

GPS VERİCİ VE ALICISI

SİNYAL SİMÜLASYONU

Yüksek Lisans Tezi

Ahmet Esat SÜZER

Eskişehir- 2017

GPS VERİCİ VE ALICISI SİNYAL SİMÜLASYONU

Ahmet Esat SÜZER

YÜKSEK LİSANS TEZİ

**Havacılık Elektrik ve Elektronik Anabilim Dalı
Danışman: Doç. Dr. Hakan OKTAL**

**Eskişehir
Anadolu Üniversitesi
Fen Bilimleri Enstitüsü
Aralık 2017**

JÜRİ VE ENSTİTÜ ONAYI

Ahmet Esat SÜZER'in “GPS Verici ve Alıcısı Sinyal Simülasyonu” başlıklı tezi 22/12/2017 tarihinde aşağıdaki jüri tarafından değerlendirilerek “Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliği”nin ilgili maddeleri uyarınca, **Havacılık Elektrik ve Elektronik**i Anabilim dalında Yüksek Lisans tezi olarak kabul edilmiştir.

Unvanı Adı Soyadı

İmza

Üye (Tez Danışmanı)	: Doç. Dr. Hakan OKTAL	
Üye	: Yard. Doç. Dr. Kadriye YAMAN	
Üye	: Yard. Doç. Dr. Ayşe KAHVECİOĞLU	

.....

Enstitü Müdürü

ÖZET

GPS VERİCİ VE ALICISI SİNYAL SİMÜLASYONU

Ahmet Esat SÜZER

Havacılık Elektrik ve Elektronik Anabilim Dalı

Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Aralık 2017

Danışman: Doç. Dr. Hakan OKTAL

1990'lı yılların başından itibaren popüler olmaya başlayan uydu sistemleri günümüzde hem sivil, hem askeri birçok alanda yaygın olarak kullanılmaktadır. Önceleri sadece askeri amaçlı kullanılan uydu temelli seyrüsefer sistemleri, zaman içerisinde sivil kullanıma da açılmış, tüm ulaşım araçlarında küresel boyutta konum belirleme amaçlı seyrüsefer yardımcısı olarak kullanılmaya başlamıştır. Farklı ülkeler tarafından geliştirilmiş uydu temelli Küresel Konum Belirleme sistemlerinden birisi de ABD tarafından geliştirilen GPS (Global Positioning System)'tir. Başlangıçta 21 operasyonel 3 yedek toplam 24 uydu ile hizmet vermesi planlanan sistem, şimdilerde toplam 31 operasyonel uydu ile kullanıcılarına hizmet vermektedir. Günümüzde GPS uyduları, mevcut L1 (1575,42 MHz) ve L2 (1227,60 MHz) frekansına ilave olarak L5 (1176,45 MHz) frekansının da eklenmesi ile toplam üç farklı frekanstan yayın yapmaktadır.

Bu çalışmanın amacı, GPS L1 sinyalinin simülasyonu ve kullanıcı arayüzü tasarımının gerçekleştirilmesidir. Bu doğrultuda öncelikle kullanıcı tarafından seçilen dört uyduda L1 sinyali üretilmiş ve daha sonra üretilen sinyallere ortam gürültüsü eklenerek alıcıya ulaşması sağlanmıştır. Alıcı düzeyinde gelen sinyallerin demodülasyon işlemi sonucunda kullanıcı için son derece önemli bilgiler içeren seyrüsefer mesajı elde edilmiştir. Simülasyonda kullanılan sinyal işleme teknikleri, MATLAB (Matrix Laboratory) R2015b programlama dili aracılığıyla gerçekleştirilmiştir. Çalışmanın son aşamasında ise alıcının gelen sinyallerden elde ettiği sonuçlar ve konum bilgilerinin sunulabilmesi amacıyla MATLAB-GUI (Graphical User Interface) kullanılarak bir arayüz tasarımı gerçekleştirilmiştir. Çalışmanın uygulama aşamasında, geliştirilen arayüz ile kullanıcının hesaplanan ve gerçek konumunun harita üzerinde gösterilme imkânı sunulmuştur. Ayrıca tasarlanan arayüz ile, uydu sinyalinin üretiminden alıcıda seyrüsefer mesajının elde edilmesine kadar gerçekleşen işlemler sonucu oluşan sinyal çıktıları incelenebilmektedir.

Anahtar Sözcükler: GNSS, Yazılım tabanlı GPS alıcı, Sinyal işleme, Uydu sinyal simülasyonu.

ABSTRACT

GPS TRANSMITTER AND RECEIVER SIGNAL SIMULATION

Ahmet Esat SÜZER

Department of Avionics

Anadolu University, Graduate School of Sciences, December 2017

Supervisor: Assoc. Prof. Dr. Hakan OKTAL

Nowadays, satellite systems that has started to be popular since early 1990s, are used in many military and civil applications. At the beginning, although the satellite-based navigation systems were used for only military purposes, they were opened up civil use over time and have begun to be used as navigation aids for all means of transport. One of the global positioning determination system developed by different countries is GPS (Global Positioning System) developed by USA. The system planned to service with 24 satellites, which was 21 operational and 3 spare, currently services to users with 31 operational satellites. Nowadays, GPS satellites broadcast in three different frequencies along with L5 (1176,45 MHz) frequency added to the existing L1 (1575,42 MHz) and L2 (1227,60 MHz) frequencies.

The aim of this study is to simulate GPS L1 signal and to design the user interface. In this framework, first of all, L1 signals are generated in four satellites selected by the user then these signals are reached to the receiver by interfering with the noise. In receiver stage after the coming signals are demodulated, navigation message including so important informations for the users is obtained. Signal processing techniques used in the simulation are implemented by means of MATLAB (Matrix Laboratory) R2015b. An interface is designed by using MATLAB-GUI (Graphical User Interface) to display the results and position informations which the receiver obtains from the coming signals. Application stage of the study enables to show the calculated and real receiver positions on the map using designed interface. Also via the interface it is possible to analyse signal outputs of process that occur from generation of satellite signal to navigation message obtained in the receiver.

Keywords: GNSS, Software-based GPS receiver, Signal processing, Satellite signal simulation.

TEŞEKKÜR

İki yıl boyunca bu çalışmanın gerçekleştirilmesinde, her zaman anlayış, hoşgörü ve destek gördüğüm değerli bilgilerini paylaşan ve kullandığı her kelimenin hayatıma kattığı önemini asla unutmayacağım saygıdeğer danışman hocam; Doç. Dr. Hakan OKTAL'a, çalışmam boyunca benden bir an olsun yardımlarını esirgemeyen, çalışma süresince tüm zorlukları benimle göğüsleyen ve hayatımın her evresinde bana destek olan Sevgili Eşim ve Aileme sonsuz teşekkürlerimi sunarım.

Ahmet Esat SÜZER

Eskişehir, 2017

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Bu tezin bana ait, özgün bir çalışma olduğunu; çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; bu çalışma kapsamında elde edilen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi; bu çalışmanın Anadolu Üniversitesi tarafından kullanılan “bilimsel intihal tespit programı”yla tarandığını ve hiçbir şekilde “intihal içermediğini” beyan ederim. Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçları kabul ettiğimi bildiririm.

Ahmet Esat SÜZER

İÇİNDEKİLER

Sayfa

BAŞLIK SAYFASI	i
JÜRİ VE ENSTİTÜ ONAYI.....	ii
ÖZET	iii
ABSTRACT.....	iv
TEŞEKKÜR	iv
ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ.....	v
İÇİNDEKİLER	vi
1. GİRİŞ	1
2. GPS SİNYAL YAPISI	8
2.1. İkili Faz Kaydırmalı Anahtarlama (Binary Phase Shift Keying, BPSK)	9
2.2. C/A (Coarse/Acquisition) Kodu ve Özellikleri	10
2.3. P-Kodu (Precision/Protected- Code) ve Özellikleri	15
2.4. Seyrüsefer Mesajı.....	18
2.5. GPS Sinyal Üretimi.....	21
3. GPS ALICI YAPISI.....	23
3.1. Anten	25
3.2. Ön Uç	26
3.2.1. Kuvvetlendirici ve filtre.....	26
3.2.2. Aşağı kaydırıcı.....	27
3.2.3. Osilatörler ve frekans sentezleyiciler	28
3.2.4. Analog-dijital dönüştürücüler ve otomatik kazanç kontrolü	29
3.3. Sinyal Yakalama	31
3.3.1. Seri arama yakalama (Serial search acquisition)	32
3.3.2. Paralel frekans alan arama yakalama (Parallel frequency space search acquisition)	33

3.3.3. Paralel kod fazı arama yakalama (Parallel code phase search acquisition)	35
3.4. İzleme	36
3.4.1. Taşıyıcı izleme (Carrier tracking)	37
3.4.2. Kod izleme (Code tracking)	38
3.4.3. Bütünleşik izleme döngüsü.....	41
3.5. Seyrüsefer Mesajının Elde Edilmesi	41
3.6. Uzaklık Hesaplama (Pseudorange)	48
3.7. Konum Hesaplama	49
4. GPS ALICI ARAYÜZÜ TASARIMI.....	53
4.1. Anasayfa	53
4.2. Başlangıç Ekranı.....	53
4.3. Sinyal Üretimi	56
4.4. Kanal.....	58
4.5. Sinyal Çözümleme.....	59
4.6. Pseudorange Hesabı.....	60
4.7. Konum Hesaplama	62
4.8. Sinyal Analizi.....	64
5. SONUÇ	67
KAYNAKÇA.....	69
EK -1: SİNYAL ANALİZİ DALGA FORMLARI	73
EK -2A: SİMÜLASYON KODLARI (Anasayfa)	77
EK -2B: SİMÜLASYON KODLARI (Sonuç Ekranı).....	81
EK -2C: SİMÜLASYON KODLARI (Konum Hesaplama)	90
ÖZGEÇMİŞ	92

ŞEKİLLER DİZİNİ

	<u>Sayfa</u>
Şekil 2.1. BPSK blok diyagramı.....	9
Şekil 2.2. BPSK örneği.....	10
Şekil 2.3. C/A kodu üretimi.....	11
Şekil 2.4. PRN15 için üretilen ilk 200 çip.....	14
Şekil 2.5. PRN21 ve PRN5 çapraz korelasyon sonucu	15
Şekil 2.6. PRN21 oto korelasyon sonucu	15
Şekil 2.7. X1A kaydedicisinin iç yapısı	17
Şekil 2.8. X1B kaydedicisinin iç yapısı.....	17
Şekil 2.9. X2A kaydedicisinin iç yapısı	18
Şekil 2.10. X2B kaydedicisinin iç yapısı.....	18
Şekil 2.11. P- kodu üretimi.....	19
Şekil 2.12. Seyrüsefer mesaj yapısı	20
Şekil 2.13. GPS sinyal üretimi.....	22
Şekil 3.1. Genel GNSS alıcı yapısı.....	23
Şekil 3.2. Genel GPS alıcı yapısı.....	24
Şekil 3.3. Mikser yapısı	27
Şekil 3.4. AGC iç yapısı	31
Şekil 3.5. Seri arama yakalama blok diyagramı	32
Şekil 3.6. Paralel frekans alan arama yakalama blok diyagramı	33
Şekil 3.7. Paralel frekans alan arama yakalama örneği	34
Şekil 3.8. Paralel frekans alan arama yakalama sonucu	35
Şekil 3.9. Paralel kod fazı arama yakalama blok diyagramı	36
Şekil 3.10. PLL yapısı	37
Şekil 3.11. Costas döngüsü.....	38

Şekil 3.12. Erken-geç izleme döngüsü	39
Şekil 3.13. Erken-geç döngüsü örnek korelasyon	40
Şekil 3.14. Altı korelatörlü DLL yapısı	40
Şekil 3.15. Bütünleşik izleme döngüsü	42
Şekil 3.16. En iyileştirilmiş izleme döngüsü	43
Şekil 3.17. İzleme döngüsü sonucu elde edilen seyrüsefer mesajı	44
Şekil 3.18. Preamble korelasyon sonucu	45
Şekil 3.19. Hata ayıklama yöntemleri	46
Şekil 3.20. TLM ve HOW alt-çerçeve yapısı	46
Şekil 3.21. Eşlik biti kodlama eşitlikleri.....	47
Şekil 3.22. GPS zamanı, z-sayacı ve kırılmış z-sayacı.....	48
Şekil 4.1. Arayüz akış diyagramı.....	54
Şekil 4.2. Arayüz anasayfası.....	55
Şekil 4.3. Parametrelerin girildiği başlangıç ekranı	57
Şekil 4.4. Faz-uyumlu sinyal algılama blok diyagramı	60
Şekil 4.5. Hesaplanan sonuçlar.....	61
Şekil 4.6. Hesaplanan ve gerçek alıcı konumları.....	63
Şekil 4.7. Sinyal analizi	66

TABLÖLAR DİZİNİ

	<u>Sayfa</u>
Tablo 2.1. GPS uyduları için C/A kodu üretimi.....	12
Tablo 2.2. P-kodu kaydedicilerinin polinom fonksiyonu ve başlangıç şartları.....	16
Tablo 2.3. Alt-çerçeve kimlik kodu	20
Tablo 4.1. Gürültü analizi	64

SİMGELER VE KISALTMALAR DİZİNİ

f_o	: GPS temel frekans değeri
f	: Frekans
c	: Işık hızı
Δt	: Zaman farkı
α	: Alfa
ρ	: Pseudorange
bps	: Bits per second (Saniyedeki bit sayısı)
Mcps	: Megachips per second (Saniyedeki megaçip sayısı)
MHz	: Mega Hertz
ABD	: Amerika Birleşik Devletleri
ADC	: Analog to Digital Converter (Analog Dijital Dönüştürücü)
AGC	: Automatic Gain Control (Otomatik Kazanç Kontrolü)
AM	: Amplitude Modulation (Genlik Modülasyonu)
AWGN	: Additive White Gauss Noise (Toplanır Beyaz Gauss Gürültü)
BCH	: The Bose, Chaudhuri and Hocquenghem Encoding (BCH Kodlama)
BOC	: Binary Offset Carrier (İkili Offset Taşıyıcı)
BPSK	: Binary Phase Shift Keying (İkili Faz Kaydırmalı Anahtarlama)
CBS	: Coğrafi Bilgi Sistemleri
CDMA	: Code-Division Multiple Access (Kod Bölmeli Çoklu Erişim)
CNS/ATM	: Communication, Navigation and Surveillance/Air Traffic Management (Haberleşme, Seyrüsefer, İzleme ve Hava Trafik Yönetimi)
CORS	: Continuously Operating Reference Station

(Sürekli Gözlem Yapan Referans İstasyonları)

C/A	: Coarse/Clear Acquisition (Serbest Edinim)
DLL	: Delay Lock Loop (Gecikme Kilitlemeli Döngü)
DO	: Dahili Osilatör
FLL	: Frequency Lock Loop (Frekans Kilitlemeli Döngü)
FM	: Frequency Modulation (Frekans Modülasyonu)
GNSS	: Global Navigation Satellite Systems (Küresel Uydu Seyrüsefer Sistemleri)
GPS	: Global Positioning System (Küresel Konumlandırma Sistemi)
GUI	: Graphical User Interface (Grafiksel Kullanıcı Arayüzü)
HOW	: Hand over Word
ICAO	: International Civil Aviation Organization (Uluslararası Sivil Havacılık Örgütü)
IF	: Intermediate Frequency (Ara Frekans)
LNA	: Low Noise Amplifier (Düşük Gürültülü Kuvvetlendirici)
LSB	: Least-Significant Bit (Düşük Değerlikli Bit)
L1	: Link1
L2	: Link2
MATLAB	: Matrix Laboratory (Matris Laboratuvarı)
MSB	: Most-Significant Bit (Yüksek Değerlikli Bit)
NCO	: Numerically Controlled Oscillator (Nümerik Kontrollü Osilatör)
NRZ	: Non-Return-to-Zero Encoding (Sıfıra Dönmeyen Kodlama)
P Code	: Precision/Protected Code (Hassas/Korumalı Kod)
PVT	: Position Velocity Time (Konum Hız Zaman)
QPSK	: Quadrature Phase Shift Keying (Dörtlü Faz Kaydırmalı Anahtarlama)
RF	: Radio Frequency (Radyo Frekansı)

RHCP	: Right Hand Circular Polarization (Sağ El Polarizasyonu)
PLL	: Phase Lock Loop (Faz Kilitlemeli Döngü)
PRN	: Pseudo Random Noise (Sözde Rasgele Gürültü)
PSD	: Power Spectral Density (Spektral Güç Yoğunluğu)
SNR	: Signal to Noise Ratio (Sinyalin Gürültüye Oranı)
TLM	: Telemetry (Telemetri)
TOW	: Time of Week (Haftanın Zamanı)
USB	: Universal Serial Bus (Evrensel Seri Veriyolu)
UTC	: Coordinated Universal Time (Eşgüdümlü Evrensel Zaman)
VGA	: Analog Amplifier Variable Gain (Ayarlanabilir Kazançlı Analog Kuvvetlendirici)
XOR	: Exclusive-OR (Özel Veya)

1. GİRİŞ

Tarihte uzun yıllardan beri insanlar bir yerden başka bir yere ulaşmak ve dünya üzerindeki konumunu belirlemek için çeşitli yollar geliştirmişlerdir [1]. İlk zamanlarda bu amaç için taşlar, kuşlar, duman, yıldızlar ve pusula gibi çeşitli yöntemler kullanılırken, günümüzde ise Küresel Uydu Seyrüsefer Sistemleri (Global Navigation Satellite Systems)' i yaygın olarak kullanılmaktadır. Küresel Uydu Seyrüsefer Sistemi, uydulardan aldıkları bilgiler doğrultusunda dünyanın herhangi bir yerinde kullanıcıların konumlarını belirlemek için kullandıkları uydu sisteminin genel ismidir [2, 3]. Uluslararası Sivil Havacılık Örgütü (ICAO-The International Civil Aviation Organization) GNSS'i "*Gerekli seyrüsefer performansını sağlamak üzere bir veya daha fazla uydu, uçak alıcıları, sistem bütünlüğünü izleme ve hassasiyet artırımını kapsayan dünya çapında konum ve zaman belirleme sistemi*" olarak tanımlamaktadır [3, 4]. 1990'lı yılların başından itibaren popüler olmaya başlayan uydu sistemleri günümüzde hem sivil, hem askeri birçok alanda yaygın olarak kullanılmaktadır. Önceleri sadece askeri amaçlı kullanılan uydu temelli seyrüsefer sistemleri, zaman içerisinde sivil kullanıma da açılmış, tüm ulaşım araçlarında küresel boyutta konum belirleme amaçlı seyrüsefer yardımcısı olarak kullanılmaya başlamıştır.

Uydu sistemlerinin ilk gelişiminin Amerika Birleşik Devletleri ve eski adı ile Sovyetler Birliği arasında yaşanan ekonomik, askeri, savunma gibi birçok alandaki rekabete dayandığı görülmektedir. 04 Ekim 1957 yılında Sputnik-1 uydusunun uzaya fırlatılması uydu seyrüsefer sistemlerinin başlangıç noktası olmuştur [5]. Sputnik uydularının uzaya gönderilmesinden sonra Amerika 1960'lı yılların başında hassas seyrüsefer için bir sistem oluşturmaya karar vermiştir ve 1964 yılında ABD Deniz Kuvvetleri Seyrüsefer Uydu Sistemi (ABD Navy Navigation Satellite System) veya diğer adıyla Transit isimli uydu seyrüsefer sistemini geliştirmiştir. Transit'den sonra ABD Deniz Kuvvetleri Zaman Seyrüsefer Sistemi (Timation) ve ABD Hava Kuvvetleri projesi 621B geliştirilmiştir. İlk operasyonel seyrüsefer sistemi olan Transit yeryüzünden 1100 km uzaklıkta 6 uydudan oluşmaktaydı ve kullandığı sinyal tipi sürekli dalgaydı [1, 6]. Uydunun en yakın olduğu zaman maksimum Doppler kaymasından bulunabiliyordu. 1967 yılında fırlatılan Timation programında yer alan atomik saatle uydu yörüngesi daha doğru tahmin edilebilirken yine atomik saat sayesinde yer kontrol istasyonunun uyduyu güncelleme sıklığı azaldı. Hava Kuvvetleri 621B projesinin en büyük gelişimi üç boyutlu bilgi sağlamasıdır. Bu sistemde taşıyıcı frekansını modüle etmek için Sözde Rasgele

Gürültü (Pseudo Random Noise) adlı kodlu sinyal kullanılmıştır ve başarılı bir teknik olduğu kanıtlanmıştır [1, 7]. Küresel Konumlandırma Sistemi (Global Positioning System), bahsedilen üç sistemin tecrübesinden faydalanarak 1973 yılında ABD Savunma Bakanlığı tarafından yapılan çalışmalar sonucu askeri amaçlı kullanılmak üzere geliştirilmiştir. GPS, başlangıçta yalnızca askeri amaçla kullanılırken, daha sonraki yıllarda sivil kullanıma da açılmıştır [4, 8]. İlk GPS uydusu 1978 yılında fırlatılmıştır [7]. Fırlatılan ilk uydular deneme uyduları olarak isimlendirilen Block I serisi olarak gönderilmiştir.[5]. Başlangıçta 6 yörüngede 21 operasyonel ve 3 yedek uydu olarak planlanan GPS, ortalama ömürlerine göre bazı uyduların hala sorunsuz olarak çalışması ve günümüzdeki yenileme çalışmaları nedeni ile 12 Block IIR, 12 Block IIR-M ve 7 Block IIF olmak üzere toplam 31 operasyonel uyduya sahiptir [6-10]. Yedek uydular, operasyonel uydulardan herhangi birinde bir arıza meydana gelmesi durumunda kullanılmak içindir.

GPS uzay, kontrol ve kullanıcı olmak üzere üç temel bölümden meydana gelmektedir. Uzay bölümü kullanıcılara radyo sinyali ile faydalı bilgi gönderen uydulardan oluşmaktadır. Uydudan gönderilen sinyal, seyrüsefer mesajı, PRN kodu ve taşıyıcıyı içermektedir. Uydular yeryüzünden yaklaşık 20.200 km uzaklığa konuşlandırılmıştır ve periyodu 11 saat 58 dakika 2,05 saniyedir [5]. Periyod süresinden dolayı bir GPS uydusu dünyanın herhangi bir bölgesinden günde iki kez geçmektedir. Uydularda çok yüksek doğruluğa sahip atomik saatler vardır. Her bir yörünge ekvatorla 55°'lik bir eğim (inclination) ve 26.560 km'lik yarıçapa sahiptir. Herhangi bir kullanıcının konumunu belirlemesi için gerekli uydu sayısı en az dördür. Her bir uydu yaklaşık 5 saat ufuk hattı üzerinde kalmaktadır [11]. Güneş enerjisi ile çalışan uyduların güneş tutulması, gölgeleme gibi güneş enerjisi kesintilerine karşı yedek bataryaları ve yörünge düzeltmeleri için de küçük ateşleyici roketleri vardır [12]. Uyduların ortalama ömürleri 7.5-10 yıl arasındadır [6]. Block I, Block II, Block IIA, Block IIR, Block IIR-M, Block IIF ve GPS III olmak üzere yedi farklı tip GPS uydusu mevcuttur.

Kontrol bölümünün ana işlevi uydu saatlerini ayarlama, uydu yörüngelerinde meydana gelen sapmaları düzeltme, GPS uydularını izleme ve oluşan hataları düzeltip yer antenleri aracılığıyla uydulara mesaj olarak gönderme, uyduların saat hatalarını belirleme ve düzeltme, batarya şarj durumu, yörünge bilgileri, uydu sağlık durumu ve doğru çalışıp çalışmadığını kontrol etme olarak sıralanabilir [4, 13]. Diğer görevi ise uydularda herhangi bir problem yaşandığında yedek uyduyu devreye sokmaktır. Bu

bölüm günde en az bir kere uyduların seyrüsefer mesajlarında gönderilen almanak, efemeris ve uydu saati gibi parametreleri düzeltmektedir. [14]. Kontrol bölümü, Ana kontrol istasyonu (Master Control Station), izleme istasyonları (Monitor Station) ve yer antenlerinden (Ground Antenna) oluşmaktadır. Ana kontrol istasyonu toplam bir tane olmak üzere ABD Colorado Springs’ de yer almaktadır. Ana kontrol istasyonuna, atmosferik veriler, uydu sağlık durumu, batarya şarj durumu, yörünge bilgileri, uydu atomik saati, uydu almanak bilgisi ham veri olarak izleme istasyonlarından gelir. Ana kontrol istasyonu gerekli düzeltmeleri yaptıktan sonra uydulara gönderilmek üzere yer antenlerine iletir. Yer antenleri de S-bandı aracılığı ile GPS uydularına göndererek gerekli düzeltmelerin yapılmasını sağlar [8, 15]. Uydularda meydana gelen hataları doğru olarak tespit edebilmek için izleme istasyonlarının koordinatları çok hassas olarak belirlenmektedir. 2017 Mayıs ayı itibariyle izleme istasyonları Colorado, Hawaii, Alaska, Cape Canaveral, USNO Washington, Ecuador, United Kingdom, Uruguay, Ascension, South Africa, Bahrain, South Korea, Australia, New Zealand, Diego Garcia ve Kwajalein olmak üzere on altı farklı yerde bulunmaktadır [16]. İzleme istasyonundan tüm uydular sürekli olarak izlenmekte ve yörüngedeki uyduların genel sağlığı, hassas irtifa, konum ve hız bilgileri kontrol edilmektedir [5]. Her izleme istasyonu, ana kontrol istasyonunun altında çalışır. Uydu yörünge bilgisi, uydu sağlık durumu ve yerel meteorolojik verileri elde eder ve daha sonra bu bilgileri ana kontrol istasyonuna gönderir [14]. Ana kontrol istasyonu da aldığı bu bilgileri düzelterek yer antenleri aracılığıyla S-bandını kullanarak ilgili uydulara gönderir. Yer antenleri uzay bölümü ile kontrol bölümü arasındaki iletişimi sağlamaktadır. Diego Garcia, Kwajalein, Ascension ve Cape Canaveral olmak üzere toplam dört farklı yerde yer antenleri bulunmaktadır [17]. Bunların görevi, ana kontrol istasyonundan düzeltilmiş olarak gönderilen efemeris bilgileri, yörünge ve saat düzeltmelerini S-bandı üzerinden ilgili uydulara göndermektir [6, 14].

Kullanıcı bölümü sivil ve askeri amaçlı zaman, konum ve hız bilgileri elde etmek isteyen alıcılardan oluşmaktadır. Alıcılar dünyanın herhangi bir yerinden uydulardan gelen sinyal ile konumlarını belirleyebilmektedir. GPS askeri ve sivil amaçlı olarak farklı alanlarda kullanılmaktadır [17].

GPS’in askeri kullanım alanları; kara, deniz ve hava araçlarının ulaşımı, arama-kurtarma, hedef bulma, füze güdümü, uçakların görüşün sınırlı ya da hiç olmadığı hava koşullarında iniş ve kalkış gibi alanlarda kullanılmaktadır [6].

Sivil olarak ise; kara, deniz ve hava araçlarının ulaşımı, Jeodezik ve Jeodinamik ölçmeler, bilimsel çalışmalar, kadastral ölçmeler, kinematik GPS destekli fotogrametrik çalışmalar, yerel ve küresel deformasyon ölçmeleri (baraj, yol, viyadük v.b), araç takip sistemleri, uçakların görüşün sınırlı ya da hiç olmadığı hava koşullarında iniş ve kalkışı, havacılıkta CNS/ATM uygulamaları, aktif kontrol ağları (CORS), CBS veri tabanlarının geliştirilmesi, turizm, tarım, ormancılık, arkeoloji, sportif faaliyetler, asayiş gibi alanlarda kullanılmaktadır [6].

Bu çalışmanın amacı gelecekte uçuşun her aşamasında kullanılması planlanan GPS uydularının temel çalışma prensipleri doğrultusunda GPS L1 sinyalinin üretilmesi, alıcıda sinyalin demodülasyon işleminin gerçekleştirilmesi ve sonuçları göstermek üzere tasarlanan kullanıcı ara yüzü geliştirilmesidir. Sinyalin üretim aşamasında farklı amaçlar için kullanılan, yazılım geliştirme aracı olan MATLAB (Matrix Laboratory) programlama dilinden faydalanılmıştır. Kullanıcı ara yüzü tasarımı da MATLAB ortamında yer alan ara yüz geliştirme ve tasarlama amaçları için kullanılan GUI (Graphical User Interface) programı ile gerçekleştirilmiştir.

Çalışmanın amacı kapsamında konu ile ilgili daha önce yapılan çalışmalar literatür araştırması kapsamında incelenmiştir. Kullanılan teknikler, yöntemler ve simülasyon için kullanılan programlar dikkate alınarak özellikle güncel olan çalışmalara aşağıda yer verilmiştir.

Elango ve Sudha (2016), laboratuvar ortamında yazılım tabanlı GPS alıcısının performansını test etmek ve değerlendirmek için düşük maliyetli, basit ve yeni bir GPS L1 sinyal simülatörü tasarlamıştır. Bu çalışmada, öncelikle bir dizüstü bilgisayar / kişisel bilgisayarda herhangi bir donanım ihtiyacı duymadan çalışan, tamamen araştırmacılara ve geliştiricilere kapsamlı bir analiz ve test esnekliği sunabilen bir GPS yazılım simülatörü önerilmiştir. Ön uç bölümü donanımsal olarak gerçekleştirilen simülasyonda RINEX protokollü gerçek GPS sinyalleri ile yazılım tabanlı temel bant işlemleri, seyrüsefer mesajını elde etme işlemleri gerçekleştirilmiş ve aynı zamanda çoklu yol etkisi sisteme dahil edilmiştir [18].

Tomé (2015), yazılım tabanlı radyo (Software Defined Radio, SDR) yaklaşımı ile GPS L1 C/A sinyali için Paralel kod fazı arama ve Paralel frekans alan arama olmak üzere iki farklı sinyal yakalama yöntemi kullanmıştır. Gelen sinyalden seyrüsefer mesajını başarılı bir şekilde elde etmek için kullanılan Costas döngüsü ile izleme yöntemi yazılım tabanlı bir alıcı tasarımı ile gerçekleştirilmiştir [19].

Landry ve Ilie (2015), Yazılım Tabanlı Radyo Aygıtı (Universal Software Radio Peripheral, USRP) donanımsal elemanı ile ürettiği GPS L1 sinyalini alıcılara göndererek tasarlanan GPS simülatörünün performansını değerlendirmiştir [20].

Soghoyan, Suleiman ve Akopian (2014), GPS sinyal yakalama aşamasını hızlandırmak için FPGA ile LabVIEW programları kullanarak LabVIEW - FPGA grafiksel kullanıcı arayüzü geliştirmiştir. Ayrıca çalışmada GPS sinyalinin simülasyonu, test edilmesi, algoritma ve performans analizi için LabVIEW ortamında GPS alıcı test platformu geliştirilmiştir [21].

Abhishek, Charitha ve Goutham (2014), gerçek GPS sinyallerini kullanarak SIMULINK ortamında alıcı tasarımı gerçekleştirmiştir [22].

Zubaroglu (2013), MATLAB ortamında yazılım tabanlı GPS alıcısının bileşenlerini modellemiş ve performanslarını değerlendirmiştir. Toplanan gerçek GPS sinyalleri kullanılarak hazırlanan modeller ile alıcı konumu hesaplanmıştır. GPS sinyallerinin MATLAB ortamında oluşturulacak yazılım tabanlı GPS alıcısı tarafından işlenebilmesi için IF sinyale dönüştürülmesi gerekmektedir. Bu amaçla “SiGE GN3Sv2 GPS Sampler” cihazı kullanılmıştır. Alıcı tasarımı ile gerçek GPS sinyalleri kullanılarak sinyal yakalama, izleme, seyrüsefer mesajını çözme, pseudorange ve konum hesabı gerçekleştirilmiştir. [23].

Akopian ve Soghoyan (2013), LabVIEW tabanlı SDR platformunu incelemiş ve LabVIEW tabanlı GPS alıcısı gerçekleştirilmiştir. Ayrıca yazılım tabanlı alıcı geliştirme platformu için LabVIEW programlama dilinin uygunluğunu ele almıştır [24].

Kassas, Bhatti ve Humphreys (2013), LabVIEW programlama dilini kullanarak GPS sinyal yakalama ve izleme aşamalarını yazılım tabanlı olarak gerçekleştirmiş ve sonuçları LabVIEW ön panelinde göstermiştir [25].

Guo, Wang ve Chao (2012), GPS IF sinyali için yazılım tabanlı simülasyon metodu önermiştir. Gecikme zamanı, en fazla ± 5 kHz Doppler frekansı, gürültü ve çoklu yol etkileri dikkate alınarak simüle edilen IF sinyalinin frekans spektrumu ile gerçek GPS IF sinyal frekans spektrumu karşılaştırılmış ve simülasyonun doğruluğu test edilmiştir [26].

Bright (2012), L2C sinyalinin yakalama ve izleme aşamalarını MATLAB ortamında gerçekleştirmiş ve sonuçta L1 alıcısına göre performansını karşılaştırmıştır [27].

Rao ve Falco (2012), GPS ve Galileo sinyallerini eş zamanlı işleyebilen bir SDR alıcı yapısını incelemiştir. Çalışmada, pseudorange ve PVT (Position, Velocity, Time) bilgilerini hesaplamıştır. Hesapladığı hız bilgilerine Kalman Filtresi ve En Küçük Kareler yöntemi uygulayarak sonuçları değerlendirmiştir [28].

Brown, Tredway ve Taylor (2011), USRP ile tasarladığı arayüzü kullanarak RF sinyali üretmiştir. Arayüzde RF sinyal için gerekli parametreler girilmekte ve USRP ile RF sinyali üretilerek alıcıya gönderilmektedir. Böylece alıcıda parametrelerin etkileri incelenebilmektedir [29].

Peña (2010), mevcut modern GPS ve Galileo sinyallerini analiz ederek ikili efemeris tahmin yöntemi ve yeni demodülasyon yöntemi gibi yeni teknikleri önermiştir. Sonuçta önerilen ile mevcut yöntemin performansı karşılaştırılmıştır [30].

Zheng (2010), öncelikle sinyal yakalamada kullanılan metotları tanıtmış ve Frekans Domeni Paralel Yakalama algoritması geliştirerek performansını incelemek üzere MATLAB ortamında gerçekleştirmiştir. Sonuçta öne sürülen yöntemin diğer metotlara göre taşıyıcı frekans ve kod fazı hesaplama doğruluğunun ve hızının arttığı ifade edilmiştir [31].

Wei vd. (2009), farklı yakalama metotlarını incelemiş ve gerçek GPS sinyallerinin yakalama aşamasının simülasyon ortamında gerçekleştirilebileceğini göstermiştir [32].

