

**T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**HAREKETLİ ARAÇLARDA KALMAN FİLTRESİ
YARDIMIYLA UYDU TAKİBİ GERÇEKLEMESİ**

**YÜKSEK LİSANS TEZİ
Fatih KOCADAĞ**

**Enstitü Anabilim Dalı : ELEKTRİK-ELEKTRONİK
MÜHENDİSLİĞİ**
Enstitü Bilim Dalı : ELEKTRONİK
Tez Danışmanı : Doç. Dr. Aşkın DEMİRKOL

Haziran 2015

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

HAREKETLİ ARAÇLARDA KALMAN FİLTRESİ
YARDIMIYLA UYDU TAKİBİ GERÇEKLEMESİ

YÜKSEK LİSANS TEZİ

Fatih KOCADAĞ

Enstitü Anabilim Dalı : ELEKTRİK-ELEKTRONİK
MÜHENDİSLİĞİ
Enstitü Bilim Dalı : ELEKTRONİK

Bu tez 04 / 06 /2015 tarihinde aşağıdaki jüri tarafından oybirliği ile kabul edilmiştir.

Doç.Dr.
Aşkın DEMİRKOL
Jüri Başkanı



Prof.Dr.
Cemil ÖZ
Üye



Doç.Dr.
Ali TANGEL
Üye



BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Fatih KOCADAĞ

17.04.2015

ÖNSÖZ

Danışmanım Doç. Dr. Aşkın DEMİRKOL'a tez boyunca ve yüksek lisans öğrenciliğim boyunca yaptığı katkılardan ve daha iyi işler yapmak için cesaretlendirdiğinden dolayı çok teşekkür ederim.

Çalışma ve hayatım boyunca Beni destekleyen, cesaretlendiren annem Safinaz KOCADAĞ'a babam Halil KOCADAĞ'a, ablam Yıldız KOCADAĞ'a, ablam Gülten KOCADAĞ'a, ağabeyim Yetiş KOCADAĞ'a çok teşekkür ederim.

Ayrıca tezin uygulama test çalışmaları aşamasında yardımcı olan ve imkân sağlayan ablam Yıldız KOCADAĞ'a ve ağabeyim Yetiş KOCADAĞ'a çok teşekkür ederim.

Ayrıca Abant İzzet Baysal Üniversitesi Bilgi İşlem Daire Başkanlığı'ndaki bütün çalışma arkadaşlarıma çok teşekkür ederim.

İÇİNDEKİLER

ÖNSÖZ.....	i
İÇİNDEKİLER.....	ii
SİMGELER VE KISALTMALAR LİSTESİ.....	v
ŞEKİLLER LİSTESİ.....	vi
TABLolar LİSTESİ.....	viii
ÖZET.....	ix
SUMMARY.....	x

BÖLÜM 1.

UYDU SİSTEMLERİ VE PARAMETRELERİ.....	1
1.1. Giriş.....	1
1.2. Uydu Sistemleri.....	2
1.3. Kullanım Amaçlarına Göre Uydular.....	5
1.3.1. Meteoroloji uyduları.....	5
1.3.2. Radyo ve tv yayın uyduları.....	6
1.3.3. Askeri uydular.....	6
1.3.4. Seyrüsefer uyduları.....	6
1.3.5. Küresel iletişim uyduları.....	6
1.4. Yörüngelerine Göre Uydular.....	6
1.4.1. Jeosenkron yörünge uyduları.....	7
1.4.2. Alçak yörünge uyduları.....	8
1.4.3. Orta yörünge uyduları.....	9
1.4.4. Yüksek yörünge uyduları.....	9
1.4.5. Diğer yörünge uydular.....	10
1.5. Uydu İletişiminde Kullanılan Frekans Bantları.....	10
1.6. Uydu Antenleri.....	12
1.6.1. Huni antenler.....	12

1.6.2.Faz dizi antenler	13
1.6.3.Parabolik antenler	14
1.7. Hareketli Araç Konum Bilgisi	15
1.8. Hareketli Araç Pozisyon Bilgisi	16
BÖLÜM 2.	
ALGILAYICI SİNYAL İYİLEŞTİRME	17
2.1. Giriş	17
2.2. Doğruluk ve Hassasiyet	17
2.3. Hata Türleri.....	17
2.4. Sinyal İyileştirme Filtreleri.....	18
2.4.1. Alçak geçiren filtre.....	19
2.4.2. Yüksek geçiren filtre	20
2.4.3. Basit hareketli ortalama.....	20
2.4.4. Bant geçiren filtre.....	22
2.4.5. Bant durduran filtre	23
BÖLÜM 3.	
KALMAN FİLTRESİ.....	24
3.1. Durum Uzayı Modeli.....	25
3.2. Gözlemleyici Tasarım Problemi	26
3.3. Kalman Filtresi	27
3.4. Ayrık Kalman Filtresi	27
3.5. Ayrık Kalman Filtresi Algoritması	30
BÖLÜM 4.	
UYGULAMA.....	32
4.1. Giriş	32
4.2. Hareketli Aracın Konum ve Pozisyon Bilgisi.....	33
4.3. Algoritma.....	36

BÖLÜM 5.	
SONUÇLAR VE ÖNERİLER.....	40
KAYNAKLAR.....	41
EKLER.....	45
ÖZGEÇMİŞ.....	61

SİMGELER VE KISALTMALAR LİSTESİ

dB	: Desibel
GEO	: Geosynchronous Equatorial Orbit
GHz	: Giga Hertz
GPS	: Global Positioning System
HEO	: High Earth Orbit
HTO	: Hohman Transfer Orbit
km	: Kilometre
LEO	: Low Earth Orbit
MEO	: Medium Earth Orbit
MHz	: Mega Hertz
ms	: Milisaniye
NASA	: National Aeronautics and Space Administration
TV	: Televizyon
UHF	: Ultra High Frequency

ŞEKİLLER LİSTESİ

Şekil 1.1. Sputnik 1 uydusunun resmi.....	1
Şekil 1.2. Uydu haberleşme birimleri	4
Şekil 1.3. 1994-2020 Yılları uydu planlaması	5
Şekil 1.4. Jeosenkron uydu.....	8
Şekil 1.5. LEO,MEO ve GEO yörüngeleri	9
Şekil 1.6. Uydu yörüngeleri ve yörünge mesafeleri.....	10
Şekil 1.7. Uydu frekans bantları.....	11
Şekil 1.8. Huni anten çeşitleri	12
Şekil 1.9. Doğrusal faz dizi anten	13
Şekil 1.10. Düzlemsel faz dizi anten	14
Şekil 1.11. Parabolik çanak anten yapısı.....	14
Şekil 1.12. Parabolik çanak anten	15
Şekil 1.13. GPS uyduları	16
Şekil 1.14. Araç pozisyon bilgisi	16
Şekil 2.1. Doğruluk hassasiyet karşılaştırması.....	18
Şekil 2.2. Alçak geçiren filtre devre şeması.....	19
Şekil 2.3. Yüksek geçiren filtre şeması	20
Şekil 2.4. Bant geçiren filtre	23
Şekil 2.5. Band durduran filtre	23
Şekil 3.1. Kalman Filtresi blok diyagramı	24
Şekil 3.2. Doğrusal durum uzay modeli.....	26
Şekil 3.3. Ayrık Kalman Filtresi döngüsü.....	30
Şekil 3.4. Kalman Filtresi çevrimi	31
Şekil 4.1. Bulunulan bölgenin yatay ve düşey açısı.....	33
Şekil 4.2. Düşey açı.....	34
Şekil 4.3. Yatay ve düşey açı	34
Şekil 4.4. Yatay açı	35

Şekil 4.5. Hareketli araç pozisyon bilgisi.....	36
Şekil 4.6. Algoritma akışı.....	37
Şekil 4.7. Yatay açı grafiği.....	38
Şekil 4.8. Düşey açı grafiği	39
Şekil 4.9. Uygulama ekran görüntüsü	39

TABLolar LİSTESİ

Tablo 4.1. Yatay açI test verileri	37
Tablo 4.2. Düşey açI test verileri	38

ÖZET

Anahtar kelimeler: Gerçek zamanlı takip, Kalman Filtresi, Parabolik çanak antenin yatay ve düşey açıları

Günümüzde gerçek zamanlı sistemler ve uygulamalar; otobüs, araba gibi hareketli araçlarda yaygın olarak kullanılmaktadır. Hareketli platformlar için gerçek zamanlı uygulamalar değerli ve gereklidir. Gerçek zamanlı sistemler hareketli platformlarla uyum sağlamış ve vazgeçilmez parçası olmuştur. Bu tez hareketli araçlarda TV uydularının Kalman Filtresi ile takibi üzerine giriş sağlamaktadır. Aracın ve uydunun konum ve pozisyon bilgisi, android tabanlı bir aygıtın algılayıcıları yardımıyla toplanmaktadır ve android tabanlı bir uygulama ile analiz edilmektedir. Uydunun yatay ve düşey açılarını sürekli hesaplayan bir uygulama geliştirildi. İvmeölçer ve magnetometre algılayıcıları aracın pozisyon bilgilerinin alınması için kullanıldı. GPS algılayıcısı, aracın enlem ve boylam bilgisini elde etmede kullanıldı. Bu bilgiler parabolik çanak antenin yatay ve düşey açılarının hesaplanmasında kullanıldı.

AN IMPLEMENTATION ABOUT REAL TIME TRACKING OF A SATELLITE WITH AID OF KALMAN FILTER

SUMMARY

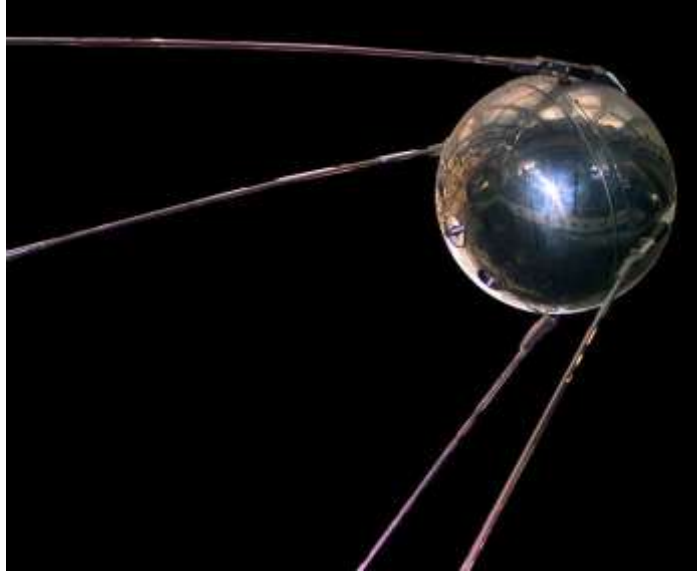
Keywords: Real time tracking, Kalman Filter, Azimuth and elevation angle of a parabolic dish antenna.

Nowadays real time systems and applications are in use widely on mobile platforms such as bus, car and etc. Real time applications are valuable and crucial to mobile platforms. Real time systems are well integrated with mobile platforms and become integral part of mobile platforms. This thesis provides an introductory level overview for real time tracking of TV satellites on moving vehicles using Kalman Filter. Information about the real time position of the vehicle and satellite are gathered with the aid of onboard sensors on an android device and then analyzed on an android based application. A real time application has been implemented to continuously calculate the azimuth and elevation angle of the satellite. Accelerometer and magnetometer sensors are used for collecting the information on the position of the moving vehicle. GPS sensor is used for collecting the latitude and longitude data of the vehicle. These data are then used for calculating the azimuth and elevation angle of a parabolic dish antenna.

BÖLÜM 1. UYDU SİSTEMLERİ VE PARAMETRELERİ

1.1. Giriş

İlk yapay uydu olan SPUTNIK 1'in Dünya yörüngesine 4 Ekim 1957 de fırlatılmasından bu yana uydu iletişimi şaşırtıcı bir değişime ve gelişime sahne oldu. Günümüzde uydular küresel iletişimde; telefon haberleşmesi, veri ağı, görüntü iletimi ve yayımlanması, televizyon ve radyo yayınlarının iletimi konusunda temel rol oynamaktadır [1]. Televizyon ve radyo yayınlarının uydu ile yapılması geniş bir bant genişliği ve kapsama alanı sunmaktadır.



Şekil 1.1. Sputnik 1 uydusunun resmi [2]

Uydu iletişim sistemlerinin diğer uzun mesafeli sistemlere göre çok önemli avantajlar sunarlar. Uydu iletişim sisteminin kurulum maliyetinin yüksek olmasına rağmen sağladığı geniş kapasite onu tercih edilir kılmaktadır.

Günümüzde, otobüs, araba gibi hareketli taşıtlarda gerçek zamanlı sistemler ve uygulamalar yaygın olarak kullanılmaktadır. Gerçek zamanlı uygulamalar, hareketli taşıtların değerli ve önemli parçaları oldular. Gerçek zamanlı uygulamalar, teknolojinin gelişmesiyle paralel olarak hareketli taşıtlara iyice uyum sağlayarak ayrılmaz bir parçası oldular. Bu tezde, televizyon uydularının hareketli araçlarda Kalman Filtresi ile gerçek zamanlı takibi konusu üzerinde çalışılmıştır.

Birinci bölümde, uyduların kullanım amaçlarına ve yörüngeye uzaklıklarına göre sınıflandırılması, uydu antenlerinin yapısı, uydu anten parametrelerinin hesaplanması, uydu yayın frekansları üzerinde durulmuştur. Bölüm sonunda hareketli araçların pozisyon bilgileri hakkında bilgi verilmiştir.

İkinci bölümde, algılayıcı sinyal iyileştirme üzerinde durulmuş, algılayıcılardan okunan verinin doğru ve tutarlı olması için yapılması gerekenler anlatılmıştır.

Üçüncü bölümde, Kalman Filtresi üzerinde durulmuş ve geniş olarak anlatılmıştır.

Dördüncü bölümde, hareketli araçlarda Kalman Filtresi ile TV uydularının gerçek zamanlı takibi uygulaması anlatılmıştır.

Beşinci bölümde uygulamanın değerlendirilmesi yapılmış ve öneriler üzerinde durulmuştur.

1.2. Uydu Sistemleri

Yapay uydu, insanoğlu tarafından geliştirilen ve gezegenlerin etrafına yerleştirilen çeşitli görevleri gerçekleştirmesi için yerleştirilen platformlardır [3]. Uydu iletişim sistemleri, farklı bölümlerden oluşan karmaşık sistemlerdir. Bir uydu sistemi incelendiğinde yer (Dünya), uzay ve kontrol birimlerinden oluştuğu görülmektedir.

