A GNSS Signal Simulator and Processor for Evaluating Acquisition and Tracking of GPS-like Signals from Satellites in LEO

by

Sergei (Sam) Joyce Bilardi

B.S., Embry-Riddle Aeronautical University, 2018

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Master of Science

Ann & H.J. Smead Department of Aerospace Engineering Sciences

2021

Committee Members:

Dr. Y. Jade Morton

Dr. Penina Axelrad

Dr. Dennis Akos

Sergei (Sam) Joyce Bilardi, (M.S., Aerospace Engineering Sciences)

A GNSS Signal Simulator and Processor for Evaluating Acquisition and Tracking of GPS-like Signals from Satellites in LEO

Thesis directed by Dr. Y. Jade Morton

Global navigation satellite systems (GNSS) have enabled accurate position, navigation, and timing (PNT) to both civilian and military users for a variety of applications. Since it has become a critical part of today's infrastructure, it is important to investigate ways to enhance or replace this service in challenging environments. These include urban canyons and indoor environments, as well as nefarious attacks, such as jamming and spoofing. Emerging low Earth orbiting (LEO) mega-constellations can be used to augment GNSS. These constellations contain a large number of satellites in order to provide global internet coverage. Although they are not optimized for PNT and use unknown signal structures, several research groups have already demonstrated the ability to track these signals and obtain position and time estimates. This thesis presents LEO signal simulation, acquisition and tracking using an in-house, Julia-language-based package, called *GNSSTools*. Topics included are an overview of GNSS, signal simulation and processing methods, and evaluation. Evaluations demonstrate the self-consistency of *GNSSTools*, and show its ability to process GPS L1 C/A and L5 simulated signals and data samples. Additionally, GPS signals are simulated with Doppler effects similar to signals observed from satellites in LEO to determine their effect on standard acquisition and tracking methods.

Dedication

There's no glory in climbing a mountain if all you want to do is to get to the top. It's experiencing the climb itself – in all its moments of revelation, heartbreak, and fatigue – that has to be the goal.

Karyn Kusama

I would like to dedicate this work to those that supported me through it by providing routine encouragement and advice. To my family and friends.

Lawrence Bilardi, Ed.D

Patricia Joyce Bilardi

Olga Bilardi

Olivia Fowler

Forrest Gasdia, Ph.D

Kari Slotten

Alex Mitchell Tomlinson

Acknowledgements

I would like to acknowledge several individuals, whose help made this thesis possible. I would like to first thank my advisor, Dr. Y. Jade Morton, for her mentorship, guidance, and patience throughout the production of this thesis, Her persistent support and expert opinions were essential to the completion of this thesis. I would also like to thank my thesis committee members, Dr. Penina Axelrad and Dr. Dennis Akos, for their valuable comments, and feedback.

Beyond my advisor and committee, there are additional individuals I would like to acknowledge. I would like to thank Harrison Bourne and Steve Taylor for providing data samples for algorithm testing and for maintaining the IT resources that were used to produce this work. I would also like to thank Dr. Brian Breitsch and Dr. Yang Wang for providing their expert opinions on GNSS signal tracking.

This work was supported by the Science, Mathematics, and Research for Transformation (SMART) scholarship program which is funded by the USD/R&E (The Under Secretary of Defense-Research and Engineering) and National Defense Education Program (NDEP) / BA-1, Basic Research.

Contents

Chapter

Tables

Table

Figures

Figure

Chapter 1

Background and Motivation

Global Navigation Satellite Systems (GNSS) have enhanced our navigational capabilities and are essential to many modern application in research, industry, and civilian and military infrastructure. The need to perform accurate positioning, navigation, and timing (PNT) in challenging environments is a major area of research. While GNSS has enabled global navigation, there are many challenging environments and situations that can impair or disable navigation. These include dense city and indoor environments as well as malicious jamming and spoofing attacks (Z. M. Kassas, Khalife, Abdallah, & Lee, 2020). To help mitigate interruption of PNT services, there is a focus on investigating new GNSS systems, particularly emerging broadband mega-constellations, that can enabled greater precision, robustness, availability, and reliability to both civilian and military users alike. These new constellations, such as OneWeb and Starlink, consist of hundreds to thousands of satellites that operate in low Earth orbit (LEO). Factors such as their sheer number and closer proximity to Earth, compared to current GNSS constellations, makes them ideal candidates for enabling heightened precision that is necessary for many applications such as vehicle automation.

This thesis focuses on presenting several methods that can help aid in evaluating future GNSS constellation designs. These include simulating GNSS signals from a LEO constellation, performing acquisition and tracking of those signals, and evaluation of the acquisition and tracking methods used. These methods have been incorporated into a single programming library named *GNSSTools*, written in the Julia programming language. Julia is a modern, high performance, and open source programming language (Bezanson, Edelman, Karpinski, & Shah, 2017). There are unique differences between signals received from satellites in LEO and medium Earth orbit (MEO). The range of observed Doppler frequency from LEO is much larger than that of MEO satellites. In addition, the Doppler frequency rate becomes significant. These factors alone, can make it difficult for default acquisition and tracking algorithms to succeed. However, standard acquisition and tracking methods can still be used, with some adjustments. For example, a three state phase lock loop (PLL) will greatly improve tracking performance instead of a typically used two state PLL.

This paper is split into five chapters. The first provides a brief overview of current GNSS systems used today and discusses emerging LEO constellations and their usefulness to navigation. The second chapter focuses on simulation of the GNSS signals from LEO satellites. However, it should be noted that *GNSSTools* can also simulate signals from any given constellation and that even arbitrary signal characteristics can be simulated as well. This chapter also discusses various topics such as GPS Gold codes and typical GNSS signal structure, Doppler effect on signals, and receiver noise models. The third chapter explains the methods used to process GNSS signals, which include coarse acquisition and signal tracking. The fourth chapter presents signal acquisition and tracking performance results for various situations as a way to demonstrate *GNSSTools's* capability to simulate and process a wide variety of situations. An emphasis is made towards simulating GNSS signals from LEO satellites and showing performance results from those situations as well. Finally, chapter five discuss the usefulness of this open source package and how it can benefit the GNSS research community as well as possible improvements that can be made to enhance it.

1.1 Modern GNSS Constellations

Most GNSS constellations are situated in MEO. Components of others, and sometimes, whole systems can be stationed in geosynchronous (GSO) and geostationary orbit (GEO). Satellites in MEO have an orbital altitude between 2,000km and 35,780km, while satellites in GSO or GEO have orbital altitudes greater than or equal to 35,780km. Satellites that orbit in LEO have altitudes between 180km and 2,000km (Riebeek & Simmon, 2009). There are multiple GNSS constellations operated by various countries today. Most offer global navigation service, while a handful offer only regional service to their respective countries (Winternitz, 2017). These systems form the backbone of today's society and ensure capabilities such as aerial and ground navigation, environmental monitoring, and commercial operation ("GPS and precision timing applications," 1996). GNSS is even used in the New York Stock Exchange and Nasdaq, which use GNSS along with atomic clocks to perform high rate transactions (Markoff, 2018; Airst, 2010).

1.1.1 Summary of Different GNSS Constellations

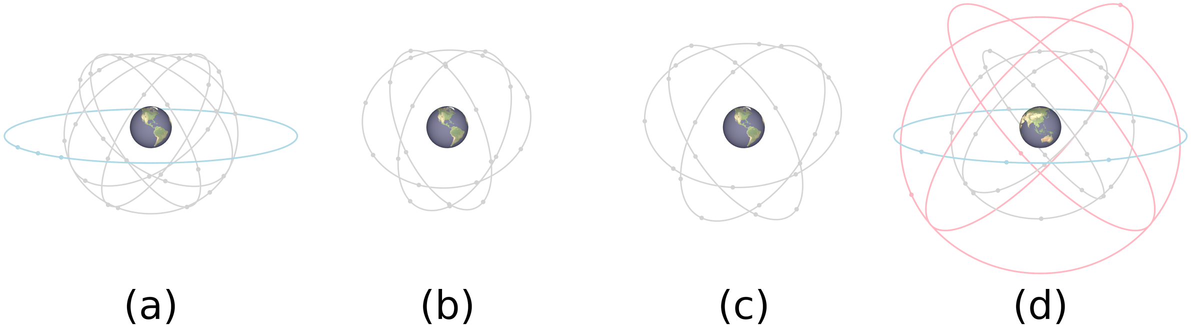


Figure 1.1: Renderings of modern-day GNSS constellations where from left to right are GPS (a), GLONASS (b), Galilieo (c), and Beidou (d). Satellites in MEO, GEO, and IGSO are grey, blue, and red, respectively. These renderings are produced using information from (Y. T. J. Morton et al., 2020a, ch. 1, 3-6, 13).

There are variety of GNSS constellations. Most provide global service, while a couple provide only regional service to their respective countries. The first operational constellations were the United States' Global Navigation Satellite (GPS) system and Russia's GLObal NAvigation Satellite System (GLONASS), which were both declared fully operational in 1995 (Y. T. J. Morton et al., 2020a, ch. 1). Since these two systems were developed, other countries began developing their own systems. The European Union and China built their own constellations, Galileo and BeiDou, respectively. These constellations mostly operate out of MEO. However, BeiDou contains satellites in GEO and inclined geosynchronous (IGSO) orbits (Y. T. J. Morton et al., 2020a, ch. 5, 6). All four of these systems have been designed with the intent to provide global navigational service. There are additional GNSS constellations that are designed only to provide regional service. Those are India's Navigation with Indian Constellation (NAVIC) and Japan's Quasi-Zenith Satellite System (QZSS) systems. NAVIC operates from GEO and GSO, while QZSS operates from IGSO (Y. T. J. Morton et al., 2020a, ch. 7, 8). Renderings and basic constellation information of some of these systems are shown in Figure 1.1 and Table 1.1, respectively.

Constellation	Satellite Num.	Planes	Incl. (deg)	Alt. (km)
GPS	34 total	—	—	—
MEO	Baseline 24, 7 extras	4	55	20,180
GEO	3 SBAS	1	0	35,785
GLONASS	Baseline 24, 2 spares, 1 testing	3	64.8	19,100
Galileo	Baseline 24, 6 spares	3	56	23,229
Beidou	35 total	—	—	—
MEO	24	3	55	21,528
GEO	3	1	0	35,786
IGSO	3	3	55	35,786

Table 1.1: Shows orbital informal information for the GPS, GLONASS, Galileo, and Beidou GNSS constellations. Information from (Y. T. J. Morton et al., 2020a, ch. 1, 3-6, 13).

All of these constellations share similarities in their design and functions. All provide positioning and timing services and consist of user, space, and control segments. They are free-to-use signals, that do not require authentication and or authorization for use, except for signals specifically for the military, such as the military P(Y) codes used in GPS on the L1 channel or 1.57542GHz (Misra & Enge, 2012, ch. 3). All GNSS systems use code division multiple access (CDMA), which allows satellites to transmit unique codes on the same frequency and not interfere with each other. It should be noted that while GLONASS uses CDMA as well, it had previously used frequency division multiple access (FDMA). Instead of each satellite transmitting on the same frequency, they each transmit on their own unique frequency (Heck, 2017; Winternitz, 2017).

In CDMA, each satellite transmits a unique psuedorandom noise (PRN) code on the same frequency. These codes are published to users who can use them to acquire each satellite's PRN code, provided the satellite is in view from their location. By correlating a known code with an signal received on the ground, one can extract the GNSS signal from the noise and perform code and carrier phase tracking to decode the navigation message. They can then use the measured code and carrier phase to estimate their position and time (Misra & Enge, 2012). A position, navigation, and time (PNT) estimate is performed using the idea of trilateration, where the signal source locations are known and the user position is unknown. The satellite atomic clocks are very accurate and correctable using clock bias, drift, and drift rate terms provided in the navigation message. Since the user clock is often not calibrated to the time in the GNSS satellites, a total of four satellites is required to estimate the user's position and time (Winternitz, 2017).

1.1.2 GNSS Accuracy

Navigational accuracy has improved an order of magnitude every thirty years, starting in the late 19th century. Very low frequency (VLF) systems such as long range navigation (LORAN) enabled hundred meter accuracy in the mid 20th century. However, it wasn't until Transit became operational a couple decades later that decameter accuracy was possible. Transit was the predecessor to GPS, which went fully into service in the 1990s and continues to offer meter level accuracy to most users in the world (T. G. R. Reid et al., 2020). GPS is guaranteed to have up to 8m and 13m of error in the horizontal and vertical directions, respectively, 95% of the time during normal conditions ("GPS SPS performance standard," 2020). However, there are ways to improve PNT estimates. For users in the United States, combining information and measurements from the Wide Area Augmentation System (WAAS) with those from GPS satellites provides horizontal and vertical positioning errors of 1.70m and 3.96m, respectively, 95% of the time during normal conditions (Y. T. J. Morton et al., 2020a, ch. 1). Higher quality receivers can provide better accuracy and achieve decimeter and millimeter accuracy when combined with survey grade antennas and more advanced processing methods (WAAS T&E Team, 2017).

1.1.3 Sources of Position and Timing Errors

There are various error sources in GNSS systems. Many of these are not from the actual system itself, but rather are caused by the environment, such as ionospheric and tropospheric delays and multipath effects. Other sources originate from the system such as clock error, which cause the largest position errors if not corrected. Satellites clocks are very accurate and are correctable using correction terms provide in the navigation message. However, the user clock is often not synced with the satellite time and is much more unstable compared to it. While user clock biases are determined simultaneously with position, they contain higher order effects that require additional processing to correct (Betz, 2015). These are not necessary for standalone PNT solutions, but may be necessary for more sensitive remote sensing applications, such as radio occultation (RO), which provides profiles of the atmosphere and ionosphere that can be used to enhance weather models and improve our understanding of atmospheric and space physics (Wang, Yang, & Morton, 2020).

The ionosphere and troposphere both play a role in PNT degradation. The ionosphere bends the GNSS signal as it propagates through it. This results in a code delay and carrier phase advance. In severe cases, it can cause errors up to 100m (Skournetou & Lohan, 2011). Single frequency measurements often require models such as the Klobuchar model to correct part of this error, but not all of it (Klobuchar, 1987). Another method used to correct this is to track signals on two separate frequencies. Since the ionospheric effect varies with frequency, its effect can be determined and removed using dual frequency measurements (Y. T. J. Morton et al., 2020a, ch. 31).

These effects, while harmful to PNT, also provide an opportunity to collect information about the ionosphere. RO is one way to remotely sense the ionosphere, but so is estimating the total electron content (TEC). Methods have been developed using stationary receivers to provide TEC gradient maps of a region (Bourne, 2016). Using a dense network of receivers, one can also estimate higher order maps (Nykiel, Zanimonskiy, Yampolski, & Figurski, 2017). The troposphere can also negatively impact PNT and its effects are usually corrected partially using simplified models (Misra & Enge, 2012).

Additional error sources also exist, such as mutlipath, which can impact PNT. Multipath effects are caused by reflections from nearby surfaces such as buildings. Cell phones are most susceptible to these effects due their antenna design (Humphreys, Murrian, van Diggelen, Podshivalov, & Pesyna, 2016). Using survey grade antennas that mitigate reflections, along with multipath estimation techniques, can help mitigate this effect (Misra & Enge, 2012). The distribution of GNSS satellites in the sky, as observed from an observer and for a given constellation, also effects the position estimate. The geometric dilution of precision (GDOP) is a measure of the position and timing uncertainty caused by the satellite geometry. Greater satellite diversity in the sky allows for lower and better GDOP. Satellites in constellations such as GPS have inclinations of 55°. For a constellation such as this, 55° is the highest latitude for a user to observe a satellite in this constellation pass directly overhead. Therefore, at higher latitudes, vertical accuracy is lower than it is at lower latitudes (T. G. Reid, Neish, Walter, & Enge, 2018).

1.1.4 Satellite-Based Augmentation System

Satellite-Based Augmentation Systems (SBAS) help improve GNSS in safety-of-life applications as well as less critical ones, such as recreational biking, hiking, and running. These systems improve PNT by providing additional information that helps aid integrity and improve PNT accuracy. One of these systems is the Wide Area Augmentation System (WAAS) which augments GPS in the North American region. This system was initially designed for applications that require greater accuracy such as instrument landing systems (ILS) (Betz, 2015, ch. 8). WASS can improve accuracy for airplanes by up to an order of magnitude (Jones, 2013). Other countries have developed SBAS to augment GPS in their own regions as well. Russia has the System for Differential Correction and Monitoring (SDCM), Europe has the European Geostationary Navigation Overlay Service (EGNOS) and Japan has the Multifunctional Transport Satellite SBAS (MSAS). In fact, Japan's QZSS is specifically designed to augment GPS over the Asia-Pacific region. It does not provide standalone service (Pullen, 2007; Y. T. J. Morton et al., 2020a, ch. 7, 8, 13).

1.2 Emerging LEO Mega-Constellations

There is always an interest in improving the reliability, availability, and accuracy of GNSS. Many of these reasons are civil safety-of-life applications such as precision navigation for aerial and ground applications. Other reasons are to support the continuation of military capability in GNSS denied environments or to protect military assets from other hostile actions such as jamming and spoofing (T. G. Reid et al., 2018). GNSS augmentations systems, both space and ground based, help improve the above performance categories. Emerging megaconstellations, by companies such as SpaceX and OnewWeb, and several others, can also offer the ability to improve GNSS in many areas. Countries such as the UK are taking interest in using OneWeb's planned LEO constellation for PNT (Inside GNSS, 2020). It is possible that these can also be useful in automated vehicles, where it is recommended that they posses better than 1.5 meters of accuracy for safe navigation (T. G. R. Reid et al., 2019). Below is a discussion on these new constellations and the benefit and challenges to using them for navigation.

1.2.1 Current and Future LEO Constellations

LEO constellations offer the opportunity of enchanced navigation. Some constellations have satellites that are around thirty-six times closer to the Earth when compared to GPS satellites. This means less free space loss. Also, the large quantity of satellites in these new constellations could enable greater satellite diversity in the sky, resulting in lower GDOP and therefore, greater accuracy (Y. T. J. Morton et al., 2020b, ch. 43).

Constellation	Satellite Num.	Planes	Incl. (deg)	Alt. (km)
Iridium	66	6	86.4	780
Starlink	1,584	24	53	550
OneWeb	648	18	87.9	1,200

Table 1.2: Shows orbital informal information for the Iridium, Starlink, and OneWeb constellations. Information from (**Iridium NEXT**, 2021; **Starlink**, 2021; **OneWeb**, 2021). A table of additional LEO constellations can be found in (Y. T. J. Morton et al., 2020b, ch. 43).

Iridium is a LEO constellation that provides true global communication services. The company guarantees that at least one satellite will be in view of a user anywhere on the Earth, barring obstructions from terrain and buildings (T. G. Reid et al., 2018). They do this by placing their satellites in six polar orbital planes consisting of eleven satellites per plane (**Iridium NEXT**, 2021). Iridium provides a service called Iridium-based Satellite Time and Location (STL), which is used to enhance navigation indoors. These signals are 30dB greater than that of GNSS signals (T. G. R. Reid et al., 2020). There are already cases where signals from non-GNSS satellites have been used for navigating GNSS denied areas, such as temporarily using OrbCom signals of opportunity (SOPs) in a UAV for navigation. It should be noted that in this case, the UAV already had a PNT solution from a GNSS constellation prior to entering a simulated GNSS denied area (Z. Z. M. Kassas, Khalife, Neinavaie, & Mortlock, 2020).

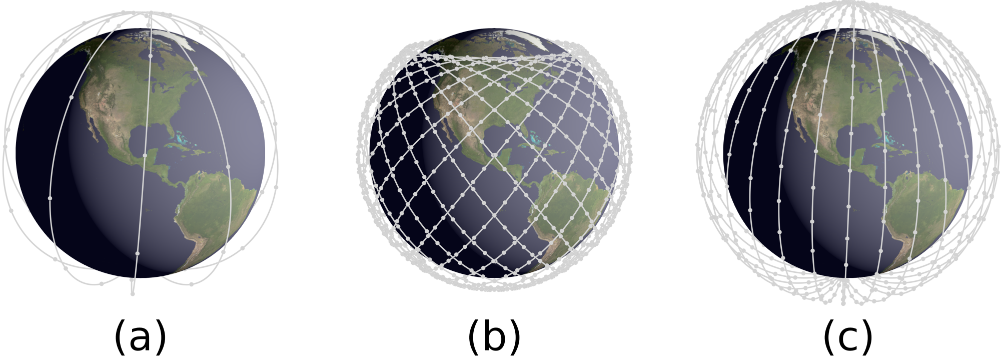


Figure 1.2: Shows renderings of the Iridium (a), Starlink (b). and OneWeb (c) constellations. Renderings are generated using information from (**Iridium NEXT**, 2021; **Starlink**, 2021; **OneWeb**, 2021).

There are additional LEO constellations that are being established. For years, SpaceX and OneWeb have been launching satellites for their planned constellations, which will have over 1500 and 600 satellites, respectively (**Starlink**, 2021; **OneWeb**, 2021). These constellations will provide global broadband internet service and are considered candidates for future PNT usage and assessing LEO constellations for navigation (T. G. Reid et al., 2018; Inside GNSS, 2020). Renderings and information on the constellations discussed so far are shown in Figure 1.2 and Table 1.2, respectively.

1.2.2 Challenges and Limitations to Using LEO Constellations for PNT

One main limitation to using the aforementioned constellations is their dedication to communications and/or broadband internet services. These constellations are not designed to perform PNT independently, since it would take considerable effort and expenses to adapt them for standalone navigation (Faragher & Ziebart, 2020). Further, even if a company produced a constellation with a dedicated PNT capability, there is no reason why they would not charge for the service. Because of this, it is often more advantageous to focus on using these LEO signals as a way to aid GNSS in situations where it is compromised or where greater accuracy is desired (T. G. R. Reid et al., 2020).

Another major challenge are the clocks on-board most satellites are inadequate for PNT. Although chip-scale atomic clocks (CSACS) are an option, they do not have the same precision as the clocks onboard GNSS satellites. In an article by Reid et al, they indicate that CSACS have an Allen Deviation of several orders of magnitude worse than that of atomic clocks on GPS satellites if they are updated every twenty-four hours. However, updating the CSACS more frequently, about once per orbit, makes the CSAC performance only one to two orders of magnitude worse than that of GPS clocks. The improved satellite geometry from the LEO constellation may also help to counter this worse performance (T. G. Reid et al., 2018). However, there is no precise orbit determination (POD) performed for LEO satellites. POD accounts for many perturbation sources such as nonuniform gravitation forces, atmospheric drag, and solar radiation pressure. Without considering these forces, propagating two-line elements (TLEs) can contain large errors, such as 3km to 10km 24 hours after TLE generation (Y. T. J. Morton et al., 2020b, ch. 43).

Signals from satellites in LEO have considerably larger Doppler frequency ranges when compared to current GNSS satellites. This is shown in Figure 1.2. The Doppler rate also becomes significant. The larger Doppler range requires a larger Doppler search area in the initial acquisition phase, while the significant Doppler rate may require a three-state PLL rather than a two-state PLL in order for the signal to be tracked successfully. Luckily, interest lies in developing robust tracking loops to counter user dynamics such as vibrations from airplanes during RO observations (Wang

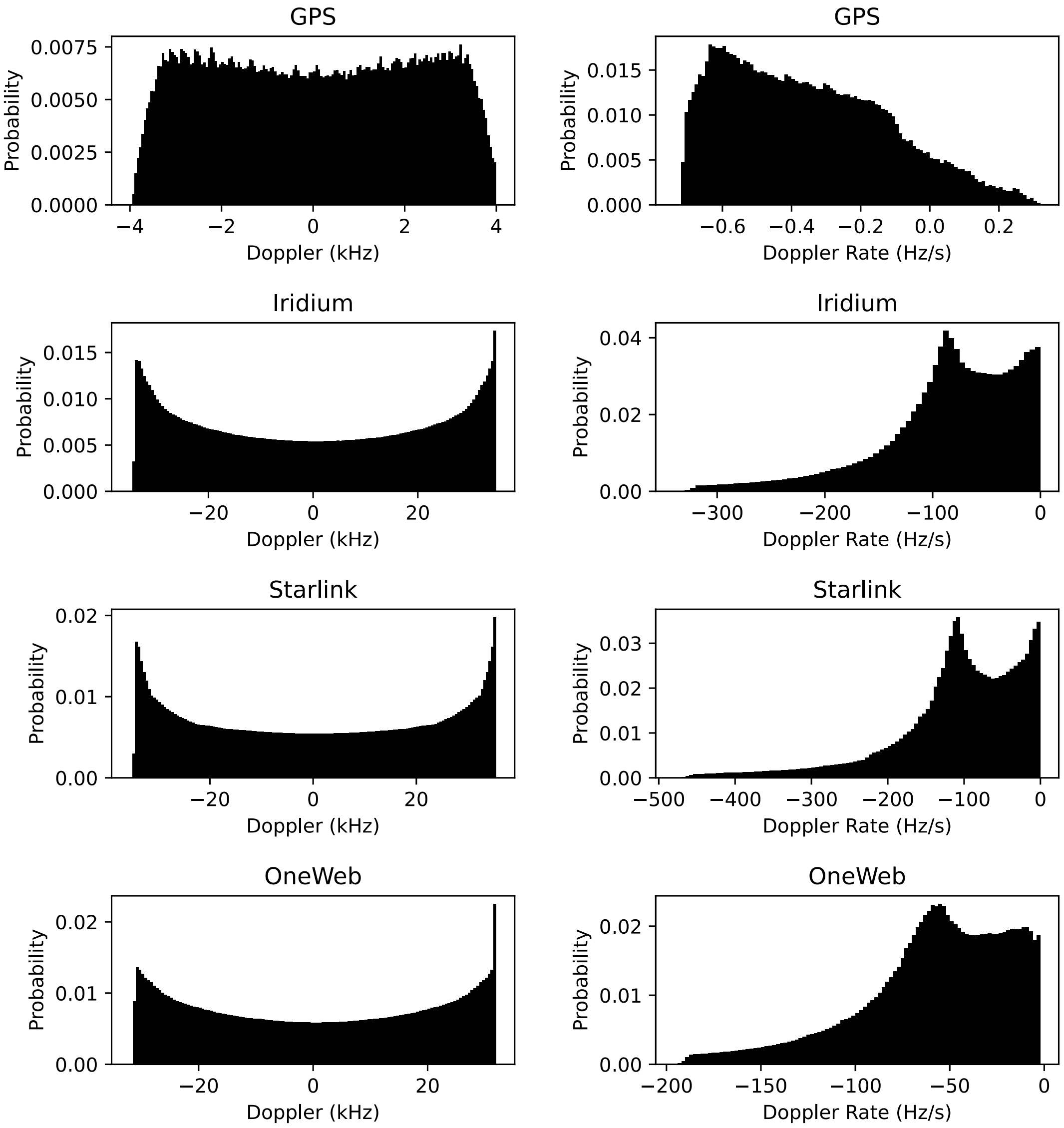


Figure 1.3: Shows the Doppler and Doppler rate probability density functions (PDFs) for the GPS, Iridium, Starlink, and OneWeb constellations at the GPS L1 frequency from 25 randomly chosen observation locations on the Earth. Figure produced using *GNSSTools*. Information for these constellations is from (European Space Agency, 2011; **Iridium NEXT**, 2021; **Starlink**, 2021; **OneWeb**, 2021).

et al., 2020). A thorough analysis of different tracking loops and their performances has also been performed (Yang, Ling, Poh, & Morton, 2017; Yang, Morton, Ling, & Poh, 2017).

1.3 Motivation for Work

Modern GNSS systems have enhanced civilian and military capabilities. They provide accurate PNT service for many applications. Although, some applications required greater precision, creating the need for augmentation systems. There are several different augmentation systems throughout the world, both ground and space based. These improve the reliability, availability, and accuracy of GPS in those areas.

Emerging LEO constellations may be able to augment modern GNSS to provide PNT in GNSS denied areas and indoor environments. Others have already used LEO signals for navigating challenging environments. While current and new LEO constellations are not dedicated for PNT, they are still an interest to various groups and there is active research focused on leveraging them for navigation. Examples of recent development towards LEO PNT are navigating using carrier Doppler shift (Psiaki, 2021) and carrier tracking and positioning with Starlink satellite signals (Z. Z. Kassas, Khalife, & Neinavaie, 2021).