Günaydın (2005), yaptığı çalışmada yazılım tabanlı alıcı tasarımını, sinyal işleme ve seyrüsefer olmak üzere iki bölüme ayırmıştır. Ön uç bölümünde NordNavT R-25 cihazı kullanılarak örneklenmiş sinyal elde edilmiş ve MATLAB ortamına aktarılmıştır. İzleme aşamasında Geleneksel ve Blok Ayarlamalı Senkronize Edilmiş Sinyal (BASS) yöntemi kullanılarak iki metodun performansı karşılaştırılmıştır. Seyrüsefer bölümünde AshtechT ve TrimbleT GPS alıcısından faydalanılmıştır. Seyrüsefer bölümünde uydu konum hesabı, pseudorange düzeltilmesi, Kalman ve En Küçük Kareler yöntemleri gerçekleştirilmiştir. Sonuçlar gerçek GPS alıcısının çıktıları ile karşılaştırıldığında, alıcı programının uygun bir şekilde çalıştığı görülmüştür [33].

Ma, Lachapelle ve Cannon (2004), yaptığı simülasyon ile gerçek ve yazılım tabanlı oluşturulan GPS sinyallerini kullanarak yakalama ve izleme performanslarını incelemiştir. RF sinyalini IF sinyale dönüştürmek için aşağı kaydırıcı kullanmıştır. Temel bant işlemleri ve seyrüsefer mesajını elde etme işlemleri gerçekleştirilmiştir. Yakalama aşamasının hızlandırılmasına yönelik önerilerde bulunmuştur [34].

Dong (2003), L1 C/A kodu için yazılım tabanlı IF GPS sinyal simülatörü geliştirmiş ve bunu gerçek zamanlı sinyaller ile test etmiştir. Sinyal simülatörü MATLAB ortamında tasarlanmış, ancak işlem hızını artırmak için C programlama dilinden faydalanılmıştır. Uydu saat hatası, iyonosferik hata, gürültü oranı gibi sinyalin yayılımdan dolayı oluşan çeşitli hataların fonksiyonu olarak ifade edilebilen dijital IF GPS sinyali için matematiksel model geliştirilmiştir. Zaman, frekans domeni ve pozisyon açısından simülatörde bulunan sonuçlar ile donanımsal olarak toplanan IF GPS sinyalleri karşılaştırılmıştır. Ayrıca çalışmada, sinyal yakalama ve izlemenin yanında taşıyıcı faz ölçümü, pseudorange ölçümü, bit senkronizasyonu, alt-çerçeve senkronizasyonu ve seyrüsefer mesajının elde edilmesi gibi işlemler gerçekleştirilmiştir [35].

Brown, Gerein ve Taylor (2000), NAVSYS tarafından geliştirilen MATLAB araç çubuğu (toolbox) ile GPS sinyalini test ve analiz etmiştir. Geliştirilen araç çubuğu PRN kodu üretme, iyonosferik etki, kod izleme ve taşıyıcı izleme gibi özellikleri içermektedir [36].

Çalışmanın ikinci bölümünde GPS sisteminin sinyal yapısı, uydulara kimlik olarak atanan C/A (Coarse/Acquisition) ve P (Precision/Protected Code) kodunun özellikleri ile üretimi, seyrüsefer mesaj içeriği, kullanılan modülasyon tekniği ve sinyal üretimi ile ilgili bilgiler detaylı bir şekilde ele alınmıştır.

Üçüncü bölümde, uydulardan gelen GPS sinyalinin anten ile alınmasından seyrüsefer mesajının çözülmesine kadar filtreleme, korelasyon, hata ayıklama gibi sinyal işleme adımlarından ayrıntılı olarak bahsedilmiştir.

Dördüncü bölümde, GPS sinyalinin simülasyonu ve uygulaması kapsamında gerçekleştirilen işlem basamakları, elde edilen ekran çıktıları ve tasarlanan ara yüz açıklanmıştır.

Günlük hayatta sıklıkla kullanılan GPS L1 sinyalinin simülasyonu ve çıktılarını göstermek için tasarlanan kullanıcı ara yüzü, çalışmanın sonuç bölümünde değerlendirilmiş ve bu konuda yapılacak araştırmalara ve çalışmalara yön vereceği düşünülen öneriler sunulmuştur.

2. GPS SİNYAL YAPISI

GPS sistemi dünya üzerindeki her noktada, her türlü hava koşulunda, kesintisiz olarak, yüksek doğrulukta seyrüsefer bilgisi sağlar [23]. 3 boyutlu konum, hız ve zaman gibi bilgiler için GPS kullanıcılarına seyrüsefer mesajı göndermektedir. Her GPS uydusu L1 (Link1) ve L2 (Link2) olarak iki frekansta yayın yaparken, günümüzde GPS sisteminin modernizasyonu ve yenileme çalışmaları kapsamında L1 ve L2 sinyallerine ilave olarak L5 frekansından da sinyal yayımı yapacaktır. L2C, sivil kullanıcıların iyonosfer etkisi gibi hataları belirleme de çift frekans özelliğinden yararlanmasını sağlamak için L2 frekansı üzerinden C/A kodu yayımlaması için ortaya çıkmıştır. L5, gönderilen Block IIF uydularında yayım yapan ve sivil amaçlı yeni bir sinyal olup, frekansı 1176,45 MHz'dir. L5 frekansının hava araçlarının güvenliği amacıyla kullanılması planlanmaktadır [6]. Yakın gelecekte GNSS sistemleri arasında yapılan işbirliği ve uyumluluk programı kapsamında GPS, L1C yayını da yapmaya başlayacaktır.

Her GPS uydusu L1 ve L2 frekansında yayım yapmaktadır. GPS uydusundan gönderilen L1 sinyali 1575,42 MHz, L2 sinyali 1227,6 MHz taşıyıcı frekansını kullanmaktadır. L1 ve L2 frekansını elde etmek için $f_o = 10,23$ MHz temel frekans değeri kullanılır.

$$L1 = 154 \times 10,23 \text{ MHz} = 1575,42 \text{ MHz}$$

$$L2 = 120 \times 10,23 \text{ MHz} = 1227,6 \text{ MHz} \quad (2.1)$$

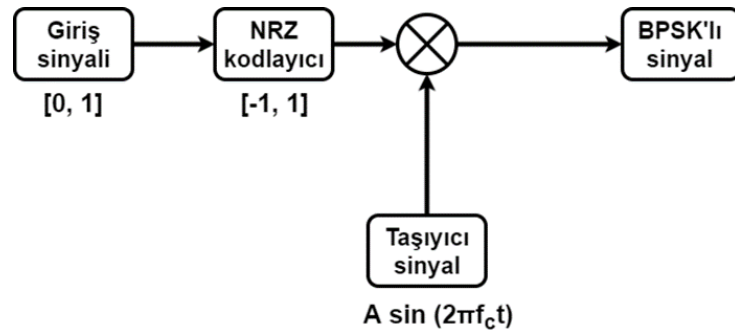
GPS uydularının L1 ve L2 frekansı üzerinden kullanıcılara gönderdiği sinyal, uydulara kimlik olarak atanan ve uyduları birbirinden ayırt etmede kullanılan Sözde Rasgele Gürültü (Pseudo Random Noise, PRN) kodu ve seyrüsefer mesajından oluşmaktadır. PRN kodunun iki önemli özelliği, sinyalin geldiği uyduyu tespit etmek ve uydü ile alıcı arasındaki mesafeden kaynaklı gecikme süresini belirlemektir. PRN kodları, ikili sayı sistemine göre oluşturulmuş 0 ve 1 sayılarından meydana gelmektedir. Geleneksel GPS uydularında iki farklı PRN kodu vardır. Birincisi sivil kullanıcılar için Serbest Edinim (Coarse/Clear Acquisition) veya C/A kodu, diğeri ise askeri uygulamalar için Hassas/Korumalı (Precision veya Protected - code) veya P-kodudur. L1 taşıyıcı frekansı üzerine sözü edilen iki PRN kodu ve seyrüsefer mesajı bindirilmiştir. Seyrüsefer mesajı kullanıcıların konum, zaman, hız gibi hassas bilgileri elde etmesine yardımcı olacak verileri içerir. L2 taşıyıcı frekansı ise yalnızca P - kodu ve seyrüsefer mesajından oluşur [6, 23]. GPS uydularında PRN kodu ve seyrüsefer mesajından oluşan L1 ve L2

sinyalini kullanıcılara iletmek için bir faz modülasyonu türü olan İkili Faz Kaydırmalı Anahtarlama (Binary Phase Shift Keying) tekniği kullanılmıştır. Kısaca BPSK, giriş sinyalinin 0 ve 1 olmasına bağlı olarak taşıyıcı sinyalin fazının 180° değiştiği modülasyon türüdür.

2.1. İkili Faz Kaydırmalı Anahtarlama (Binary Phase Shift Keying, BPSK)

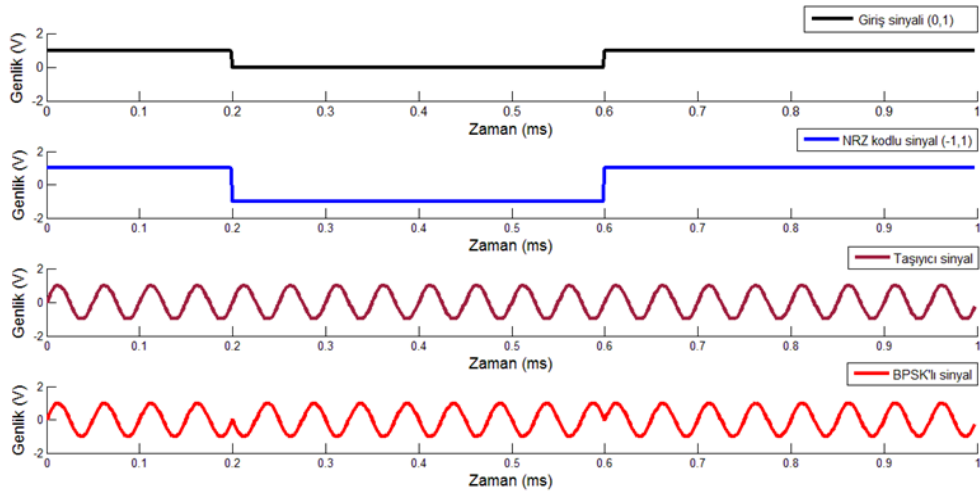
Uyduya dayalı konum belirleme sistemlerinde sinyal iletimi için Genlik Modülasyonu (Amplitude Modulation, AM), Frekans Modülasyonu (Frequency Modulation, FM) gibi analog modülasyon tekniklerinden ziyade İkili Faz Kaydırmalı Anahtarlama (Binary Phase Shift Keying, BPSK), Dörtlü Faz Kaydırmalı Anahtarlama (Quadrature Phase Shift Keying, QPSK), İkili Offset Taşıyıcı (Binary Offset Carrier, BOC) gibi sayısal modülasyon teknikleri kullanılmaktadır. BPSK modülasyonunda her sembolde bir bit gönderilmektedir. Sayısal veri iletiminde sinyaller 0 ve 1 ile ifade edilmektedir. Sayısal olarak temsil edilen sinyaller, taşıyıcı sinyal vasıtası ile boşluğa gönderilmektedir. BPSK modülasyonunda sinyaller 0° ve 180° faz kaydırmalı olarak iletilmektedir [37]. Yani giriş sinyali 0 olduğunda taşıyıcı sinyalin fazı değişmezken, 1 olduğunda taşıyıcı sinyalin fazı 180° değişmektedir.

BPSK sinyalini üretmek için gerekli olan modülatör blok diyagramı Şekil 2.1’de gösterilmektedir. BPSK modülasyonu için 0 ve 1 ikili sayılarının Sıfıra Dönmeyen kodlama (Non-Return-to-Zero Encoding) veya tek kutuptan çift kutba dönüşümü (Unipolar to Bipolar Conversion) ile -1 ve 1 sayılarına çevrilmesi gerekmektedir. Söz konusu dönüşüm gerçekleştiği takdirde taşıyıcı sinyalin fazında değişim gerçekleşir. Blok diyagramda yer alan çarpıcı (mixer) taşıyıcı sinyalin fazını değiştirmek için kullanılmaktadır.



Şekil 2.1. BPSK blok diyagramı

Taşıyıcı sinyalin faz açısı değişimi giriş sinyalinin bit değişimi sırasında gerçekleşir. Eğer sinyalde 1'den 0'a veya 0'dan 1'e bir değişim gerçekleşirse, taşıyıcı sinyalin fazı 180° değişir. Bahsedilen durumun daha net anlaşılması için Şekil 2.2'de örnek bir uygulama ele alınmıştır. Şekilde görüldüğü üzere, girişe ikili kod dizisi (1, 0, 0, 1, 1) sayısal veri uygulanmıştır. İkili kod dizisi NRZ kodlayıcı katında (1, -1, -1, 1, 1) verisine dönüştürülür. Nihai olarak taşıyıcı sinyal, NRZ kodlu veri dizisi ile çarpılır ve BPSK modülasyonu tamamlanır. Dikkat edilirse, BPSK sinyalinin iki noktasında 180° 'lik faz değişimi meydana gelmiştir. Bahsedilen faz değişimleri, giriş sinyalindeki 0'dan 1'e veya 1'den 0'a geçişin olduğu noktalardır.



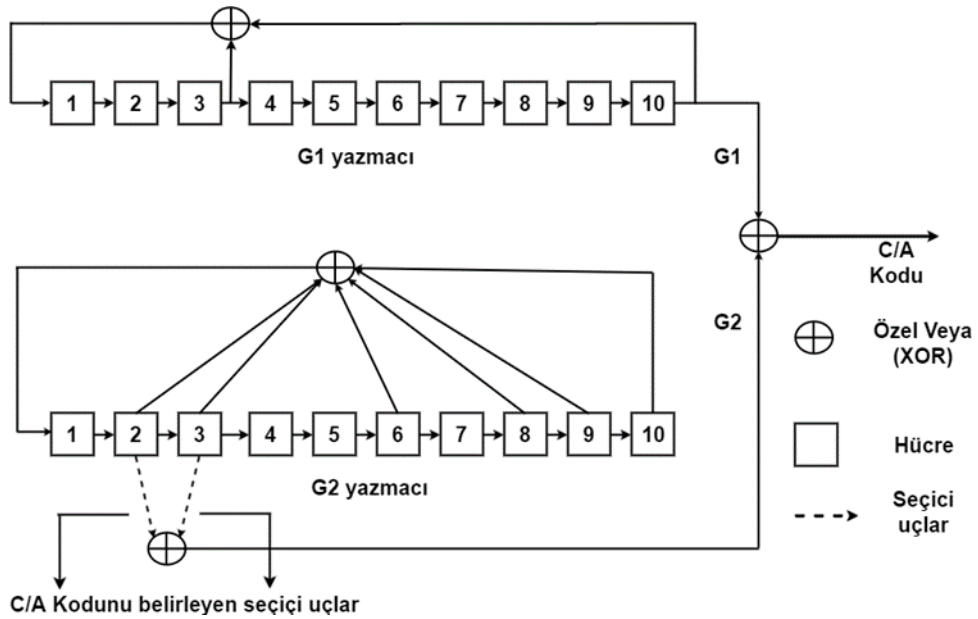
Şekil 2.2. BPSK örneği

2.2. C/A (Coarse/Acquisition) Kodu ve Özellikleri

PRN kodu rasgele bir sayı dizisine benzemekte olup belirli bir algoritma veya kurala göre kaymalı kaydediciler (shift register) kullanılarak üretilmektedir [7]. Uydularda PRN kodları kaymalı kaydedicilerdeki farklı hücrelerden alınan çıkış bitleri ile elde edilmektedir [7]. Farklı hücrelerden alınmasından dolayı her bir PRN kodu birbirinden farklı olmaktadır. Bu yüzden de her bir uydu için farklı bir PRN kodu tahsis edilmektedir. Yani PRN kodunun her bir uydu için birer kimlik niteliğinde olduğu söylenebilir. Uydular için 1'den 37'ye kadar tanımlı toplam 37 adet PRN kodu mevcuttur. Örneğin PRN 11, 11 numaralı uyduya, PRN 2, 2 numaralı uyduya atanmıştır. Bu sayede alıcı, gelen sinyalin hangi uydudan alındığını belirlemektedir. PRN kodları sadece sinyalin alındığı uyduyu belirlemez, bunun yanında alıcı ve uydu arasındaki mesafeyi hesaplamak için kullanılmaktadır. Uydu tarafından gönderilen sinyal alıcı ve verici

arasındaki mesafeden dolayı alıcıya belirli bir gecikme ile ulaşmaktadır. Bu gecikme zamanı, alıcıda PRN kodu ile tespit edildikten sonra ışık hızı ile çarpılarak aradaki mesafe hesaplanmaktadır. C/A kodu 1023 bit uzunlukta ve periyodu 1 milisaniyedir. C/A kodunun periyodu çok kısa seçilmesinin nedeni GPS alıcılarının uydulara en kısa sürede kilitlenmesini sağlamaktır [6]. Periyodu 1 milisaniye olduğu için 1,023 MHz'lik veri aktarım hızına sahiptir. C/A kodunda çip (chip) uzunluğu 1 mikro saniye olduğu için dalga boyu yaklaşık 293 metre ve çözünürlüğü de 3 metredir. C/A kodu, P-kodlu alıcıların daha uzun süreli olan P koduna kilitlenmesi için gerekli olan süreyi azaltmakta da kullanılmaktadır [6].

C/A kodlar, Altın kodlar (Gold codes) olarak bilinen ailenin üyeleridir ve çip adı verilen 0 ve 1 sayılarından meydana gelmektedir ve her 0 ve 1 sayısına "chip" adı verilmektedir [6, 23]. C/A kodu üretimi için G1 ve G2 olmak üzere iki tane kaymalı kaydedici kullanılmaktadır. G1 ve G2 kaydedicilerinde 10 farklı hücre bulunmaktadır. Kaydedicilerde seçilen çıkış hücrelerine bağlı olarak farklı C/A kodları üretilmektedir. G1 ve G2'nin başlangıç şartları ikilik sayı sisteminde 1111111111 olarak verilmektedir. Kaymalı kaydediciler ile C/A kodu üretimini gösteren blok diyagram Şekil 2.3'de verilmiştir.



Şekil 2.3. C/A kodu üretimi [7]

Şekil 2.3’ de görüldüğü gibi G1 kaydedicisindeki 3 ve 10 numaralı hücrelerden elde edilen değerler Özel Veya (Exclusive OR, XOR) kapısından geçirilerek elde edilen çıkış G1 kaydedicisinin 1 numaralı hücre girişine uygulanır. G2 kaydedicisindeki 2, 3, 6, 8, 9 ve 10 numaralı hücrelerden elde edilen değerler için de Özel Veya işlemi uygulanır. Elde edilen sonuç G2 kaydedicisinin 1 numaralı hücre girişine uygulanır. Anlatılan adımlar devamlı olarak tekrarlanır. Şekil 2.3’de verilen blok diyagrama göre G1 ve G2 için polinom fonksiyonu ifade edilmek istenirse sırasıyla Eşitlik 2.2’deki gibi yazılabilir.

$$G_1(x) = 1 + x^3 + x^{10}$$

$$G_2(x) = 1 + x^2 + x^3 + x^6 + x^8 + x^9 + x^{10} \quad (2.2)$$

Yine aynı şekilde, G1 kaydedicisindeki 10 numaralı hücre çıkışı ile G2’den uydu numarasına göre seçilen iki hücrenin XOR kapısında birleştirilmesi sonucu C/A kodu üretilmektedir. Şekilde verilen blok diyagramdaki C/A kodu, seçilen G2 hücrelerine göre 10. uydu için üretilmektedir. G2 kaydedicisinden seçilen hücre numaralarına göre uydulara atanmış olan C/A kodları Tablo 2.1’de sıralanmıştır.

Tablo 2.1. GPS uyduları için C/A kodu üretimi [7, 38]

Uydu Numarası	PRN Numarası	G2 Seçici Uçlar	İlk 10 C/A Kodu (Octal)
1	1	$2 \oplus 6$	1440
2	2	$3 \oplus 7$	1620
3	3	$4 \oplus 8$	1710
4	4	$5 \oplus 9$	1744
5	5	$1 \oplus 9$	1133
6	6	$2 \oplus 10$	1455
7	7	$1 \oplus 8$	1131
8	8	$2 \oplus 9$	1454
9	9	$3 \oplus 10$	1626
10	10	$2 \oplus 3$	1504
11	11	$3 \oplus 4$	1642
12	12	$5 \oplus 6$	1750
13	13	$6 \oplus 7$	1764
14	14	$7 \oplus 8$	1772
15	15	$8 \oplus 9$	1775
16	16	$9 \oplus 10$	1776
17	17	$1 \oplus 4$	1156
18	18	$2 \oplus 5$	1467
19	19	$3 \oplus 6$	1633
20	20	$4 \oplus 7$	1715
21	21	$5 \oplus 8$	1746

Tablo 2.1. (Devam) *GPS uyduları için C/A kodu üretimi [7, 38]*

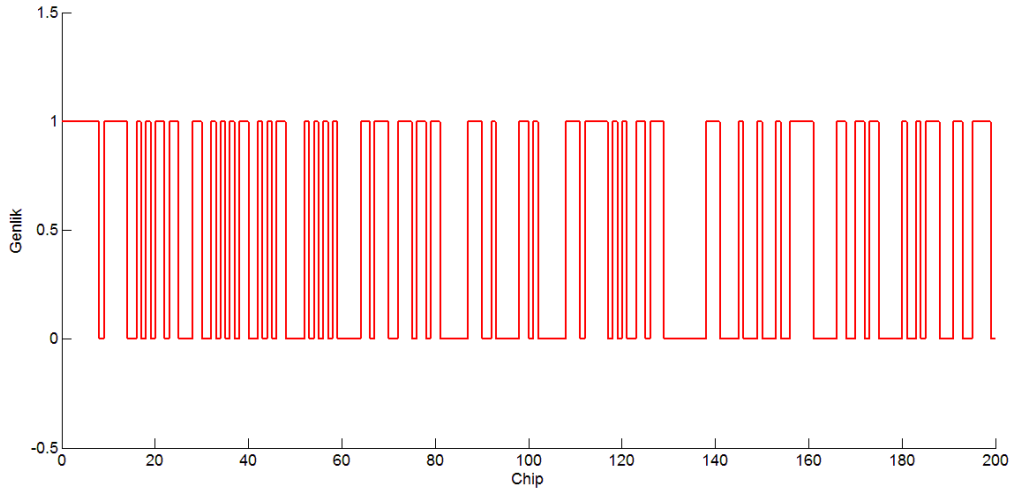
22	22	$6 \oplus 9$	1763
23	23	$1 \oplus 3$	1063
24	24	$4 \oplus 6$	1706
25	25	$5 \oplus 7$	1743
26	26	$6 \oplus 8$	1761
27	27	$7 \oplus 9$	1770
28	28	$8 \oplus 10$	1774
29	29	$1 \oplus 6$	1127
30	30	$2 \oplus 7$	1453
31	31	$3 \oplus 8$	1625
32	32	$4 \oplus 9$	1712
**	33	$5 \oplus 10$	1745
**	34*	$4 \oplus 10$	1713
**	35	$1 \oplus 7$	1134
**	36	$2 \oplus 8$	1456
**	37*	$4 \oplus 10$	1713

*34 ve 37 aynı C/A koduna sahiptir.

**GPS uyduları C/A kodlarını kullanmamaktadır. Bunlar rezervedir.

Tablo 2.1'in ilk sütunu uydu numaralarını, ikinci sütunu uydu numarasına karşılık gelen PRN numarasını, üçüncü sütun uydu numarasına göre G2 kaydedicisinde seçilecek hücre numaralarını, dördüncü sütun ise ilgili C/A kodunun ilk 10 bitinin sekizlik (octal) tabandaki karşılığını göstermektedir. Örneğin 22. uydunun C/A kodu üretilmek istenirse, G2 çıkışı 6. ve 9. hücrelerden alınmalıdır ve üretilen ilk 10 bitin değeri ise sekizlik tabanda 1763 olmalıdır. Eğer üretilen C/A kodunun ilk 10 biti bahsedilen değer ile eşleşmez ise, üretilen kod hatalıdır [7]. Örnek olması açısından 15. uydu için 1023 bit uzunluklu C/A kodu Tablo 2.1 referans alınarak MATLAB ortamında üretilmiştir ve ilk 200 çip Şekil 2.4'de gösterilmiştir

C/A kodlarının en önemli özelliklerinden birisi oto korelasyon (auto correlation) değerinin yüksek, çapraz korelasyon (cross correlation) değerinin düşük olmasıdır [5]. Korelasyon iki değişken arasındaki ilişkiyi göstermektedir. Bu iki değişken arasındaki ilişki derecesi de korelasyon katsayısı ile ifade edilir. Korelasyon katsayısı -1 ile 1 arasında değer almaktadır. Korelasyon katsayısının işareti pozitif ise değişkenler arasında pozitif ilişki, korelasyon katsayısının işaretinin negatif olması durumunda değişkenler arasında negatif bir ilişki olduğu ifade edilir. Bu değer 0 olması durumunda da iki değişken arasında herhangi bir ilişkinin bulunmadığı söylenilebilir [39].

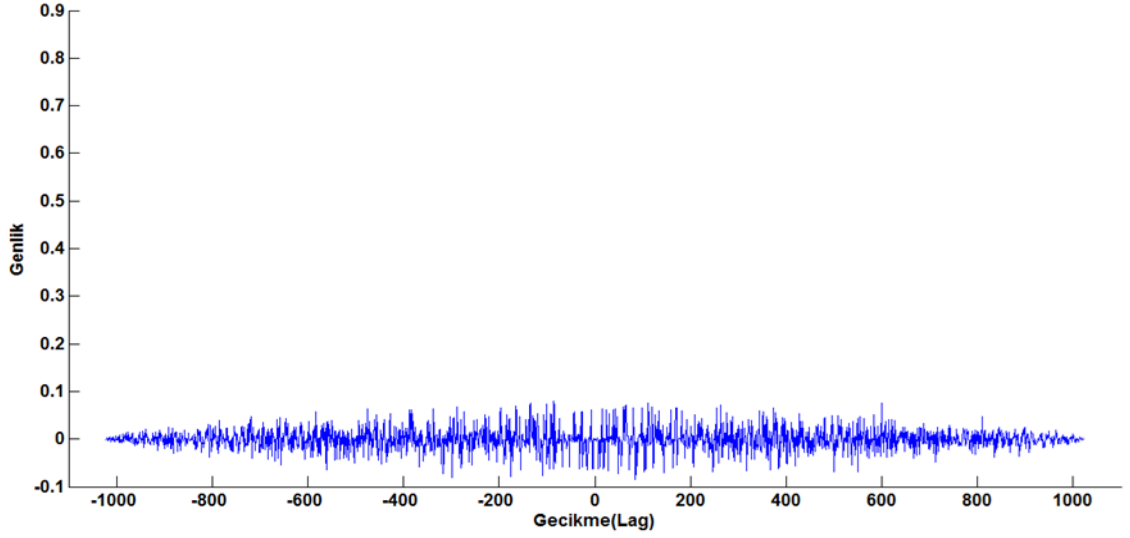


Şekil 2.4. PRN15 için üretilen ilk 200 çip

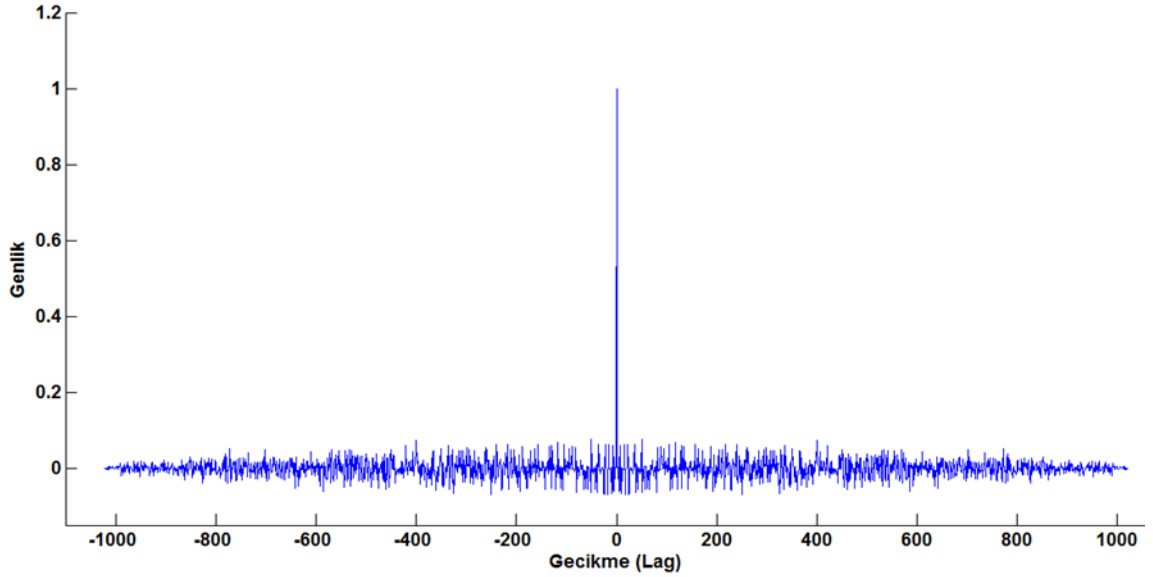
Oto korelasyon, bir değişkenin kendisi ile belirli bir miktar geciktirilmiş hali arasındaki benzerliği gösteren bir ölçüt iken, çapraz korelasyon ise bir değişken ile başka bir değişkenin bir miktar geciktirilmiş hali arasındaki benzerliğin ölçütüdür. C/A kodunun korelasyon özelliği sinyal yakalamada büyük bir avantaj sağlamaktadır [7].

Gelen sinyal tüm C/A kodları ile korelasyon işlemine tabi tutulur ve yüksek korelasyon değerli olan kod, uydunun C/A kodu olarak belirlenir. Teorik olarak iki farklı kod dizisi ideal olarak birbirine dik (orthogonal) ise çapraz korelasyon değeri sıfır olacaktır. Altın kodlar tam olarak birbirine dik değildir bu yüzden de C/A kodlarının çapraz korelasyon değeri sıfır değildir ancak çok düşüktür. Şekil 2.5’ de 21 ve 8. uydulara ait C/A kodlarının çapraz korelasyon sonucunu, Şekil 2.6’da ise 21. uyduya ait oto

korelasyon sonucu gösterilmiştir. Şekil 2.5’ den de görüldüğü üzere 21 ve 8. uyduların C/A kodlarının oto korelasyon değerleri sıfır değil ancak çok düşük değerlerdedir ve bu iki C/A kodu arasında korelasyonun bulunmadığı anlamına gelmektedir. Şekil 2.6’ya göre de 21. uydunun kendi ile olan oto korelasyon değerinin çok yüksek olduğu görülmektedir.



Şekil 2.5. PRN21 ve PRN5 çapraz korelasyon sonucu



Şekil 2.6. PRN21 oto korelasyon sonucu

2.3. P-Kodu (Precision/Protected- Code) ve Özellikleri

P- kodu askeri kullanıcılara koordinat, zaman, konum gibi bilgileri yüksek doğrulukta L1 ve L2 frekansı ile göndermek için kullanılmaktadır. Sivil kullanıcıların erişimi engellenmiştir. 10,23 MHz temel frekans değeri ile üretilen P kodu veri hızı 10, 23 Mcps, dalga boyu yaklaşık 29,3 metre ve çözünürlüğü ise 30 cm mertebelerindedir. P-kodu elektronik karıştırma (jamming) ve aldatmaya (spoofing) karşı korumak için A-S (anti-Spoofing) özelliği kullanılarak kriptolanmıştır. Kripto kodu olarak veri hızı 10,23 Mcps' dan daha düşük olan W-kod kullanılmaktadır ve kriptolanan P- koduna Y Kodu

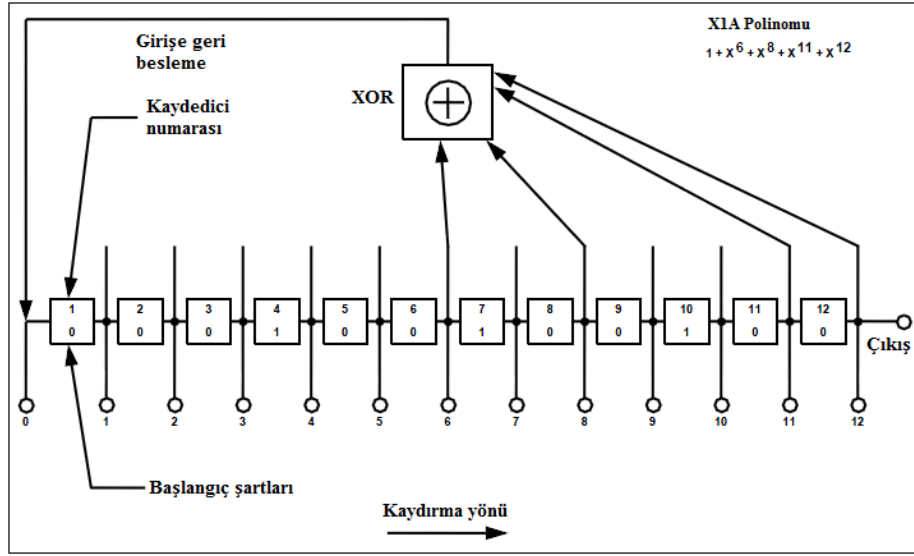
veya P(Y) kodu denilmektedir. P (Y) kod kullanımı sadece yetkili (askeri ve sivil güvenlik) kullanıcılara açık hale gelmiştir [6, 40].

P- kodunu üretmek için C/A kodunda olduğu gibi yine kaymalı kaydedicilerden faydalanılmaktadır. P-kodu X1A, X1B, X2A ve X2B olmak üzere 12 bitlik 4 kaymalı kaydediciden üretilmektedir. Bunun sonucunda 235471014 chip uzunluklu kod dizisi elde edilir. P- kodunun veri aktarım hızının 10.23 Mcps olduğu düşünüldüğünde bu kod dizisinin uzunluğu 266,41 gün veya 38.058 haftaya denk gelmektedir [40]. Kod uzunluğunun tamamı birer haftalık toplam 37 haftaya bölünmüştür ve her bir uyduya 37 haftalık kod uzunluğunun bir haftalık bölümü tahsis edilmiştir. Kodlar her GPS haftası başında (Cumartesi gece yarısı) tekrar etmektedir. Eğer P kodu her hafta sıfırlanmasa idi, toplam süresi yaklaşık 37 hafta olacaktı. 37 haftalık periyot birer haftalık periyotlara ayrılarak toplam 37 tane farklı P- kodu elde edilmiştir [6, 41].