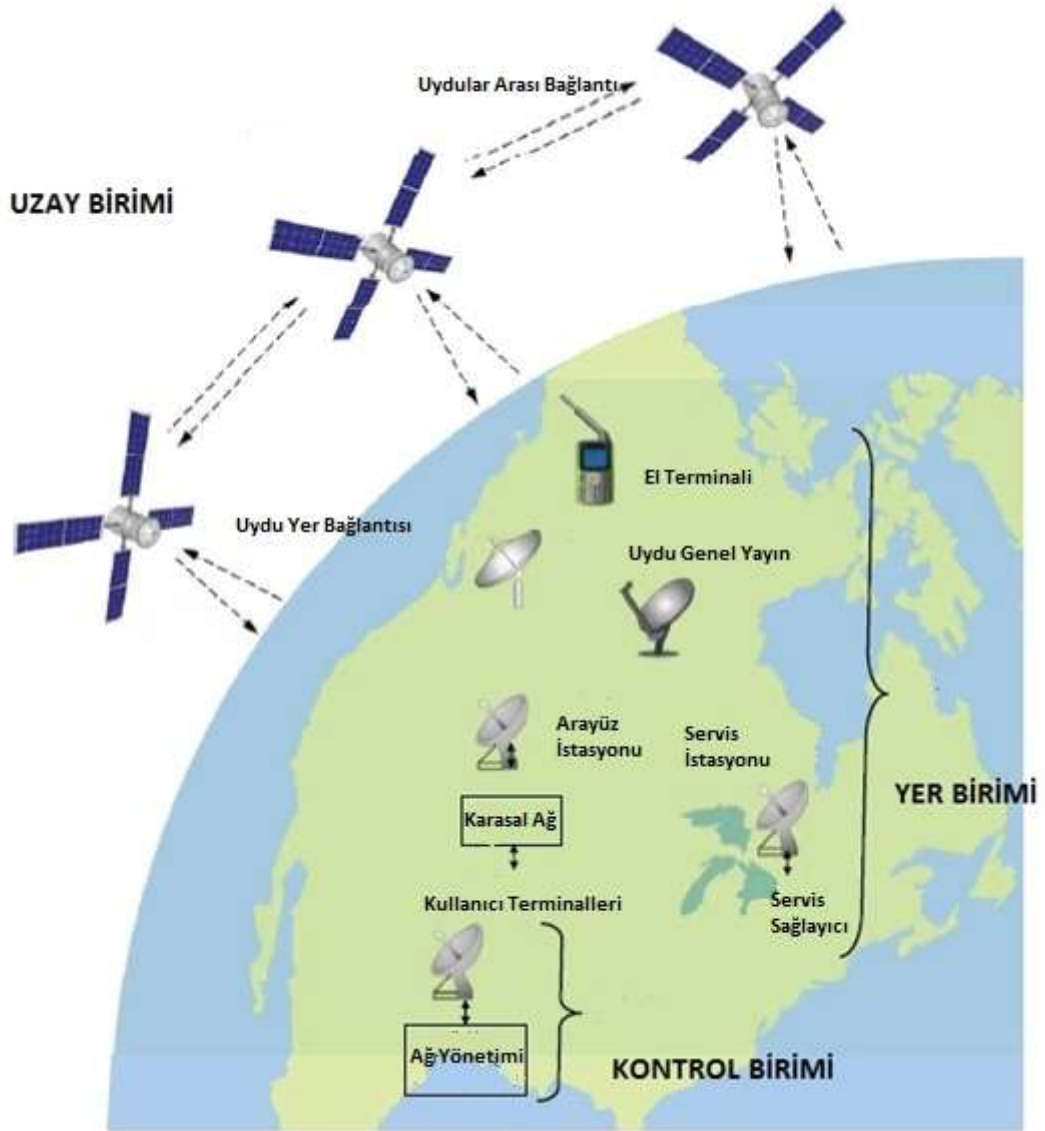
Yer birimi, uydunun tasarım amacına göre, iletişim trafiğinin yönetildiği yer sistemlerini içeren, kullanıcı birimi olarak da adlandırabileceğimiz birimdir [4]. Bu

birimde genel yayım hizmet servisleri ve kullanıcıların uzay birimi ile etkileşimi sağlayan sistemlerin bütünüdür.

Uzay birimi, uyduları ve uyduyu izleme, telemetri yani uydunun uzaktan kontrolünü, “telecommand” ve uyduda uzaktan yönetim ve lojistik faaliyetlerini içeren yer birimleri ve sistemleridir. Uydu, aktif elemanlarını kullanarak uzayda iletişimi sağlayan platformdur. Bu platform tasarım amacına farklılık gösterebilir temel olarak, güç kaynağı, yörünge kontrol, hareket kontrol, ısı kontrolü, telemetri, telecommand gibi sistemleri bünyesinde barındırır [5]. İzleme, uzaktan kontrol ve yönetim gibi sistemler uydunun açılı, yörünge ve kontrol fonksiyonlarını içeren alt sistemlerdir ve kontrol birimi olarak da adlandırılabilir. Kontrol birimi, yönetim, izleme bakım ve denetim sistemlerini içeren bütün yer sistemleridir.

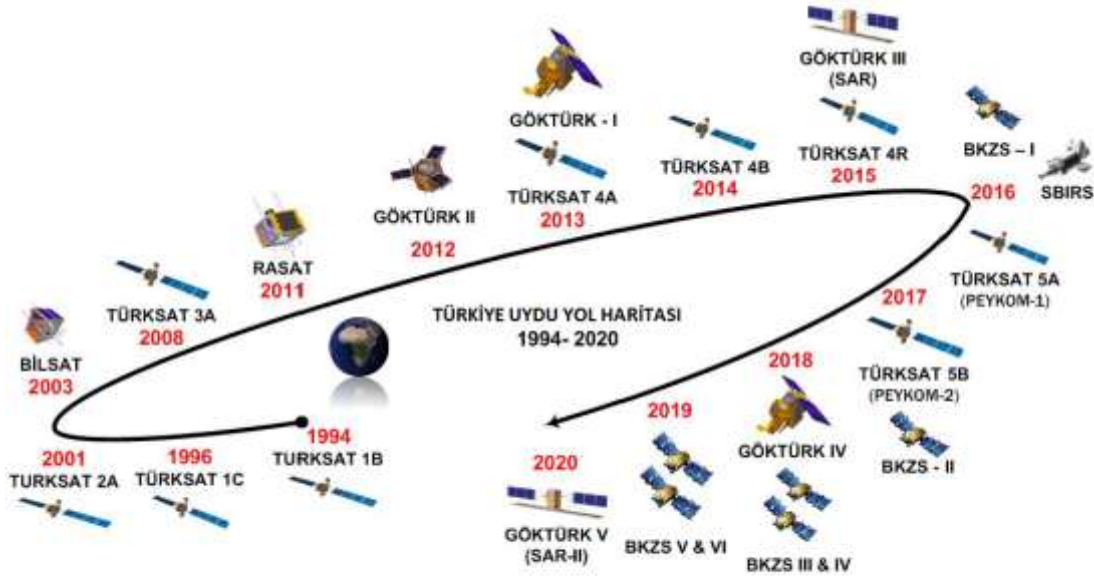
Uydu iletişim sistemlerinde veri haberleşmesi, yer biriminden uzay birimine doğru “uplink”, uzay biriminden yer birimine doğru ise “downlink” olarak adlandırılır.

Sabit uydu servisleri (Fixed Satellite Service), uydu ile dünya üzerinde belirli noktalar (istasyonlar) arasındaki radyo iletişim servsidir. Örneğin bir TV yayının uyduya genel yayım için yükleme bağlantısı servsidir. Diğer uydu servisleri ise, Mobil Uydu Servisi (Mobile Satellite Service), bir veya daha fazla hareketli yer istasyonları ile uydu arasındaki radyo bağlantısı servsidir. Bu bir uyduya bağlanabilen el terminali bile olabilir. Genel Yayım Uydu Servisi (Broadcasting Satellite Service) uyduya yüklenen veya uydudan genele yayım için yapılan dünya yüzeyinden küçük uydu antenleri ile alımı yapılan yayım hizmetidir [6].



Şekil 1.2. Uydu haberleşme birimleri [7]

Ülkemizde, ilk uydu yayını, Türksat 1A fırlatım aşamasında bilinmeyen bir nedenle düşmesinin ardından, 1994 yılında Türksat 1B'nin fırlatılmasıyla başlanmıştır. Uydu teknolojisi çalışmaları ise hızla devam etmemiştir. Bu konuda yıllara göre yapılan işleri resmeden şekil aşağıda verilmiştir.



Şekil 1.3. 1994-2020 Yılları uydu planlaması [8]

1.3. Kullanım Amaçlarına Göre Uydular

Uydular, uzayda belirli görevleri icra etmek için tasarlanırlar. Günümüzde uydular, özellikle iletişim, uzaktan algılama ve seyrüsefer sistemleri olmak üzere çok geniş alanlarda faaliyet göstermektedirler. Bu görevleri icra eden uyduların yerleştirildiği yörüngeler, icra edecekleri görevin özelliklerine bağlı olarak, farklılıklar göstermektedir.

1.3.1. Meteoroloji uyduları

Bu tür uydular, dünyadaki iklim durumlarını sürekli olarak izler ve hava tahminlerinde bulunurlar. Bunu hava fotoğraflarını çekerek, tarayıcıları (kızıl ötesi vb.) ile atmosferi tarayarak bilgi toplarlar ve elde edilen bilgileri radyo bağlantısı ile dünyadaki istasyonlara iletirler. Elde edilen bilgiler, hava tahminleri ve iklim değişiklikleri hakkında bilgi elde edilmesinde kullanılır [9].

1.3.2. Radyo ve tv yayın uyduları

Bu tür uydular, yüzlerce TV ve radyo kanalının çok geniş alanlara yayını yaparlar. Bu tür yayınların alınabilmesi uydu antenleri ile sağlanır. Örnek olarak Türksat 4A gibi uydular. Yayını alımı ise parabolic çanak antenler ile gerçekleştirilir.

1.3.3. Askeri uydular

Bu tür uydular askeri görevleri icra etmek için tasarlanırlar. Örnek olarak istihbarat toplama, askeri amaçlı iletişim servisleri, uydu üzerinden silah kontrol sistemleri vb. verilebilir.

1.3.4. Seyrüsefer uyduları

Bu tür uydular, dünya üzerinde konum belirlenmesi amacıyla dünya yörüngesine yerleştirilmiş uydulardır. Bu uydulardan alınan sinyaller yardımıyla konum tespiti yapılır.

1.3.5. Küresel iletişim uyduları

Küresel iletişim uyduları, ülkeler arasında telefon omurga yapısının uydu üzerinden sağlanması amacıyla ilk zamanlarda kullanıldı. Fakat daha sonra fiber optik kablolar bu konuda kullanılmaya başlandı. Küresel haberleşme uyduları iki uzak mesafe arasında bağlantı sağlayan uydular da örnek olarak verilebilir [10].

1.4. Yörüngelerine Göre Uydular

Bir uydu, bir roket yardımıyla etrafına yerleştirildiği gezegenin, güneş etrafında dönüşü gibi benzer bir yolu izler. Uyduların, gezegen etrafında dönerken izledikleri yola yörünge denir. Uyduların ana yapıları planlanırken, kullanma alanları, uzaydaki ömrü, üzerindeki cihazlar ve bunların ömürleri, güç sistemleri ve diğer yardımcı sistemler göz önüne alınarak tasarlanır. Gezegenlerin ve yapay uyduların uzaydaki hareketlerini açıklayan yasalara Kepler yasaları denir. Güneş Sisteminde bulunan

gezegenlerin hareketlerini açıklayan üç matematiksel yasadır. Alman matematikçi ve astronom Johannes Kepler (1572-1630) tarafından keşfedilmişlerdir. Kepler, Tycho Brahe tarafından yapılan gözlemler sonucu elde edilen verileri inceleyerek, Brahe'in gezegenlerin konumları ile tutmuş olduğu kayıtların görece basit olan üç adet matematiksel ifade ile açıklanabileceğini bulmuştur [11].

1. Yasa: Her gezegen, odak noktalarının birinde güneşin bulunduğu bir elips yörünge üzerinde hareket eder.
2. Yasa: Bir gezegeni güneşe bağlayan çizgi eşit zaman aralıklarında eşit alanlar tarar.
3. Yasa: Bir gezegenin yörüngesel periyodunun karesi, dolandığı elipsin ana eksen uzunluğunun küpü ile doğru orantılıdır.

Yörüngelerine göre uydular, alçak yörünge, orta yörünge, jeosenkron yörünge ve yüksek yörünge uyduları olarak gruplandırılır.

1.4.1. Jeosenkron yörünge uyduları

Jeosenkron yörünge uyduları, “geosynchronous equatorial orbit” GEO uyduları olarak bilinir. Dünya ile eş zamanlı olarak dönerler. Dünyada belirli bir noktadan bakıldığında sabit görünürler. Bu tür uyduların ayak izi büyüklüğü dünyanın 1/3 kadardır. Yörüngeleri daireseldir. Bu tür uyduların beklenen ömürleri yaklaşık 15 yıldır. Yerküreden, 37,786 km (yaklaşık olarak 36,000 km) uzağa yerleştirilmelidirler. Dünyanın dönüş hızıyla ve doğu yönlü olarak hareket etmelidirler. Dünyaya göre eğimleri sıfır derece olmalıdır [12].

Güneş ve ayın kütle çekim etkisiyle yörüngelerinde bozulmalar olur. Dünyanın dairesel olmamasıda dönüş hızlarının sürekli ayarlamasını gerektirir. Merkez kaç kuvvetindeki değişmelerde yörüngelerinde bozulmalar meydana getirir. Bu tür uydular TV ve radyo yayıncılığında ayrıca telefon şebekesinin omurgası olarakda kullanılabilirler.

GEO, uydularında işaret gecikmeleri yüksektir. İşaret en az 72000 km seyahat eder fakat geniş bir kapsama alanları vardır.



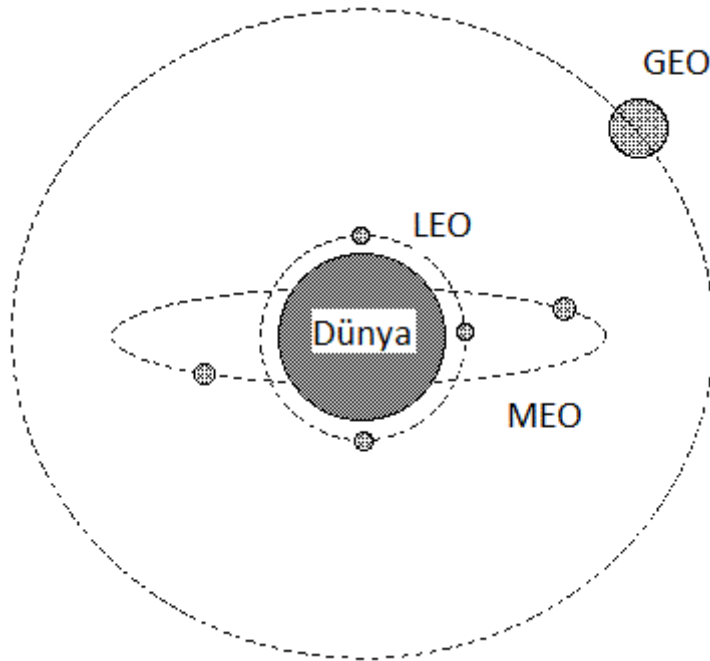
Şekil 1.4. Jeosenkron uydu [13]

1.4.2. Alçak yörünge uyduları

Bu tür uydular, yeryüzünden 160-2000 km yukarıya yerleştirilirler. Küçük bir yörüngede döndüklerinden dolayı, periyotları da 88 ile 127 dakika arasındadır ve kısadır. LEO (low Earth orbit) yörüngesindeki uydular, dünya üzerindeki her noktadan yüksek yükselme açısı gerektiğinden, yüksek kalitede iletişim bağlantısı sağlarlar. Her LEO uydusu, yeryüzünden kısa bir süre görünür. Yerküreye yakın olduklarından, işaret gecikmeleri az olur. Bu yüzden küresel iletişim sistemlerinde ve uzaktan algılamada kullanılırlar. LEO uydularının genel problemi, kısa ömürleridir [14].

1.4.3. Orta yörünge uyduları

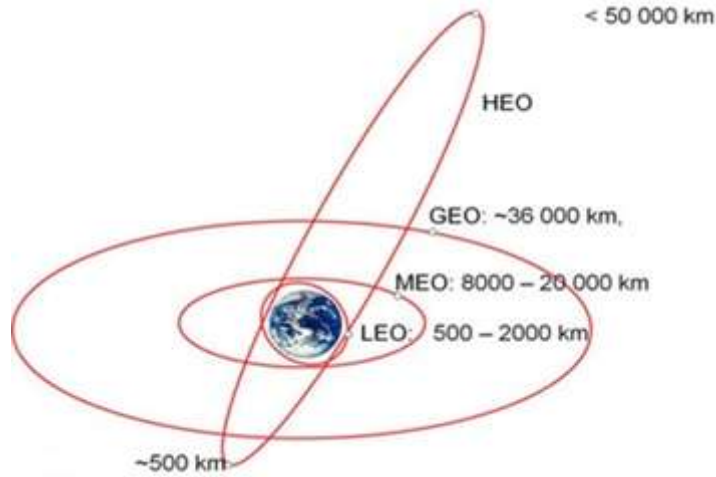
MEO (Medium Earth orbit), uyduları, LEO ve GEO uyduları arasına yaklaşık yeryüzünden 1,600 km ile 19,300 km arasındaki yörüngeye yerleştirilirler. 10000 km deki yörüngeyi kullanan bir MEO sistemi, bir düzine uyduyu gerektirir. Bu LEO'dan az, GEO'dan fazladır. Uydu periyotları 120 dakika ile 12 saat arasında değişir. Geniş bir alanı kapsar. Buna rağmen gecikmeleri 70-80 ms'dir ve yüksek iletim güçlü özel antenlere ihtiyaç duyarlar [15].



