533
45	0,4563	2,0354	2,0859
50	0,3262	2,0090	2,0353

GKF algoritmasında, sensör ölçümleri, sensörlerin ölçüm gürültü varyansları karışık sırayla güncellendiğinde, sensör sayısı arttırıldığında elde edilen, GKF algoritmasının konum kestirim hatası sonuçları

Birinci sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları karışık sırayla güncellendiğinde, sensör sayısı arttırıldığında elde edilen, GKF algoritmasının kestirim hata grafiği, Şekil 4.33'te gösterilmiştir.



Şekil 4.33. Birinci sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları karışık sırayla güncellendiğinde, sensör sayısı arttırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

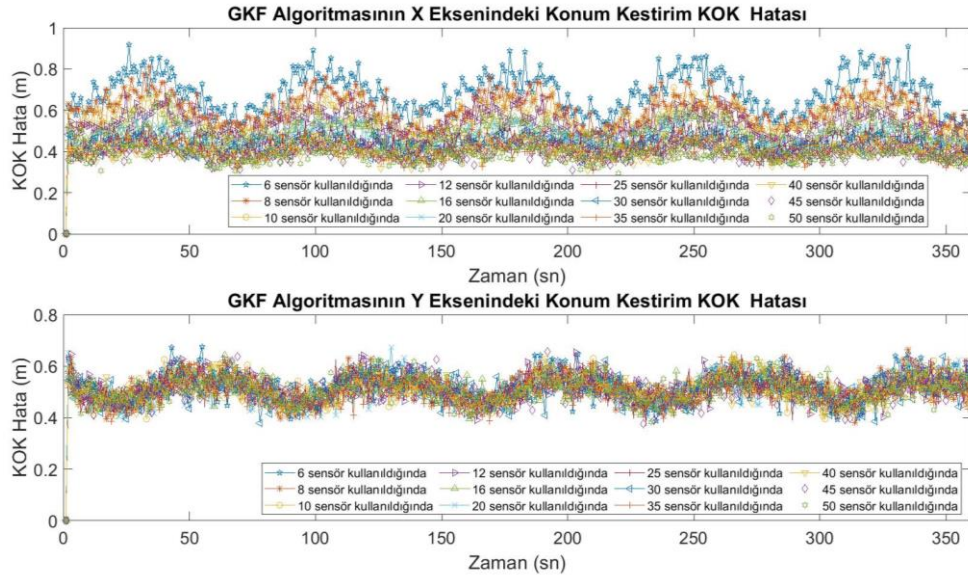
Ölçüm gürültü varyansları karışık sırayla güncellendiğinde, ölçüm gürültü varyansının artıp azalmasına bağlı olarak kestirim hatası da artıp azalmaktadır. GKF algoritmasında durum öngörüm değerleri, ölçüm hatası yüksek olan sensörle güncellendiğinde kestirim hatası artarken, düşük olan sensörle güncellenince kestirim hatası azalmaktadır. O nedenle, sensör ölçümleri karışık sırayla güncellendiğinde, kestirim hatası da ölçüm gürültü varyansına bağlı olarak artıp azaldığından, hata grafiği daha dalgalıdır. Şekil 4.33 incelendiğinde, sensör sayısı yaklaşık 30 sensöre kadar arttıkça, x eksenindeki konum kestirim hatasının azaldığı, ancak sensör sayısı arttırılmaya devam edildiğinde, hatanın daha fazla azalmadığı görülmektedir. Bunun sebebi, 1. sensör geometrisinde, sensör sayısı arttıkça, sensörler nesneden uzaklaştığından, sensörlerden algoritmaya ekstra bilgi gelmediği için kestirim hatası azalmamaktadır. 1. sensör geometrisinde, sensörlerin y eksenindeki konum bileşenleri aynı olduğu için, y eksenindeki konum kestiriminde, her bir sensörden GKF algoritmasına aynı bilgi gelmektedir. Bu nedenle, y eksenindeki konum kestirim hatası, sensör sayısı artsa da değişmemektedir.

Çizelge 4.11. GKF algoritmasının, 1. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları karışık sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları

Sensör Geometrisinde Kullanılan Sensör Sayısı	GKF Algoritmasının X Eksenindeki Ortalama KOK Kestirim Hatası (m)	GKF Algoritmasının Y Eksenindeki Ortalama KOK Kestirim Hatası (m)	GKF Algoritmasının Net Ortalama KOK Kestirim Hatası (m)
6	0,8643	0,7428	1,1396
8	0,8038	0,7259	1,0831
10	0,7090	0,6986	0,9953
12	0,6344	0,6767	0,9275
16	0,5520	0,6577	0,8586
20	0,5009	0,6408	0,8133
25	0,4721	0,6332	0,7898
30	0,4469	0,6274	0,7702
35	0,4325	0,6270	0,7616
40	0,4180	0,6247	0,7516
45	0,4006	0,6195	0,7377
50	0,3900	0,6135	0,7269

GKF algoritmasında, sensör ölçümleri, ölçüm gürültü varyansları küçükten büyüğe doğru sırayla güncellendiğinde, sensör sayısı arttırıldığında elde edilen, GKF algoritmasının konum kestirim hatası sonuçları

Birinci sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiğinde, sensör sayısı arttırıldığında elde edilen, GKF algoritmasının kestirim hata grafiği, Şekil 4.34'te gösterilmiştir.



Şekil 4.34. Birinci sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

Ölçüm gürültüsü varyansı küçükten büyük sensöre doğru ölçümler güncellendiğinde, GKF algoritması, ölçüm gürültü varyansı küçük olan 1. sensörden başlayarak güncelleme yaptığından, kestirim hatası düşüktür. 2. sensörün ölçüm gürültüsü, 1. sensörden yüksek olduğundan, 2. sensör ölçümü geldiğinde GKF algoritması, 1. sensör ölçümüyle elde ettiği kestirime daha fazla güvenerek kestirim üretir. Bu şekilde, sensör sayısı arttığında, eklenen yeni sensörlerin ölçüm gürültü varyansı arttıkça, GKF algoritması kendi kestirimine daha fazla güvenerek, kestirim hatasını azaltır. Dolayısıyla, sensör sayısı arttıkça, ölçüm gürültü varyansı da arttığı için, GKF algoritması sensör sayısı artsa bile, sensör ölçümlerini daha az kullanarak kestirim ürettiğinden, sensör sayısının artmasına bağlı olarak kestirim hatasındaki azalma miktarı daha azdır.

Çizelge 4.12. GKF algoritmasının, 1. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansı küçükten büyüğe doğru sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları

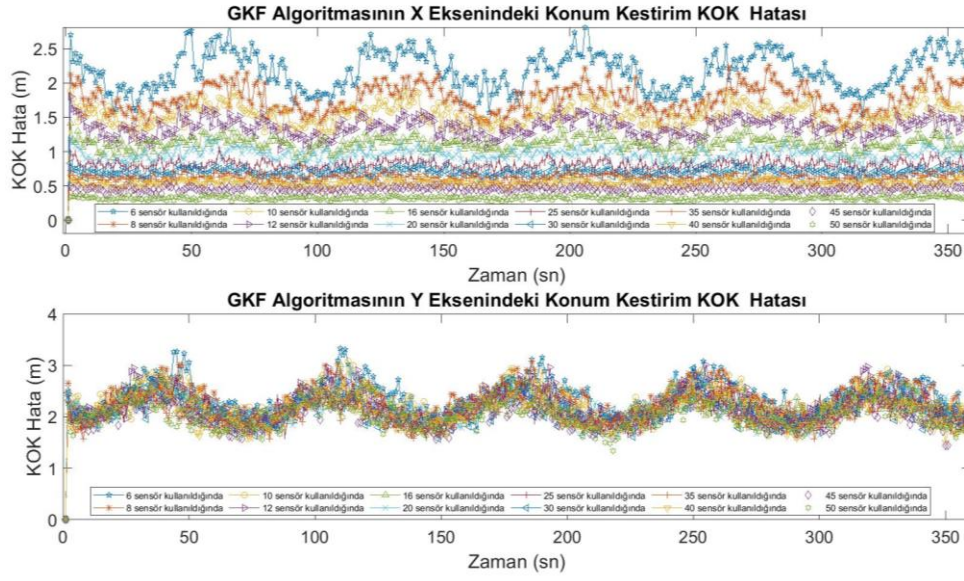
Sensör Geometrisinde Kullanılan Sensör Sayısı	GKF Algoritmasının X Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Y Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Net Ortalama KOK Konum Kestirim Hatası (m)
6	0,6824	0,5191	0,8573
8	0,6013	0,5147	0,7915
10	0,5583	0,5141	0,7589
12	0,5279	0,5138	0,7366
16	0,4891	0,5102	0,7067
20	0,4632	0,5099	0,6888
25	0,4390	0,5076	0,6711
30	0,4246	0,5070	0,6613
35	0,4079	0,5068	0,6505
40	0,4026	0,5067	0,6471
45	0,3930	0,5060	0,6407
50	0,3849	0,5035	0,6337

4.5.2. GKF algoritmasında ikinci sensör geometrisi kullanıldığında elde edilen sonuçlar

GKF algoritmasında 2. sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültü varyansları büyükten küçüğe doğru sırayla, karışık sırayla ve küçükten büyüğe doğru sırayla güncellendiğinde elde edilen sonuçlar, 4.5.2 numaralı bölümün alt bölümlerinde sırayla incelenmiştir.

GKF algoritmasında, sensör ölçümleri, sensörlerin ölçüm gürültü varyansları büyükten küçüğe doğru sırayla güncellendiğinde, sensör sayısı arttırıldığında elde edilen, GKF algoritmasının konum kestirim hatası sonuçları

İkinci sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla güncellendiği durumda, sensör sayısı arttırıldığında elde edilen, GKF algoritmasının kestirim hatası grafiği, Şekil 4.35'te gösterilmiştir.



Şekil 4.35. İkinci sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

Şekil 4.35 incelendiğinde, 2. sensör geometrisinde kullanılan sensör sayısı 30'a kadar artırıldığında, GKF algoritmasının x eksenindeki konum kestirim hatasının azaldığı, ancak daha fazla artırıldığında ise daha fazla azalmayarak, yaklaşık sabitlendiği görülmektedir. Bunun sebebi, sensör sayısı arttırılmaya devam edildiğinde, eklenen yeni sensörler nesneden uzaklaştığı için hedef nesneden daha fazla bilgi elde edilemediği için, x eksenindeki konum kestirim hatası daha fazla azalmamıştır. Sensörler y eksenine dik şekilde yerleştirildiği için, her bir sensörün y eksenindeki bileşeni aynıdır. Her bir sensörden, GKF algoritmasına aynı bilgi geldiği için, sensör sayısı artsa da y eksenindeki kestirim hatası aynı kalmıştır.

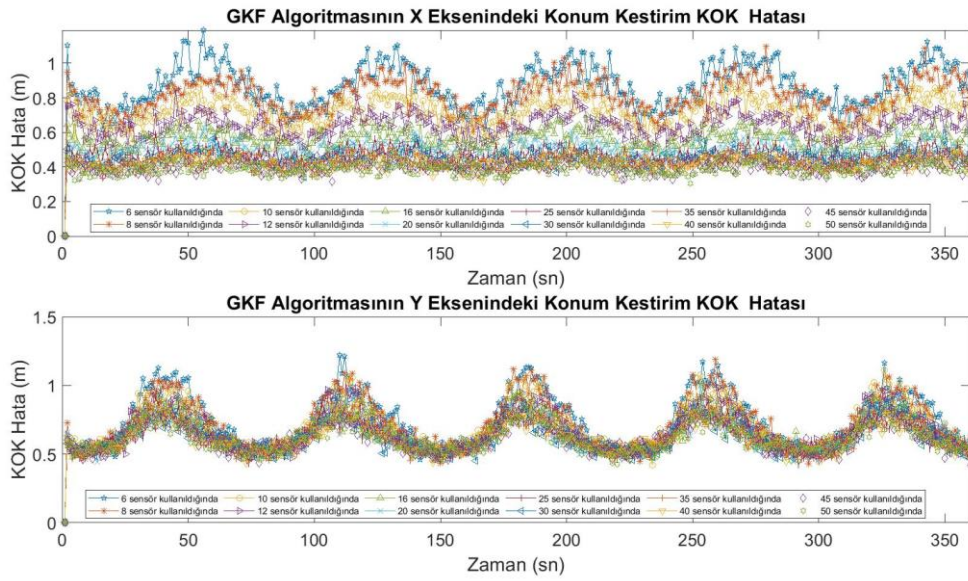
Çizelge 4.13. GKF algoritmasının, 2. sensör geometrisinde sensör sayısı artırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansı büyükten küçüğe doğru sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları

Sensör Geometrisinde Kullanılan Sensör Sayısı	GKF Algoritmasının X Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Y Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Net Ortalama KOK Konum Kestirim Hatası (m)
6	2,1713	2,3686	3,2132
8	1,7662	2,3019	2,9014
10	1,5301	2,2929	2,7565
12	1,3499	2,2471	2,6213
16	1,1091	2,1903	2,4551
20	0,9569	2,1638	2,3659
25	0,8155	2,1236	2,2748
30	0,7022	2,1118	2,2254
35	0,6224	2,0763	2,1675
40	0,5404	2,0788	2,1479
45	0,4566	2,0266	2,0774
50	0,3271	1,9986	2,0251

Sensör ölçümleri, ölçüm gürültüsü varyansları büyük olan sensörden başlayarak güncellendiğinde, başlangıçta 6 adet sensör kullanıldığında, sensörlerin yüksek ölçüm hatası sebebiyle kestirim hatası da yüksektir. Daha sonra, sensör sayısı artırıldığında ve sensörlerin ölçüm gürültü varyansları azaldıkça, sensörlerden daha yüksek doğrulukla ölçüm alınmakta ve bu nedenle algoritma, Kalman kazancını artırarak, sensör ölçümünü daha çok kullanmaktadır. Böylece, durum öngörümü değerini sensör ölçümüyle daha fazla düzelterek kestirim ürettiği için, sensör sayısı arttıkça kestirim hatası da oldukça azalmıştır.

GKF algoritmasında, sensör ölçümleri, ölçüm gürültü varyansları karışık sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının konum kestirim hatası sonuçları

İkinci sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları karışık sırayla güncellendiği durumda, sensör sayısı artırıldığında elde edilen kestirim hata grafiği, Şekil 4.36'da gösterilmiştir.