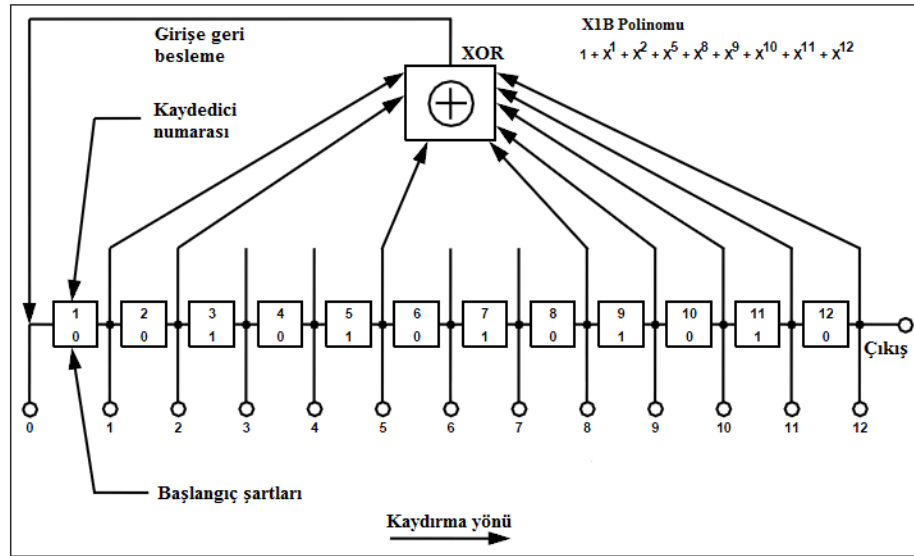
P-kodunu üretmek için kullanılan X1A, X1B, X2A ve X2B kaymalı kaydedicilerine ait blok diyagramlar sırasıyla Şekil 2.7, 2.8, 2.9 ve 2.10’ da gösterilmektedir. X1A’nın blok diyagramına bakıldığında ilgili kaydedicilerden alınan çıkışlara göre polinom fonksiyonu $1 + x^6 + x^8 + x^{11} + x^{12}$ olarak yazılabilir. Bu fonksiyonun anlamı, 6, 8, 11 ve 12. kaydedici çıkışlarından elde edilen bitler Özel Veya (Exclusive-OR) kapısı işlemine tabi tutulur ve elde edilen sonuç geri besleme ile 1 numaralı kaydedicinin girişine uygulanır. Bu işlem sürekli olarak devam eder. X1A, X1B, X2A ve X2B’ ye ait polinom fonksiyonları ve başlangıç şartları Tablo 2.2’ de verilmiştir. Yeni GPS haftasının başı veya mevcut GPS haftasının sonunda kaydedicilerin tamamı eş zamanlı olarak başlangıç şartına dönmektedir.

Tablo 2.2. P-kodu kaydedicilerinin polinom fonksiyonu ve başlangıç şartları [41, 42]

Kaydedici isimleri	Polinom fonksiyonu	Başlangıç şartları
X1A	$1 + x^6 + x^8 + x^{11} + x^{12}$	001001001000
X1B	$1 + x^1 + x^2 + x^5 + x^8 + x^9 + x^{10} + x^{11} + x^{12}$	010101010100
X2A	$1 + x^1 + x^3 + x^4 + x^5 + x^7 + x^8 + x^9 + x^{10} + x^{11} + x^{12}$	100100100101
X2B	$1 + x^2 + x^3 + x^4 + x^8 + x^9 + x^{12}$	010101010100



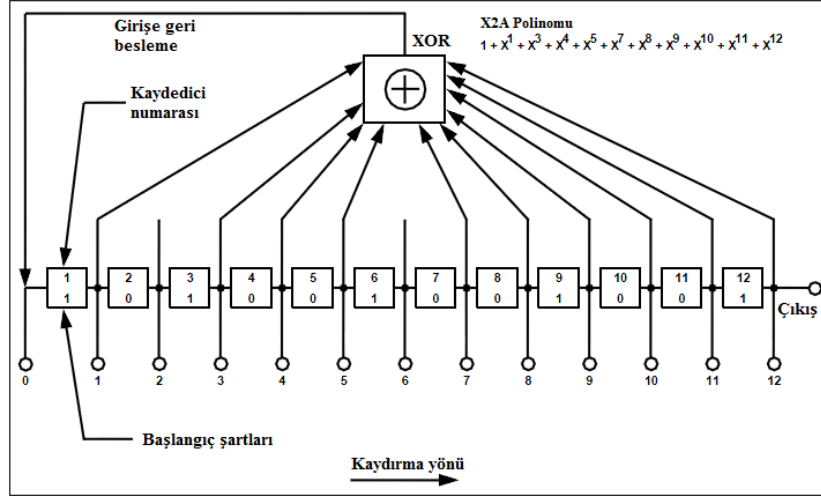
Şekil 2.7. X1A kaydedicisinin iç yapısı [41, 42]



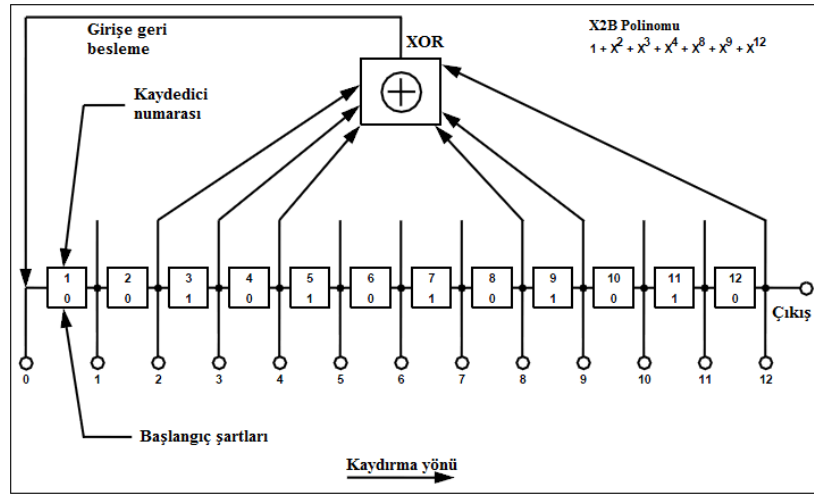
Şekil 2.8. X1B kaydedicisinin iç yapısı [41, 42]

P- kodunun X1 ve X2 kaydedicileri ile üretilmesini gösteren blok diyagram Şekil 2.11' de verilmiştir. X1 kaydedicisi, X1A ve X1B kaydedicilerine XOR kapısı işlemi uygulandıktan sonra elde edilmektedir. Aynı şekilde X2 kaydedicisi de, X2A ve X2B kaydedicilerine XOR kapısı işlemi uygulandıktan sonra elde edilmektedir. P-kodu üretilmeden hemen önce X2 kaydedicisine, gecikme bloğu ile uydu numarasına bağlı olarak chip gecikmesi verilerek $X2_i$ elde edilir. X1 ve $X2_i$ kaydedicilerine tekrar XOR kapısı işlemi uygulandıktan sonra nihai olarak P-kodu üretilmiş olur [14, 41]. Burada yer alan i , 1' den 37' e kadar olup uydu numarasına bağlı olarak verilecek chip gecikmesini

temsil etmektedir. Yani eğer 3. uydu P- kodu üretecek ise, X2 kaydedicisine 3 çip gecikme verildikten sonra X1 kaydedicisi ile XOR işlemi gerçekleştirilir.



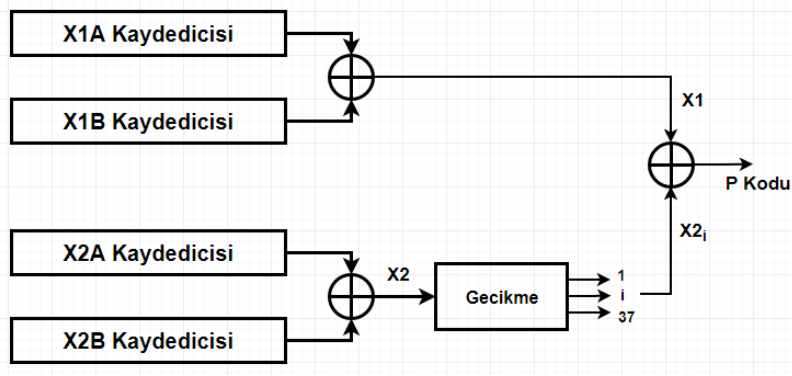
Şekil 2.9. X2A kaydedicisinin iç yapısı [41, 42]



Şekil 2.10. X2B kaydedicisinin iç yapısı [41, 42]

2.4. Seyrüsefer Mesajı

Daha önceki bölümde de bahsedildiği gibi C/A ve P-kodu uyduları belirlemek için kullanılan veri dizisi olarak tanımlanabilir. Bu kodlarla birlikte alıcı için hayati öneme sahip seyrüsefer mesajı iletilmektedir. Bu mesaj kullanıcıya görünür uyduların hassas konumu ve iletilen sinyalin zamanını hesaplamak için önemli bilgiler sağlamaktadır [14]. Aynı zamanda mesafe ölçüm hatasını etkileyen birçok hata ve saat hatası gibi düzeltmeler için kullanılır [14].



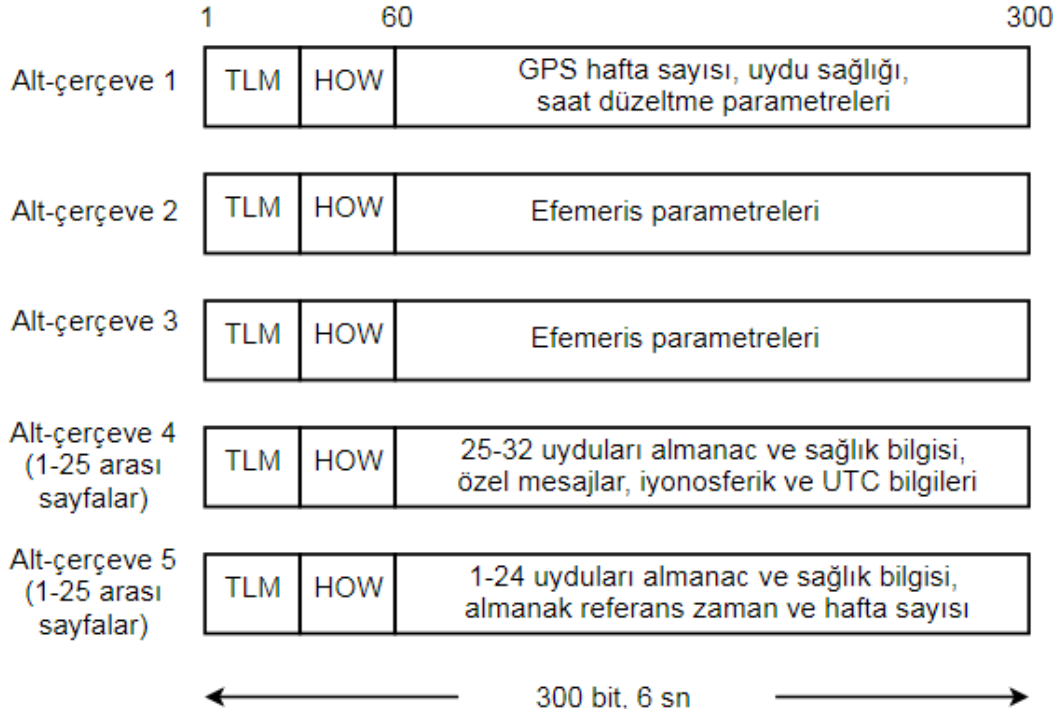
Şekil 2.11. P- kodu üretimi[14]

50 bps veri hızına sahip seyrüsefer mesajı 1500 bit uzunluğa sahip çerçevelerden (frame) oluşmaktadır. Bir çerçevenin iletim süresi 30 saniyedir. Her bir çerçeve beş alt-çerçeveye (sub-frame) ayrılmıştır ve her biri 300 bit uzunluğunda ve süresi 6 saniyedir [43]. Her bir alt-çerçeve 30 bit uzunlukta on kelimeden (word) oluşmaktadır. Alt-çerçeve 1, 2 ve 3 her çerçevede sürekli olarak tekrar eder. Fakat alt-çerçeve 4 ve 5’de aynı yapıya sahip farklı veriler içeren toplam 25 sayfadan (page) oluşmaktadır [43]. Seyrüsefer mesajının tamamının alınması için gerekli süre 12,5 dakikadır [43]. Şekil 2.12’ de çerçeve, alt-çerçeve ve kelimelerin içyapısı gösterilmektedir.

Her alt-çerçevenin ilk iki kelimesi Telemetry (TLM) ve Handover Word (HOW) ile başlar. İlk önce TLM daha sonra HOW iletilmektedir [38]. TLM 30 bit uzunluğundadır ve 8 bitlik 10001011 başlangıç (preamble) dizisi, 16 rezerve biti ve 6 bitlik parite (parity) bitinden oluşmaktadır. Her TLM 8 bitlik başlangıç (preamble) dizisi ile başlar. Parite bitleri alınan veri dizisinde meydana gelen hataları belirlemek ve düzeltmek için kullanılmaktadır. Başlangıç (Preamble) dizisi alt-çerçeve başlangıcını tespit etmek, senkronizasyonu sağlamak ve seyrüsefer mesajını eşleştirmek için kullanılır [7, 43].

30 bit uzunluklu HOW, her alt-çerçeve veya sayfanın ikinci kelimesidir ve 6 saniyede bir tekrarlanır. HOW, 17 bitlik haftanın zamanı (time-of-week, TOW) ile başlar ve 3 bitlik alt-çerçeve kimliği (sub-frame ID), 6 bitlik parite biti ile sonlanır [38, 41, 42]. TOW bilgisi alıcılar için hayati öneme sahiptir. Uydunun gönderdiği sinyal kullanıcılara aradaki mesafeden dolayı belirli bir gecikme ile ulaşır. TOW bilgisinde uydunun gönderdiği sinyalin çıkış zamanı burada yer almaktadır. Alıcı HOW bilgisini çözdükten sonra gelen sinyalin uydudan çıkış zamanını belirler ve çıkış zamanı ile o andaki kendi dahili saati arasındaki zaman farkına bakar. Aradaki fark uydu ile alıcı arasındaki gecikme

süresini verir. Uydu sinyallerinin boşlukta ışık hızı ile ilerlediği düşünülduğünde bu sürenin belirlenmesi, aynı zamanda alıcı verici arası mesafenin de belirlenmesi anlamına gelmektedir.



Şekil 2.12. Seyrüsefer mesaj yapısı [38, 41-43]

Alt-çerçeve kimliği, alt-çerçevenin numarasını vermektedir. Bu veriler alt-çerçeveleri tespit etmek için kullanılmaktadır. Örneğin alt-çerçeve 3 için, alt-çerçeve kimliği 011 olarak belirlenmiştir [38]. Tablo 2.3’de alt-çerçeve numaralarına karşılık gelen kodlar sıralanmıştır.

Tablo 2.3. Alt-çerçeve kimlik kodu [38, 41, 42]

Alt-çerçeve (Subframe)	Kimlik Kodu
1	001
2	010
3	011
4	100
5	101

Alıcıların konumlarını belirlemesi için hata kaynaklarından dolayı meydana gelen düzeltmeler ve uyduların koordinat bilgileri gibi hayati öneme sahip parametreler alt-

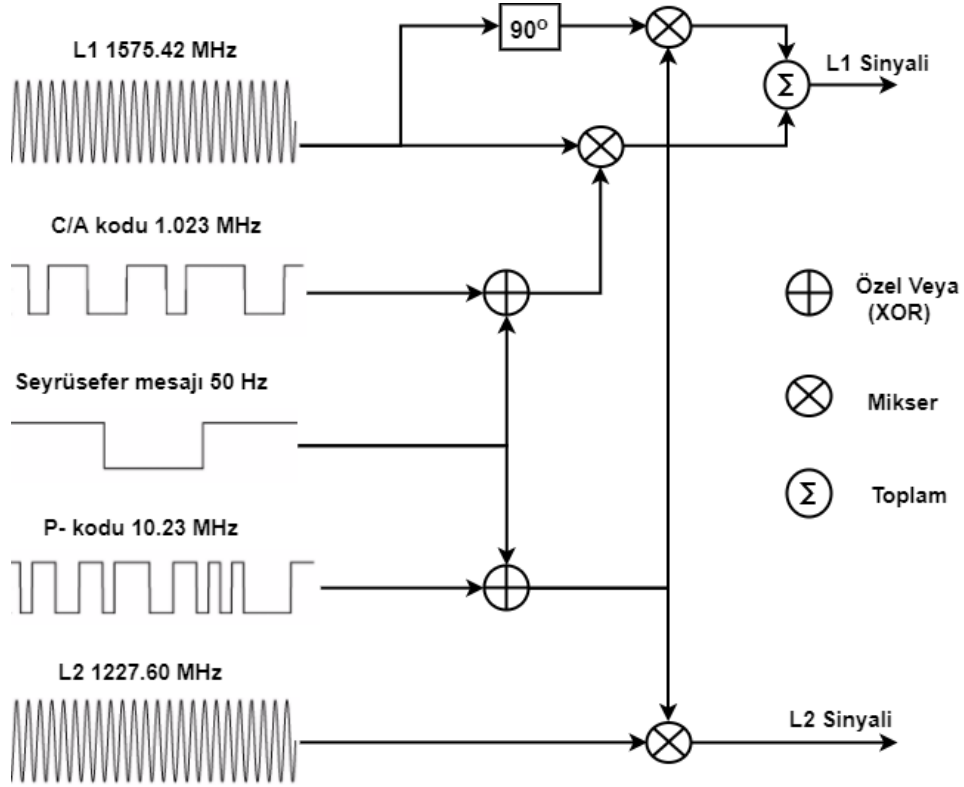
çerçeveler ile gönderilmektedir. Parametrelerin gönderilmesi tek bir alt-çerçeve ile mümkün olmamaktadır. Bu yüzden her bir alt-çerçevede farklı bir bilgi gönderilmektedir. Alt-çerçeve 1’de genel olarak gönderilen parametreler Uydu Saat ve Sağlık Bilgisi (Satellite Clock and Health Data) verisidir. Bu veri altında Hafta Numarası (Week No), Uydu Doğruluğu (Satellite Accuracy), Uydu Sağlığı (Satellite Health), Uydu Saat Düzeltme Parametreleri (Satellite Clock Correction Parameters) gibi bilgiler gönderilir [38, 41, 42]. Özellikle uydu saat düzeltme parametreleri hassas mesafe hesabı için oldukça önemlidir [14]. Alt-çerçeve 2 ve 3’de genel olarak Uydu Efemeris (Satellite Ephemeris) parametreleri gönderilmektedir. Efemeris bilgisini bir uydunun yörüngedeki pozisyonunu hassas bir şekilde belirlemek için gereken, her uydunun kendi hassas koordinatlarını içeren ve geçerliliği birkaç saate kadar olan veriler olarak tanımlamak mümkündür. Alt-çerçeve 4’ün içerdiği bilgiler, 25 – 32 arasındaki uydular için almanak (almanac) ve sağlık bilgisi (health information), iyonosferik hata verisi (ionospheric data) ve Eşgüdümlü Evrensel Zaman (Coordinated Universal Time, UTC) verileri ile uydu konfigürasyonlarını (Satellite Configuration for 32 satellite) içermektedir. Almanak bilgisi, efemeris bilgisinin aksine birkaç aya kadar geçerlidir ve tüm uyduların hassas olmayan yaklaşık koordinat bilgilerini ihtiva eder. Kısacası, almanak bilgisi her uyduya ait hassas olmayan koordinat bilgilerini içerirken, efemeris bilgisi uydunun doğruluğu çok yüksek olan kendi koordinat bilgilerini içermektedir [43]. Alt-çerçeve 5, 1-24 arası uydular için almanak ve sağlık bilgisi, almanack referans zamanı (almanac reference time) ve almanack referans hafta sayısı (Almanac reference week number) bilgilerini içerir [38, 41, 42].

2.5. GPS Sinyal Üretimi

GPS L1 sinyali C/A, P-kod ve seyrüsefer mesajından oluşurken, L2 sinyali sadece P-kod ve seyrüsefer mesajından oluşmaktadır. GPS uydularından gönderilen L1 ve L2 sinyallerini GPS uyduları kullanıcılara göndermek için BPSK modülasyonu kullanılmaktadır. GPS sinyalinin üretimini gösteren blok diyagram Şekil 2.13’de gösterilmiştir.

Şekil 2.13’ de görüldüğü üzere, öncelikle C/A kodu ve 50 bps hıza sahip seyrüsefer mesajına XOR işlemi uygulanır. Elde edilen sinyal BPSK tekniği kullanılarak 1575,42 MHz frekanslı taşıyıcı sinyal üzerine bindirilir. P-kodu ve seyrüsefer mesajından oluşan sinyal hem 1575,42 hem de 1227,60 MHz frekanslı taşıyıcı sinyal üzerine

bindirilir. 1575,42 MHz'lik L1 sinyaline P- kodlu seyrüsefer mesajı 90° faz farklı olarak eklenir. Faz farkının amacı, C/A ve P-kodlu sinyallerin birbirine karışmasını engellemektir. Sonuç olarak L1 sinyali hem C/A kodlu hem de 90° faz farklı P- kodlu seyrüsefer mesajından oluşurken, L2 sinyali sadece P- kodlu seyrüsefer mesajını içerir.

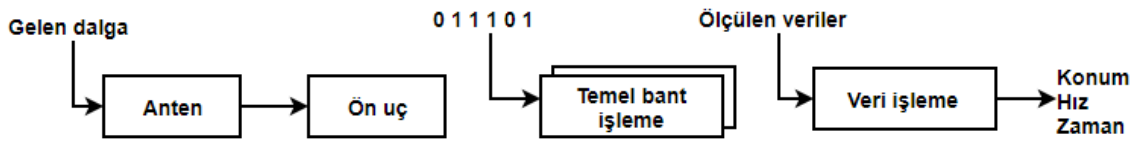


Şekil 2.13. GPS sinyal üretimi

3. GPS ALICI YAPISI

GPS alıcısı, GPS uydularının yayınladığı sinyali alıp kullanıcının konum, hız ve zaman (PVT) bilgilerini belirlemek için kullanılan aygıtlara verilen genel bir isimdir. Geçmişte GPS alıcıları sadece donanımsal (hardware) olarak üretilirken, günümüzde yazılım (software) tabanlı alıcıların üretimi de her geçen gün artmaktadır. Yazılım tabanlı alıcılar, USB portlu aygıtlar gibi bilgisayara takılarak uydulardan gelen sinyalleri işledikten sonra kullanıcının hız, zaman ve konumunu belirlemektedir. Geleneksel GPS alıcılarında yakalama (acquisition) ve izleme (tracking) gibi aşamalarda sinyal işleme donanımsal olarak gerçekleştirilirken, yazılım tabanlı alıcılarda sinyal genellikle, analog / dijital dönüştürücü ile sayısallaştırıldıktan sonra temel bant işlemleri (baseband processing) yazılımsal olarak gerçekleştirilir [5]. Yazılım tabanlı alıcılarda ön uç kısmı hariç geri kalan işlemler yazılımla yapılmaktadır. Bu yüzden de donanım elemanları üzerinde etkili olan sıcaklık gibi olumsuz faktörlerden etkilenmemektedirler. Ayrıca alınan sinyallerin dijital ortamda test edilme ve değerlendirilme esnekliğine sahip olması yeni ve farklı algoritmalar geliştirip uygulama imkanı sağlamaktadır [5].

Alıcıda, uydulardan gelen sinyali kullanıcı açısından faydalı bilgiye dönüştürme işlemi çeşitli basamaklarından oluşmaktadır. Bu basamaklar genellikle bütün alıcı yapılarında aynı olmasına rağmen tasarım kriterlerine göre bazı farklılıklar içerebilir. Şekil 3.1, genel bir GNSS alıcı yapısını göstermektedir.

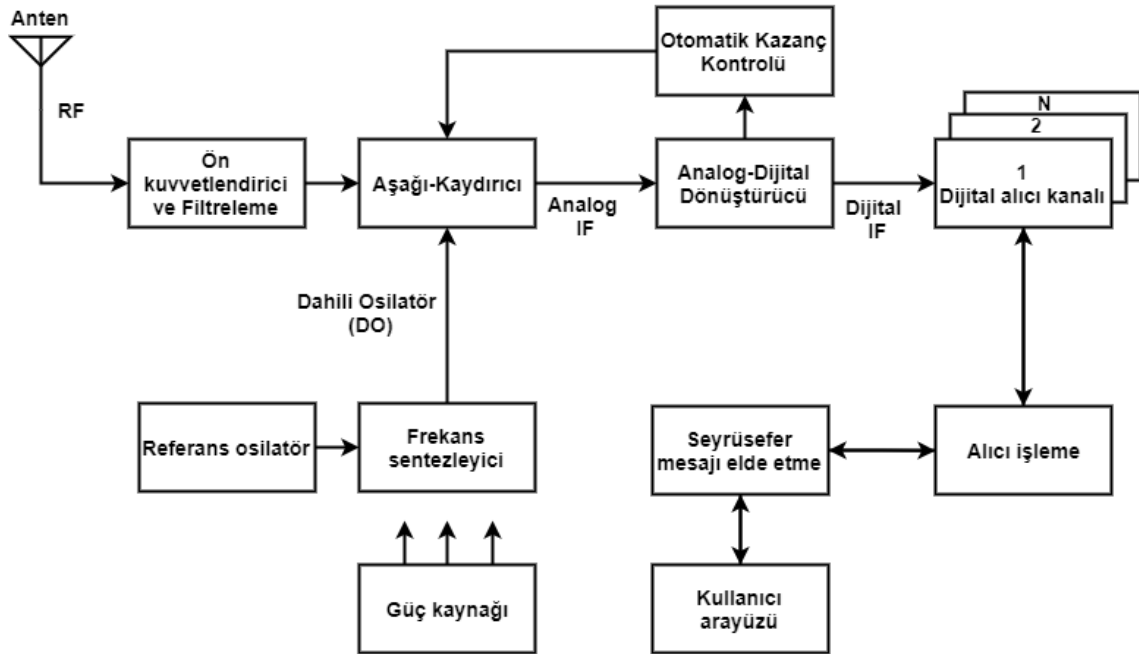


Şekil 3.1. Genel GNSS alıcı yapısı [44]

Uydulardan gelen elektromanyetik sinyal öncelikle antenler ile alınıp elektrik sinyaline dönüştürülür. Ön uç (Front End) bölümüne gelmeden önce sinyale ön filtre ve kuvvetlendirme işlemleri uygulanır ve daha sonra sinyale ön uç bölümünde kuvvetlendirme, filtreleme, aşağı kaydırma (down - converter) ve sayısallaştırma işlemleri uygulanır. Temel bant işleme bölümünde (baseband processing) yakalama (acquisition), izleme (tracking) ve korelasyon (correlation) gibi alıcı için son derece önemli işlemler gerçekleştirilir. Veri işleme bölümü (application processing), temel bant

işlemden sonra elde edilen sinyalin kullanıcının konum, zaman ve hız gibi bilgilerini hesaplayıp çeşitli amaçlar için kullandığı bölümü oluşturmaktadır [44].

GNSS alıcı yapısı genel olarak anten, ön uç, temel bant işleme ve uygulama bölümlerinden oluşmaktadır. Ancak uydu sistemlerinin farklı sinyal yapılarına sahip olması nedeniyle her uydu sisteminin alıcı yapısı birbirine göre çeşitli farklılıklar göstermektedir. Bu çalışma GPS uydularının simülasyonu çerçevesinde gerçekleştirildiği için genel olarak GPS alıcı yapılarının üzerinde durulmuştur. Şekil 3.2, bir GPS alıcısının uydudan gelen bir sinyali kullanıcıya konum, hız ve zaman gibi faydalı bilgiye dönüştürme aşamalarını göstermektedir. Bu aşamalar, anten (antenna), ön kuvvetlendirici (preamplifier), aşağı kaydırıcı (down-converter), analog dijital dönüştürücü (ADC), otomatik kazanç kontrolü (AGC), dijital alıcı kanalı, alıcı işleme, seyrüsefer mesajı elde etme ve kullanıcı arayüzü olarak sıralanabilir. Güç kaynağı alıcı için gerekli olan çalışma gerilimini sağlarken, frekans sentezleyici referans osilatör değerini kullanarak aşağı kaydırma işlemi için gerekli olan dahili osilatör ile ihtiyaç duyulan frekansı düzenlemektedir [14].



Şekil 3.2. Genel GPS alıcı yapısı [14]

3.1. Anten

Antenler, elektromanyetik dalgayı elektrik sinyaline veya elektrik sinyalini elektro manyetik dalgaya dönüştürmede kullanılan elemanlardır [43, 45]. Alıcı ve verici anteni olmak üzere iki grupta incelenebilir. Verici anten, sistemden aldığı güç sayesinde çevresine elektromanyetik dalga yayar. Çevresinde var olan elektromanyetik dalgaları alarak sisteme güç aktaran antene de alıcı anteni adı verilmektedir. Boşluktaki elektromanyetik dalga gelen sinyal ile gürültüden oluşmaktadır. Alıcı anteni, bu elektromanyetik dalgayı alıcı girişlerindeki uçlara elektrik akım ve gerilim değeri olarak dönüştürmektedir [45].

Antenlerde, gelen elektromanyetik dalganın geliş yönü, frekansı ve polarizasyonu oldukça önemlidir. Yatay, dikey ve dairesel olmak üzere üç farklı polarizasyon vardır. Eğer alıcı anteni gönderilen sinyalle aynı yön, frekans ve polarizasyona sahip değilse bu sinyal alınamaz veya çok zayıf olarak alınır. Bu açıdan düşünüldüğünde antenin üç boyutlu bir filtre olduğu söylenebilir [45].

GPS sinyalleri dairesel polarizasyonlu olup, sağ el polarizasyona (Right Hand Circular Polarization, RHCP) sahip antenler ile alınabilmektedir. Bir GPS alıcısının performansı, alıcıya ulaşan sinyalin gücü, kayıpsız olması ve istenmeyen sinyallerden arınmış olmasına bağlıdır. GPS antenleri, GPS sinyalleri için uygun frekanstaki ve bant genişliğindeki sinyalleri alıp bunun dışındaki sinyalleri bastırabilmelidir [23]. Bir GPS anteni, maksimum sayıda sinyal alabilmek için geniş bir alan açısını kapsamalıdır [7]. Antenler, çoklu yol (multipath) etkisini en aza indirmelidir. Çünkü bu etki kullanıcının konumunu belirleme de çok büyük hatalara sebep olabilmektedir [7]. Bütün bu özelliklerin dışında GPS antenleri kazançlarına göre aktif ve pasif olmak üzere iki çeşittir. Pasif antenler aldığı sinyale herhangi bir kazanç katmadan iletirken, aktif antenler sinyalleri kuvvetlendirmektedir. GPS anteni ile alıcı arasında birkaç metreden fazla mesafe olduğu durumlarda, kablodan kaynaklı kayıpları telafi edebilmek için aktif anten kullanılmalıdır. Ancak, aktif antenler pasif antenlerin aksine çalışmak için güç beslemesine ihtiyaç duyarlar [23].

3.2. Ön Uç

3.2.1. Kuvvetlendirici ve filtre

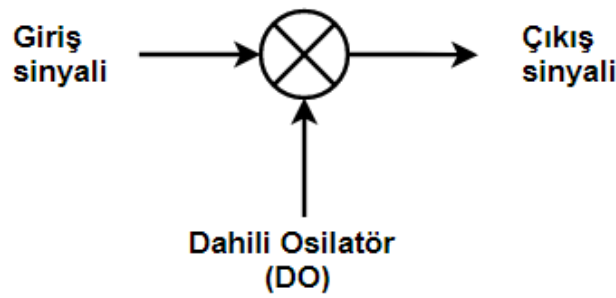
Antenden sonra gelen ilk eleman ya filtre ya da kuvvetlendiricidir. Filtre yalnızca belirli frekansların geçmesini ve diğer frekans değerlerinin zayıflatılmasını sağlamaktadır. Alçak geçiren, yüksek geçiren, bant geçiren ve bant durduran olmak üzere farklı filtre çeşitleri mevcuttur [43]. Direnç (R), kapasitör (C) ve indüktörden (L) oluşan filtreler pasif filtre iken transistör ile işlemsel kuvvetlendiricilerden (Operational Amplifier) oluşan filtreler aktif filtre olarak isimlendirilmektedir. Aktif filtreler, filtreleme işlemini yapabilmek için güce gereksinim duyarken, pasif filtreler herhangi bir güce ihtiyaç duymazlar. Filtre tasarlamadaki ana amaç, istenen veya istenmeyen frekans aralığında keskin geçişler sağlamak ve minimum geçiş kaybı (insertion loss) sağlamaktır. Filtreler, geçiş kaybı ve bant genişliği ile karakterize edilir [43]. Filtrelerde zayıflama veya geçiş kaybı, herhangi bir sinyalin filtreleme yapıldıktan sonra gücünde meydana gelen azalma miktarı olarak tanımlanabilir. Geçiş kaybı, idealde sıfırdır ancak uygulamalarda bu değer sıfır olmamaktadır. Filtrenin geçiş kaybı ne kadar düşük ise filtrenin o kadar iyi olduğu söylenebilir. Bant genişliği filtrelerde diğer önemli bir parametredir. Bant genişliği, bir veri iletişim ortamının ya da haberleşme kanalının kapasitesini ifade etmek için kullanılır [43]. Diğer bir deyişle, bir kanal üzerinde taşınabilecek maksimum frekansa sahip sinyal kanalının bant aralığıdır. Uydulardan gelen sinyalin frekansı Doppler etkisi gibi faktörlerden dolayı nominal değerden farklılık göstermektedir. Filtrenin bant genişliği gelen sinyalin frekansındaki değişimlere uygun olarak ayarlanmış ise bu durumda gelen sinyal sağlıklı bir şekilde alınıp filtrelenebilir.

Kuvvetlendiriciler, herhangi bir sinyalin büyüklüğünü artırmada kullanılan aktif elemanlar olup filtrelerin aksine işlevini yerine getirmesi için güce ihtiyaç duymaktadır. İdeal kuvvetlendiriciler sadece sinyalin genliğini artırırken, pratikte kullanılan kuvvetlendiriciler bunun yanında sinyale belirli oranda bir gürültü seviyesi ekler. Bu gürültü kuvvetlendiricinin sahip olduğu elektronik elemanlar ve termal etkinin sonucu oluşan gürültüdür. Kuvvetlendirici tasarımında amaç, gelen sinyalin genliğini artırmak ve minimum seviyede gürültü eklenmesini sağlamaktır [43]. Bir kuvvetlendirici kazanç, belirli frekans aralığı ve gürültü faktörü olmak üzere üç parametre ile ifade edilir. Kazanç, gelen sinyalin genliğini artırma oranı olarak tanımlanır ve genellikle dB olarak ifade edilir. Gürültü faktörü ise, kuvvetlendirilmiş sinyale eklenen gürültü miktarını tanımlar ve dB olarak ifade edilir [43].