Şekil 1.5. LEO, MEO ve GEO yörüngeleri [16]

1.4.4. Yüksek yörünge uyduları

Yüksek yörünge uyduları (HEO- High Earth Orbit), dünyaya olan mesafeleri en fazla olan uydulardır. Jeosenkron yörünge uydularının bulunduğu kuşaktan sonraki kuşaktır. Yörüngelerinde yirmi dört saatten fazla sürede dolaşırlar. Dönüş hızları Dünya'nın dönüş hızından düşüktür [17].



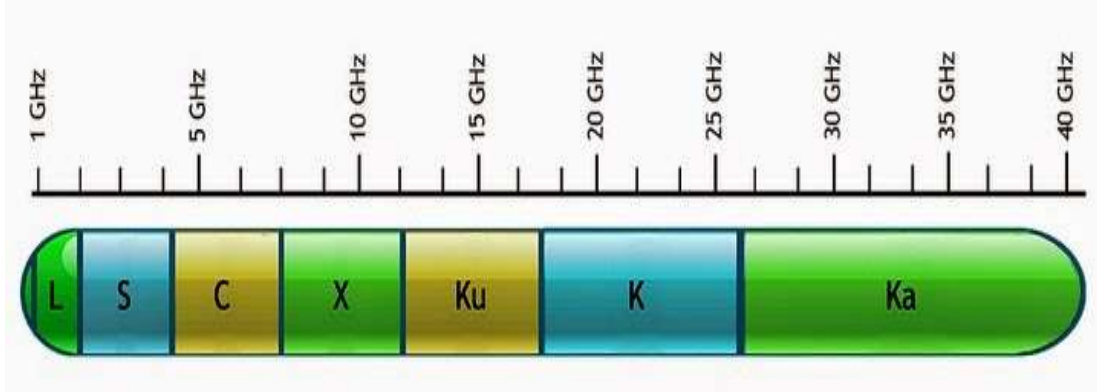
Şekil 1.6. Uydu yörüngeleri ve yörünge mesafeleri [18]

1.4.5. Diğer yörünge uydular

Diğer yörünge uydularına bakıldığında, Güneş senkron uydular; irtifasının ve yörünge eğikliğinin, yörüngede yer alan uydunun gezegen üzerindeki bir noktadan her bir geçişini aynı yerel zamana denk getirecek şekilde seçildiği yörüngelerdir. Bu tip bir yörüngede uydu devamlı olarak aynı miktarda Güneş ışığına maruz kalacaktır. Görüntü alan gözlem uyduları ve hava tahmin uyduları için kullanışlı bir yörüngedir. Hohman Transfer Orbit (HTO), ise yüksek derecede eliptik yörüngeye sahip orta yörünge uydusudur. Kutupsal Yörünge Uyduları Polar Orbit, yörüngeleri kuzey ve güney kutup bölgesinden geçen uydulardır. Burada değinilmeyen farklı tip yörüngesel uydularda vardır [19].

1.5. Uydu İletişiminde Kullanılan Frekans Bantları

Hızlı gelişen uydu teknolojileri ve uygulamaları sürekli olarak artmaktadır. Uydular sadece radyo iletişimi amaçlı değil aynı zamanda astronomi, hava tahmini, yayıncılık ve haritalama gibi bir çok uygulama tarafından kullanılır. Çok çeşitli uydu frekans bantları, tahsis edilen aralıklar kolaylıkla kullanılabilirler. Yüksek frekans bantları çok geniş bant genişliği sağlarlar. Uydu teknolojilerinde kullanılan frekans bantlarını inceleyelim.



Şekil 1.7. Uydu frekans bantları [20]

L - Bandı: 1 - 2 GHz arasındaki frekans aralığıdır ve GPS uyduları ve uydu telefon sistemleri tarafından kullanılır.

S - Bandı: 2 – 4 GHz arasındaki frekans aralığıdır ve meteoroloji radarları ve iletişim uyduları, özellikle NASA tarafından iletişim ISS ve uzay mekikleri ile iletişim amaçlı kullanılır.

C- Bandı: 4 - 8 GHz arasındaki frekans aralığıdır öncelikli olarak uydu iletişimi amaçlı kullanılır.

X – Bandı: 8 – 12 GHz arasındaki frekans aralığıdır ve öncelikli olarak askeri amaçlı bir çok radar uygulamasında kullanılır. Alt bantları ise hava gözlem, hava trafik kontrolü, deniz kıyı trafik kontrolü ve araç takip ve hız tespit gibi uygulamalarda kullanılır.

Ku- Bandı: 12 – 18 GHz arasındaki frekans aralığıdır ve uydu iletişimde kullanılır. Avrupada astra uydusunda olduğu gibi genel (TV, Radyo) yayın frekansıdır.

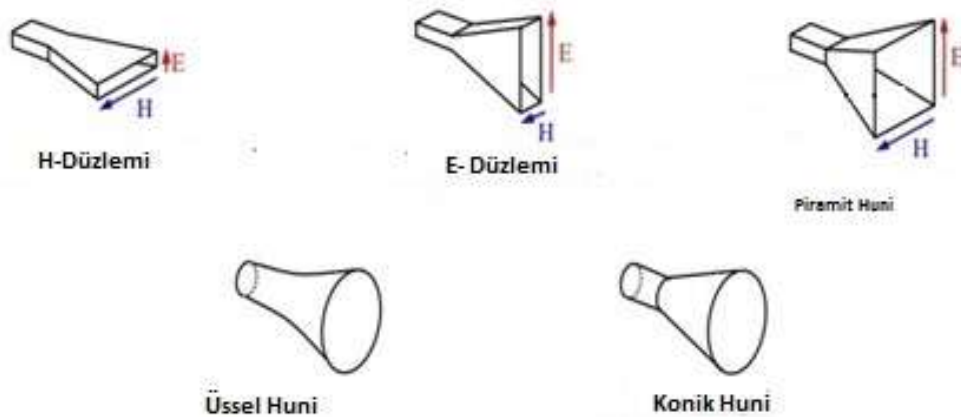
Ka – Bandı: 26 – 40 GHz arasındaki frekans aralığıdır. İletişim uyduları tarafından ve askeri uçaklar tarafından yakın mesafe hedef radarı olarak kullanılır [21].

1.6. Uydu Antenleri

Anten, elektrik enerjisini elektromanyetik ışımaya dönüştüren (verici anten) ya da elektromanyetik ışımayı elektrik enerjisine dönüştüren (alıcı anten) bir iletken ya da iletkenlerden oluşan sistemdir. Uydu iletişimde yaygın olarak kullanılan anten türleri, huni antenler, faz dizi antenler ve çanak antenlerdir.

1.6.1. Huni antenler

Huni antenler, metal bir koni ya da piramit biçiminde yapılmış ve dalga kılavuzu ile beslenen yönlü antenlerdir. UHF bandında ve 300 MHz'in üzerindeki mikrodalga uygulamalarında kullanılır. Huni antenler genellikle, parabolik çanak antenlerde besleyici olarak ve diğer antenlerin kazanç ölçümlerinde ve kalibrasyonunda kullanılırlar. Huni antenler genellikle yönlü ışımaya sahiptirler. Huni antenlerinin ön yüzündeki açıklık, anten açıklığı olarak adlandırılır. Huni antenler genellikle 10-20 dB arasında değişmekte olan ve 25 dB'e kadar da olabilen yüksek anten kazançlarına sahiptirler. Oldukça geniş empedans bant genişlikleri vardır. Giriş empedansları, geniş bir frekans aralığında yavaşça değişirler. Pratikte huni antenler için bant genişliği, çalışma frekansı arttığında anten kazancıda artar. Bunun nedeni anten açıklığı dalga boyu ile ölçülür. Yüksek frekanslarda dalga boyu küçülür. Böylece anten açıklığı da genişler. Aşağıda huni anten çeşitleri görülmektedir [22].

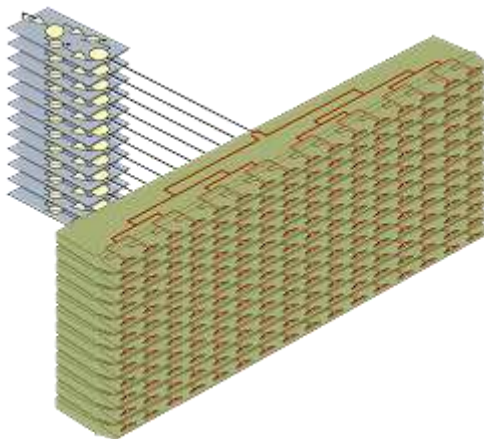


Şekil 1.8. Huni anten çeşitleri [23]

1.6.2. Faz dizi antenler

Faz dizi antenler, her bir ışına elemanı farklı faz açılarıyla besleyen bir anten grubudur. Bu antenler sinyallerde girişim prensibine göre çalışır. Aynı fazdaki sinyaller birbirine eklenerek kuvvetlenirler, zıt faza sahip sinyaller ise birbirini yok ederler. İki kaynaktan (huni antenden) eşzamanlı iki darbe gönderildiğinde bir girişim meydana gelir; sinyal ana yönde kuvvetlenirken, yan yönlerde zayıflar. Burada her iki besleyiciden yayınlanan sinyaller aynı fazdan beslenmektedir. Sinyal bu durumda ana yönde kuvvetlenir. Bir faz dizi antenin, en öne çıkan özelliği; her bir yayın elemanı için faz kayma miktarının ayarlanabilmesi ve bunun sonucu olarak ışına ana yönünü değiştirilebilmesidir. Hem yatay ve hem de düşey düzlemde ışına demeti elde edebilmek için bir anten düzleminde çok sayıda ışına elemanı kullanmak gerekir. Işına sinyalleri, bir faz kaydırma devresi ile kontrol edildiğinden, yayılma yönlerinin de elektronik olarak kontrol edilmesi mümkün olur [24].

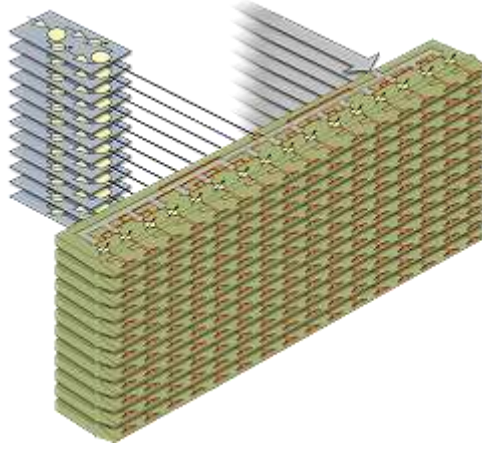
Doğrusal faz dizi antenler, ortak bir faz-kaydırıcı tarafından kontrol edilen ışına elemanları satırlarından meydana gelmektedir. Her bir satır için bir faz kaydırıcı devre gerekir. Çok sayıda, birbiri üzerine düşey istikamette konulmuş, doğrusal antenler bir anten düzlemi oluştururlar. Basit yerleşimi vardır fakat tarama işlemi sadece bir düzlemde yapılabilir.



Şekil 1.9. Doğrusal faz dizi anten [25]

Düzlemsel faz dizi antenler, her birisi kendisine ait faz kaydırıcısına sahip tekli elemanlardan oluşur. Her bir ışına elemanı için bir faz kaydırıcı devre gerekir.

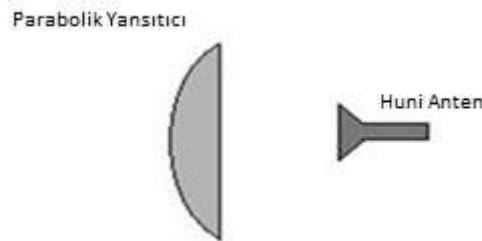
Elemanlar bir matriste dizildiği gibi dizilmiş olup, elemanların düzlemsel yerleşimi toplam anteni oluşturur. Tarama işlemi iki düzlemde yapılabilir fakat düzenleme karmaşıklıkça kontrol için çok sayıda devreye ihtiyaç duyulur.



Şekil 1.10. Düzlemsel faz dizi anten [26]

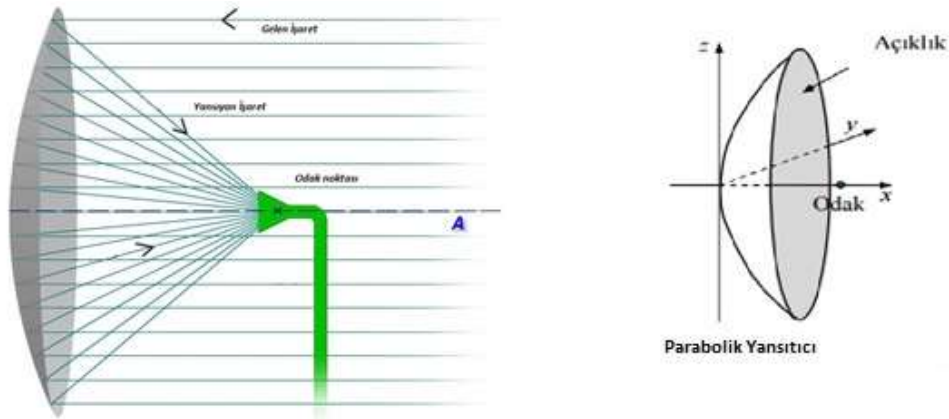
1.6.3. Parabolik antenler

Parabolik Antenler içinde en yaygın olanı parabolik yansıtıcı antenler olup karasal mikrodalga sistemlerinde, uydu iletişimde kullanılır. Parabolik yansıtıcı yüzey bir parabolün eksenini etrafında döndürülmesi ile elde edilen yüzeydir, buna paraboloid denir. Paraboloidin x eksenine dik kesiti daire şeklindedir ve anten açıklığı olarak adlandırılır [27].



Şekil 1.11. Parabolik çanak anten yapısı

Parabolik yansıtıcılar 30-40 dB gibi yüksek kazançla sahiptirler. Bant genişlikleri oldukça yüksektir. Küçük parabolik çanak antenler 2-28 GHz frekans aralığında çalışırlar. Geniş parabolik çanak antenler ise 30-300MHz aralığında çalışırlar. Basit yapısı aşağıdaki şekilde gösterilmiştir. Antenin yapısı incelendiğinde bir parabolik yansıtıcı ve bu yansıtıcıya yönelmiş bir besleyici antenden (huni anten) oluşur. Yansıtıcı çanak dalga boyundan büyük olmalıdır.



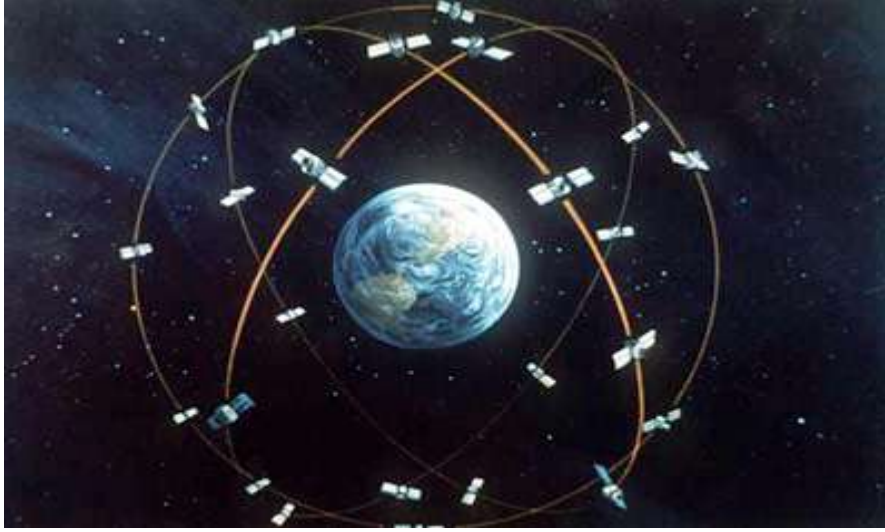
Şekil 1.12. Parabolik çanak anten [28]

Çanak antenlerin en güçlü ışın ve alma yaptıkları yön çanağın eksenidir. Bu durum odakta bir nokta kaynağı için teorik olarak doğrudur. Uygulamada odakta bir huni anten ya da ikincil yansıtıcı olacağından, parabolik yansıtıcıdan yansıyan ışın biraz dağılmaktadır. Buna karşılık, çanak antenlerin hüzm genişlikleri oldukça dardır [29].