Şekil 4.36. İkinci sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları karışık sırayla güncellendiğinde, sensör sayısı arttırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

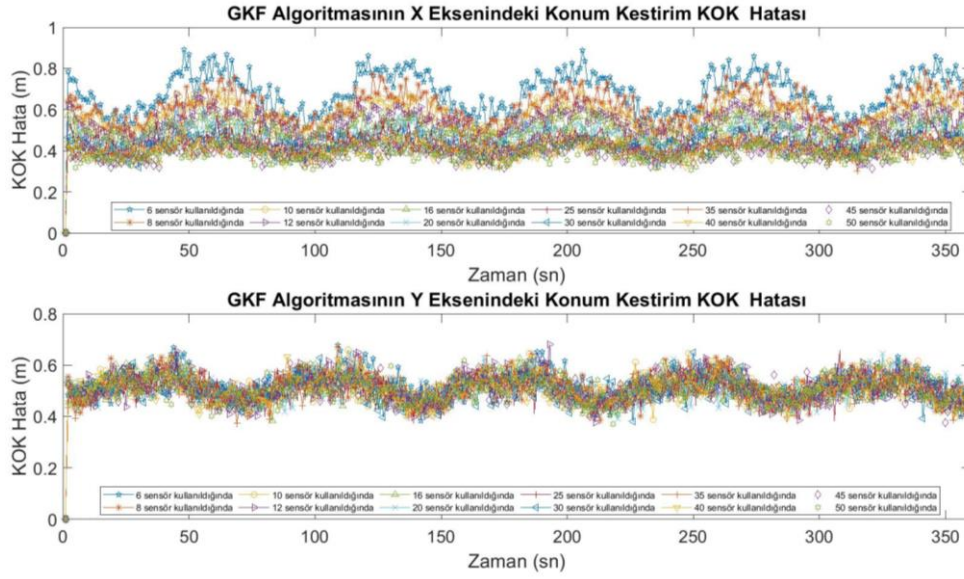
Şekil 4.36'da verilen grafik incelendiğinde, sensör ölçümleri, ölçüm gürültü varyansları karışık sırayla güncellendiğinde, kestirim hatasının, daha fazla dalgalandığı görülmektedir. Bunun sebebi, ölçüm gürültü varyanslarının karışık sırayla güncellenmesi sebebiyle, ölçüm gürültü varyansının artıp azalmasına bağlı olarak kestirim hatasının da artıp azalmasıdır. Ölçüm gürültü varyansı arttığında, ölçüm hatası da artmaktadır. Bu durumda, ölçüm hatasından kaynaklı kestirim hatası artmaktadır. Ölçüm gürültü varyansı azaldığında ise ölçüm hatası azalmakta ve filtre, kestirim üretmek için sensör ölçümünden gelen bilgiyi daha fazla kullanarak kestirim üretmektedir. Bu durumda ise kestirim hatası azalmaktadır.

Çizelge 4.14. GKF algoritmasının, 2. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansı karışık sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları

Sensör Geometrisinde Kullanılan Sensör Sayısı	GKF Algoritmasının X Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Y Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Net Ortalama KOK Konum Kestirim Hatası (m)
6	0,8703	0,7387	1,1415
8	0,8089	0,7227	1,0847
10	0,7109	0,6971	0,9956
12	0,6368	0,6742	0,9273
16	0,5543	0,6540	0,8573
20	0,5029	0,6386	0,8128
25	0,4724	0,6307	0,7880
30	0,4474	0,6247	0,7683
35	0,4339	0,6211	0,7576
40	0,4190	0,6239	0,7515
45	0,4010	0,6162	0,7351
50	0,3912	0,6106	0,7251

GKF algoritmasında, sensör ölçümleri, ölçüm gürültü varyansları küçükten büyüğe doğru sırayla güncellendiğinde, sensör sayısı arttırıldığında elde edilen, GKF algoritmasının konum kestirim hatası sonuçları

İkinci sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiği durumda ve sensör sayısı arttırıldığında elde edilen, GKF algoritmasının kestirim hata grafiği, Şekil 4.37’de gösterilmiştir.



Şekil 4.37. İkinci sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

Ölçüm gürültüsü varyansı küçükten büyük sensöre doğru ölçümler güncellendiğinde, GKF algoritması, ölçüm gürültü varyansı küçük olan 1. sensörden başlayarak güncelleme yaptığından, kestirim hatası düşüktür. 1. sensör ölçümü kullanılarak elde edilen kestirim değeri, 2. sensör ölçümü kullanıldığında, algoritmanın durum öngörüm değeri olmakta ve 2. sensör ölçümüyle güncellenerek kestirim üretilmektedir. 2. sensörün ölçüm gürültüsü, 1. sensörden yüksek olduğundan, 2. sensör ölçümü geldiğinde Kalman kazancı azalmakta ve algoritma, 1. sensör ölçümüyle elde ettiği kestirime daha fazla güvenerek kestirim üretmektedir. Sensör sayısı arttıkça ve yeni sensörlerin ölçüm gürültü varyansı arttıkça, GKF algoritması kendi kestirimine daha fazla güvenerek kestirim hatasını azaltır. Bu nedenle sensör sayısı arttıkça kestirim hatası azalmış, ancak sensör sayısı daha fazla arttırıldığında eklenen yeni sensörler nesneden uzaklaştığı için ekstra bilgi sağlamadığından, kestirim hatası daha fazla azalmamıştır.

Çizelge 4.15. GKF algoritmasının, 2. sensör geometrisinde sensör sayısı artırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları

Sensör Geometrisinde Kullanılan Sensör Sayısı	GKF Algoritmasının X Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Y Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Net Ortalama KOK Konum Kestirim Hatası (m)
6	0,6797	0,5218	0,8569
8	0,6047	0,5150	0,7942
10	0,5629	0,5135	0,7619
12	0,5302	0,5116	0,7367
16	0,4897	0,5088	0,7061
20	0,4644	0,5073	0,6877
25	0,4415	0,5070	0,6722
30	0,4228	0,5046	0,6583
35	0,4153	0,5043	0,6532
40	0,4047	0,5058	0,6477
45	0,3944	0,5040	0,6399
50	0,3861	0,5020	0,6333

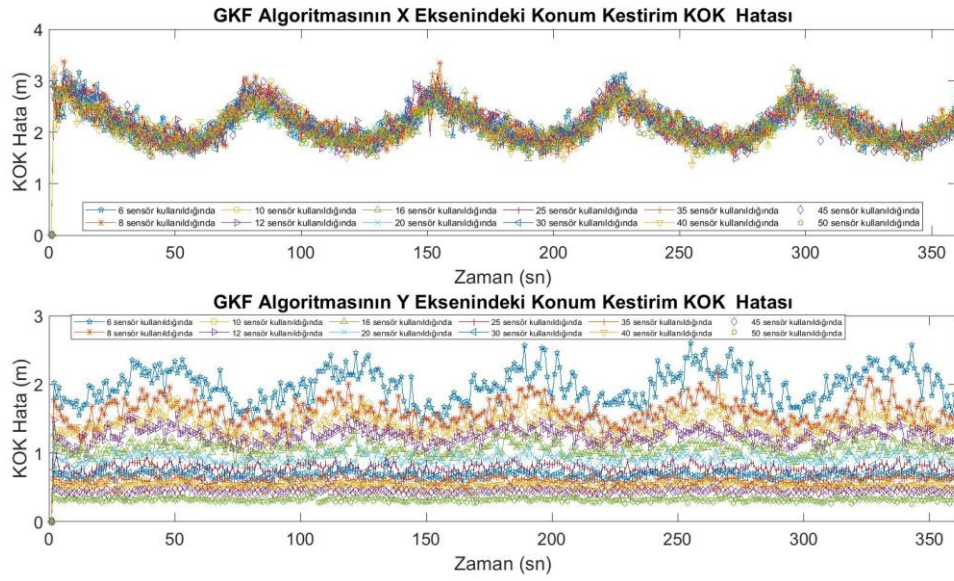
4.5.3. GKF algoritmasında üçüncü sensör geometrisi kullanıldığında elde edilen sonuçlar

GKF algoritmasında 3. sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe, karışık ve küçükten büyüğe doğru sırayla güncellendiğinde elde edilen kestirim hata sonuçları, 4.5.3 numaralı bölümün alt bölümlerinde incelenmiştir.

GKF algoritmasında, sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının konum kestirim hatası sonuçları

GKF algoritmasında, üçüncü sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe sırayla güncellendiği durumda, sensör sayısı artırıldığında elde edilen, kestirim hata grafiği, Şekil 4.38’de gösterilmiştir. Sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe, karışık ve küçükten büyüğe doğru sırayla güncellendiği her bir durum için kestirim hata grafikleri incelendiğinde, GKF algoritmasının x eksenindeki kestirim hatası, sensör sayısı artsa da aynı kalmıştır. Bunun

sebebi, 3. sensör geometrisinde, sensörler x eksenine dik şekilde yerleştirildiği için, her bir sensörün x eksenindeki konum bileşenleri aynıdır ve her bir sensörden GKF algoritmasına aynı bilginin gelmesidir. y eksenindeki konum kestirim hata grafikleri incelendiğinde ise, y eksenindeki konum kestirim hatası sensör sayısı artırıldığında azalmakta, ancak sensör sayısı arttırılmaya devam edildiğinde, hata daha fazla azalmamaktadır. Bunun sebebi, üçüncü sensör geometrisinde sensör sayısı arttıkça, sensörler nesneden uzaklaştığından, sensör geometrisine eklenen yeni sensörlerden algoritmaya ekstra bilginin gelmemesidir.



Şekil 4.38. Üçüncü sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

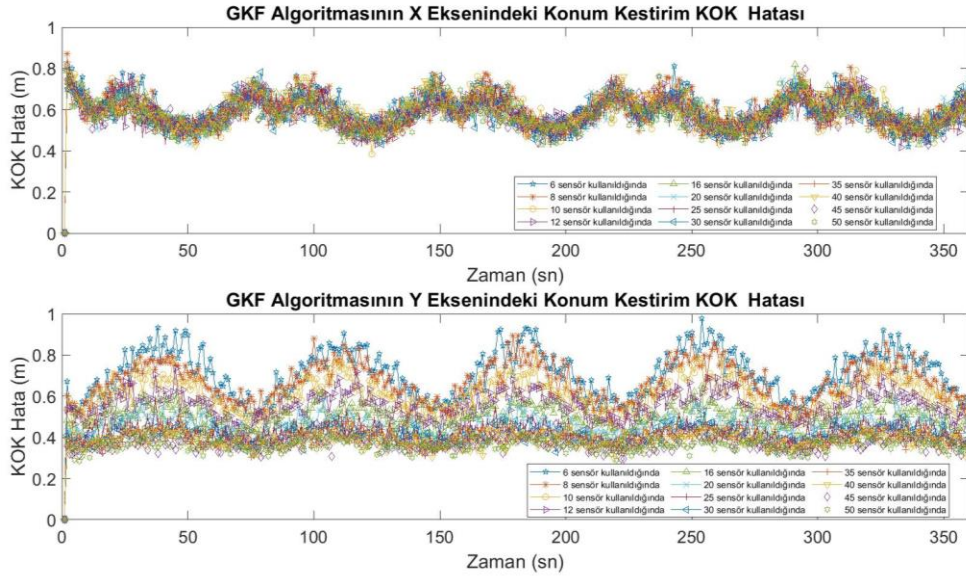
3. sensör geometrisinde sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla güncellendiğinde x ve y eksenlerindeki konum kestirim hataları başlangıçta yüksektir. Bunun sebebi, ölçüm gürültü varyansı büyük olduğu için kestirim hatasının da yüksek olmasıdır. Sensör sayısı artırıldığında, sensörlerden algoritmaya, konumu tespit edilmek istenen nesne hakkında daha fazla bilgi geldiğinden ve sensörlerin ölçüm gürültü varyansları azaldıkça, daha doğru ölçümlerle güncelleme yapılarak kestirim üretildiğinden, y eksenindeki kestirim hatası oldukça azalmıştır. Ancak, sensör sayısı daha çok arttığında, sensörler nesneden uzaklaştığından, sensörlerden ekstra bilgi elde edilemediği için, y eksenindeki kestirim hatası daha fazla azalmamıştır.

Çizelge 4.16. GKF algoritmasının, 3. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları

Sensör Geometrisinde Kullanılan Sensör Sayısı	GKF Algoritmasının X Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Y Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Net Ortalama KOK Konum Kestirim Hatası (m)
6	2,1884	1,9770	2,9491
8	2,2110	1,5965	2,7271
10	2,1847	1,3988	2,5941
12	2,1969	1,2543	2,5297
16	2,1550	1,0487	2,3966
20	2,1296	0,9146	2,3176
25	2,1460	0,7807	2,2835
30	2,1433	0,6805	2,2487
35	2,1313	0,6033	2,2150
40	2,1180	0,5224	2,1636
45	2,0983	0,4423	2,1623
50	2,0719	0,3187	2,0962

GKF algoritmasında, sensör ölçümleri, ölçüm gürültü varyansları karışık sırayla güncellendiğinde, sensör sayısı arttırıldığında elde edilen, GKF algoritmasının konum kestirim hatası sonuçları

Üçüncü sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları karışık sırayla güncellendiğinde, sensör sayısı arttırıldığında elde edilen, kestirim hata grafiği, Şekil 4.39’da gösterilmiştir.



Şekil 4.39. Üçüncü sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları karışık sırayla güncellendiğinde, sensör sayısı arttırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

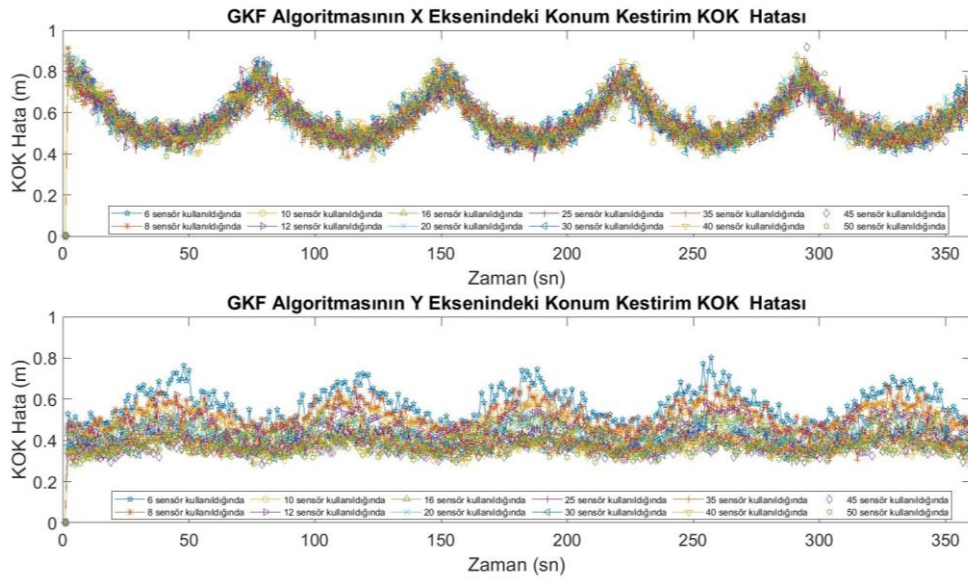
Şekil 4.39’da verilen grafik incelendiğinde, sensör ölçümleri, ölçüm gürültü varyansları karışık sırayla güncellendiğinde, kestirim hatasının, ölçüm gürültü varyansları küçükten büyüğe doğru sıralandığında elde edilen kestirim hatalarıyla karşılaştırıldığında, daha fazla dalgalandığı görülmektedir. Çünkü sensör ölçümleri, ölçüm gürültü varyansları karışık sırayla güncellendiğinde, ölçüm gürültü varyansının artıp azalmasına bağlı olarak ölçüm hatası da artıp azalmaktadır. Dolayısıyla, ölçüm hatası arttığında, ölçüm hatasından kaynaklı kestirim hatası artmakta, ölçüm hatası azaldığında ise, GKF algoritması bu ölçümleri daha çok kullanarak kestirim üreterek, kestirim hatasını azaltmaktadır. Bu nedenle, ölçüm gürültü varyansının artıp azalmasına bağlı olarak, kestirim hatası da artıp azaldığından, kestirim hata grafiği daha dalgalıdır.

Çizelge 4.17. GKF algoritmasının, 3. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları karışık sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları

Sensör Geometrisinde Kullanılan Sensör Sayısı	GKF Algoritmasının X Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Y Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Net Ortalama KOK Konum Kestirim Hatası (m)
6	0,6046	0,7066	0,9299
8	0,6013	0,6653	0,8967
10	0,5972	0,6013	0,8474
12	0,5913	0,5482	0,8063
16	0,5859	0,4905	0,7641
20	0,5827	0,4509	0,7367
25	0,5819	0,4272	0,7218
30	0,5807	0,4079	0,7096
35	0,5856	0,3984	0,7016
40	0,5792	0,3865	0,6931
45	0,5767	0,3723	0,6864
50	0,5754	0,3643	0,6810

GKF algoritmasında, sensör ölçümleri, sensörlerin ölçüm gürültü varyansları küçükten büyüğe doğru sırayla güncellendiğinde, sensör sayısı arttırıldığında elde edilen, GKF algoritmasının konum kestirim hatası sonuçları

Üçüncü sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiği durumda ve sensör sayısı arttırıldığında elde edilen, GKF algoritmasının kestirim hatası grafiği, Şekil 4.40'ta gösterilmiştir.