Antenden gelen sinyalin gücü oldukça düşüktür. Bu yüzden alıcılarda genellikle antenden sonra sinyal gücünü artırmak için ön kuvvetlendiriciler kullanılır. Çoğu ticari alıcılarda anten ve ön kuvvetlendirici bütünleşik durumdadır [46]. Düşük Gürültülü Kuvvetlendirici (Low Noise Amplifier, LNA) düşük gürültü oranına sahiptir. Bu özelliğinden dolayı GPS alıcısının ön kuvvetlendirme aşamasında genellikle LNA tipi kuvvetlendiriciler tercih edilmektedir. Ayrıca GPS alıcılarında anten ve ön kuvvetlendirici arasında pasif bir bant geçiren ön filtre devresi, istenmeyen RF girişimini en aza indirmek için kullanılabilir [14].

3.2.2. Aşağı kaydırıcı

Aşağı kaydırıcılar, gelen RF sinyalini daha düşük bir frekans değeri olan Ara Frekans (Intermediate Frequency, IF) sinyaline dönüştürmek için kullanılmaktadır. Gelen RF sinyalinin frekansı çok yüksek olmasından dolayı alıcıda işlem yapmak oldukça zordur. Bunun üstesinden gelmek için aşağı kaydırıcı kullanılarak aynı sinyal daha düşük frekanslı sinyale dönüştürülmektedir. Bu sayede hem sinyal daha düşük frekansa sahip olur, hem de temel bant işlemi kolaylaşır. Ayrıca daha düşük frekansa sahip olduğunda analog-dijital dönüşümü de kolaylaşmaktadır [43]. IF frekansına dönüştürmek için frekans sentezleyici tarafından üretilen dahili osilatörlerden faydalanılmaktadır. Şekil 3.3, aşağı kaydırma işleminin blok diyagramını göstermektedir. Aşağı kaydırma işlemi matematiksel olarak Eşitlik 3.1'deki gibi yazılabilir.



Şekil 3.3. Mikser yapısı [45]

$$\sin(2\pi f_1 t) \sin(2\pi f_2 t) = \frac{1}{2} [\cos(2\pi (f_1 - f_2)t) - \cos(2\pi (f_1 + f_2)t)] \quad (3.1)$$

Burada $\sin(2\pi f_1 t)$ gelen sinyali, $\sin(2\pi f_2 t)$ dahili olarak üretilen sinyali ifade etmektedir. Elde edilen yüksek frekanslı $\cos(2\pi (f_1 + f_2)t)$ bileşenini elimine edip sadece

düşük frekanslı $\cos(2\pi(f_1-f_2)t)$ bileşeni kalması için çarpım sonucu alçak geçiren filtreden geçirilir.

Mikser işleminin sonucunda gelen sinyaldeki faydalı bilgilerde bir değişim veya hata meydana gelmez, sadece RF sinyalinin frekans değeri düşürülür [14]. Aşağı kaydırma işlemi, tek aşamadan oluşmak zorunda değildir. Bu durum direk olarak tasarım kriterlerine bağlıdır. Tek bir ön uç tasarımında çoklu aşağı kaydırıcı kullanılabilir. Her aşağı kaydırma işlemi için bir dahili osilatör kullanılmalıdır [14].

GPS uydularında aşağı kaydırma işleminde, gelen sinyal ile alıcıda dahili osilatörde üretilen sinyal mikser kullanılarak çarpılır ve bunun sonucunda alt ve üst yan bant sinyalleri elde edilir. İşlem kolaylığı açısından daha düşük frekanslı alt yan bant sinyali seçilir ve bu sinyal IF sinyali olarak adlandırılır. Bu işlemler sonucunda uydudan gönderilen sinyalin içerisinde yer alan seyrüsefer mesajı, PRN kodu gibi bilgilerde herhangi bir değişim meydana gelmemektedir [14]. Sadece gelen sinyal, daha düşük bir frekans değerine indirilmektedir.

3.2.3. Osilatörler ve frekans sentezleyiciler

Frekans sentezleyiciler, alıcılarda farklı frekansta osilatör ve saat frekanslarını üretmek için kullanılmaktadır. Referans osilatörlerin görevi, frekans sentezleyiciler tarafından kullanılan kararlı sinüzoidal sinyali üretmektir [45].

Referans osilatörler, temel frekans değerini (ortak değer 10 MHz) kullanarak MHz mertebelerinde farklı frekans değerleri için sinüzoidal sinyal üretir. Frekans sentezleyiciler ile dahili osilatörler, referans osilatör kullanılarak elde edilir [45]. Dahili osilatör (DO) frekansını 10 kHz veya 100 kHz mertebelerinde temel frekans değeri olarak elde etmek için Faz Kilitlemeli Döngü (Phase Lock Loop, PLL) yapısı kullanır. Bu değer, gerekli DO frekanslarını, saat frekanslarını, dahili olarak üretilen C/A kodu frekansını, alıcı devresinin diğer frekans ve zamanlama gereksinimlerini sağlamak için farklı değerlerle çarpılır [45].

Referans osilatörler zamanla idealden uzaklaştığı için kod ve taşıyıcı izleme performansını kötü yönde etkileyen faz gürültüsüne neden olurlar. Faz gürültüsü, bir sinyalde meydana gelen kısa süreli faz dalgalanmalarından kaynaklanan gürültü olarak tanımlanır ve bu dalgalanmalar, sinyalin her iki tarafına yayılmış bir gürültü spektrumu gibi görünen yan bantlar olarak kendini gösterir [45]. Frekans sentezleyiciler, faz

gürültüsünden oldukça fazla etkilenmektedir. Buna bağlı olarak istenilen frekans değerinin dışındaki frekanslarda sahte dalga biçimleri üretmektedir. Üretilen sahte dalga biçimleri de alıcının ön uç performansını oldukça etkilemektedir. Bu faz gürültüsü Allan sapması yöntemi ile karakterize edilebilmekte ve böylece etkisi ortadan kaldırılmaktadır [45]. Allan varyansı, bir veri dizisini zaman domeninde analiz etme ve osilatörde frekans kararlılığı sağlamak için kullanılır. Uygulaması ve hesaplaması basit olan bu yöntem, bir sistem veya sinyaldeki var olan gürültüyü belirleyip tanımlamak için popüler olarak kullanılmaktadır. Allan sapması ise Allan varyansının karekökü alınarak hesaplanmaktadır [47, 48].

İdeal osilatör, sabit frekanslı saf sinüzoidal dalga üretirken, gerçek osilatörler ideal durumdan farklılık göstermektedir [45]. Uygulamada DO, arzu edilen saf sinüs dalgası değildir. Buna ilave olarak faz gürültüsü, ortak DO frekansının harmonikleri ve referans osilatör frekansının harmonikleri ortaya çıkmaktadır. Ortak DO frekansının harmonikleri bastırabildiği için endişe verici bir durum değildir. Ancak faz gürültüsü ve referans osilatör frekansının harmonikleri kolay kolay bastırılmamaktadır. Ayrıca güç kaynağı gerilimindeki dalgalanmaların sonucu olarak çıkış frekansı, yavaş yavaş nominal değerden sapmaktadır. Titreşim ve gürültü gibi etkilerin sonucu olarak da frekansın rasgele dalgalanmasına sebep olmaktadır. Bu frekans kaymasını deterministik olarak modellemek için sonlu bir zaman aralığında Taylor serisi yaklaşımı kullanılmakta ve böylece oluşan hata ortadan kaldırılmaktadır [45].

3.2.4. Analog-dijital dönüştürücüler ve otomatik kazanç kontrolü

Ön uç bölümünde yer alan son eleman analog-dijital dönüştürücülerdir (ADC). ADC'ler analog sinyali, dijital sinyale dönüştürmede kullanılan elemanlardır [43]. Analog bir sinyalin işlenmesi çok karmaşık ve donanım gerektiren bir durum olduğu için ADC'ler aşağı kaydırma işleminden gelen analog IF sinyalini dijitale dönüştürür ve sinyal işleme tekniklerinin daha rahat kullanımına olanak sağlar.

ADC işlemi, genel olarak bant sınırlama, örnekleme aşamalarından meydana gelmektedir [45]. Bant sınırlama gerçekte bir filtre görevi görür ve örneklenmiş sinyal frekansı dışındaki frekans değerlerinde sinyali zayıflatır. Kısaca bant sınırlama, belirli bir frekans aralığı dışında gücü çok düşük olan sinyaller için kullanılır. Bant sınırlı sinyal, belirli bir frekans aralığında gücü yüksek, diğer değerlerde yaklaşık sıfır olan sinyallerdir. Örneğin telefon ses sinyali 400 -3400 Hz arasında bir bant sınırlı sinyaldir. Örnekleme,

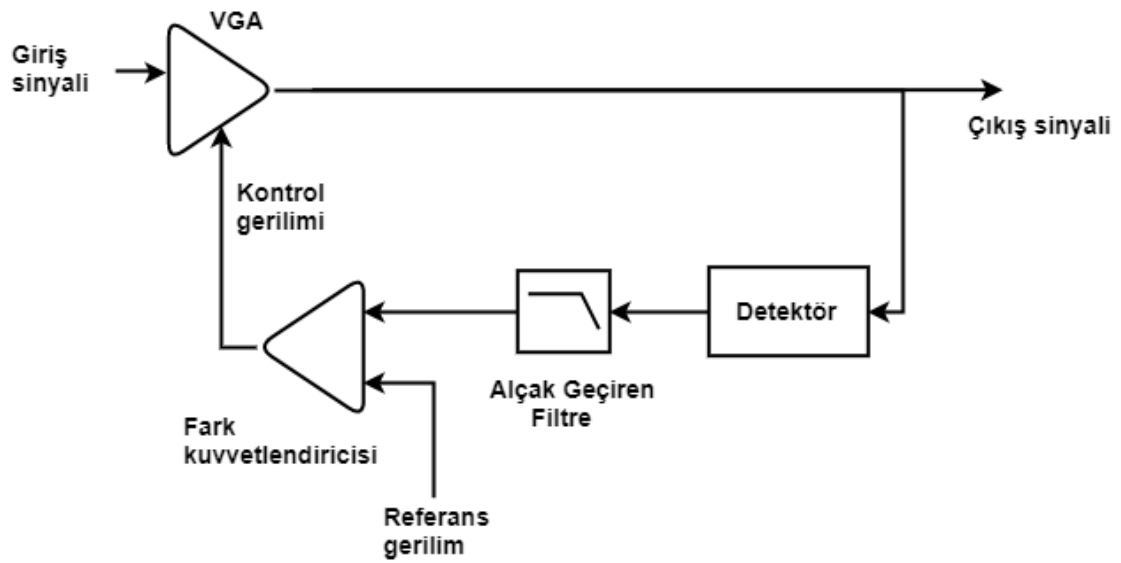
sürekli zaman sinyalinin belirli aralıklarda temsil edilen ayrık zamanlı sayısal sinyale dönüştürülme işlemi olarak tanımlanabilir [45].

ADC' nin çevrim zamanı, örnekleme frekansı, çözünürlük, örnekleme seviyesi, doğruluk ve polarite gibi önemli özellikleri vardır [49]. Bunlardan birisi de çözünürlüktür. Çözünürlük, ADC' nin analog girişindeki en küçük değer değişimine karşılık, çıkışında dijital farklılık oluşturma yeteneğidir. Uygulamada ADC'ler için çözünürlük denildiğinde çıkıştaki bit sayısı akla gelmektedir. Çeşitli üretici firmalar tarafından 8,10,12,14,15,16,18,20,24 bit çözünürlükte değişik ADC entegreleri üretilmiştir. Dijital çıkışın bit sayısının fazla olması çözünürlüğü artırır. Analog sinyalin doğrusal olmaması ve gürültü bulunması çözünürlüğü azaltır. Analog -dijital dönüştürücüde çözünürlüğün yüksek olması istenir [49].

Otomatik Kazanç Kontrolü (AGC), ADC'deki bit sayısını en etkili şekilde kullanmak üzere giriş dalga formunun seviyesini ADC'ye göre ayarlamak için kullanılır [45]. Bazı alıcı tasarımlarında ADC, giriş analog sinyalini uygun bir şekilde örneklemek için yeterli bit sayısına sahip olabilir. Bu durumda AGC'ye ihtiyaç duyulmamaktadır. Ayrıca tek bitlik ADC için de herhangi bir AGC'ye gerek yoktur. Ancak uygulamalarda çoğu kez örnekleme sayısı bir bitten fazla olduğu için AGC'ye ihtiyaç duyulmaktadır. Geleneksel AGC'ler kazancı Ayarlanabilir Kazançlı Analog Kuvvetlendiricilerden (Analog Amplifier Variable Gain, VGA) oluşmaktadır. Bu yaklaşım düşük maliyetli uygulamalarda halen tercih edilmektedir [45]. Bu yaklaşımın olumsuz yanı esnekliğe sahip olmamasıdır. Tipik bir VGA, sabit kazançlı bir kuvvetlendirici ve değişken bir zayıflatıcının kaskad bağlanmasından oluşmaktadır. Şekil 3.4'de, tipik bir analog AGC blok diyagramı verilmiştir [45]. VGA'nın çıkışı, referans değerle karşılaştırılır ve aradaki oluşan fark kazancı ayarlamak üzere geri beslenir.

Şekilde yer alan detektör, yöntem olarak doğrusal veya doğrusal olmayan giriş-çıkış bağıntısı kullanarak kuvvetlendiricinin çıkış gücünü ölçer. Detektör çıkışından alınan sinyal alçak geçiren filtre yardımıyla filtrelenir ve bir girişi referans voltaj olan fark kuvvetlendiricisine uygulanır. Sonuçta filtreli detektör çıkışı ve referans voltaj arasındaki fark VGA'nın kazancını belirler [45].

Analog AGC ve dijital zayıflatıcı AGC olmak üzere iki farklı uygulaması mevcuttur. Analog AGC’de kullanılan zayıflatıcı sabit kazançlıdır. Diğerinde zayıflatıcının kontrolü dijital olarak yapıldığı için dijital zayıflatıcı AGC adı verilmektedir. İkisinin arasındaki en önemli fark, dijital zayıflatıcı AGC için herhangi bir VGA’ya ihtiyaç duymaması ve bunun sonucu olarak kuvvetlendirici tasarımının basitleşmesidir. Ayrıca dijital zayıflatıcı AGC’ de geri bildirim, kuvvetlendirici çıkışından değil ADC çıkışından sağlanır. Bu özellik ile AGC’nin daha hassas bir şekilde ayarlanması ve gelişmiş kazanç kontrolü seçeneğine sahip olması sağlanır [45].



Şekil 3.4. AGC iç yapısı [45]

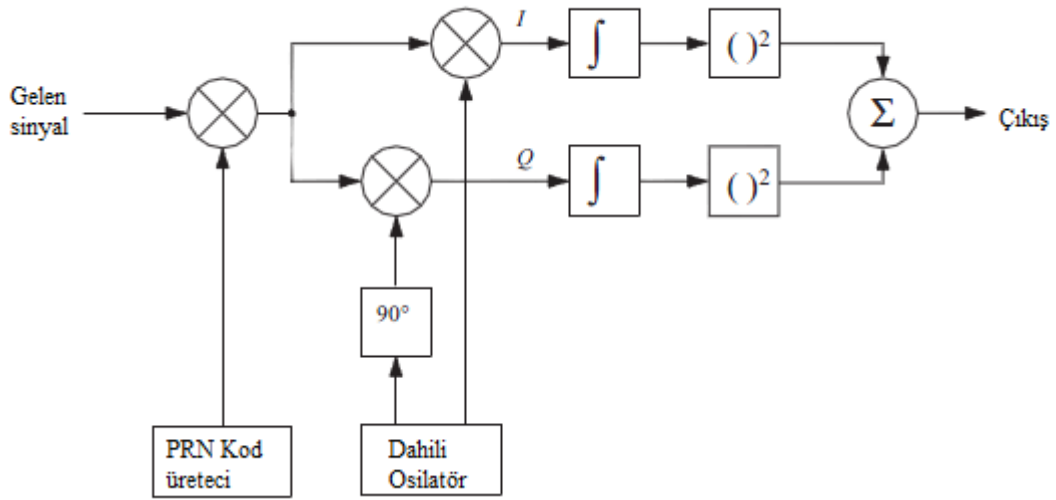
3.3. Sinyal Yakalama

Sinyal yakalamanın ana amacı, görülebilen uyduların belirlenip bu uydulardan gelen sinyallerin kod gecikmesini ve taşıyıcı frekansını belirlemektir [43]. Uydudan gelen sinyalin frekansı nominal değerinden farklılık göstermektedir. Bu farklılığın nedeni Doppler etkisidir. Doppler etkisi, dalga özelliği gösteren herhangi bir fiziksel varlığın frekans ve dalga boyunun hareketli veya sabit bir gözlemci tarafından farklı zaman ve/veya konumlarda farklı algılanması olayıdır. Doppler etkisi sonucunda sinyalin frekansında bir miktar değişim meydana gelmektedir. Bu duruma ise Doppler frekans kayması adı verilmektedir. Uydular açısından dünya üzerindeki sabit kullanıcı için en fazla $\pm 5\text{kHz}$, mobil kullanıcı için ise $\pm 10\text{ kHz}$ frekans kayması meydana gelmektedir. Uydu ile kullanıcı arasındaki mesafeden dolayı gelen sinyal belirli bir gecikme ile alınır.

Bunun sonucunda gelen sinyal içerisinde yer alan PRN kodunun başlangıcında gecikme meydana gelir. Bu durum kod gecikmesi olarak adlandırılır. Sinyalin alıcıya ulaştığı anda 1023 çip uzunluklu PRN kodunda meydana gelen gecikmenin belirlenmesi hem sinyalin doğru olarak çözülmesi hem de alıcı ile uydu arasındaki mesafenin doğru olarak hesaplanması açısından son derece önemlidir [23]. Uydudan gönderilen sinyal frekansındaki değişim ve kod gecikmesini belirlemek için Seri Arama Yakalama (Serial Search Acquisition), Paralel Frekans Alanı Arama Yakalama (Parallel Frequency Space Search Acquisition) ve Paralel Kod Faz Arama Yakalama (Parallel Code Phase Search Acquisition) olmak üzere üç farklı yöntem kullanılır [43].

3.3.1. Seri arama yakalama (Serial search acquisition)

Seri Arama Yakalama, GPS gibi sistemlerin kullandığı Kod Bölmeli Çoklu Erişim (Code-Division Multiple Access, CDMA)'de sıklıkla kullanılan bir yöntemdir [43].



Şekil 3.5. Seri arama yakalama blok diyagramı [43]

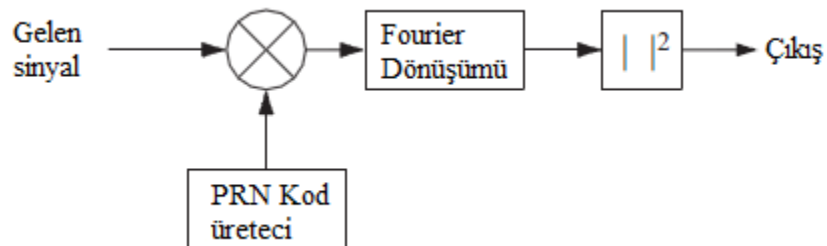
Şekil 3.5’de görüldüğü gibi yöntemin genel prensibi, gelen sinyal ile dahili olarak üretilen PRN kodu ve sonrasında taşıyıcı frekansların çarpımına dayanmaktadır. PRN kod üretici, her bir uyduya karşılık gelen PRN kod dizisini üretir. Üretilen PRN kod dizisi 0 ile 1022 çip arası belirli bir kod fazına sahiptir. Gelen sinyal başlangıçta dahili olarak üretilen PRN kod dizisi ile çarpılır. Gelen sinyalin fazının bilinmemesi sebebiyle sinyalin doğru algılandığından emin olmak için I ve Q sinyali kullanılmaktadır. Burada dahili osilatör taşıyıcı sinyali, I ve 90° faz farklı Q olmak üzere iki sinyal olarak üretilir. Daha

sonra elde edilen sinyal, dahili olarak üretilen taşıyıcı sinyalle çarpılır. I ve Q sinyallerinin bir C/A kodu süresine tekabül eden 1 milisaniye aralıklarla integrali alınarak kareleri toplanır. Bulunan sonuç önceden belirlenen eşik değerini geçerse, frekans ve kod fazının doğru olarak belirlendiği sonucuna varılır.

Bu yöntem, frekans ve kod fazı olarak iki farklı tarama işlemi gerçekleştirmektedir. 500 Hz adımlarla $IF \pm 10$ kHz 'lık tüm olası taşıyıcı frekansları için frekans taraması ($2 \times 10000 / 500 + 1 = 41$) ve 1023 farklı kod fazı için bir çip adımlarla kod fazı taraması yapılır. Neticede toplam 41.943 (41×1023) kombinasyon gerçekleştirilir. Bu sonuca göre frekans ve kod fazı tarama işlemlerinin olası tüm değerlerini sıralı olarak denemenin oldukça zaman alan bir prosedür olduğu görülebilir.

3.3.2. Paralel frekans alan arama yakalama (Parallel frequency space search acquisition)

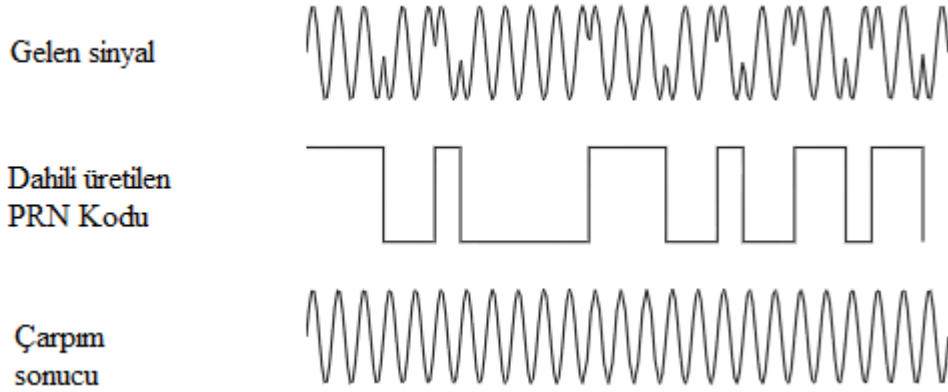
Seri arama yakalama yönteminde gerçekleştirilen kod fazı ve taşıyıcı frekansı tarama işlemi sonucu oluşan 41.943 kombinasyonun oldukça zaman alan bir prosedür olduğu ifade edilmişti. Eğer iki parametreden herhangi biri arama prosedüründen çıkarttırılırsa veya paralel olarak uygulanırsa, yakalama performansı önemli ölçüde artacaktır [43]. Paralel Frekans Alan Arama Yakalama, sadece kod fazı tarama işlemi gerçekleştirir ve böylece sinyal yakalama hızı seri arama metoduna göre önemli ölçüde artar. Bu yöntem, dahili üretilen PRN kodları ve Fourier dönüşümünü kullanır. Fourier dönüşümü ile sinyalin zaman domeninden frekans domenine dönüştürme işlemi gerçekleştirir [43].



Şekil 3.6. Paralel frekans alan arama yakalama blok diyagramı [43]

Şekil 3.6’da gösterildiği üzere yöntem, gelen sinyal dahili olarak üretilen PRN kod dizisi ile çarpılması sonucu elde edilen sinyalin Fourier dönüşümünün alınması yolu ile sinyal yakalama işlevini yerine getirmektedir. Dahili olarak üretilen PRN kodu ile

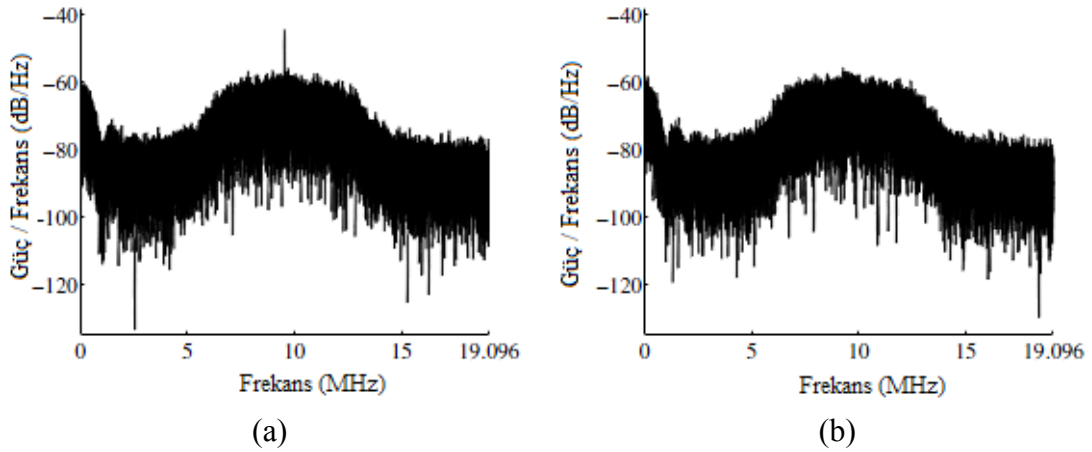
gelen sinyalin PRN kod fazı tam olarak eşleştğinde Fourier dönüşümü sonucunda yüksek bir değer elde edilmektedir. Şekil 3.7, gelen sinyal ile dahili olarak üretilen PRN kodunun tam olarak eşleşmesi sonucunda oluşan taşıyıcı sinyal frekansını göstermektedir. Bu durum sadece gelen sinyal ile dahili üretilen PRN kodunun tam olarak eşleşmesi sonucu oluşmaktadır.



Şekil 3.7. Paralel frekans alan arama yakalama örneği [43]

Şekil 3.8’de iki farklı spektral güç yoğunluğu (Power Spectral Density, PSD) gösterilmektedir. Şekil 3.8.a’da görüldüğü gibi, gelen sinyal ile dahili üretilen PRN kodunun tam eşleşmesi sonucu elde edilen sinyalin Fourier dönüşümü alınmış ve bu sonuca göre gelen sinyalin 10 MHz civarında yüksek bir değere sahip olduğu görülmektedir. Gelen sinyal ile dahili üretilen PRN kodunun tam eşleşmemesi durumunda elde edilen sinyalin Fourier dönüşümü sonucu ise Şekil 3.8.b’de gösterilmektedir. Bu durumda frekans spektrumunda yüksek bir tepe değeri elde edilememektedir.

Seri Arama metodunda olası tüm kod fazı ve taşıyıcı frekans tarama işlemi yapılırken, bu yöntemde sadece kod fazı tarama işlemi yapılmaktadır. Frekans domeni dönüşümü uygulamasına bağlı olarak, seri arama yöntemine kıyasla bu yöntemin daha hızlı bir şekilde uygulanması mümkün olabilmektedir.



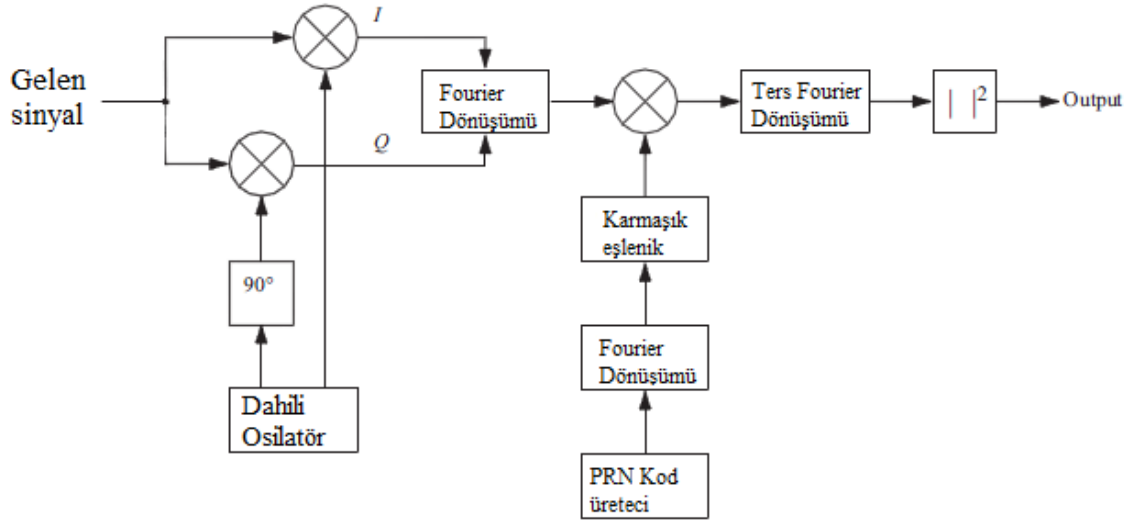
Şekil 3.8. Paralel frekans alan arama yakalama sonucu [43]

3.3.3. Paralel kod fazı arama yakalama (Parallel code phase search acquisition)

Kod faz tarama adımının miktarı (1023 çip) frekans tarama adımından (41 frekans) oldukça fazladır. Bir önceki yöntem, frekans tarama adımını paralelleştirerek 41 olası frekans tarama adımını ortadan kaldırmıştır. Eğer kod fazı tarama işlemi paralel hale getirilebilirse, paralel frekans alanı arama algoritmasındaki 1023'e kıyasla sadece 41 adımda yakalama işlemi tamamlanabilir [43]. Kod faz tarama işlemi paralelleştirip sadece 41 adımda frekans taraması gerçekleştiren yöntem Paralel Kod Fazı Arama Yakalama olarak adlandırılmaktadır. Şekil 3.9, paralel kod fazı arama yönteminin blok diyagramını göstermektedir. Şekilden de görüldüğü gibi, ilk aşama gelen sinyali taşıyıcı sinyalden arındırmaktır. Bunun için gelen sinyal dahili olarak üretilen sinüs ve kosinüs taşıyıcı sinyal ile çarpılır. Bu işlemin sonucunda sırasıyla I ile Q sinyalleri elde edilir. Fourier dönüşümü gerçekleştirilmeden önce karmaşık giriş sinyali oluşturmak için I ve Q sinyalleri $x(n) = I(n) + j Q(n)$ olacak şekilde birleştirilir. Dahili olarak üretilen fazsız PRN kodu, Fourier dönüşümü yardımıyla frekans domenine dönüştürülerek karmaşık eşleniği alınır.

Fourier dönüşümü alınan karmaşık giriş sinyali ile PRN kodunun Fourier dönüşümü çarpılır. Elde edilen sinyali frekans domeninden zaman domenine dönüştürmek için Ters Fourier Dönüşümü kullanılır. Zaman domenine çevrilmiş olan sinyalin mutlak değeri, çarpılan iki sinyalin korelasyon sonucudur. Elde edilen sinyalin mutlak değerinin alınması sonucunda bulunan değer, gelen sinyal ile dahili üretilen PRN

kodu arasındaki korelasyonu verir. İki sinyal arasında bir korelasyon olduğunda spektral güç yoğunluğunda yüksek bir değer elde edilir.



Şekil 3.9. Paralel kod fazı arama yakalama blok diyagramı [43]

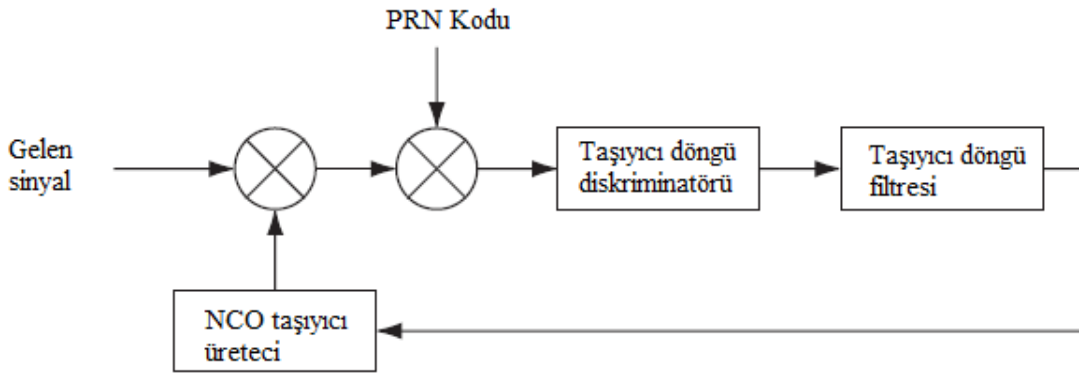
Frekans tarama işleminde, 41 frekans değerinin her biri için bir Fourier ve bir Ters Fourier dönüşümü yapılmaktadır. Bu sebeple metodun hesaplama verimliliği bu fonksiyonların performansına bağlıdır [43]. Üretilen PRN kodlarının Fourier dönüşümü her yakalama için sadece bir kez gerçekleştirilmelidir. Bundan dolayı 1023 farklı kod fazının dikkate alınmasına gerek duyulmadan kod-fazsız PRN kodları üretilmektedir. Böylece diğer metotlara göre oldukça hızlı bir performans sergilemektedir.

3.4. İzleme

Uyduların yakalanması işlemi esnasında her uydudan gelen sinyalin taşıyıcı frekansı ve PRN (C/A) kod fazı yaklaşık olarak bulunur. Takip işleminin amacı zamanla değişen bu bilgileri daha hassas bir şekilde bulmak, takip etmek ve sinyaldeki kullanıcı açısından önemli olan seyrüsefer bilgisini elde etmektir [7, 43]. Bu amaç için kod ve taşıyıcı frekans/fazı olmak üzere iki izleme modu kullanılır. Taşıyıcı izleme modunda, gelen sinyalin frekansında meydana gelen değişimleri belirlemek için Faz Kilitlemeli Döngü (Phase Lock Loops, PLL) veya Frekans Kilitlemeli Döngü (Frequency Lock Loop, FLL) kullanılırken, kod izleme modunda kod fazındaki meydana gelen değişimleri tespit etmek için Gecikme Kilitlemeli Döngü (Delay Lock Loop, DLL) kullanılır [7, 43, 45].

3.4.1. Taşıyıcı izleme (Carrier tracking)

Seyrüsefer mesajını hatasız demodüle etmek için taşıyıcı frekansının çok hassas bir şekilde belirlenmesi gerekmektedir. Bunun için PLL veya FLL yapıları kullanılmaktadır. PLL, taşıyıcının fazında meydana gelen değişimleri, FLL ise frekansında meydana gelen değişimleri belirlememizi sağlar. Taşıyıcı izleme aşamasında genellikle PLL yapısı tercih edildiği için bu bölümde PLL yapıları üzerinde durulacaktır. Şekil 3.10'da, PLL için temel bir blok diyagram gösterilmektedir.