1.7. Hareketli Araç Konum Bilgisi

Hareketli araçlarda konum bilgisini, GPS (Global Positioning System, Küresel Konumlama Sistemi) alıcıları ile tespit edebiliriz. GPS, Dünya üzerinde herhangi engelsiz bir görüş hattında, dört veya daha fazla uydusu ile her türlü hava koşulunda yer ve zaman bilgileri sağlayan uzay tabanlı uydu navigasyon sistemidir. Düzenli olarak kodlanmış bilgi yollayan bir uydu ağıdır ve uydularla arasındaki mesafeyi ölçerek Dünya üzerindeki kesin yeri tespit etmeyi sağlar. Küresel ölçekte çalışan bu tür sistemlerin ilkidir. Bu sistem, Amerika Birleşik Devletleri Savunma Bakanlığı'na ait, yörüngelerinde sürekli olarak ilerleyen uydulardan oluşur. Savunma Bakanlığı

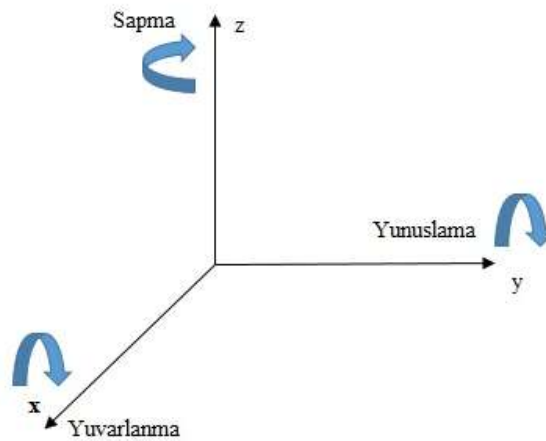
tarafından desteklenen sistem, bir GPS alıcısı ile kullanılacak şekilde herkes tarafından erişilebilirdir. Uydular bir tür radyo sinyali yayarlar ve yeryüzündeki GPS alıcıları bu sinyalleri alıp yorumlayarak konum belirlenmesini gerçekleştirir [30].



Şekil 1.13. GPS uyduları [31]

1.8. Hareketli Araç Pozisyon Bilgisi

Araç pozisyon bilgisini, ivme ölçer ve gyro sensörleri ile ölçebiliriz. aracın üç eksendeki x, y ve z eksenlerine göre durumunu açısal olarak belirtir. Üç eksendeki hareketler. Pitch: Yunuslama, Yaw: Sapma, Roll: Yuvarlanma olarak adlandırılır [32].



Şekil 1.14. Araç pozisyon bilgisi

BÖLÜM 2. ALGILAYICI SİNYAL İYİLEŞTİRME

2.1. Giriş

Algılayıcılar, gürültü ve zamanla bozunma nedeniyle fiziksel büyüklükleri kusursuz olarak ölçemezler. Veri üzerinde, bu problemler hatalar meydana getirir. Bu hataları azaltacak algoritmalar ve teknikler üzerinde çalışılmaktadır. Algılayıcının çıkışından alınan verileri süzerler ve algılayıcı gruplarından alınan verileri ise biraraya getirirler. Bu bölümde algılayıcılardan alınan verilerin iyileştirilmesi üzerinde duracağız [33].

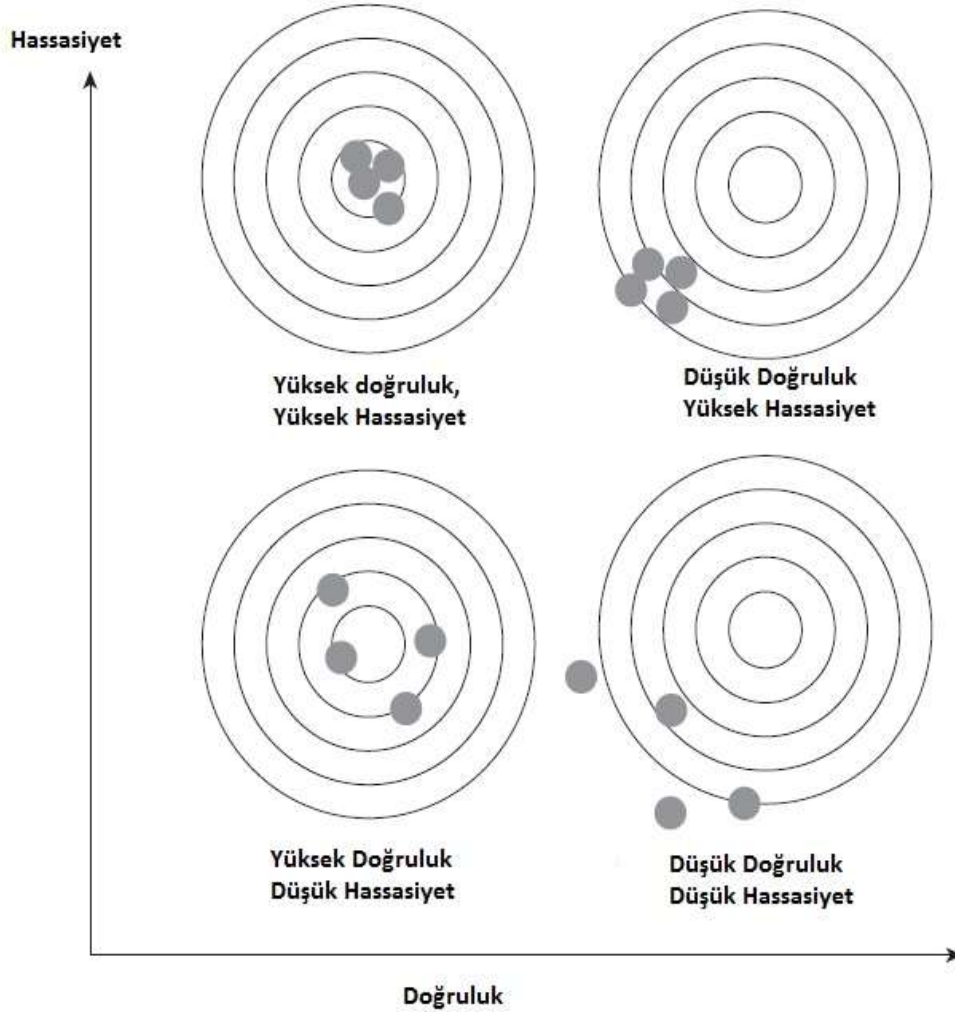
2.2. Doğruluk ve Hassasiyet

Algılayıcı doğruluğu ve hassasiyeti irdelemek için iki değeri bilmemiz gerekir. Bunlardan biri algılayıcının ölçmeye çalıştığı gerçek değerdir. Diğeri ise ölçtüğü yani algılayıcının çıkışından okunan değerdir. Yüksek doğruluğun anlamı ölçülen değer gerçek değere olan yakınlığıdır. Yüksek duyarlılık ise ölçülen değerlerin birbirine olan yakınlığıdır [34].

2.3. Hata Türleri

Algılayıcılardan veri okunurken karşılaşılabileceğiniz hataların türlerinin bilinmesi önemlidir. Algılayıcıdan okunan verilerin hatalarının tespitini sağlayan algoritmaların geliştirilmesinde anahtar rol oynar. Hata türleri, insan kaynaklı hatalar, sistematik hatalar ve rassal hatalardır. İnsan kaynaklı hatalar, insanlardan kaynaklanan; şöyleki yanlış okuma kaynaklı hatalar örnek olarak verilebilir. Sistematik hatalar ise ölçümün doğruluğunu etkileyen genel sabit değerli hatalardır. Örneğin bir pusuladan mıknatıs yakınında okuma yapılmasıdır. Bazı durumlardada kalibrasyon yapılması bu hatayı önler. Rassal hatalar ise değeri belli olmayan ölçümü etkileyen hatalardır. Belirli tekniklerle veriden ayrıştırılamazlar. Gürültü, ölçülen

değerdeki rassal oynamalardır. Gürültü kahverengi gürültü, beyaz gürültü gibi gruplandırılabilir. Sapma, zamanla oluşan ölçülen değer gerçek değerden uzaklaşmasıdır. Sıfır offset, sıfır büyüklüğü ölçülürken, ölçülen çıkış değerinin sıfır olmadığı durumdur. Bu durumda algılayıcı sıfırlanmalıdır. Ölçümde gecikmeler ve veri kayıplarında yaşanabilmesi olası durumlara örnek verilebilir [35].



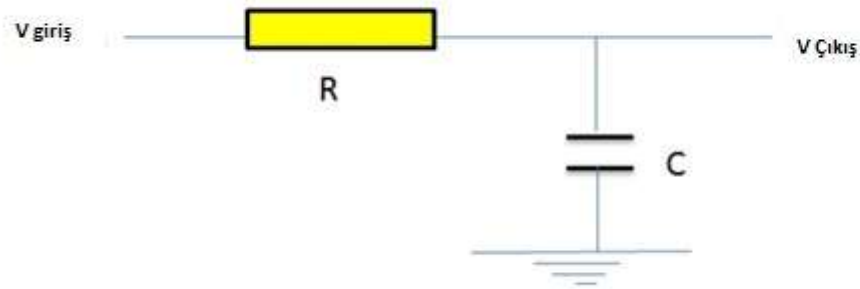
Şekil 2.1. Doğruluk hassasiyet karşılaştırması [36]

2.4. Sinyal İyileştirme Filtreleri

Sinyal iyileştirme filtreleri, algılayıcılardan alınan yada elde edilen veri kümesinin içindeki önemli olan veriyi elde etmeye yarar. Sinyalden gürültü gibi bozucu verileri atılmasını sağlar. Şimdi işaret iyileştirme filtrelerinin bazılarını inceleyeceğiz.

2.4.1. Alçak geçiren filtre

Girişten gelen işaretin yüksek frekanslı bileşenlerini bloklar alçak frekanslı bileşenlerini geçiren filtredir.



Şekil 2.2. Alçak geçiren filtre devre şeması [37]

Ağırlıklı yumuşatma (tesviye), alçak geçiren sayısal filtrenin gerçekleştirilmesinde genellikle kullanılan bir metottür. İşlem sabit bir parametre ile ölçülen değerin çarpılarak yeni değerin elde edilmesidir.

$$\text{Yeni Değer} = (\text{son değer}) + x_i * a - (\text{son değer}) * a$$

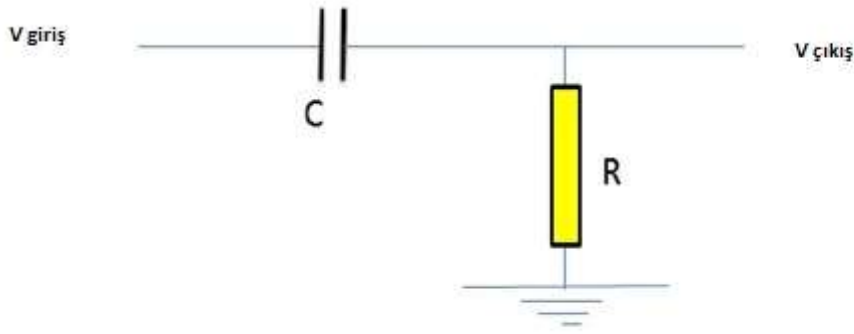
Alçak geçiren filtrenin java dilinde yazılmış kod örneği aşağıdadır.

```
float a = 0.1f;
x = event.values[0];
mLowPassX = lowPass(x, mLowPassX);

// simple low-pass filter
float lowPass(float current, float last)
{
    return last * (1.0f - a) + current * a;
}
```


2.4.2. Yüksek geçiren filtre

Girişten gelen işaretin yüksek frekanslı bileşenlerini geçirip alçak frekanslı bileşenlerini bloklayan filtredir.



Şekil 2.3. Yüksek geçiren filtre şeması [38]

Sayısal yüksek geçiren filtrenin java dilinde yazılmış kod örneği aşağıdadır.

```

    x = event.values[0];
    mHighPassX = highPass(x, mLastX, mHighPassX);
    mLastX = x;

float highPass(float current, float last, float filtered)
{
    return a * (filtered + current - last);
}

```

2.4.3. Basit hareketli ortalama

Basit hareketli ortalama, dönen yada yürüyen ortalama olarak adlandırılır. En güncel k kadar verinin aritmetik ortalamasıdır [39]. İşaret iyileştirmede tercih edilen bir yöntemdir. Basit hareketli ortalama, yeni gözlem değeri elde edildiğinde ortalama hesabına bu yeni değeri ilave ederken, bir önceki ortalamaya dahil edilen en eski

dönem değerini çıkarır. Böylece en son k kadar gözlem değerlerinin ortalaması değer olarak alınır.

Basit hareketli ortalamaya ait java dilinde yazılmış kod örneği aşağıdadır.

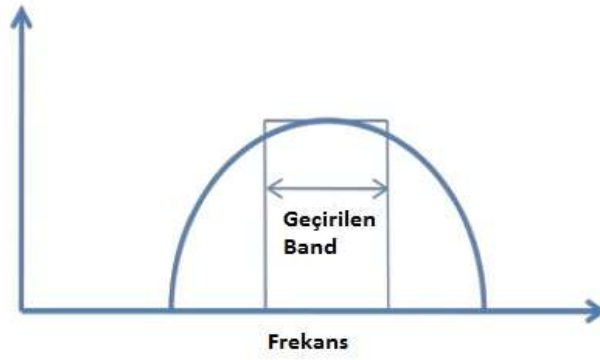
```
public class MovingAverage
    {
        private float circularBuffer[];
        private float avg;
        private int circularIndex;
        private int count;
        public MovingAverage(int k)
            {
                circularBuffer = new float[k];
                count = 0;
                circularIndex = 0;
                avg = 0;
            }
        /* Get the current moving average. */
        public float getValue()
            {
                return avg;
            }
        public void pushValue(float x)
            {
                if (count++ == 0)
                {
                    primeBuffer(x);
                }

                float lastValue = circularBuffer[circularIndex];
                avg = avg + (x - lastValue) / circularBuffer.length;
                circularBuffer[circularIndex] = x;
                circularIndex = nextIndex(circularIndex);
            }
    }
```

```
public long getCount()
    {
        return count;
    }
private void primeBuffer(float val)
    {
        for (int i = 0; i < circularBuffer.length; ++i)
            {
                circularBuffer[i] = val;
            }
        avg = val;
    }
private int nextIndex(int curIndex)
    {
        if (curIndex + 1 >= circularBuffer.length)
            {
                return 0;
            }
        return curIndex + 1;
    }
}
```

2.4.4. Bant geçiren filtre

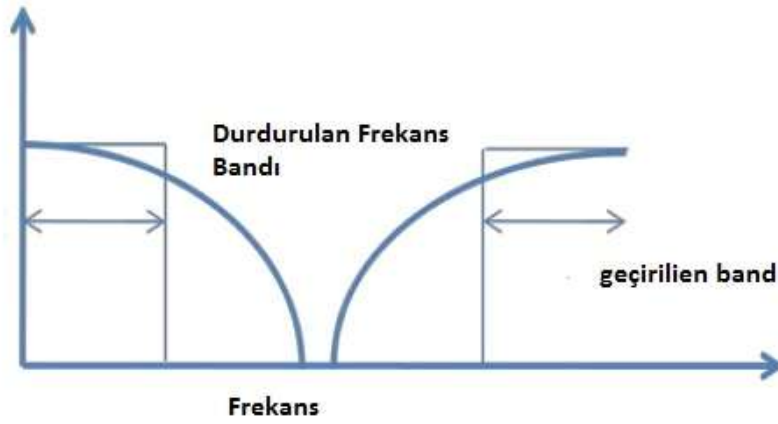
Bazı uygulamalarda, belirli bir bandın yani frekans aralığındaki sinyal bileşenlerinin geçirilmesi gerekir bu tür uygulamalarda alçak geçiren ve yüksek geçiren filtrelerin beraber kullanılmasıyla bant geçiren filtre tasarlanır.



Şekil 2.4. Bant geçiren filtre [40]

2.4.5. Bant durduran filtre

Belli bir frekans aralığını alt ve üst değerlerinin geçiren aradaki değerlerin durduran filtredir. Alçak geçiren ve yüksek geçiren filtrenin paralel bağlanmasıyla elde edilir.

