Circuits and Systems
Mekelweg 4,
2628 CD Delft
The Netherlands
https://cas.tudelft.nl/

CAS-2022-4537165

M.Sc. Thesis

The impact of jamming and spoofing on GNSS signals

Pim Jansen B.Sc.

Abstract

Global Navigation Satellite Systems (GNSSs) have become a critical part of the infrastructure of modern society. Radio interference can introduce position or timing errors in systems that use GNSS or, in a worst-case scenario, block the reception of GNSS signals in full. Part of this critical infrastructure is, among others, power plants, banks, and transport.

Interference of GNSS signals could originate from nature, such as solar activity or ionospheric effects. Other interference could originate from unintentional sources (e.g., radio signals from a malfunctioning radio tower) or intentional sources such as a jamming or spoofing device. The latter is what this thesis will focus on.

This thesis consists of two parts. The first part is about jamming and elaborates on the impact of seven different forms of jamming on two types of GNSS receivers, a time-worn receiver and a cutting-edge receiver. The cutting-edge receiver has as option to turn on Interference Mitigation (IM). The performance of both receivers is the roughly the same in case the IM is turned off on the cutting-edge receiver. However, when the IM is turned on the cutting-edge receiver clearly is more resillient to the jamming signals.

In the second part of the thesis various types of spoofing are discussed. Due to time and hardware restrictions it was not possible to perform synchronous spoofing, which is an advanced form of spoofing. Instead, various concepts are discussed that describe how synchronous spoofing could be achieved.

TU Delft

Faculty of Electrical Engineering, Mathematics and Computer Science

Delft University of Technology

The impact of jamming and spoofing on GNSS signals

Master thesis

Thesis

submitted in partial fulfillment of the
requirements for the degree of

Master of Science

in

Embedded Systems

by

Pim Jansen B.Sc.
born in 's-Hertogenbosch, The Netherlands

This work was performed in:

Circuits and Systems Group
Department of Microelectronics
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Delft University of Technology

Delft University of Technology
Department of
Microelectronics

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for acceptance a thesis entitled **"The impact of jamming and spoofing on GNSS signals"** by **Pim Jansen B.Sc.** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: 13-06-2022

Chairman:

dr.ir. G.J.M. Janssen

Committee Members:

dr.ir. C. C. J. M. Tiberius

dr. ir. A. J. van Genderen

ir. A. Vroom

Preface

This thesis project was completed for the Embedded Systems masters program at the faculty EEMCS, Delft University of Technology. This research project was offered by CGI Nederland B.V. and was carried out under the supervision of both TU Delft and CGI. For 9 months, research has been done in searching for and evaluating vulnerabilities in Global Navigation Satellite Systems, specifically in the Global Positioning System (GPS). This thesis is the product of reading papers, discussions, the development of Matlab and C code and practical evaluation tests.

P. Jansen
Delft, May 2022

Acknowledgments

It would not have been possible to do the research for this thesis without the help of my advisors G.J.M. Janssen and A. Vroom. I would like express my sincerest appreciation to G.J.M. Janssen for his valuable insights, suggestions, and reserving time for meeting with me weekly over the course of this thesis. These meetings have been very pleasant and helpfull for me, by asking pressing questions he was able to drive me to do even better. Moreover, his feedback on my earlier version of this thesis was elaborate, and significantly aided to this final result.

I would like to thank A. Vroom for guiding me through the process of creating this thesis and getting me up to speed in the world of GNSS. Because of the discussions I had with him and his knowledge on GNSS I was able to make progress faster than I would have otherwise. It was very kind and nice of him that he was always available whenever I had a questions.

Furthermore, I would like to thank T. v.d. Oever and J. Gaňez Fernandez for their time and insights during the weekly lunch presentations.