Şekil 3.10. PLL yapısı [43]

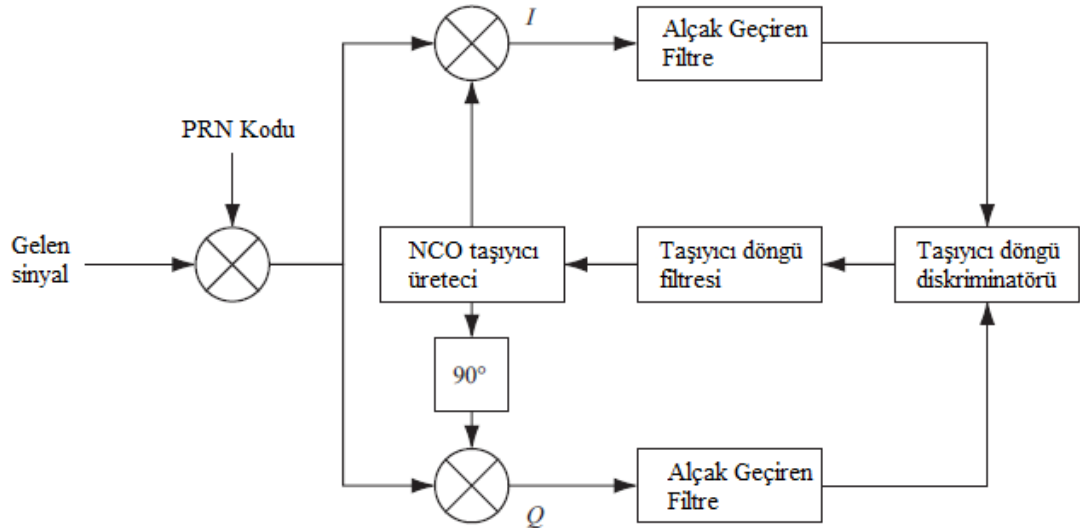
Blok diyagrama göre gelen sinyal taşıyıcı frekanstan ve PRN kodundan ayrıldıktan sonra diskriminatör bloğuna girer. Diskriminatör bloğu, taşıyıcı sinyalin faz hatasını bulmaktadır. Diskriminatörün çıkışı faz hatasıdır ve bir filtreden geçirilerek taşıyıcı frekansını üreten nümerik kontrollü osilatöre (NCO - Numerically Controlled Oscillator) geri besleme olarak verilir. Bu sayede gelen sinyalin taşıyıcı frekansı çok hassas bir şekilde üretilebilmektedir. NCO, ayrık zamanlı sinüs veya kosinüs fazlı dalga üretmek için kullanılır ve bu sayede alıcı hassas frekanslı sinyal üretebilmektedir [14, 50].

Sıradan bir PLL'nin kullanılması sonucu oluşan problem, 180° faz kaymasına karşı duyarlı olmasıdır [43]. Bu durumda PLL'in çıkışı faz hatası oluşturmaktadır. GPS sisteminde kullanılan BPSK modülasyonundan dolayı seyrüsefer mesaj bitleri geçişinde 180° faz farkı olmaktadır bu yüzden de GPS alıcısında kullanılan PLL'in 180° faz kaymalarına karşı faz hatası üretmemesi gerekmektedir. 180° faz farkına duyarsız olan ve seyrüsefer mesaj bitlerinin değişiminden etkilenmeyen PLL yapısı Costas döngüsüdür. Costas döngüsünün blok diyagramı Şekil 3.11'de gösterilmektedir. Costas döngüsünde

faz hatası 0 ve 180° olduğunda diskriminatör çıkışı faz hatası üretmemektedir. Bu sayede seyrüsefer bit değişimi sonucu oluşan 180°'lik faz kaymaları, sinyal takibini etkilememektedir [23, 43]. Bu yapıda, üretilen taşıyıcı sinyal I (in-phase) ve Q (quadrature) olmak üzere iki kola ayrılmaktadır. Costas döngüsü iki çarpımdan oluşmaktadır. Birincisi gelen sinyal ile dahili olarak üretilen taşıyıcı sinyalin çarpımı, ikincisi ise 90° faz farklı taşıyıcı sinyal ile gelen sinyalin çarpımıdır. Costas döngüsünün ana amacı, enerjinin tamamını I kolunda tutmaya çalışmak ve bu koldaki sinyalin takibini yapmaktır [23, 43]. I kolundaki sinyalin takibini yapabilmek için ise NCO taşıyıcı üreticiye bir geri besleme yapılması gerekmektedir [23, 43]. Gelen sinyalin fazıyla üretilen taşıyıcı sinyalin fazı arasındaki fark matematiksel olarak Eşitlik 3.2'deki gibi yazılabilir.

$$\varphi = \tan^{-1} \left(\frac{Q_k}{I_k} \right) \quad (3.2)$$

Burada φ gelen sinyal ile üretilen taşıyıcı sinyalin arasındaki faz farkını, Q_k k. uydunun Q kolu korelasyon değeri, I_k k. uydunun I kolu korelasyon değerini ifade etmektedir.



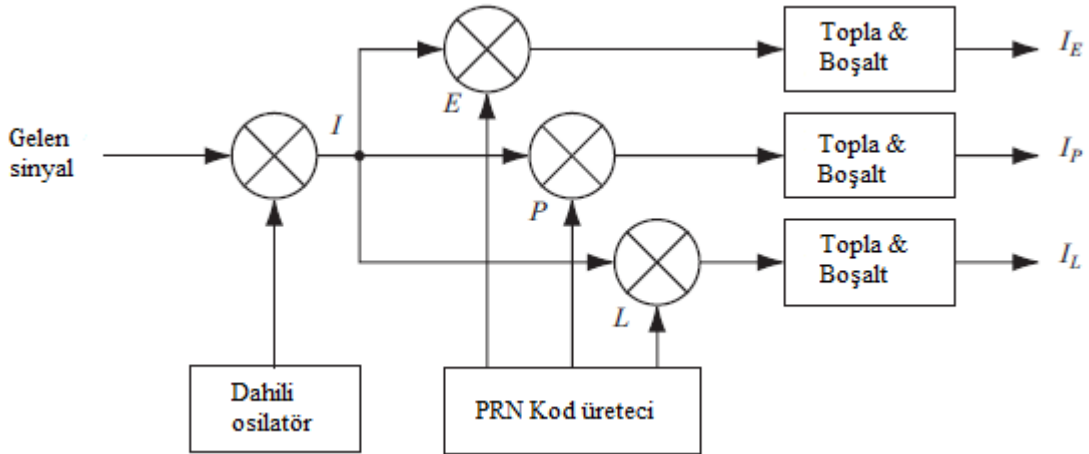
Şekil 3.11. Costas döngüsü [43]

3.4.2. Kod izleme (Code tracking)

Kod izlemenin amacı, gelen sinyaldeki PRN kodunun fazını takip etmek ve belirlemektir. GPS alıcısında kod izleme, erken geç izleme döngüsü (Early-Late Loop)

olarak adlandırılan bir gecikmeli kilit döngüsü (Delay Lock Loop) ile gerçekleştirilir [23, 43]. DLL'in temel çalışma prensibi gelen sinyal ile PRN kodunun üç farklı fazına (Early, Prompt, Late) korelasyon işlemi uygulamaktır.

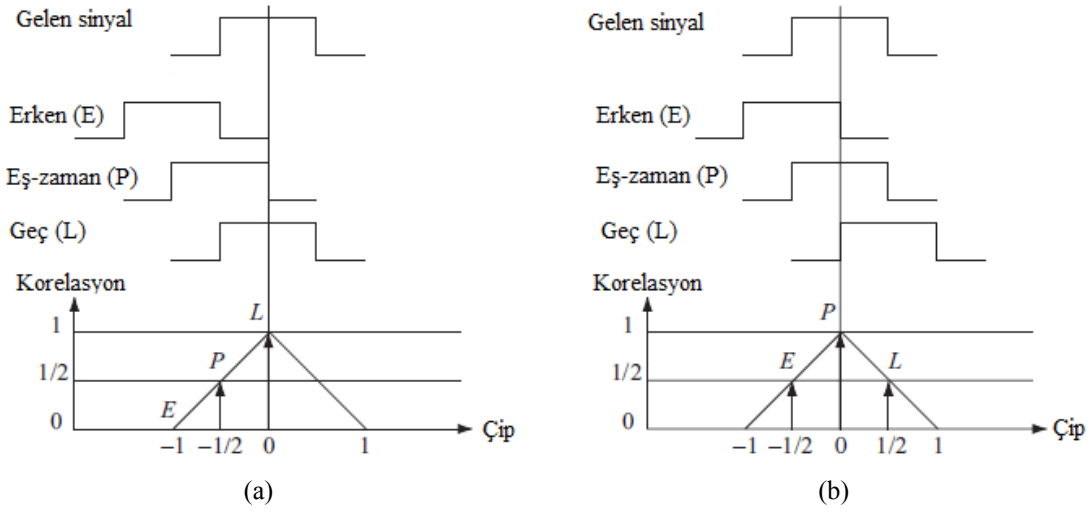
DLL'in ilk aşamasında, gelen sinyal ile dahili olarak üretilen taşıyıcı sinyal çarpılarak gelen sinyal taşıyıcı frekanstan ayrılma işlemi yapılmaktadır (Şekil 3.12). Daha sonra elde edilen sinyal 0.5 çip aralıklarla üretilen Early, Late ve Prompt sinyalleri ile tek tek çarpılır ve elde edilen üç çıkışa Topla ve Boşalt (Integrate and Dump) işlemi uygulanır. Topla & Boşalt, bir sinyalin belirli bir zaman dilimi içerisinde integralinin veya toplamının belirlenmesini sağlamaktadır. Bu işlemin sonucunda sinyal, gürültüden arındırılmaktadır. Bu işlemin sonucunda elde edilen sayısal büyüklükler, gelen sinyal ile ilgili Early, Prompt ve Late kodları arasındaki korelasyon büyüklüğünü vermektedir [23]. Neticede I_E , I_P ve I_L olmak üzere üç korelasyon sonucu karşılaştırılarak en yüksek korelasyona sahip olan sinyal belirlenerek kod gecikmesi tespit edilir.



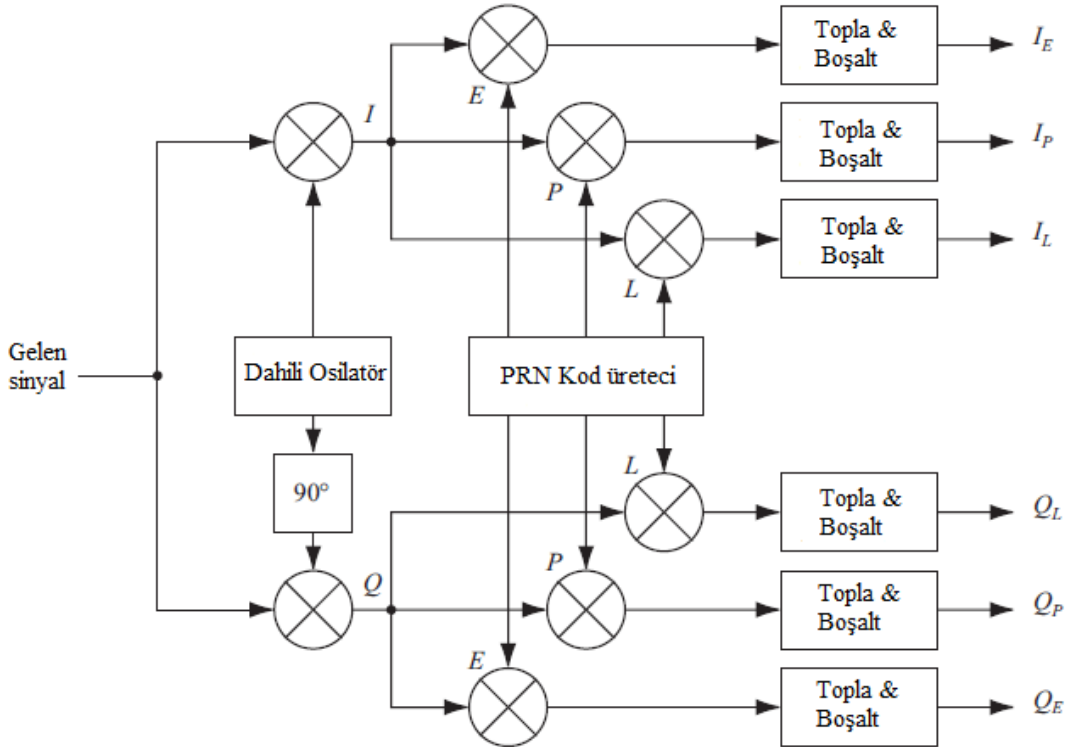
Şekil 3.12. Erken-geç izleme döngüsü [43]

Örnek bir kod izlemeye ait korelasyon sonuçları şekil 3.13'de verilmiştir. Şekil 3.13.a'da gelen sinyale, sırasıyla Early, Prompt ve Late faza sahip kodlarla korelasyon işlemi uygulanır. Bunun sonucunda Late fazlı sinyal en yüksek korelasyona sahip olmaktadır. Bu durum kod fazının azaltılması gerektiğini ifade etmektedir. Şekil 3.13.b'deki gelen sinyal korelasyon işlemi sonucunda Prompt sinyalle en yüksek korelasyona sahip olmaktadır. Bu durum kod fazının doğru olarak belirlenip yakalandığını ifade etmektedir [23].

Üç korelatörlü bir DLL devresi, taşıyıcı sinyalin hem frekans, hem de faz açısının birlikte doğru olarak yakalanabildiği durumlarda iyi sonuç vermektedir [23, 43]. Fakat üretilen taşıyıcı sinyaldeki meydana gelen bir faz hatası sinyalin daha gürültülü olmasına sebep olmakta ve bunun sonucunda DLL devresinin kod takibi yapmasını oldukça zorlaştırmaktadır. Bundan dolayı bir GPS alıcısındaki DLL yapısı çoğu zaman Şekil 14’de görüldüğü gibi tasarlanmaktadır.



Şekil 3.13. Erken-geç döngüsü örnek korelasyon [43]



Şekil 3.14. Altı korelatörlü DLL yapısı [43]

Üretilen taşıyıcı sinyalin fazında meydana gelen hatadan bağımsız olması için altı korelatörlü DLL yapısı kullanılmaktadır. Dahili olarak üretilen taşıyıcı sinyal ile gelen sinyal aynı fazda ise I kolu bütün enerjiye sahip olmaktadır. Ancak aralarında bir faz hatası olursa, enerji I ve Q kolları arasında paylaşılmaktadır. DLL'in performansının, PLL'in performansından bağımsız olması istenirse, izleme döngüsü kodu izlemek için hem I hem de Q kollarını kullanmalıdır [43].

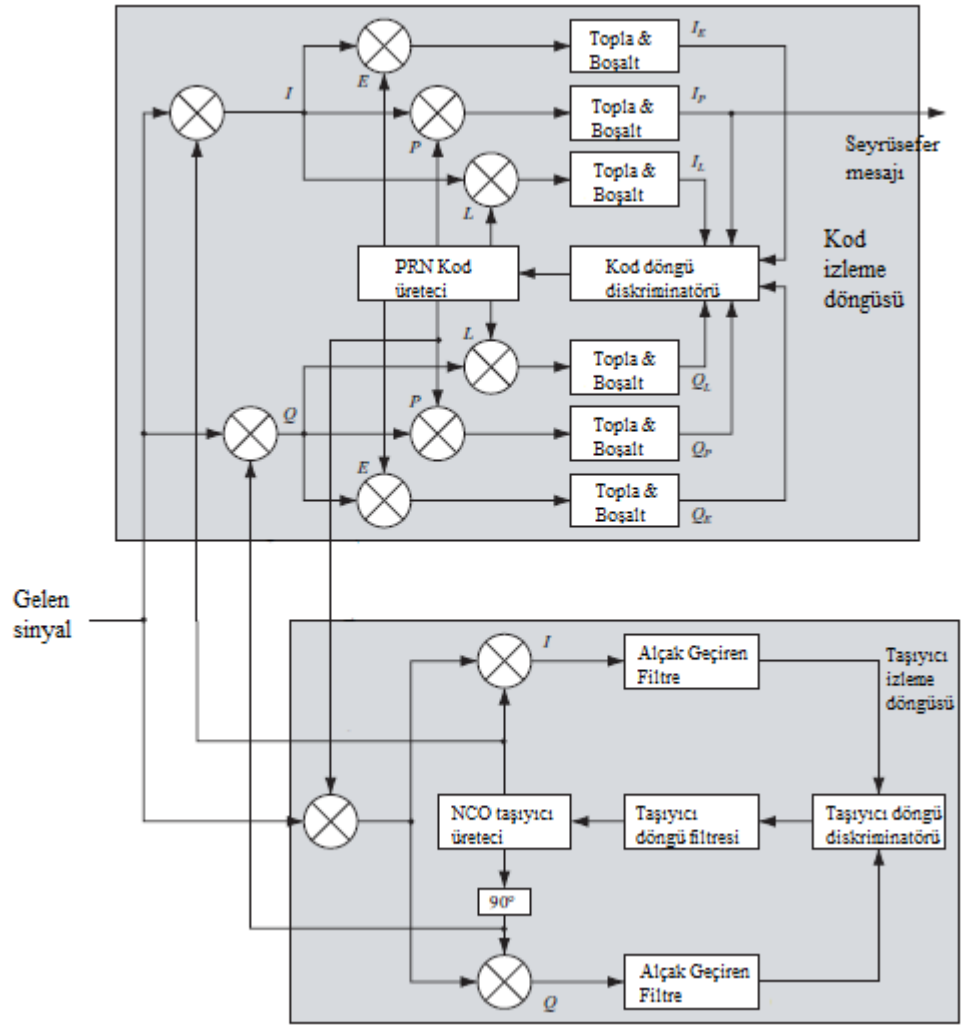
3.4.3. Bütünleşik izleme döngüsü

Kod izleme döngüsü, gelen sinyalin kod fazını DLL yardımı ile belirlemede kullanılırken, taşıyıcı izleme döngüsü taşıyıcı frekansını PLL yardımı ile belirlemede kullanılmaktadır. Bütünleşik izleme döngüsü, kod izleme ve taşıyıcı izleme döngülerinin birleşiminden oluşmaktadır. Bu yapının en önemli avantajı hesaplamaları en aza indirerek işlem süresini azaltmaktır [23, 43]. Şekil 3.15, bütünleşik izleme döngü blok diyagramını göstermektedir. Şekil 3.15'de görüldüğü üzere, frekans izleme döngüsünde gelen sinyali PRN kodundan ayırmak için kullanılan kodun kod izleme döngüsünden geldiği görülmektedir. Aynı şekilde kod izleme döngüsünde, gelen sinyali taşıyıcı sinyalden ayırmak için gerekli sinyalin taşıyıcı frekans izleme döngüsünden geldiği görülmektedir. Blok diyagramda yer alan toplam on bir çarpma işlemi vardır ve bunlar en çok zaman alan işlemlerdir. Çarpım işlemlerinin sayısını azaltmak döngünün daha hızlı çalışmasına katkı sağlar [43]. Bu amaçla birleştirilmiş izleme döngülerinin en iyileştirilmiş hali Şekil 3.16'da gösterilmektedir.

En iyileştirilmiş bütünleşik döngüde, taşıyıcı döngü diskriminatörü I ve Q girişlerine, kod izleme döngüsünden I_p ve Q_p sinyalleri uygulanmıştır. Böylece Costas döngüsündeki üç çarpma işlemine gerek kalmamıştır. Bunun sonucunda hesaplama ve işlem zamanı azaltılarak izleme döngüsü hızlandırılmıştır [23, 43].

3.5. Seyrüsefer Mesajının Elde Edilmesi

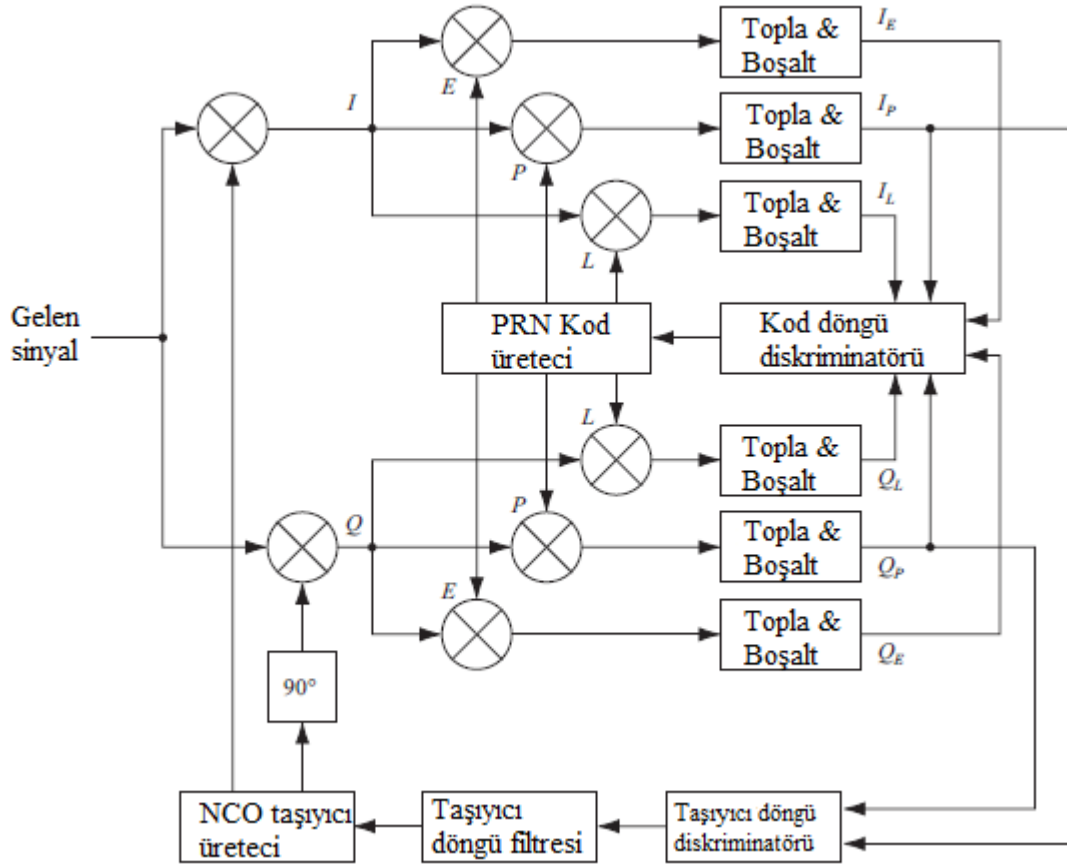
İzleme döngüsü sonucu oluşan sinyal, belirli eşik değerine göre 1 ve -1 olarak değiştirildiğinde seyrüsefer mesajı elde edilir. Teorik olarak izleme döngüsü sonucunda her 1 milisaniyede bir bitlik bilgi elde edilebilir. Ancak gelen sinyallerin zayıf ve gürültülü sinyaller olması sebebiyle seyrüsefer mesajının bir bitlik süresi olan 20 milisaniye aralıklarla ortalama bir değer hesaplanmaktadır. Bulunan değer 1 veya -1 olarak değiştirilmesi sonucu bir bitlik bilgi elde edilir [43].



Şekil 3.15. Bütünleşik izleme döngüsü [43]

Seyrüsefer mesaj bitlerinin doğru bir şekilde elde edilebilmesi için, öncelikle her bir bitlik bilgiyi içeren 20 milisaniyelik bölümlerin belirlenmesi gerekmektedir. Bu amaçla ilk bit değişiminin olduğu (+1'den -1'e ya da tam tersi) yer bulunur ve buradan başlanarak her 20 milisaniyelik bölümden bir seyrüsefer mesaj biti elde edilir [23, 43]. Bir bit değişiminin olduğu başlangıç zamanı belirlendiğinde, seyrüsefer mesaj bit süresinin 20 milisaniye olduğu göz önünde bulundurulduğunda, tüm bit değişimlerinin gerçekleştiği süreler de belirlenebilir. Bu değişimler ilk belirlenen bit değişiminden itibaren 20 milisaniye aralıklarla gerçekleşir. Şekil 3.17'de izleme döngüsü sonucunda elde edilen sinyalin, ilk bit değişiminin gerçekleştiği yerden itibaren 300 milisaniye uzunluğundaki bir bölümü gösterilmektedir. 300 milisaniyelik sinyal, gelen sinyali PRN kodundan ayırdıktan sonra elde edilen seyrüsefer mesajını göstermektedir. Burada bit

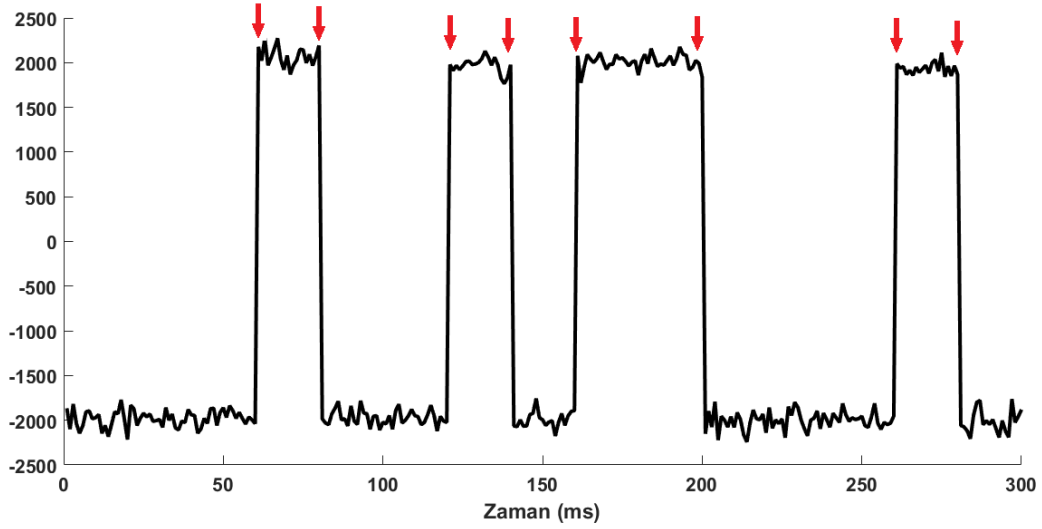
değişimlerinin olduğu yerler ok sembolü ile işaretlenmiştir. Buna göre sinyalin her 20 milisaniyelik ortalama değerinin hesaplanması sonucunda ilk 15 bit “-1 -1 -1 1 -1 -1 1 -1 1 1 -1 -1 -1 1 -1” olarak elde edilir.



Şekil 3.16. En iyileştirilmiş izleme döngüsü [43]

İzleme döngüsünden gelen sinyal yukarıda anlatılan işlemler sonucunda 50 Hz’lik bit dizisine çevrilerek seyrüsefer mesajı elde edilir. Bu mesajda yer alan bitler kullanılarak Bölüm 2.4’de anlatılan seyrüsefer mesajına ait çerçevelerin, alt çerçevelerin ve sözcüklerin bulunarak içerdiği bilgilerin çözülmesi gerekmektedir [23, 43].

GPS seyrüsefer mesajını çözmek için gerekli olan ilk adım, bir alt çerçevenin başlangıç yerini belirlemektir. Her alt çerçevenin başlangıcı preamble olarak adlandırılan bit dizisinden oluşur. Alt çerçevenin başlangıcını belirlemek için kullanılan preamble dizisi 8 bitlik 10001011 şeklindedir [23, 43].



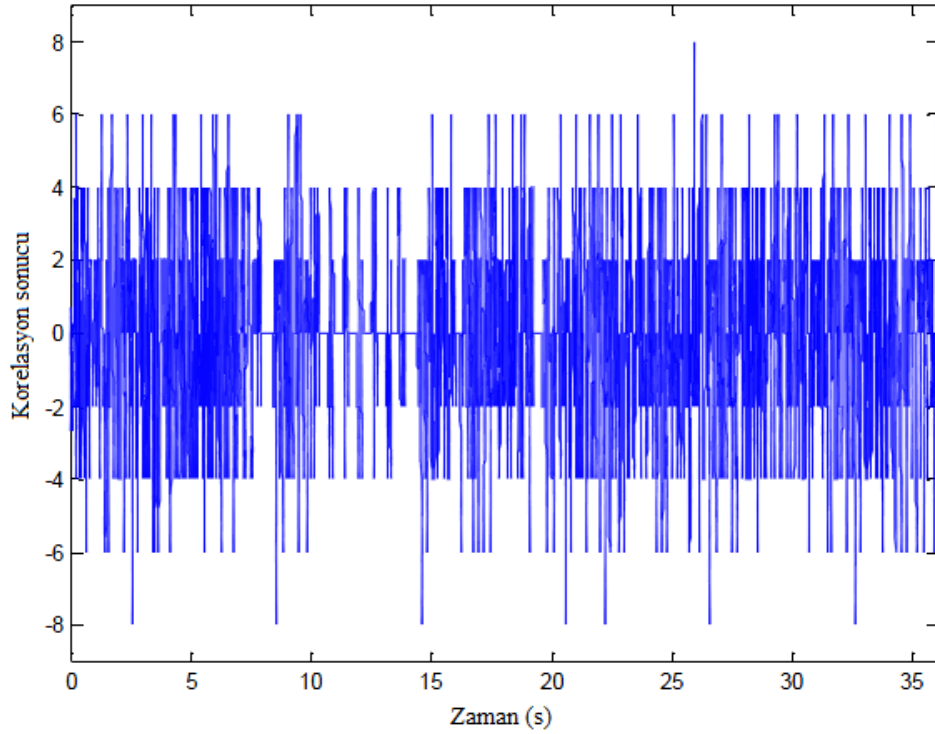
Şekil 3.17. İzleme döngüsü sonucu elde edilen seyrüsefer mesajı [23, 43]

Burada dikkat edilmesi gereken önemli bir durum, izleme aşamasında kullanılan Costas döngüsünün 180° 'lik faz farklı sinyali de izleme yeteneğine sahip olması sebebiyle bu 8 bitlik preamble dizisi ters çevrilmiş 01110100 olarak da görülebileceği unutulmamalıdır. Bu iki bit dizisinden birisi, gelen seyrüsefer mesajının içerisinde yer alabilir. Bu ayırım, preamble arama işlemi aracılığıyla yapılabilir. Bu işlem sonucunda ardışık preamble dizisinin, iki alt çerçevenin iletimi arasındaki süreye denk gelen her 6 saniyede bir tekrarlanıp tekrarlanmadığına bakılır ve 6 saniyede bir tekrar eden preamble dizisi dışındaki aynı bit dizilimine sahip yerler preamble dizisi olarak değerlendirilmez [23, 43].

Preamble arama işlemi korelasyon ile gerçekleştirilir. Korelasyon yapabilmek için 0 olan bitlerin değeri -1 olarak alınır. İzleme döngüsünden elde edilen seyrüsefer mesaj bitleri ile 8 bitlik preamble dizisine korelasyon işlemi uygulanması sonucu preamble'ın eşleştiği yerde +8 veya tersinin eşleştiği yerde -8 bulunur [43]. Şekil 3.18, izleme döngüsünden elde edilen 36 saniyelik seyrüsefer mesajı ile preamble arasındaki korelasyon sonucunu göstermektedir.

Şekil 3.18'de görüldüğü üzere, 36 saniyelik seyrüsefer mesajının içerisinde 6 (36 / 6 s) alt çerçeve olmasına rağmen korelasyon sonucunda 6 noktadan fazla yerde 8 veya -8 değeri elde edilmiştir. Burada sadece 6 saniyede bir tekrarlayan ve eşlik (parite) kontrolünde hata olmayan yerler alt çerçeve preamble'ı olarak alınmalıdır. Şekilde, 6

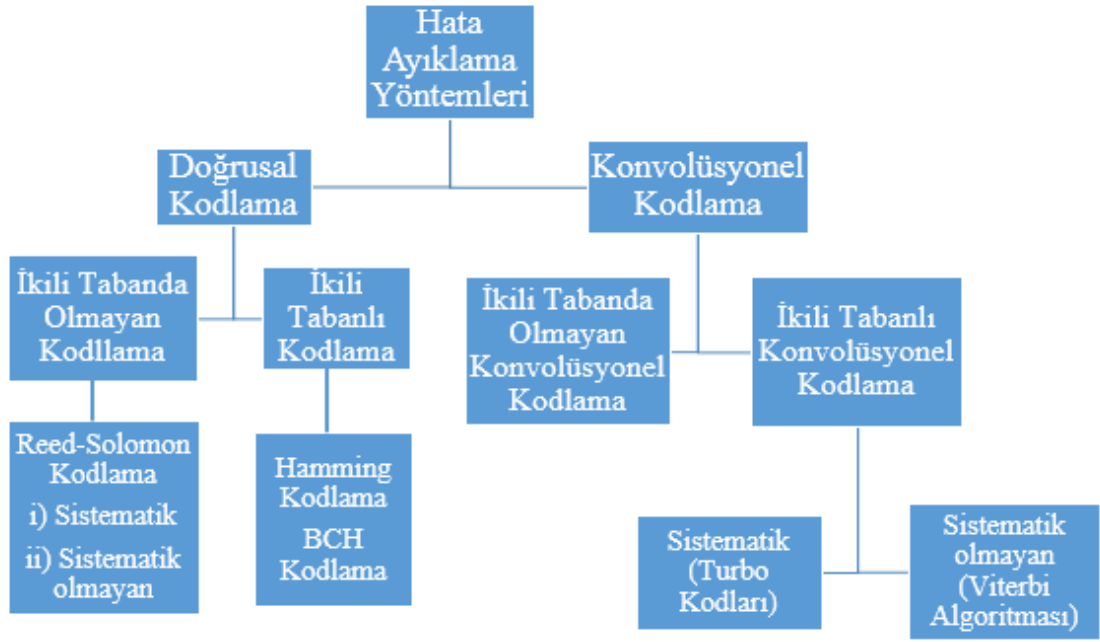
saniyelik aralıklarla korelasyon sonucunun -8 olduğu yerler görülmektedir. Bu yerler, 6 alt çerçevenin başlangıç yerini göstermektedir [23].



Şekil 3.18. Preamble korelasyon sonucu [23]

Preamble yerlerinin doğru olarak bulunması sonucunda alt çerçevelerin başlangıcı belirlenir ve seyrüsefer mesajı elde edilir. Gerekli matematiksel işlem ve dönüşümler yapıldıktan sonra seyrüsefer mesajı içeriği çözülerek kullanıcı açısından önemli olan efemeris bilgisi hesaplanır. Fakat korelasyon sonucu, preamble dizisinin ters çevrildiğini yani -8 olduğunu ifade ederse bütün seyrüsefer mesajı ters çevrilmelidir [23, 43].