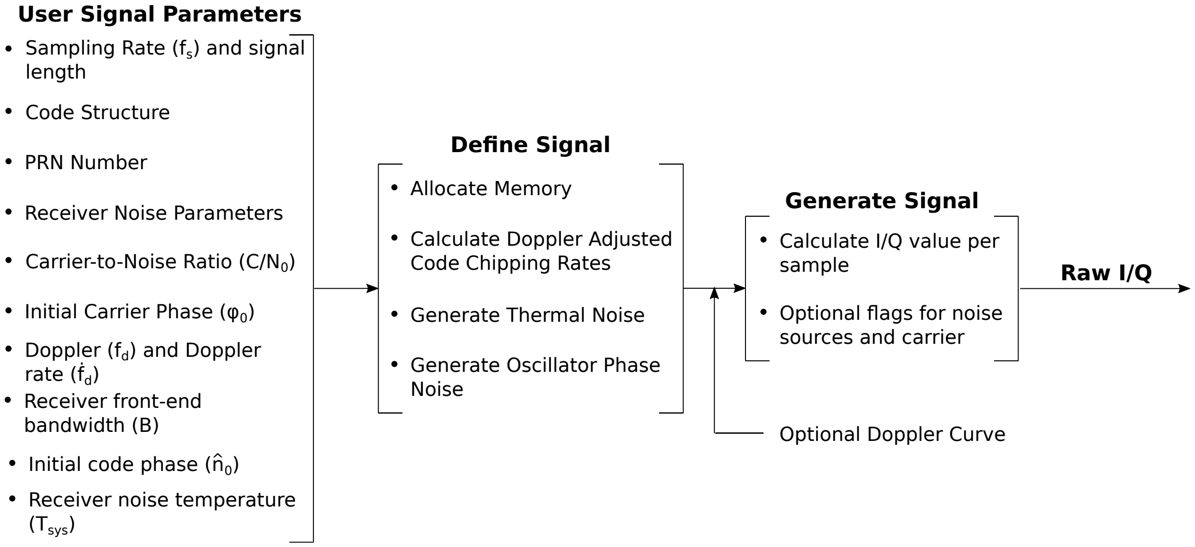
This work aims to provide additional insight into using LEO signals for PNT by presenting a Julia language module, *GNSSTools*, that simulates and processes GNSS signals. Processing includes acquisition and code and carrier phase tracking of these signals. The simulation segment produces signals that in addition to the GNSS codes, contain thermal and oscillator phase noise and appropriate Doppler effects for a given satellite constellation. *GNSSTools* currently simulates GPS L1 C/A and L5 signal codes, but is designed to enable definition of custom codes, making it easier to add new GNSS signals later. This work will discuss the steps of simulating these signals in Chapter 2 and acquiring and tracking them in Chapter 3. Various simulated scenarios regarding GPS and LEO constellations and the results of acquiring and tracking the resulting signals will be discussed in Chapter 4.

It should be noted that there are limitations to *GNSSTools* and assumptions made throughout this work. First, ionospheric, tropospheric, and multipath effects are not simulated and accounted for, since it was considered out of scope for this thesis. Also, it is assumed that the LEO signals are GNSS like, where a known ranging code is used to extract the appropriate GNSS code from the signal. Finally, navigation data bits simulated in later chapters are randomly generated and do not contain accurate navigation information.

Chapter 2

GNSS Signal Simulation

This chapter discusses the methods used to simulate GNSS signals. Topics, such as generating GNSS codes, Doppler effects, constellation, and receiver thermal and oscillator phase noise are discussed. These methods are used to produce GNSS signals that are used in subsequent chapters for processing and evaluation of a LEO PNT system.



User Signal Parameters
• Sampling Rate (f_s) and signal length
• Code Structure
• PRN Number
• Receiver Noise Parameters
• Carrier-to-Noise Ratio (C/N_0)
• Initial Carrier Phase (φ_0)
• Doppler (f_d) and Doppler rate ($\dot{f}_d$)
• Receiver front-end bandwidth (B)
• Initial code phase ($\hat{n}_0$)
• Receiver noise temperature (T_{sys})
Define Signal
• Allocate Memory
• Calculate Doppler Adjusted Code Chipping Rates
• Generate Thermal Noise
• Generate Oscillator Phase Noise
Generate Signal
• Calculate I/Q value per sample
• Optional flags for noise sources and carrier
Raw I/Q
Optional Doppler Curve

Figure 2.1: Diagram showing the process of simulating GNSS signals in *GNSSTools*.

Figure 2.6 shows the high level process to simulating a GNSS signal. Information about the signal code structure is required along with additional parameters such as the receiver thermal and oscillator noise parameters, desired carrier-to-noise ratio (C/N_0), initial code and carrier phase, and Doppler and Doppler rates. Note that an optional Doppler curve can be provided as well.

2.1 GNSS Codes

In all GNSS constellations, each satellite transmits its own ranging code. Ranging codes are used to produce an estimate of the user to satellite range. All codes, except for those that are reserved for government and military use, are published and can be generated locally. Correlating these local replicas with a GNSS signal provides the user with an estimate of the code phase. Decoding the navigation data, that is overlayed on top of the ranging code, also provides the user with an absolute estimate of the range between the user and the satellite that the signal originates from in addition to satellite orbit and time information needed to estimate a user's position and time.

	L1 C/A	L5
Center Frequency (MHz)	1575.42	1176.45
Code Length	1023	10230
Chipping Rate (MHz)	1.023	10.23
Nav. Message Data Rate (bps)	50	50 (100 sps)
Channels	Data Only	Data and Pilot
Minimum Signal Power (dBW)	-158.5	-157

Table 2.1: Shows information about the GPS L1 C/A and L5 codes that can be generated in *GNSSTools*. Information from (Flores, 2020a, 2020b).

The codes discussed in this section will be the legacy GPS L1 C/A and modern L5 codes, since they are implemented in *GNSSTools*. Table 2.1 shows information about these two codes. L5 contains two channels, pilot and data, while L1 C/A only contains a data channel. Pilot channels are signals with no navigation data bits and are useful for situations that require long integration times, that would otherwise be impossible without additional post processing methods. The following will detail the generation of these two codes types. A discussion of general signal generation using arbitrary codes will follow.

2.1.1 Generating L1 C/A Codes

The L1 C/A codes are length 1023 Gold codes that are designed to have auto- and cross-correlation properties that are appropriate for GNSS. A unique set of these codes are used for ranging and enable signals from each satellite to be differentiated from each other (Misra & Enge, 2012). Generating these codes starts with generating a sequence of bits using two 10 bit linear-feedback shift registers, called G_1 and G_2. The output of G_1 is the same regardless of the satellite and it is added to the output, G_{2i}, which is the output from the phase selector. The phase selector uses two taps, S_1 and S_2, at locations unique to each satellite to generate this code. The sum of the G_1 and G_{2i} sequences is the L1 C/A code for a specific satellite (Flores, 2020a). Figure 2.2 summarizes this process. The symbol $\oplus$ means modulo 2 addition, or exclusive or (XOR) operation between two numbers that are either 1 or 0.

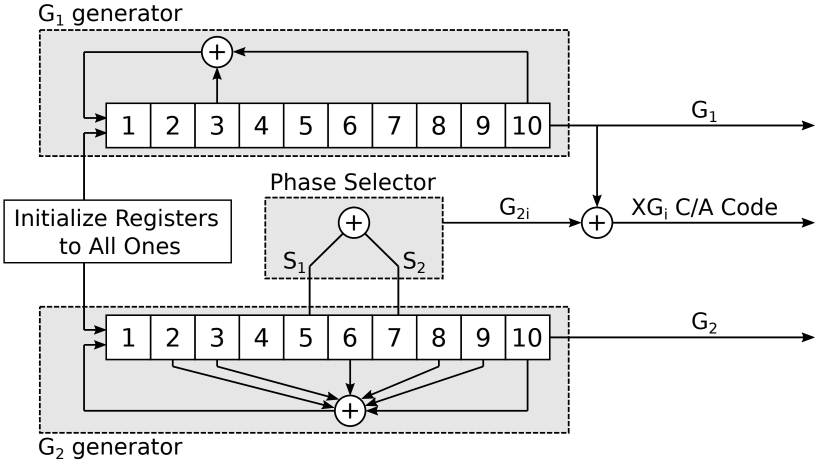


Figure 2.2: Summarizes the process of generating L1 C/A codes for a given satellite. The phase selector takes two values from S_1 and S_2 locations in the G_2 register, to generate the G_{2i} code. S_1 and S_2 are unique to each satellite. Figure recreated based off the figure on page 63 in (Misra & Enge, 2012).

The chipping rate of the L1 C/A codes is 1.023MHz and they repeat every 1ms (Misra & Enge, 2012). The precision that can be achieved is proportional to the wavelength of each chip in the code.

$$\lambda_{code} = \frac{c}{f_{code}} \tag{2.1}$$

where c is the speed of light, approximately 3×10^8m/s, and f_{code} is the chipping rate of the code sequence. The wavelength of each chip in the L1 C/A code is about 300m, which results in an expected range precision of 0.3m, assuming that only thermal noise is present (Misra & Enge, 2012).

PRN numbers are used to identify a satellite and its corresponding ranging code. The instructions for generating these codes can be found in the respective interface specification document (ICD) of that constellation and set of ranging codes. Refer to the ICD for the L1 C/A codes for a list of the phase selector taps and their corresponding PRN numbers (Flores, 2020a).

2.1.2 Generating L5 Codes

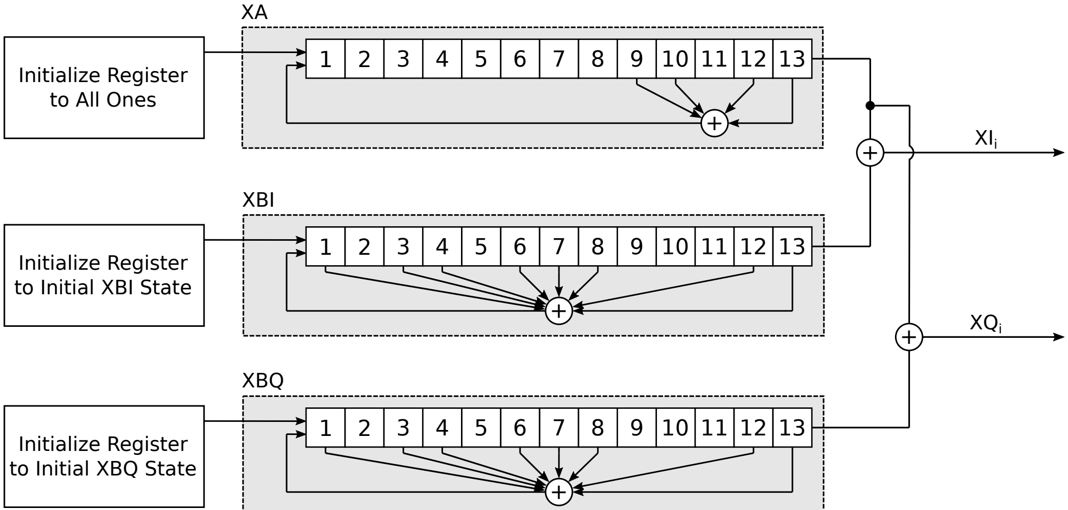

XA
Initialize Register to All Ones
1 2 3 4 5 6 7 8 9 10 11 12 13
XBI
Initialize Register to Initial XBI State
1 2 3 4 5 6 7 8 9 10 11 12 13
XBQ
Initialize Register to Initial XBQ State
1 2 3 4 5 6 7 8 9 10 11 12 13
XI_i
XQ_i

Figure 2.3: Summarizes the process of generating L5I and L5Q ranging codes for a given satellite. Figure produced using information from (Flores, 2020b).

The L5 codes are in phase quadrature. On the I and Q channels are the data and pilot codes, respectively. The ranging codes are ten times the chipping rate of the L1 C/A codes. Since they too repeat every 1ms, the codes have a length of 10,230 (Flores, 2020b). Unlike L1 C/A, these codes are not part of the Gold code family, because their length is not equal to $2^n - 1$, where n is the number of bits inside the shift registers (Gold, 1967). In most text, including the L5 code ICD, the codes on the I and Q channels are called, L5I and L5Q, respectively (Flores, 2020b).

Figure 2.3 summarizes the process to generating L5 codes for a specific PRN number. The process is similar to that of generating L1 C/A codes. There are three 13 bit shift registers, XA, XBI, and XBQ. XA is initialized to all ones and its output is added to the XBI and XBQ register outputs to produce the values of the L5I and L5Q chips, respectively. All three registers produce length 10,230 codes that repeat. Although, the XA register does not run to completion. Instead, it produces the first 8,090 bits, initializes back to all ones, produces the first 2,040 bits, and then resets again.

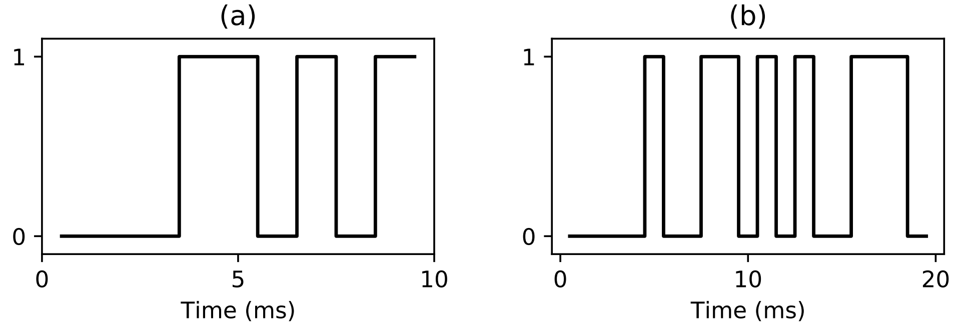


Figure 2.4: Shows the 10ms (a) and 20ms (b) Neuman-Hofman sequences that are overlayed onto the L5I and L5Q ranging codes, respectively.

The XBI and XBQ registers are initialized with a 13 bit long sequence that is unique to each PRN number. The initial XBI and XBQ register states are also not the same for a given PRN number. See the ICD for the L5 codes for a list of these initial states (Flores, 2020b). Finally, an overlay code is mixed with both channel codes. A 10ms and a 20ms Neuman-Hofman sequence is mixed onto the L5I and L5Q codes, respectively. The chipping rate of this code is 1kHz. Figure 2.4 shows the two Neuman-Hofman sequences (Misra & Enge, 2012).

2.1.3 Defining Signals using Custom Codes

GNSSTools takes a generalized approach to generating arbitrarily defined signals. In *GNSSTools* a GNSS signal is described primarily by its center frequency, and set of codes on both the I and Q channels. As shown in Figure 2.5, a set of N and M codes on the I and Q channels, respectively, can be mixed together and modulated onto a carrier to produce an output signal. A received signal in *GNSSTools* is described as the following.

$$s(t) = Ae^{j(2\pi\{f_{IF}+f_d(t)\}t+\phi_o+\phi_{code}(t)+\nu(t))} + w(t) \tag{2.2}$$

where A is the carrier amplitude, f_{IF} is the intermediate frequency, $f_d(t)$, is the Doppler frequency at time t, ϕ_o is the initial carrier phase, and $\phi_{code}(t)$ is the phase between the I and Q channel code values, shown below.

$$\phi_{code}(t) = tan^{-1}\left(\frac{C_Q(t)}{C_I(t)}\right) \tag{2.3}$$

C_I and C_Q are the mixed values of all code layers for the I and Q channels respectively at time t. $\nu(t)$ and $w(t)$ are the oscillator phase and thermal noise introduced into the receiver, which will be discussed in later sections. It is assumed that the clock on-board the satellite contains no instability.

	C_I	C_Q	ϕ_{code} (°)
BPSK(C_I)	1	—	0
	-1	—	180
BPSK(C_Q)	—	1	90
	—	-1	270
QPSK(C_I, C_Q)	1	1	45
	-1	1	135
	-1	-1	225
	1	-1	315

Table 2.2: Shows BPSK of I or Q only channel codes and QPSK of codes on both the I and Q channels.

The product of codes is determined separately for each channel and is modulated onto the

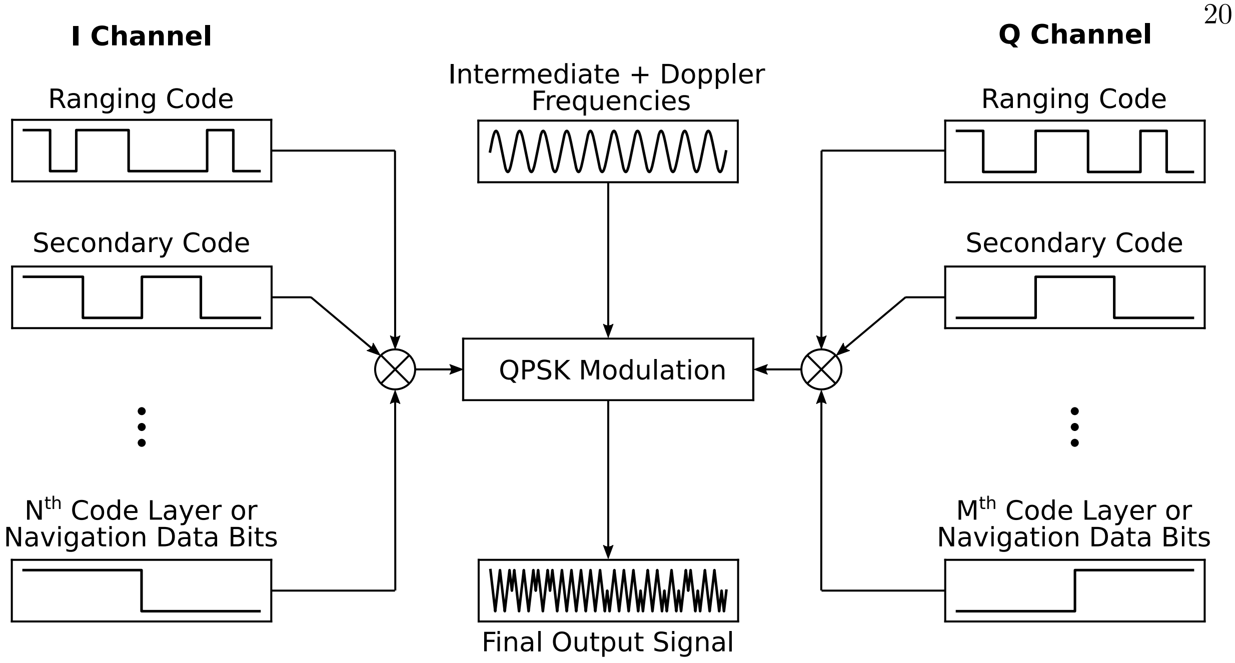

I Channel
Q Channel
Intermediate + Doppler Frequencies
Ranging Code
Secondary Code
QPSK Modulation
Ranging Code
Secondary Code
N th Code Layer or Navigation Data Bits
M th Code Layer or Navigation Data Bits
Final Output Signal

Figure 2.5: Summarizes the process to generating an arbitrarily defined signal with N codes on the I channel and M codes on the Q channel. The user specifies the signal center frequency and the intermediate frequency after down conversion in the receiver, along with each code layer on both channels. Not shown are the addition of thermal and phase noise and ADC quantization steps.

signal either through binary phase shift keying (BPSK) or quadrature phase shift keying (QPSK) depending on the channels present. Table 2.2 shows the process for calculating ϕ_{code} based off whether I and/or Q channel codes are being simulated. The signal, $s(t)$, is also sampled and quantized, resulting in a discrete time signal that is sampled at the sampling frequency, f_s, and with integer amplitude steps between $\pm 2^{n-1}$, where n is the simulated bit depth of the signal.

2.2 Satellite Constellations

This section discusses defining satellite constellations and calculating the expected Doppler and Doppler rate that would be observed on Earth. This helps provide insight into one of the primary differences between signals transmitted from satellites in MEO and LEO orbits. A summary of simulating Doppler effects in GNSS signals follows.

2.2.1 Satellite Orbit Overview & Defining Constellations

Satellite orbits are described by six orbital parameters, shown in Figure 2.6. The satellite orbit is propagated forward in time using these orbital parameters. Each satellite constellation is defined as either a Walker Delta or Walker Star constellation. Each constellation consists of t satellites which are distributed into p planes. Walker Delta and Walker Star constellations have orbital planes with longitude of ascending node spacing $\Delta\Omega$ equal to the following (T. G. Reid et al., 2018).

$$\Delta\Omega_{\Delta} = \frac{2\pi}{p} \tag{2.4}$$

$$\Delta\Omega_{\star} = \frac{\pi}{p} \tag{2.5}$$

where $\Delta\Omega_{\Delta}$ and $\Delta\Omega_{\star}$ refer to $\Delta\Omega$ for a Walker Delta and Walker Star constellation, respectively. The number of satellites in a given plane are equally distributed in a plane where the angular spacing (true anomaly difference) between each satellite is $2\pi/s$. s is the number of satellites in a given plane, which is given by t/p. There is also a true anomaly offset in adjacent planes, which is $\Delta f = \pi/s$ (T. G. Reid et al., 2018).

The GPS space segment is a Walker Delta constellation. Of the megaconstellations discussed in Chapter 1, SpaceX's Starlink constellation is also a Walker Delta, but Iridium's and OneWeb's constellations are Walker Star constellations. Figures 1.1 and 1.2 show modern GNSS and LEO constellations. The reader can refer to (Misra & Enge, 2012) for an in depth discussion on propagating satellite orbits.

2.2.2 Obtaining the Expected Doppler and Doppler Rate

Orbit propagation and coordinate transformations are done using a Julia-language-based package called *SatelliteToolbox* (Chagas, de Sousa, Louro, & dos Santos, 2018). The number of satellites and number of planes and their inclinations are used along with the equations discussed

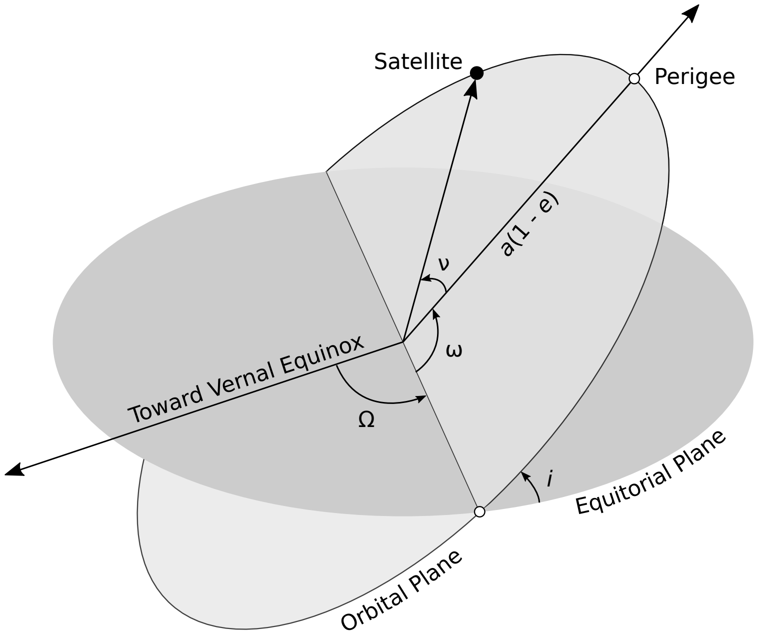


Figure 2.6: Shows the six Keplerian elements that describe a satellite's orbit. These include the semi-major axis (a), eccentricity (e), inclination (i), longitude of ascending node (Ω), argument of periapsis (ω), true anomaly (ν). Figure created based off inspiration from Figure 4.11 in (Misra & Enge, 2012) and (Snyder, 2007).

above to determine the argument of longitude spacing between planes, spacing between satellites in each plane, and the true anomaly phase difference between adjacent planes. The state vector of each satellite in the Earth centered Earth fixed (ECEF) frame is its position, velocity, and acceleration at time t. The satellite range is the magnitude of the difference between the satellite and observer positions.

$$r_r = |\vec{r}_s - \vec{r}_0| \tag{2.6}$$

The line of site velocity (also known as radial velocity and range rate), denoted $\dot{r}_r$, is the dot product between the satellite velocity vector and a vector pointing from the observer to the satellite. It is also the time derivative of r_r.

$$\dot{r}_r = \frac{\dot{\vec{r}}_s \cdot (\vec{r}_s - \vec{r}_o)}{|\vec{r}_s - \vec{r}_o|} \tag{2.7}$$

where $\vec{r}_s$ and $\dot{\vec{r}}_s$ are the satellite position and velocity vectors. $\vec{r}_o$ is the observer position. The range acceleration is the derivative of $\dot{r}_r$ (Ellis, Rheeden, & Dowla, 2020).

$$\ddot{r}_r = \frac{\ddot{\vec{r}}_s \cdot (\vec{r}_s - \vec{r}_o)}{|\vec{r}_s - \vec{r}_o|} + \frac{\dot{\vec{r}}_s \cdot \dot{\vec{r}}_s}{|\vec{r}_s - \vec{r}_o|} + \left(\frac{\dot{\vec{r}}_s \cdot (\vec{r}_s - \vec{r}_o)}{|\vec{r}_s - \vec{r}_o|}\right)^2 \frac{1}{|\vec{r}_s - \vec{r}_o|} \tag{2.8}$$

Finally, the Doppler and Doppler rate are defined as the following and scale with the transmitted carrier frequency, f_c.

$$f_d = -\frac{f_c \dot{r}_r}{c} \tag{2.9}$$

$$\dot{f}_d = -\frac{f_c \ddot{r}_r}{c} \tag{2.10}$$

where f_d and $\dot{f}_d$ are the Doppler frequency and Doppler frequency rate, respectively. c is the speed of light.

2.2.3 Simulating Doppler Effects in GNSS Signals

As shown in Figure 1.3, the expected Doppler frequency observed by a stationary user for a GPS constellation is between ±4kHz. The Doppler rate is also -0.73Hz/s, at the worst. Of course, user motion can make this higher. For a LEO constellation though, a stationary observer would observe a Doppler frequency between $\pm 35kHz$. The Doppler rate is also -475Hz/s at worst. Large Doppler ranges can result in longer acquisition times and significant Doppler rates may require the use of third order signal tracking loops. Methods for including these effects are discussed below.

The Doppler effect on GNSS signals is the ratio between the Doppler and carrier frequencies.

$$f_{d_{code}} = \frac{f_{code} f_d}{f_c} \tag{2.11}$$

The code Doppler rate is the time derivative of $f_{d_{code}}$.

$$\dot{f}_{d_{code}} = \frac{f_{code} \dot{f}_d}{f_c} \tag{2.12}$$

The adjusted chipping rate, $f_{code_{adj}}(t)$, is a function of the Doppler, Doppler rate, and t. Integrating Equation 2.13 results in Equation 2.14, the code phase at time t.

$$f_{code_{adj}}(t) = f_{code} + f_{d_{code}} + \dot{f}_{d_{code}} t \tag{2.13}$$

$$n(t) = n_0 + \left(f_{code} + f_{d_{code}}\right) t + \frac{1}{2}\dot{f}_{d_{code}} t^2 \tag{2.14}$$

The carrier phase is also shown in Equation 2.15. By default, $GNSSTools$ uses a user specified Doppler and Doppler rate to determine the current code and carrier phase as a function of time. For short periods of time, this is sufficient. However for longer simulation runs, $f_d(t)$ is nonlinear and a slightly different approach is used to determined the code and carrier phase at time, t.

$$\phi(t) = \phi_0 + 2\pi\left(f_{IF} + f_d + \frac{1}{2}\dot{f}_d t\right) t \tag{2.15}$$

Equations 2.14 and 2.15 are modified to become Equations 2.16 and 2.17, respectively. Here, the code and carrier phase are now functions of the integrated Doppler function. As shown in Figure 2.1, an optional Doppler curve can be given to simulated nonlinear Doppler effects in the simulated signal. The Doppler function is numerically integrated over each sample period to produce the code and carrier phase at each sample.

$$n(t) = n_0 + f_{code} t + \frac{f_{code}}{f_c}\int_0^t f_d(t')dt' \tag{2.16}$$

$$\phi(t) = \phi_0 + 2\pi\left(f_{IF} t + \int_0^t f_d(t')dt'\right) \tag{2.17}$$

2.3 Receiver Noise Models

This section will provide an overview of the major noise sources in a GNSS receiver system, which include thermal and oscillator phase noise. Sampling and quantization noise also occur from an ADC, where 1-bit and 4-bit ADCs can contribute to approximately 2dB and 0.17dB, respectively, of quantization noise into a signal (Y. T. J. Morton et al., 2020a, ch. 18). Thermal noise in a receiver system is a function of the receiver's equivalent noise temperature and bandwidth, while oscillator phase noise is a result of clock instabilities inside a receiver (Misra & Enge, 2012).