Şekil 4.40. Üçüncü sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

Ölçüm gürültüsü varyansı küçükten büyük sensöre doğru ölçümler güncellendiğinde, GKF algoritması, başlangıçta ölçüm gürültü varyansı küçük olan 1. sensör ölçümünü kullandığı için kestirim hatası düşüktür. 2. sensörden ölçüm alındığında, 1. sensör ölçümü kullanılarak elde edilen kestirim değeri, algoritmanın durum öngörüm değeri olmakta ve 2. sensör ölçümüyle güncellenerek kestirim üretilmektedir. Ancak, 2. sensörün ölçüm gürültüsü, 1. sensörden yüksek olduğundan, 2. sensör ölçümü geldiğinde Kalman kazancı azalmakta ve algoritma 1. sensör ölçümüyle elde ettiği kestirime daha fazla ağırlık vererek kestirim üreterek, kestirim hatasını azaltmaktadır. Ancak sensör sayısı daha fazla artırıldığında eklenen yeni sensörler nesneden uzaklaştığı için ekstra bilgi sağlamadığından, kestirim hatası daha fazla azalmamıştır. Ayrıca, ölçümler, ölçüm gürültü varyansı büyüğe doğru sırayla güncellendiğinde, sensör sayısı arttığında kestirim hatasındaki azalma miktarı, ölçüm gürültü varyansı küçükten büyüğe doğru sırayla güncellendiği durumdan daha fazladır. Bunun sebebi, ölçüm gürültü varyansı küçükten büyüğe doğru ölçümler güncellendiğinde, sensör sayısı artınca, sensör geometrisine eklenen yeni sensörlerin ölçüm gürültü varyansları da arttığından, GKF algoritması yeni eklenen sensör ölçümlerini daha az kullanarak kestirim üretmektedir. Bu nedenle, kestirim hatası sensör ölçümleriyle fazla azaltılamadığından, sensör sayısı artışı kestirim hatasını fazla azaltmamıştır.

Çizelge 4.18. GKF algoritmasının, 3. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları

Sensör Geometrisinde Kullanılan Sensör Sayısı	GKF Algoritmasının X Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Y Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Net Ortalama KOK Konum Kestirim Hatası (m)
6	0,5839	0,5685	0,8149
8	0,5759	0,5158	0,7731
10	0,5752	0,4876	0,7540
12	0,5742	0,4634	0,7378
16	0,5719	0,4352	0,7186
20	0,5708	0,4156	0,7060
25	0,5706	0,3982	0,6958
30	0,5681	0,3849	0,6900
35	0,5679	0,3796	0,6862
40	0,5663	0,3710	0,6770
45	0,5660	0,3632	0,6725
50	0,5654	0,3570	0,6687

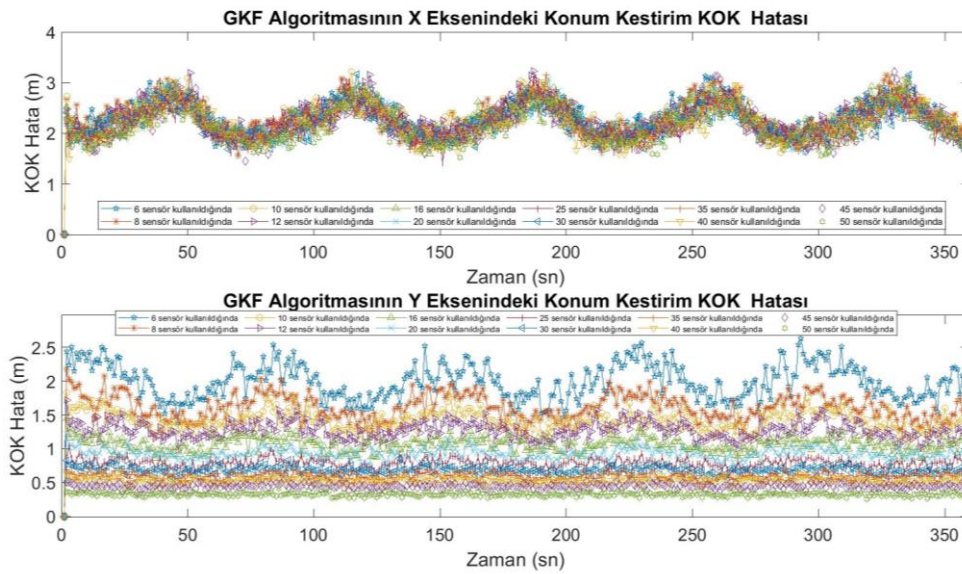
4.5.4. GKF algoritmasında dördüncü sensör geometrisi kullanıldığında elde edilen sonuçlar

GKF algoritmasında 4. sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültü varyansları büyükten küçüğe doğru sırayla, karışık sırayla ve küçükten büyüğe doğru sırayla güncellendiğinde elde edilen kestirim hatası sonuçları, 4.5.4 numaralı bölümün alt bölümlerinde incelenmiştir. Sensör ölçümlerinin, ölçüm gürültü varyansları büyükten küçüğe, karışık ve küçükten büyüğe doğru sırayla güncellendiği her bir durum için, GKF algoritmasının x eksenindeki konum kestirim hatası, sensör sayısı artışından etkilenmeyerek aynı kalmıştır. Çünkü 3. sensör geometrisinde, sensörler x eksenine dik olarak yerleştirildiğinden, her bir sensörün x eksenindeki konum bileşenleri aynıdır. Bu nedenle, her bir sensörden GKF algoritmasına aynı bilgi gelmektedir. y eksenindeki konum kestirim hatasıysa, sensör sayısı arttırıldığında azalmakta, ancak sensör sayısı arttırılmaya devam edildiğinde, hata daha fazla azalmamaktadır. Bunun sebebi ise, dördüncü sensör

geometrisinde sensör sayısı arttıkça, sensörler nesneden uzaklaştığından, sensör geometrisine eklenen yeni sensörlerden algoritmaya ekstra bilginin gelmemesidir.

GKF algoritmasında, sensör ölçümleri, ölçüm gürültü varyansları büyükten küçüğe doğru sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının konum kestirim hatası sonuçları

GKF algoritmasında, dördüncü sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla güncellendiği durumda, sensör sayısı artırıldığında elde edilen, kestirim hata grafiği, Şekil 4.41’de gösterilmiştir.



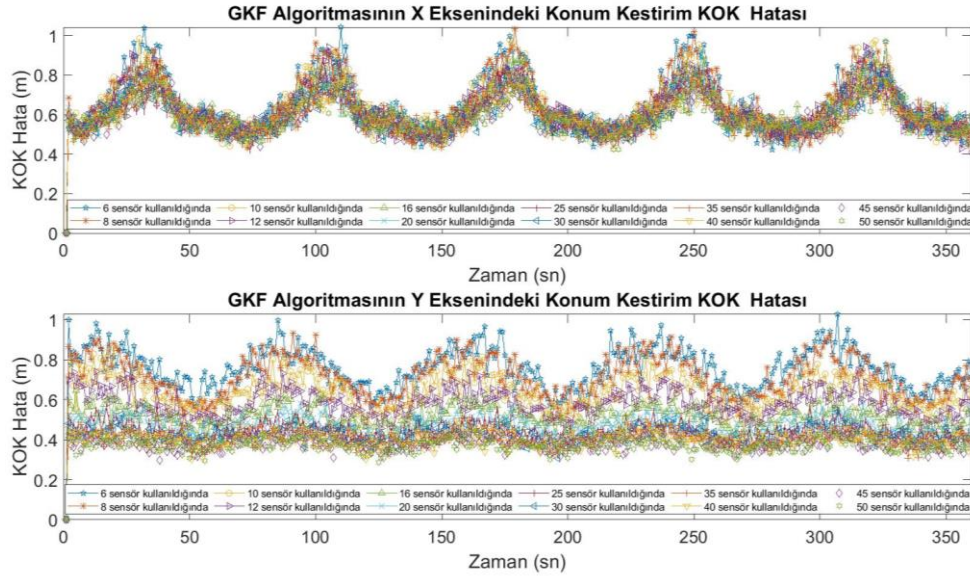
Şekil 4.41. Dördüncü sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

Sensörün ölçüm gürültü varyansı yüksek olduğunda, ölçüm hatası da yüksek olmaktadır. Ölçüm gürültü varyansı azaldığında ise, sensörün ölçüm hatası da azalmaktadır. GKF algoritmasında, ölçüm gürültü varyansı büyük olan sensörden başlayarak ölçümler güncellendiğinde, sensörün ölçüm hatası yüksek olduğu için, kestirim hatası da yüksek olmaktadır. Ancak, sensör sayısı artıp, sensör geometrisine eklenen yeni sensörlerin ölçüm gürültü varyansları azaldıkça, ölçüm hatası daha düşük olan sensör ölçümleri filtreye beslenmektedir. Bu nedenle, y eksenindeki konum kestirim hatası önemli ölçüde azalmaktadır. Çünkü ölçüm gürültü varyansı küçük olan sensör ölçümleri kullanıldığından, Kalman kazancı artmakta ve algoritma bu ölçümleri daha çok kullanarak kestirim üreterek, kestirim hatasını daha çok azaltmaktadır.

Çizelge 4.19. GKF algoritmasının, 4. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları

Sensör Geometrisinde Kullanılan Sensör Sayısı	GKF Algoritmasının X Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Y Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Net Ortalama KOK Konum Kestirim Hatası (m)
6	2,3272	1,9915	3,0629
8	2,3221	1,6124	2,8270
10	2,3034	1,4099	2,7006
12	2,2981	1,2635	2,6225
16	2,2584	1,0560	2,4931
20	2,2167	0,9212	2,4005
25	2,2288	0,7844	2,3628
30	2,2127	0,6827	2,3156
35	2,2006	0,5949	2,2795
40	2,1987	0,5242	2,2603
45	2,1855	0,4432	2,2299
50	2,1382	0,3192	2,1619

GKF algoritmasında, sensör ölçümleri, ölçüm gürültü varyansları karışık sırayla güncellendiğinde, sensör sayısı arttırıldığında elde edilen, konum kestirim hatası sonuçları Dördüncü sensör geometrisi kullanıldığında ve ölçüm gürültüsü varyans değerleri karışık sırayla ölçümler güncellendiğinde, sensör sayısı arttırılarak elde edilen, GKF algoritmasının konum kestirim hatası grafiği, Şekil 4.42’de gösterilmiştir.



Şekil 4.42. Dördüncü sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları karışık sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

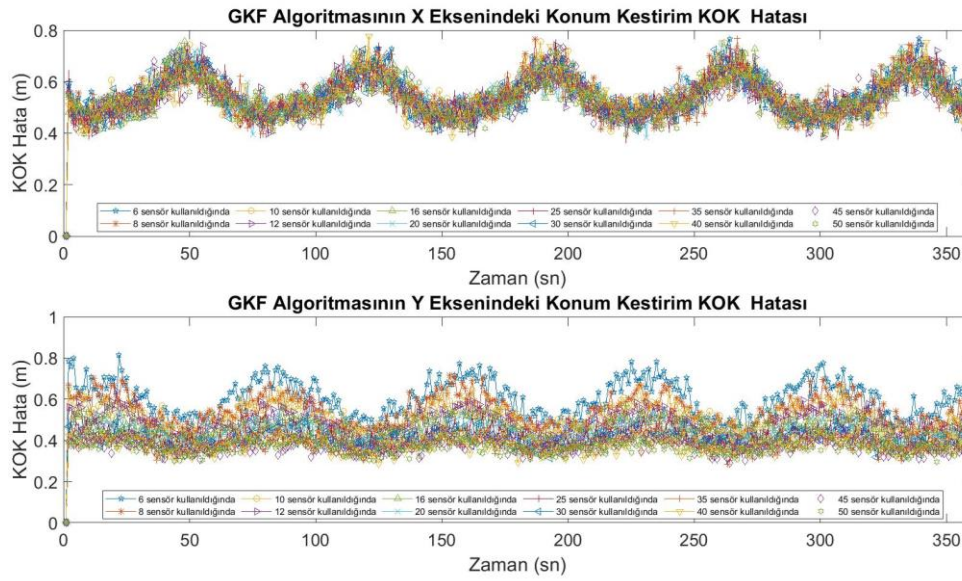
Sensör ölçümleri, ölçüm gürültü varyansları karışık sırayla güncellendiğinde, ölçüm gürültü varyansının artıp azalmasına bağlı olarak ölçüm hatası da artıp azalmaktadır. Ölçüm gürültü varyansı arttığında ölçüm hatası artmakta, ölçüm gürültü varyansı azaldığında ise ölçüm hatası azalmaktadır. Dolayısıyla, ölçüm hatası arttığında, ölçüm hatasından kaynaklı kestirim hatası artmakta, ölçüm hatası azaldığında ise, GKF algoritması bu ölçümleri daha çok kullanarak kestirim üreterek, kestirim hatasını azaltmaktadır.

Çizelge 4.20. GKF algoritmasının, 4. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları karışık sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları

Sensör Geometrisinde Kullanılan Sensör Sayısı	GKF Algoritmasının X Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Y Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Net Ortalama KOK Konum Kestirim Hatası (m)
6	0,6559	0,7569	1,0015
8	0,6495	0,7115	0,9633
10	0,6372	0,6376	0,9014
12	0,6258	0,5786	0,8523
16	0,6140	0,5115	0,7991
20	0,6067	0,4693	0,7670
25	0,6039	0,4440	0,7495
30	0,6015	0,4220	0,7347
35	0,6015	0,4102	0,7280
40	0,5963	0,3976	0,7167
45	0,5966	0,3820	0,7084
50	0,5912	0,3728	0,6989

GKF algoritmasında, sensör ölçümleri, ölçüm gürültü varyansları küçükten büyüğe doğru sırayla güncellendiğinde, sensör sayısı arttırıldığında elde edilen, GKF algoritmasının konum kestirim hatası sonuçları

Dördüncü sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiği durumda ve sensör sayısı arttırıldığında elde edilen, GKF algoritmasının konum kestirim hatası grafiği, Şekil 4.43'te gösterilmiştir.



Şekil 4.43. Dördüncü sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

Ölçüm gürültüsü varyansı küçükten büyük sensöre doğru ölçümler güncellendiğinde, GKF algoritması, başlangıçta ölçüm gürültü varyansı küçük olan 1. sensör ölçümünü kullandığı için kestirim hatası düşüktür. 2. sensörden ölçüm alındığında, 1. sensör ölçümü kullanılarak elde edilen kestirim değeri, algoritmanın durum öngörüm değeri olmakta ve 2. sensör ölçümüyle güncellenerek kestirim üretilmektedir. Ancak, 2. sensörün ölçüm gürültüsü, 1. sensörden yüksek olduğundan, 2. sensör ölçümü geldiğinde Kalman kazancı azalmakta ve algoritma 1. sensör ölçümüyle elde ettiği kestirime daha fazla ağırlık vererek kestirim üreterek, kestirim hatasını azaltmaktadır. Fakat ölçüm gürültü varyansı büyük olan sensörden, küçük olana sensöre doğru ölçümler güncellendiğinde, sensör sayısı arttıkça kestirim hatası daha fazla azalmıştır. Çünkü ölçüm gürültü varyansı küçük olan sensörlerden ölçüm hatası daha düşük olan ölçümler geldikçe, Kalman kazancı artmakta ve bu ölçümler daha fazla kullanılarak kestirim hatası daha çok azaltılmaktadır.