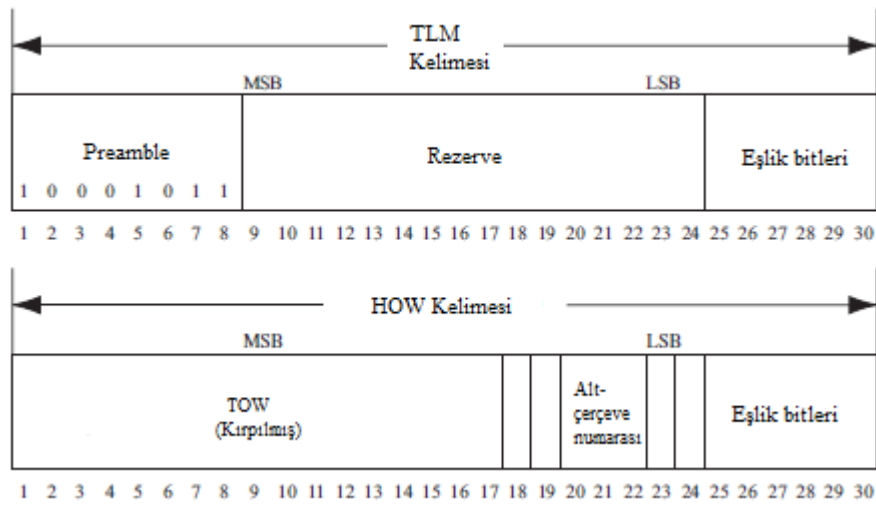
Hata ayıklama yöntemleri, alınan sinyalde herhangi bir hata olup olmadığını tespit etmek için kullanılmaktadır. Bu yöntemler, alınan sinyalde herhangi bir hata varsa kullanılan kodlama tekniğine bağlı olarak düzeltilebilmektedir [51, 52]. Günümüzde kullanılan çeşitli hata ayıklama yöntemleri mevcuttur. Şekil 3.19’da hata ayıklama yöntemlerinin sınıflandırılması gösterilmektedir. Şekilde de görüldüğü üzere Hamming, BCH (The Bose, Chaudhuri and Hocquenghem), Reed-Solomon, Turbo ve Viterbi gibi birçok kodlama tekniği vardır [52]. GPS, hata ayıklama yöntemi olarak en temel kodlama tekniği olan Hamming yöntemini kullanmaktadır [14].



Şekil 3.19. Hata ayıklama yöntemleri [51]

Doğru olarak bulunan her preamble dizisi alt çerçevesinin başlangıcını ifade etmektedir. Her bir alt çerçeve 300 bit uzunluğunda ve 30 bitlik on kelimedenden oluşmaktadır. Her alt çerçevenin ilk iki kelimesi Şekil 3.20’de gösterilen yapıya sahiptir.

Her 30 bitlik kelime 6 eşlik biti ve 24 bitlik seyrüsefer mesajını içermektedir. Eşlik biti, alınan seyrüsefer mesajının içeriğinde herhangi bir hata olup olmadığını kontrol etmek için kullanılır [43]. Bu eşlik bitleri Şekil 3.21’de yer alan eşitlikler yardımıyla hesaplanır. Şekilde yer alan $\oplus$ sembolü XOR işlemini ifade etmektedir.



Şekil 3.20. TLM ve HOW alt-çerçeve yapısı[43]

$D_1 - D_{24}$ kelimedede yer alan veri bitlerini gösterirken, $D_{25} - D_{30}$ eşlik bitlerini göstermektedir. Ayrıca D_{29}^* ve D_{30}^* ise bir önceki kelimeden gelen son iki eşlik bitini ifade etmektedir. Seyrüsefer mesajı alındığında gelen bitlerin doğru bir şekilde alınıp alınmadığının kontrolü eşlik bitleri kullanılarak mutlaka gerçekleştirilmelidir [43].

Eşlik biti kontrolü başarılı bir şekilde gerçekleştirildikten veya gelen verilerde herhangi bir hata olmadığı anlaşıldıktan sonra seyrüsefer mesaj içeriği çözülebilir. Burada en önemli nokta alınan alt çerçevenin GPS uydusundan gönderildiği zamanı belirlemektir [43].

Eşlik biti kodlama

$$\begin{aligned}
 D_1 &= d_1 \oplus D_{30}^* \\
 D_2 &= d_2 \oplus D_{30}^* \\
 D_3 &= d_3 \oplus D_{30}^* \\
 &\vdots \\
 D_{24} &= d_{24} \oplus D_{30}^* \\
 D_{25} &= D_{29}^* \oplus d_1 \oplus d_2 \oplus d_3 \oplus d_5 \oplus d_6 \oplus d_{10} \oplus d_{11} \oplus d_{12} \oplus d_{13} \oplus d_{14} \oplus d_{17} \oplus d_{18} \oplus d_{20} \oplus d_{23} \\
 D_{26} &= D_{30}^* \oplus d_2 \oplus d_3 \oplus d_4 \oplus d_6 \oplus d_7 \oplus d_{11} \oplus d_{12} \oplus d_{13} \oplus d_{14} \oplus d_{15} \oplus d_{18} \oplus d_{19} \oplus d_{21} \oplus d_{24} \\
 D_{27} &= D_{29}^* \oplus d_1 \oplus d_3 \oplus d_4 \oplus d_5 \oplus d_7 \oplus d_8 \oplus d_{12} \oplus d_{13} \oplus d_{14} \oplus d_{15} \oplus d_{16} \oplus d_{19} \oplus d_{20} \oplus d_{22} \\
 D_{28} &= D_{30}^* \oplus d_2 \oplus d_4 \oplus d_5 \oplus d_6 \oplus d_8 \oplus d_9 \oplus d_{13} \oplus d_{14} \oplus d_{15} \oplus d_{16} \oplus d_{17} \oplus d_{20} \oplus d_{21} \oplus d_{23} \\
 D_{29} &= D_{30}^* \oplus d_1 \oplus d_3 \oplus d_5 \oplus d_6 \oplus d_7 \oplus d_9 \oplus d_{10} \oplus d_{14} \oplus d_{15} \oplus d_{16} \oplus d_{17} \oplus d_{18} \oplus d_{21} \oplus d_{22} \oplus d_{24} \\
 D_{30} &= D_{29}^* \oplus d_3 \oplus d_5 \oplus d_6 \oplus d_8 \oplus d_9 \oplus d_{10} \oplus d_{11} \oplus d_{13} \oplus d_{15} \oplus d_{19} \oplus d_{22} \oplus d_{23} \oplus d_{24}
 \end{aligned}$$

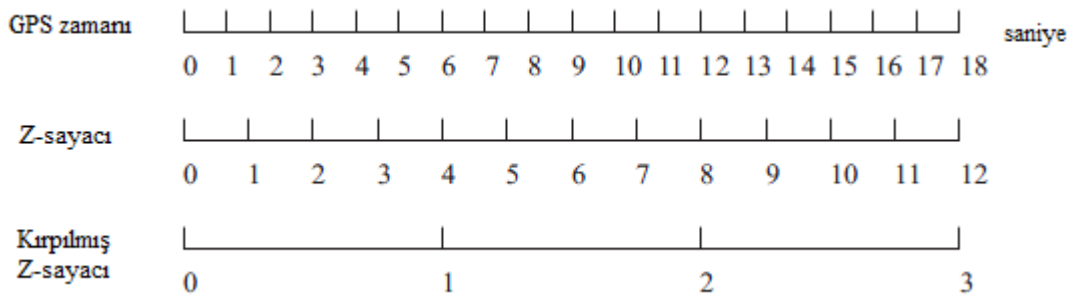
Şekil 3.21. Eşlik biti kodlama eşitlikleri [43]

GPS saati, bir haftadaki saniye sayısıdır ve her yeni haftanın başlangıcında sıfırlanır. Her alt çerçevenin ikinci kelimesinin ilk 17 biti kırpılmış hafta zamanı (Time of Week, TOW) bilgisini içerir. Bu bitler aynı zamanda Z-sayacı olarak da ifade edilir. Z-sayacı dahili olarak her uyduda mevcut GPS haftasının başlangıcından itibaren 1.5 saniye aralıklarla üretilir [7, 43]. Z-sayacı, yüksek değerlikli (most-significant bits, MSBs) 19 bitlik hafta zamanı ve düşük değerlikli (least-significant bits, LSBs) 10 bitlik hafta sayısı (Week Number) olmak üzere 29 bitten oluşur. Uydulardan gönderilen seyrüsefer mesajında 27 bit olan Z-sayacının ilk 10 bitlik hafta sayısı alt çerçevenin ilk üç kelimesinde yer almaktadır. 17 bitlik TOW ise her alt çerçevenin HOW (Handover Word) kısmında bulunmaktadır. Geriye kalan düşük değerlikli 2 bit kırpılmış Z-sayacı kullanılarak elde edilmektedir [43].

TOW, 1,5 saniyelik zamana sahiptir yani her 1,5 saniyede bir değeri artmaktadır. Bir hafta 604.800 ($24 \times 7 \times 3600$) saniye olduğu için TOW 0 ve 403.199 ($604.800 / 1,5 - 1$) arasında değişmektedir. TOW Cumartesiye Pazara bağlayan gece yarısı sıfırlanarak GPS hafta sayısını artırır. 17 bitlik kırpılmış TOW bir çerçeve süresine eşit olan 6 ($1,5 \times 4$) saniyelik zamana sahip olduğu için 0 ve 100.799 ($403.200 / 4 - 1$) arasında değer almaktadır. Şekil 3.22’de görüldüğü gibi, 17 bitlik kırpılmış TOW, 4 ile çarpıldığında gerçek 19 bitlik TOW elde edilir. Böylece 0 ve 100.799 arası değişen kırpılmış TOW değeri 0 ile 403.199 olarak değişecektir [7, 43].

Yüksek değerlikli 10 bitlik Z-sayacı, hafta sayısını vermektedir. Hafta sayısı, 5 Ocak 1980 gecesi/ 6 Ocak 1980 sabahından itibaren geçen hafta sayısını göstermektedir. Toplam hafta sayısı 0 – 1023 arasında olup, bu değerden sonra GPS hafta sayısı sıfırlanmaktadır. Bu yüzden kullanıcılar, GPS saatinden takvim tarihine dönüştürme işlemi yaparken önceki 1024 haftayı da hesaba katmalıdır. HOW’da yer alan kırpılmış Z-sayacı bir sonraki seyrüsefer mesajı alt çerçevesinin gönderilme zamanına karşılık gelir veya diğer bir deyişle bir sonraki alt çerçevenin TOW bilgisini verir. HOW’daki kırpılmış Z-sayacı 6 ile çarpılıp çıkan sonuçtan 6 saniye çıkarıldığında o andaki alt çerçevenin gönderilme zamanı belirlenir [43].

TLM ve HOW kelimelerinden bulunan zaman bilgileri elde edildikten sonra kullanıcı için hayati öneme sahip seyrüsefer mesajı çözülebilir. Ayrıca HOW yardımıyla gelen sinyalin alıcıya ulaşma zamanı belirlendiği için uydu ve alıcı arasındaki pseudorange büyüklüğü de hesaplanabilir.



Şekil 3.22. GPS zamanı, z-sayacı ve kırpılmış z-sayacı [43]

3.6. Uzaklık Hesaplama (Pseudorange)

Konum hesaplamadan önce belirlenmesi gereken en önemli büyüklük alıcı ve uydu arasındaki mesafedir. Alıcı ve uydu saatlerinin senkronize olmamasından

kaynaklanan hatadan dolayı alıcı ile uydu arasındaki uzaklık olması gerekenden farklı çıkmakta ve bu büyüklükte pseudorange olarak ifade edilmektedir. Pseudorange, uydudan gönderilen sinyalin çıkış zamanı ile alıcıya varış zamanı arasında geçen sürenin ışık hızı ($\sim 3 \times 10^8$ m/s) ile çarpılması sonucu hesaplanır. Alıcıya varış zamanı, gelen sinyaldeki alt çerçevenin başlangıcının belirlenmesi ile bulunabilir. Pseudorange, matematiksel olarak Eşitlik 3.3'deki gibi ifade edilebilir.

$$\rho = c \cdot \Delta t \quad (3.3)$$

Burada ρ pseudorange, c ışık hızı ve Δt ise sinyalin çıkış ve varış zamanı arasındaki farkı temsil etmektedir. Pseudorange büyüklüğü belirlendikten sonra seyrüsefer mesajındaki uyduların koordinat bilgilerinin yer aldığı efemeris verileri yardımıyla kullanıcı konumu hesaplanır.

3.7. Konum Hesaplama

Alıcının yerine getirmesi gereken son adım, kullanıcı konumunu hesaplamaktır. Konum, pseudorange ve seyrüsefer mesajı içerisinde gelen efemeris verilerinden elde edilen uydu konumları kullanılarak hesaplanır [43].

Alıcı her uydudan gelen sinyallerin varış zamanını belirler ve bu süreye göre pseudorange'i hesaplar. Daha sonra seyrüsefer mesajı içerisinde yer alan efemeris verisinden uyduların koordinat bilgilerini elde eder. Alıcı konumunu bulabilmek için Eşitlik 3.4'de yer alan denklem kullanılır.

$$\rho_i = \sqrt{(x_i - x_u)^2 + (y_i - y_u)^2 + (z_i - z_u)^2} + cb_u \quad (3.4)$$

Burada ρ_i i. uydu ile alıcı arasındaki uzaklığı, x_i i. uydunun x koordinatını, y_i i. uydunun y koordinatını, z_i i. uydunun z koordinatını, x_u alıcının x koordinatını, y_u alıcının y koordinatını, z_u alıcının z koordinatını, c ışık hızını ve b_u alıcı ve uydu saatleri arasındaki zaman farkını göstermektedir.

3 boyutta konum hesabı için alıcının x, y, z koordinat ve zaman bilgileri olmak üzere dört parametrenin belirlenmesi gerekir. Bu yüzden de alıcı dört veya daha fazla uydu sinyaline ihtiyaç duymaktadır. En az dört uydudan alınan seyrüsefer mesajı ile konum bulmak için Eşitlik 3.4'de yer alan ifade kullanılarak dört farklı denklem elde edilir. x, y, z koordinatı ve alıcı saat hatası bu dört denklem ile belirlenebilir. Ancak eşitlikte yer alan karesel ifadelerden dolayı sonuçta iki farklı çözüm elde edilir ve bu

durumda doğru olan çözümü belirlemek gerekmektedir. O halde kullanılan bu yöntem her zaman doğru çözümü vermeyebilir. Oluşan problemin üstesinden gelmek için doğrusal olmayan dört bilinmeyenli denklem, Taylor serisine açılıp ikinci ve daha üst dereceden terimler ihmal edilerek doğrusallaştırılır. Daha sonra En Küçük Kareler metodu uygulanarak iteratif çözüm elde edilir ve sonuçta kullanıcının x, y, z koordinatı ve alıcı saat hatası hesaplanır [6, 23, 43].

Eşitlik 3.4’de yer alan denklemin doğrusallaştırılması sonucu elde edilen ifade Eşitlik 3.5’de verilmiştir.

$$\begin{bmatrix} \delta\rho_1 \\ \delta\rho_2 \\ \delta\rho_3 \\ \delta\rho_4 \\ \vdots \\ \delta\rho_n \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 1 \\ a_{21} & a_{22} & a_{23} & 1 \\ a_{31} & a_{32} & a_{33} & 1 \\ a_{41} & a_{42} & a_{43} & 1 \\ & \cdot & & \\ & \cdot & & \\ & \cdot & & \\ a_{n1} & a_{n2} & a_{n3} & 1 \end{bmatrix} \begin{bmatrix} \delta x_u \\ \delta y_u \\ \delta z_u \\ \delta cb_u \end{bmatrix} \quad (3.5)$$

$$a_{i1} = \frac{x_i - x_u}{\rho_i - cb_u} \quad a_{i2} = \frac{y_i - y_u}{\rho_i - cb_u} \quad a_{i3} = \frac{z_i - z_u}{\rho_i - cb_u} \quad (3.6)$$

Eşitlik 3.5’de yer alan matris formun daha basitleştirilmiş hali Eşitlik 3.7’de verilmiştir.

$$\delta\rho = a \delta x \quad (3.7)$$

Burada $\delta\rho$ ve δx vektör iken, a matris formundadır. Bu ifadelerin genişletilmiş hali Eşitlik 3.8’de gösterilmektedir.

$$\delta\rho = [\delta\rho_1 \ \delta\rho_2 \ \delta\rho_3 \ \dots \ \delta\rho_n]^T$$

$$\delta x = [\delta x_u \ \delta y_u \ \delta z_u \ \delta cb_u]^T$$

$$a = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 1 \\ a_{21} & a_{22} & a_{23} & 1 \\ a_{31} & a_{32} & a_{33} & 1 \\ a_{41} & a_{42} & a_{43} & 1 \\ & \cdot & & \\ & \cdot & & \\ & \cdot & & \\ a_{n1} & a_{n2} & a_{n3} & 1 \end{bmatrix} \quad (3.8)$$

Burada $[]^T$ ifadesi bir matrisin transpozunu ifade etmektedir. α kare matris olmadığı için direk olarak tersi alınamaz. Eşitlik 3.8 doğrusal bir denklemdir ve bir doğrusal denklemde bilinmeyenlerden daha fazla denklem mevcutsa en küçük kareler yöntemi çözümü bulmak için kullanılabilir. Sonuçta bu eşitliğe En Küçük Kareler metodu uygulanırsa Eşitlik 3.9'daki denklem elde edilir.

$$\delta x = [a^T a]^{-1} a^T \delta \rho \quad (3.9)$$

$$\delta v = \sqrt{(\delta x_u)^2 + (\delta y_u)^2 + (\delta z_u)^2 + (\delta cb_u)^2} \quad (3.10)$$

Eşitlik 3.10'da yer alan δv istenilen sonuca ulaşım ulaşılmadığını belirlemek için kullanılan niceliktir. Bu nicelik keyfi olarak belirlenen eşik değerinden daha küçük olduğunda iterasyon sonlanmaktadır.

İstenilen çözümü bulabilmek için Eşitlik 3.9 iteratif yolla çözümlenmelidir. Bunun için takip edilmesi gereken adımlar aşağıdaki gibi sıralanabilir [7]:

- i) x_{u0} , y_{u0} , z_{u0} ve cb_{u0} için başlangıç şartı değeri seçilir. Örneğin pozisyon olarak dünyanın merkezi (0, 0, 0) ve saat hatası olarak sıfır alınabilir. Yani tüm değişkenler için başlangıç değerleri sıfır olarak seçilir.
- ii) Eşitlik 3.4 kullanılarak her bir uydu için pseudorange ρ_i hesaplanır. Burada i uydu numarasını göstermektedir. Bu ρ_i değeri ölçülen değerden farklıdır. Ölçülen ve hesaplanan değer arasındaki fark $\delta \rho_i$ olarak tanımlanır.
- iii) Hesaplanan $\delta \rho_i$ yardımı ile Eşitlik 3.6'da yer alan a_{i1} , a_{i2} ve a_{i3} değerleri hesaplanır.
- iv) δx_u , δy_u , δz_u , δcb_u değerlerini hesaplamak için Eşitlik 3.9 kullanılır.
- v) δx_u , δy_u , δz_u , δcb_u ve Eşitlik 3.10 yardımı ile δv niceliği hesaplanır.
- vi) δv önceden keyfi olarak belirlenmiş eşik değeri ile karşılaştırılır ve bu değerden büyükse sıradaki işlemlere devam edilir.
- vii) Hesaplanan δx_u , δy_u , δz_u , δcb_u nicelikleri başlangıçta seçilen x_{u0} , y_{u0} , z_{u0} pozisyon ve cb_{u0} saat hatası değerlerinin üzerine eklenir. Sonuçta yeni pozisyon ve saat hatası x_{u1} , y_{u1} , z_{u1} , cb_{u1} olarak ifade edilir. Bu değerler bir sonraki adımda başlangıç pozisyon ve saat hatası olarak kullanılır.

- viii) δv , eşik değerinden küçük olana kadar i'den vii'ye kadar yer alan adımlar tekrarlanır. Bulunan çözüm kullanıcı konumu ile alıcı saat hatasını verir ve x_u , y_u , z_u ve cb_u olarak ifade edilir.

Genel olarak yukarıdaki iterasyon metodunda hesaplanan δv değeri hızlı bir şekilde azalmaktadır ve seçilen eşik değerine bağlı olarak iterasyon metodu genellikle 10'dan az iterasyonda istenilen çözüme ulaşabilmektedir [7].

4. GPS ALICI ARAYÜZÜ TASARIMI

GPS sisteminin sinyal yapısı, kullanılan modülasyon tekniği, sinyal üretimi ile ilgili bilgiler çalışmanın 2. Bölümü, GPS sinyalinin demodüle edilme aşamasında gerçekleştirilen işlemler ile ilgili bilgiler de 3. Bölümde detaylı olarak verilmiştir. Bu bölümde, çalışmanın amacı doğrultusunda uygulama alanı olarak seçilen GPS L1 sinyalinin üretilmesi ile ilgili işlem adımları, elde edilen sinyalin demodülasyon işlemi ve geliştirilen kullanıcı arayüzü detaylı bir şekilde açıklanmıştır.

GPS L1 sinyalinin simülasyonu için tasarlanan arayüzde gerçekleşen adımlar Şekil 4.1'deki akış diyagramında gösterilmektedir. Akış diyagramı referans alınarak simülasyonda gerçekleştirilen işlemler ve kullanılan parametreler başlıklar halinde ilerleyen bölümlerde anlatılmıştır.

4.1. Anasayfa

Anasayfa bölümü simülasyon arayüzünün giriş ekranını oluşturmaktadır. Anasayfada yer alan 'Simülasyonu Başlat (Start Simulation)' butonuna basıldığında başlangıç parametrelerinin girildiği ekran açılmaktadır. Şekil 4.2, simülasyon arayüzünün giriş ekranı olan anasayfayı göstermektedir.

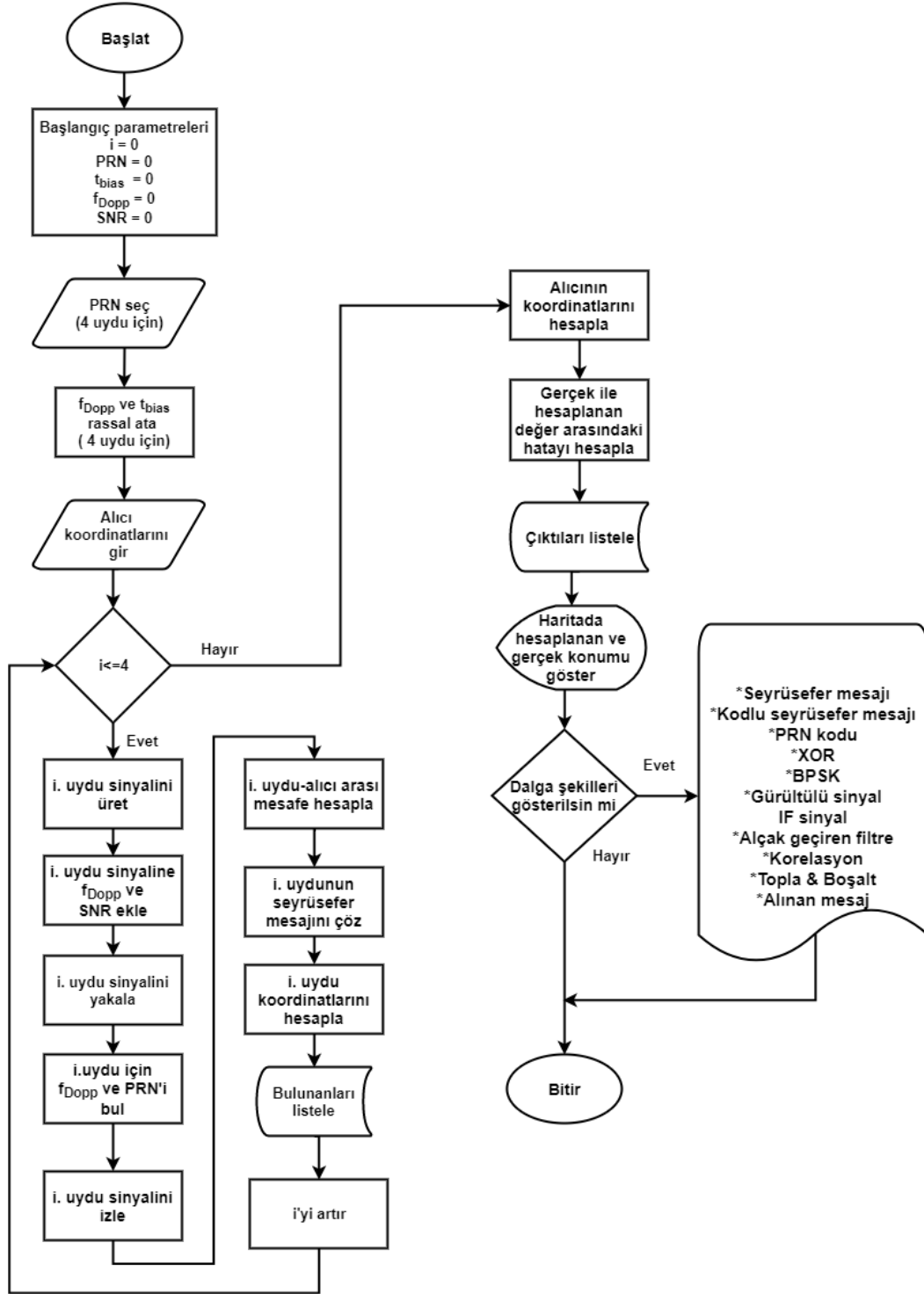
4.2. Başlangıç Ekranı

Başlangıç ekranı, dört farklı uydunun kullanıcı tarafından keyfi olarak seçilen PRN kodları, Sinyal Gürültü Oranı (Signal to Noise, SNR) ile gerçek uydular referans alınarak rassal atanan Doppler frekansı ve alıcı saat hatasının belirlendiği bölümdür.

Simülasyonun gerçek dünya ile tam olarak uyumlu olabilmesi için kullanıcı 1'den 32'ye kadar olan PRN kodları arasında tercih yapabilmektedir. Uygun olmayan uydu koordinat bilgisi girilmesinin önüne geçmek için kullanıcı tarafından seçilen PRN koduna bağlı olarak, gerçek GPS uydularına ait Kartezyen koordinat bilgileri referans alınarak oluşturulmuş veri tabanından ilgili uyduya ait anlık (X,Y,Z) koordinatları otomatik olarak atanmaktadır.

SNR değeri, dB cinsinden alınan sinyal gücünün gürültüye oranı olarak tanımlanır. SNR ne kadar yüksek ise sinyal o kadar az gürültü içerir. Ne kadar düşük olursa alınan sinyalin bozulma ihtimali o kadar yüksek olmaktadır. Bu yüzden SNR değerinin yüksek olması istenen bir durumdur. Kullanıcının aldığı GPS sinyalinin ortalama SNR değeri -29 ve -21 dB arasında değişmektedir [53]. Bu durum dikkate

alınarak başlangıç parametrelerinde gerekli olan SNR değerinin skalası -40 ile 0 dB arasında geniş bir aralık alınarak istenilen gürültü seviyesinde sinyal üretme ve bu sayede gürültünün sinyale olan etkisini inceleme imkânı sunulmaktadır.



Şekil 4.1. Arayüz akış diyagramı



Şekil 4.2. Arayüz anasayfası

4 uydu için PRN kodları ve SNR değerleri belirlendikten sonra alıcının enlem ve boylam bilgilerinin girilmesi gerekmektedir. Alıcı enlem ve boylam bilgilerinin girilmesindeki amaç, uydulardan gelen sinyalleri çözüp seyrüsefer mesajını kullanarak hesaplanan konum bilgisi ile gerçek konum bilgisi arasındaki hatayı belirlemektir. Enlem ve boylam bilgilerinin gösterilmesi için Derece-Dakika-Saniye, Derece-Dakika ve Ondalık Derece gibi çeşitli birimler kullanılmaktadır. Bu simülasyon çalışmasında, kolaylık olması açısından alıcının enlem ve boylam bilgileri Ondalık Derece biriminde girilmektedir.

Simülasyonu başlatmadan önceki son adım, Doppler frekans ve alıcı saat hatasını belirlemektir. Doppler kayması, uydu ve alıcının hareketli olmasından kaynaklanan, uydudan gönderilen sinyal ile alıcıya gelen sinyal arasındaki frekans farkıdır. Sabit kullanıcı için Doppler frekans kayması en fazla ± 5 kHz iken, mobil kullanıcı için ise ± 10 kHz aralığındadır. Bu çalışmada kullanıcının mobil olduğu varsayılarak Doppler frekansı ± 10 kHz arasında alınmıştır. Uydu ve alıcı hareketine bağlı olarak değişiklik gösteren Doppler frekansı, 'Seç (Select)' butonuna basıldığında her bir uyduya ± 10 kHz aralığında rassal olacak şekilde atanmaktadır. Konum hesaplamada ortaya çıkan hatalardan birisi de,

alıcı ile uydu saatinin birbiriyle senkronize olmamasından kaynaklanmaktadır. Eğer alıcı saati ve uydu saati birbiriyle senkronize ise hesaplanan uzaklık değeri hata olmazken, uydu ile alıcı arasındaki mikro saniyelik bir zaman farkı dahi hesaplanan uzaklık değeri büyük hatalara sebep olabilmektedir. Bu büyüklüğün etkisi de göz önünde bulundurularak gerçekleştirilen simülasyon çalışmasında alıcı saat hatası, başlangıç parametrelerinde yer alan 't_bias' bölümünde gösterilmektedir. Alıcı saat hatası, alıcının kullandığı dahili saatten kaynaklanan bir hatadır. Bu yüzden de alıcı saat hatası için herhangi bir tipik değer söz konusu değildir. Bu durum dikkate alınarak alıcı saat hatası, 10 ile 50 mikro saniye arasında makul bir aralık belirlenmiştir. Seç butonuna basıldığında 't_bias' bölümüne 10 ile 50 mikro saniye aralığında rassal bir sayı, alıcı saat hatası olarak atanmaktadır. Simülasyonun ön hazırlık evresi olan başlangıç parametrelerinin tamamı girildikten sonra 'Başlat (Start)' butonuna basılarak simülasyon çalıştırılmaktadır (Şekil 4.3).

4.3. Sinyal Üretimi

GPS sinyalinin üretimi ile ilgili detaylı bilgiler 2. Bölümde verilmiştir. Bu bölümde sadece simülasyonda yapılan işlemler anlatılmıştır. Başlangıç parametrelerinin girilip simülasyonun çalıştırılması sonrası gerçekleştirilen ilk adım GPS L1 sinyalinin üretilmesidir. Bu aşamada sinyal üretimi için gerekli parametreler, seyrüsefer mesajını oluşturan uydu koordinat bilgileri ve PRN kodudur.

Seyrüsefer mesaj yapısı uydu koordinatları, iyonosferik hata düzeltmesi, saat düzeltmesi gibi detaylı bilgiler içermektedir. Bu yüzden seyrüsefer mesajını oluşturmak için bazı matematiksel dönüşüm ve hesaplamaların yapılması gerekmektedir. Simülasyon çalışmasında işlem kolaylığı olması açısından seyrüsefer mesajı sadece uydunun koordinat bilgileri kullanılarak oluşturulmuştur. Onluk tabanda girilen kartezyen koordinat (X,Y,Z) bilgileri, ikilik (binary) tabana dönüştürülerek seyrüsefer mesajı elde edilmiştir.

GPS'te kullanılan Hamming, alınan sinyalde herhangi bir hata olup olmadığını tespit etmek için kullanılan hata ayıklama yöntemlerinden birisidir. Hamming yöntemi, alınan sinyaldeki bir bitlik hatayı tespit ederek düzeltir veya sadece iki bitlik hatayı tespit

GPS Signal Simulation

INPUTS

1st Satellite Coordinates

X	3.21922e+07
Y	1.7659e+07
Z	1.40924e+07

2nd Satellite Coordinates

X	2.85298e+07
Y	2.42268e+07
Z	3.02152e+07

3rd Satellite Coordinates

X	2.30406e+06
Y	2.32879e+07
Z	1.1917e+07

4th Satellite Coordinates

X	1.67291e+07
Y	3.08722e+06
Z	2.32037e+07

Real Receiver Coordinates

Latitude (decimal)	39.813
Longitude (decimal)	30.523

PRN Number

PRN10

Doppler Freq. (kHz)

6.5

SNR (dB)

-25

PRN Number

PRN16

Doppler Freq. (kHz)

-7.5

SNR (dB)

-20

PRN Number

PRN2

Doppler Freq. (kHz)

8.5

SNR (dB)

-20

PRN Number

PRN13

Doppler Freq. (kHz)

3.0

SNR (dB)

-25

t_b bias (microsec.)

50.0

Select

Start

Şekil 4.3. Parametrelerin girildiği başlangıç ekranı

eder. Hamming yöntemi ile oluşturulan eşlik biti kullanılarak alınan sinyalde herhangi bir hata olup olmadığı tespit edilir ve varsa bir bitlik hata düzeltilir. Bu çalışma kapsamında gürültü ve diğer bozucu etkilerden kaynaklanan hataları düzeltmek için GPS sisteminde de kullanılan Hamming kodlaması tercih edilmiştir. Seyrüsefer mesajının Hamming ile kodlanması sonucu 5,040 saniye uzunluğunda kodlu seyrüsefer mesajı elde edilir. Daha sonra kodlu sinyale, aynı süreye sahip PRN kodu ile XOR işlemi uygulanır ve sonuçta oluşan sinyal, BPSK modülasyonu kullanılarak 1575,42 MHz frekanslı taşıyıcı sinyal üzerine bindirilerek alıcılara gönderilir.

4.4. Kanal

GPS uydularının gönderdiği sinyaller alıcılara uzay boşluğu aracılığıyla gönderilir. Bu sinyaller uzay boşluğundan alıcıya ulaşana kadar gürültü gibi birçok bozucu etkiye maruz kalmakta ve bunun sonucunda da kullanıcı gürültülü bir sinyal almaktadır.

Kanal aşaması, üretilen sinyale gürültü, alıcı-uydu arasındaki mesafeden kaynaklanan gecikme süresi ve Doppler frekansının eklendiği bölümdür. Gürültü eklemek için sinyale Toplanır Beyaz Gauss Gürültü (Additive White Gauss Noise, AWGN) kanal modeli uygulanmıştır. AWGN kanal modeli, alınan işareti bozma yönünde eklenmiş istatistiksel olarak bağımsız gürültü örneklerinden oluşmaktadır. Aynı zamanda AWGN kanalının genellikle aynı ve davranışının zamanla değişmediği kabul edilmektedir. Üretilen sinyale, başlangıç parametrelerinde belirlenen SNR değeri kullanılarak gürültü eklenmiş ve böylece gerçek GPS sinyali ile uyumlu hale getirilmiştir.