2.3.1 Thermal Noise

The carrier-to-noise density ratio (C/N_0) is a normalized measure of the signal power against the noise power (Y. T. J. Morton et al., 2020a, ch. 18).

$$\frac{C}{N_0} = \frac{P_S}{kT_{sys}} \tag{2.18}$$

where P_S is the signal power and is related to the signal amplitude ($P_S = A^2/2$ for a sinusoid and $P_S = A^2$ for a complex exponential) (Misra & Enge, 2012). k is the Boltzmann constant (1.38×10^{-23}J/K), and T_{sys} is the receiver equivalent noise temperature, which is the sum of the thermal noise from the receiver itself and the antenna.

$$T_{sys} = T_A + T_R \tag{2.19}$$

where T_A is the antenna noise temperature, typically approximately 100K (Misra & Enge, 2012). T_R is the receiver noise temperature and is a function of the noise figure (NF) of the receiver, or $10log_{10}(F)$, where F is the noise factor. NF is typically around 4dB, making $T_{sys} =$ 535K (J. Y. Morton, 2019). T_0 is 290K.

$$T_R = T_o(F - 1) \tag{2.20}$$

Equation 2.21 is the receiver noise power. It is a function of the receiver bandwidth, B. For example, for signals L1 C/A and L5, the bandwidth would be at least twice their chipping rates, 2.046MHz and 20.46MHz, respectively. The thermal noise in the signal, denoted $w(t)$ in Equation 2.2, can be modelled as a zero mean Gaussian, $w \sim \mathcal{N}(0, P_N)$.

$$P_N = kT_{sys}B \tag{2.21}$$

2.3.2 Oscillator Phase Noise

The oscillator phase noise, denoted $\nu(t)$ in Equation 2.2, is due to clock instability. It can be usually formulated as the fractional frequency of an oscillator, relative to its nominal frequency, f_0 (Riley & Howe, 2000).

$$y(t) = \frac{\Delta f}{f_0} \tag{2.22}$$

The phase noise is best described by it's one sided power spectral density (PSD) function, denoted $S_y(f)$, and is often modeled as the weighted sum of f^α noise, as shown in Equation 2.23 (Rebeyrol et al., 2006). These include the following coefficients used to model phase noise.

$$S_y(f) = \sum_{\alpha=-2}^{2} h_\alpha f^\alpha \tag{2.23}$$

- h_{-2}: Random Walk frequency noise
- h_{-1}: Flicker frequency noise
- h_0: White frequency noise
- h_1: Flicker phase noise
- h_2: White phase noise

There is a relationship between the PSD of the fractional frequency to the frequency and phase fluctuations, $S_{\Delta f}(f)$ and $S_\nu(f)$, respectively (Rebeyrol et al., 2006).

$$S_{\Delta f}(f) = f_0^2 S_y(f) \tag{2.24}$$

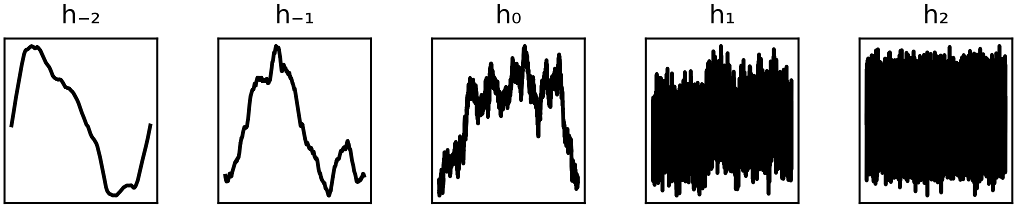


Figure 2.7: Shows the time domain equivalent of each component of $h_\alpha f^{\alpha-2}$ phase noise.

$$S_\nu(f) = \frac{f_0^2 S_y(f)}{f^2} \tag{2.25}$$

Equation 2.23 can be substituted into Equation 2.25 and used to calculate the PSD of phase noise for an oscillator with a given set of h parameters. Figure 2.7 shows the different components of phase noise as the time domain equivalent to Equation 2.26.

$$S_\nu(f) = f_0^2 \sum_{\alpha=-2}^{2} h_\alpha f^{\alpha-2} \tag{2.26}$$

It should be noted that the above is for the single sided PSD and that specification documents present the oscillator phase noise at every frequency decade relative to the carrier frequency in a two sided PSD, $L(\Delta f)$. This differs by approximately a factor of 2 or 3dB when compared to the one sided PSD (Chiou, 2010).

$$L(\Delta f) \approx \frac{S_\nu(f)}{2} \tag{2.27}$$

The oscillator h parameters can be fit to $L(\Delta f)$ and equivalent h parameters for different oscillator $L(\Delta f)$ PSDs are shown in Tables 2.2 and 2.1, respectively in (Chiou, 2010). Figure 2.8 shows the calculated one sided PSDs, $S_\nu(f)$, for each oscillator listed and the values of $2L(\Delta f)$, which the author had taken from the specification documents of each oscillator.

For phase noise simulation, it is only a matter of converting the one sided PSD, $S_\nu(f)$, into the two sided equivalent PSD. Since the PSD only represents the magnitude and does not contain phase information, random phase noise can be multiplied to produce the equivalent Fourier transform of

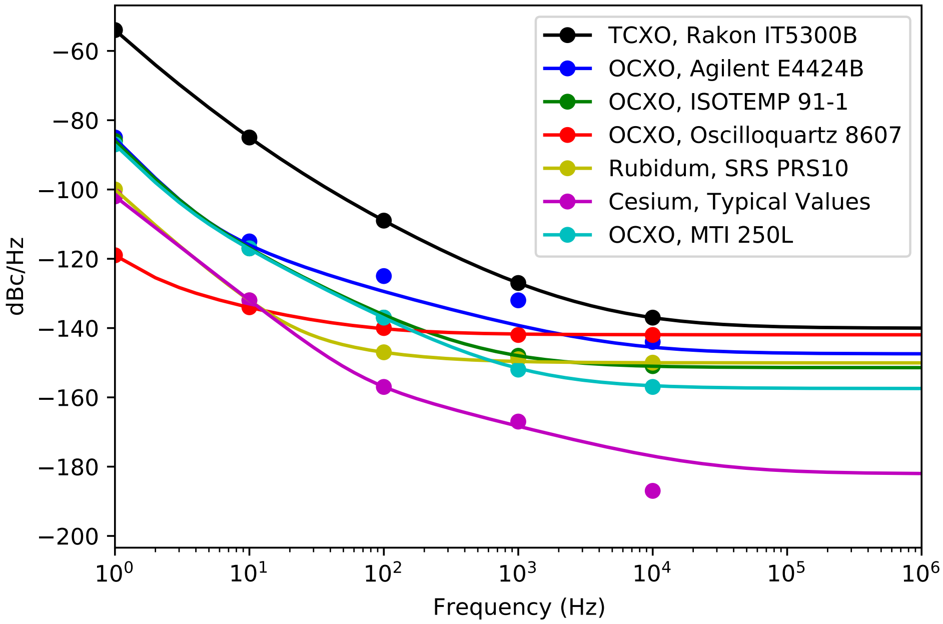


Figure 2.8: Shows the PSDs of the simulated oscillator noise for various oscillators. Lines are the simulated PSD, while points are the phase noise PSD values at each frequency decade from each oscillator's respective specification document. Figure produced using oscillator PSD values and h parameters from (Chiou, 2010).

the time domain phase noise (Michael, 2015). The inverse Fourier transform yields $\nu(t)$. Figure 2.8 shows $\nu(t)$ for each $f^{\alpha-2}$.

$$W_\nu(f) = \left(\frac{f_c}{f_0}\right)^2 S_\nu(f) \tag{2.28}$$

Note that the values listed in (Chiou, 2010) are also unnormalized h parameters. These differ from typical normalized h parameters listed in (Curran, Lachapelle, & Murphy, 2012), which are h_α/f_0^2. f_0 for oscillators in Figure 2.8 is 10MHz, except for the Rakon IT5300B TCXO oscillator which has an f_0 of 16.3676MHz. This is important since the phase noise scales with the received frequency, as shown in Equation 2.28, where $W_\nu(f)$ represents the oscillator phase noise at a receiver center frequency (Chiou, 2010).

Chapter 3

Processing GNSS Signals

As mentioned in the previous chapter, GNSS signals are designed in such a way to be acquired and tracked by correlating known replicas, even when they are well below the noise floor. Successful processing of these signals will involve different stages. First, the signals from each satellite must be acquired. A common approach is to perform a blind search over every possible combination of PRN, Doppler frequency, and code phase. While the Doppler frequency estimate is coarse, this process reveals which satellite signals are trackable and thus which ones can be used for navigation.

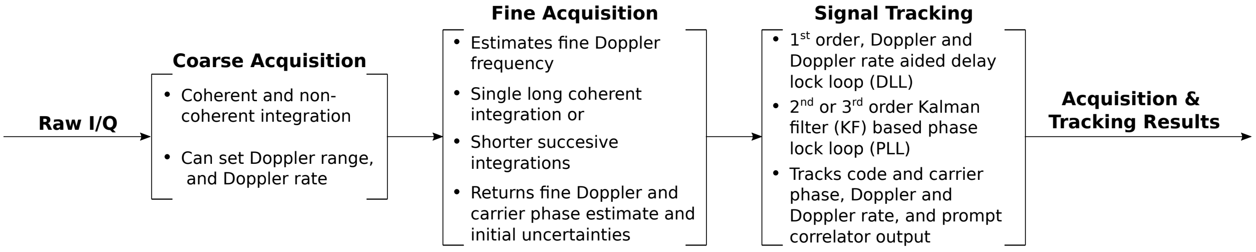

Raw I/Q
Coarse Acquisition
• Coherent and non-coherent integration
• Can set Doppler range, and Doppler rate
Fine Acquisition
• Estimates fine Doppler frequency
• Single long coherent integration or
• Shorter succesive integrations
• Returns fine Doppler and carrier phase estimate and initial uncertainties
Signal Tracking
• 1st order, Doppler and Doppler rate aided delay lock loop (DLL)
• 2nd or 3rd order Kalman filter (KF) based phase lock loop (PLL)
• Tracks code and carrier phase, Doppler and Doppler rate, and prompt correlator output
Acquisition & Tracking Results

Figure 3.1: Diagram showing different processing stages of GNSS signals.

After acquisition, the Doppler frequency estimate needs to be refined in order to perform signal tracking. Signal tracking enables fractional code and carrier phase, Doppler frequency, and Doppler frequency rate estimates of the received signal. The signal tracking stage is what allows for accurate PNT. This chapter will discuss each of these processing stages in more detail. A summary of each one is shown in Figure 3.1.

3.1 Acquisition

Acquisition is an important step during signal processing. Through acquisition, the receiver determines which satellites signals are receivable and their respective approximate Doppler frequency and code phase estimate. This process is made possible due to the correlation properties of the GNSS signal codes. While signal tracking methods provide precise code and carrier phase measurements for PNT, they must be initialized with the coarse estimates from this stage. This section will summarize the acquisition stage in GNSS signal processing, with topics including GNSS signal correlation, coarse acquisition, coherent and noncoherent integration, and fine acquisition.

3.1.1 GNSS Signal Correlation

After down conversion, an incoming signal, $s_r(t)$, contains the sum of signals from each satellite, $s^i(t)$, where i denotes a given satellite and $\nu(t)$ and $w(t)$ are the receiver oscillator phase and thermal noise, respectively.

$$\begin{aligned} s_r(t) &= \left(\sum s^i(t)\right) e^{j\nu(t)} + w(t) \\ &= \left(\sum A^i e^{j\left(2\pi\left\{f_{IF}+f_d^i(t)\right\}t+\phi_o^i+\phi_{code}^i(t)\right)}\right) e^{j\nu(t)} + w(t) \end{aligned} \tag{3.1}$$

The received signal undergoes carrier wipeoff, where the carrier frequency of the signal is removed. In this case, the intermediate frequency component is removed (Misra & Enge, 2012).

$$s_w(t) = s_r(t)e^{-j2\pi f_{IF}t} \tag{3.2}$$

A local replica, $r^i(t)$, is then generated where $\hat{f}_d^i$ and $\hat{\phi}_{code}^i$ are the estimates of the Doppler frequency and code phase for a given PRN, i.

$$r^i(t) = e^{j\left(2\pi\hat{f}_d^i t+\hat{\phi}_{code}^i(t)\right)} \tag{3.3}$$

The circular cross-correlation of the local replica, $r^i(t)$, and the signal, $s_w(t)$, yields the correlation result $R(t)$. Since the received signals being processed are discrete-time signals, the discrete correlation is performed on the discrete time wipeoff signal, $s_w[n]$, and replica, $r[n]$ (Tsui, 2004).

$$\begin{aligned} R[n] &= r^i[n] \circledast s_w[n] \\ &= \sum_{m=0}^{N-1} s_w[m] r^i[n+m] \end{aligned} \tag{3.4}$$

where N is the number of data samples. Since most GNSS constellations use CDMA, such as GPS, each satellite's psueorandom noise code is unique and can be easily distinguished from other satellite codes. Figure 3.2 shows the auto-correlation of the L1 C/A code for PRN 1 and the cross-correlation for the PRN 1 and PRN 2 codes. If the Doppler frequency estimate and PRN are the same, the cross-correlation becomes the auto-correlation and there will be a peak at the beginning of the code sequence, shown in Figure 3.2a. The code was offset by 500 chips. The initial code phase at the beginning of the data segment is denoted n_0. n_0 in Figure 3.2a is 524.

The GNSS PRN codes are such that they provide distinct correlation peaks when the Doppler frequency and code phase estimates are correct. When the PRN code replica does not match the PRN code in the received signal, the result is similar to Figure 3.2b. The expectation value of the auto- and cross-correlation values for a PRN code of length N with amplitude of 1 can be described as the following (Misra & Enge, 2012).

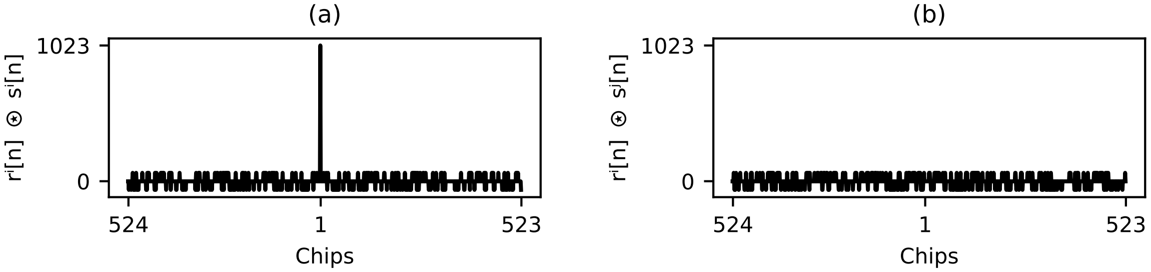


Figure 3.2: Shows the auto-correlation function of the L1 C/A code for PRN 1 (a) and the cross-correlation function of L1 C/A codes for PRN 1 and 2 (b). The PRN 1 code, $s^i[n]$, was offset by 500 chips and the codes have amplitudes of ±1.

$$E\{R[n]\} = \begin{cases} N & n = n_0 \\ 0 & n \neq n_0 \end{cases} \tag{3.5}$$

and the standard deviation of the cross-correlation mean is

$$\sigma_{n \neq n_0} = \sqrt{N} \tag{3.6}$$

The signal-to-noise ratio (SNR) of the peak is the ratio of the signal and noise powers.

$$\frac{S}{N} = \frac{P_S}{P_N} = \left(\frac{S}{\sigma_N}\right)^2 \tag{3.7}$$

where S is the amplitude of the correlation peak, and σ_N is the standard deviation of the noise. There is one more case that will be useful in Section 2, where signal tracking is discussed. In almost all situations, the replica is never initially aligned with any integer chip in the incoming signal. The replica is usually partially aligned with a given chip in the code. If the replica code is partially aligned over the chip n_0, then the expected correlation function is similar to that of correlating two pulses. Switching to the continuous time form of the cross-correlation, $R(\tau)$, and where T_{chip} is the period of a single chip in the GNSS code, the expected correlation is the following. (Misra & Enge, 2012).

$$E\{R(\tau)\} = \begin{cases} N\left(\frac{\tau}{T_{chip}} + 1\right) & -T_{chip} < \tau < 0 \\ N\left(\frac{-\tau}{T_{chip}} + 1\right) & 0 < \tau < T_{chip} \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

3.1.2 Coarse Acquisition

Coarse acquisition is the method by which the coarse Doppler frequency and code phase are estimated. The Doppler frequency of the incoming signal is unknown without any prior information about the user location and GPS constellation status. Cell phones use reference stations, such as

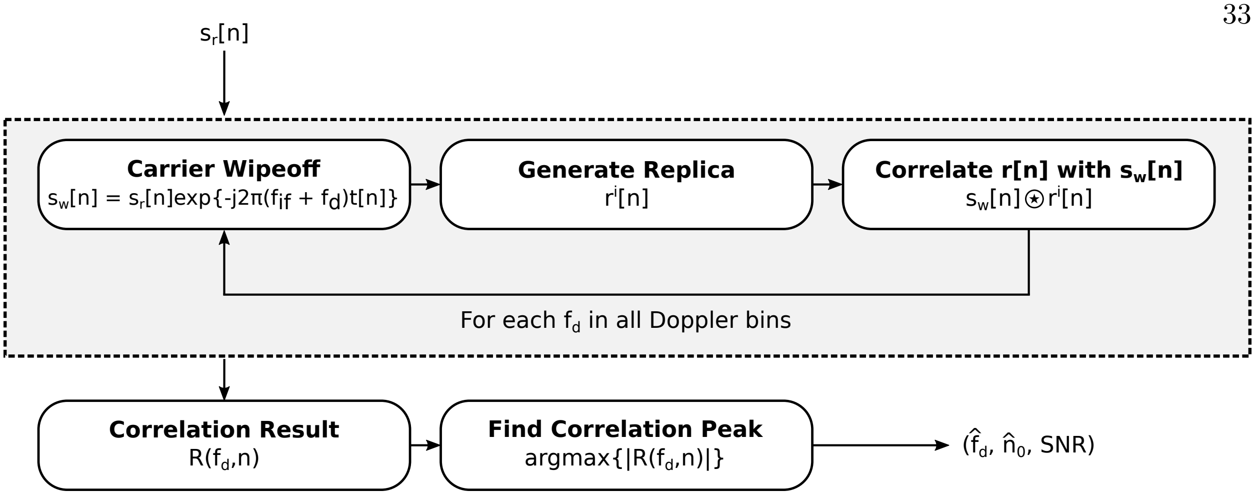


Figure 3.3: Diagram of coarse acquisition of a GNSS signal for a specific PRN. The correlation at each possible combination of Doppler frequency and code phase is performed to produce a two dimensional correlation map, where the peak (if any) is the coarse Doppler frequency and code phase estimate of the received signal.

nearby cellular towers, to acquire this information and quickly acquire GNSS signals (Betz, 2015). However, the methods discussed here assume a standalone GNSS receiver with no prior information about it and the satellite constellation.

Coarse acquisition is performed over each possible code phase and Doppler frequency, since neither are known. While the Doppler frequency is not known, it is expected to fall within a range of Doppler frequencies, between $-f_{d_{max}}$ and $f_{d_{max}}$ (Misra & Enge, 2012). This is dependent on the GNSS constellation that the receiver is processing signals from. For processing signals from GPS satellites, a range of ±5kHz is typically used (Tsui, 2004).

Figure 3.3 shows the process of coarse acquisition. The carrier of the incoming signal is removed along with the current Doppler frequency estimate. A local replica is generated and correlated against the signal that has undergone carrier wipeoff. The correlation result for this Doppler frequency estimate is stored in a 2D array. One axis represents the Doppler frequency bin, while the other represents the code phase. This process is repeated for all frequency bins. The peak, if it exists, represents the coarse estimate of the Doppler frequency and code phase of the incoming signal for that PRN (Tsui, 2004). This process is then repeated again for each PRN. Since only the magnitude of the correlation result is kept (by taking the absolute value of the correlation map), the phase information in the correlation result is lost. As long as the carrier of the local replica is in-phase/quadrature, a mismatch in the initial phase estimate of the signal will not cause fading in the correlation result (Misra & Enge, 2012).

A single frequency bin has a width equal to $1/T$, where T is the integration time, the amount of time being correlated between the signal and local replica (Tsui, 2004). For GPS signals, T is at least 1ms, making the Doppler bin spacing 1kHz. Therefore, 11 Doppler bins would be required to cover a Doppler frequency range of ±5kHz with a 1ms integration time.

The correlation performed in Figure 3.3 can be the trivial implementation of the correlation, known as shift and multiply. Although, this method is significantly slower than the alternative method, which is to perform the correlation in the frequency domain, shown below (Tsui, 2004). The latter method is used in *GNSSTools*.

$$r^i[n] \circledast s_w[n] = \mathcal{F}^{-1}\left\{\overline{\mathcal{F}\left(r[n]\right)}\mathcal{F}\left(s[n]\right)\right\} \tag{3.9}$$

where $\overline{\mathcal{F}\left(r[n]\right)}$ is the complex conjugate of the Fourier transform of the local replica signal and $\mathcal{F}\left(s[n]\right)$ is the Fourier transform of the received signal. The above is performed for each Doppler frequency bin during acquisition.

3.1.3 Probability of Detection

Signal detection is important in determining whether a signal has been acquired or not. Ideally, one would only track signals that they are confident actually exist. Therefore, criteria for successful acquisition, based off the likelihood of detection and false alarms is used to set a threshold. Acquisition peaks above this defined threshold are considered real and subsequently tracked, while peaks below this threshold are considered noise and are not tracked.

There are two situations that can occur when detecting a signal. There is a noise-only case, where no signal exists and a signal-plus-noise case. The noise-only case takes the magnitude of the complex correlation result. The in-phase and quadrature components are each described by Gaus-

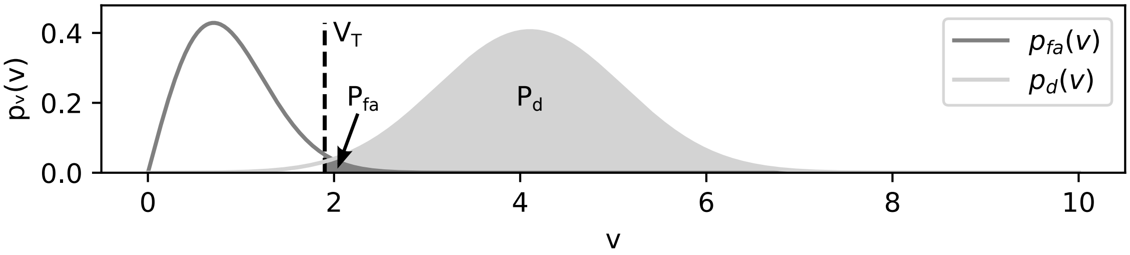


Figure 3.4: Shows the PDFs for the false alarm and detection, $p_{fa}(v)$ and $p_d(v)$, respectively. The area under each curve from V_T to ∞ is the corresponding probability.

sian distributions if there is no signal. However, the magnitude of these components, $\sqrt{I^2+Q^2}$, is part of a Rayleigh distribution, which is used to model the noise-only case of the correlation result. The probability of false alarm, f_{fa} is defined as the area of the Rayleigh distribution, shown in Figure 3.4, between a threshold value, V_T, and ∞ (Richards, Scheer, & Holm, 2016).

$$P_{fa} = \int_{V_T}^{\infty} \frac{v}{\sigma_N^2} e^{-v^2/2\sigma_N^2} dv \tag{3.10}$$

where σ_n is the noise standard deviation. Equation 3.10 results in an analytical relationship between the probability of false alarm, P_{fa} and the threshold value, V_T.

$$V_T = \sqrt{2\sigma_N^2 \ln\left(1/P_{fa}\right)} \tag{3.11}$$

The PDF describing the signal-plus-noise case, denoted $p_d(v)$ in Figure 3.4, is a Rician distribution. The probability of detection, P_d, is also calculated by numerically integrating Equation 3.12 from V_T to ∞ (J. Y. Morton, 2019).

$$P_d = \int_{V_T}^{\infty} \frac{v}{\sigma_N^2} e^{-\left(v^2/2\sigma_N^2+SNR\right)} I_0\left(\frac{v\sqrt{2SNR}}{\sigma_N}\right) dv \tag{3.12}$$

where SNR is the signal-to-noise ratio of the signal, and $I_0(\cdot)$ is the zero order, modified Bessel function of the first kind. P_{fa} can be set first to determine the threshold, V_T. Then, P_d can be calculated numerically. After establishing V_T, values below it will be considered noise and those above it will be considered a real signal. P_{fa} will always be nonzero, unless P_d is 1. If this is so, then any acquisition result will be considered real and signal tracking will begin.

3.1.4 Coherent and Noncoherent Integration

As mentioned above, the integration time is the length of time that a local replica is correlated with an incoming signal. A single correlation, with an integration time of T, is called coherent integration. It has a processing gain of the following (Tsui, 2004).

$$G_c = 10\log_{10}\left(\frac{B_{in}}{B_{out}}\right) \tag{3.13}$$

where B_{in} and B_{out} are the input and output bandwidth, in Hz. The input bandwidth is the bandwidth of the receiver, while the output bandwidth is $1/T$. The carrier-to-noise ratio (C/N_0), in dB·Hz, is related to the bandwidth of the receiver and the signal-to-noise ratio (SNR), in dB.

$$C/N_0 = SNR - 10\log_{10}(B) \tag{3.14}$$

Substituting Equation 3.14 into Equation 3.13, results in Equation 3.15. Here, a coherent integration of time T, in seconds, results in an expected SNR that is a function of C/N_0 and T.

$$SNR_c = C/N_0 + 10\log_{10}(T) \tag{3.15}$$

Longer coherent integration times result in a higher correlation peak SNR. New modern GNSS codes, such as L5, have dedicated code-less channels, also called pilot channels, that enable long coherent integration times. These channels do not have navigation data bits on them, preventing reduced correlation power due to the integration time overlapping data bit flips. Long integration times can also suffer reduced correlation power when the Doppler frequency deviates from the local replica's estimate of the Doppler. In addition, longer integration times result in a significantly greater number of calculations during course acquisition. Since the Doppler bin width is related the coherent integration time by $1/T$, then the Doppler frequency bin width will decrease with increasing integration time. This results in a greater number of Doppler frequency bins, requiring more correlations than required for a shorter integration time. An example of coherent integration is shown in Figure 3.5a, where a local replica of length $2T$ is correlated with a data segment of length $2T$. Note that T in the figure represents the length of the ranging code, making the integration time two code lengths long.