Pim Jansen B.Sc.
Delft, The Netherlands
13-06-2022

Contents

List of Figures

List of Tables

List of variables

ϵ	Distance error relative to the true user's location.
σ	Standard deviation.
θ	Initial phase offset of signal.
τ	Pulse width.
ϵ_θ	Estimate of phase offset on received GPS signal.
ϵ_τ	Time delay offset between received PRN and replica PRN code.
ϵ_{f_D}	Estimate of remaining Doppler shift present on received GPS signal.
τ_i	Code phase of an authentic GPS signal.
ϕ_i	Carrier phase of an authentic GPS signal.
ϵ_{loss}	Squaring loss as result of non-coherent integration.
τ_{sl}	Code phase of a spoofed GPS signal.
ϕ_{sl}	Carrier phase of a spoofed GPS signal.
δt	Timing error.
δt^{sat}	Error of satellite clock.
δt_u	Timing offset error of receiver relative to atomic clock in satellite.
A	Amplitude of a signal.
C_i	Spreading code of an authentic GPS signal.
C_{sl}	Spreading code of a spoofed GPS signal.
D_i	Navigation data of an authentic GPS signal.
D_{sl}	Navigation data of a spoofed GPS signal.
G_{preamp}	Netto pre-amplification in dB.
N	Number of satellites where the receiver can receive the radio signals form.
NF_{amp}	Antenna low noise amplifier noise figure in dB.
NF_{rx}	Noise figure of the receiver module in dB.
NF_{sys}	Noise figure of a system in dB.
N_s	Number of satellites for which spoofed radio signals are simulated.
$N_{s(m)}$	Vector length of $s(m)$.
N_{samp}	Number of samples in a window.
P_{rec}	Received signal power in dBW.
$R_c(\epsilon_\tau)$	Autocorrelation function of the received PRN and the PRN replica.
R_k	Pseudorange from satellite to user receiver, where k is the satellite number.
S_{eff}	Effective magnitude correlation peak.
T	Chirp repetition interval.
T_{ant}	Antenna noise temperature (typically 130 K).
T_s	Signal period in s.
c	Speed of light ($c = 2.998 \cdot 10^8$ m/s).
d	Duty cycle.
dt_p	Propagation time it takes a radio signal to travel from a satellite to a user in s.
f_{IF}	Intermediate frequency in Hz.
f_{IM}	Image frequency caused by the local oscillator in Hz.

f_{LO}	Frequency local oscillator in Hz.
f_{RF}	Frequency received signal in Hz.
f_{bw}	Bandwidth of a chirp, $f_{bw} = f_{stop} - f_{start}$ in Hz.
f_c	Center frequency of GPS (1,575.42 MHz).
f_d	Doppler offset in Hz.
f_{rand}	Random frequency created by a 32-bit LFSR used for the broadband jamming signal.
f_{samp}	Sampling frequency in Hz.
f_{start}	Start frequency of a chirp in Hz.
f_{stop}	End frequency of a chirp in Hz.
g	Geometric distance between satellite and user receiver in m.
k	Chirp repetition rate.
k_b	Boltzmann constant ($1.3806 \cdot 10^{-23} m^2 kgs^{-2} K^{-1}$).
$max(bin_N)$	The maximum correlation within a frequency bin.
$mean(bin_N)$	The mean correlation of a frequency bin.
n	Number of windows that a signal is subdivided in.
$s(m)$	Vector of a signal of which a PSD is to be calculated.
t_1	Time of transmission at location of satellite.
t_u	Time user receives navigation message.
$var(bin_N)$	The variation in correlation within a frequency bin.
x_0	Approximation of receiver's x location.
x_u	x location user receiver.
y_0	Approximation of receiver's y location.
y_u	y location user receiver.
z_0	Approximation of receiver's z location.
z_u	z location user receiver.

Acronyms

ADC	Analog to Digital Converter.
AGC	Automatic Gain Controller.
AM	Amplitude Modulation.
AWGN	Additive White Gaussian Noise.
BOC	Binary Offset Carrier.
BPF	Band Pass Filter.
BPSK	Binary Phase-Shift Keying.
C/A	Coarse/Acquisition.
C/N0	Carrier-to-Noise ratio.
CDMA	Code Division Multiple Access.
CW	Continuous Wave.
DAC	Digital-to-Analog Converter.
DLL	Delay Lock Loop.
DSP	Digital Signal Processing.
FFT	Fast Fourier Transform.
FLL	Frequency Lock Loop.
FPGA	Field-Programmable Gate Array.
GNSS	Global Navigation Satellite System.
GPS	Global Positioning System.
HOW	Transference.
I&D	Integrate & Dump.
IF	Intermediate Frequency.
IFE	Integrated Front-End.
IFFT	Inverse Fast Fourier Transform.
IM	Interference Mitigation.
J/S	Jamming/Signal.
LFSR	Linear Feedback Shift Register.
LHCP	Left-Hand Circular Polarization.
LoS	Line-of-Sight.
LPF	Low Pass Filter.
NLoS	Non-Line-of-Sight.
NTP	Network Time Protocol.
PLL	Phase Lock Loop.

PNT	Positioning, Naviation and Timing.
PPD	Personal Privacy Devices.
PPS	Pulse Per Second.
PRN	Pseudorandom Noise.
PSD	Power Spectral Density.
RAIM	Receiver Autonomous Integrity Monitoring.
RF	Radio Frequency.
RHCP	Right-Hand Circular Polarization.
RSS	Root Squared Squared.
SDR	Software Defined Radio.
SNR	Signal-to-Noise Ratio.
STFT	Short-Time Fourier Transform.
SV	Satellite Vehicle.
TLM	Telemetry.
VCO	Voltage-Controlled Oscillator.