Çizelge 4.21. GKF algoritmasının, 4. sensör geometrisinde sensör sayısı artırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları

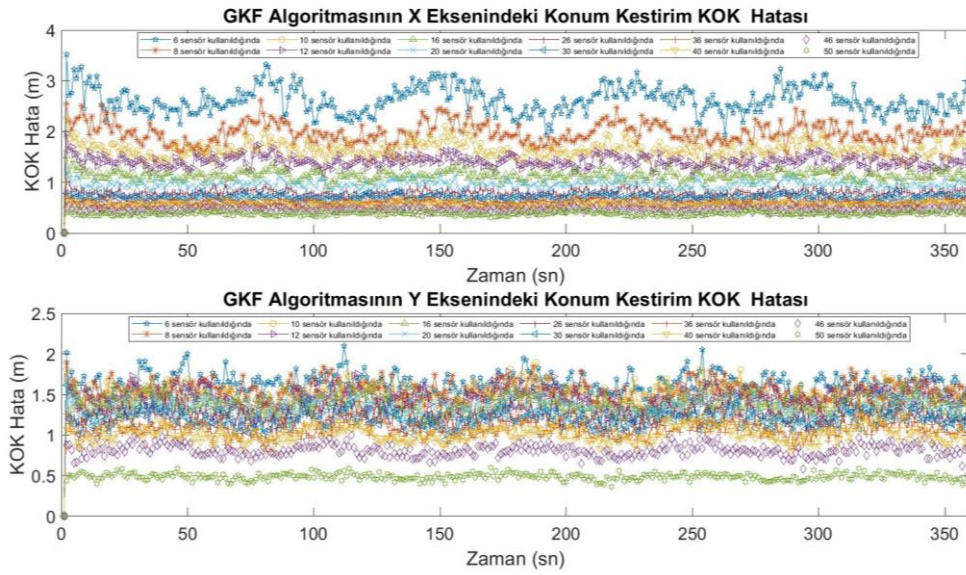
Sensör Geometrisinde Kullanılan Sensör Sayısı	GKF Algoritmasının X Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Y Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Net Ortalama KOK Konum Kestirim Hatası (m)
6	0,5511	0,5976	0,8129
8	0,5480	0,5383	0,7681
10	0,5434	0,5080	0,7438
12	0,5455	0,4780	0,7253
16	0,5427	0,4530	0,7070
20	0,5416	0,4317	0,6926
25	0,5419	0,4124	0,6809
30	0,5404	0,4001	0,6723
35	0,5407	0,3867	0,6647
40	0,5399	0,3791	0,6597
45	0,5398	0,3735	0,6564
50	0,5350	0,3673	0,6489

4.5.5. GKF algoritmasında beşinci sensör geometrisi kullanıldığında elde edilen sonuçlar

GKF algoritmasında 5. sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla, karışık sırayla ve küçükten büyüğe doğru sırayla güncellendiğinde elde edilen kestirim hatası sonuçları, 4.5.5 numaralı bölümün alt bölümlerinde incelenmiştir.

GKF algoritmasında, sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının konum kestirim hatası sonuçları

GKF algoritmasında, beşinci sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe sırayla güncellendiği durumda, sensör sayısı artırıldığında elde edilen kestirim hata grafiği, Şekil 4.44'te gösterilmiştir.



Şekil 4.44. Beşinci sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

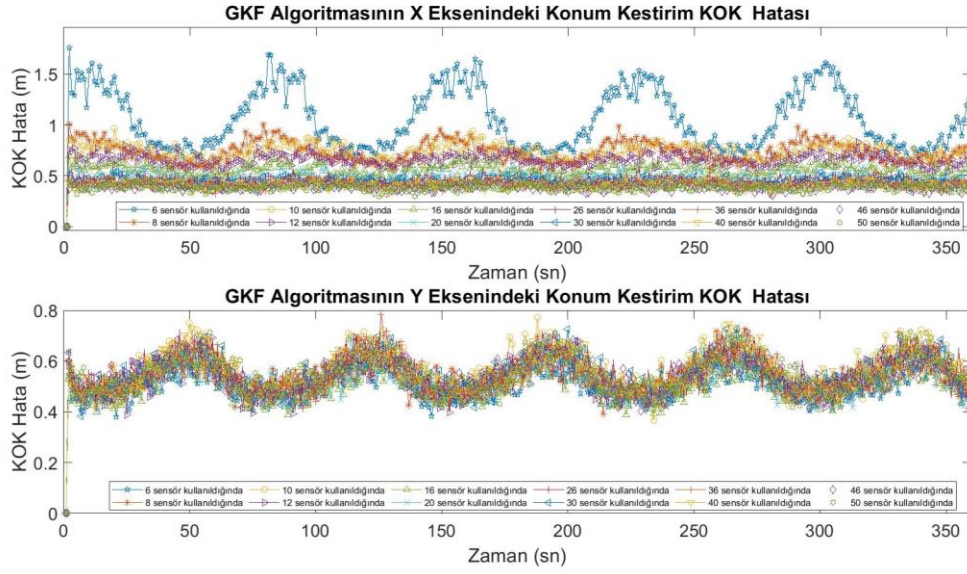
Sensör sayısı arttığında ve ölçüm gürültü varyansı büyük olan sensörden, küçük sensöre doğru ölçümler güncellendiğinde, sensör sayısı arttıkça, ölçüm gürültü varyansı daha küçük sensör ölçümleriyle güncelleme yapılmaktadır. Bu nedenle, x eksenindeki kestirim hatası azalmıştır. Şekil 4.44 incelendiğinde, sensör sayısı daha çok artırılıp, 46 ve 50 adet sensör kullanıldığında ise y eksenindeki kestirim hatasının daha çok azaldığı görülmektedir. Bunun sebebi, 5. sensör geometrisinde sensör sayısı 46, 50 sensöre kadar artırıldığında, 50., 49., 48. gibi yeni eklenen sensörler nesneye daha yakındır. Ayrıca bu sensörlerin ölçüm gürültüsü varyans değerleri de daha küçük olduğundan, GKF algoritması bu sensörlerin ölçümlerini daha çok kullanarak, y eksenindeki kestirim hatasını daha çok azaltmıştır.

Çizelge 4.22. GKF algoritmasının, 5. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları

Sensör Geometrisinde Kullanılan Sensör Sayısı	GKF Algoritmasının X Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Y Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Net Ortalama KOK Konum Kestirim Hatası (m)
6	2,6184	1,6261	3,0822
8	1,9782	1,5369	2,5050
10	1,6318	1,4904	2,2099
12	1,4054	1,4423	2,0137
16	1,1318	1,3658	1,7738
20	0,9583	1,3173	1,6289
26	0,7693	1,2474	1,4655
30	0,7060	1,1893	1,3830
36	0,6138	1,0818	1,2438
40	0,5589	1,0144	1,1581
46	0,4680	0,8069	0,9327
50	0,3861	0,4855	0,6203

GKF algoritmasında, sensör ölçümleri, sensörlerin ölçüm gürültü varyansları karışık sırayla güncellendiğinde, sensör sayısı arttırıldığında elde edilen, GKF algoritmasının konum kestirim hatası sonuçları

Beşinci sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyans değerleri karışık sırayla güncellendiğinde, sensör sayısı arttırıldığında elde edilen, GKF algoritmasının kestirim hatası grafiği, Şekil 4.45'te gösterilmiştir.



Şekil 4.45. Beşinci sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları karışık sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

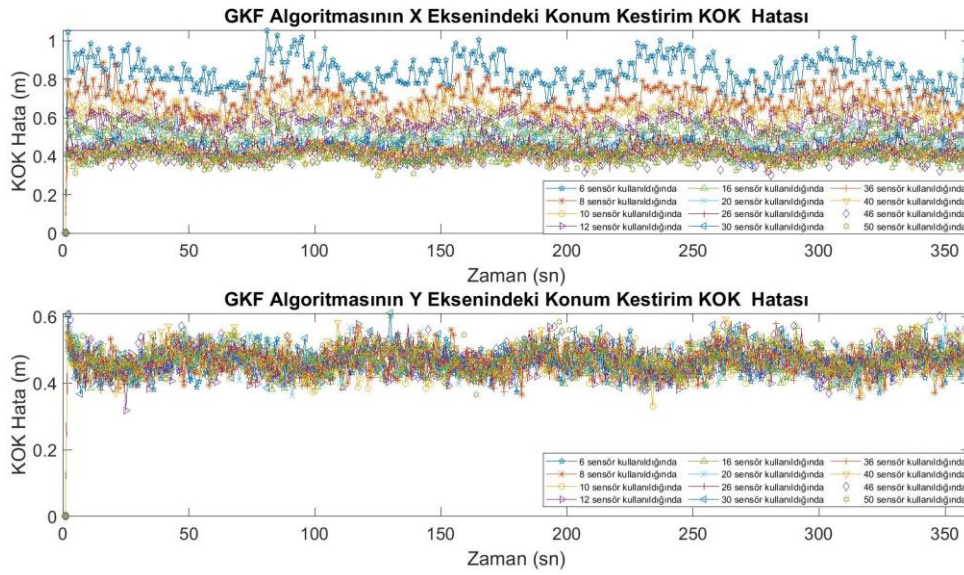
5. sensör geometrisinde, sensörler nesnenin altında ve üstünde olacak şekilde y eksenine dik şekilde yerleştirildiğinden, y eksenindeki bileşenleri aynıdır. Bu nedenle, her bir sensörden, GKF algoritmasına aynı bilgi geldiği için, y eksenindeki konum kestirim hatası yaklaşık aynıdır. Bundan dolayı, sensör sayısının artırılması, y eksenindeki konum kestirim hatasını etkilememiştir. X eksenindeki kestirim hatası ise, 6 sensör kullanıldığında yüksektir. Ancak sensör sayısı artırıldığında, sensörlerden, nesnenin x eksenindeki konumu hakkında daha fazla bilgi geldiği için kestirim hatası azalmıştır.

Çizelge 4.23. GKF algoritmasının, 5. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları karışık sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları

Sensör Geometrisinde Kullanılan Sensör Sayısı	GKF Algoritmasının X Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Y Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Net Ortalama KOK Konum Kestirim Hatası (m)
6	1.0563	0,5645	1,1700
8	0.7326	0,5530	0,9231
10	0,7305	0,5529	0,9161
12	0,6540	0,5509	0,8551
16	0,5517	0, 5495	0,7786
20	0,4886	0,5458	0,7325
26	0,4557	0,5447	0,7101
30	0,4372	0,5392	0,6941
36	0,4209	0,5337	0,6797
40	0,4123	0,5202	0,6637
46	0,3916	0,5061	0,6399
50	0,3842	0,5032	0,6331

GKF algoritmasında, sensör ölçümleri, ölçüm gürültü varyansları küçükten büyüğe doğru sırayla güncellendiğinde, sensör sayısı arttırıldığında elde edilen, GKF algoritmasının konum kestirim hatası sonuçları

Beşinci sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiği durumda ve sensör sayısı arttırıldığında elde edilen, GKF algoritmasının kestirim hatası grafiği, Şekil 4.46'da gösterilmiştir.



Şekil 4.46. Beşinci sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

5. sensör geometrisi incelendiğinde, sensör sayısı arttırıldığında, sensör geometrisine eklenen yeni sensörler, nesneye daha yakın yerleştirilmiştir. Örneğin, 30 adet sensör kullanılan geometride, 16. sensörden başlamak üzere, sensör geometrisine eklenen daha sonraki sensörler nesneye yakın olmaktadır. Ölçüm gürültü varyansı küçük olan sensörden, büyük olan sensöre doğru ölçümler güncellendiğinden, sensör sayısı artınca, sensör nesneye yaklaşırsa bile, ölçüm gürültü varyansı arttığı için, GKF algoritması bu sensörlerden alınan ölçümleri daha az kullanmaktadır. Bu nedenle, sensör sayısı artışı, algoritmanın kestirim hatasının azalmasına katkıda bulunmamaktadır. O yüzden sensör sayısı artsa da y eksenindeki konum kestirim hatası sabit kalmıştır. X eksenindeki konum kestirim hatası ise, sensör sayısı belli sayıya kadar artırıldığında, algoritma ölçüm gürültü varyansı düşük olan bu sensör ölçümlerini kullanarak kestirim hatasını azaltmıştır. Sensör sayısı artırılmaya devam ettiğinde ve ölçüm gürültü varyansı küçük olan sensörden, büyük olan sensöre doğru sırayla güncelleme yapıldığında, sensör geometrisine eklenen yeni sensörlerin ölçüm gürültü varyansları büyük olmaktadır. Bu nedenle, kestirim üretilirken bu sensör ölçümleri daha az kullanıldığından, daha fazla artırılan sensör sayısı, x eksenindeki konum kestirimine daha fazla katkıda bulunmamıştır. Bu sebeple kestirim hatası, sensör sayısı daha çok arttığında sabit kalmıştır.

Çizelge 4.24. GKF algoritmasının, 5. sensör geometrisinde sensör sayısı artırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları

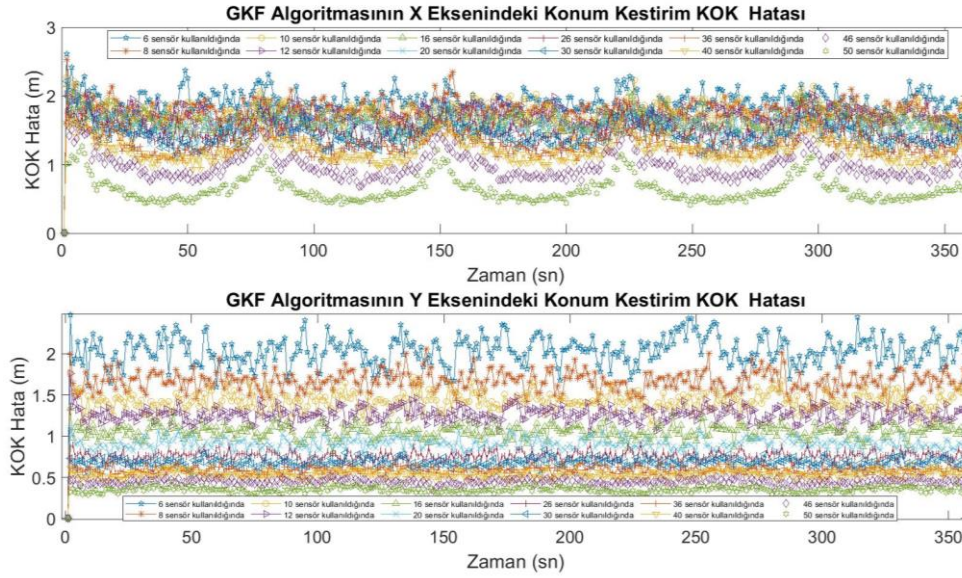
Sensör Geometrisinde Kullanılan Sensör Sayısı	GKF Algoritmasının X Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Y Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Net Ortalama KOK Konum Kestirim Hatası (m)
6	0,8362	0,4781	0,9632
8	0,6923	0,4766	0,8404
10	0,6181	0,4752	0,7796
12	0,5689	0,4745	0,7408
16	0,5138	0,4743	0,6992
20	0,4791	0,4741	0,6740
26	0,4479	0,4656	0,6460
30	0,4307	0,4612	0,6310
36	0,4192	0,4593	0,6218
40	0,4127	0,4573	0,6159
46	0,3996	0,4563	0,6065
50	0,3916	0,4546	0,6000

4.5.6. GKF algoritmasında altıncı sensör geometrisi kullanıldığında elde edilen sonuçlar

GKF algoritmasında, 6. sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültü varyansları büyükten küçüğe doğru, karışık ve küçükten büyüğe doğru sırayla güncellendiğinde elde edilen kestirim hatası sonuçları, 4.5.6 numaralı bölümün alt bölümlerinde sırayla incelenmiştir.