Noncoherent integration is the sum of the power from individual coherent integrations. It can be used to gain additional correlation power while avoiding the effects of data bit transitions. If short coherent integration times are used then the Doppler bin widths are large, resulting in fewer Doppler bins, and thus fewer correlations. Also, the Doppler frequency estimate does need to be precise for long nocoherent times, as opposed to long coherent times, where a drift in the Doppler

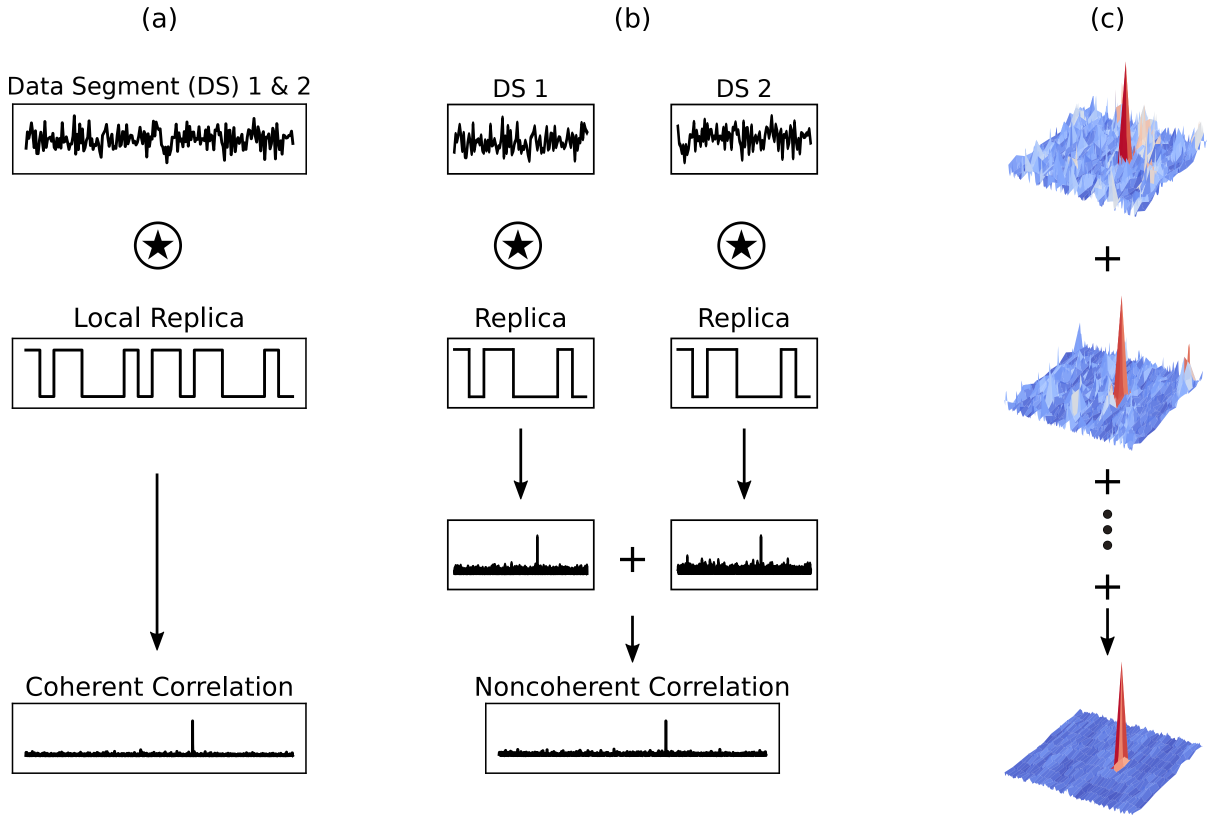


Figure 3.5: Shows coherent integration (a), noncoherent integration (b), and noncoherent coarse acquisition (c), where each correlation map is produced using the method in Figure 3.3 and then added together. Data segments (DS) 1 and 2 are depicted as one code length each.

frequency can result in the Doppler frequency drifting over multiple frequency bins. A noncoherent integration occurs by summing M coherently integrated segments, each with length T. An example of this is shown in Figure 3.5b. Here the data segment in Figure 3.5a is split into two segments. Both are length T. Each one is correlated with a replica and squared element-wise to remove the carrier phase from the correlation result. The results are summed to produce the final noncoherent correlation result.

The noncoherent processing gain is a function of the coherent processing gain and noncoherent integration loss, that are a function of M, the number of noncoherent integrations, each with coherent integration time, T (Tsui, 2004).

$$G_M = 10\log_{10}(M) - L(M) \tag{3.16}$$

$L(M)$ is the noncoherent integration loss term, shown below (Tsui, 2004).

$$L(M) = 10\log_{10}\left(\frac{1+\sqrt{1+9.2MD_c(1)}}{1+\sqrt{1+9.2D_c(1)}}\right) \tag{3.17}$$

where $D_c(1)$ is the ideal detectability factor, shown in Equation 3.18 (Tsui, 2004).

$$D_c(1) = \left\{erf^{-1}(1-2P_{fa}) - erf^{-1}(1-2P_d)\right\}^2 \tag{3.18}$$

where $erf^{-1}(y)$ is the inverse error function, P_{fa} is the probability of false alarm, and P_d is the probability of detection. P_{fa} and P_d are typically around 10^{-7} and 0.9, respectively (Tsui, 2004). If coherent and nocoherent integration is being performed, then the total processing gain is the sum of the coherent and noncoherent processing gains.

$$\begin{aligned} G_T &= G_C + G_M \\ &= 10\log_{10}\left(\frac{B_{in}}{B_{out}}\right) + 10\log_{10}(M) - L(M) \end{aligned} \tag{3.19}$$

3.2 Fine Acquisition

As discussed in Section 3.1.2, the coherent integration time, T, determines the resolution of the frequency bins of the Doppler search area. For an integration time of 1ms, the frequency bins are 1kHz wide. This is often insufficient for a carrier tracking loop, which requires a more precise Doppler frequency estimate than what coarse acquisition provides. The simplest approach is to perform a long coherent integration, which results in smaller frequency bins in the frequency domain. According to Tsui, tracking loops require an initial Doppler frequency estimate with an accuracy of tens of Hz. Therefore, the integration time, T, would need to be on the order of 100ms. For the L1 C/A and the data channels on L2 and L5, multiple navigation data bit flips can occur within this time, diminishing the correlation power (Tsui, 2004). This approach, although supported in *GNSSTools*, requires a greater amount of memory to perform than an alternative method that measures the phase difference between consecutive correlations. This value is proportional to the Doppler frequency of the incoming signal.

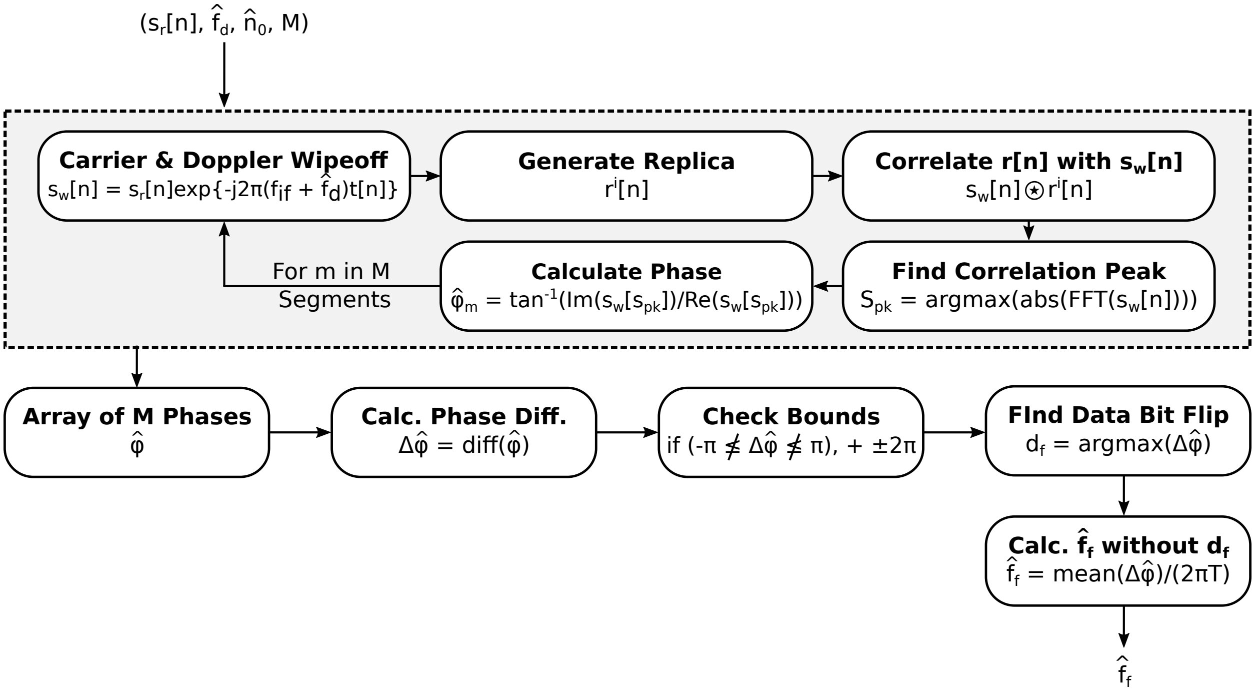

$(s_r[n], \hat{f}_d, \hat{n}_0, M)$
Carrier & Doppler Wipeoff
$s_w[n] = s_r[n]\exp\{-j2\pi(f_{if} + \hat{f}_d)t[n]\}$
Generate Replica
$r^i[n]$
Correlate r[n] with $s_w[n]$
$s_w[n] \circledast r^i[n]$
Find Correlation Peak
$S_{pk} = \text{argmax}(\text{abs}(\text{FFT}(s_w[n])))$
Calculate Phase
$\hat{\varphi}_m = \tan^{-1}(\text{Im}(s_w[s_{pk}])/\text{Re}(s_w[s_{pk}]))$
For m in M Segments
Array of M Phases
$\hat{\varphi}$
Calc. Phase Diff.
$\Delta\hat{\varphi} = \text{diff}(\hat{\varphi})$
Check Bounds
if $(-\pi \nleq \Delta\hat{\varphi} \nleq \pi)$, + ±2π
FInd Data Bit Flip
$d_f = \text{argmax}(\Delta\hat{\varphi})$
Calc. $\hat{f}_f$ without d_f
$\hat{f}_f = \text{mean}(\Delta\hat{\varphi})/(2\pi T)$
$\hat{f}_f$

Figure 3.6: Shows the fine acquisition method described in (Tsui, 2004) and (J. Y. Morton, 2019).

As summarized in Figure 3.6, the coarse Doppler estimate from coarse acquisition is used to again carrier wipeoff M segments of data with integration times, T. The excess Doppler is then determined by measuring the phase in each of the M segments. To do so, the wiped off signal is correlated with a local replica. By doing so, the result is a carrier, assuming no navigation data bit flip, and the value of the peak in the corresponding frequency bin in the frequency domain, Z, will yield the carrier phase (Tsui, 2004).

$$\hat{\phi} = \tan^{-1}\left(\frac{\Im(Z)}{\Re(Z)}\right) \tag{3.20}$$

The difference in phase between consecutive integrations, each with an integration time of T, is related to the residual Doppler frequency, also called the fine Doppler frequency. This is shown in Equation 3.21. In implementation, phase measurements are taken in M integrations. Consecutive phases are differenced to yield a vector of $\Delta\hat{\phi}$, which is averaged to produce the mean phase difference.

$$\hat{f}_f = \frac{\hat{\phi}_i - \hat{\phi}_{i-1}}{2\pi T} \tag{3.21}$$

Since the bandwidth of the integrations, which are separated by T, is $1/T$, there exists an ambiguity for phase differences less than $-\pi$ and above π. This is resolved by adding 2π if $\Delta\hat{\phi}$ is less than $-\pi$ and by subtracting 2π if $\Delta\hat{\phi}$ is greater than π (J. Y. Morton, 2019). This method is an alternative to an FFT approach. Both methods are supported in *GNSSTools.* It should be noted that the carrier phase method in *GNSSTools* assumes that there will only be one navigation data bit flip. While newer GPS signals such as L5 have a pilot channel that contains no data bit flips, there is an Neuman-Hoffman overlay code of length 20ms. If this overlay code is not included in the local replica while using this method, it will create the effect of data bit flips in the resulting phase measurements, flipping them by π. Therefore it is important to align not just to the ranging code but also to the overlay code in order to perform this method successfully.

3.3 GNSS Signal Tracking

After the signal is acquired, tracking algorithms track its code and carrier phase, Doppler frequency, and if desired, its Doppler frequency rate. Results from this stage are used to produce pseudorange and phase measurements, which are used for estimating the user position and time. Various estimation techniques exist and can be deployed to estimate these states through indirect measurements. The residuals of the code and carrier phase are measured at intervals of the integration time, T. A delay lock loop (DLL) and phase lock loop (PLL) are used to track the code and carrier phase. Below is a discussion of this process.

3.3.1 Correlators

A DLL and PLL are used to track the code phase, carrier phase, Doppler frequency, and Doppler frequency rate. Both of these require measurements to be taken that indicate how well the local replica matches the incoming signal. Typically, three correlators, or local replicas, are generated and correlated with the incoming signal. Each correlator is spaced 1/2 the chip width from the other. The prompt correlator is aligned with the most recent estimate of the code phase. The early and late correlators are advanced and delayed by 1/2 the chip width, respectively. This is shown in Figure 3.7.

After correlating these local replicas with the incoming signal, the correlation peak values of the early, prompt, and late correlators become Z_E, Z_P, and Z_L, respectively. These values are used with the discriminators, discussed in the next subsection, to determine the code and carrier phase residuals, or the difference between the estimated and measured states.

3.3.2 Discriminators

Discriminators use the correlator outputs to isolate the desired estimate to measure. Different discriminators are used for the code and carrier tracking loops. For code tracking, the code phase error is measured. There are different discriminators used to extract this error, that include early-minus-late, non-coherent, power, and normalized (Misra & Enge, 2012). In *GNSSTools*, a normalized, early-minus-late discriminator, Equation 3.22, is used to measure the code phase error, $\delta\tau$. The output of the discriminator is related to the code phase error, Equation 3.23 (Y. T. J. Morton et al., 2020a, ch. 15).

$$l_{\tau_4} = \frac{|Z_E - Z_L|}{|Z_E| + |Z_L|} \tag{3.22}$$

$$l_\tau = \frac{2}{T_C}\delta t \tag{3.23}$$

where T_C is the single code chip period and $\delta t = \delta n T_C$, where δn is the code phase error in chips. Equation 3.24 is similar to the carrier phase discriminator, except Z is Z_P in this case. Both atan and atan2 can be used, with the latter providing better accuracy. Although, atan2 is susceptible to bit flips, while atan is not (Y. T. J. Morton et al., 2020a, ch. 15).

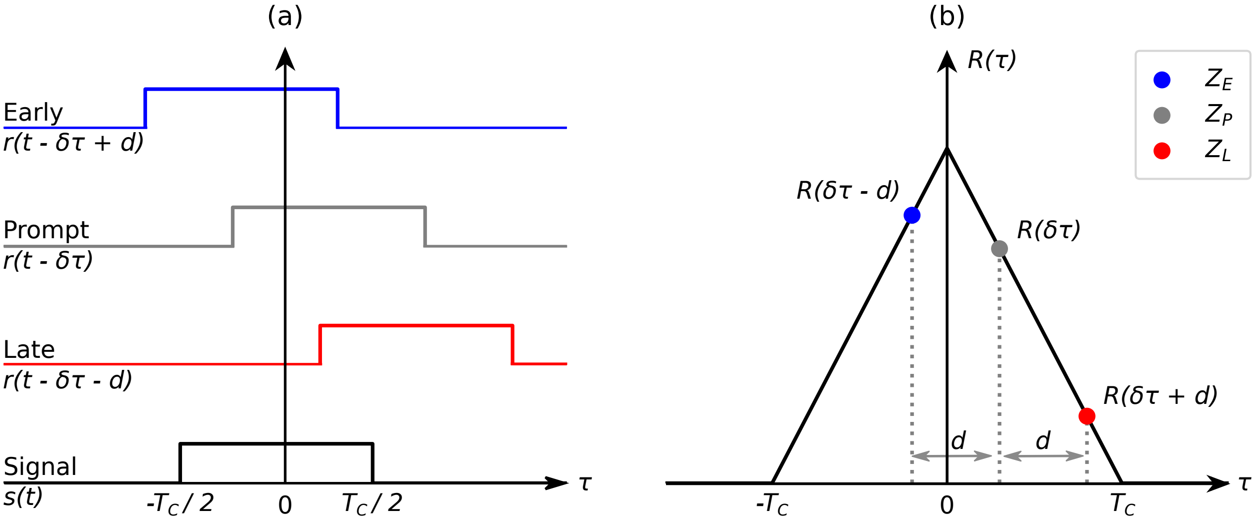


Figure 3.7: Shows how early, prompt, and late correlators are used to estimate the sub-chip code phase. (a) shows the early, prompt, and late correlators and their positions relative to an incoming signal, where the error in the current code phase estimate is $\delta\tau$. (b) shows the auto-correlation correlation peak of $s(t)$ and the values of the correlators. (a) produced using information from (Y. T. J. Morton et al., 2020a, ch. 15). (b) recreated based off Figure 15.6 in (Y. T. J. Morton et al., 2020a, ch. 15).

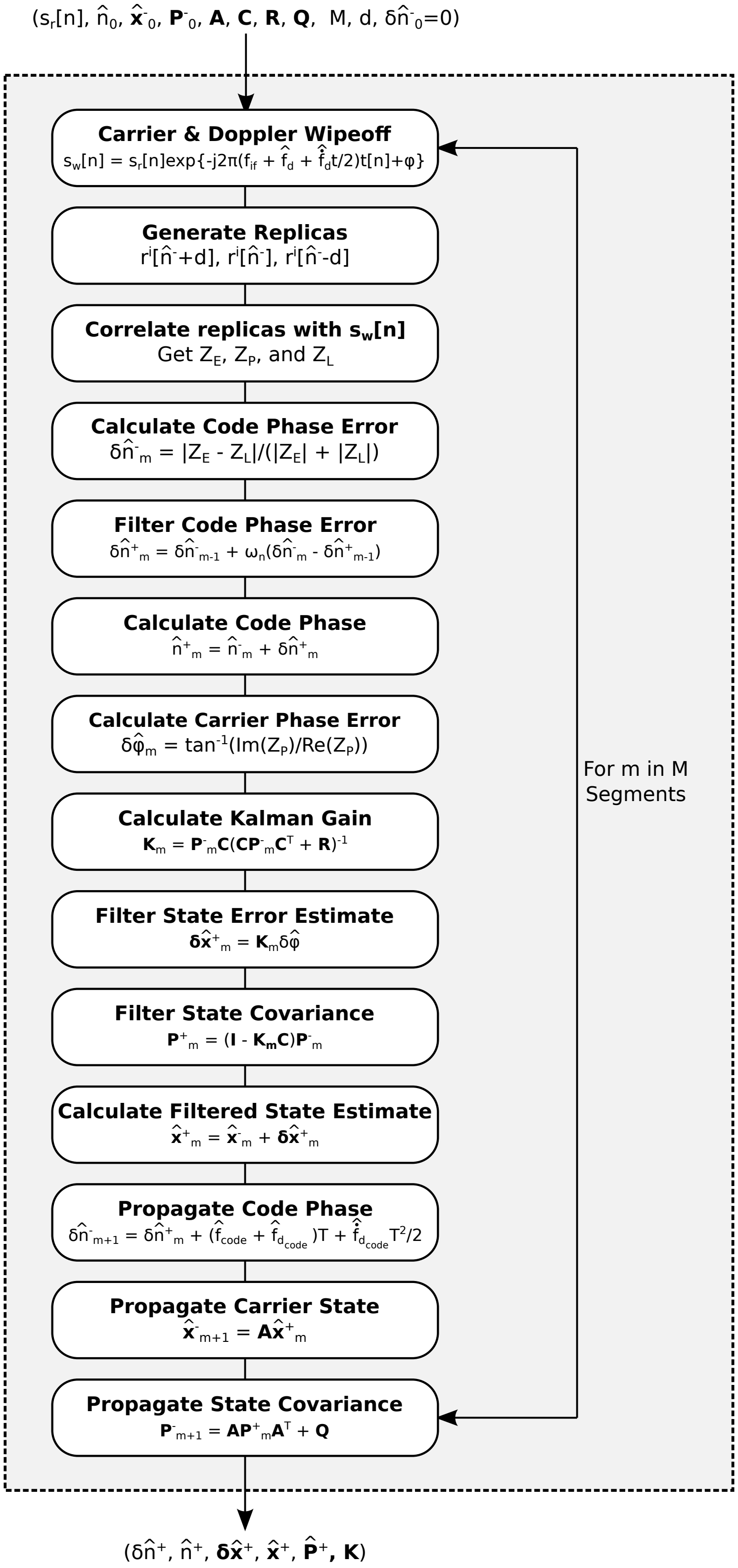

$(s_r[n], \hat{n}_0, \hat{\mathbf{x}}^-_0, \mathbf{P}^-_0, \mathbf{A}, \mathbf{C}, \mathbf{R}, \mathbf{Q}, M, d, \delta\hat{n}^-_0=0)$
Carrier & Doppler Wipeoff
$s_w[n] = s_r[n]\exp\{-j2\pi(f_{if} + \hat{f}_d + \hat{\dot{f}}_d t/2)t[n]+\varphi\}$
Generate Replicas
$r^i[\hat{n}^-+d], r^i[\hat{n}^-], r^i[\hat{n}^--d]$
Correlate replicas with $s_w[n]$
Get Z_E, Z_P, and Z_L
Calculate Code Phase Error
$\delta\hat{n}^-_m = |Z_E - Z_L|/(|Z_E| + |Z_L|)$
Filter Code Phase Error
$\delta\hat{n}^+_m = \delta\hat{n}^-_{m-1} + \omega_n(\delta\hat{n}^-_m - \delta\hat{n}^+_{m-1})$
Calculate Code Phase
$\hat{n}^+_m = \hat{n}^-_m + \delta\hat{n}^+_m$
Calculate Carrier Phase Error
$\delta\hat{\varphi}_m = \tan^{-1}(\mathrm{Im}(Z_P)/\mathrm{Re}(Z_P))$
Calculate Kalman Gain
$\mathbf{K}_m = \mathbf{P}^-_m\mathbf{C}(\mathbf{C}\mathbf{P}^-_m\mathbf{C}^T + \mathbf{R})^{-1}$
Filter State Error Estimate
$\boldsymbol{\delta}\hat{\mathbf{x}}^+_m = \mathbf{K}_m\delta\hat{\varphi}$
Filter State Covariance
$\mathbf{P}^+_m = (\mathbf{I} - \mathbf{K_m}\mathbf{C})\mathbf{P}^-_m$
Calculate Filtered State Estimate
$\hat{\mathbf{x}}^+_m = \hat{\mathbf{x}}^-_m + \boldsymbol{\delta}\hat{\mathbf{x}}^+_m$
Propagate Code Phase
$\delta\hat{n}^-_{m+1} = \delta\hat{n}^+_m + (\hat{f}_{code} + \hat{f}_{d_{code}})T + \hat{\dot{f}}_{d_{code}}T^2/2$
Propagate Carrier State
$\hat{\mathbf{x}}^-_{m+1} = \mathbf{A}\hat{\mathbf{x}}^+_m$
Propagate State Covariance
$\mathbf{P}^-_{m+1} = \mathbf{A}\mathbf{P}^+_m\mathbf{A}^T + \mathbf{Q}$
For m in M
Segments
$(\delta\hat{n}^+, \hat{n}^+, \boldsymbol{\delta}\hat{\mathbf{x}}^+, \hat{\mathbf{x}}^+, \hat{\mathbf{P}}^+, \mathbf{K})$

Figure 3.8: Shows a diagram of the signal tracking loop. $\hat{f}_{d_{code}}$ and $\hat{\dot{f}}_{d_{code}}$ are calculated using Equations 2.11 and 2.12.

$$\delta\phi = tan^{-1}\left(\frac{\Im(Z_P)}{\Re(Z_P)}\right) \tag{3.24}$$

The code and carrier phase error measurements above are used with the DLL and PLL, discussed below, to track not only the code and carrier phase, but also the Doppler frequency and Doppler frequency rate. Additional discriminators for measuring the code and carrier phase can be found in (Y. T. J. Morton et al., 2020a, ch. 15).

3.3.3 Code and Carrier Tracking

The code and carrier phase measurements are used to estimate the pseudorange between the receiver and the satellite and estimate the receiver's position. In order to be precise, a delay and phase lock loop are used to track the code and carrier phase, respectively. A proportional-integral filter (PIF) is one implementation of the DLL and PLL (Yang, Ling, et al., 2017). A first order DLL tracks the code phase and its responsiveness is related to its explicitly defined bandwidth. The bandwidth should be set higher when the signal dynamics are high. However, when there are low signal dynamics, the filter will yield noisier filtered estimates than those from a filter with a lower bandwidth. A rate-aided DLL is aided by the PLL, resulting in a lower bandwidth requirement. As the PLL tracks the carrier phase and Doppler frequency, the Doppler estimate is used to propagate the code phase state to the next time step (Y. T. J. Morton et al., 2020a, ch. 15).

$$\delta\hat{n}_m = \delta\hat{n}_{m-1} + T\left(f_{d_{code}} + \omega_n\left(\delta n_m - \delta\hat{n}_{m-1}\right)\right) \tag{3.25}$$

where T is the integration time and $\omega_n = 4B$, where B is the bandwidth of the filter. δn and $\delta\hat{n}$ are the measured and filtered code phase errors, respectively (Y. T. J. Morton et al., 2020a, ch. 15). If the Doppler rate is estimated, a modified version of Equation 3.25 is used.

$$\delta\hat{n}_m = \delta\hat{n}_{m-1} + T\left(f_{d_{code}} + \frac{1}{2}\dot{f}_{d_{code}}T + \omega_n\left(\delta n_m - \delta\hat{n}_{m-1}\right)\right) \tag{3.26}$$

The PLL can track the carrier phase, Doppler, and Doppler rate, if it is third order. There are various implementations for it, including PIF, Wiener (WF), and Kalman filter (KF). A second order filter is typically used, which tracks only the carrier phase and Doppler, and can be used to aid the first order DLL (Yang, Ling, et al., 2017). Third order filters are desired for tracking LEO satellite signals because these signals can have a significantly higher Doppler rate than those observed from MEO satellite signals, as shown in Figure 1.3. Figure 3.8 outlines the signal tracking process.

3.3.4 Carrier Tracking: Kalman Filter Approach

As mentioned above, there are several implementations of the carrier tracking loop. PIF, WF, and KF filters are discussed in detail with generalized PLLs in (Yang, Ling, et al., 2017) and compared against each other in (Yang, Morton, et al., 2017). An application of the KF approach is in (Wang et al., 2020). This section is paraphrased from the references cited above, which were instrumental in implementing this in *GNSSTools*.