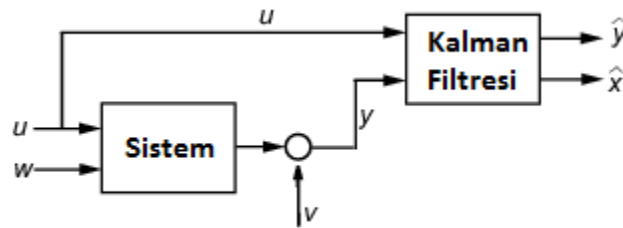


Şekil 2.5. Band durduran filtre [41]

BÖLÜM 3. KALMAN FİLTRESİ

Bir sistemin durumlarını gösteren durum değişkenlerinin gözlemlenmesi, eğer mümkün olmuyorsa kestirimi, tahmin edilmesi lineer sistem teorisinin önemli çalışma alanlarıdır. Ortam ve çevre koşullarının sisteme etkisi olarak kabul edilen gürültü nedeniyle durum değişkenlerinin veya sistemin durumlarının yeterince gözlemlenemeyeceği açıktır. Tam olarak değeri tahmin edilemeyen durum değişkenlerinin kestiriminden dolayı, olasılık tabanlı yaklaşımlar bu sürecin sürekli merkezinde yer almışlardır [42].

Kalman Filtresi 1960 yılında Macar asıllı Amerikalı Rudolf Emil Kalman tarafından geliştirilmiştir ve bazen “Optimal Kestirimci” bazen “Lineer Kestirimci” bazende “Durum Gözlemleyici” olarak bilinir. Kalman Filtresi, Wiener Filtrenin devamı olarak, ondan farklı olarak giriş işaretinin durağan olmayan sistemlerde uygulanabilirliği ile önemli katkı sağlamıştır. Bayes teorisinden yararlanan ve raslantısal olan bu yaklaşım günümüzde bir çok alanda uygulanmaktadır. Kalman Filtresi en küçük kareler yönteminde esinlenerek sabit hatayı değil hatanın değişimini (hata varyansını) dikkate almaktadır [43]. Sistemin blok diyagramı aşağıda verilmiştir.



Şekil 3.1. Kalman Filtresi blok diyagramı [44]

Kalman Filtresi, sisteme verdiğimiz girişe (u) ve ölçülen gürültülü çıkışa (y) bakarak sistemin gerçek çıkışını ve durumunu tahmin etmeye çalışır. Burada (w)'ya işlem (proses) gürültüsü denir ve sistem modelinde yukarıda sayılan nedenlerden dolayı meydana gelen idealden sapmaları temsil eder. Ayrıca (v)'ye ölçüm gürültüsü denir ve ölçümde meydana gelen sıkıntıları temsil eder. Kalman filtresinde bu gürültülerin sisteme aşağıdaki gibi etki ettiği varsayılır [45].

$$\hat{x} = Ax + Bu + Gw \quad (4.1)$$

$$\hat{y} = Cx + Du + Hw + v \quad (4.2)$$

Kalman Filtresinin sistemin gerçek çıkışı için yaptığı tahmin \hat{y} ile, ve sistemin durumu için yaptığı tahmin \hat{x} ile gösterilir.

3.1. Durum Uzayı Modeli

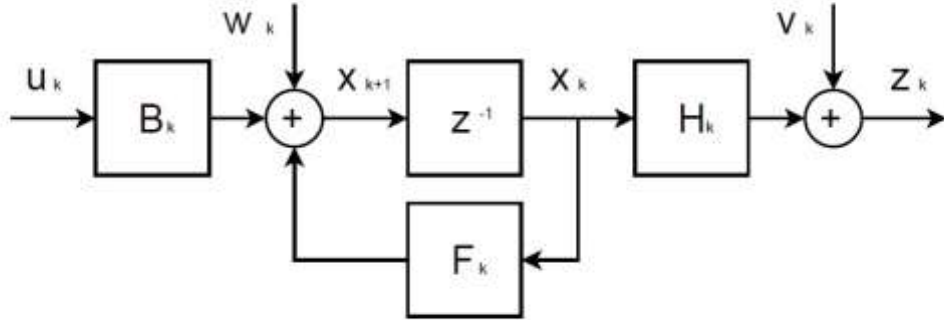
Doğrusal durum uzay modeli, $f(x, u, w)$ ve $h(x, v)$ fonksiyonlarının durum ve girdide doğrusal olduğu modeldir. Doğrusal olan bu fonksiyonlar, hesaplamaları doğrusal cebire indirgeyen F , B ve H matrisleri ile ifade edilebilir. Bu şekildeki durum uzay modeli şu şekilde gösterilir:

$$x_{k+1} = f(x_k, u_k, w_k) \text{ ve } z_k = h(x_k, v_k) \quad (4.3)$$

$$x_{k+1} = F_k x_k + B_k u_k + w_k \quad (4.4)$$

$$z_k = H_k x_k + v_k \quad (4.5)$$

Bu doğrusal modelde hesaplama ve analiz kolaydır. Kullanıcıya denetlenebilirlik, gözlenebilirlik, frekans yanıtı gibi özellikleri irdeleme imkânı sunar. Doğrusal durum modelleri ya gerçek doğrusal süreçler, ya da doğrusal olmayan süreçlerin basit bir şekilde birinci derece Taylor yaklaşımı aracılığı ile doğrusal hale getirilmiş biçimleri üzerine kurulmuş modellerdir [46].



Şekil 3.2. Doğrusal durum uzay modeli [47]

3.2. Gözlemleyici Tasarım Problemi

Doğrusal sistem teorisi alanında, gözlemleyici tasarım problemi olarak bilinen genel bir problem vardır. Temel problem, çıkışları bilinen bir doğrusal sistemin iç durumlarının tahmin edilmesidir. Bu, kara kutu (black box) problemine benzediği düşünülen, sadece çıkışların izlenebildiği fakat sistemin içindekilerin direk gözlenemediği bir durumdur.

Temel probleme yapılan bir çok yaklaşım, durum uzayı modeli olarak daha önce ifade edildi. Sistem durumunun geçişini modelleyen tipik bir sisteme durum modeli vardır. Bu genellikle doğrusal rassal fark denklemi olarak bilinir. Aşağıda ifade edilmiştir:

$$\chi_k = A \chi_{k-1} + B u_k + w_{k-1} \quad (4.6)$$

Buna ek olarak sistemin durumları ile ölçümler arasındaki ilişkiyi tanımlayan ölçüm modeli vardır:

$$z_k = H_k \chi_k + v_k \quad (4.7)$$

Burada, w_k ve v_k terimleri, giriş ve ölçüm gürültülerini temsil eden rassal terimlerdir. Beyaz gürültü olan w_k , sıfır ortalamalı ve Q kovaryansına sahiptir, girişle ilişkisizdir. Beyaz gürültü olan v_k , sıfır ortalamalı ve R kovaryansına sahiptir, girişle ilişkisizdir yani ortogondur [48].

3.3. Kalman Filtresi

Sistemin önceki bilgileriyle birlikte giriş ve çıkış bilgilerinden sistemin durumlarını tahmin edilebilen filtredir. Kalman Filtresi, geleneksel tahmin edicilerde olduğu gibi filtreleme özelliğine rağmen, sistemin ölçülemeyen durumlarını tahmin etmek için çok güçlü ve yeteneklidir. Gürültülü veriler üzerinde özyinelemeli gerçek zamanlı çalışarak hataları, enaz-kareler eğriye sığdırma yöntemi ile filtre eder ve sistemin fiziksel karakteristiklerinin modellenmesi ile üretilen gelecek durumun matematiksel tahminine göre optimize eder.

Kalman Filtresi pek çok farklı alanda sistemin durumunu, değerlerini kestirebilen (tahmin edebilen) bir yöntemdir. Matematiksel olarak doğrusal sistemlerin (lineer sistemlerin-hareket denklemleri birinci derece olan denklemler) durumlarını tahmin eder. Pratikte çok faydalı bir filtredir. Ayrıca teorik yönüde güçlüdür. Çünkü mevcut filtreler içinde kestirim hatasını (tahmini) minimize eden (gittikçe azaltan) tek filtredir [49].

3.4. Ayrık Kalman Filtresi

Kalman Filtresi, $x \in \mathbb{R}^n$ 'de doğrusal stokastik fark denklemi ile ifade edilen ayrık zamanlı kontrol edilen sürecin, durumlarını tahmin eder:

$$\chi_k = A \chi_{k-1} + B u_k + w_{k-1} \quad (4.6)$$

Ölçüm değeri ile $z \in \mathbb{R}^m$

$$z_k = H_k \chi_k + v_k \quad (4.7)$$

w_k ve v_k terimleri, giriş ve ölçüm gürültülerini temsil eden rassal terimlerdir. Birbirinden bağımsız normal dağılıma sahip, beyaz gürültü olduğu varsayılır.

$$P(w) \sim N(0, Q) \quad (4.8)$$

$$P(v) \sim N(0, R) \quad (4.9)$$

Pratikte, giriş gürültü kovaryansı Q ve ölçüm gürültü kovaryansı R matrisleri her ölçüm aralığında değişir fakat biz bunların sabit olduğunu varsayırız.

Doğrusal rassal fark denklemindeki $n \times n$ boyutundaki A matrisi, $k-1$ bir önceki durumdan, k durumuna olan durum geçiş matrisidir. Her k durum değişiminde A matrisinin değişmesi beklenir fakat pratikte sabit olduğu varsayılır. B matrisi, $n \times 1$ boyutunda kontrol girişi u , $u \in \mathbb{R}$ her k durumu ile ilgilidir. H matrisi, $m \times n$ boyutunda ölçüm denklemindeki k durumundaki ölçüm değerleri ile ilgilidir. Her k durum değişiminde değişmesi beklenir fakat pratikte sabit kabul edilir [43].

$\hat{x}_k^- \in \mathbb{R}^n$, k durumunda tahmin edilen priori/öngörü durum tahmini (süper minus olarakda adlandırılır) bilgisidir. $\hat{x}_k \in \mathbb{R}^n$ ifadesi, k durumundaki z_k ölçüm değeri ile tahminde bulunulan sonraki “posteriori” durum tahmini olarak adlandırılır. Priori ve posteriori öngörü/sonraki tahmin hataları aşağıdaki gibi tanımlanır:

$$\text{Öngörü (priori) tahmin hatası, } e_k^- = x_k - \hat{x}_k^- \quad (4.10)$$

$$\text{Sonraki (posteriori) tahmin hatası, } e_k = x_k - \hat{x}_k \quad (4.11)$$

$$\text{Öngörü (priori) tahmin hata kovaryansı, } P_k^- = E[e_k^- e_k^{-T}] \quad (4.12)$$

$$\text{Sonraki (posteriori) tahmin hata kovaryansı, } P_k = E[e_k e_k^T] \quad (4.13)$$

Kalman Filtresi denklemlerinden elde edilen, sonraki durum (posteriori) tahmin değerini, öngörü durum (priori) tahmin değeri ve güncel ölçüm değeri z_k ile ölçüm tahmini $H\hat{x}_k^-$ arasındaki farktan hesaplayan, Kalman Filtresinin olasılık orijinli denklemi aşağıda verilmiştir.

Kalman Filtresinin olasılık orijinli denklemi:

$$\hat{x}_k = \hat{x}_k^- + K(z_k - K\hat{x}_k^-) \quad (4.14)$$

Denklemdaki $(z_k - K\hat{x}_k^-)$ değeri, ölçüm tahmini “rezidü” olarak adlandırılır. Rezidü tahmin edilen ölçüm değeri ile ölçülen değer arasındaki farkı yansıtır. Rezidü sıfır ise iki değer birbirine eşittir.

Denklemdaki $n \times m$ boyutlu K matrisi, sonraki “posteriori” hata kovaryansını minimize eden değer, “Kalman Kazancı” olarak seçilir. Yukarıdaki denklemlerden elde edilen bu kazanç,

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad (4.15)$$

$$K_k = \frac{P_k^- H^T}{(H P_k^- H^T + R)} \quad (4.16)$$

denklemi ile hesaplanır. Denkleme baktığımızda, ölçüm hata kovaryansı R sıfıra yaklaşırken; kazanç K ,

$$\lim_{R \rightarrow 0} K_k = H^{-1} \quad (4.17)$$

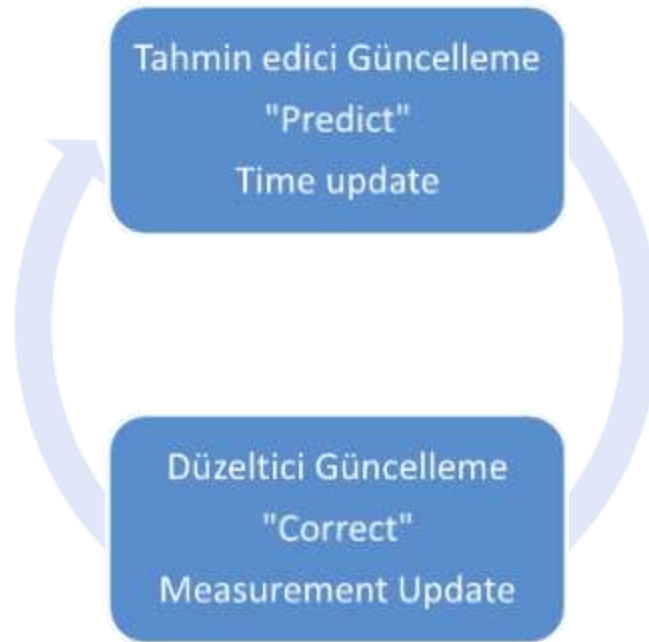
ile ifade edilir. Diğer bir taraftan, öngörü “priori” tahmin hata kovaryansı P_k^- sıfıra yaklaşırken; kazanç K ,

$$\lim_{P_k^- \rightarrow 0} K_k = 0 \quad (4.18)$$

ile ifade edilir [50].

3.5. Ayrık Kalman Filtresi Algoritması

Kalman Filtresi, geri besleme, yani sistemin gürültülü ölçüm değerlerinin kullanarak, bir sistemin durumlarını tahmin eder. Kalman Filtresi denklemleri iki grupta; zaman güncelleme denklemleri ve ölçüm güncelleme denklemleri olarak toplanabilir. Zaman güncelleme denklemleri, bir sonraki durumun değeri tahmin etmek için, güncel durumu ve hata kovaryansını kullanır. Zaman güncelleme denklemleri, tahmin edici denklemler olarak adlandırılır. Ölçüm güncelleme denklemleri ise, geliştirilmiş sonraki “improved posteriori” tahmini elde etmede kullanılır. Ölçüm güncelleme denklemleri ise doğrulayıcı, düzeltici denklemler olarak adlandırılır.



Şekil 3.3. Ayrık Kalman Filtresi döngüsü

Tahmin edici güncelleme denklemleri:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \quad (4.19)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (4.20)$$

Tahmin edici denklemler; k-1 durumundaki bilgileri kullanarak k durumundaki sistemin durum ve kovaryansını tahmin eder.

Düzeltilici güncelleme denklemleri:

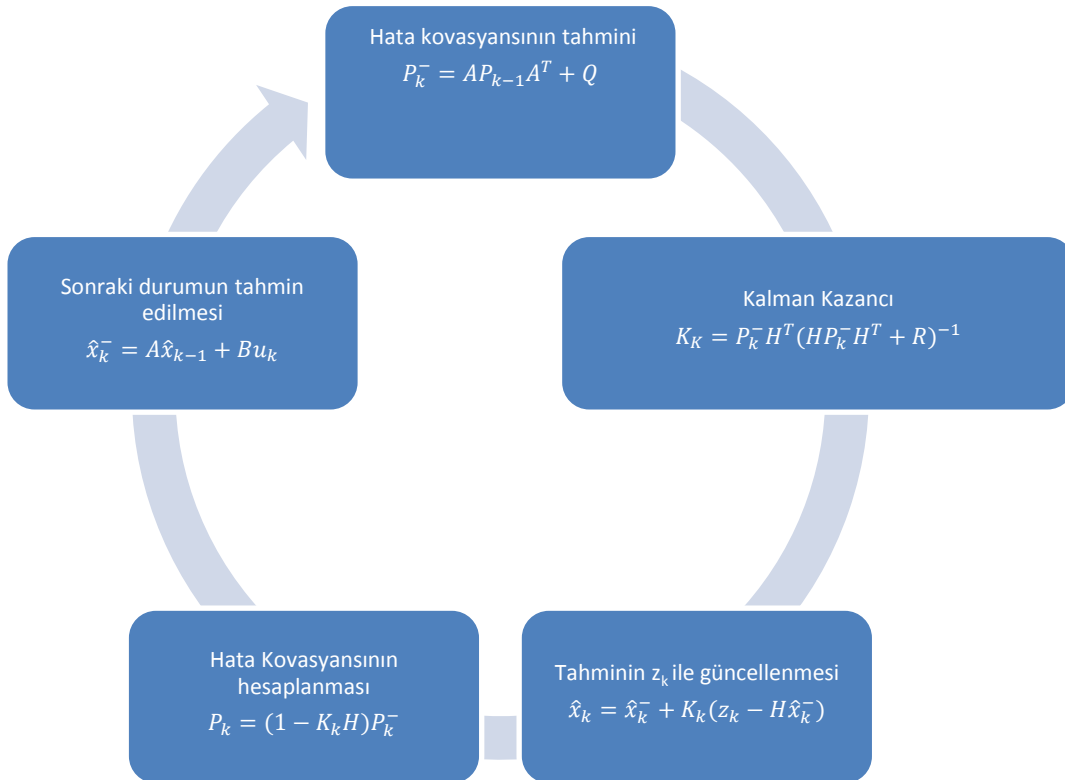
$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad (4.15)$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H \hat{x}_k^-) \quad (4.21)$$

$$P_k = (1 - K_k H) P_k^- \quad (4.22)$$

Düzeltilici güncelleme denklemlerinde ise ilk önce Kalman kazancı hesaplanır. Sonraki “posteriori” durum değeri z_k değeri ölçülerek, güncellenir. Sonraki adımda ise “posteriori” hata kovaryansı elde edilir.