GKF algoritmasında, sensör ölçümleri, ölçüm gürültü varyansları büyükten küçüğe doğru sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının konum kestirim hatası sonuçları

GKF algoritmasında, altıncı sensör geometrisi kullanıldığında ve ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla ölçümler güncellendiği durumda, sensör sayısı artırıldığında elde edilen kestirim hata grafiği, Şekil 4.47’de gösterilmiştir.



Şekil 4.47. Altıncı sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

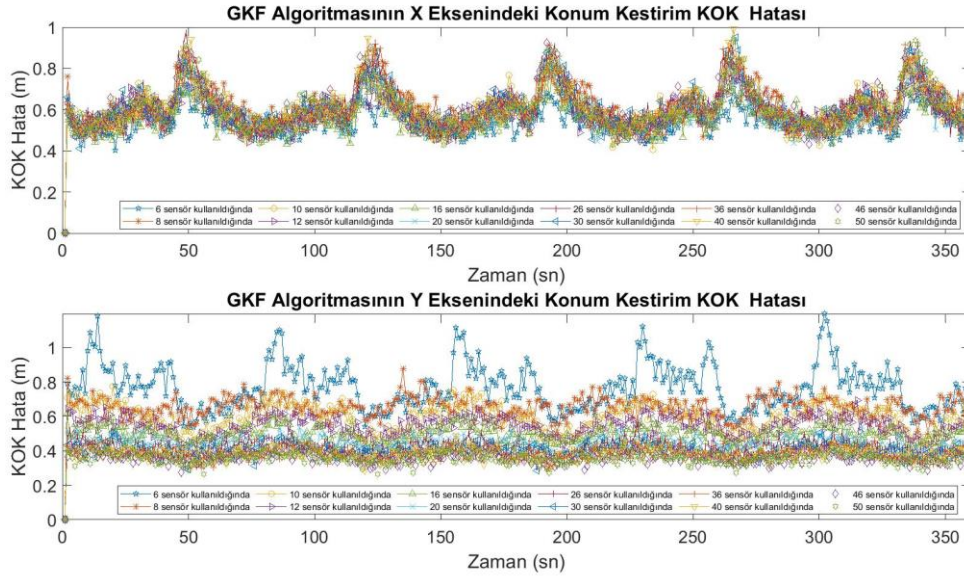
6. sensör geometrisinde sensörler x eksenine dik şekilde yerleştirildiğinden, her bir sensörden GKF algoritmasına, x eksenindeki konum kestirimi için yaklaşık aynı bilgi gelmektedir. Bu nedenle, sensör sayısı 40'a kadar arttırıldığında, x eksenindeki kestirim hatası sabit kalmıştır. Ancak sensör sayısı arttırılmaya devam edilip, 46, 50 adet sensör kullanıldığında, sensör geometrisine eklenen yeni sensörler nesneye daha yakın olmaktadır. Ayrıca, ölçümler, ölçüm gürültü varyansı büyükten küçüğe doğru güncellendiğinde, nesneye yakın olan 46., 47., 48., 49. ve 50. sensörlerin ölçüm gürültü varyansları küçüktür. Ölçüm gürültü varyansları küçük olduğundan ölçüm hataları da düşük olduğu için, GKF algoritması Kalman kazancını arttırmakta ve bu ölçümleri daha çok kullanarak daha küçük hatayla kestirim üretmektedir. Bu nedenle, 46 ve 50 adet sensör kullanıldığında, x eksenindeki kestirim hatası daha çok azalmıştır. Her bir sensörün y eksenindeki bileşenleri farklı olduğu için, sensör sayısı arttıkça, her bir sensörden, GKF algoritmasına ekstra bilgi gelmekte ve algoritma bu bilgileri daha çok kullanarak kestirim hatasını azaltmaktadır.

Çizelge 4.25. GKF algoritmasının, 6. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları büyükten küçüğe doğru sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları

Sensör Geometrisinde Kullanılan Sensör Sayısı	GKF Algoritmasının X Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Y Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Net Ortalama KOK Konum Kestirim Hatası (m)
6	1,9032	2,0317	2,7838
8	1,7751	1,6623	2,4319
10	1,7399	1,4176	2,2443
12	1,6856	1,2598	2,1043
16	1,6246	1,0516	1,9352
20	1,5648	0,9038	1,8070
26	1,4938	0,7601	1,6760
30	1,4227	0,6782	1,5760
36	1,3067	0,5886	1,4331
40	1,2222	0,5358	1,3344
46	0,9839	0,4414	1,0783
50	0,6416	0,3477	0,7297

GKF algoritmasında, sensör ölçümleri, sensörlerin ölçüm gürültü varyansları karışık sırayla güncellendiğinde, sensör sayısı arttırıldığında elde edilen, GKF algoritmasının konum kestirim hatası sonuçları

Altıncı sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyans değerleri karışık sırayla güncellendiğinde, sensör sayısı arttırıldığında elde edilen, GKF algoritmasının kestirim hatası grafiği, Şekil 4.48’de gösterilmiştir.



Şekil 4.48. Altıncı sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları karışık sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

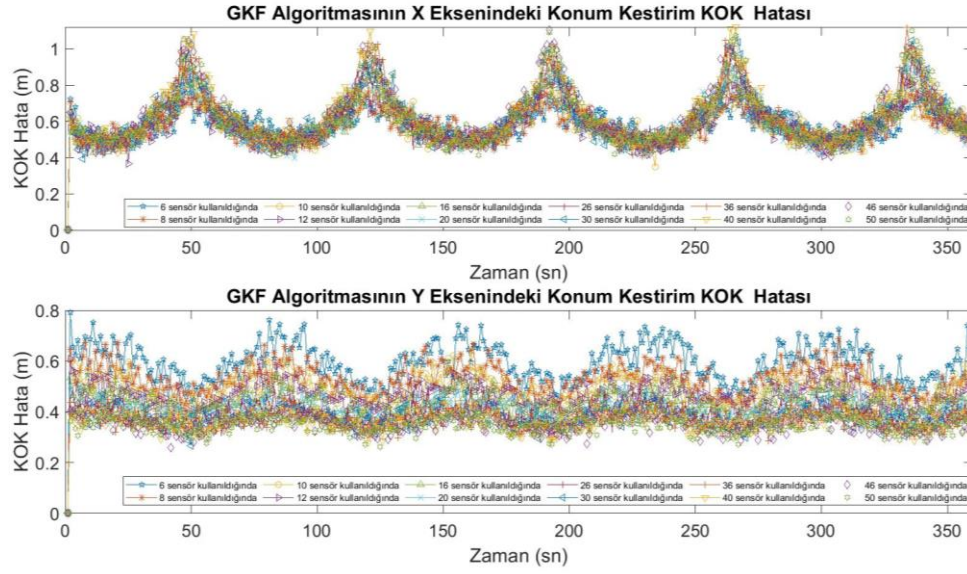
Sensör ölçümleri karışık sırayla güncellendiğinde, x eksenindeki kestirim hata grafiğinin daha fazla dalgalandığı görülmektedir. Bunun nedeni, ölçüm gürültü varyansı karışık sırayla güncellendiğinden, ölçüm gürültü varyansı yüksek olan ölçüm geldiğinde, kestirim hatası ölçüm hatasından dolayı yüksek, ölçüm gürültü varyansı küçük olan ölçüm geldiğinde ise kestirim hatası düşük olmaktadır. Ölçüm gürültü varyansı ise karışık sırayla güncellendiğinden dolayı kestirim hatası da daha çok artıp azaldığından, grafik daha dalgalı olmaktadır. 6. sensör geometrisinde sensörler x eksenine dik yerleştirildiğinden, her bir sensörün y eksenindeki bileşenleri farklıdır. Bu nedenle GKF algoritmasına, y eksenindeki konum kestirimi için, her bir sensörden farklı bilgi gelmektedir. Ayrıca, sensör sayısı arttıkça, nesnenin konumu hakkında daha fazla bilgi gelmekte ve GKF algoritması ise bu ölçümleri kullanarak, y eksenindeki konum kestirim hatasını azaltmaktadır.

Çizelge 4.26. GKF algoritmasının, 6. sensör geometrisinde sensör sayısı arttırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları karışık sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları

Sensör Geometrisinde Kullanılan Sensör Sayısı	GKF Algoritmasının X Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Y Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Net Ortalama KOK Konum Kestirim Hatası (m)
6	0,6168	0,7626	0,9808
8	0,6162	0,6531	0,8979
10	0,6162	0,5883	0,8519
12	0,6122	0,5403	0,8165
16	0,6077	0,4921	0,7819
20	0,6069	0,4422	0,7509
26	0,6044	0,4179	0,7348
30	0,5951	0,3988	0,7163
36	0,5911	0,3850	0,7054
40	0,5887	0,3795	0,7004
46	0,5598	0,3610	0,6661
50	0,5456	0,3549	0,6509

GKF algoritmasında, sensör ölçümleri, sensörlerin ölçüm gürültü varyansları küçükten büyüğe doğru sırayla güncellendiğinde, sensör sayısı arttırıldığında elde edilen, GKF algoritmasının konum kestirim hatası sonuçları

Altıncı sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiği durumda ve sensör sayısı arttırıldığında elde edilen, GKF algoritmasının kestirim hatası grafiği, Şekil 4.49’da gösterilmiştir.



Şekil 4.49. Altıncı sensör geometrisi kullanıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiğinde, sensör sayısı artırıldığında elde edilen, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hataları

Şekil 4.49 ve Şekil 4.47 incelendiğinde, sensör sayısı arttıkça ve sensör ölçümleri, ölçüm gürültü varyansları küçükten büyüğe doğru sırayla güncellendiğinde, y eksenindeki konum kestirim hatası, varyans büyükten küçüğe sırayla güncellendiğindeki hataya göre daha az azalmıştır. Bunun sebebi, sensör sayısı arttığında ve ölçüm gürültü varyansı küçükten büyüğe doğru sırayla ölçümler güncellendiğinde, eklenen yeni sensörlerin ölçüm gürültü varyanslarının yüksek olmasıdır. Ölçüm gürültü varyansı yüksek olduğundan, GKF algoritması bu ölçümleri daha az kullanmakta ve kestirime daha az etki ettirmektedir. Bu nedenle, sensör sayısı artsa da kestirim hatası, daha az azalmıştır.

Çizelge 4.27. GKF algoritmasının, 6. sensör geometrisinde sensör sayısı artırıldığında ve sensör ölçümleri, ölçüm gürültüsü varyansları küçükten büyüğe doğru sırayla güncellendiğinde elde edilen ortalama KOK konum kestirim hataları

Sensör Geometrisinde Kullanılan Sensör Sayısı	GKF Algoritmasının X Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Y Eksenindeki Ortalama KOK Konum Kestirim Hatası (m)	GKF Algoritmasının Net Ortalama KOK Konum Kestirim Hatası (m)
6	0,6351	0,5903	0,8671
8	0,6305	0,5270	0,8217
10	0,6299	0,4877	0,7966
12	0,6259	0,4616	0,7777
16	0,6214	0,4316	0,7565
20	0,6182	0,4095	0,7415
26	0,6043	0,3883	0,7183
30	0,5968	0,3766	0,7057
36	0,5873	0,3681	0,6931
40	0,5842	0,3637	0,6881
46	0,5819	0,3526	0,6803
50	0,5752	0,3480	0,6723

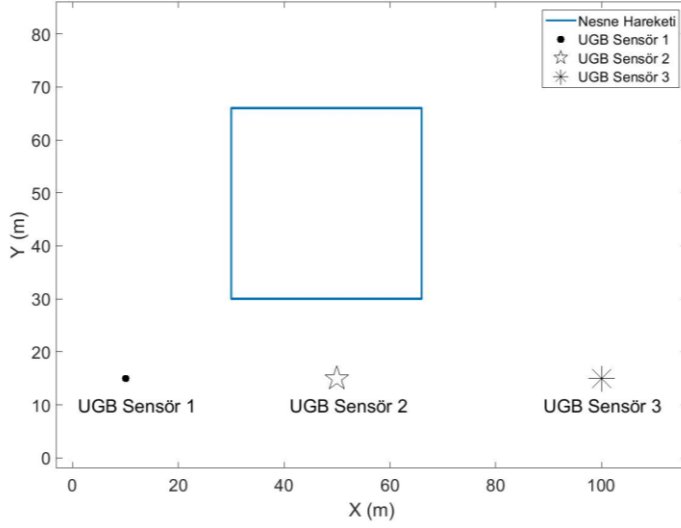
4.6. Nesne Hareketinin Genişletilmiş Kalman Filtre Algoritmasının Kestirim Hatası Üzerindeki Etkilerinin İncelenmesi

Nesne hareketinin, GKF algoritmasının kestirim hatası üzerindeki etkilerini incelemek için, dairesel hareket ve karesel hareket modelleri kullanılmıştır.

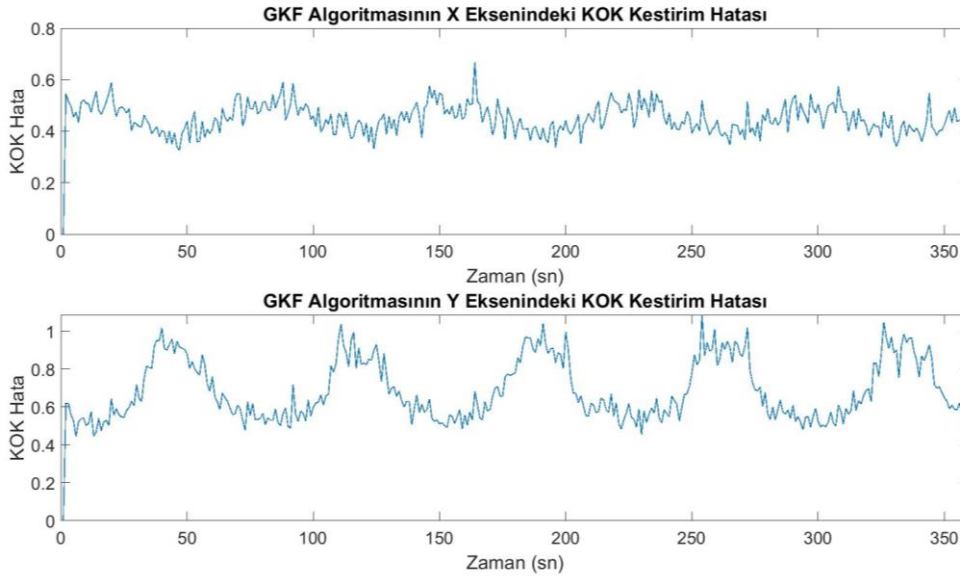
4.6.1. Nesne karesel hareket yaptığı ve 1. sensör geometrisi kullanıldığında, GKF algoritmasının konum kestirim hatası

Nesne hareketinin, GKF algoritmasının kestirim hatası üzerindeki etkilerinin incelenmesi için 1. sensör geometrisi kullanılmış ve nesne karesel şekilde hareket ettirilmiştir. Nesne hareketi ve sensör konumları Şekil 4.50’de gösterilmiştir. Nesne hem dairesel hem de karesel harekette, $1 \frac{m}{sn}$ sabit hızla, 360 saniye boyunca hareket etmektedir. Dairesel harekette, nesne $0,0873 \frac{rad}{sn}$ açısal hızla hareket etmektedir. 1. sensör (10, 15) metre, 2. sensör (50, 15) metre ve 3. sensör ise (100, 15) metre konumunda bulunmaktadır. Sensörlerin ölçüm gürültüsü

varyansları $0,5 \text{ m}^2$ olarak alınmıştır. Nesnenin karesel hareket modeli Şekil 4.50 ve GKF algoritmasının kestirim hatası grafiği Şekil 4.51’de gösterilmiştir.