Standard PIF based PLLs are known as "traditional" PLLs. Their performance is primarily controlled by setting the filter bandwidth and dampening factors. However, adaptive PIF based PLLs exist and significantly outperform fixed gain PIF filters in dynamic, low C/N_0 signal environments. Another adaptive PLL filter are KFs. These filters have an equivalent filter bandwidth, but their steady-state gains are set based off the expected receiver oscillator h-parameters, h_{-2} and h_0, and signal dynamics, q_a. The state space model for this filter is the following:

$$\mathbf{x}_{m+1} = \mathbf{A}\mathbf{x}_m + \mathbf{w}_m \tag{3.27}$$

$$\mathbf{y}_m = \mathbf{C}\mathbf{x}_m + \mathbf{v}_m \tag{3.28}$$

where $\mathbf{A}$ is the state transition matrix, $\mathbf{C}$ is the measurement model. The value of $\mathbf{x}_m$ depends on whether the two- or three-state model is used, Equations 3.29 and 3.30, respectively.

$$\mathbf{x}_{2-state} = \begin{bmatrix} \phi_m & \omega_m \end{bmatrix}^T \tag{3.29}$$

$$\mathbf{x}_{3-state} = \begin{bmatrix} \phi_m & \omega_m & \dot{\omega}_m \end{bmatrix}^T \tag{3.30}$$

where ϕ is the carrier phase, ω is the carrier angular rate, and $\dot{\omega}$ is the carrier angular acceleration. Note that $\omega = 2\pi\left(f_{IF} + f_d\right)$ The noise sources, $\mathbf{w}_m$ and $\mathbf{v}_m$ are the process and measurement noises, respectively. The process noise reflects the random dynamics that occur as the states propagate forward in time. Since the noise is modelled as Gaussian, $\mathbf{w}_m \sim \mathcal{N}(0, \mathbf{Q})$, where $\mathbf{Q}$ is the covariance matrix. The covariance for the two- and three-state PLL are Equations 3.31 and 3.32, respectively.

$$\mathbf{Q}_{2-state} = (2\pi f_c)^2 \begin{bmatrix} Tq_\phi + \frac{T^3}{3}q_\omega & \frac{T^2}{2}q_\omega \\ \frac{T^2}{2}q_\omega & Tq_\omega \end{bmatrix} \tag{3.31}$$

$$\mathbf{Q}_{3-state} = (2\pi f_c)^2 \begin{bmatrix} Tq_\phi + \frac{T^3}{3}q_\omega + \frac{T^5 q_a}{20c^2} & \frac{T^2}{2}q_\omega + \frac{T^4 q_a}{8c^2} & \frac{T^3 q_a}{6c^2} \\ \frac{T^2}{2}q_\omega + \frac{T^4 q_a}{8c^2} & Tq_\omega + \frac{T^3 q_a}{3c^2} & \frac{T^2 q_a}{2c^2} \\ \frac{T^3 q_a}{6c^2} & \frac{T^2 q_a}{2c^2} & \frac{T q_a}{c^2} \end{bmatrix} \tag{3.32}$$

where q_ϕ and q_ω are the contribution by the oscillator phase noise and are related to the oscillator h-parameters, h_0 and $h_{-}{-}2$. q_a is the acceleration squared due to platform dynamics in $\left(m^2/s^6\right)/Hz$. c is the speed of light.

$$q_\phi = \frac{h_0}{2} \tag{3.33}$$

$$q_\omega = 2\pi^2 h_{-2} \tag{3.34}$$

The state transition matrix matrix, $\mathbf{A}$, propagates the state vector $\mathbf{x}_m$ to $\mathbf{x}_{m+1}$, in time increments of T. At the end of each PLL loop, the estimated state is propagated to the next time step, where the predicted state is used to generate the replica signal.

$$\mathbf{A}_{2-state} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \tag{3.35}$$

$$\mathbf{A}_{3-state} = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \tag{3.36}$$

where T is the integration time. The measurement matrix, $\mathbf{C}$, maps the states to the measurements. The two- and three-state measurement matrices are Equations 3.37 and 3.38.

$$\mathbf{C}_{2-state} = \begin{bmatrix} 1 & T \end{bmatrix} \tag{3.37}$$

$$\mathbf{C}_{3-state} = \begin{bmatrix} 1 & \frac{T}{2} & \frac{T^2}{6} \end{bmatrix} \tag{3.38}$$

The measurement obtained at each integration period is the difference between the true and expected carrier phase, $\delta\phi$, since most of the carrier phase is removed during carrier wipeoff. The measurement noise, v_m, is also considered Gaussian and is $\mathbf{v}_m \sim \mathcal{N}(0, \mathbf{R})$, where $\mathbf{R}$ is the measurement covariance matrix.

$$\mathbf{R} = \sigma^2_{\delta\phi} = \frac{1}{2TC/N_0}\left(1 + \frac{1}{2TC/N_0}\right) \tag{3.39}$$

The state estimate, $\mathbf{x}_m$, also has uncertainty, which is modelled as $\mathbf{x}_m \sim \mathcal{N}(0, \mathbf{P}_m)$, where $\mathbf{P}$ is the state covariance matrix. In the iterative KF approach, $\mathbf{P}_m$ varies over m steps, which results in adjustment of the Kalman gain, $\mathbf{K}_m$, At each time step, the most recent, filtered state estimate, $\mathbf{x}_m^+$, is propagated to the next time step using the following:

$$\mathbf{x}_{m+1}^- = \mathbf{A}\mathbf{x}_m^+ \tag{3.40}$$

This produces $\mathbf{x}_{m+1}^-$, which is used to generate a local replica. This replica is then correlated with the incoming signal and the code and carrier phase errors are measured, δn_{m+1}^- and $\delta\phi_{m+1}^-$, respectively. $\mathbf{x}_{m+1}^{-}$ has a corresponding state covariance matrix of $\mathbf{P}_{m+1}^{-}$, which is determined by propagating $\mathbf{P}_{m}^{+}$ using the following:

$$\mathbf{P}_{m+1}^{-} = \mathbf{A}\mathbf{P}_{m}^{+}\mathbf{A}^{T} + \mathbf{Q} \tag{3.41}$$

The propagated state covariance is then used with the phase error measurement to calculate the Kalman gain, $\mathbf{K}_{m+1}$, and afterwards, determine the corrected state covariance, $\mathbf{P}_{m+1}^{+}$.

$$\mathbf{K}_{m+1} = \left(\mathbf{P}_{m+1}^{-}\mathbf{C}\right)\left(\mathbf{C}\mathbf{P}_{m+1}^{-}\mathbf{C}^{T} + \mathbf{R}\right)^{-1} \tag{3.42}$$

$$\mathbf{P}_{m+1}^{+} = \left(\mathbf{I} - \mathbf{K}_{m+1}\mathbf{C}\right)\mathbf{P}_{m+1}^{-} \tag{3.43}$$

Finally, the state estimate is corrected.

$$\mathbf{x}_{m+1}^{+} = \mathbf{x}_{m+1}^{-} + \mathbf{K}_{m+1}\delta\phi_{m+1} \tag{3.44}$$

The corrected state, $\mathbf{x}_{m+1}^{+}$ is then propagated to the next time step using Equation 3.40, and used to generate the local replica. Using the Ricatti equation, the steady state Kalman gain, the point at which the filter settles to an optimal gain, can be determined. The h-parameters, h_{-2} and h_0, the platform dynamics, q_a, and the measurement covariance, $\mathbf{R}$, will determine the value of the steady state gain, $\bar{\mathbf{K}}$.

3.3.5 Equivalent Bandwidth of the KF PLL

The PIF based PLL has a bandwidth that is explicitly defined. It is either static or variable, the latter achieved using adaptive PLL algorithms. The KF based PLL has an implicitly defined bandwidth, where the steady state bandwidth is indirectly set by q_a, h_0, h_{-2}, and $\mathbf{R}$ (Yang, Ling, et al., 2017). However, the equivalent bandwidth can be estimated by obtaining the transfer function of the KF and integrating over all frequency. The transfer function of the KF based PLL is similar to that of the PIF based PLL. For the two state Kalman gain at time step m, $\mathbf{K}_m = \begin{bmatrix} k_1 & k_2 \end{bmatrix}^T$. The equivalent bandwidth of the KF is the following (Won, Pany, & Eissfeller, 2012).

$$B_k = \frac{1}{4}\left(k_1 + \frac{k_2}{k_1}\right) \tag{3.45}$$

In order to determine the equivalent bandwidth of the three state KF based PLL, its transfer function is first determined from Figure 3.9. Figure 3.9 is similar to Figure 2 in (Yang, Ling, et al., 2017), although the filter gains are changed to the variable Kalman gains for the three state KF, $\mathbf{K}_m = \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix}^T$.

$$H(s) = \frac{\hat{\phi}}{\phi} = \frac{k_1 s^2 + k_2 s + k_3}{s^3 + k_1 s^2 + k_2 s + k_3} \tag{3.46}$$

The equivalent noise bandwidth of the filter at time step m is the bandwidth of an equivalent boxcar filter (Y. T. J. Morton et al., 2020a, ch. 15). The integration of Equation 3.47, results in Equation 3.48 (O'Driscoll, Petovello, & Lachapelle, 2010; Brown, 2012).

$$B_k = \frac{1}{|H(0)|^2} \int_{-j\infty}^{j\infty} H(s)H(-s)\frac{ds}{2\pi j} \tag{3.47}$$

$$= \frac{1}{4}\left(k_1 + \frac{k_2^2}{k_2 k_1 - k_3}\right) \tag{3.48}$$

The equivalent noise bandwidth of the two-state KF based PLL gain at steady state was estimated, shown in Figure 3.10.

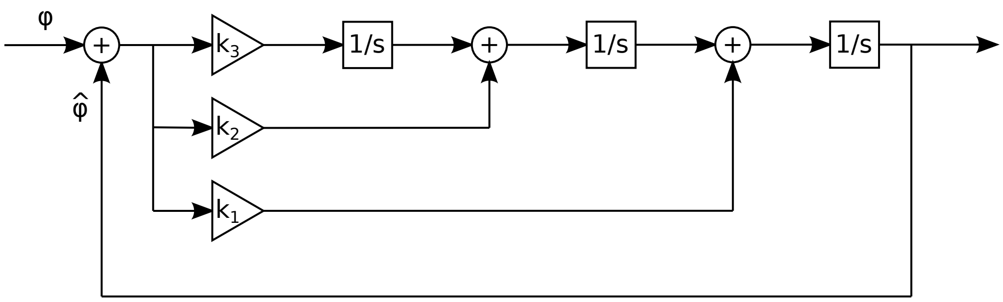


Figure 3.9: Shows the analog three state PLL where the gains are replaced by the variable Kalman gains, k_1, k_2, and k_3.

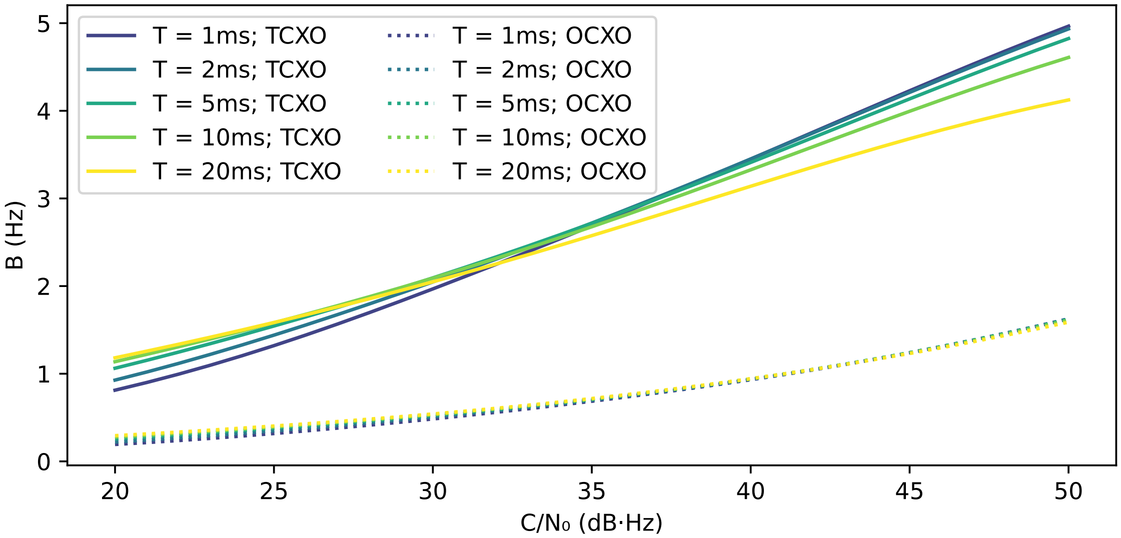


Figure 3.10: Shows the equivalent bandwidth for the two-state KF PLL for various C/N_0, T, and oscillator h-parameters (TCXO and OCXO).

Here, the steady state Kalman gain was calculated for various C/N_0, q_a, T, and oscillator h-parameters. The two-state gains were calculated and used to calculate the equivalent noise bandwidth. Figure 3.10 shows that the equivalent bandwidth of the KF increases with increasing C/N_0. This is consistent to what is shown in Figure 6 in (Sun, Qin, & Niu, 2013).

Chapter 4

Results

A suite of simulation and processing functions in *GNSSTools* are used on both real and simulated L1 C/A and L5 signals. L5 signals have two bands, the data (I) and pilot (Q) channels. Since the pilot channel has no databits, integration times of 1ms or any integer multiple of the length of the Neuman-Hofman sequence (20ms) are possible without databit flips. For all L5 signals, only the L5Q channel is processed.

This chapter contains three major sections. The first presents evaluations of the runtime of major stages such as simulation, acquisition, and tracking. It also presents the performance of the acquisition and tracking stages by comparing experimentally determined values of the probability of detection, P_d, and phase noise jitter, σ_ϕ, with their respective theoretical values. The second section presents acquisition and tracking results from both real and simulated data. The third and fourth sections compare the acquisition and tracking performance for four constellations, GPS, Iridium, Starlink, and OneWeb. Finally, the fifth section discusses a way to narrow the Doppler and Doppler rate distributions, shown in Figure 1.3.

4.1 Evaluation

GNSSTools is evaluated to understand its performance in three main areas. The most important are the acquisition and tracking performance. However, the runtime required to simulate and process signals is useful as well, as there are many factors to these processes that significantly affect the length of time spent on simulation, acquisition, and tracking.

4.1.1 Acquisition Performance

The probability of detection, P_d, is used as the metric to compare between the expected and experimental acquisition performance. For a given C/N_0, receiver noise temperature, T_{sys}, sampling rate, f_s, bandwidth, B, and integration time, T, the amplitude and thermal noise standard deviations are Equations 4.1 and 4.2, respectively.

$$A = \sqrt{2kT_{sys}}10^{C/N_0/20} \tag{4.1}$$

$$\sigma = \sqrt{kBT_{sys}} \tag{4.2}$$

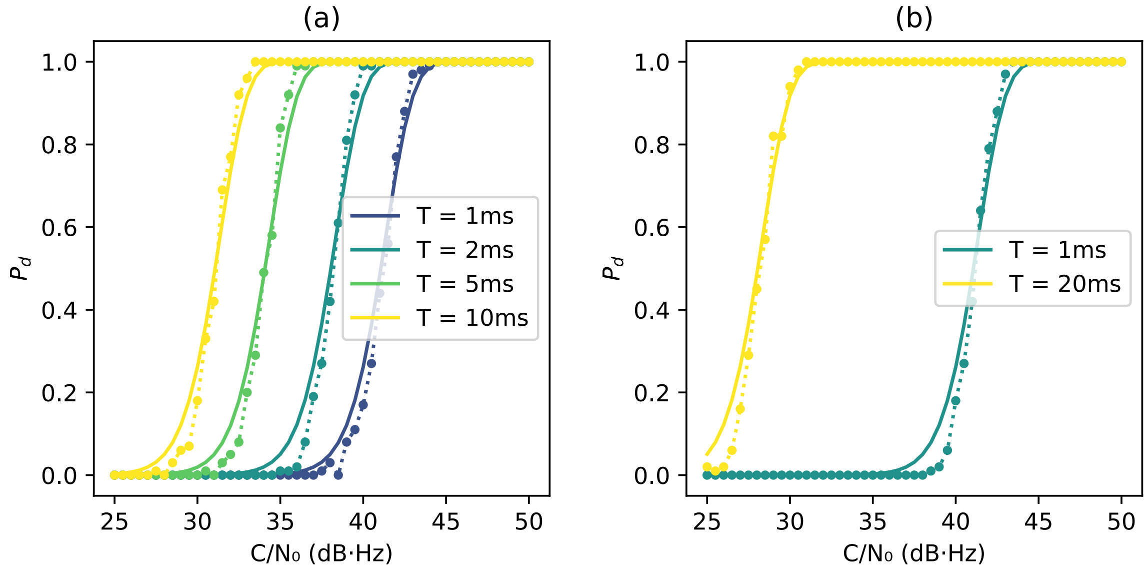


Figure 4.1: Shows the theoretical P_d (solid lines) and the P_d based off 100 simulated signals (dotted lines) for various integration times, T. (a) shows results for the L1 C/A code, while (b) shows results for the L5Q signal.

When the signal is coherently integrated by a local replica after undergoing carrier wipeoff, the peak power, P_S and noise power, P_N are Equations 4.3 and 4.4, respectively.

$$P_S = \frac{(NA)^2}{2} \tag{4.3}$$

$$P_N = N\sigma^2 \tag{4.4}$$

where N is the number of samples coherently integrated, shown in Equation 4.5. The ratio of P_S, and P_N is the expected SNR, as shown in Equation 3.7.

$$N = f_S T \tag{4.5}$$

The expected SNR is used with Equation 3.12 to determine the expected P_d. The experimental P_d of the acquisition result is determined by comparing the peak value to the threshold value, V_T over M iterations. For each iteration, V_T, is calculated based off the noise standard deviation of the correlation result and the probability of false alarm, P_{fa}, which is set to 10^{-7}, using Equation 3.11. If the peak value is above V_T, then the acquisition is considered successful, otherwise, it is considered failed. The number of successes over the total number of iterations is the experimental P_d. The Figure 4.1 shows the results for the L1 C/A (a) and L5Q (b) codes for various integration time, T. The theoretical and experimental P_d agree, indicating that the acquisition process is performing as expected.

4.1.2 Tracking Performance

Tracking performance is measured by comparing the theoretical and measured phase noise jitter, $\Delta\phi$. The total filtered phase noise jitter is a function of the phase jitter due to wideband noise, σ_ϕ, colored noise, σ_{ϕ_c}, and dynamics, σ_{ϕ_d}, shown below. σ_ϕ is the total phase noise jitter at the output of the PLL (Razavi, Gebre-Egziabher, & Akos, 2008). The noise variance at the input of the PLL, or the output of the arctan discriminator is determined by Equation 3.39.

$$\sigma_\phi = \sqrt{\sigma_{\phi_0}^2 + \sigma_{\phi_c}^2} + \frac{\sigma_{\phi_d}}{3} \tag{4.6}$$

$$\sigma_{\phi_0}^2 = \frac{B}{C/N_0\,(1-2BT)}\left(1+\frac{1}{2TC/N_0)}\right) \tag{4.7}$$

$$\sigma_{\phi_c}^2 = \frac{0.9048\,(2\pi)^3\,h_{-2}}{B^3} + \frac{0.8706\,(2\pi)^2\,h_{-1}}{B^2} + \frac{1.2566\,(2\pi)\,h_0}{B} \tag{4.8}$$

σ_{ϕ_0} is the phase jitter due to white noise, while $\sigma_{\phi_c}^2$ is phase jitter due to oscillator phase noise. For these simulations, the "receiver" is considered stationary, thus the dynamics are zero, making σ_{ϕ_d} zero as well. A signal of 1 second is simulated and tracked 25 times for each combination of C/N_0 and T. The unfiltered and filtered σ_ϕ are measured by calculating the standard deviation of the $\tan^{-1}$ discriminator and KF filter phase outputs, respectively, when at steady state. The steady state Kalman gain is used with Equations 4.6-4.8 to calculate the expected filtered phase error, while Equation 3.39 is used to calculate the expected unfiltered phase error.

Figures 4.2 and 4.3 show the performance of the 2-state tracking loop when working with L1 C/A signals (a) and L5Q signals (b). Figure 4.2 shows the unfiltered phase noise jitter, while Figure 4.3 shows the filtered phase noise jitter. Theoretically, the jitter should be larger with small

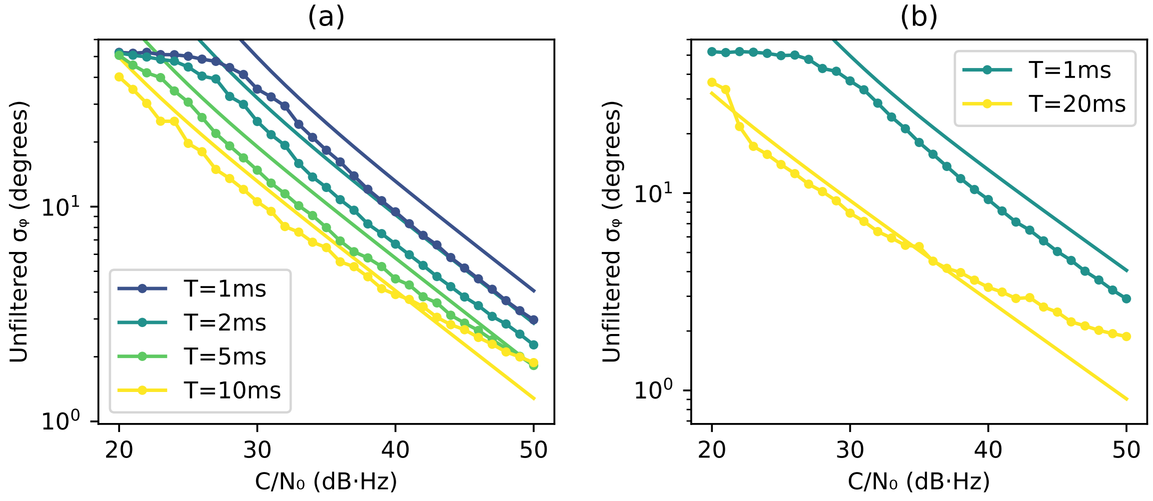


Figure 4.2: Shows the tracking performance for the L1 C/A code (a) and L5Q code (b). Solid lines are the theoretical unfiltered σ_ϕ, while dotted are the average unfiltered σ_ϕ from tracking 25 simulated signals per pair of C/N_0 and T.

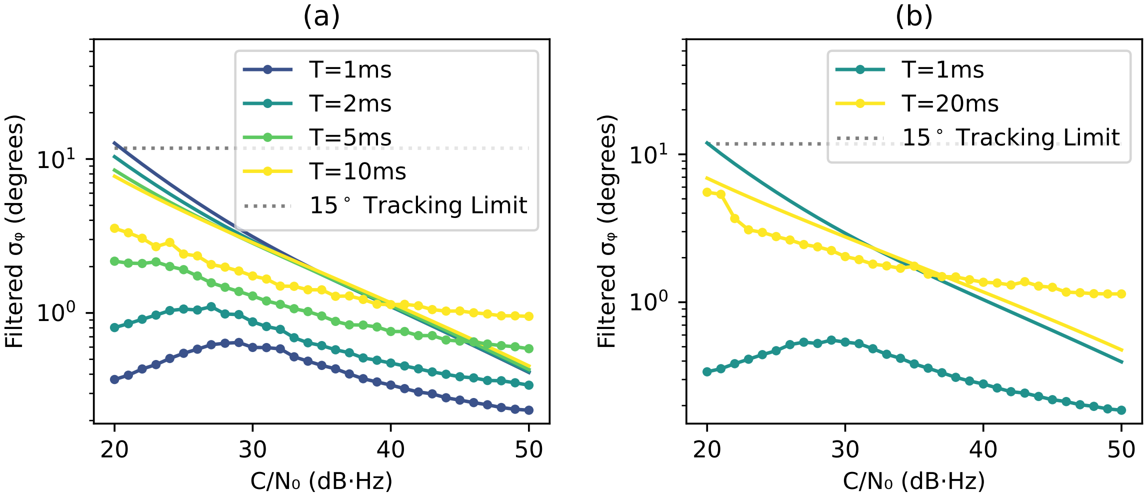


Figure 4.3: Shows the tracking performance for the L1 C/A code (a) and L5Q code (b). Solid lines are the theoretical filtered σ_ϕ, while dotted are the average filtered σ_ϕ from tracking 25 simulated signals per pair of C/N_0 and T.

T. This is the case for the unfiltered phase noise jitter. For the filtered jitter, it appears the that the noise jitter increases with increasing T, rather than decreases. This differs from the value calculated using Equation 4.6, which shows σ_ϕ decreasing with increasing T. The author was not able determine the reason for the differences between the expected and measured filtered σ_ϕ.

4.1.3 Speed of Signal Simulation, Acquisition, and Tracking Stages

The signal simulation, acquisition, and tracking stages were benchmarked to determine the expected runtime of each stage with varying cases. Benchmarks were performed on an Intel® Core™ 1.90GHz 4 core i7-8650U CPU. The absolute runtimes may change with each computer, but it is not expected that the relative change in runtimes will.

Figure 4.4 shows the runtimes for signal simulations of various length of time, sampling frequency, and number of codes in each signal. For example, the L1 C/A signal has one ranging code and an overlay of navigation databits. L5I has a ranging code, Neuman-Hofman sequence, and databits. L5Q only has the ranging code and Neuman-Hofman sequence overlay code. The top row shows the runtime for a simulated signal with thermal and phase noise sources and ADC

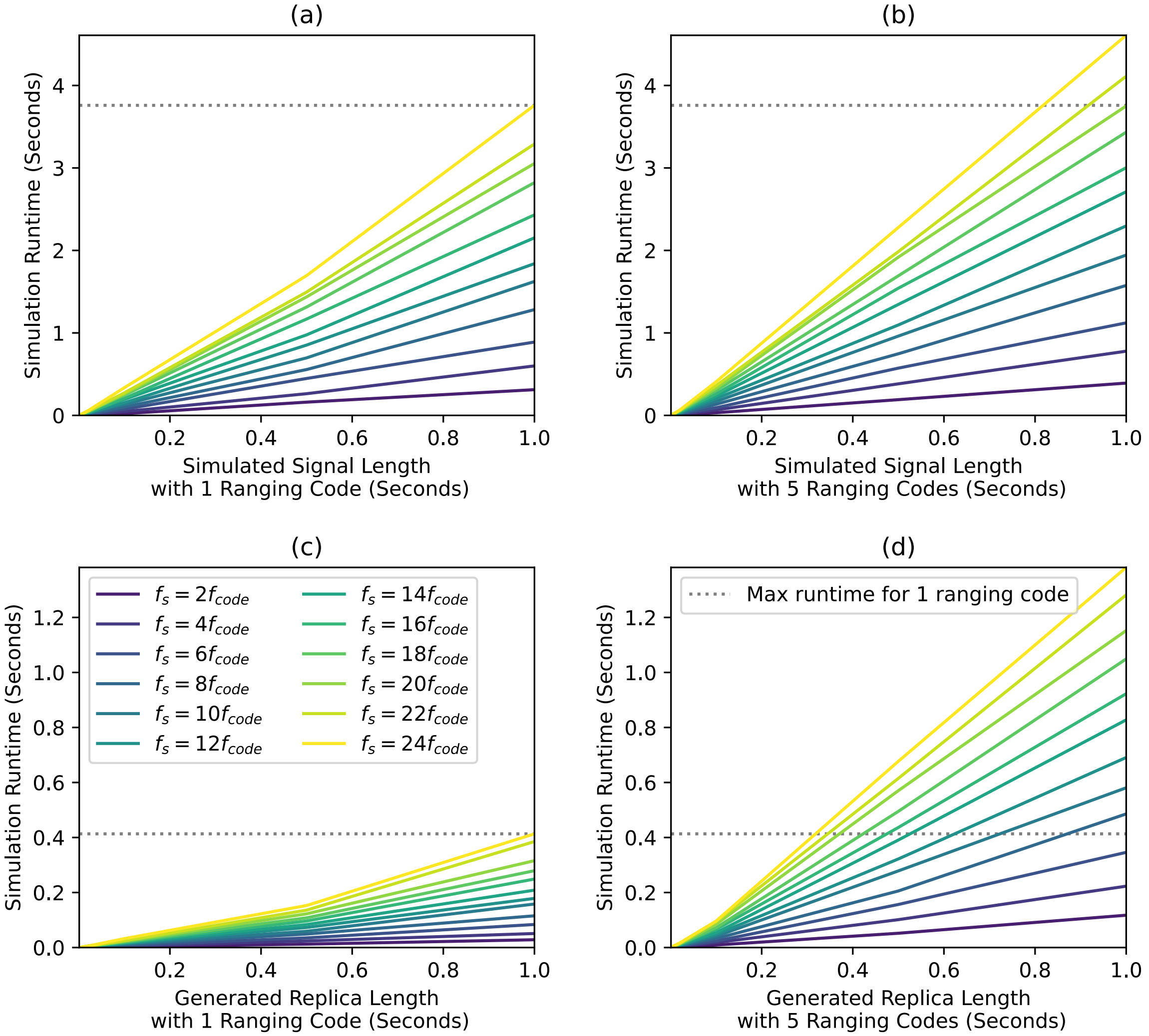


Figure 4.4: Shows the runtime for generating signals with noise sources (top row) and local replicas, without noise sources (bottom row).

quantization. The bottom row shows the runtime for local replicas that do not have noise and are not quantized. Simulating the noise sources and quantizing the signal results in a significant increase of time required for simulation, approximately 4 times that of the local replica. For a signal with five code layers, a local replica with a sampling rate up to $18f_{code}$, or 18 times the primary ranging code chipping rate, can be generated in real time.