Bu, tahmin edici ve düzeltilici döngü her tekrarlandığında, yani bir önceki değerler kullanılarak, sonraki değerler (tahminler) elde edilir [51].



Şekil 3.4. Kalman Filtresi çevrimi

BÖLÜM 4. UYGULAMA

4.1. Giriş

Günümüzde gerçek zamanlı sistemler ve uygulamalar, otobüs araba gibi hareketli araçlarda yaygın olarak kullanılmaktadır. Hareketli platformlar için gerçek zamanlı uygulamalar değerli ve gereklidir. Gerçek zamanlı sistemler hareketli platformlarla uyum sağlamış ve vazgeçilmez parçası olmuştur. Bu uygulama hareketli araçlarda TV uydularının Kalman Filtresi ile takibi üzerine giriş sağlamaktadır.

Hareketli araçlarda anten yardımıyla uydu işaretlerini takip edilmesi ile TV izlemek mümkün olmaktadır. Anten hareket ederken, uydu işaretlerinin alınması için gerçek zamanlı takip gereklidir. Uydunun pozisyon bilgisi ve hareketli aracın pozisyon ve konum bilgisi gerçek zamanlı TV izlemek için gerekli matematiksel hesaplar yapmak için gereklidir. Hareketli aracın konum ve pozisyon bilgisi hareketten dolayı sürekli olarak değişmektedir. Bundan dolayı, takip edilen uydunun işaretlerinin anten tarafından alınması için antenin pozisyonu sürekli olarak güncellenmelidir.

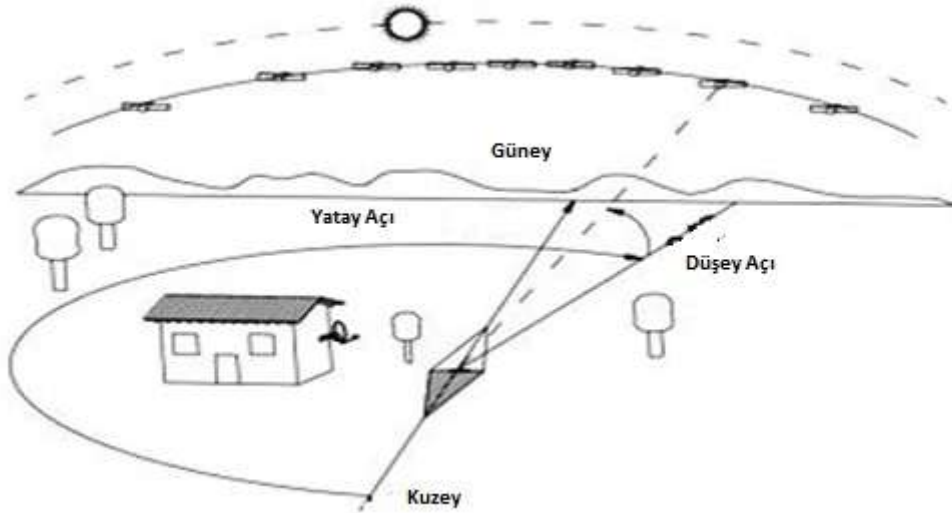
Hareketli aracın pozisyon ve konum bilgisi yaygın olarak kullanılan GPS, ivmeölçer ve magnetometre algılayıcıları yardımıyla elde edilir. Hareketli aracın konum bilgisi ve pozisyon bilgisi sürekli olarak bu algılayıcılar tarafından uygulamaya sağlanmalıdır. Bu veriler uygulama tarafından hareketli aracın anteninin yükseklik ve yatay açı bilgilerinin hesaplanmasında kullanılmaktadır. Bu açılar izleme için Kalman Filtresi tarafından kullanılmaktadır [52].

Bulunanan pozisyon ve konum bilgilerine göre antenin hesaplanan yatay açı ve dikey açı fark bilgisi Kalman Filtresi geçirilerek, uydu işaretlerini takibi

sağlanmıştır. Uygulamada 42° Doğuda bulunan TÜRKSAT 4A uydusu için hesaplamalar yapılmıştır.

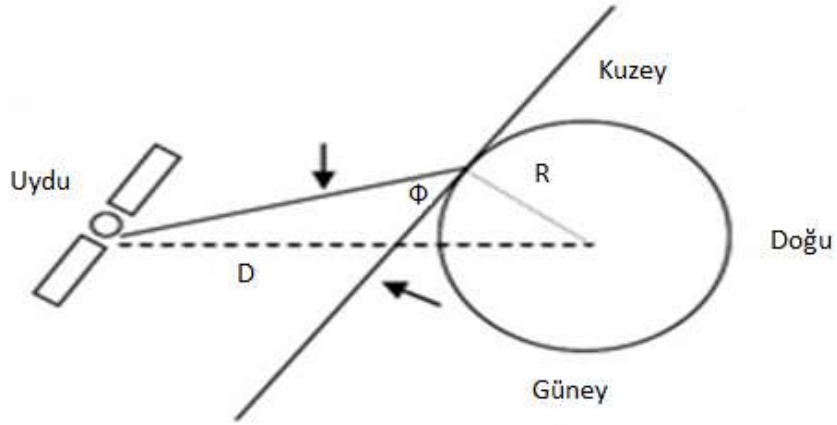
4.2. Hareketli Aracın Konum ve Pozisyon Bilgisi

Uydunun işaretlerinin bir anten yardımıyla izlememesi için hareketli aracın pozisyon ve konum bilgisine ihtiyaç duyulmaktadır. Hareketli aracın Dünya üzerindeki konum bilgisi GPS algılayıcısı tarafından sağlanmaktadır. Hareketli aracın pozisyon bilgisi ise ivmeölçer ve magnetometre algılayıcısı tarafından sağlanmaktadır. Bütün bu veriler izlenen uydu için antenin yatay ve düşey açı parametrelerinin hesaplanmasından kullanılmaktadır.



Şekil 4.1. Bulunulan bölgenin yatay ve düşey açısı [53]

Hareketli aracın konum bilgisi, aracın hızına bağlı olarak değişme göstermektedir. Bu değişme çok hızlı olan bir değişme değildir. Çünkü uydu ile hareketli araç arasındaki mesafe çok fazladır. Konum bilgisi, sürekli olarak geliştirilen uygulamaya GPS algılayıcısı tarafından sağlanmaktadır. Hareketli aracın bulunduğu enlem ve boylam bilgilerinden o konuma ait uydu anteninin yatay ve düşey açıları hesaplanmaktadır. Bu işlem sürekli olarak veya belirli aralıklarla yapılmalıdır. Bulunulan bölgenin yatay ve düşey açıları aşağıdaki formüllere göre hesaplanmalıdır [54].



Şekil 4.2. Düşey açı [55]

Antenin düşey açısının hesaplanması:

$$R = \text{Dünyanın Yarıçapı} \quad (5.1)$$

$$D = \text{Dünya Merkezi ile Uydu Arasındaki Mesafe} \quad (5.2)$$

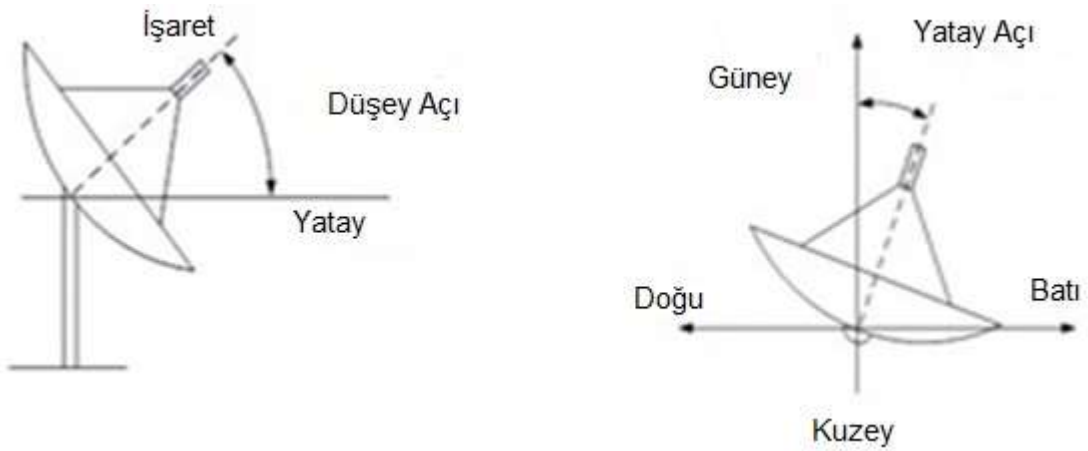
$$N = \text{Uydu Boylamı} - \text{Anten Boylamı} \quad (5.3)$$

$$B = \cos^{-1}(\cos(\text{Anten Boylamı}) \cdot \cos N) \quad (5.4)$$

$$P = \frac{R}{D} \quad (3) \quad (5.5)$$

$$\phi = \text{Düşey Açı}$$

$$\phi = \tan^{-1}\left(\frac{\cos B - P}{\sin B}\right) \quad (4) \quad (5.6)$$



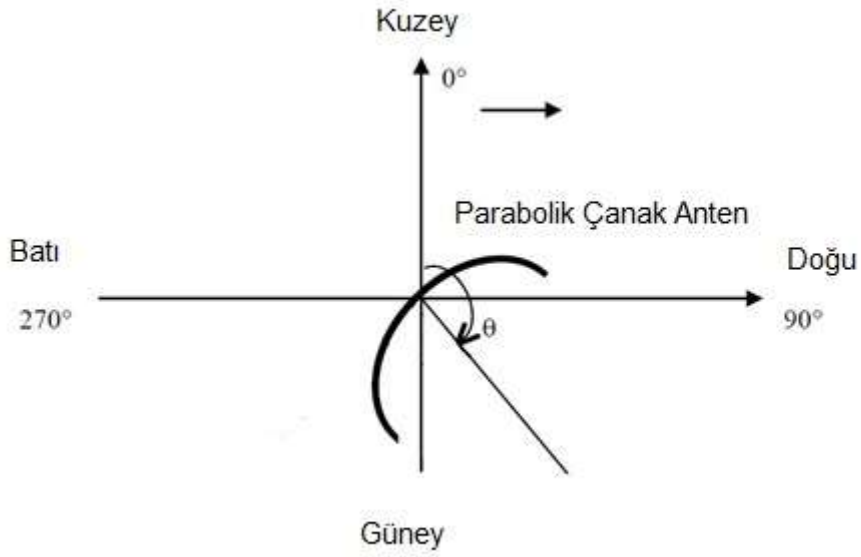
Şekil 4.3. Yatay ve düşey açı [56]

Antenin yatay açısının hesaplanması:

$$\vartheta = \text{Yatay Açı} \quad (5.7)$$

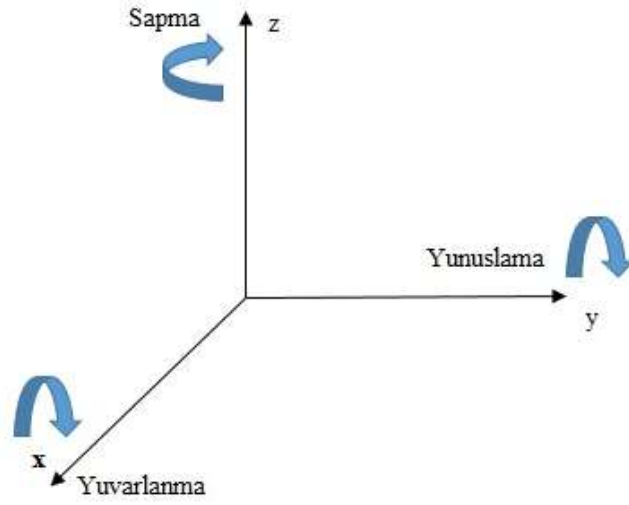
$$N = \text{Uydu Boylamı} - \text{Anten Boylamı} \quad (5.8)$$

$$\vartheta = \tan^{-1}\left(\frac{\tan N}{\sin(\text{Anten Enlemi})}\right) \quad (5.9)$$



Şekil 4.4. Yatay açı [57]

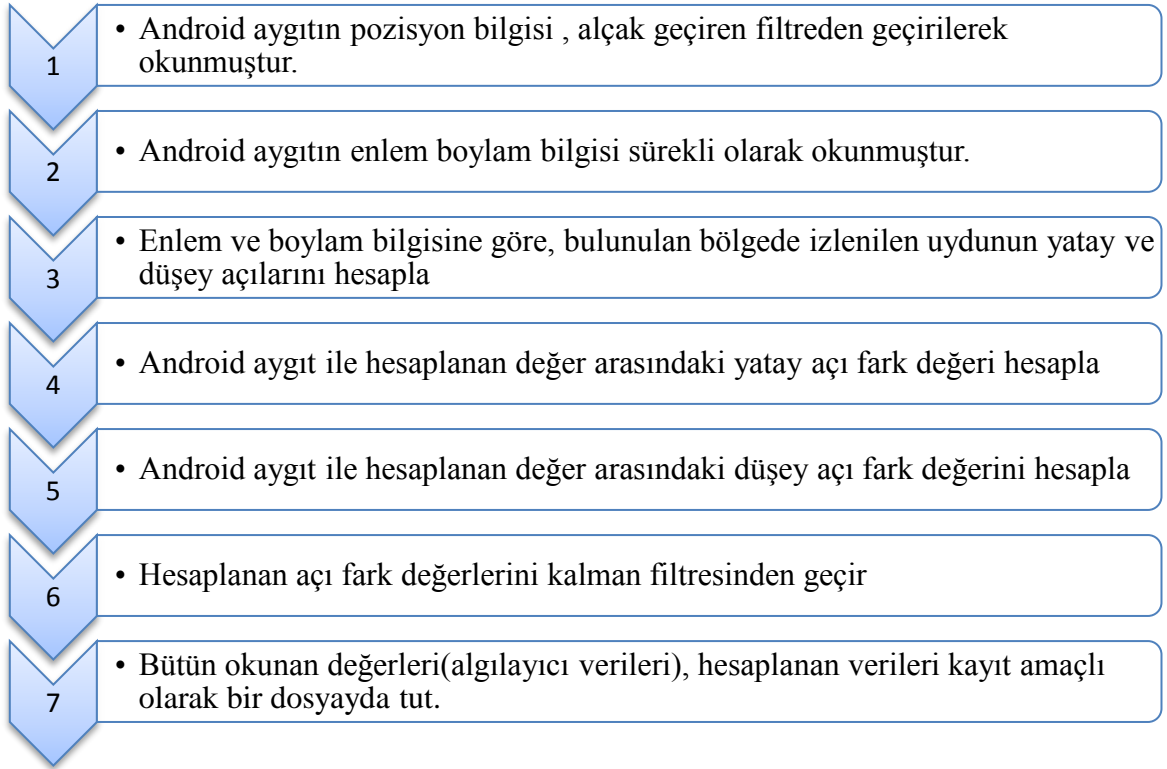
Hareketli aracın pozisyon bilgisi, sürekli olarak ivmeölçer ve magnetometre algılayıcıları tarafından alınır. Bu algılayıcılardan gelen veriler harekete uygun olarak iyileştirilmelidir. İyileştirme amaçlı olarak alçak geçiren filtre, yürüyen ortalama filtresi gibi filtreler kullanılabilir ayrıca algılayıcılar gürültü ve etkileşim ve girişime karşı iyi izole edilmelidir.