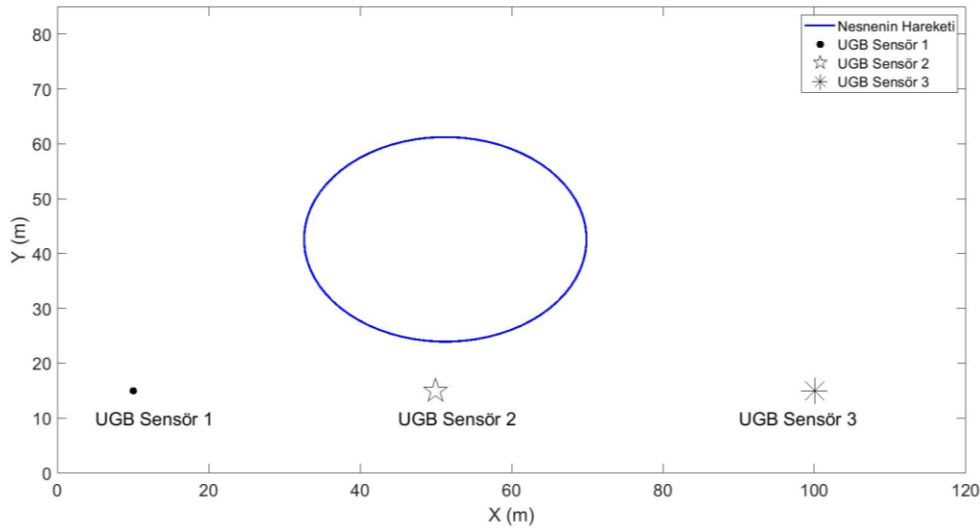


Şekil 4.50. Nesnenin karesel hareket modeli ve sensör konumları



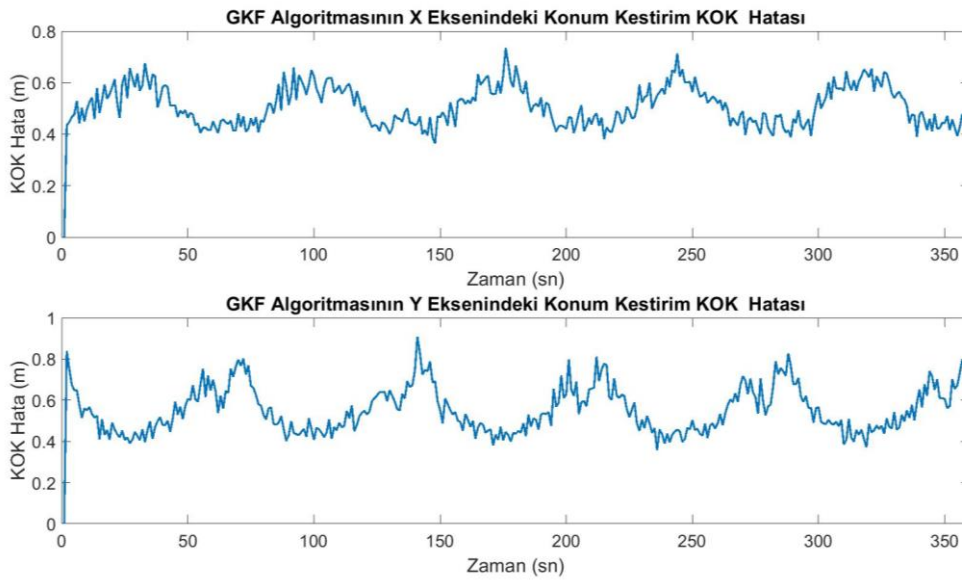
Şekil 4.51. Nesne karesel şekilde hareket ettiğinde ve 1. sensör geometrisi kullanıldığında GKF algoritmasının, nesnenin x ve y eksenlerindeki konum kestirim hataları

Nesne karesel şekilde hareket ettiğinde, GKF algoritmasının x ve y eksenlerindeki ortalama konum kestirim hataları sırasıyla $0,5471$ metre ve $0,5377$ metre olarak elde edilmiştir.



Şekil 4.52. Nesnenin dairesel hareket modeli ve sensör konumları

Şekil 4.52’de nesnenin dairesel hareket modeli ve sensör konumları gösterilmiştir. Nesne dairesel hareket yaptığında ve 1. sensör geometrisi kullanıldığında, GKF algoritmasının x ve y eksenlerindeki KOK konum kestirim hata grafikleri Şekil 4.53’te gösterilmiştir.

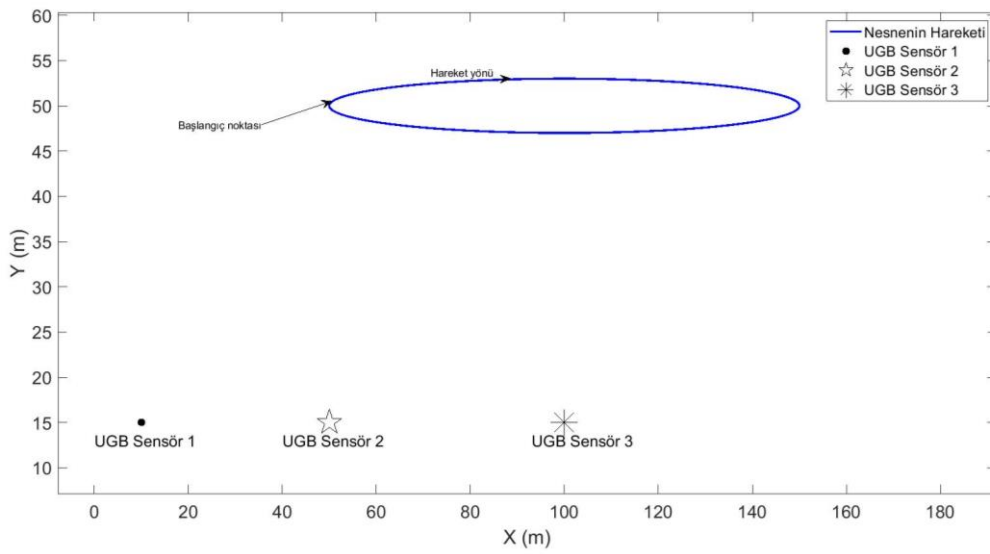


Şekil 4.53. Nesne dairesel şekilde hareket ettiğinde ve 1. sensör geometrisi kullanıldığında GKF algoritmasının, nesnenin x ve y eksenlerindeki konum kestirim hataları

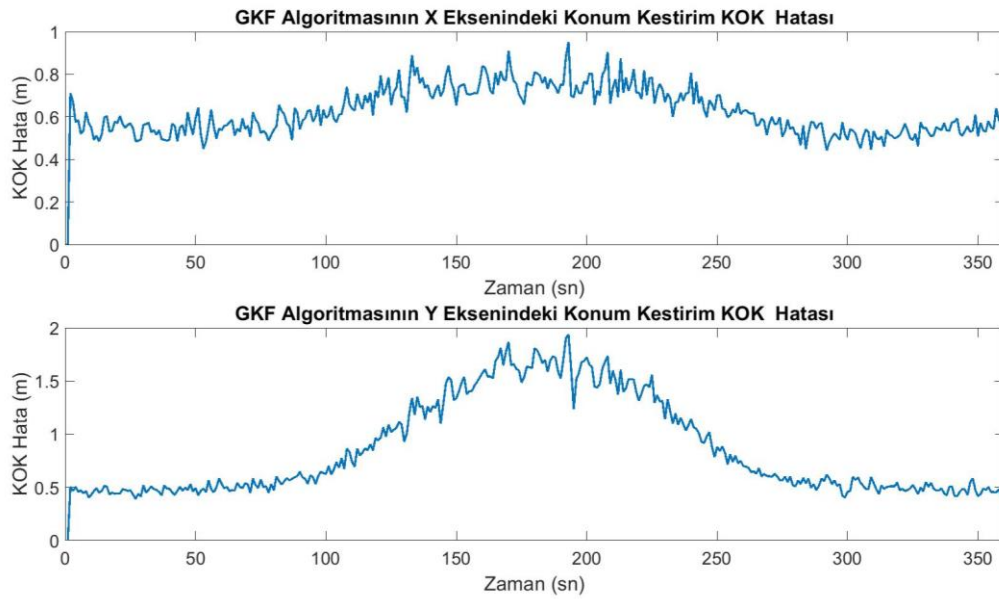
Nesne dairesel hareket ettiğinde, GKF algoritmasının x ve y eksenlerindeki ortalama KOK konum kestirim hataları sırasıyla 0,5087 metre ve 0,5524 metre olarak elde edilmiştir. Nesne dairesel ve karesel hareket yaptığında, hareket şekilleri benzer olduğundan, bu hareketler GKF algoritmasında benzer matematiksel modeller ile modellenmektedir. Ayrıca, her iki

harekette de nesne sensörlere aynı uzaklıkta bulunduğundan, GKF algoritmasının konum kestirim hatası yaklaşık aynı çıkmıştır.

Nesnenin sensörlere yaklaşıp uzaklaştığı bir senaryoda, GKF algoritmasının kestirim hatasının nasıl değiştiğini incelemek için, Şekil 4.54'te verilen hareket modeli kullanılmıştır. Nesne hareketine, 2 boyutlu kartezyen koordinat sistemi üzerinde (50, 50) metre noktasından başlayarak, Şekil 4.54'te gösterildiği yönde 360 saniye boyunca hareket etmektedir. Nesne, başlangıçta sensörlerden uzağa doğru hareket etmekte ve daha sonra tekrar sensörlere yaklaşarak hareketini tamamlamaktadır.



Şekil 4.54. Nesnenin hareket modeli ve sensör konumları



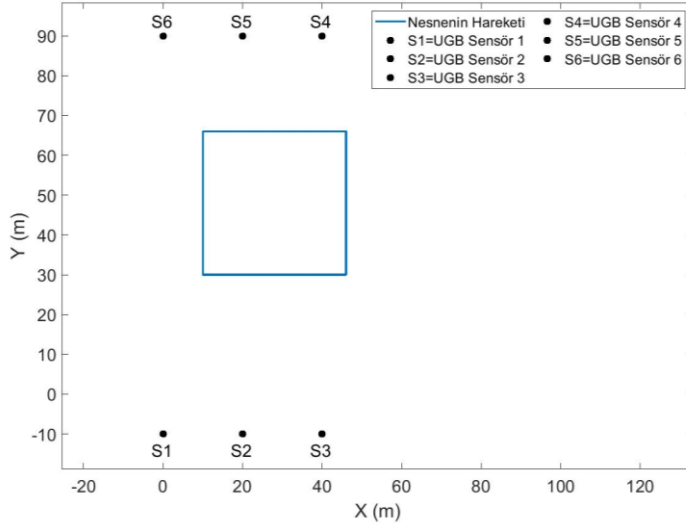
Şekil 4.55. Nesne dairesel şekilde hareket ettiğinde ve 1. sensör geometrisi kullanıldığında GKF algoritmasının, nesnenin x ve y eksenlerindeki konum kestirim hataları

Şekil 4.55'te, nesne Şekil 4.54'te gösterildiği şekilde hareket ettiğinde, GKF algoritmasının, nesnenin x ve y eksenlerindeki konum kestirim hataları gösterilmiştir. x ve y eksenlerindeki konum kestirim hata grafikleri incelendiğinde, hatanın başlangıçta arttığı ve daha sonra tekrar azaldığı görülmektedir. Nesne başlangıçta sensörlerden uzağa doğru hareket etmektedir. Bu durumda nesne sensörlerden uzaklaştığından, sensörlerden alınan bilgi miktarı azalmakta ve buna bağlı olarak GKF algoritmasının konum kestirim hatası artmaktadır. Yaklaşık olarak 180. saniyeden sonra, nesne sensörlere daha yakın olacak şekilde hareket ettiğinde ise, GKF algoritmasının x ve y eksenlerindeki konum kestirim hatası azalmaktadır. Bunun sebebi, nesne sensörlere daha yakın hareket ettiğinde, sensörlerden alınan bilgi miktarı artmakta ve GKF algoritması ise bu ölçümleri daha çok kullanarak kestirim hatasını azaltmaktadır. Bu nedenle, nesne sensörlerden uzağa doğru hareket ettiğinde, GKF algoritmasının konum kestirim hatası artmakta, sensörlere yakın hareket ettiğinde ise kestirim hatası azalmaktadır.

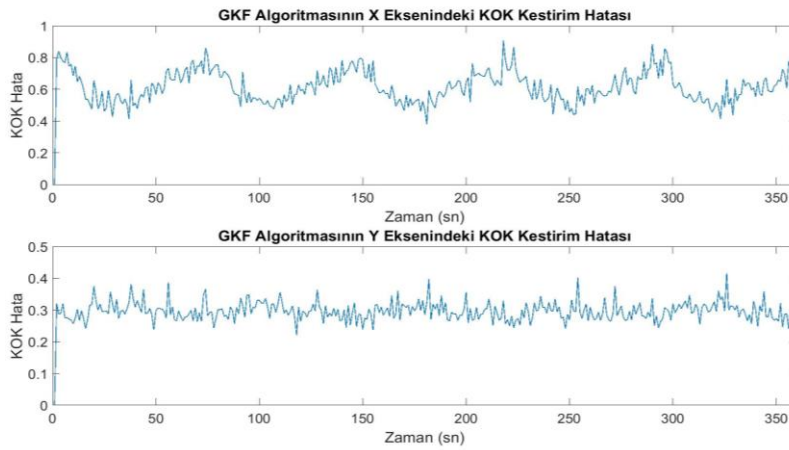
4.6.2. Nesne karesel hareket yaptığı ve 5. sensör geometrisi kullanıldığında, GKF algoritmasının konum kestirim hatası

Nesne hareketinin, GKF algoritmasının kestirim hatası üzerindeki etkilerinin incelenmesi için, 5. sensör geometrisi kullanılmış ve nesne karesel şekilde hareket ettirilmiştir. 5. sensör geometrisinde, 1. sensör (0, -10) metre, 2. sensör (20, -10) metre ve 3. sensör (40, -10) metre, 4. sensör (40, 90) metre, 5. sensör (20, 90) metre ve 6. sensör ise (0, 90) metre konumunda bulunmaktadır. Sensörlerin ölçüm gürültüsü varyansları $0,5 \text{ m}^2$ alınmıştır. Nesnenin karesel

hareket modeli Şekil 4.56 ve GKF algoritmasının kestirim hatası grafiği Şekil 4.57’de gösterilmiştir.