Figure 4.5 shows the runtimes for the acquisition stage. The top row shows the runtimes for

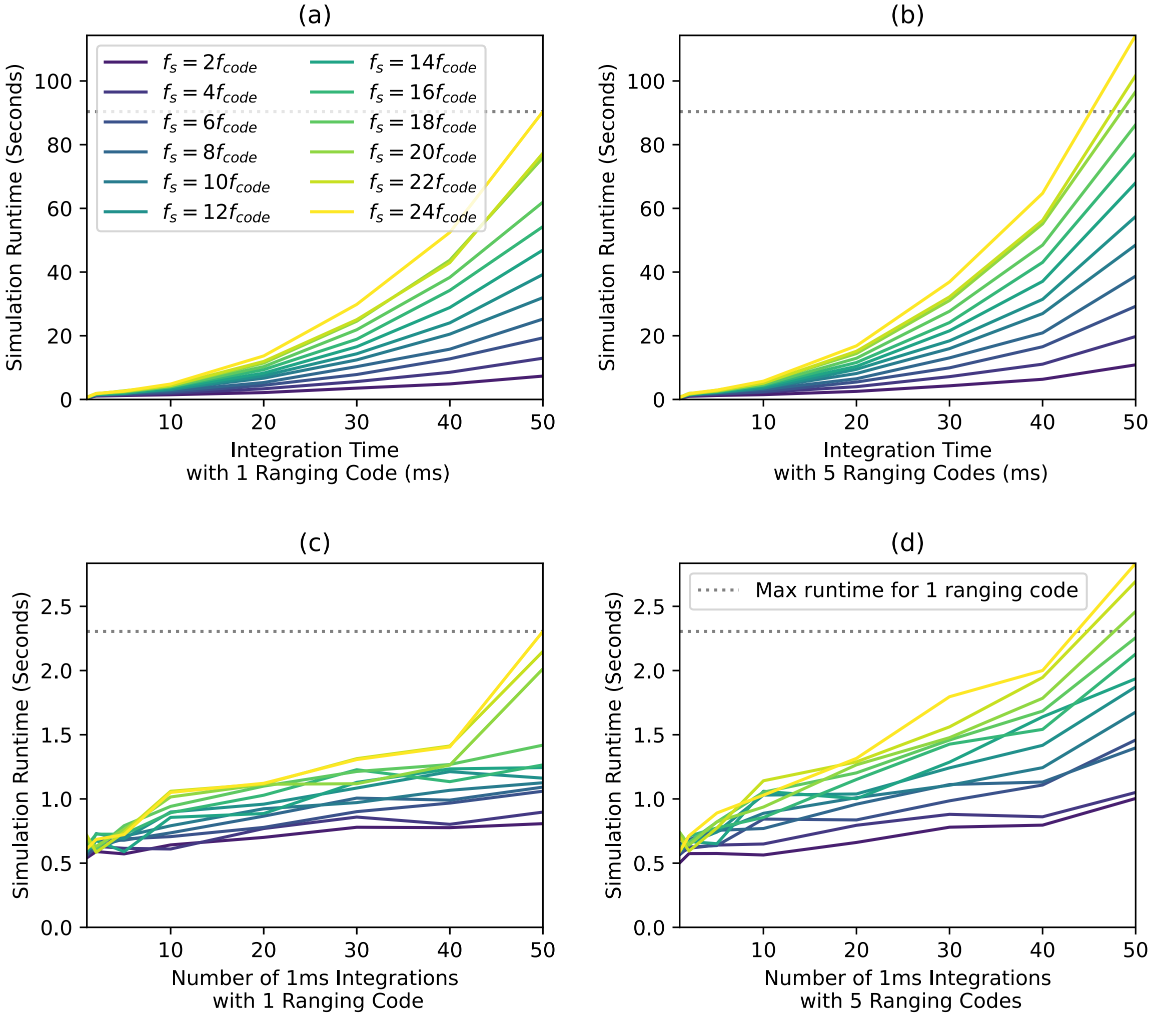


Figure 4.5: Shows the runtime for performing coherent integrations of time T (top row) and M noncoherent integrations (bottom row).

various coherent integration times, while the bottom row shows the runtimes for various numbers of noncoherent integrations, each with an integration time of 1ms. An acquisition stage relying on long coherent integrations will be significantly slower than a process that noncoherently integrates over the same time frame. As the coherent integration time goes up, the bin width goes down, since $\Delta f = 1/T$. Therefore, the number of Doppler frequency bins to search increases with T. Figure 4.6 shows the runtimes for the tracking stage. Here, the significant factor in the runtime of the

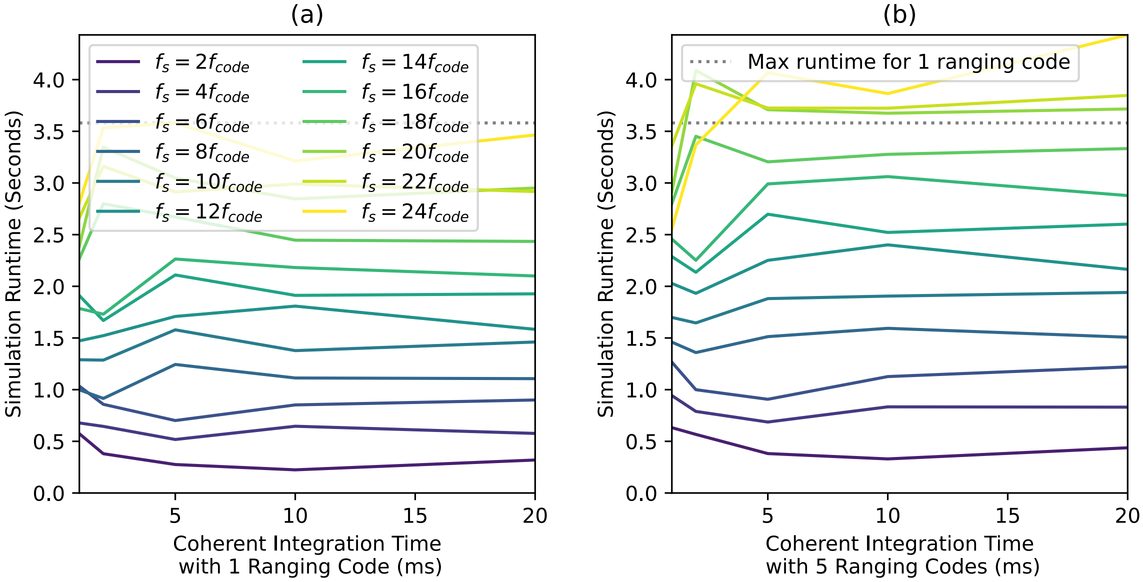


Figure 4.6: Shows the runtime for tracking 1s signals.

stage was the sampling rate.

4.2 Simulation and Processing Examples

GNSSTools can be used to simulated and process L1 C/A and L5 signals. This section demonstrates this ability by presenting various examples. First real L1 C/A and L5Q data are processed. The three cases are L1 C/A signals from a SeNSe lab receiver, simulated L1 C/A signals from Oriolia's Skydel GNSS Signal Simulator, and L5 signals from a SeNSe lab receiver. The next set of examples are simulated using *GNSSTools*. They are L1 C/A signals, L5 signals, and a single L1 C/A signal from a LEO satellite. For all signals simulated using *GNSSTools*, the h-parameters for a Rakon IT5300B TCXO oscillator were used. During acquisition, any PRN that had a P_d of ≥ 0.9 based off its correlation peak value and threshold value, was considered acquired and are shown in the tables below. For each data set, one PRN was chosen arbitrarily to perform carrier tracking. Acquisition and tracking parameters, such as the acquisition coherent and noncoherent integration times, were chosen based off trial and error, with an aim to maximize the number of signals detected and processed. Acquisition and tracking results were visually inspected to fine tune each of these parameters.

4.2.1 Results from Processing Data Samples

The first data set was collected on 2020-12-11 21:17:45 UTC in Boulder, Colorado. The receiver center frequency was centered on the GPS L1 band and the bandwidth was large enough for at least the L1 C/A code. The sampling rate was 25MHz. Two seconds of the data set was processed. A search for detectable signals was performed by attempting acquisition of PRNS 1-32 with a coherent integration time of 2ms and noncoherently integrated 10 times. Table 4.1 shows the acquisition results for all acquired PRN signals. PRN 9 was arbitrarily chosen to undergo tracking. Figure 4.7 shows the tracking results. Values for additional parameters are shown in Table 4.2.

PRN	$\hat{n}_0$ (chips/1023)	$\hat{f}_d$ (Hz)	SNR (dB)
3	798.6	-2000	23.8
4	122.7	1000	22.1
6	588.5	-1000	15.7
9	952.1	3000	18.4
16	339.2	1000	23.8
22	856.0	-3000	21.1
26	255.8	-2000	19.9
27	244.7	3000	16.4
31	267.0	-3000	18.3

Table 4.1: Shows the acquisition results for a sample of L1 C/A data.

Parameter	Value	Parameter	Value
Acq. Integration Time	2ms	Acq. # of Noncoherent Integrations	10
Fine Acq. Integration Time	10ms	Fine Acq. Method	FFT
Tracking Integration Time	1ms	q_a	$1\,(m/s)^2$
h_0	1×10^{-21}	h_2	2×10^{-20}
DLL Bandwidth	1Hz		

Table 4.2: Shows the parameters used for processing the L1 C/A sample data set.

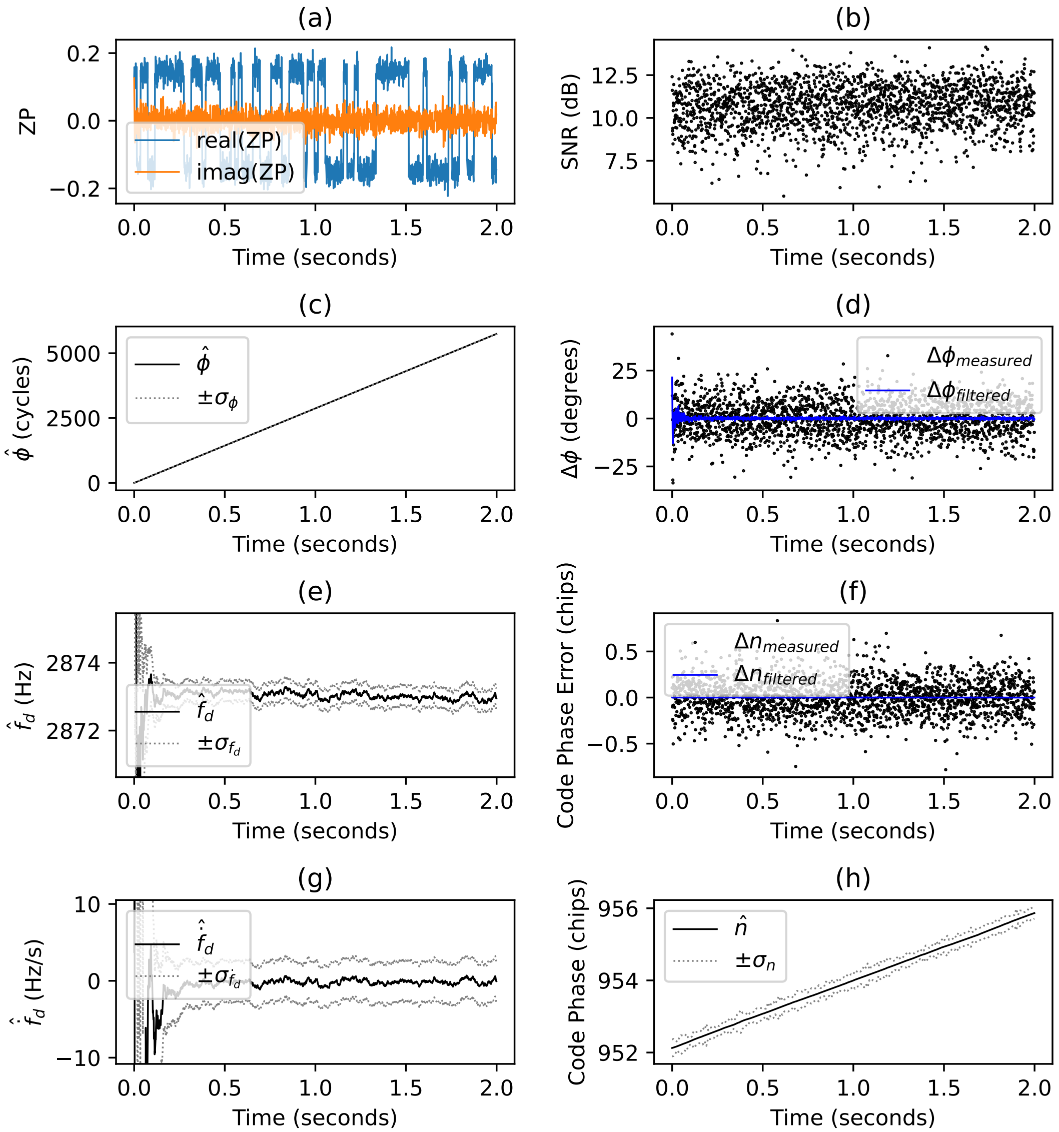


Figure 4.7: Shows the tracking results for PRN 9 in the L1 C/A data sample.

The second data set was simulated using Orolia's *Skydel GNSS Signal Simulation Software*, not *GNSSTools*. The receiver center frequency was centered on 1.582GHz. Relative to the GPS L1 center frequency of 1.57542GHz, the signal contained an IF of -6.58MHz The sampling rate was 60MHz. Two seconds of the data set was processed. A search for detectable signals was performed by attempting acquisition of PRNS 1-32 with a coherent integration time of 2ms and noncoherently integrated 20 times. Table 4.3 shows the acquisition results for all acquired PRN signals. PRN 19 was arbitrarily chosen to undergo tracking. Figure 4.8 shows the tracking results. Values for additional parameters are shown in Table 4.4.

PRN	$\hat{n}_0$ (chips/1023)	$\hat{f}_d$ (Hz)	SNR (dB)
2	487.5	2500	17.0
3	1022.3	-2500	16.1
6	824.6	1000	21.9
12	374.8	3000	18.4
17	361.4	-1000	24.0
19	909.2	0	24.3
24	656.1	-2000	15.2
28	683.3	-3000	20.2

Table 4.3: Shows the acquisition results for a sample of L1 C/A data, simulated using Skydel.

Parameter	Value	Parameter	Value
Acq. Integration Time	2ms	Acq. # of Noncoherent Integrations	20
Fine Acq. Integration Time	10ms	Fine Acq. Method	FFT
Tracking Integration Time	1ms	q_a	$1\,(m/s)^2$
h_0	1×10^{-21}	h_2	2×10^{-20}
DLL Bandwidth	1Hz		

Table 4.4: Shows the parameters used for processing the L1 C/A Skydel sample data set.

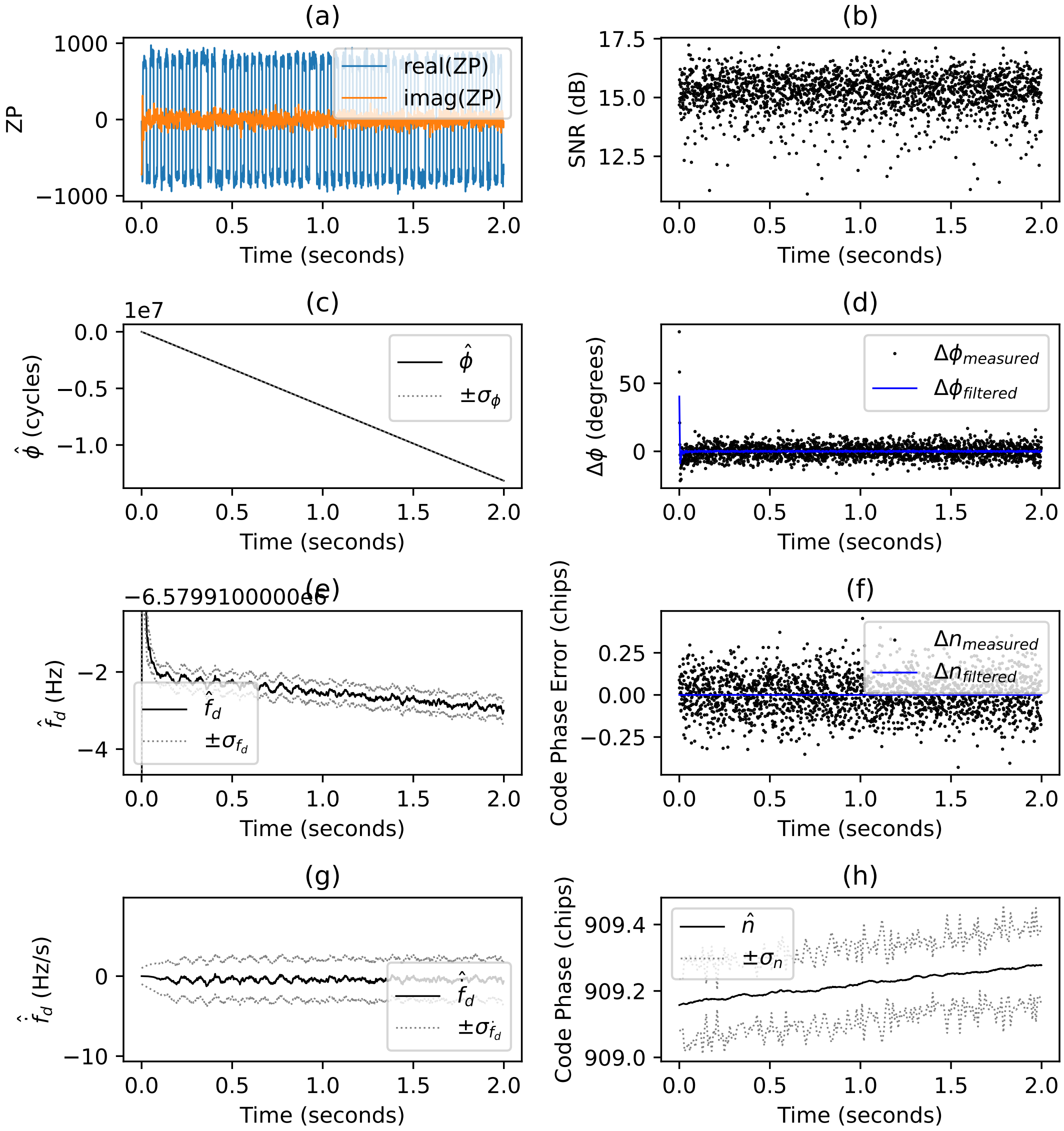


Figure 4.8: Shows the tracking results for PRN 19 in the L1 C/A Skydel data sample.

The third data set was collected on 2020-12-11 21:13:00 UTC in Boulder, Colorado. The receiver center frequency was centered on the GPS L5 band and the bandwidth was large enough for the L5 code. The sampling rate was 25MHz. Two seconds of the data set was processed. A search for detectable signals was performed by attempting acquisition of PRNS 1-32 with a coherent integration time of 20ms and was not noncoherently integrated. Table 4.5 shows the acquisition results for all acquired PRN signals. PRN 3 was arbitrarily chosen to undergo tracking. Figure 4.9 shows the tracking results. Values for additional parameters are shown in Table 4.6.

PRN	$\hat{n}_0$ (chips/20)	$\hat{f}_d$ (Hz)	SNR (dB)
3	6.43	-1450	28.7
4	8.18	950	28.4
6	13.03	-650	19.9
9	0.69	2150	22.9
26	5.84	-1200	28.2
27	12.90	2350	21.1

Table 4.5: Shows the acquisition results for a sample of L5 data. Only the Q channel was processed.

Parameter	Value	Parameter	Value
Acq. Integration Time	20ms	Acq. # of Noncoherent Integrations	1
Fine Acq. Integration Time	100ms	Fine Acq. Method	FFT
Tracking Integration Time	20ms	q_a	$1\,(m/s)^2$
h_0	1×10^{-21}	h_2	2×10^{-20}
DLL Bandwidth	1Hz		

Table 4.6: Shows the parameters used for processing the L5 sample data set.

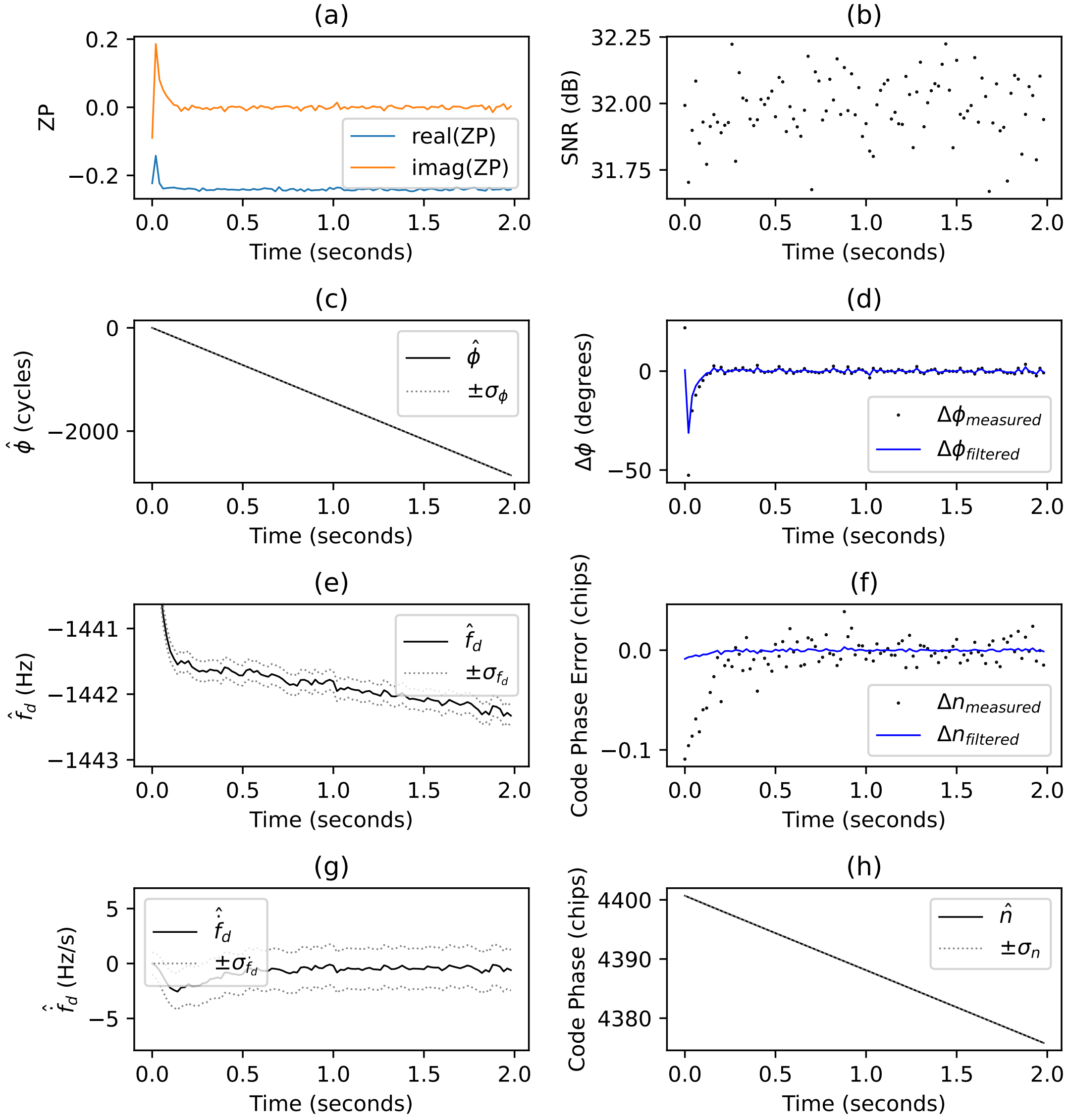


Figure 4.9: Shows the tracking results for PRN 3 in the L5 data sample. Only the Q channel is processed.

4.2.2 Results from Processing Simulated Signals

The first simulated data set is of an L1 C/A signal. The receiver center frequency was centered on the GPS L1 band and the bandwidth was set to be 20 times the L1 C/A chipping rate. The sampling rate was 25MHz. Two seconds of the data set was processed. The signal was simulated with 9 random PRN codes. For each code, the initial carrier and code phase was chosen from a uniform distribution bounded by 0 to 2π and 1 to 1023, respectively. The C/N_0 was also chosen randomly between 35dB·Hz and 45dB·Hz. Additionally, the Doppler and Doppler rate were sampled from the GPS Doppler distribution, shown in Figure 1.3, for each PRN. The h-parameters used to generate the oscillator phase noise were those associated with the Rakon IT55300B TCXO oscillator, shown in Figure 2.8. These h-parameters are used for the remainder of the chapter.

PRN	$\hat{n}_0$ (chips/1023)	$\hat{f}_d$ (Hz)	SNR (dB)
9	259.2	-2500	18.5
15	479.2	3750	19.4
18	315.8	3500	21.8
20	375.6	1250	16.8
22	382.5	-500	19.7
23	118.0	-250	19.6
25	139.1	2000	15.3
26	194.4	2000	15.2
30	227.1	-500	22.5

Table 4.7: Shows the acquisition results for a sample of L1 C/A data, simulated using *GNSSTools*.

A search for detectable signals was performed by attempting acquisition of PRNS 1-32 with a coherent integration time of 4ms and was noncoherently integrated 20 times. Table 4.7 shows the acquisition results for all acquired PRN signals. PRN 22 was arbitrarily chosen to undergo tracking. Figure 4.10 shows the tracking results. Values for additional parameters are shown in Table 4.8.

Parameter	Value	Parameter	Value
Acq. Integration Time	4ms	Acq. # of Noncoherent Integrations	20
Fine Acq. Integration Time	10ms	Fine Acq. Method	FFT
Tracking Integration Time	1ms	q_a	$1\,(m/s)^2$
h_0	1×10^{-21}	h_2	2×10^{-20}
DLL Bandwidth	1Hz		

Table 4.8: Shows the parameters used for processing the L1 C/A simulated data set.

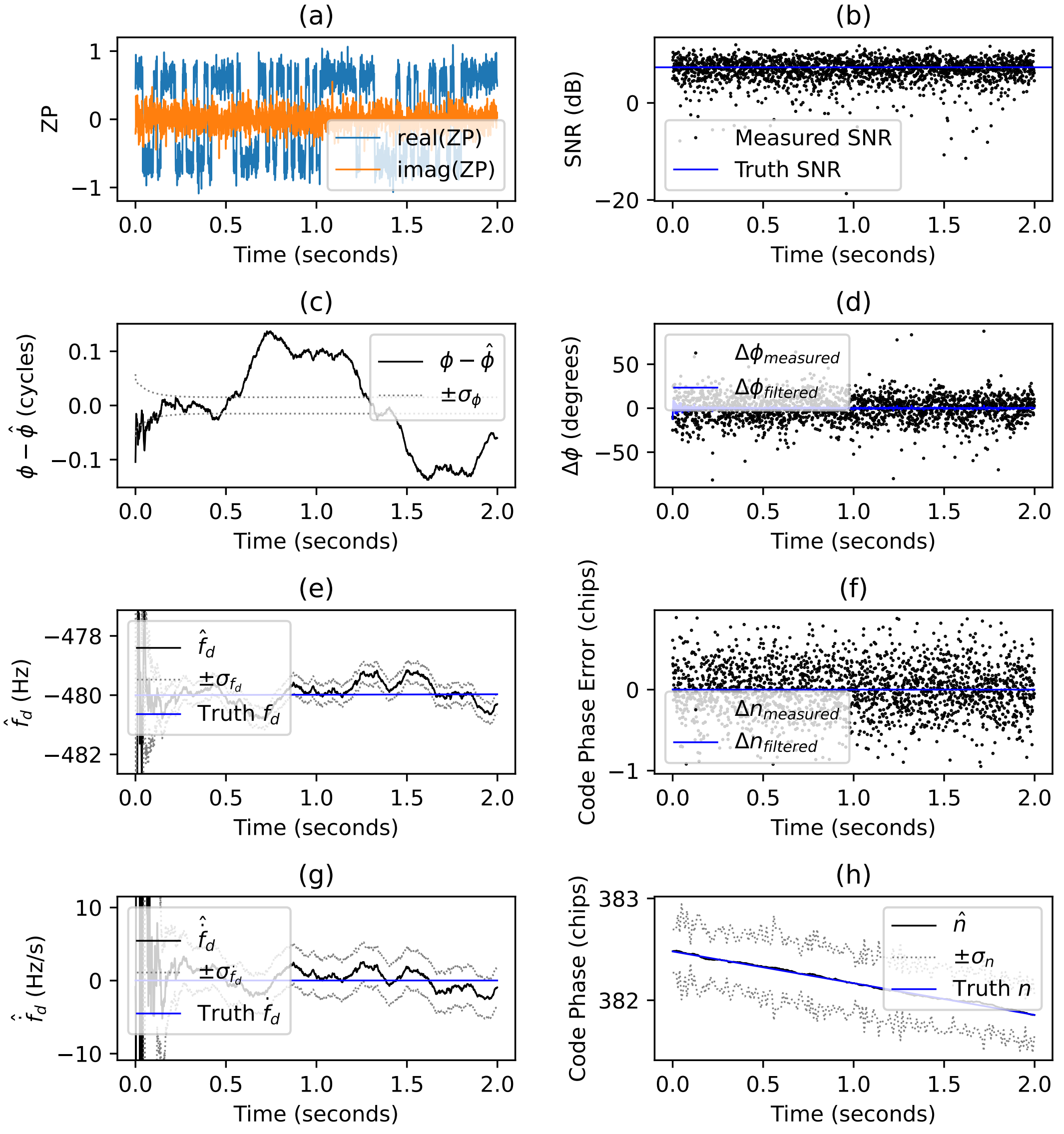


Figure 4.10: Shows the tracking results for PRN 22 in the L1 C/A simulated data sample.