GPS'in çalışma prensibine uygun hale getirilebilmesi için Doppler etkisi ve gecikme süresi üretilen sinyale uygulanmalıdır. Simülasyonda başlangıç parametrelerinde her bir uyduya rassal olarak atanan Doppler frekansı, uydudan gönderilen sinyal frekansının (1575,42 MHz) Doppler frekansı kadar değiştirilmesi ile sisteme dahil edilmektedir. Uydu ile alıcı arasındaki mesafeden dolayı uydudan gönderilen sinyal alıcıya belirli bir gecikme süresi ile ulaşmaktadır. Belirli gecikme süresi, başlangıçta girilen alıcı ile uydu arasındaki uzaklığın belirlenmiş ve ışık hızına bölünerek gecikme süresi hesaplanmıştır. Daha sonra bu gecikme süresine denk gelecek şekilde PRN koduna çip gecikmesi uygulanmıştır. Simülasyonun bu aşamasında gönderilen sinyale belirli bir gecikme süresi uygulanmakta ve daha sonra sinyal alıcıya ulaşmaktadır.

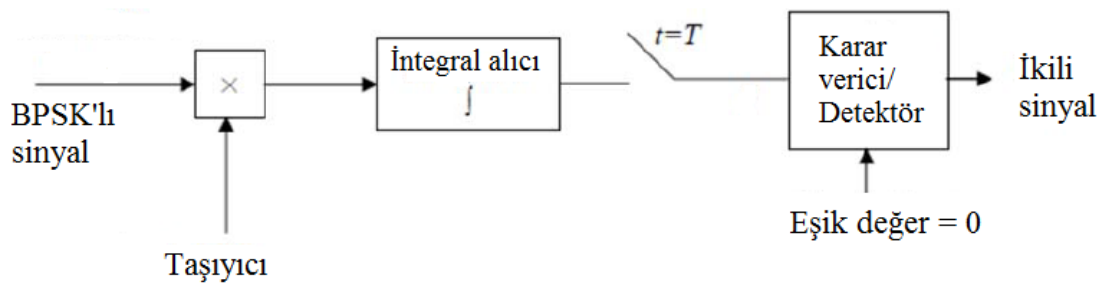
4.5. Sinyal Çözümleme

Sinyal çözümleme bölümü, uydulardan gelen sinyalin alıcı tarafından çözülmesinden seyrüsefer mesajını elde etme sürecine kadar gerçekleşen aşamaları içermektedir. Uydudan gelen RF ($1575,42 \text{ MHz} \pm \text{Doppler frekansı}$) sinyalin, frekansının çok yüksek olması ve temel bant işlemlerinin kolay yapılabilmesi için daha düşük frekansa sahip Ara Frekans (IF) sinyaline dönüştürülmesi gerekmektedir. Aşağı kaydırma (Down-converter), gelen RF sinyalin IF sinyale dönüştürülmesi için kullanılmaktadır. Bunun için gelen sinyal ile dahili osilatör tarafından üretilen $1573,42 \text{ MHz}$ frekanslı sinyalin, mikser yardımıyla çarpılması sonucu frekans spektrumunda $\text{RF} + 1573,42 \text{ MHz}$ ve $\text{RF} - 1573,42 \text{ MHz}$ frekanslı iki sinyal elde edilmektedir. Bu sinyallere, Alçak Geçiren Filtre uygulandığında $\text{RF} + 1573,42$ ($1575,42 \pm \text{Doppler} + 1573,42$) MHz 'li sinyal bastırılır ve sadece $\text{RF} - 1573,42$ ($1575,42 \pm \text{Doppler} - 1573,42$) MHz frekansa sahip IF sinyali geçer. RF sinyalinin IF sinyale dönüşmesi sırasında, seyrüsefer mesajı ve Doppler frekansında herhangi bir değişim veya hata söz konusu değildir. Sadece RF sinyalinin frekans değeri düşürülmektedir. Uygulamalarda donanım elemanları ile gerçekleştirilen aşağı kaydırma (Down-converter) işlemi, yazılım tabanlı simülasyon çalışması çerçevesinde yapay olarak MATLAB ortamında gerçekleştirilmektedir.

Aşağı kaydırma işlemi tamamlandıktan sonraki ilk adım sinyal yakalamadır. Sinyal yakalamanın ana amacı, görülebilen uyduların belirlenip bu uydulardan gelen sinyallerinin kod gecikmesini ve taşıyıcı frekansını belirlemektir. Yapılan çalışmada, diğer metotlara göre daha hızlı ve uygulaması açısından daha kolay olan Paralel Kod Fazı Arama yöntemi kullanılmıştır. Bu yöntemde 500 Hz aralıklarla üretilen taşıyıcı sinyal ve PRN kodları kullanılarak 41 ($10 \text{ kHz} / 500 \text{ Hz} + 1$) adımda sinyal yakalama aşaması tamamlanmaktadır. Gelen IF sinyale, Paralel Kod Fazı Arama yöntemi uygulandığında en yüksek korelasyon değerine sahip olan sonuç dikkate alınmakta ve buna göre gelen sinyalin PRN kodu, Doppler frekansı ve taşıyıcı frekansı belirlenmektedir. Daha sonra belirlenen PRN kodu ile dahili olarak üretilen aynı PRN kodu arasında oto korelasyon işlemi uygulanması sonucu kod gecikmesi tespit edilmektedir. Elde edilen gecikme süresi ışık hızı ile çarpıldığında alıcı ve uydu arasındaki mesafe hesaplanır. Bu mesafe bilgisi, alıcı konumunu belirlemek için son derece önemli bir parametredir.

Sinyal yakalama işlemi tamamlandıktan sonra gelen sinyaldeki seyrüsefer mesaj bitlerini belirlemek için haberleşme alanında ve özellikle de BPSK modülasyonunda

sıklıkla tercih edilen Faz-uyumlu Sinyal Algılama (Coherent Signal Detection) yapısı kullanılmıştır. Bu yöntemde gelen sinyalin frekans ve faz bilgisinin mutlaka bilinmesi gereklidir. Bu faz ve frekans bilgisi, genellikle bir PLL veya Costas döngüsünden elde edilir. Eğer sinyalin frekans ve fazı doğru olarak belirlenmiş ise Coherent yöntemi oldukça iyi sonuçlar vermektedir. Şekil 4.4’de, genel Coherent yapısının blok diyagramı gösterilmektedir. Buna göre, PLL veya Costas döngüsü kullanılarak gelen sinyalin frekansı belirlenir. Alıcıda dahili olarak aynı frekanslı bir sinyal üretilerek gelen sinyalle çarpılır. Elde edilen sinyalin bir PRN kodu uzunluğunda denk gelen 1 milisaniye boyunca integralinin alınması sonucu oluşan, önceden belirlenen eşik değeri olan 0’dan büyükse ise 1, küçük ise 0 olarak değerlendirilerek sinyal ikilik tabana dönüştürülür. Bu işlemin sonucunda elde edilen sinyal hem PRN kodu, hem de seyrüsefer mesajını içermektedir. Seyrüsefer mesajını elde etmek için sinyal PRN kodundan ayrılarak seyrüsefer mesajındaki hataları ayıklayıp doğru olarak belirlemek için sinyale Hamming kod çözme işlemi uygulanır ve bu işlemin sonucunda seyrüsefer mesajı elde edilir. Seyrüsefer mesajının ikilik tabandan onluk tabana dönüştürülmesi sonucu uyduların X, Y, Z koordinatları belirlenir. Alıcının konumunu hesaplayabilmesi için en az dört farklı uydudan sinyal alması gerektiği için bu adımlar dört farklı uydu için gerçekleştirilir. Uydulardan gelen seyrüsefer mesajından elde edilen dört farklı X, Y, Z uydu koordinatları ve uydu-alıcı arasındaki mesafe bilgileri kullanılarak alıcı konumu hesaplanabilir.



Şekil 4.4. Faz-uyumlu sinyal algılama blok diyagramı [54]

4.6. Pseudorange Hesabı

Alıcının konumunu hesaplamadan önce belirlenmesi gereken en önemli büyüklüklerden birisi alıcı ile uydu arasındaki mesafedir. Alıcı ve uydu saatlerinin senkronize olmaması durumunda ortaya çıkan hatalardan dolayı aralarındaki ölçülen

mesafe olması gerekenden farklı çıkmakta ve bu büyüklükte pseudorange olarak ifade edilmektedir.



Şekil 4.5. Hesaplanan sonuçlar

Pseudorange hesabı, gelen sinyalin çözümü sonucunda belirlenen kod gecikmesi kullanılarak hesaplanmaktadır. Kod gecikmesi, alıcı-uydu arasındaki mesafeden dolayı gönderilen sinyalin belirli bir gecikme ile alıcıya ulaşmasıdır. Boşlukta dalgaların ışık hızı ile hareket ettiği düşünüldüğünde pseudorange, kod gecikmesi ile ışık hızının çarpılması sonucu elde edilebilir. Pseudorange son derece önemli bir büyüklüktür. Çünkü konum hesabı yapılırken alıcının bilinmeyen koordinat bilgisi ve saat hatasını belirlemek için uydu koordinatları ile birlikte kullanılmaktadır. Şekil 4.5, simülasyon sonucu birbirinden farklı dört uydu için hesaplanan pseudorange değerlerini göstermektedir. Şekilde görüldüğü gibi, alıcı ile uydu arasındaki gerçek pseudorange mesafesi ve hesaplanan pseudorange büyüklüğü birbirinden farklıdır. Bu aradaki fark, alıcı saatinin sebep olduğu hatadan kaynaklanmaktadır.

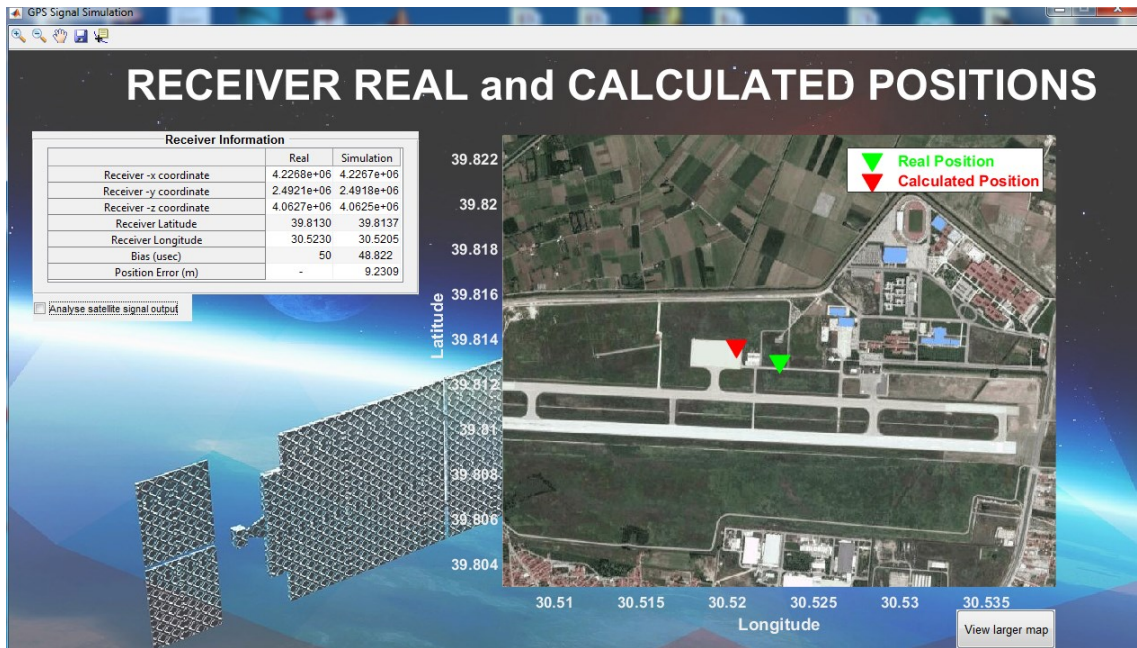
4.7. Konum Hesaplama

Bu bölümde, alıcının gerçek ve hesaplanan konum bilgisi arasında ortaya çıkan hata oranı karşılaştırılarak yapılan çalışmanın performansı analiz edilmektedir. Yukarıda anlatılan işlemlerin temel amacı, alıcının dünya üzerindeki konumunu belirlemektir. Konum hesabı ile ilgili detaylı bilgiler Bölüm 3.7’de verilmiştir. Bu bölümde, sadece simülasyon kapsamında yapılan işlemler ve alıcının gerçek ve hesaplanan koordinat, enlem-boylam, saat hatası değerlerinin karşılaştırılması yapılmaktadır.

Başlangıç parametrelerinde, alıcıya ait enlem-boylam bilgileri girilmektedir. Ondalık Derece birimli bu enlem-boylam bilgileri World Geodetic System 1984 (WGS84) koordinat sistemi kullanılarak kartezyen (X, Y, Z) koordinat sistemine dönüştürülür. Simülasyon tamamlandığında hem enlem- boylam, hem de X, Y, Z koordinat bilgileri ‘Gerçek ve Hesaplanan Konum (Real and Calculated Position)’ ekranındaki ‘Gerçek’ (Real)’ sekmesinde gösterilmektedir. ‘Simülasyon (Simulation)’ sekmesinde ise simülasyon sonucunda hesaplanan alıcının X, Y, Z koordinatları ile WGS84 dönüşümü kullanılarak bulunan enlem-boylam bilgileri gösterilmektedir.

Ayrıca başlangıçta alıcı saat hatası olarak atanan ‘t_bias’ değerinin hesaplamalar sonucu bulunan değeri de gösterilmektedir. Böylece gerçek ve hesaplanan değerler arasında bir kıyaslama yapılarak oluşan hata oranı belirlenebilmektedir.

Simülasyonun en son aşaması, enlem-boylam bilgisine dönüştürülen alıcı konumunun Google şirketinin sunduğu Google Statik Harita (Google Static Map) hizmeti kullanılarak harita üzerinde gösterilmesidir. Hem gerçek hem de hesaplanan alıcı konumu kullanılarak simülasyon sonucunda oluşan konum hatası Şekil 4.6'daki harita üzerinde gösterilmektedir. Ayrıca iki farklı enlem ve boylam arasındaki kuş bakışı mesafeyi ölçmede kullanılan Haversine formülü yardımıyla, gerçek ve hesaplanan konum arasındaki hatanın büyüklüğü bulunmaktadır. Bulunan bu konum hatası da 'Simülasyon' sekmesinde metre biriminde gösterilmektedir.



Şekil 4.6. Hesaplanan ve gerçek alıcı konumları

Yapılan çalışmada gürültünün konum hatasına olan etkisini inceleyebilmek için sinyale çeşitli SNR değerleri eklenerek konum belirlenmiştir. Gürültünün etkisini doğru olarak tespit edebilmek için Doppler kayması ile saat hatası her SNR değeri için sabit tutulmuştur. Simülasyon kapsamında sinyale uygulanan SNR değerlerinin konum hatasına olan etkisi Tablo 4.1'de verilmiştir. Tabloda görüldüğü üzere, SNR değeri arttıkça elde edilen konum hatasında artış meydana gelmektedir. Sinyalde SNR değeri düştükçe de konum hatası yaklaşık sıfır olarak hesaplanmaktadır.

Tablo 4.1. Gürültü analizi

Gerçek Alıcı Bilgileri						
SNR (dB)	0	-10	-25	-30	-35	-40
x-koordinatı	4227273,399	4227273,399	4227273,399	4227273,399	4227273,399	4227273,399
y-koordinatı	2491346,435	2491346,435	2491346,435	2491346,435	2491346,435	2491346,435
z-koordinatı	4062698,616	4062698,616	4062698,616	4062698,616	4062698,616	4062698,616
Enlem (Ondalık)	39,81300	39,81300	39,81300	39,81300	39,81300	39,81300
Boylam (Ondalık)	30,51299	30,51299	30,51299	30,51299	30,51299	30,51299
Saat hatası (μs)	50,0	50,0	50,0	50,0	50,0	50,0
Hesaplanan Alıcı Bilgileri						
SNR (dB)	0	-10	-25	-30	-35	-40
x-koordinatı	4227284,257	4227284,257	4227197,683	4226167,460	4226651,714	**
y-koordinatı	2491352,338	2491352,338	2491295,646	2490735,296	2491405,716	**
z-koordinatı	4062706,573	4062706,573	4062669,964	4061640,391	4062266,750	**
Enlem (Ondalık)	39,812983	39,812983	39,813326	39,812962	39,812927	**
Boylam (Ondalık)	30,512995	30,512995	30,512937	30,513408	30,517282	**
Saat hatası (μs)	50,040977	50,040977	49,698560	44,983175	48,390213	**
Konum hatası (m)	0,1043522	0,1043522	2,0908173	4,6449647	14,1069241	**

** Hesaplama sonuçlarının bulunamadığını ifade etmektedir.

4.8. Sinyal Analizi

Sinyal analizinin ana amacı, GPS sinyalinin üretim aşamasından seyrüsefer mesajının çözülmesine kadar gerçekleşen işlemler sonucunda elde edilen sonuçların detaylı bir şekilde incelenmesi ve simülasyonda yapılan herhangi bir değişikliğin sinyal sonucuna olan etkisinin gözlemlenmesidir. 4 uydu için gerçekleştirilen aşamaların aynı olmasının yanında MATLAB R2015b yazılımının veri kapasite sınırı nedeniyle sadece birinci uydunun sinyal çıkışlarının bazı bölümleri gösterilmektedir. Simülasyonun sonucunda verici ve alıcı kısmında elde edilen sinyal dalga formlarını göstermek üzere tasarlanan arayüz, Şekil 4.7’de gösterilmektedir. Bu amaçla, birinci uydu için gösterilen sinyal çıktıları ve ifade ettikleri anlamlar başlıklar halinde aşağıda açıklanmıştır.

Navigation Message: Seyrüsefer mesajı içerisinde girilen uydu (X, Y, Z) koordinatlarının ikilik tabana dönüştürülmesi sonucu oluşan sinyal gösterilmektedir.

Encoded Data: Kodlu seyrüsefer mesajı içerisinde oluşturulan seyrüsefer mesajına, hataları ayıklamak için kullanılan Hamming kodlama tekniği uygulanması sonucu oluşan sinyal gösterilmektedir.

PRN Code: Uydulara kimlik olarak atanan PRN kodunu göstermektedir.

Exclusive-OR: Elde edilen kodlu seyrüsefer mesajı ile PRN koduna XOR işlemi uygulanması sonucu oluşan sinyal gösterilmektedir.

BPSK: XOR işleminden sonra elde edilen sinyalin BPSK tekniği kullanılarak 1575,42 MHz frekanslı taşıyıcı üzerine bindirilmesi sonucu oluşan sinyal gösterilmektedir.

Noise: Uydu tarafından gönderilen BPSK'lı sinyalin boşlukta maruz kaldığı gürültü sonucu oluşan sinyal gösterilmektedir.

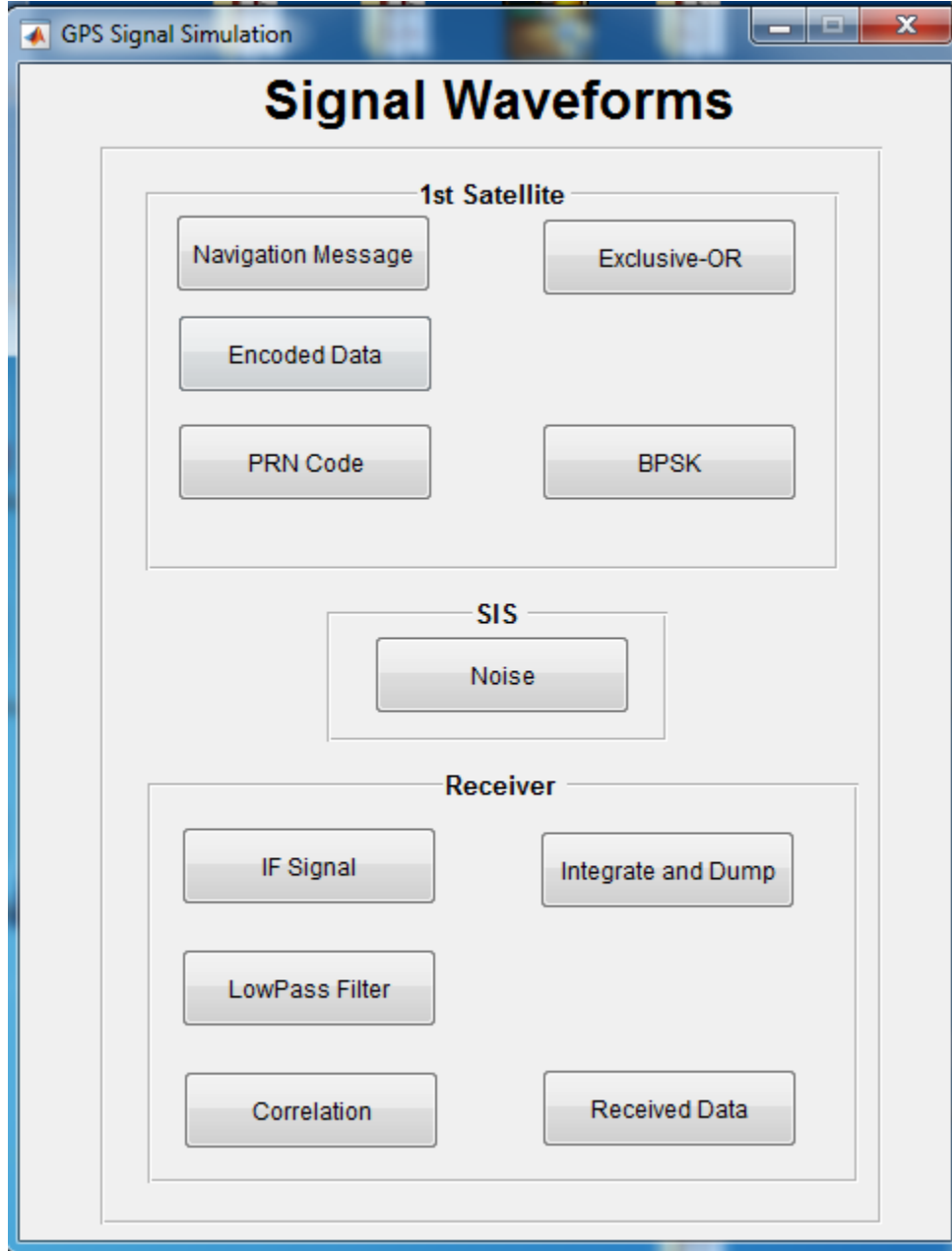
IF Signal: Yüksek frekanslı gelen sinyal ile alıcıdaki Dahili Osilatörün ürettiği sinyalin mikser kullanılarak çarpılması sonucu elde edilen düşük frekanslı IF sinyali gösterilmektedir.

Lowpass: Elde edilen IF sinyalini, gürültü ve diğer bozucu etkilerden arındırmak için Alçak Geçiren Filtre uygulanması sonucu oluşan sinyal gösterilmektedir.

Correlation: Uydunun PRN kodu ile alıcıda üretilen aynı PRN kodunun korelasyon işlemi yapılması sonucu elde edilen sonuç gösterilmektedir. Bu sayede gelen sinyalin PRN kodunda meydana gelen gecikme belirlenebilmektedir.

Integrate and Dump: Gelen sinyalin 1 milisaniye süre ile integrali alınarak gürültüden arındırılması sonucu Coherent detektörün çıkışında Topla & Boşalt (Integrate and Dump) kullanılarak 1 veya 0 olmasına karar verilmesi sonucunda elde edilen sinyal gösterilmektedir.

Received Data: Seyrüsefer mesajının PRN kodundan ayrılması ve Hamming tekniği kullanılarak oluşan hataların ayıklanması sonucu alıcının elde ettiği sinyal gösterilmektedir.



Şekil 4.7. Sinyal analizi

Örnek bir simülasyonun çalışmasının gerçekleştirilmesi sonucunda 1. uydu için elde edilen sinyal çıktıları EK-1’de verilmiştir.

5. SONUÇ

GPS, ABD tarafından geliştirilen ve günümüzde 31 operasyonel uydu ile küresel boyutta hizmet veren Küresel Uydu Seyrüsefer Sistemi (GNSS)'dir. Kullanıcının konum, hız ve zaman gibi önemli bilgileri GPS uydularının gönderdiği sinyaller kullanılarak belirlenmektedir. Günlük hayatta sıklıkla kullanılan GPS; L1, L2 ve L5 olmak üzere üç farklı frekans bandında yayın yapmaktadır.

Bu çalışmada gelecekte havacılık uygulamalarında yaygın bir şekilde kullanılacak olan GPS sisteminin genel çalışma prensibi, sinyal yapısı ve alıcı yapısı incelenmiştir. Bu kapsamda sivil kullanıcıların erişimine açık olması nedeniyle seçilen GPS L1 sinyalinin üretilmesi, alıcıda çözülmesi ve elde edilen sonuçları göstermek üzere kullanıcı arayüzünün geliştirilmesi amaçlanmıştır.

GPS L1 sinyalinin üretimi, çözülmesi ve kullanıcı arayüzünün geliştirilmesi aşamalarında matematiksel problemlerin çözümü ve analizi için uygun bir yazılım aracı olan R2015b versiyonuna sahip MATLAB ile MATLAB-GUI'den yararlanılmıştır. Bu çalışmada L1 sinyalinin simülasyonu hem verici hem de alıcı düzeyinde gerçekleştirilmiştir. Geliştirilen arayüz aracılığıyla, dört farklı uyduda üretilen L1 sinyalinin alıcıda çözülmesi sonucunda hesaplanan alıcı konumu ile gerçek alıcı konumu harita üzerinde karşılaştırılmış; ortam gürültüsü, alıcı saat hatası ve Doppler kayması kavramlarının konum hatasına etkileri irdelenmiştir. Ayrıca tasarlanan arayüz ile L1 sinyalinin üretimi ve çözülmesi aşamalarında gerçekleştirilen işlemler sonucunda elde edilen sinyal çıktılarını inceleme imkanı sunulmuştur.

Yapılan çalışma kapsamında elde edilen sonuçlar değerlendirildiğinde konum hatasının, doğrudan sinyalin maruz kaldığı gürültü oranına bağlı olarak değiştiği bunun yanında hesaplama yolu ile bulunabilen alıcı saat hatası ve Doppler frekansının da bu hataya kısmen katkı sağladığı gözlemlenmiştir.

Sürekli olarak modernize edilen GPS sistemi için literatür araştırması dikkate alındığında GPS L1 sinyali için tasarlanan bu çalışmanın bugüne kadar yapılan diğer çalışmalardan farklılıkları şu şekilde özetlenebilir;

- GPS L1 sinyali için hem alıcı, hem de verici düzeyinde hiçbir donanım elemanı kullanılmadan tamamen yazılım tabanlı simülasyon geliştirilmesi

- Simülasyonda kullanıcının gürültü değeri girmesi sağlanarak gürültünün sinyale olan etkisinin incelenmesi
- Sinyal üretiminden alıcıda seyrüsefer mesajının çözülmesine kadar olan aşamalarda elde edilen sinyal dalga formlarının gösterilmesi
- Doppler frekansı ve alıcı saat hatası dikkate alınarak çalışmaya dahil edilmesi
- Hesaplanan ve gerçek konumun harita üzerinde gösterilmesi

Bu çalışma, gerçek zamanlı GPS sinyallerini kullanarak alıcının konum ve hız değerlerini bulan bir yazılım tabanlı GPS alıcısının geliştirilmesine temel teşkil etmektedir. Bu çalışmanın devamında, GNSS uygulamaları kapsamında GLONASS, Galileo ve COMPASS gibi diğer uydu seyrüsefer sistemlerini de kapsayacak şekilde yazılım tabanlı bir GNSS alıcısının geliştirilmesi düşünülebilir.

KAYNAKÇA

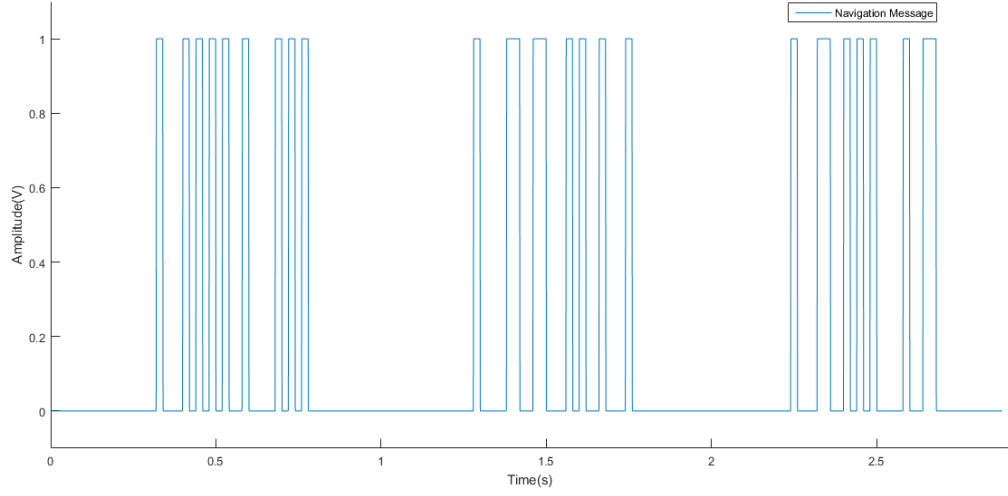
- [1] Pace, S., Frost, G., Lachow, I., Frelinger, D., Fossum, D., Wassem, D., Pinto, M. (1995). *The global positioning system: assessing national policies*. Santa Monica: RAND Corporation.
- [2] <http://literature.cdn.keysight.com/litweb/pdf/5991-2288EN.pdf> (Eriřim tarihi: 07.12.2017)
- [3] Arslan, E. (2016). *Galileo uydu seyrüsefer sistemi E5 sinyal simülasyonu*. Yüksek Lisans Tezi. Eskişehir: Anadolu Üniversitesi, Fen Bilimleri Enstitüsü.
- [4] Subedi, P. (2013). *Software simulator and signal analysis for Galileo E5 band signals*. Master of Science Thesis. Finland: Tampere University of Technology.
- [5] Zheng, S. (2005). *Signal acquisition and tracking for a software GPS receiver*. Master of Science Thesis. Blacksburg: Virginia Polytechnic Institute and State.
- [6] Kahveci, M. ve Yıldız, F. (2012). *GPS/GNSS uydularla konum belirleme sistemleri teori ve uygulama*. Ankara: Nobel Akademik Yayıncılık.
- [7] Bao, J. and Tsui, Y. (2000). *Fundamentals of global positioning system receivers: a software approach*. New York: John Wiley & Sons.
- [8] El-Rabbany, A. (2002). *Introduction to GPS - the global positioning system*. United States of America (USA): Artech House.
- [9] Siddiqui, B. (2009). *Simulink-based acquisition unit for Galileo E1 CBOC modulated signals*. Master of Science Thesis. Finland: Tampere University of Technology.
- [10] <https://www.gps.gov/systems/gps/space/> (Eriřim tarihi: 07.12.2017)
- [11] Milli Eğitim Bakanlığı. (2011). *GPS*. Ankara: Milli Eğitim Yayınları.
- [12] <https://sites.google.com/a/teknometcom.com/index/Gps-Nedir/uzay-boeluemue> (Eriřim tarihi: 01.12.2017)
- [13] Zhang, J. (2013). *Advanced signal processing in multi-mode multi-frequency receivers for positioning applications*. PhD Thesis. Finland: Tampere University of Technology.
- [14] Kaplan, E. and Hegarty, C. (2006). *Understanding GPS principles and applications*. London: Artech House.
- [15] Prasad, R. and Ruggieri, M. (2005). *Applied satellite navigation using GPS, Galileo, and augmentation systems*. London: Artech House.
- [16] <https://www.gps.gov/systems/gps/control/> (Eriřim tarihi: 07.12.2017)
- [17] Maini, A. and Agrawal, V. (2011). *Satellite technology principles and applications*. United Kingdom: John Wiley & Sons.

- [18] Elango, G. A. and Sudha, G. F. (2016). Design of complete software GPS signal simulator with low complexity and precise multipath channel model. *Journal of Electrical Systems and Information Technology* 3, 161-180.
- [19] Tomé, G. (2015). *Post-processed acquisition & tracking of GPS C/A L1 signals: a software-defined receiver approach*. Master of Science Thesis. Lisbon: Técnico Lisboa. Department of Electrical and Computer Engineering.
- [20] Landry, R. and Ilie, I. (2015). Software-defined GNSS simulator: a step forward. *InsideGNSS*, 54-60.
- [21] Soghoyan, A., Suleiman, A. and Akopian, D. (2014). A development and testing instrumentation for GPS software defined radio with fast FPGA prototyping support. *IEEE Transactions on instrumentation and Measurement* (s. 2001-2012). IEEE.
- [22] Abhishek, C., Charitha, A. and Goutham, D. (2014). GPS receiver implementation using simulink. *International Journal of Advance Engineering and Research Development*, 1-5.
- [23] Zubaroğlu, T. (2013). *Yazılım tabanlı GPS almaçlarının incelenmesi*. Yüksek Lisans Tezi. Ankara: Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü.
- [24] Akopian, D. and Soghoyan, A. (2013). A labview-based fast prototyping software defined GPS receiver platform. *Global Conference on Signal and Information Processing* (s. 1230-1233). Austin: IEEE.
- [25] Kassas, Z., Bhatti, J. and Humphreys, T. (2013). A graphical approach to GPS software-defined receiver implementation. *Global Conference on Signal and Information Processing* (s. 1226-1229). Austin: IEEE.
- [26] Guo, J., Wang, W. and Chao, B. (2012). Model of GPS IF signal and its simulation. *Geo-spatial Information Science*, 100-103.
- [27] Bright, M. (2012). *GPS L2 C signal survey and the development of the emergent MATLAB L2C (emal2) receiver*. Master of Science Thesis. Austin: The University of Texas, Department of Engineering.
- [28] Rao, M. and Falco, G. (2012). SDR joint GPS/Galileo receiver from theory to practice. *International Journal of Aerospace Sciences*, 1-7.
- [29] Brown, A., Tredway, R. and Taylor, R. (2011). GPS signal simulation using open source GPS receiver platform. *21st Virginia Tech Symposium on Wireless Personal Communications*, (s. 1-6). Virginia
- [30] Peña, A. (2010). *Optimization of demodulation performance of the GPS and Galileo navigation message*. PhD Thesis. Toulouse: Université de Toulouse, l'Institut National Polytechnique de Toulouse.
- [31] Zheng, Y. (2010). A software-based frequency domain parallel acquisition algorithm for GPS signal. *Anti-Counterfeiting Security and Identification in Communication* (s. 298-301). Chengdu: IEEE.