Şekil 4.5. Hareketli araç pozisyon bilgisi

4.3. Algoritma

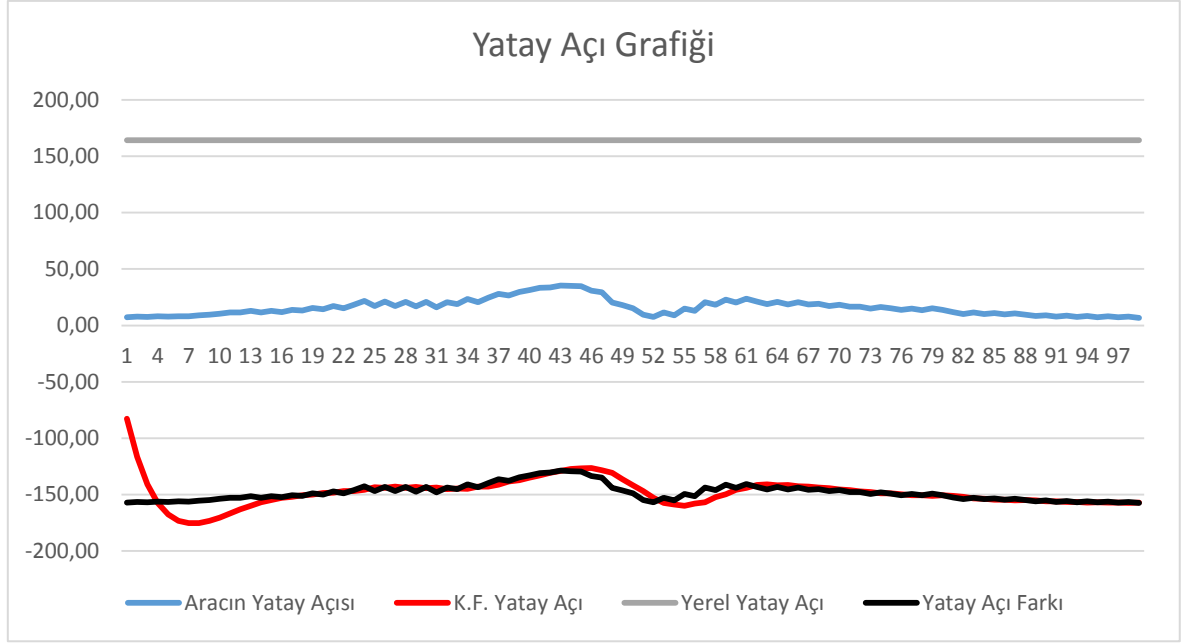
Geliştirilen gerçek zamanlı uygulama bir otobüste Bolu, Yalova şehirleri arasında test edildi. Geliştirilen uygulama bir android tabanlı aygıtın (burada hareket eden araç olarak düşünüldü) yatay ve düşey açılarını ve konum bilgisini toplayarak hesaplamalarda kullandık. Android tabanlı aygıt, burada hareket eden aracı temsil etmektedir. Uygulamada, pozisyon ve konum bilgileri gerçek zamanlı sürekli toplanarak aygıtta bir dosyada depolanmaktadır. Bütün toplanan veriler parabolic bir antenin, takip edilen uydunun işaretinin alınması için yatay ve düşey açılarının hesaplanmasında kullanıldı. Android aygıtın yatay ve düşey açıları ile uydunun izlenebilmesi için gereken yatay ve düşey açıların arasındaki fark sürekli olarak takip için kalman filtresinden geçirilmiştir. Kalman filtresi hesaplamaları için açık kaynak kodlu bir kod parçacığı kullanılmıştır. Uygulamanın akış diyagramını Şekil 4.6 gösterilmektedir.



Şekil 4.6. Algoritma akışı

Tablo 4.1. Yatay açı test verileri

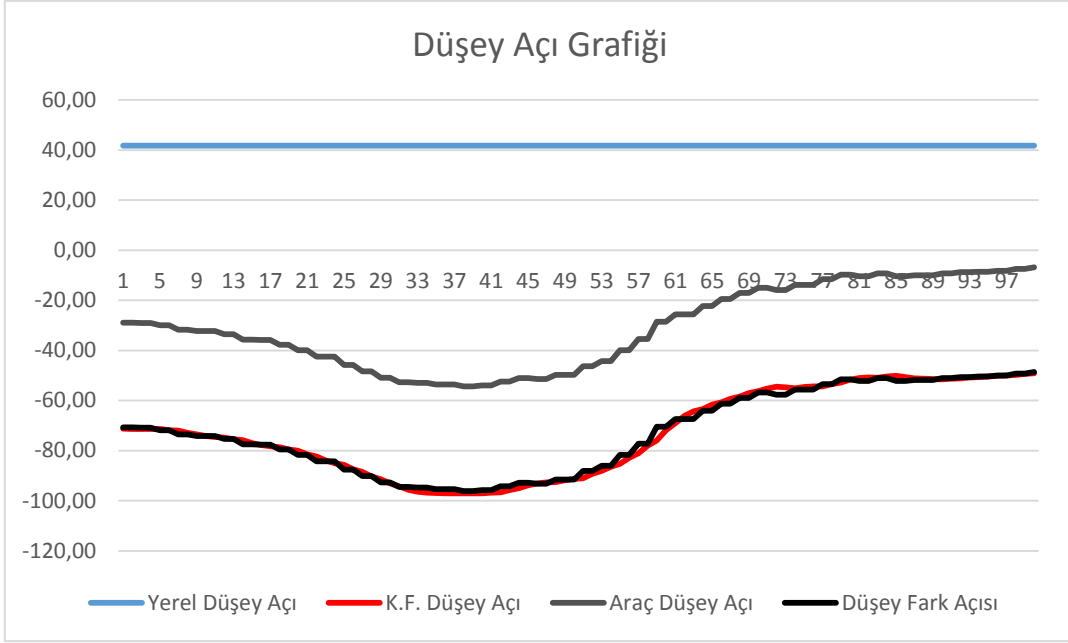
A	B	C	D	E
Örnek No:	Zaman	Uydunun Yatay Açısı	Aracın Yatay Açısı	K.F. Hesaplanan Yatay Açı
1661	100647	164.3°	280.6°	116.1°
4936	104507	163.7°	331.2°	167.5°
5914	105521	163.4°	294.5°	131.2°



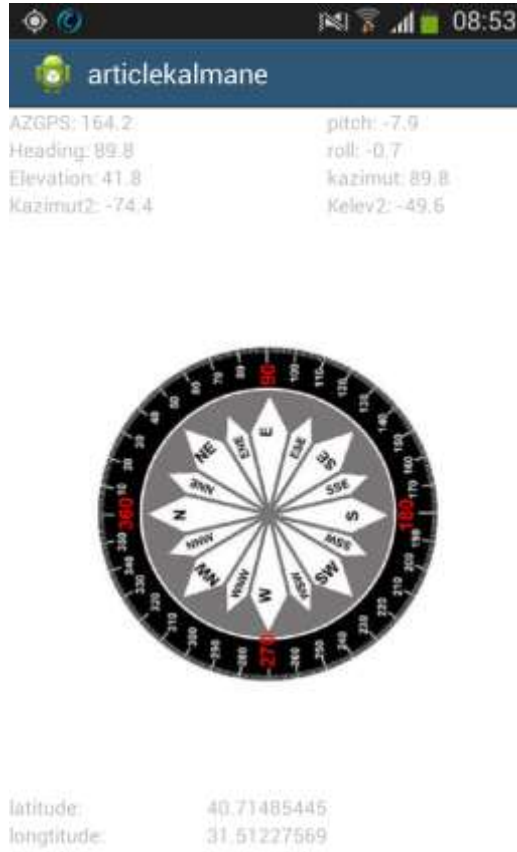
Şekil 4.7. Yatay açı grafiği

Tablo 4.2. Düşey açı test verileri

A	B	C	D	E
Örnek No:	Zaman	Uydunun Düşey Açısı	Aracın Düşey Açısı	K.F. Hesaplanan Düşey Açı
1661	100647	41.8°	0.9°	40.9°
4936	104507	41.6°	2.7°	38.8°
5914	105521	41.5°	-0.9°	42.5°



Şekil 4.8. Düşey açı grafiği



Şekil 4.9. Uygulama ekran görüntüsü

BÖLÜM 5. SONUÇLAR VE ÖNERİLER

Bu çalışmada gerçek zamanlı çalışan bir uygulama geliştirilerek, gerçek veriler üzerinde çalışılmıştır. Geliştirilen uygulama ile hareket eden bir araç için parabolik çanak antenin yatay ve düşey açılarının gerçek zamanlı tahminleri Kalman Filtresi yardımıyla yapılmıştır. Gerçek zamanlı uygulama Android bir cihaz üzerinde çalıştırılarak testler hareketli bir araçta yapılmıştır. Yapılan çalışma sonuçlarını incelediğimizde hareketli aracın hızına, yani konum değişiminin yavaş olmasına bağlı olarak anten parametrelerinin çok hızlı değişmediği gözlenmiştir. Hareketli araçlarda uydu izlemede en etkili parametrenin aracın yön ve pozisyon değişimlerinin etkili olduğu gözlenmiştir. Ayrıca hareket eden araçta oluşan sarsıntıların algılayıcılardaki veri okuma süreçlerini olumsuz etkilediği gözlenmiştir.

Bu çalışma sayesinde, gerçek zamanlı bir ortamda testler yapılarak, gerçek zamanlı bir sistem tecrübe edilmiştir.

Gelecekte bu konu üzerinde yapılabilecek çalışmalar üç grupta toplanabilir. İlk grupta İşaret iyileştirme konusu incelenebilir. Algılayıcılar üzerinden alınan verilerin daha sağlıklı okunabilmesi ile birlikte gürültüden ve bozucu etkilerden arındırılarak gerçeğe yakın verilerin işlenmesi sağlanabilir. İkinci grupta yapılacak çalışmalar, elde edilen veriler kullanılarak hareketli araçlarda uydu takibi üzerine yeni izleme algoritmaları üzerinde ve var olan izleme algoritmaları ile kıyaslanarak performans analizi yapılabilir. Üçüncü grupta bu konu üzerinde anten dizileri yardımıyla uydu takibi üzerinde çalışmalar yapılabilir. Ayrıca alınan uydu sinyalinin seviyesinin ve kalitesinin artırılması üzerine çalışmalar yapılabilir.

KAYNAKLAR

- [1] THE STAFF OF LAB-VOLT LTD. , Satellite Orbits, Covarage, And Antenna Alignment, Canada, pp. 7, 2011.
- [2] http://en.wikipedia.org/wiki/Sputnik_1, Eriřim Tarihi: 15.03.2015.
- [3] http://tr.wikipedia.org/wiki/Yapay_uydu, Eriřim Tarihi: 15.03.2015.
- [4] THE STAFF OF LAB-VOLT LTD., Principles of Satellite Communications, Canada, pp. 29, 2014.
- [5] International Telecommunication Union, Handbook on Satellite Communications, pp.3-10 April, 2002.
- [6] http://www.mu.ac.in/myweb_test/Satelight%20Comm..pdf, Eriřim Tarihi: 31.03.2015.
- [7] https://www.labvolt.com/downloads/86311_F0.pdf, Eriřim Tarihi: 15.03.2015.
- [8] http://rewreward.blogspot.com/2014_02_01_archive.html, Eriřim Tarihi: 15.03.2015.
- [9] http://www.mu.ac.in/myweb_test/Satelight%20Comm..pdf, Eriřim Tarihi: 31.03.2015.
- [10] http://www.mu.ac.in/myweb_test/Satelight%20Comm..pdf, Eriřim Tarihi: 31.03.2015.
- [11] RODDY DENNIS, Satellite Communications, pp.21-27. 2001.
- [12] http://www.mu.ac.in/myweb_test/Satelight%20Comm..pdf, Eriřim Tarihi: 31.03.2015.
- [13] http://www.european-security.com/n_index.php?id=5822, Eriřim Tarihi: 15.03.2015.
- [14] http://en.wikipedia.org/wiki/Low_Earth_orbit, Eriřim Tarihi: 01.04.2015.
- [15] http://en.wikipedia.org/wiki/Medium_Earth_orbit, Eriřim Tarihi: 01.04.2015.

- [16] http://www.cse.wustl.edu/~jain/cis788-97/ftp/satellite_data/index.html, Erişim Tarihi: 15.03.2015.
- [17] http://en.wikipedia.org/wiki/High_Earth_orbit, Erişim Tarihi: 01.04.2015.
- [18] <http://www.cpi.com/capabilities/ssa.html>, Erişim Tarihi: 15.03.2015.
- [19] http://www.mu.ac.in/myweb_test/Satelight%20Comm..pdf, Erişim Tarihi: 31.03.2015.
- [20] <http://www.avkomsat.com/#!/kabandi/cfvg>, Erişim Tarihi: 15.03.2015.
- [21] http://www.esa.int/Our_Activities/Telecommunications_Integrated_Applications/Satellite_frequency_bands, Erişim tarihi: 21.03.2015.
- [22] <http://www.antenna-theory.com/antennas/aperture/horn.php>, Erişim tarihi: 31.03.2015.
- [23] [http://akademik.maltepe.edu.tr/~demironer/ELK%20412%20Telsiz%20ve%20Mobil%20A%F0lar/Ders%20Notlar%FD/Bolum%202-Antenler%20ve%20Yay%FDlma%20\(23.02.2015\).pdf](http://akademik.maltepe.edu.tr/~demironer/ELK%20412%20Telsiz%20ve%20Mobil%20A%F0lar/Ders%20Notlar%FD/Bolum%202-Antenler%20ve%20Yay%FDlma%20(23.02.2015).pdf), Erişim Tarihi: 15.03.2015.
- [24] <http://www.radartutorial.eu/06.antennas/Phased%20Array%20Antenna.en.html>, Erişim tarihi 31.03.2015.
- [25] <http://www.radartutorial.eu/06.antennas/Phased%20Array%20Antenna.en.html>, Erişim tarihi 31.03.2015.
- [26] <http://www.radartutorial.eu/06.antennas/Phased%20Array%20Antenna.en.html>, Erişim tarihi 31.03.2015.
- [27] <http://akademik.maltepe.edu.tr/~demironer/ELK%20412%20Telsiz%20ve%20Mobil%20A%F0lar/Ders%20Notlar%FD/Bolum%202-Antenler%20ve%20Yay%FDlma%20%2823.02.2015%29.pdf>, Erişim Tarihi: 31.03.2015.
- [28] <http://www.radartutorial.eu/06.antennas/pic/parabol1.print.png>, Erişim Tarihi: 15.03.2015.
- [29] <http://www.antenna-theory.com/antennas/reflectors/dish.php>, Erişim tarihi: 31.03.2015.
- [30] <http://tr.wikipedia.org/wiki/GPS>, Erişim Tarihi: 31.03.2015.
- [31] <http://www.navigasyon.com.tr/gps.html>, Erişim Tarihi: 15.03.2015.
- [32] http://en.wikipedia.org/wiki/Axes_conventions, Erişim Tarihi: 31.03.2015.

- [33] MILETTE G., STROUD A., Professional Android Sensor Programming, pp 103, 2012.
- [34] <http://www.sophia.org/tutorials/accuracy-and-precision--3>, Eriřim Tarihi: 01.04.2015.
- [35] MILETTE G., STROUD A., Professional Android Sensor Programming, pp 105-107, 2012.
- [36] MILETTE G., STROUD A., Professional Android Sensor Programming, pp 104, 2012.
- [37] <http://nptel.ac.in/courses/112103174/pdf/mod2.pdf>, Eriřim Tarihi: 15.03.2015.
- [38] <http://nptel.ac.in/courses/112103174/pdf/mod2.pdf>, Eriřim Tarihi: 15.03.2015.
- [39] http://en.wikipedia.org/wiki/Moving_average , Eriřim Tarihi: 31.03.2015.
- [40] <http://nptel.ac.in/courses/112103174/pdf/mod2.pdf>, Eriřim Tarihi: 15.03.2015.
- [41] <http://nptel.ac.in/courses/112103174/pdf/mod2.pdf>, Eriřim Tarihi: 15.03.2015.
- [42] DEMİRKOL Ařkın, Sinyaller&Sistemler, Sistem Transformasyonu, pp.561, İstanbul, Eylül 2013.
- [43] DEMİRKOL Ařkın, Sinyaller&Sistemler, Sistem Transformasyonu, pp.561, İstanbul, Eylül 2013.
- [44] https://kasnakoglu.files.wordpress.com/2014/01/ders7_ck01.pdf, Eriřim Tarihi: 31.03.2015.
- [45] https://kasnakoglu.files.wordpress.com/2014/01/ders7_ck01.pdf, Eriřim Tarihi: 31.03.2015.
- [46] <http://80.251.40.59/science.ankara.edu.tr/ozturk/Dersler/ist522/Tercumeler/EKFMehmet.pdf>, Eriřim Tarihi: 31.03.2015.
- [47] <http://archimede.bibl.ulaval.ca/archimede/fichiers/25229/api.html>, Eriřim Tarihi: 15.03.2015.
- [48] DEMİRKOL Ařkın, Sinyaller&Sistemler, Sistem Transformasyonu, pp.562, İstanbul, Eylül 2013.
- [49] ÇAYIROĐLU İbrahim, Fen ve Teknoloji Bilgi Paylařımı, Kalman Filtresi ve Programlama, Karabük, 2012-1.