The second simulated data set is of an L5 signal. The receiver center frequency was centered on the GPS L5 band and the bandwidth was set to be twice times the L5 chipping rate. The sampling rate as 25MHz. Two seconds of the data set was processed. The simulation process was similar to that of the L1 C/A code simulation. However, the initial code phase was randomly chose from a uniform distribution bounded between 1 and 10230. All other simulation steps are the same.

PRN	$\hat{n}_0$ (chips/20)	$\hat{f}_d$ (Hz)	SNR (dB)
2	16.22	3750	17.1
3	10.84	800	20.9
12	16.64	-1500	19.7
17	13.51	1950	15.8
20	4.09	-3000	18.1
25	2.55	-2450	18.8
26	11.34	-750	15.8
29	15.23	-2500	20.4
32	7.83	-3100	18.5

Table 4.9: Shows the acquisition results for a sample of L5 data, simulated using *GNSSTools*. Only the Q channel was processed.

A search for detectable signals was performed by attempting acquisition of PRNS 1-32 with a coherent integration time of 20ms and was not noncoherently integrated. Table 4.9 shows the acquisition results for all acquired PRN signals. PRN 2 was arbitrarily chosen to undergo tracking. Figure 4.11 shows the tracking results. Values for additional parameters are shown in Table 4.10.

Parameter	Value	Parameter	Value
Acq. Integration Time	20ms	Acq. # of Noncoherent Integrations	1
Fine Acq. Integration Time	100ms	Fine Acq. Method	FFT
Tracking Integration Time	20ms	q_a	$1\,(m/s)^2$
h_0	1×10^{-21}	h_2	2×10^{-20}
DLL Bandwidth	1Hz		

Table 4.10: Shows the parameters used for processing the L5 simulated data set. Only the Q channel was processed.

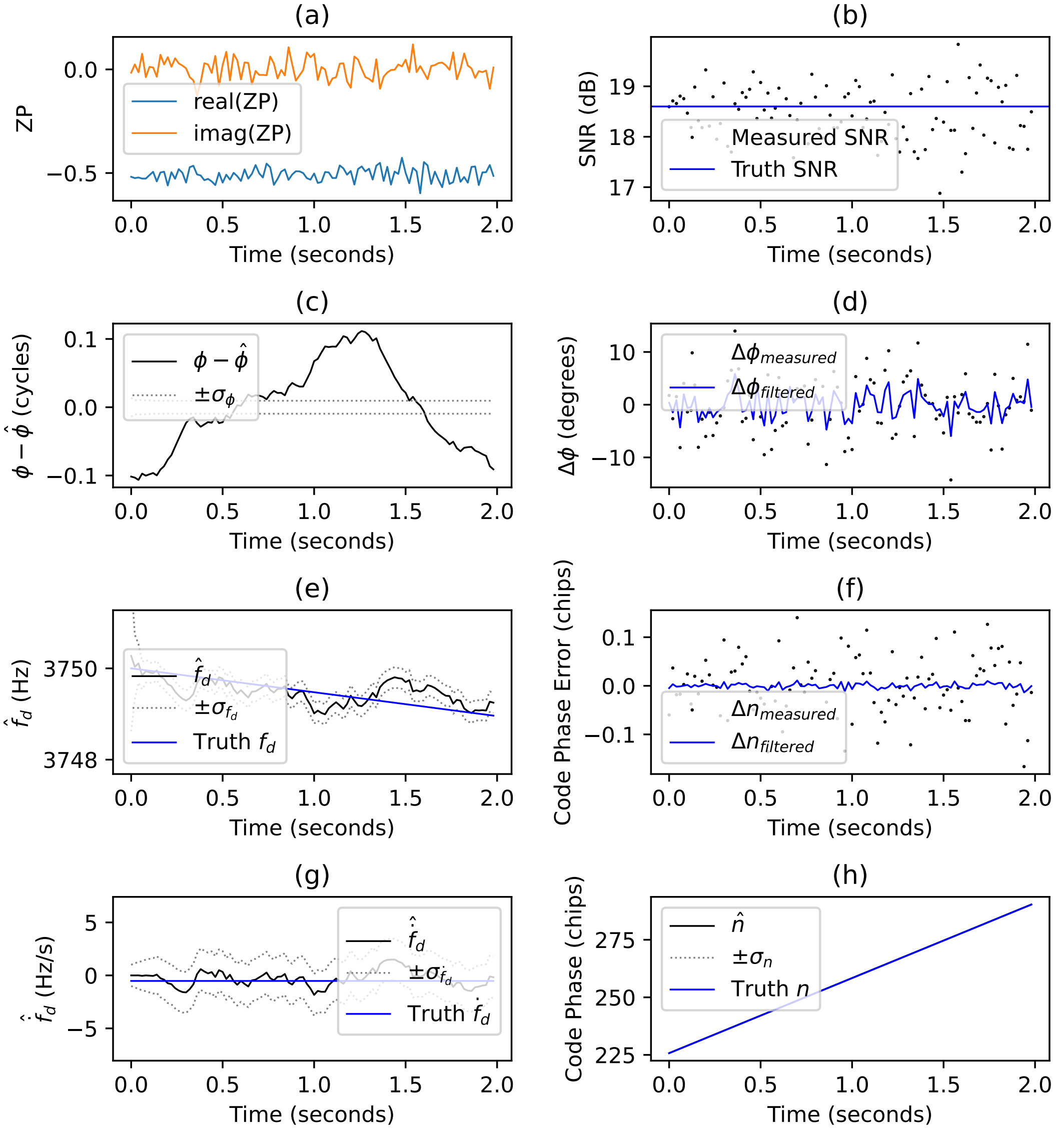


Figure 4.11: Shows the tracking results for PRN 2 in the L5 simulated data sample. Only the Q channel is processed.

The third simulated data set is of a GPS L1 C/A signal from a LEO satellite. An Iridium constellation was defined and the satellite that passed over the receiver with the highest elevation during the time frame of the simulation (10min) was chosen for simulating and processing the signal from it. Only this signal was simulated with no other PRNs. The sampling rate was 5MHz with an intermediate frequency of 1.25MHz. The C/N_0 was 45dB·Hz throughout the whole pass. While it would not be in reality, the simulation still shows the PLLs ability to track this large change in Doppler over a long period of time.

The coarse acquisition process was aided to find the correct Doppler frequency quicker. However, it was not centered to the exact Doppler, but to a Doppler value that was an integer multiple of $1/T$, where T is the coherent integration time used for acquisition. After acquisition, standard fine acquisition and tracking methods, used in previous examples, were used to process the signal. Table 4.11 shows the acquisition and tracking parameters used to track this signal while Figure 4.12 shows the tracking results.

Parameter	Value	Parameter	Value
Acq. Integration Time	1ms	Acq. # of Noncoherent Integrations	1
Fine Acq. Integration Time	10ms	Fine Acq. Method	FFT
Tracking Integration Time	1ms	q_a	$1\,(m/s)^2$
h_0	1×10^{-21}	h_2	2×10^{-20}
DLL Bandwidth	1Hz		

Table 4.11: Shows the parameters used for processing the L1 C/A simulated data as if it was transmitted from an iridium like satellite.

Figures 4.12b, d, and f show spikes in SNR, $\Delta\phi$, and code phase error, respectively. By looking at Figure 4.12h, it is realized that the spikes in the former three subplots line up with the center of the 1ms L1 C/A ranging code. Due to the varying Doppler and Doppler rate, the ranging code is either less than or greater than 1ms, if the Doppler frequency is postive or negative, respectively. In either situation, the initial code phase at the start of each 1ms integration drifts. It advances or recesses until it wraps back to the beginning of the code, shown in Figure 4.12h. Since data bits were simulated in this data set, data bit transitions can negatively affect the correlation power. This is most noticeable when the initial code phase is half the number of chips in the ranging code (≈ 511). Here, a data bit will cause the correlation power to be nearly zero. This is evident in Figure 4.12b, where the measured SNR is around -25dB, when these data bit flips occur. Modifying the current *GNSSTools* implementation in the future can resolve this issue by keeping most of each

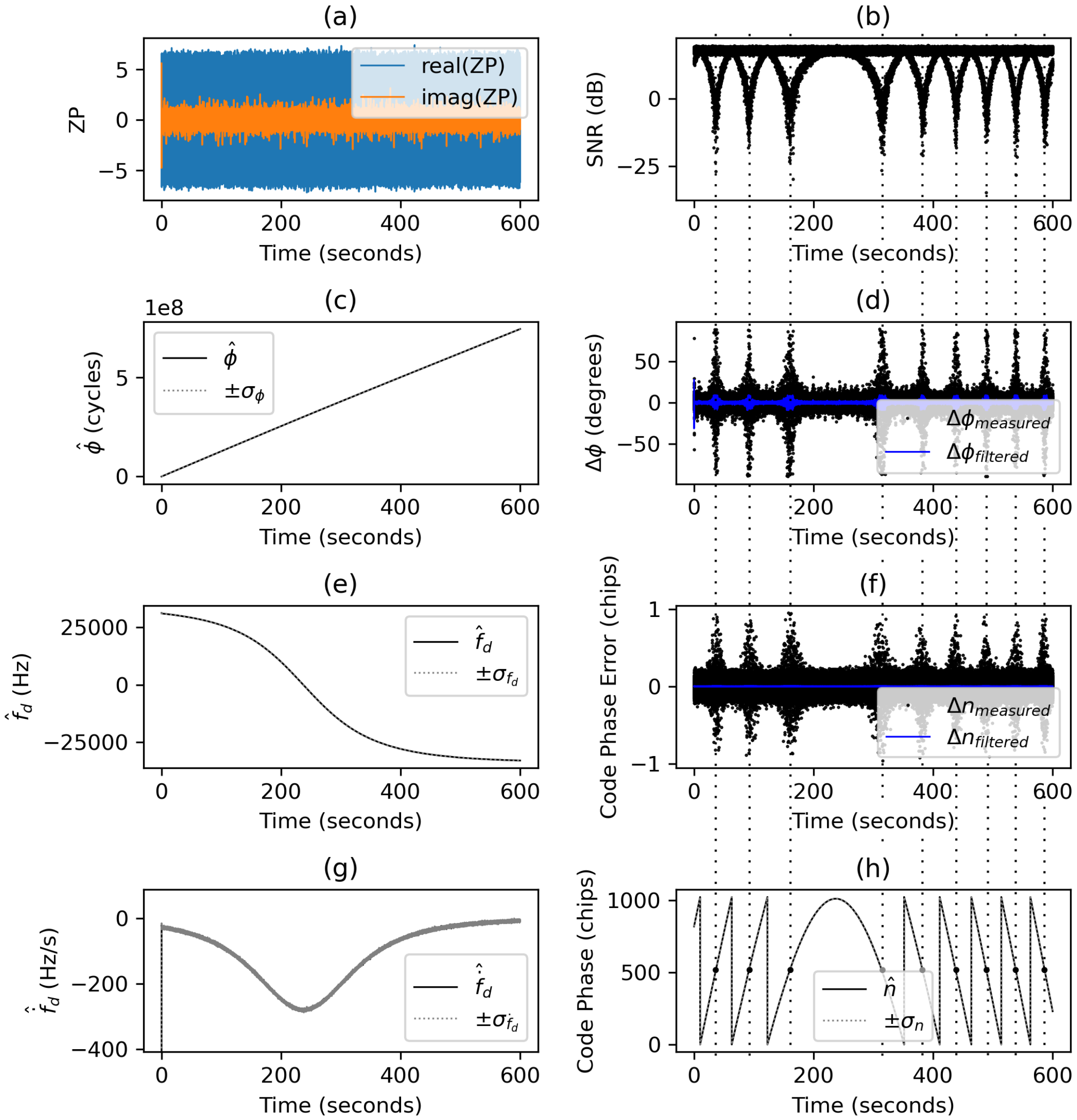


Figure 4.12: Shows the tracking results for PRN 26 in the L1 C/A simulated data sample. This simulated signal is simulated as if it is from a satellite in an Iridium-like constellation over a 10 minute time span. The vertical, dotted lines on the right half of the figure point to the large errors caused by navigation data bit flips.

successive ranging code in the integration window rather than blindly integrating every T.

4.3 GPS and LEO Constellation Comparisons: Acquisition

Signal acquisition becomes a more time consuming step for signals from LEO satellites than those from MEO, due to the high Doppler range to search. As shown in Table 4.12, the number of Doppler bins to search increases by a factor of ~ 8 for all integration times. This will cause the acquisition process to take that many times longer than what is showed in Figure 4.5.

Constellation	Max f_d (kHz)	T=1ms	2ms	5ms	10ms	20ms	20ms	100ms
GPS	4.0	9	17	41	81	161	401	799
Iridium	34.8	71	141	349	697	1393	3479	6955
Starlink	35.2	73	143	353	705	1409	3521	7041
OneWeb	32.0	65	129	321	641	1281	3201	6399

Table 4.12: Shows the number of Doppler bins to search for when acquiring signals from each constellation for various integration times.

The second issue for acquiring LEO signals is the high Doppler rate that can be encountered

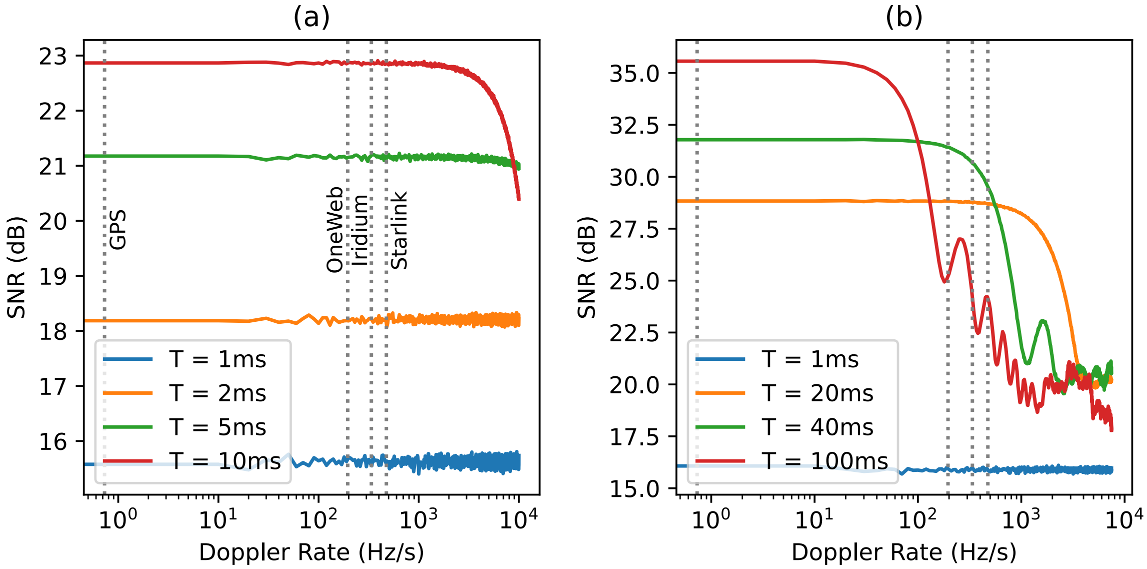


Figure 4.13: Shows the affect on the acquired peak SNR due to high Doppler rates for the L1 C/A code (a) and L5Q code (b). The C/N_0 of the signals simulated are 45dB·Hz.

when a satellite is near zenith. Based off Figure 1.3, the maximum Doppler rates that can be expected from the GPS, Iridium, Starlink, OneWeb constellations are -0.730Hz/s, -336.4Hz/s, -475.0Hz/s, and -195.5Hz/s, respectively. Unless the receiver is assisted, by having access to the approximate time and recent orbital elements for a given LEO constellation, the initial Doppler rate of any given incoming signal is not known. Therefore, the local replica would have an $\dot{f}_d$ of zero. If the Doppler rate is very high, as in the case of the signals from LEO satellites, then it may negatively affect the power of the correlation peak. This is shown in Figure 4.13, where the receiver is assumed to be stationary. For low integration times such as 1ms, 2ms, and 5ms, there is a negligible affect on the peak correlation power. For integration times 10ms and 20ms, a significant affect can be seen past $\sim$ 1000Hz/s. However, this is much greater than the maximum Doppler rate from the constellations being compared, so it would have no affect on acquisition. 40ms integrations experience around a 2dB attenuation due to the maximum Starlink Doppler rate and a negligible attenuation due to the maximum Iridium Doppler rate. However, 100ms integrations experience a 10dB attenuation at the maximum Iridium Doppler rate and an additional 2.5dB attenuation at the maximum Starlink Doppler rate.

4.4 GPS and LEO Constellation Comparisons: Tracking

Large Doppler rates also affect signal tracking performance. During acquisition, the Doppler rate is not known. If the acquisition stage does not estimate a value for the initial Doppler rate, then it can be initialized to zero, if a 3-state KF-based PLL is used. For a 2-state KF-based PLL defined in (Yang, Ling, et al., 2017), only the carrier phase and Doppler frequency are tracked. The only dynamics modelled are white noise, a function of the integration time and C/N_0, and colored noise from the receiver oscillator, defined by the oscillator h-parameters. Since this 2-state PLL is optimize to perform under near zero dynamic conditions, it fails to track signals with high $\dot{f}_d$, as shown in Figure 4.14.

Here, four cases were tested. Each case has either a high or low f_d and $\dot{f}_d$, shown in Table 4.13. In Figures 4.14-4.16, Cases 1 and 4 always succeed. In cases 2 and 3 a signal is simulated

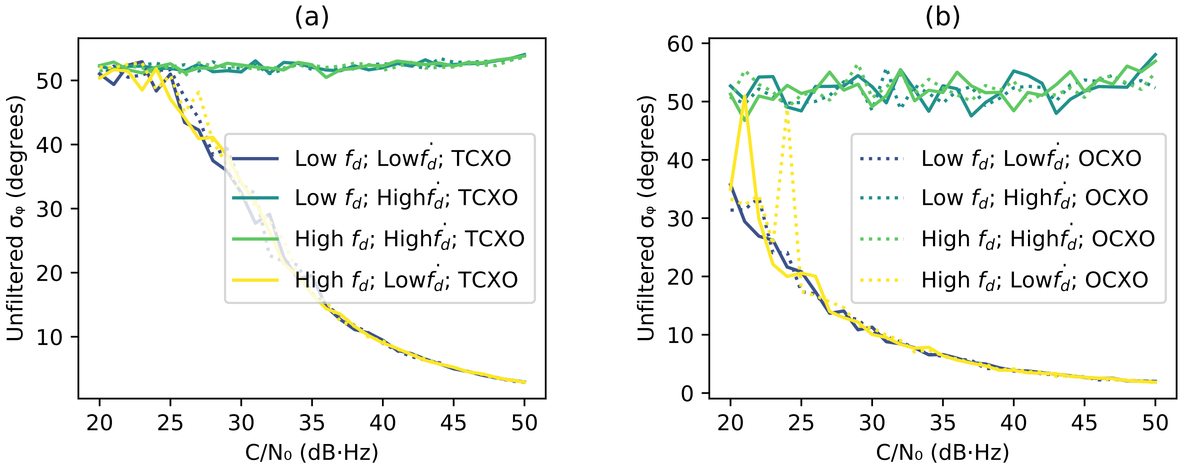


Figure 4.14: Shows the 2-state PLL tracking performance for different Doppler and Doppler rate combinations using integration times of 1ms (a) and 10ms (b).

with a high $\dot{f}_d$, which a 2-state PLL fails to track in Figure 4.14.

In Figure 4.15, a 3-state KF-based PLL is used where the initial $\dot{f}_d$ is set to zero, assuming that the acquisition stage did not provide an initial estimate of the Doppler rate. The high Doppler rate is similar to large dynamics in the signal and the 3-state PLL would fail to track it if q_a is set low. Therefore, q_a is set very high, 1000 (m^2/s^6)/Hz to ensure the filter bandwidth is high. In doing so, the signal with high Doppler rate can be tracked, but only to around 32dB·Hz.

In Figure 4.16, a noisy initial estimate of $\dot{f}_d$ is used to initialize the PLL. This noisy estimate has a standard deviation of $\sigma_{f_d} = 1$Hz. Because $\dot{f}_d$ is not initialized to zero, q_a is set to 0.1 (m^2/s^6)/Hz. It can be seen that the filter is able to successfully track signals in cases 2 and 3 just as well as in cases 1 and 4 for an integration time of 1ms. For $T = 10$ms, there is lower performance

	f_d (kHz)	$\dot{f}_d$ (Hz/s)
Case 1	4	-0.730
Case 2	4	-475.0
Case 3	35.2	-475.0
Case 4	35.2	-0.730

Table 4.13: Shows the cases of Doppler and Doppler rates to test 2- and 3-state KF-based PLL tracking loops on.

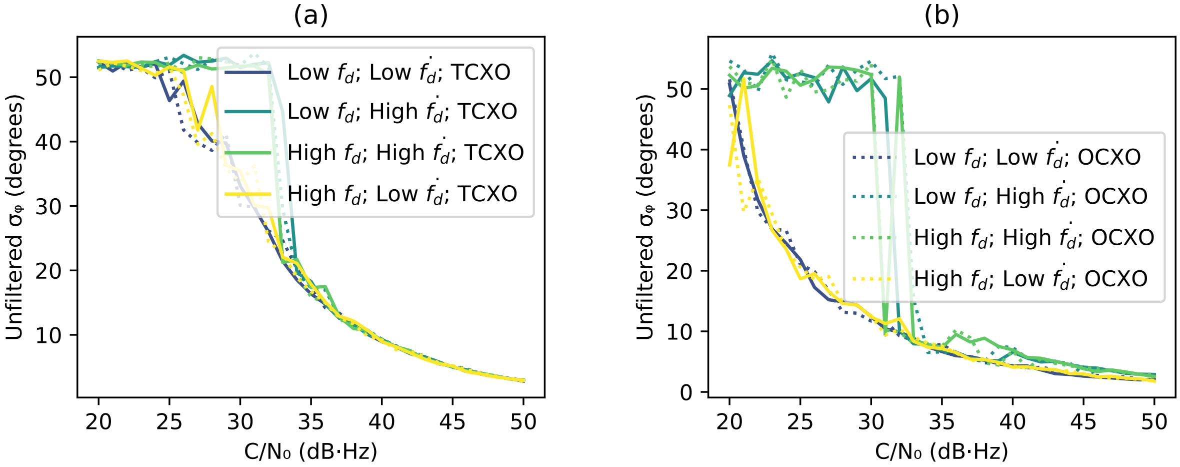


Figure 4.15: Shows the 3-state PLL tracking performance for different Doppler and Doppler rate combinations using integration times of 1ms (a) and 10ms (b). q_a is set to be very large (1000 $(m^2/s^6)/Hz$) and the initial $\dot{f}_d$ is set to zero.

at low C/N_0. However, this may also be due to the low number of iterations performed for these figures and that there were less $\Delta\phi$ samples per run for $T = 10$ms than there were for $T = 1$ms. Based off the these results, it is best to provide an initial Doppler rate estimate to the three-state PLL, allowing a realistic q_a to be set. As shown in Figure 1.3, the range of Doppler and Doppler

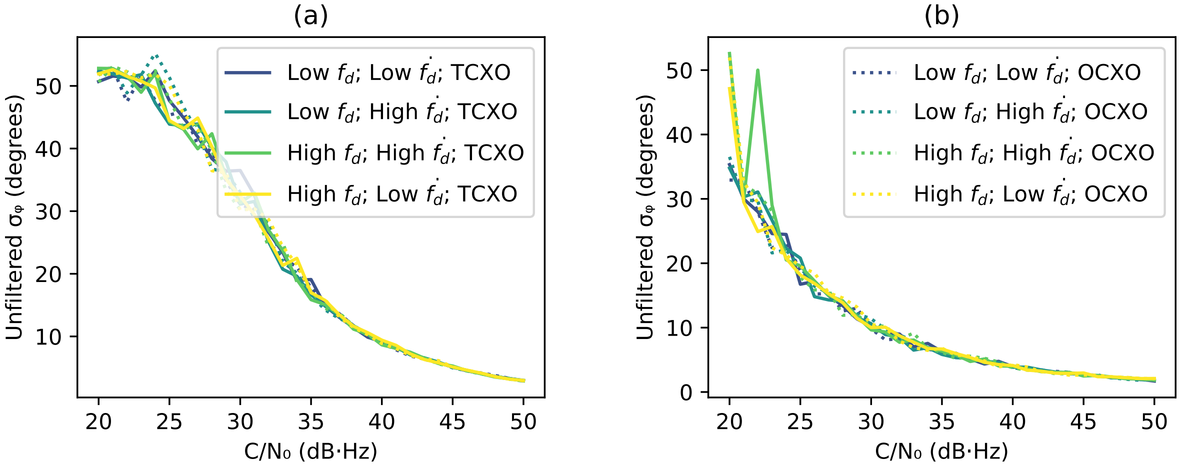


Figure 4.16: Shows the 3-state PLL tracking performance for different Doppler and Doppler rate combinations using integration times of 1ms (a) and 10ms (b). q_a is set to 0.1 $(m^2/s^6)/Hz$ and the initial $\dot{f}_d$ is set to the value of the $\dot{f}_d$ used for simulating the signal in each iteration.

rate for LEO constellations is large. However, the following section will discuss a way to reduce this range.