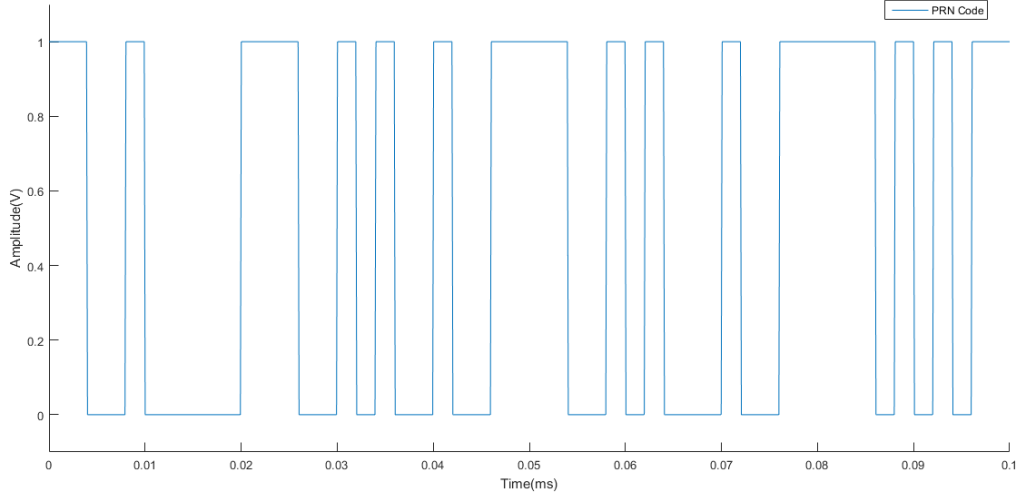
- [32] Wei, Z., Ke, Z., Binbin, W. and Heejong, S. (2009). Simulation and analysis acquisition of GPS c/a code signals in GPS system. *Computer Network and Multimedia Technology*. Wuhan: IEEE.
- [33] Günaydın, E. (2005). Implementation of software GPS receiver. Master of Science Thesis. Ankara: Middle East Technical University, Institute of Science.
- [34] Ma, C., Lachapelle, G. and Cannon, M. (2004). Implementation of a software GPS receiver. *ION GNSS 2004* (s. 1-15). California: ION.
- [35] Dong, L. (2003). *IF GPS signal simulator development and verification*. Master of Science Thesis. Canada: University of Calgary, Department of Geomatics Engineering.
- [36] Brown, A., Gerein, N. and Taylor, K. (2000). Modeling and simulation of GPS using software signal generation and digital signal reconstruction. *ION National Technical Meeting*,. Anaheim: ION.
- [37] Xiong, F. (2006). *Digital modulation techniques*. United States of America (USA): Artech House.
- [38] U.S. Department of Defence. (1995). *Global positioning system standart positioning service signal specification* (2.nd Edition). Washington D.C.
- [39] Weeks, M. (2007). *Digital signal processing using matlab and wavelets*. Hingham: Infinity Science Press LLC.
- [40] Hoffman-Wellenhof, B., Lichtenegger, H. and Wasle, E. (2008). *GNSS-global navigation satellite systems GPS, GLONASS, Galileo & more*. New York: Springer.
- [41] GPS Directorate. (2013). *Systems engineering & integration interface specification (IS-GPS-200H)*. El Sugonda.
- [42] ARINC Research Corporation. (2000). *Interface control document (ICD-GPS 200C)*, El Sugonda.
- [43] Borre, K., Akos, D., Bertelsen, N., Rinder, P. and Jensen, S. (2007). *A software-defined GPS and Galileo receiver: a single-frequency approach*. Boston: Birkhauser.
- [44] http://www.navipedia.net/index.php/Generic_Receiver_Description (Erişim tarihi: 13.12.2017)
- [45] Betz, J. W. (2016). *Engineering satellite-based navigation and timing*. New Jersey: John Wiley & Sons.
- [46] Doberstein, D. (2012). *Fundamentals of GPS receiver: a hardware approach*. New York: Springer.
- [47] http://cache.freescale.com/files/sensors/doc/app_note/AN5087.pdf (Erişim tarihi: 15.12.2017)

- [48] El-Sheimy, N., Hou, H. and Niu, X. (2008). Analysis and modeling of inertial sensors using Allan variance. *IEEE Transactions on Instrumentation and Measurement*, 140-149.
- [49] Milli Eğitim Bakanlığı. (2012). *ADC-DAC devreleri*. Ankara: Milli Eğitim Yayınları.
- [50] Skanda, V. (2012). *Sine wave generator using numerically controlled oscillator module*. ABD: Microchip Technology Inc.
- [51] Satyanarayana, R., & Neethu, S. (2016). Analysis and implementation of Reed Salomon codes for Forward Error Correction using LabVIEW. *International Research Journal of Engineering and Technology (IRJET)*, 1057-1065.
- [52] Lint, J. (1999). *Introduction to coding theory*. Heidelberg: Springer.
- [53] Joseph, A. (2010). Measuring GNSS signal strength. *InsideGNSS*, 20-25.
- [54] http://muhendislik.istanbul.edu.tr/elektrikelektronik/wp-content/uploads/2013/12/iletisim_sayisal_modulasyon.pdf (Erişim tarihi: 15.12.2017)

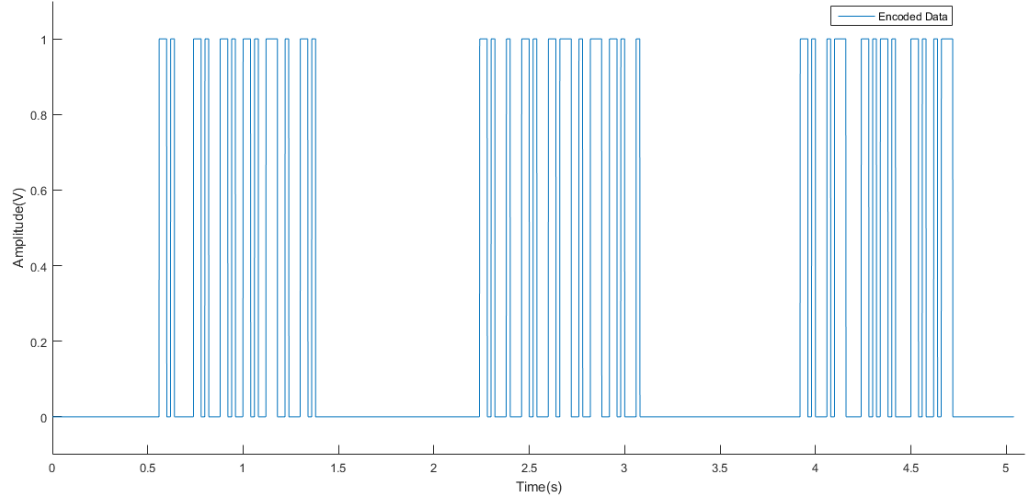
EK -1: SİNYAL ANALİZİ DALGA FORMLARI



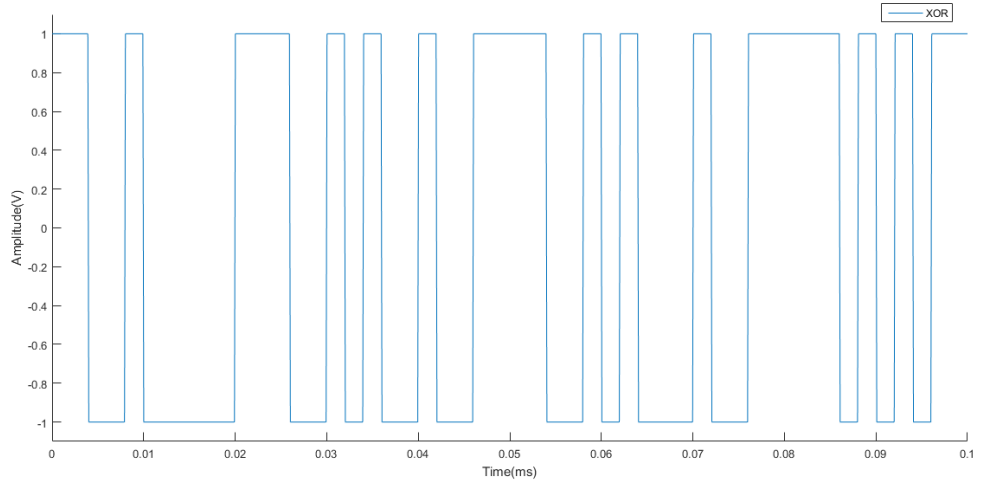
Şekil E1. 1. Seyrüsefer mesajı



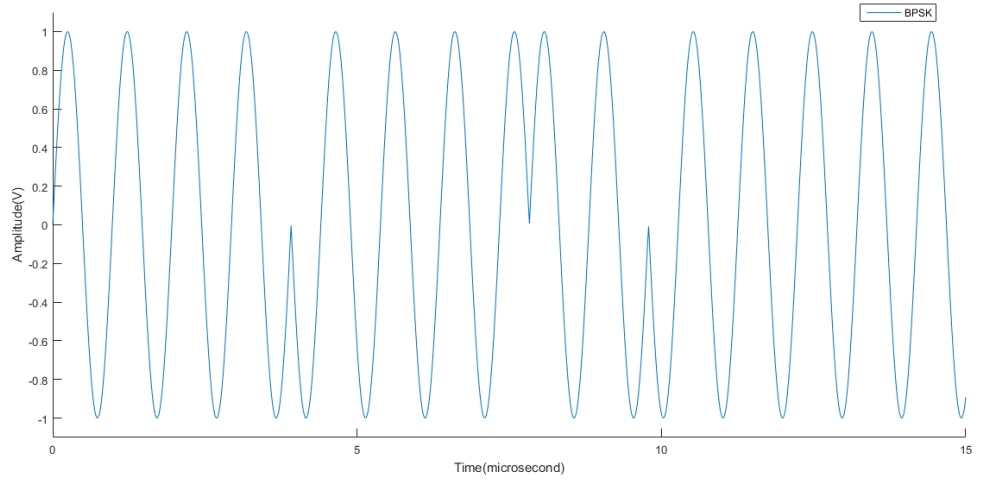
Şekil E1. 2. PRN kodu



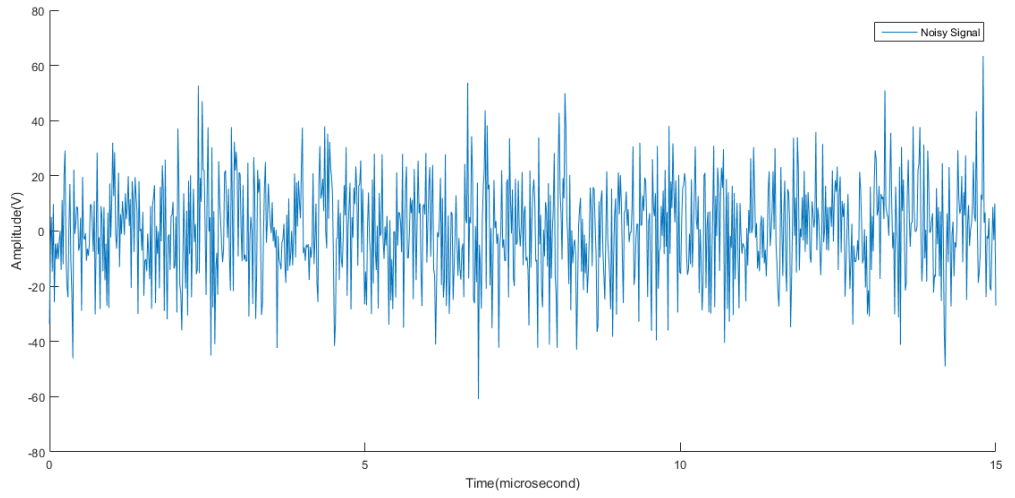
Şekil E1. 3. Kodlu seyrüsefer mesajı



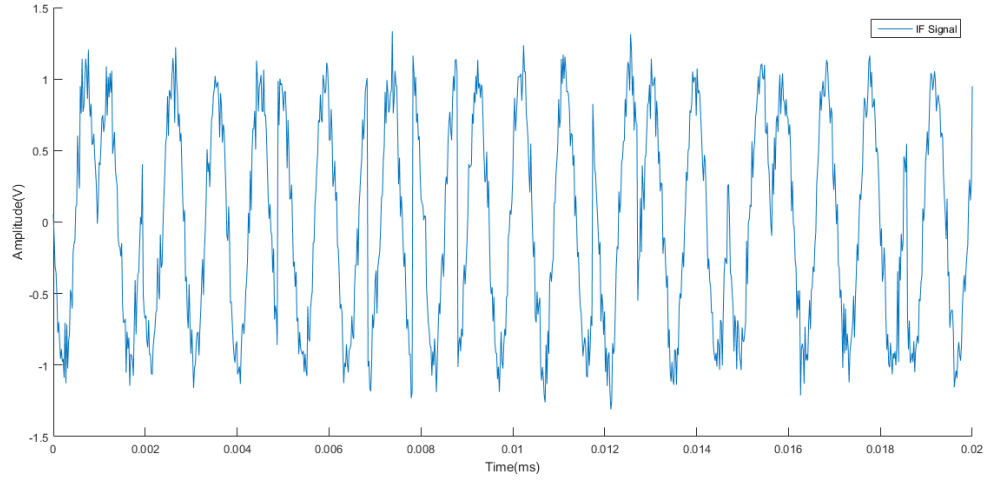
Şekil E1. 4. XOR



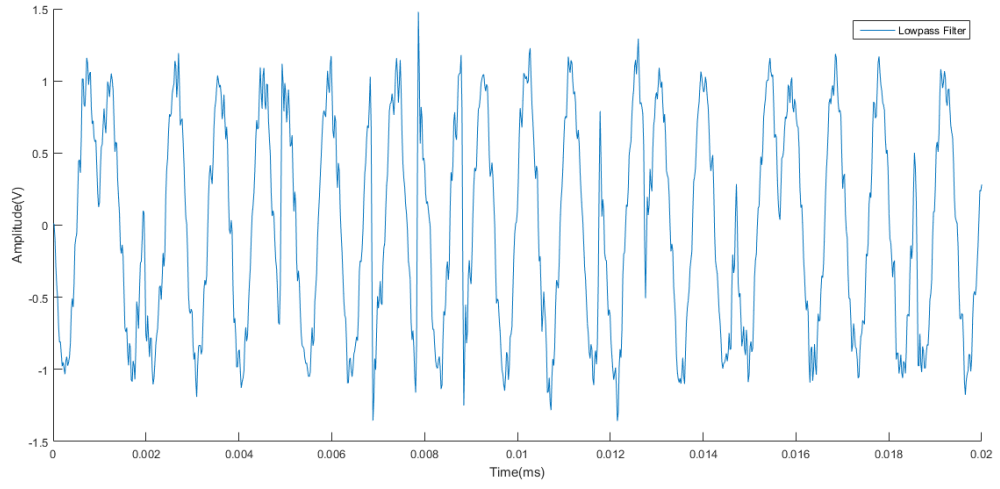
Şekil E1. 5. BPSK



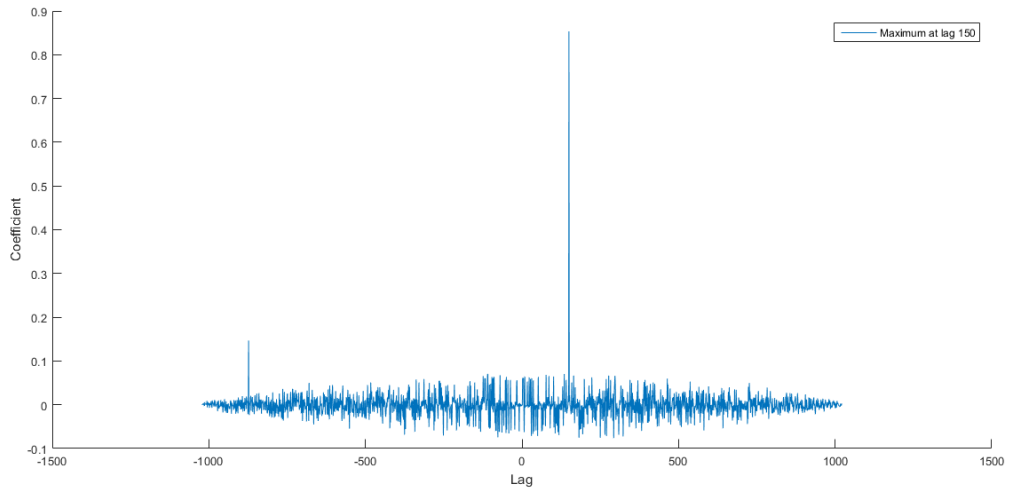
Şekil E1. 6. Gürültülü sinyal



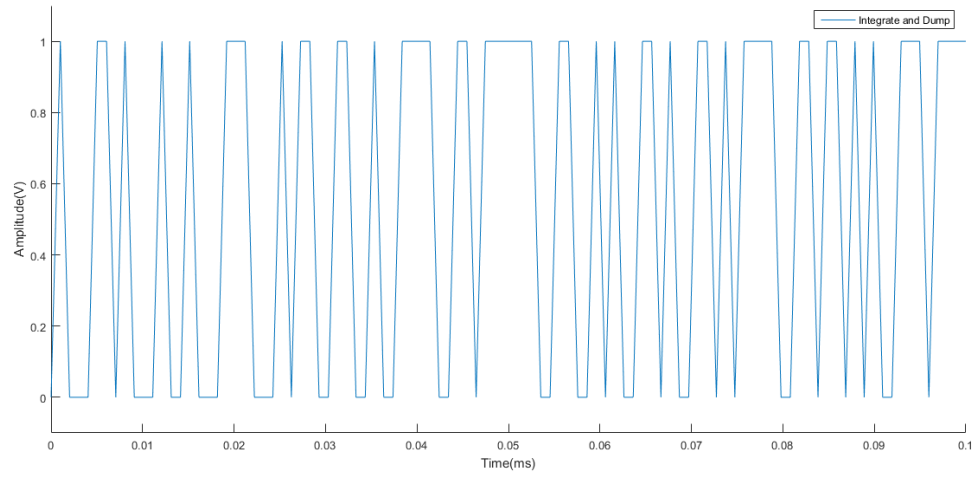
Şekil E1. 7. IF sinyali



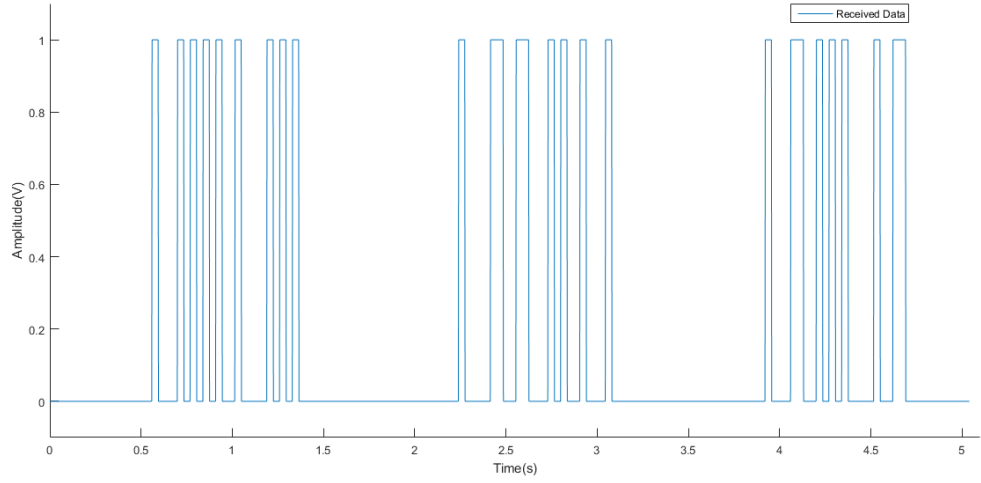
Şekil E1. 8. Alçak geçiren filtre



Şekil E1. 9. Korelasyon



Şekil E1. 10. *Topla & boşalt*



Şekil E1. 11. *Alınan mesaj*

EK -2A: SİMÜLASYON KODLARI (Anasayfa)

```
function varargout = Home_page(varargin)

% HOME_PAGE MATLAB code for Home_page.fig

%   HOME_PAGE, by itself, creates a new HOME_PAGE or raises the existing
%   singleton*.

%
%   H = HOME_PAGE returns the handle to a new HOME_PAGE or the handle to
%   the existing singleton*.

%
%   HOME_PAGE('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in HOME_PAGE.M with the given input
%   arguments.

%
%   HOME_PAGE('Property','Value',...) creates a new HOME_PAGE or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before Home_page_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to Home_page_OpeningFcn via varargin.

%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".

%
% See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text to modify the response to help Home_page
% Last Modified by GUIDE v2.5 22-Nov-2017 10:31:27
% Begin initialization code - DO NOT EDIT

gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @Home_page_OpeningFcn, ...
```

```

        'gui_OutputFcn', @Home_page_OutputFcn, ...
        'gui_LayoutFcn', [] , ...
        'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Home_page is made visible.
function Home_page_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Home_page (see VARARGIN)

% Choose default command line output for Home_page
handles.output = hObject;

% create an axes that spans the whole gui
ah = axes('unit', 'normalized', 'position', [0 0 1 1]);
% import the background image and show it on the axes
bg = imread('mission.jpg');
% bg= imrotate(bg,180);
c=imagesc(bg);

```

```

set(c, 'AlphaData', 1);

% % prevent plotting over the background and turn the axis off
% set(ah,'handlevisibility','off','visible','off')

% % making sure the background is behind all the other uicontrols
% uistack(ah, 'bottom');

% set(handles.text1,'BackgroundColor',get(ah,'Color'));
set(ah,'visible','off')

text(90,45,'GPS Signal Simulation
','Color','w','FontSize',20,'Fontweight','bold','BackgroundColor','none','parent',ah);

% text(180,85,'Interface (GPS
SSI)','Color','w','FontSize',18,'Fontweight','bold','BackgroundColor','none','parent',ah);

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Home_page wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Home_page_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)

```

```

% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
Main_Screen();

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

EK -2B: SİMÜLASYON KODLARI (Sonuç Ekranı)

```
function varargout = Final_Screen(varargin)

% FINAL_SCREEN MATLAB code for Final_Screen.fig

%   FINAL_SCREEN, by itself, creates a new FINAL_SCREEN or raises the existing
%   singleton*.

%

%   H = FINAL_SCREEN returns the handle to a new FINAL_SCREEN or the
%   handle to

%   the existing singleton*.

%

%   FINAL_SCREEN('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in FINAL_SCREEN.M with the given input
%   arguments.

%

%   FINAL_SCREEN('Property','Value',...) creates a new FINAL_SCREEN or raises
%   the

%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before Final_Screen_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to Final_Screen_OpeningFcn via varargin.

%

%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".

%

% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Final_Screen

% Last Modified by GUIDE v2.5 22-Nov-2017 09:46:38

% Begin initialization code - DO NOT EDIT
```



```

gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @Final_Screen_OpeningFcn, ...
                  'gui_OutputFcn', @Final_Screen_OutputFcn, ...
                  'gui_LayoutFcn', [] , ...
                  'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Final_Screen is made visible.
function Final_Screen_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Final_Screen (see VARARGIN)

% Choose default command line output for Final_Screen
handles.output = hObject;

% create an axes that spans the whole gui

```

```

ah = axes('unit', 'normalized', 'position', [0 0 1 1]);
% import the background image and show it on the axes
bg = imread('mission.jpg');
c=imagesc(bg);
set(c, 'AlphaData', .75);
% prevent plotting over the background and turn the axis off
set(ah,'handlevisibility','off','visible','off')
% making sure the background is behind all the other uicontrols
uistack(ah, 'bottom');
set(ah,'visible','off')

text(100,30,'RECEIVER REAL and CALCULATED POSITIONS
','Color','w','FontSize',35,'Fontweight','bold','BackgroundColor','none','parent',ah);

real_x_cor_rec = getappdata(0,'real_x_cor_rec'); % Real Receiver Coord. Value
real_y_cor_rec = getappdata(0,'real_y_cor_rec');
real_z_cor_rec = getappdata(0,'real_z_cor_rec');
real_t_bias = getappdata(0,'real_t_bias'); % Real T_bias Value

first_satellite = getappdata(0,'first_satellite');
% first_satellite=evalin('base','first_satellite');

calc_sat1_cor_x = first_satellite.sat_cor(1);
calc_sat1_cor_y = first_satellite.sat_cor(2);
calc_sat1_cor_z = first_satellite.sat_cor(3);
calc_sat1_pseudorange = first_satellite.pseudorange;

second_satellite = getappdata(0,'second_satellite');
% second_satellite=evalin('base','second_satellite');

```

```

calc_sat2_cor_x = second_satellite.sat_cor(1);
calc_sat2_cor_y = second_satellite.sat_cor(2);
calc_sat2_cor_z = second_satellite.sat_cor(3);
calc_sat2_pseudorange = second_satellite.pseudorange;

```

```

third_satellite = getappdata(0,'third_satellite');
% third_satellite=evalin('base','third_satellite');

```

```

calc_sat3_cor_x = third_satellite.sat_cor(1);
calc_sat3_cor_y = third_satellite.sat_cor(2);
calc_sat3_cor_z = third_satellite.sat_cor(3);
calc_sat3_pseudorange = third_satellite.pseudorange;

```

```

fourth_satellite = getappdata(0,'fourth_satellite');
% fourth_satellite=evalin('base','fourth_satellite');

```

```

calc_sat4_cor_x = fourth_satellite.sat_cor(1);
calc_sat4_cor_y = fourth_satellite.sat_cor(2);
calc_sat4_cor_z = fourth_satellite.sat_cor(3);
calc_sat4_pseudorange = fourth_satellite.pseudorange;

```

```

calc_rec_cor = calculationReceivercoordinate (calc_sat1_cor_x, calc_sat1_cor_y,
calc_sat1_cor_z,calc_sat2_cor_x, calc_sat2_cor_y, calc_sat2_cor_z,...
    calc_sat3_cor_x, calc_sat3_cor_y, calc_sat3_cor_z, calc_sat4_cor_x, calc_sat4_cor_y,
calc_sat4_cor_z,calc_sat1_pseudorange, calc_sat2_pseudorange,...
    calc_sat3_pseudorange, calc_sat4_pseudorange );

```

% Latitude, Longitude and Altitude Conversion to Cartesian Coordinate

```

a= 6378137.0 ;% semi-major axis (m)
b= 6356752.3141403; % semi-minor axis (m)

```

```

f = (a-b)/a; % flattening
e2= 2*f-f^2 ; % eccentricity squared
h = 1000 ; % ellipsoid height (m)
eover=(a^2-b^2)/b^2;

p_real = (real_x_cor_rec^2 + real_y_cor_rec^2 )^.5;
teta_real = atan((real_z_cor_rec*a)/(p_real*b));
real_rec_lat = atan((real_z_cor_rec+eover*b*(sin(teta_real))^3)/(p_real-
e2*a*(cos(teta_real))^3));
real_rec_lon = atan2(real_y_cor_rec,real_x_cor_rec);

real_rec_lat_degree = real_rec_lat*180/pi;
real_rec_lon_degree = real_rec_lon*180/pi;

p_calc = (calc_rec_cor(1)^2 + calc_rec_cor(2)^2 )^.5;
teta_calc = atan((calc_rec_cor(3)*a)/(p_calc*b));
calc_rec_lat = atan((calc_rec_cor(3)+eover*b*(sin(teta_calc))^3)/(p_calc-
e2*a*(cos(teta_calc))^3));
calc_rec_lon = atan2(calc_rec_cor(2),calc_rec_cor(1));

calc_rec_lat_degree = calc_rec_lat*180/pi;
calc_rec_lon_degree = calc_rec_lon*180/pi;

handles.real_lat = real_rec_lat_degree; guidata(hObject, handles);
handles.real_lon = real_rec_lon_degree; guidata(hObject, handles);
handles.calc_lat = calc_rec_lat_degree; guidata(hObject, handles);
handles.calc_lon = calc_rec_lon_degree; guidata(hObject, handles);

%% Law of Haversine % Two point (latitude and longitude) distance

R= 6371 ; % R= 6371 km is radius of earth.

```

```

a = sin((handles.real_lat-handles.calc_lat)/2)^2 +
cos(handles.real_lat)*cos(handles.calc_lat)*sin((handles.real_lon-
handles.calc_lon)/2)^2;

c=2*atan2(sqrt(a),sqrt(1-a));

d=R*c;

position_error=d;

calc_t_bias=calc_rec_cor(4)/3e8; % calc_rec_cor (4) is meter.
calc_t_bias_microsecond = calc_t_bias*1e6;


setappdata(0,'real_x_cor_rec',real_x_cor_rec);
setappdata(0,'real_y_cor_rec',real_y_cor_rec);
setappdata(0,'real_z_cor_rec',real_z_cor_rec);
setappdata(0,'calc_rec_x',calc_rec_cor(1));
setappdata(0,'calc_rec_y',calc_rec_cor(2));
setappdata(0,'calc_rec_z',calc_rec_cor(3));


setappdata(0,'real_rec_lat',real_rec_lat_degree);
setappdata(0,'real_rec_lon',real_rec_lon_degree);
setappdata(0,'calc_rec_lat',calc_rec_lat_degree);
setappdata(0,'calc_rec_lon',calc_rec_lon_degree);


setappdata(0,'real_t_bias',real_t_bias);
setappdata(0,'calc_t_bias',calc_t_bias_microsecond);
setappdata(0,'p_error',position_error);


final_res = {real_x_cor_rec, calc_rec_cor(1); real_y_cor_rec, calc_rec_cor(2);
real_z_cor_rec, calc_rec_cor(3);real_rec_lat_degree, calc_rec_lat_degree;...
real_rec_lon_degree,calc_rec_lon_degree;real_t_bias , sprintf('
%1.3f',calc_t_bias_microsecond);'          -',position_error};

```

```

set(handles.uitable2,'data',final_res);
axes(handles.axes2); % axes2 eksenini için işleme başlama komutu
plot(real_rec_lon_degree,real_rec_lat_degree, 'vg','markersize',15,'markerfacecolor','g');

% text(real_rec_lon_degree-100/1e5,real_rec_lat_degree+45/1e5,'Real
Position','Fontweight','bold','FontSize',10,'Color','b')

set(handles.axes2,'XColor',[0.941 0.941 0.941],'fontweight','bold','FontSize',13);
set(handles.axes2,'YColor',[0.941 0.941 0.941],'fontweight','bold','FontSize',13);
ylabel('Latitude'); xlabel('Longitude');

hold on ;

plot(calc_rec_lon_degree,calc_rec_lat_degree, 'vr','markersize',15,'markerfacecolor','r');

% text(calc_rec_lon_degree-300/1e5,calc_rec_lat_degree-45/1e5,'Calculated
Position','Fontweight','bold','FontSize',10,'Color','r')

% r_defined=0.0025;
r_defined = 0.01;
xlim([real_rec_lon_degree-r_defined real_rec_lon_degree+r_defined ])
ylim([real_rec_lat_degree-r_defined real_rec_lat_degree+r_defined]);

show_map();

hold off ;

lgnd=legend({'\color {green} Real Position','\color{red} Calculated
Position'},'FontSize',13);
% lgnd=legend('\color[rgb]{0.929,0.694,0.125} Real Position','\color{red} Calculated
Position','FontSize',10);
% set(lgnd,'color','none');

```

```

% set( lgnd, 'Box', 'off' ) ;

% Update handles structure
guidata(hObject, handles);
% UIWAIT makes Final_Screen wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Final_Screen_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

real_description = sprintf('%f %f',handles.real_lat,handles.real_lon );
real_name = 'Real position';

calc_description = sprintf('%f %f',handles.calc_lat,handles.calc_lon );
calc_name = 'Calculated position';

wmmarker(handles.real_lat, handles.real_lon, 'Description', real_description, ...

```

```

        'FeatureName', real_name,...
        'OverlayName', real_name,...
        'Color','g');

wmmarker(handles.calc_lat, handles.calc_lon, 'Description', calc_description, ...
        'FeatureName', calc_name,...
        'OverlayName', calc_name,...
        'Color','r');

% --- Executes on button press in checkbox1.
function checkbox1_Callback(hObject, eventdata, handles)
% hObject    handle to checkbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox1
if (get(handles.checkbox1,'Value')==1)
    Sub_Screen_2();
else
    close(Sub_Screen_2())
end

```


EK -2C: SİMÜLASYON KODLARI (Konum Hesaplama)

```
function [output] = calculationReceivercoordinate(sat1_x, sat1_y, sat1_z, sat2_x, sat2_y,
sat2_z, sat3_x, sat3_y, sat3_z, sat4_x, sat4_y, sat4_z,
sat1_pse, sat2_pse, sat3_pse, sat4_pse)

nsat=4;

% sp(1:3, 1:nsat); % satellite position which has the following format
%
% pr(1:nsat); % is the measured pseudo-range which has the format as
% % pr=[pr1 pr2 pr3 ... prnn]T;

sp = [sat1_x sat1_y sat1_z; sat2_x sat2_y sat2_z; sat3_x sat3_y sat3_z; sat4_x sat4_y
sat4_z]';

pr = [sat1_pse sat2_pse sat3_pse sat4_pse];

nn=nsat; % is the number of satellites

% ***** Select initial guessed positions and clock bias *****
x_guess=0; y_guess=0; z_guess=0; bu=0;

gu(1)=x_guess; gu(2)=y_guess; gu(3)=z_guess;

% Calculating rao the pseudo-range as shown in Equation (2.1) the
% clock bias is not included

for j=1:nsat
rao(j)=((gu(1)-sp(1,j))^2+(gu(2)-sp(2,j))^2+(gu(3)-sp(3,j))^2)^.5;
end

% generate the fourth column of the alpha matrix in Eq. 2.15
alpha(:,4)=ones(nsat,1);

error=1;

while error>1e-2;

for j=1:nsat;

for k=1:3;

alpha(j,k)=(gu(k)-sp(k,j))/(rao(j)); % find first
%3 columns of alpha matrix

end

end
```

```

drao= pr - (rao + ones(1,nsat)*bu);%** find delta rao
% includes clock bias
drao=drao';
% dl=pinv(alpha'*alpha)*alpha'*drao; % Equation (2.16)
dl=pinv(alpha)*drao; % Equation (2.16)
bu= bu + dl(4); % new clock bias
for k=1:3;
gu(k)=gu(k) + dl(k); %**find new position
end
% error=sqrt(dl(1)^2+dl(2)^2+dl(3)^2); % find error
error=(dl(1)^2+dl(2)^2+dl(3)^2); % find error
for j=1:nsat;
rao(j)=((gu(1)-sp(1,j))^2+(gu(2)-sp(2,j))^2+(gu(3)- sp(3,j))^2)^.5; % find new rao
without clock bias
end
end
xuser=gu(1);
yuser=gu(2);
zuser=gu(3);
bias=bu;
output= [xuser yuser zuser bias];

```