- [50] WELCH G. BISHOP G., “SIGGRAPH 2001 Course 8 An Introduction to the Kalman Filter”,pp.19-20, 2001, <http://www.cs.unc.edu/~{welch, gb}>, Eriřim Tarihi:31.03.2015.
- [51] WELCH G. BISHOP G, “An Introduction to the Kalman Filter”,pp. 5-7 2006, <http://www.cs.unc.edu/~welch/kalman>, Eriřim Tarihi: 31.03.2015.
- [52] KOCADAĐ F.,DEMİRKOL A., 2nd International Conference On Signal Processing and Integrated Networks, Real Time Tracking of TV Satellites on Moving Vehicles Using Kalman Filter, February 19-20, New DELHI, 2015.
- [53] http://www.techtir.ie/howto/align_sat_dish, Eriřim Tarihi: 15.03.2015.
- [54] GÖZDE H., “Design Of A Two Dimensional Microprocessor Based Parabolic Antenna Controller,” Gazi University Institute Of Science And Technology M. Sc. Thesis, pp.8-11, August 2003.
- [55] GÖZDE H., “Design Of A Two Dimensional Microprocessor Based Parabolic Antenna Controller,” Gazi University Institute Of Science And Technology M. Sc. Thesis, pp.8, August 2003.
- [56] http://www.techtir.ie/howto/align_sat_dish, Eriřim Tarihi: 15.03.2015.
- [57] GÖZDE H., “Design Of A Two Dimensional Microprocessor Based Parabolic Antenna Controller,” Gazi University Institute Of Science And Technology M. Sc. Thesis, pp.8, August 2003.

EKLER

EK A: Uygulamanın Android java kodu

```
package com.example.vaio.articlekalman;

import android.app.Activity;
import android.content.Context;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.animation.Animation;
import android.view.animation.RotateAnimation;
import android.widget.ImageView;
import android.widget.TextView;
import java.io.BufferedWriter;
```

```
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Random;
import java.util.logging.FileHandler;

public class MainActivity extends Activity implements
SensorEventListener, LocationListener
{
    public static final String FILENAME = "artickalmane.txt";
    ///file operations

    SimpleDateFormat sdf = new SimpleDateFormat("yyyyMMdd_HHmms");
    String currentDateandTime = "";

    protected LocationManager locationManager;
    protected LocationListener locationListener;
    protected Context context;

    Location location;

    @Override
    public void onProviderDisabled(String provider) {
        Log.d("Latitude", "disable");
    }

    String data="";
```

```

GPSTracker gps;
// define the display assembly compass picture
private ImageView image;
// record the compass picture angle turned
private float currentDegree = 0f;
float azimuth;
float pitch;
float roll;// View to draw a compass
float degree=0f;
float k2azimut=0f;
float k2elevation=0f;
////////////////////kalman variables////////////////////

float zk=0f;
float xkex=0f;
float xk=0f;
float pk=1;
float pkex=0f;
float r= (float) 0.1;
float kk=0;

////////////////////kal2////////////////////

JKalman kalman = new JKalman(4, 2);

Random rand = new Random(System.currentTimeMillis() % 2011);
double x = 0;
double y = 0;
// constant velocity
double dx = rand.nextDouble();

```

```

double dy = rand.nextDouble();

// init
Matrix s = new Matrix(4, 1); // state [x, y, dx, dy, dxy]
Matrix c = new Matrix(4, 1); // corrected state [x, y, dx, dy, dxy]

Matrix m = new Matrix(2, 1); // measurement [x]

// transitions for x, y, dx, dy
double[][] tr = { {1, 0, 1, 0},
                  {0, 1, 0, 1},
                  {0, 0, 1, 0},
                  {0, 0, 0, 1} };

////////////////////////////////////

// device sensor manager
Sensor accelerometer;
Sensor magnetometer;

TextView tvHeading;
public static TextView textlat;
public static TextView textlong;
public static TextView kalmanpredict;

double azgps=0d;
double elevation=0d;

double latitude=0d;

```

```
double longitude=0d;
double b=0d;

public static TextView textazgps;
public static TextView textelevation;
public static TextView textpitch;
public static TextView textroll;
public static TextView kalman1;
public static TextView kalmanazimut;
public static TextView kalmanelevation;

private SensorManager mSensorManager;

// Acquire a reference to the system Location Manager

// Define a listener that responds to location updates

static final float ALPHA = 0.15f;

protected float[] lowPass( float[] input, float[] output ) {
    if ( output == null ) return input;

    for ( int i=0; i<input.length; i++ ) {
        output[i] = output[i] + ALPHA * (input[i] - output[i]);
    }
    return output;
}

LocationManager locMgr;
```

```
MyLocationListener locLstnr;
```

```
private void writeToFile( String mytext)
```

```
{
```

```
    Log.i("TEST", "SAVE");
```

```
    try {
```

```
        String fpath = "/storage/emulated/0/"+"sonuclar"+" .txt";
```

```
        File file = new File(fpath);
```

```
        // If file does not exists, then create it
```

```
        if (!file.exists()) {
```

```
            file.createNewFile();
```

```
        }
```

```
        FileWriter fw = new FileWriter(file.getAbsolutePath(),true);
```

```
        BufferedWriter bw = new BufferedWriter(fw);
```

```
        bw.write(mytext);
```

```
        bw.close();
```

```
        Log.d("Sucess", "Sucess");
```

```
    } catch (IOException e) {
```

```
        e.printStackTrace();
```

```
    }
```

```
}
```

```
@Override
```

```
public void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.comm);
```

```
    try {
```

```
        FileHandler myFileHandler = new FileHandler();
```

```

    } catch (IOException e) {
        e.printStackTrace();
    }

    locationManager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);

locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 200,
1, this);

    gps=new GPSTracker(MainActivity.this);

    mSensorManager =
(SensorManager)getSystemService(SENSOR_SERVICE);
    accelerometer =
mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
    magnetometer =
mSensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);

    image = (ImageView) findViewById(R.id.imageViewCompass);
    // TextView that will tell the user what degree is he heading
    tvHeading = (TextView) findViewById(R.id.tvHeading);
    textlat = (TextView) findViewById(R.id.textlat);
    textlong = (TextView) findViewById(R.id.textlong);
    kalmanpredict= (TextView) findViewById(R.id.latitude);

    textazgps=(TextView) findViewById(R.id.textViewazgps);
    textelevation=(TextView) findViewById(R.id.textViewelevation);
    textpitch=(TextView) findViewById(R.id.textViewpitch);
    textroll=(TextView) findViewById(R.id.textViewroll);
    kalman1=(TextView) findViewById(R.id.textViewkalman);
    kalmanazimut=(TextView) findViewById(R.id.textViewkalmanazi);

```



```
kalmanelevation=(TextView) findViewById(R.id.textViewkalmanelevation);

// kalmanpredict.setText("Başlıyor");

// initialize your android device sensor capabilities
// mSensorManager = (SensorManager)
getSystemService(SENSOR_SERVICE);

String locationProvider = LocationManager.GPS_PROVIDER;

m.set(0, 0, x);
m.set(1, 0, y);

kalman.setTransition_matrix(new Matrix(tr));

// 1s somewhere?
kalman.setError_cov_post(kalman.getError_cov_post().identity());

gps = new GPSTracker(MainActivity.this);

locMgr =
(LocationManager)getSystemService(Context.LOCATION_SERVICE);
locLstnr = new MyLocationListener();
}

private void makeUseOfNewLocation(Location location) {
}

@Override
```

```
protected void onResume() {
    super.onResume();

    // for the system's orientation sensor registered listeners

    mSensorManager.registerListener(this, accelerometer,
    SensorManager.SENSOR_DELAY_UI);

    mSensorManager.registerListener(this, magnetometer,
    SensorManager.SENSOR_DELAY_UI);

}

@Override
protected void onPause() {
    super.onPause();

    mSensorManager.unregisterListener(this);
}

float[] mGravity;
float[] mGeomagnetic;

@Override
public void onLocationChanged(Location location) {

    latitude = location.getLatitude();
    longitude = location.getLongitude();
    // latitude = gps.getLatitude();
    // longitude = gps.getLongitude();

    Log.d("Success", "on Location changed Success");
```

```

        textlat.setText( Double.toString(latitude));
        textlong.setText( Double.toString(longitude));
    }

    @Override
    public void onStatusChanged(String provider, int status, Bundle extras) {

    }

    @Override
    public void onProviderEnabled(String provider) {

    }

    public void onSensorChanged(SensorEvent event) {

        locMgr.requestLocationUpdates(LocationManager.GPS_PROVIDER, 0, 0,
this);

        textlat.setText( Double.toString(latitude));
        textlong.setText( Double.toString(longitude));

        if (event.sensor.getType() == Sensor.TYPE_ACCELEROMETER)
            mGravity = lowPass( event.values.clone(),mGravity);

        if (event.sensor.getType() == Sensor.TYPE_MAGNETIC_FIELD)
            mGeomagnetic = lowPass( event.values.clone(),mGeomagnetic);
        if (mGravity != null && mGeomagnetic != null) {
            float R[] = new float[9];

```

```
float I[] = new float[9];
boolean success = SensorManager.getRotationMatrix(R, I, mGravity,
mGeomagnetic);
if (success) {
    float orientation[] = new float[3];
    SensorManager.getOrientation(R, orientation);
    azimuth =orientation[0];
    pitch=orientation[1];
    roll=orientation[2];

    azimuth=azimut*360/(2*3.14159f);
    // degree=Math.round(azimut);

    if(azimut<0)
    {
        azimuth=azimut+360;
    }

    //azimut=(float)Math.toDegrees(azimut);
    pitch=(float)Math.toDegrees(pitch);
    roll=(float)Math.toDegrees(roll);

    azimuth=azimut*10;
    pitch=pitch*10;
    roll=roll*10;

    azimuth=Math.round(azimut);
    pitch=Math.round(pitch);
    roll=Math.round(roll);
```

```

azimut=azimut/10;
pitch=pitch/10;
roll=roll/10;

degree=azimut*10;
degree=degree/10;

degree=azimut;

////////////////////////////////////
//calculation of gps azimut

// N

double p=0.15;

b=Math.toDegrees(Math.acos((Math.cos(Math.toRadians(latitude))*Math.cos(Math.toRadians(42-longitude)))); //indegrees

elevation= Math.toDegrees((Math.atan((Math.cos(Math.toRadians(b)) - p) / Math.sin(Math.toRadians(b)))));

elevation=elevation*10;
elevation=Math.round(elevation);
elevation=elevation/10;

azgps= Math.toDegrees(Math.atan(Math.tan(Math.toRadians(42-longitude))/(Math.sin(Math.toRadians(latitude)))));

azgps=180-azgps;//şimdilk dogru

```

```

    azgps=azgps*10;
    azgps=Math.round(azgps);
    azgps=azgps/10;

    textazgps.setText("AZGPS: " + Double.toString(azgps) );
    textelevation.setText("Elevation: " + Double.toString(elevation) );

}

//////////////////////////////////Kalman//////////////////////////////////

//update//
xkex=azimut;
xk=xkex;
pk=pkex;
zk=azimut;

///measurement update////

kk=pk/(pk+r);
xk=xkex+kk*(zk-xk);
pk=(1-kk)*pk;

//////////////////////////////////

xkex=xk;
pkex=pk;

//////////////////////////////////kalman 2//////////////////////////////////

```

```

s = kalman.Predict();

// function init :)
// m.set(1, 0, rand.nextDouble());
x =azimut-azgps;
y = pitch-elevation;

m.set(0, 0,x);
m.set(1, 0,y);

c = kalman.Correct(m);

////////////////////////////////////////kalman2////////////////////////////////////////
k2azimut=((float)s.get(0,0))*10;
k2azimut=(Math.round(k2azimut));k2azimut=k2azimut/10;
k2elevation=((float)s.get(1,0))*10;
k2elevation=(Math.round(k2elevation));k2elevation=k2elevation/10;

////////////////////////////////////////

tvHeading.setText("Heading: " + Float.toString(degree));
textpitch.setText("pitch: " + Float.toString(pitch));
textroll.setText("roll: " + Float.toString(roll) );

kalman1.setText("kazimut: " + Float.toString(xk));

kalmanazimut.setText("Kazimut2: " +Float.toString(k2azimut));
kalmanelevation.setText("Kelev2: " +Float.toString(k2elevation));

```

```

// create a rotation animation (reverse turn degree degrees)
RotateAnimation ra = new RotateAnimation(
    currentDegree,
    -xk,
    Animation.RELATIVE_TO_SELF, 0.5f,
    Animation.RELATIVE_TO_SELF,
    0.5f);

// how long the animation will take place
ra.setDuration(210);

// set the animation after the end of the reservation status
ra.setFillAfter(true);

// Start the animation
image.startAnimation(ra);
currentDegree = -xk;

currentDateandTime =sdf.format(new Date());

data=currentDateandTime+";"+"Latitude:"+Double.toString(latitude)+" "+"Longtitude:"+Double.toString(longitude)+" "+"azimut:"+azimut+" "+"elevation:"+elevation+" "+"kalman2azimut:"+Float.toString(k2azimut)+" "+"kalman2elevation:"+Float.toString(k2elevation)+" "+"kalman1 denoise:"+xk+" "+"pitch:"+pitch+" "+"roll:"+roll+" "+"gpsazimuthesap:"+azgps+" "+"gpselevation:"+elevation+"\n";

writeToFile(data);

// degree = azimut;// orientation contains: azimut, pitch and roll

}

```



```
// mCustomDrawableView.invalidate();

//Math.round(event.values[0]);

}
@Override
public void onAccuracyChanged(Sensor sensor, int accuracy) {
    // not in use
}

}
```

ÖZGEÇMİŞ

Fatih KOCADAĞ, 21.02.1979'da Bolu'da doğdu. İlk, orta öğrenimini Bolu ilindeki okullarda, lise eğitimini İstanbul Kuleli Askeri Lisesi'nde tamamladı. 2002 yılında başladığı Karadeniz Teknik Üniversitesi, Elektrik Elektronik Mühendisliği bölümü eğitimine devam ederken, 2003 yılında aynı okulda Bilgisayar Mühendisliği bölümünde çift anadal programına başladı. 2007 yılında Elektrik Elektronik Mühendisliği, 2009 yılında ise Bilgisayar Mühendisliği'nden mezun oldu. 2007-2009 yılları arasında Schott Orim Cam Tic. Sanayi A.Ş. Üretim Geliştirme Mühendisi olarak çalıştı. Bu yıllar arasında üretim bantlarının geliştirilmesi, makine ve süreç iyileştirmelerinin yapılması ve Yalın Üretim Teknikleri üzerine çalışmalarda bulundu, Çek Cumhuriyeti'nde kurulan yeni cam işleme tesisinin kurulumunda görev aldı. 2009-2012 yılları arasında Nortel Netaş Networks A.Ş. Teknoloji Geliştirme Mühendisi olarak çalıştı. Bu yıllar arasında CICM Voip tabanlı sistemin Global Ürün Destek Mühendisi olarak, AA2 SIP tabanlı sistemin yazılım geliştirme bölümünde ve aynı sistemin Patch (Yama) grubunda görev aldı. 2012 yılından itibaren Abant İzzet Baysal Üniversitesi Bilgi İşlem Daire Başkanlığı yazılım bölümünde Uzman olarak görev yapmaktadır.