4.5 Narrowing the Doppler Search Area and Approximating the Doppler Rate

For GNSS receivers that are assisted, as is the case for cell phones, the satellite orbital elements, time, and approximate position of the receiver are known. With this information, the acquisition process can be much faster. In addition, the PLL can be initialized with the Doppler rate calculated using the published orbital elements. In a cold start, the orbital elements, time, and receiver position are not known. Since the Doppler search area is so large, it would be beneficial to narrow the search area by only considering high elevation passes of satellites that are at low elevation at the time of acquisition. In doing so, both the Doppler and Doppler rate can be approximated

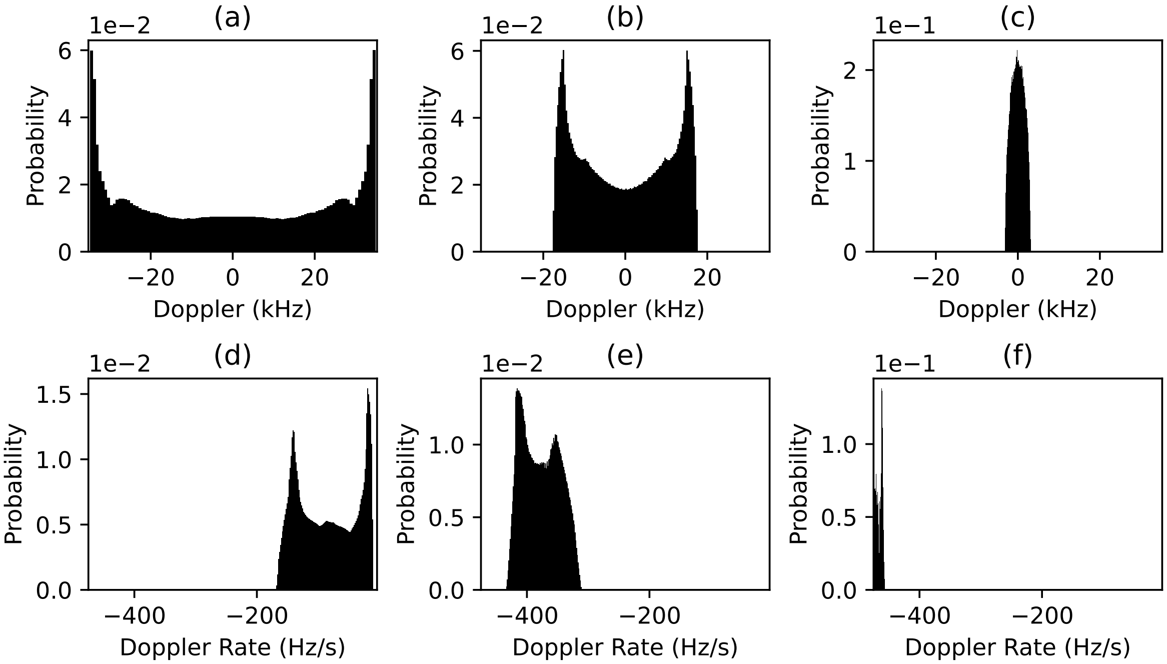


Figure 4.17: Shows the Doppler frequency (top) and Doppler rate (bottom) distributions for Starlink satellites with elevations $10° - 15°$ (left), $60° - 65°$ (middle), and $85° - 90°$ (right). Results are for satellite passes with a maximum elevation of at least 10°.

for use in the acquisition and tracking stages.

Figures 4.14 and 4.15 depict cold start scenarios where the Doppler search was completely blind and the PLL was not initialized with any estimate of the Doppler rate. These situations do not consider the Doppler and Doppler rate distributions of the satellite constellation that the signal is transmitted from. For signals transmitted from satellites in either Iridium, Starlink, and OneWeb constellations, there are vast Doppler ranges to search and significant Doppler rates, that should not be assumed to be zero during initialization of the PLL. The Doppler and Doppler rate ranges can be narrowed by placing constraints on the elevations of satellites to process. It may not be advantageous to begin tracking signals from satellites at zenith. Instead, it may be better to begin tracking signals from satellites that are rising, with low elevation. This will yield more time to decode the navigation message. However, the expected Doppler range for signals from satellites with elevations between 10° and 15° still spans the whole Doppler range when considering satellites

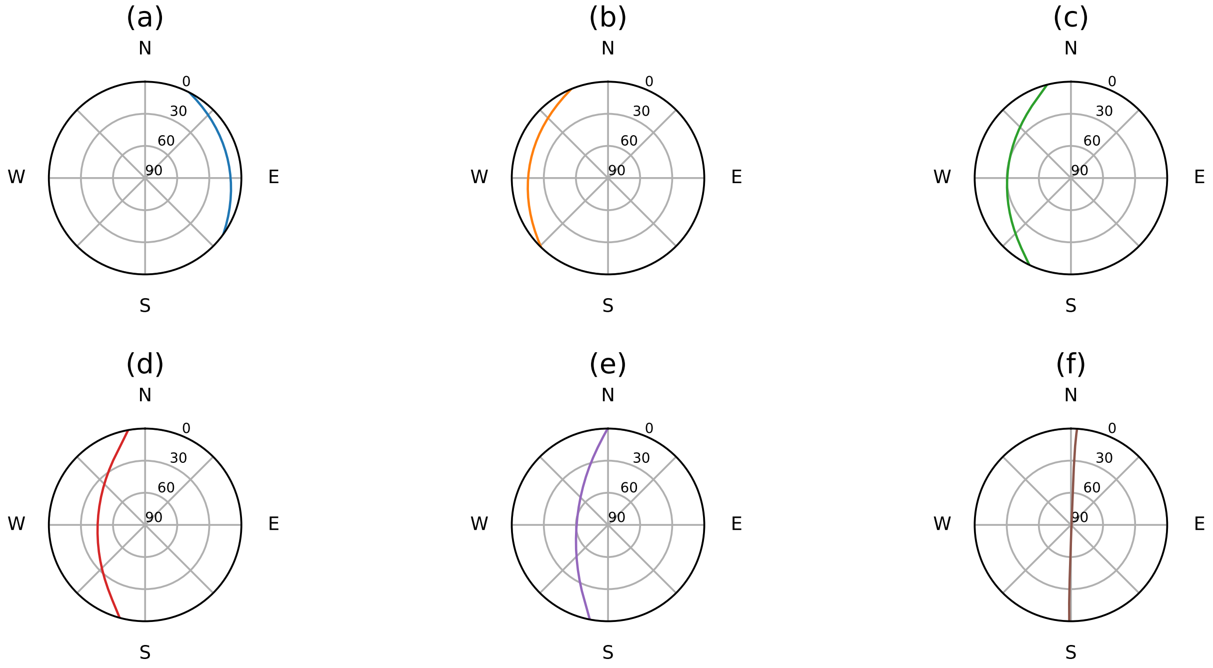


Figure 4.18: Shows sky plots of individually, propagated Iridium satellite passes. The maximum elevation of these passes are 10° (a), 15° (b), 30° (c), 45° (d), 60° (e), and 90° (f). The Doppler and Doppler rates for a given point on each of the passes are shown via the respective contour lines in Figures 4.19(b-d).

in elevations between 10° and 90°, shown in Figure 4.17. Figure 4.17 shows that the smallest range in Doppler and Doppler rate that can be achieved by processing signals from satellites with elevations between 85° and 90°, shown in Figures 4.17c and 4.17f, respectively.

The large Doppler and Doppler rate range for signals from low elevation satellites, in Figures 4.17a and 4.17d, are caused by satellite passes that peak at low elevations. In the case of Figure 4.17, it is between 10° and 15°. Examples of high and low elevation passes are shown in Figure 4.18. Since the peak elevation of a satellite pass plays a role in the observed Doppler and Doppler rates, it can be specified in order to reduce the Doppler search range and provide an initial Doppler rate estimate. Figure 4.20 shows the joint probability distribution of the Doppler and Doppler rate for the GPS,

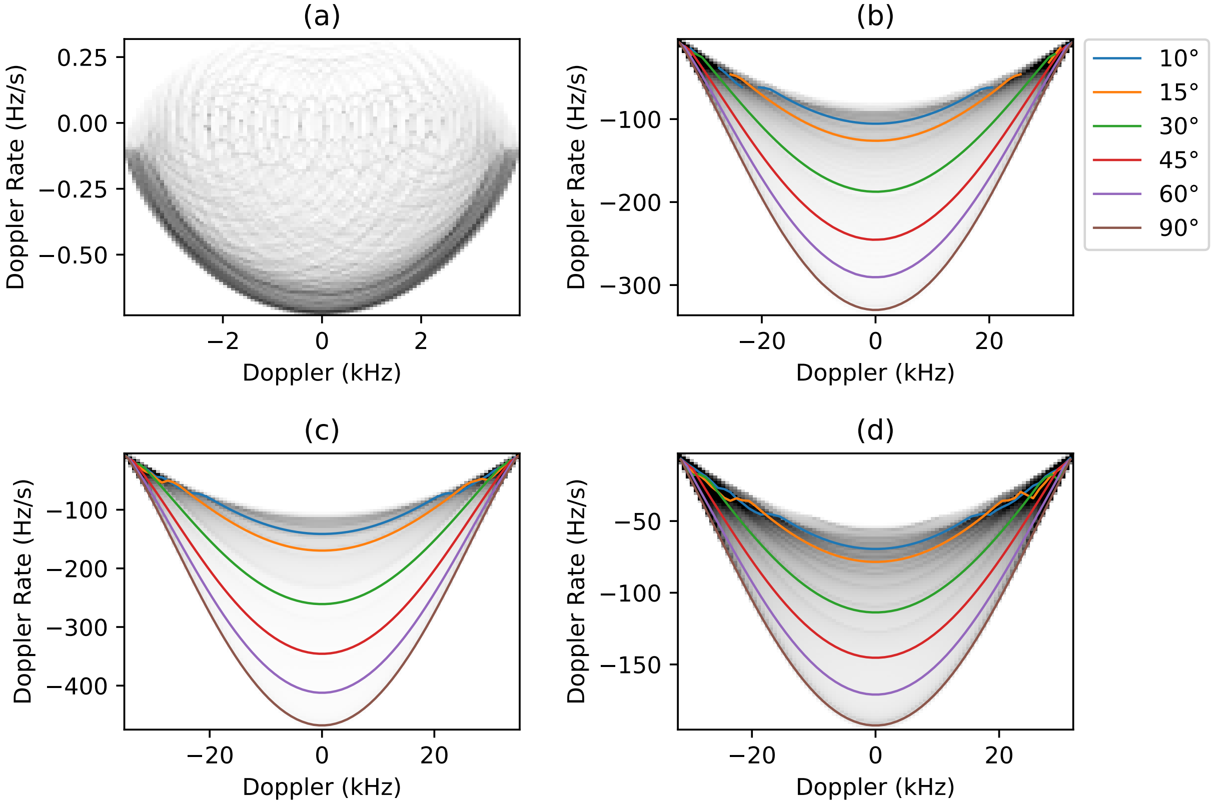


Figure 4.19: Shows the Doppler and Doppler rate joint probability distributions for the GPS (a), Iridium (b), Starlink (c), and OneWeb (d) constellations. The contour lines on (b)-(d) show the Doppler and Doppler rates associated with a satellite pass with a maximum elevation equal to their respective values in the legend.

Iridium, Starlink, and OneWeb constellations. The contour lines for the Iridium, Starlink, and OneWeb distributions are the Doppler and Doppler rates observed during their respective satellite passes in Figure 4.18. It can be seen that the peak elevation of each satellite pass results in unique pairs of Doppler and Doppler rates for either half of the Doppler axis on the joint PDF. Since rising satellites are of interest, only positive Doppler frequencies would be used for acquisition. The only region of the PDF that is not unique for any peak elevation are the extreme negative and positive Dopplers. Satellites, irregardless of their path across the sky, will be observed with these extreme Dopplers at the beginning and ending of their pass. Figure 4.20 shows two points on this topic. The first is that considering satellite passes with peak elevations of at least 30°, and filtering out satellites with elevations between 10° and 15°, results in two, relatively narrow, Doppler peaks in Figure 4.20a. The range of Doppler rates is also diminished, shown in 4.20d. Using the positive Doppler half of Figure 4.20a and Figure 4.20d, an incoming signal from a rising

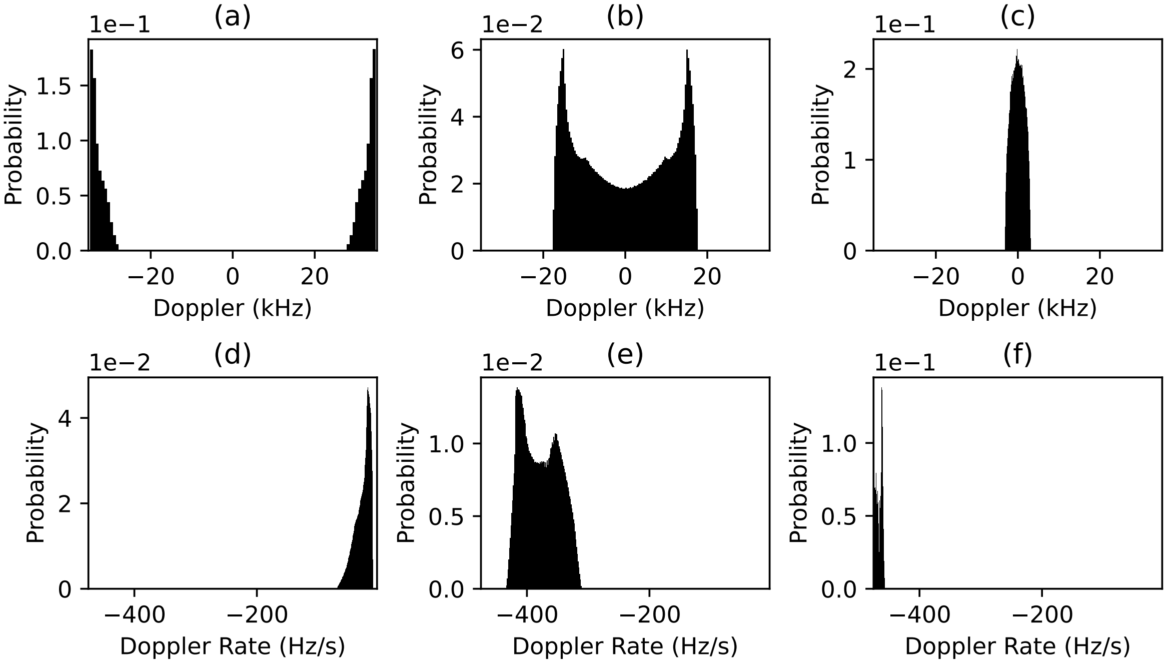


Figure 4.20: Shows the Doppler frequency (top) and Doppler rate (bottom) distributions for Starlink satellites with elevations $10° - 15°$ (left), $60° - 65°$ (middle), and $85° - 90°$ (right). Results are for satellite passes with a maximum elevation of at least 30°.

LEO satellite can be acquired much more quickly. A three state PLL can be initialized with the code phase, carrier phase, and Doppler estimates from the coarse and fine acquisition stages and with the Doppler rate from the PDFs above. This semi-cold start process only needs to occur for one satellite, since the almanac and approximate time can be obtained from the navigation message, if any. If an approximate estimate of the receiver location exists, then the initial Doppler and Doppler rate estimates for additional signal acquisition and tracking can be determined from the orbital elements.

Chapter 5

Conclusion

This thesis discussed GNSS signal simulation, acquisition, and tracking using an in-house Julia-language-based package called *GNSSTools*. This module can be used to simulate signals with various ranging code layers, Doppler frequency (static, linear, and non-linear), and thermal and oscillator phase noise sources. It provides methods for acquiring and tracking signals in both real and simulated data sets. The tracking algorithm is a 2- and 3-state KF-based PLL. The state, and various other filter parameters, can be set by the user.

After brief overviews of signal simulation, acquisition, and tracking, the performance of the acquisition and tracking stages were evaluated and compared to the expected performances. The acquisition stage has similar performance to the expected performance, while the KF-based PLL has measured unfiltered and filtered carrier phase errors that differ from the expected values. The runtimes of the signal simulation and processing stages were also presented and discussed. After evaluation, a variety of examples were presented. These examples included processed real data from one of the SeNSe lab's receivers, as well as a data set created using Orolia's *Skydel GNSS Signal Simulator*. Simulated data sets using *GNSSTools*, were also presented to showcase the situations that can be generated. For all cases, all simulated signals were acquired successfully. One signal from each data set was arbitrarily chosen to undergo signal tracking.

The ability to acquire signals with high Doppler rates was discussed, with an evaluation of the effect on the peak correlation power due to high Doppler rates. The significantly large Doppler search area that would need to be searched for signals from LEO satellites was also noted. In addition, a demonstration of tracking signals with extreme cases of Doppler and Doppler rates showed that either an initial estimate of the Doppler rate be used to initialize the PLL or that q_a must be set high in order for a 3-state PLL to track the high Doppler rate. However, using a high q_a results in failed tracking around 32dB·Hz.

Finally, a way to minimize the Doppler search range and provide an initial Doppler rate estimate to the PLL was discussed. By enforcing two requirements, low peak elevation satellite passes and satellites that are at low elevation at the time of acquisition, the Doppler and Doppler rate distributions narrow significantly. Only positive Doppler frequencies need to be searched because rising satellites are worth more to process than setting satellites, since there is more time to decode the navigation message, if any, and obtain orbital information for processing signals from additional satellites.

There are limitations to *GNSSTools* that should be noted. This package does not include all error sources that exist in real signals. For example, it does not simulate ionospheric and tropospheric effects, multipath, and satellite clock errors. These are major sources of error in PNT. All results from this thesis are likely overestimates of the performance of each processing stage. Including these sources of error in the future would greatly improve the accuracy of these evaluations. It may also better inform future developers of the affect these sources have on signals from LEO satellites.

References

Airst, M. J. (2010). Case study on GPS network timing integrity & financial markets:impacts on infrastructure when GPS service is disrupted. In Space-based positioning navigation & timing: Tenth meeting.

Betz, J. W. (2015). Engineering satellite-based navigation and timing. John Wiley & Sons.

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. SIAM Review, 59(1), 65–98. doi: 10.1137/141000671

Bourne, H. W. (2016). An algorithm for accurate ionospheric total electron content and receiver bias estimation using gps measurements (mathesis, Colorado State University). Retrieved from http://hdl.handle.net/10217/173454

Brown, R. (2012). Introduction to random signals and applied Kalman filtering : with MATLAB exercises. Hoboken, NJ: J. Wiley & Sons.

Chagas, R. A. J., de Sousa, F. L., Louro, A. C., & dos Santos, W. G. (2018, oct). Modeling and design of a multidisciplinary simulator of the concept of operations for space mission pre-phase a studies. Concurrent Engineering, 27(1), 28–39. doi: 10.1177/1063293x18804006

Chiou, T. (2010). Design of a doppler-aided GPS navigation system for weak signals caused by strong ionospheric scintillation (mathesis). Standford University.

Curran, J. T., Lachapelle, G., & Murphy, C. C. (2012). Digital GNSS PLL design conditioned on thermal and oscillator phase noise. IEEE Transactions on Aerospace and Electronic Systems, 48(1), 180-196. doi: 10.1109/taes.2012.6129629

Ellis, P., Rheeden, D. V., & Dowla, F. (2020). Use of doppler and doppler rate for RF geolocation using a single LEO satellite. IEEE Access, 8, 12907–12920. doi: 10.1109/access.2020.2965931

European Space Agency. (2011). GLONASS Space Segment. Web. Retrieved from https://gssc.esa.int/navipedia/index.php/GLONASS_Space_Segment

Faragher, R., & Ziebart, M. (2020). OneWeb LEO PNT: progress or risky gamble? Web. Retrieved from https://insidegnss.com/oneweb-leo-pnt-progress-or-risky-gamble/

Flores, A. (2020a). IS-GPS-200: NAVSTAR GPS space segment navigation user segment interfaces [Computer software manual].

Flores, A. (2020b). IS-GPS-705: NAVSTAR GPS space segment user segment L5 interfaces [Computer software manual].

Global positioning system (GPS) standard positioning service (SPS) performance standard: 5th edition (Tech. Rep.). (2020). 6000 Defense Pentagon, Washington, DC 20301-6000: United States Air Force.

Gold, R. (1967, oct). Optimal binary sequences for spread spectrum multiplexing (corresp.). IEEE Transactions on Information Theory, 13(4), 619–621. doi: 10.1109/tit.1967.1054048

GPS and precision timing applications (Tech. Rep.). (1996). Palo Alto, CA: Hewlett Packard. (Application Note 1272)

Heck, J. (2017). Positioning in a multi-global navigation satellite systems (GNSS) world. In Noaa ngs webinar series.

Humphreys, T. E., Murrian, M., van Diggelen, F., Podshivalov, S., & Pesyna, K. M. (2016, apr). On the feasibility of cm-accurate positioning via a smartphone's antenna and GNSS chip. In 2016 IEEE/ION position, location and navigation symposium (PLANS). IEEE. doi: 10.1109/plans.2016.7479707

Inside GNSS. (2020). UK acquires OneWeb leo constellation, but won't work for satnav — or maybe it will. Web. Retrieved from `https://insidegnss.com/uk-acquires-oneweb-leo-constellation-but-wont-work-for-satnav/`

Iridium NEXT. (2021). Web. Retrieved from `https://earth.esa.int/web/eoportal/satellite-missions/i/iridium-next`

Jones, T. L. (2013). Fact Sheet – Wide Area Augmentation System (WAAS). Web. Retrieved from `https://www.faa.gov/news/fact_sheets/news_story.cfm?newsId=14974&omniRss=fact_sheetsAoc&cid=103_F_S`

Kassas, Z. M., Khalife, J., Abdallah, A., & Lee, C. (2020, oct). I am not afraid of the jammer: Navigating with signals of opportunity in GPS-denied environments. Institute of Navigation. doi: 10.33012/2020.17737

Kassas, Z. Z., Khalife, J., & Neinavaie, M. (2021, September). The first carrier phase tracking and positioning results with starlink LEO satellite signals. IEEE Transactions on Aerospace and Electronic Systems, 1–1. doi: 10.1109/taes.2021.3113880

Kassas, Z. Z. M., Khalife, J., Neinavaie, M., & Mortlock, T. (2020, July). Opportunity comes knocking overcoming GPS vulnerabilities with other satellites' signals. Inside Unmanned Systems, 30–35. Retrieved from `https://cdn.coverstand.com/61060/665176/93eb6b2b914b1d99e4b1a7d1771a993d43d19aed.2.pdf`

Klobuchar, J. (1987, may). Ionospheric time-delay algorithm for single-frequency GPS users. IEEE Transactions on Aerospace and Electronic Systems, AES-23(3), 325–331. doi: 10.1109/taes.1987.310829

Markoff, J. (2018). Time split to the nanosecond is precisely what Wall Street wants. Web. Retrieved from `https://www.nytimes.com/2018/06/29/technology/computer-networks-speed-nasdaq.html`

Michael. (2015, June). TimeseriesFromPSD(Sxx, fs, T, plot_on). MATLAB Central File Exchange. Retrieved from `https://www.mathworks.com/matlabcentral/fileexchange/47342-timeseriesfrompsd-sxx-fs-t-plot_on`

Misra, P., & Enge, P. (2012). Global positioning system: signals, measurements, and performance (Second ed.). Massachusetts: Ganga-Jamuna Press.

Morton, J. Y. (2019). Course on advanced GNSS software and applications. (University of Colorado Boulder)

Morton, Y. T. J., Diggelen, F., Spilker, J. J., Parkinson, B. W., Lo, S., & Gao, G. (Eds.). (2020a). Position, navigation, and timing technologies in the 21st century (Vol. 1). Piscataway, NJ: Wiley.

Morton, Y. T. J., Diggelen, F., Spilker, J. J., Parkinson, B. W., Lo, S., & Gao, G. (Eds.). (2020b). Position, navigation, and timing technologies in the 21st century (Vol. 2). Piscataway, NJ: Wiley.

Nykiel, G., Zanimonskiy, Y., Yampolski, Y., & Figurski, M. (2017, oct). Efficient usage of dense GNSS networks in central europe for the visualization and investigation of ionospheric TEC variations. Sensors, 17(10), 2298. doi: 10.3390/s17102298

O'Driscoll, C., Petovello, M. G., & Lachapelle, G. (2010, dec). Choosing the coherent integration time for Kalman filter-based carrier-phase tracking of GNSS signals. GPS Solutions, 15(4), 345–356. doi: 10.1007/s10291-010-0194-4

OneWeb. (2021). Web. Retrieved from https://directory.eoportal.org/web/eoportal/satellite-missions/o/oneweb#top

Psiaki, M. L. (2021, aug). Navigation using carrier doppler shift from a LEO constellation: TRANSIT on steroids. NAVIGATION, Journal of the Institute of Navigation, 68(3), 621–641. doi: 10.1002/navi.438

Pullen, S. (2007). Worldwide trends in gnss development and their implications for civil user performance and safety. In Japan gnss/gps symposium. Tokyo, Japan. Retrieved from https://web.stanford.edu/group/scpnt/gpslab/pubs/papers/Pullen_JapanGNSS_2007.pdf

Razavi, A., Gebre-Egziabher, D., & Akos, D. M. (2008, apr). Carrier loop architectures for tracking weak GPS signals. IEEE Transactions on Aerospace and Electronic Systems, 44(2), 697–710. doi: 10.1109/taes.2008.4560215

Rebeyrol, E., Macabiau, C., Ries, L., Issler, J.-L., Bousquet, M., & Boucheret, M.-L. (2006). Phase noise in gnss transmission/reception system. In Proceedings of the 2006 national technical meeting of the institute of navigation (p. 698-708).

Reid, T. G., Neish, A. M., Walter, T., & Enge, P. K. (2018). Broadband LEO constellations for navigation. Navigation, 65(2), 205–220. doi: 10.1002/navi.234

Reid, T. G. R., Chan, B., Goel, A., Gunning, K., Manning, B., Martin, J., ... Tarantino, P. (2020, apr). Satellite navigation for the age of autonomy. In 2020 IEEE/ION position, location and navigation symposium (PLANS). IEEE. doi: 10.1109/plans46316.2020.9109938

Reid, T. G. R., Houts, S. E., Cammarata, R., Mills, G., Agarwal, S., Vora, A., & Pandey, G. (2019). Localization requirements for autonomous vehicles. SAE Intl. J CAV 2(3):2019. doi: 10.4271/12-02-03-0012

Richards, M. A., Scheer, J. A., & Holm, W. A. (2016). Principles of modern radar, vol. I: basic principles (Vol. 1; D. R. Kay, Ed.). Raleigh, NC: The Institution of Engineering and Technology. Retrieved from https://www.ebook.de/de/product/11344959/principles_of_modern_radar.html

Riebeek, H., & Simmon, R. (2009). Catalog of Earth Satellite Orbits. Web. Retrieved from https://earthobservatory.nasa.gov/features/OrbitsCatalog

Riley, W., & Howe, D. A. (2000). Handbook of frequency stability analysis (techreport). 325 Broadway, Boulder, CO 80305: National Institute of Standards and Technology.

Skournetou, D., & Lohan, E. S. (2011). Personal satellite services (G. Giambene & C. Sacchi, Eds.). Springer Berlin Heidelberg. doi: 10.1007/978-3-642-23825-3_35

Snyder, L. (2007, October). Orbit1.svg. Wikipedia. Retrieved from https://en.wikipedia.org/w/index.php?title=File:Orbit1.svg

Starlink. (2021). Web. Retrieved from https://directory.eoportal.org/web/eoportal/satellite-missions/s/starlink

Sun, X., Qin, H., & Niu, J. (2013, July). Comparison and analysis of GNSS signal tracking performance basedon Kalman filter and traditional loop. WSEAS Transactions on Signal Processing, 9(3).

Tsui, J. B.-Y. (2004). Fundamentals of global positioning system receivers. John Wiley & Sons, Inc. doi: 10.1002/0471712582

WAAS T&E Team. (2017). Global positioning system (GPS) standard positioning service (SPS) performance analysis report (Tech. Rep.). Atlantic City International Airport, NJ 08405:

William J. Hughes Technical Center.

Wang, Y., Yang, R., & Morton, Y. J. (2020). Kalman filter-based robust closed-loop carrier tracking of airborne GNSS radio-occultation signals. IEEE Transactions on Aerospace and Electronic Systems, Institute of Electrical and Electronics Engineers (IEEE), 56(5), 3384–3393. doi: 10.1109/taes.2020.2972248

Winternitz, L. (2017). Introduction to GPS and other global navigation satellite systems. In 42nd annual time and frequency metrology seminar.

Won, J.-H., Pany, T., & Eissfeller, B. (2012, oct). Characteristics of kalman filters for GNSS signal tracking loop. IEEE Transactions on Aerospace and Electronic Systems, 48(4), 3671–3681. doi: 10.1109/taes.2012.6324756

Yang, R., Ling, K.-V., Poh, E.-K., & Morton, Y. (2017). Generalized GNSS signal carrier tracking: part i—modeling and analysis. IEEE Transactions on Aerospace and Electronic Systems, Institute of Electrical and Electronics Engineers (IEEE), 53(4), 1781–1797. doi: 10.1109/taes.2017.2673998

Yang, R., Morton, Y., Ling, K.-V., & Poh, E.-K. (2017). Generalized GNSS signal carrier tracking—part II: optimization and implementation. IEEE Transactions on Aerospace and Electronic Systems, Institute of Electrical and Electronics Engineers (IEEE), 53(4), 1798–1811. doi: 10.1109/taes.2017.2674198