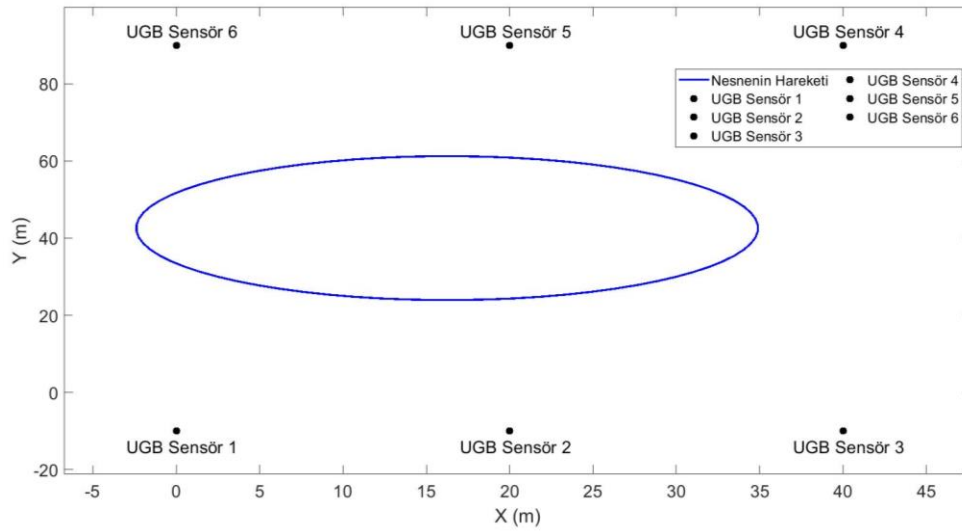


Şekil 4.56. Nesnenin karesel hareket modeli ve sensör konumları

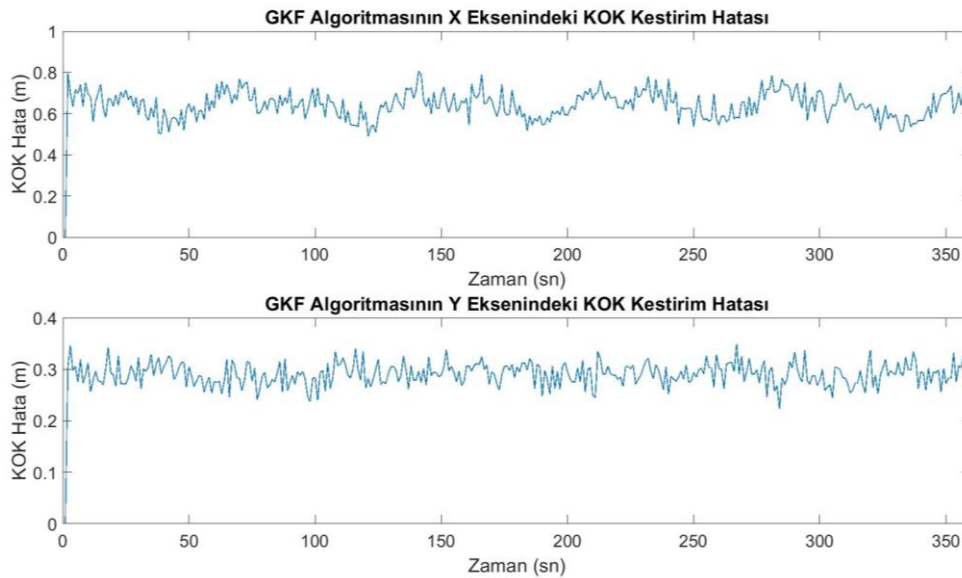


Şekil 4.57. Nesne karesel şekilde hareket ettiğinde ve 5. sensör geometrisi kullanıldığında GKF algoritmasının, nesnenin x ve y eksenlerindeki konum kestirim hataları

Nesne karesel şekilde hareket ettiğinde, GKF algoritmasının x ve y eksenlerindeki ortalama konum kestirim hataları sırasıyla 0,6152 metre ve 0,2957 metre olarak elde edilmiştir.



Şekil 4.58. Nesnenin dairesel hareket modeli ve sensör konumları



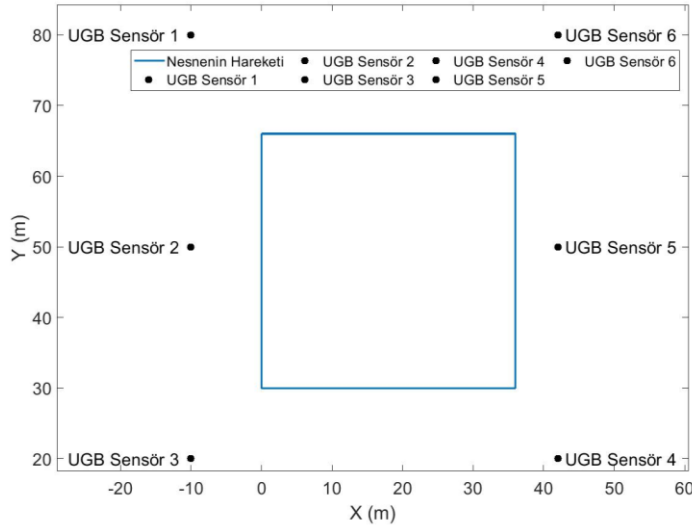
Şekil 4.59. Nesne dairesel hareket ettiğinde ve 5. sensör geometrisi kullanıldığında GKF algoritmasının, nesnenin x ve y eksenlerindeki konum kestirim hataları

Nesne dairesel hareket ettiğinde, GKF algoritmasının x ve y eksenlerindeki ortalama KOK konum kestirim hataları sırasıyla 0,6434 metre ve 0,2871 metre olarak elde edilmiştir. 5. sensör geometrisi kullanıldığında ve nesne dairesel ve karesel hareket yaptığında, hareket şekilleri benzer olduğu için ve her iki harekette de nesne sensörlere aynı uzaklıkta bulunduğu için, GKF algoritmasının konum kestirim hatası yaklaşık aynı çıkmıştır.

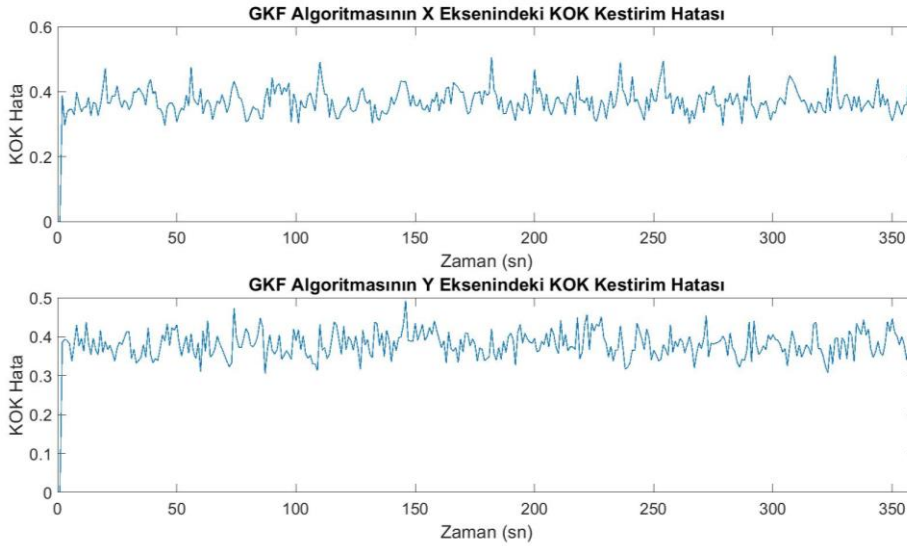
4.6.3. Nesne karesel hareket yaptığında ve 6. sensör geometrisi kullanıldığında, GKF algoritmasının konum kestirim hatası

Nesne hareketinin, GKF algoritmasının kestirim hatası üzerindeki etkilerinin incelenmesi için 6. sensör geometrisi kullanılmış ve nesne karesel şekilde hareket ettirilmiştir. 6. sensör

geometrisinde, 1. sensör (-10, 80) metre, 2. sensör (-10, 50) metre ve 3. sensör (-10, 20) metre, 4. sensör (42, 20) metre, 5. sensör (42, 50) metre ve 6. sensör ise (42, 80) metre konumunda bulunmaktadır. Sensörlerin ölçüm gürültüsü varyansları $0,5 \text{ m}^2$ alınmıştır. Nesnenin karesel hareket modeli Şekil 4.60 ve GKF algoritmasının kestirim hatası grafiği Şekil 4.61’de gösterilmiştir.



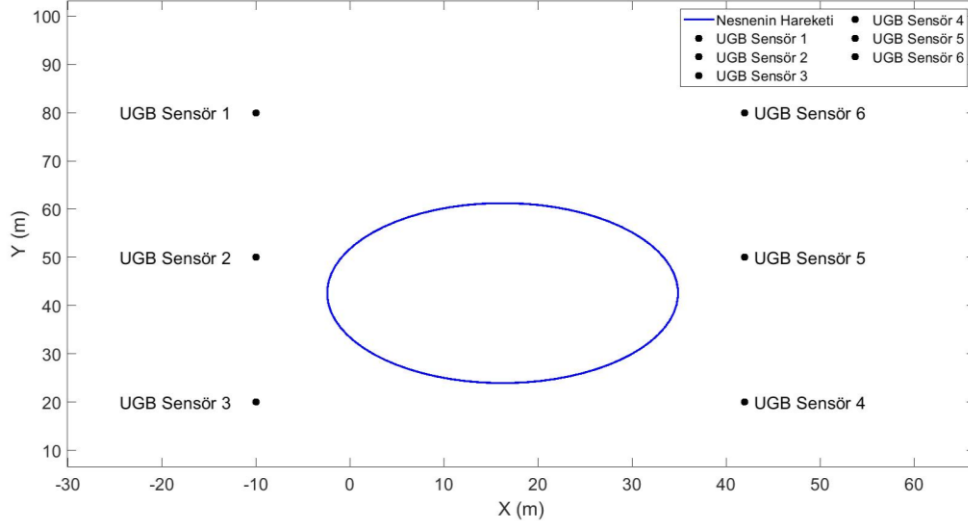
Şekil 4.60. Nesnenin karesel hareket modeli ve sensör konumları



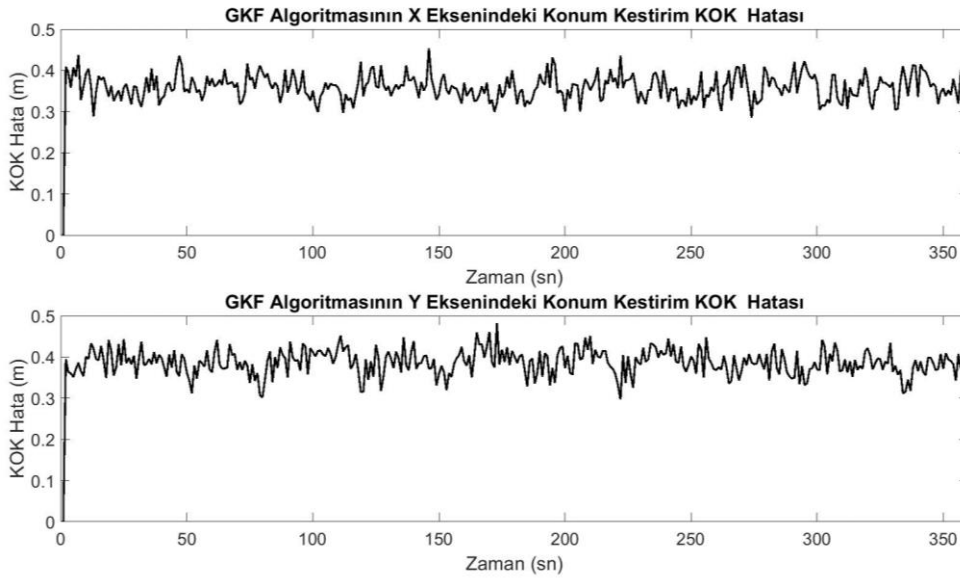
Şekil 4.61. Nesne karesel hareket ettiğinde ve 6. sensör geometrisi kullanıldığında GKF algoritmasının, x ve y eksenlerindeki konum kestirim hataları

Nesne karesel hareket ettiğinde, GKF algoritmasının x ve y eksenlerindeki ortalama KOK konum kestirim hataları sırasıyla 0,3697 metre ve 0,3799 metre olarak elde edilmiştir. Nesne dairesel hareket ettiğinde elde edilen kestirim hata grafikleri Şekil 4.61’de gösterilmiştir. Nesne dairesel hareket ettiğinde GKF algoritmasının x ve y eksenlerindeki ortalama konum

kestirim hataları sırasıyla 0,356 metre ve 0,3847 metre olarak elde edilmiştir. Karesel ve dairesel hareket şekilleri benzer olduğundan ve her iki harekette de nesne sensörlere aynı uzaklıkta bulunduğundan, GKF algoritmasının konum kestirim hatası yaklaşık aynı çıkmıştır.



Şekil 4.62. Nesnenin dairesel hareket modeli ve sensör konumları



Şekil 4.63. Nesne dairesel hareket ettiğinde ve 6. sensör geometrisi kullanıldığında GKF algoritmasının, x ve y eksenlerindeki konum kestirim hataları

5. SONUÇ VE ÖNERİLER

Tez çalışması kapsamında, ultra geniş bant sensörü kullanılarak, kapalı alanda nesne konumlandırma probleminde, GKF algoritmasının başarımlarının belirlenmesiyle, farklı sensör geometrilerinin ve sensör sayısının GKF algoritmasının kestirim performansı üzerindeki etkileri araştırılmıştır. Bu amaçla, farklı sensör geometrileri tasarlanıp GKF algoritmasında kullanılmış ve 100 Monte Carlo simülasyonu yapılarak KOK türünden konum kestirim hataları elde edilmiştir. Elde edilen kestirim hataları incelendiğinde, sensör geometrisi değiştiğinde, konum kestirim hatasının da değiştiği görülmektedir. Sensör geometrisinin, genişletilmiş Kalman filtre algoritmasının kestirim hatasını etkileme sebebi, sensörün bulunduğu yer değiştirildiğinde, jakobyian matris değişmektedir. Jakobyian matris değiştiği için ise Kalman kazancı değişmektedir. Kalman kazancı değiştiğinde, algoritmanın kestirim üretmek için sensör ölçümünden gelen bilgiye verdiği ağırlık değiştiği için algoritmanın kestirim değeri ve kestirim değerinin değişmesiyle beraber de kestirim hatası değişmektedir. Bu nedenle, sensör geometrisi GKF algoritmasının kestirim hatasını etkilemektedir. Elde edilen sonuçlar, GKF algoritması kullanarak, düşük hatalarla konum kestirimi elde edebilmek için sensör geometrisini optimum şekilde seçmek, tasarlamak gerektiğini göstermektedir.

Tez çalışması kapsamında, sensör sayısının GKF algoritmasının kestirim hatası üzerindeki etkileri, sensörlerin ölçüm gürültüsü varyans değerleri eşit olduğu durum için ve farklı olduğu durum için ayrı ayrı incelenmiştir. Bu amaçla, tez çalışması kapsamında tasarlanan farklı sensör geometrilerinin her birisinde, kullanılan sensör sayıları 50 adede kadar artırılarak, sensör sayılarının GKF algoritmasının kestirim hatası üzerindeki etkileri incelenmiştir. Sensör sayısı artışının, kestirim hatasını azaltması sensör geometrisine bağlı olduğu sonucuna ulaşılmıştır. Örneğin, 1., 2., 3. ve 4. sensör geometrileri kullanıldığında, sensör sayısı bulgular ve tartışma bölümünde anlatıldığı üzere, belli sayıya kadar arttırıldığında, kestirim hatası azalmaktadır. Ancak, sensör sayısı daha fazla arttırıldığında, kestirim hatası daha fazla azalmamaktadır. Bunun sebebi, 1., 2., 3. ve 4. sensör geometrilerinde, sensör sayısı arttıkça, sensörler nesneden uzaklaştığından, sensör sayısı artsa da sensörlerden bilgi alınamadığı için, bu sensörler, GKF algoritmasının kestirimine etki etmemektedir. Bu nedenle, sensör sayısı daha fazla artsa da kestirim hatası azalmamaktadır.

Tez çalışması kapsamında, nesne hareketinin, GKF algoritmasının kestirim hatası üzerindeki etkileri de incelenmiştir. Bu amaçla, nesne dairesel ve karesel hareket yaptığında, GKF

algoritmasının konum kestirim hataları elde edilmiştir. Karesel ve dairesel hareket şekilleri benzer olduğundan, her iki hareket modeli de matematiksel açıdan benzer şekilde modellenmektedir. Ayrıca, nesnenin karesel ve dairesel hareket yaptığı her iki durumda da nesne ile sensörler arası uzaklıklar aynıdır. Bu nedenle, nesnenin karesel ve dairesel hareket yaptığı her iki durum için de, GKF algoritmasının konum kestirim hataları yaklaşık aynı çıkmıştır.

Sensör geometrisi GKF algoritmasının kestirim hatasını etkilediği için, yüksek doğruluklarla konum kestirimi elde edebilmek amacıyla, optimum sensör geometrisinin elde edilmesine yönelik yapılabilecek çalışmalar, gelecekte yapılabilecek çalışmalar olarak önerilmektedir. Ayrıca, sensör sayısının belirli bir miktar artışı, GKF algoritmasının kestirim hatasını azaltmaktadır. Ancak öte yandan, sensör sayısının artışı aynı zamanda, kapalı alanda nesne konumlandırma probleminin çözümünde maliyeti de artırmaktadır. Bu nedenle, hem sensör sayısının artışına bağlı olarak GKF algoritmasının konum kestirim hatasındaki azalmalar, hem de sensör sayısının artışının getirdiği maliyet göz önüne alınarak, optimum sensör sayısının belirlenmesi de oldukça önemlidir. Bu nedenle, gelecekteki çalışmalar için, GKF algoritması kullanılarak, kapalı alanda nesne konumlandırma problemlerinin çözümünde, optimum sensör geometrisi ve optimum sensör sayısının belirlenmesi üzerine yapılabilecek çalışmalar önerilmektedir.

KAYNAKLAR

- Alarifi, A., Al-Salman, A., Alsaleh, M., Alnafessah, A., Al-Hadhrami, S., Al-Ammar, M. A., & Al-Khalifa, H. S. (2016). Ultra Wideband Indoor Positioning Technologies: Analysis and Recent Advances. *Sensors*, 16(5), Art. 5.
- Asmaa, L., Hatim, K. A., & Abdelaaziz, M. (2014). Localization algorithms research in wireless sensor network based on Multilateration and Trilateration techniques. *2014 Third IEEE International Colloquium in Information Science and Technology (CIST)*, 415-419.
- Bar-Shalom, Y., Li, X. R., Kirubarajan, T. (2001). *Estimation with Application to Tracking and Navigation*. America: Wiley – Interscience Publication, 386.
- Baran, T. (2018). Türkiye’de Savunma Sanayi Sektörünün İncelenmesi ve Savunma HAarcamalarının Ekonomi Üzerindeki Etkilerinin Değerlendirilmesi. *Uluslararası İktisadi ve İdari Bilimler Dergisi*, 4(2), Art. 2.
- Belhadi, Z., & Fergani, L. (2014). Fingerprinting methods for RFID tag indoor localization. *2014 International Conference on Multimedia Computing and Systems (ICMCS)*, 717-722.
- Boonsriwai, S., & Apavatjirut, A. (2013). Indoor WIFI localization on mobile devices. *2013 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, 1-5.
- Botler, L., Spörk, M., Diwold, K., & Römer, K. (2020). Direction Finding with UWB and BLE: A Comparative Study. *2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 44-52.
- Bouet, M., & Dos Santos, A. L. (2008). RFID tags: Positioning principles and localization techniques. *2008 1st IFIP Wireless Days*, 1-5.
- Brida, P., & Machaj, J. (2013). A Novel Enhanced Positioning Trilateration Algorithm Implemented for Medical Implant In-Body Localization. *International Journal of Antennas and Propagation*, 2013.
- Chan, M., & Zhang, X. (2017). Reader leveled RFID localization approaches and their simulation. *2017 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, 1-6.
- Choliz, J., Eguizabal, M., Hernandez-Solana, A., & Valdovinos, A. (2011). Comparison of Algorithms for UWB Indoor Location and Tracking Systems. *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*, 1-5.
- Costin, A. M., & Teizer, J. (2015). Fusing passive RFID and BIM for increased accuracy in indoor localization. *Visualization in Engineering*, 3(1), 1-20.
- De Oliveira, L. S., Rayel, O. K., & Leitao, P. (2021). Low-Cost Indoor Localization System Combining Multilateration and Kalman Filter. *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*, 1-6.
- Deak, G., Curran, K., & Condell, J. (2012). A survey of active and passive indoor localisation systems. *Computer Communications*, 35(16), 1939-1954.
- Duong, N.-S., & Thi, T.-M. D. (2021). Smartphone indoor positioning based on enhanced BLE beacon multi-lateration. *TELKOMNIKA Telecommunication, Computing, Electronics and Control*, 19(1), 51-62.
- Do, V. A., Rana, B., & Hong, I.-P. (2021). Identification of Wi-Fi and Bluetooth Signals at the Same Frequency using Software Defined Radio. *Journal of IKEEE*, 25(2), 252-260.
- Gharat, V., Colin, E., Baudoin, G., & Richard, D. (2017). Indoor performance analysis of LF-RFID based positioning system: Comparison with UHF-RFID and UWB. *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 1-8.

- Goldoni, E., Savioli, A., Risi, M., & Gamba, P. (2010). Experimental analysis of RSSI-based indoor localization with IEEE 802.15.4. *2010 European Wireless Conference (EW)*, 71-77.
- Handojo, A., Lim, R., Octavia, T., & Anggita, J. K. (2019). Museum Visitor Activity Tracker using Indoor Positioning System. *TIMES-iCON 2019 4th Technology Innovation Management and Engineering Science International Conference*, 1-5.
- Hayward, S. J., van Lopik, K., Hinde, C., & West, A. A. (2022). A Survey of Indoor Location Technologies, Techniques and Applications in Industry. *Internet of Things*, 20, 100608.
- Hazas, M., & Hopper, A. (2006). Broadband ultrasonic location systems for improved indoor positioning. *IEEE Transactions on Mobile Computing*, 5(5), 536-547.
- Huang, B., & Wang, Q.-G. (2006). Overview of Emerging Bayesian Approach to Nonlinear System Identification
- Huang, C.-H., Lee, L.-H., Ho, C. C., Wu, L.-L., & Lai, Z.-H. (2015). Real-Time RFID Indoor Positioning System Based on Kalman-Filter Drift Removal and Heron-Bilateration Location Estimation. *IEEE Transactions on Instrumentation and Measurement*, 64(3), 728-739.
- İlçi, V., Güllal, V., Alkan, R., & Çizmecici, H. (2015). *Trilateration Technique for WiFi-Based Indoor Localization*.
- Kadaba, A. S., Bhat, A. R., Maheshwari, V. M., Surekha, T. P., & Girijamba, L. D. (2018). Indoor Positioning System using Ultrasound. *International Journal of Engineering Research & Technology*, 6(13).
- Kim, B., Bong, W., & Kim, Y. C. (2011). Indoor localization for Wi-Fi devices by cross-monitoring AP and weighted triangulation. *2011 IEEE Consumer Communications and Networking Conference (CCNC)*, 933-936.
- Kirbaş, İ., & Arslan, K. (2020). KAPALI ALAN KONUMLANDIRMA SİSTEMLERİ İÇİN DÜĞÜM PROTOTİPİ GELİŞTİRME. *Mühendislik Bilimleri ve Tasarım Dergisi*, 8(2), Art. 2.
- Kong, X., Wu, C., You, Y., & Yuan, Y. (2023). Hybrid Indoor Positioning Method of BLE and PDR Based on Adaptive Feedback EKF With Low BLE Deployment Density. *IEEE Transactions on Instrumentation and Measurement*, 72, 1-12.
- Korogodin, I. V., Dneprov, V. V., & Mikhaylova, O. K. (2019). Triangulation Positioning by Means of Wi-Fi Signals in Indoor Conditions. *2019 PhotonIcs & Electromagnetics Research Symposium - Spring (PIERS-Spring)*, 2339-2345.
- Kumar, S., & Moore, K. B. (2002). The Evolution of Global Positioning System (GPS) Technology. *Journal of Science Education and Technology*, 11(1), 59-80.
- Lanir, J., Kuflik, T., Sheidin, J., Yavin, N., Leiderman, K., & Segal, M. (2017). Visualizing museum visitors' behavior: Where do they go and what do they do there? *Personal and Ubiquitous Computing*, 21(2), 313-326.
- Lau, S.-Y., Lin, T.-H., Huang, T.-Y., Ng, I.-H., & Huang, P. (2009). A measurement study of zigbee-based indoor localization systems under RF interference. *Proceedings of the 4th ACM International Workshop on Experimental Evaluation and Characterization*, 35-42.
- Li, N., & Becerik-Gerber, B. (2011). Performance-based evaluation of RFID-based indoor location sensing solutions for the built environment. *Advanced Engineering Informatics*, 25(3), 535-546.
- Lima, D. (2017). Neural Network Training Using Unscented and Extended Kalman Filter. *Robotics&Automation Engineering Journal*, 2.

- Liu, H., Darabi, H., Banerjee, P., & Liu, J. (2007). Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6), 1067-1080.
- Montaser, A., & Moselhi, O. (2014). RFID indoor location identification for construction projects. *Automation in Construction*, 39, 167-179.
- Muqaibel, A., Safaai-Jazi, A., & Riad, S. (2004). Ultra Wideband vs. Narrowband Communications.
- Munoz, D., Bouchereau, F., Vargas, C., & Enriquez, R. (2009). CHAPTER 3—Location Information Processing. İçinde D. Munoz, F. Bouchereau, C. Vargas, & R. Enriquez (Ed.), *Position Location Techniques and Applications* (ss. 67-102). Academic Press.
- Papapostolou, A., & Chaouchi, H. (2011). RFID-assisted indoor localization and the impact of interference on its performance. *Journal of Network and Computer Applications*, 34(3), 902-913.
- Pu, Y.-C., & You, P.-C. (2018). Indoor positioning system based on BLE location fingerprinting with classification approach. *Applied Mathematical Modelling*, 62, 654-663.
- Retscher, G., Gikas, V., Hofer, H., Perakis, H., & Kealy, A. (2019). Range validation of UWB and Wi-Fi for integrated indoor positioning. *Applied Geomatics*, 11(2), 187-195.
- Rong Li, X., & Jilkov, V. P. (2003). Survey of maneuvering target tracking. Part I. Dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4), 1333-1364.
- Rukaiya, Qureshi, R., & Enam, R. (2015). RSSI based Node Localization using Trilateration in Wireless Sensor Network. *Bahria University Journal of Information & Communication Technologies*, 8, 58-64.
- Sánchez, A., Elvira, S., De Castro, A., Gonzalez de Rivera, G., Ribalda, R., & Garrido, J. (2009). *Low cost indoor ultrasonic positioning implemented in FPGA*. 2709-2714.
- Sang, C. L., Adams, M., Korthals, T., Hörmann, T., Hesse, M., & Rückert, U. (2019). A Bidirectional Object Tracking and Navigation System using a True-Range Multilateration Method. *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 1-8.
- Sanpechuda, T., & Kovavisaruch, L. (2008). A review of RFID localization: Applications and techniques. *2008 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, 2, 769-772.
- Schejbal, V., Cermak, D., Nemec, Z., Pidanic, J., Konecny, J., Bezousek, P., & Fiser, O. (2006). Multipath Propagation of UWB Through-Wall Radar and EMC Phenomena. *Radioengineering*, 15.
- Shorey, R., & Miller, B. A. (2000). The Bluetooth technology: Merits and limitations. *2000 IEEE International Conference on Personal Wireless Communications. Conference Proceedings (Cat. No.00TH8488)*, 80-84.
- Subhan, F., Hasbullah, H., Rozyyev, A., & Bakhsh, S. T. (2011). Indoor positioning in Bluetooth networks using fingerprinting and lateration approach. *2011 International Conference on Information Science and Applications*, 1-9.
- Tan, P., Tsinakwadi, T., Xu, Z., & Xu, H. (2022). Sing-Ant: RFID Indoor Positioning System Using Single Antenna with Multiple Beams Based on LANDMARC Algorithm. *Applied Sciences*, 12, 6751.
- Thaljaoui, A., Val, T., Nasri, N., & Brulin, D. (2015). BLE localization using RSSI measurements and iRingLA. *2015 IEEE International Conference on Industrial Technology (ICIT)*, 2178-2183.

- Yang, J., Wang, Z., & Zhang, X. (2015). An iBeacon-based indoor positioning systems for hospitals. *International Journal of Smart Home*, 9(7), 161-168.
- Yim, J., Park, C., Joo, J., & Jeong, S. (2008). Extended Kalman Filter for wireless LAN based indoor positioning. *Decision Support Systems*, 45(4), 960-971.
- Wessels, A., Wang, X., Laur, R., & Lang, W. (2010). Dynamic indoor localization using multilateration with RSSI in wireless sensor networks for transport logistics. *Procedia Engineering*, 5, 220-223.
- Zafari, F., Papapanagiotou, I., Devetsikiotis, M., & Hacker, T. (2017). *An iBeacon based Proximity and Indoor Localization System* (arXiv:1703.07876). <http://arxiv.org/abs/1703.07876>
- Zegeye, W. K., Amsalu, S. B., Astatke, Y., & Moazzami, F. (2016). WiFi RSS fingerprinting indoor localization for mobile devices. *2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 1-6.
- Zhang, L., Zhang, H., Cui, X. R., & Gulliver, T. A. (2012). Ultra Wideband Indoor Positioning Using Kalman Filters. *Advanced Materials Research*, 433-440, 4207-4213.
- Zhang, W., Zhu, X., Zhao, Z., Liu, Y., & Yang, S. (2020). High Accuracy Positioning System Based on Multistation UWB Time-of-Flight Measurements. *2020 IEEE International Conference on Computational Electromagnetics (ICCEM)*, 268-270.
- Zhou, N., Lau, L., Bai, R., & Moore, T. (2021). Novel prior position determination approaches in particle filter for ultra wideband (UWB)-based indoor positioning. *NAVIGATION: Journal of the Institute of Navigation*, 68(2), 277-292.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : KA***** G****
Uyruğu : T.C.
Doğum tarihi ve yeri :
Medeni hali :
Telefon :
Faks :
e-mail :



Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Yüksek lisans	SBTÜ/ Savunma Teknolojileri	2023
Lisans	CÜ / Elektrik Elektronik Mühendisliği	2015
Lise	Halil Rıfat Paşa Lisesi	2011

İş Deneyimi

Yıl	Yer	Görev
2020-Halen	Sivas Bilim ve Teknoloji Üniversitesi	Araştırma Görevlisi
2015-2016	Necron Energy	Teknik Servis Mühendisi

Yabancı Dil

İngilizce

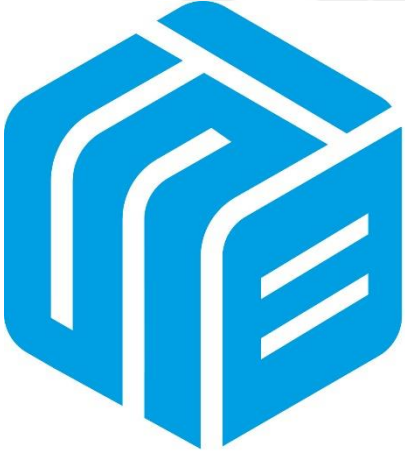
Yayınlar

-

Hobiler

Yüzme, Yürüyüş Yapmak

KÖKLERDEN GÖKLERE...



**SİVAS
BİLİM VE TEKNOLOJİ
ÜNİVERSİTESİ**