1 Introduction

Global Navigation Satellite System (GNSS) is the general term for satellite navigation systems. A well-known GNSS is the Global Positioning System, better known as GPS, which is developed by the United States. The European equivalent of GPS is called Galileo. Such systems are very complex and often consist of more than 30 satellites. Most people will know GNSS from using navigation apps on their smartphones. However, another characteristic GNSS has is that it can be used as a precise time source. If receivers are correctly configured, they can achieve almost the same time accuracy as atomic clocks but at a much lower cost. A GNSS can tell the time up to about 40 ns accuracy. A wide range of critical industries uses this functionality of a GNSS, such as power plants for synchronization and banks and the stock exchange for timestamping of transactions. It is even used for telecommunication networks. GNSSs have been identified as a vital process by the NCTV, which is the National Coordinator of Terrorism and Safety of the Netherlands [1]. They have become the backbone of the connected world [2].

In the last couple of years, significant advances have been made in the car industry toward autonomous driving. Accurate positioning information is required for autonomous driving applications, like lane detection, route guidance, collision avoidance and vehicle platooning. If GNSS is tampered with, this could have significant consequences for these systems[i] since their operation relies on GNSS functioning properly.

The Positioning, Naviation and Timing (PNT) signals are weak once they arrive on Earth. This makes these signals very vulnerable to interference which could be unintentional or intentional, as GNSSs can be compromised if specific signals are broadcasted on the same frequencies as used for GNSS. Unintentional interference sources could be the multipath propagation of GNSS signals, malfunctioning of broken electronics transmitting out-of-band, or a solar storm, making it harder for signals to reach Earth from space. Two types of intentional interference are jamming and spoofing. With jamming, a denial-of-service attack, adversaries are deliberately ”blinding” GNSS receivers. Usually, jamming is caused by a device transmitting powerful signals in the GNSS band. This can cause a range of effects, such as loss of tracking, increased pseudorange errors, denial of acquisition and false signal detection [3]. Often Personal Privacy Devices (PPD) are used for jamming. With jamming, the adversary is frequently the user of a GNSS who does not want his location to be disclosed. An example of this is a driver that wants to avoid paying a toll. The PPD can jam the frequency band, making it impossible for the car's receiver to derive its location. This way, the driver can avoid being tracked on toll roads [4]. Another form of intentional interference is spoofing. This is a more sophisticated way of tricking the receiver. With spoofing, counterfeit GNSS signals are transmitted to make the dependent system believe it is somewhere

[i] e.g., power plants, banks, stock exchange, and transport

that it is not. For this to work, it needs the receivers to believe that this transmitted signal is the original GNSS signal. Spoofing intends not to disrupt the system, as with jamming, but to feed the receiver with false information.

Due to the development of Software Defined Radios (SDRs), it has become easier to create and broadcast complicated radio signals. In the early 90s, the term SDR was defined by its creator Joseph Mitola [5]. SDR software and hardware today are available at relatively low prices [6], which has enabled malicious entities to deploy jamming and spoofing attacks more easily.

Most users take the correct operation of GNSS for granted. They do not consider that a GNSS can also be tampered with as is the case with other electronic systems (e.g., viruses and malware that are spread on computers and data servers). A subdivision can be made between GNSS signals that are available for civilians, unencrypted and freely available, and the GNSS signals that are reserved for governmental purposes. These signals are encrypted and can not be spoofed so easily. Since the GNSS signals for civilians are not encrypted, they are tampered with more easily.

It is of the utmost importance that awareness is raised concerning the vulnerabilities of GNSSs. So far, there have been very few implementations of jamming or spoofing detectors in receivers. Making it possible to detect that an entity is being attacked via implementation of recognition software in receivers of jamming or spoofing is step one. Once an entity knows it is being attacked, it can then undertake action.

This thesis will focus on the civilian GNSS signals. These signals were chosen as they are unprotected, although they are widely used in society. The jamming and spoofing of the governmental GNSS signals are beyond the scope of this thesis. These signals are protected by a keying system and redundancy checks, which makes them less vulnerable.

1.1 Research objectives

The goal of this thesis is to implement jamming signals and synchronous spoofing in the GNSS simulator of CGI Nederland B.V.

CGI's software-based signal simulator does not yet possess any jamming signals. Various types of jamming shall be implemented within the simulator. Subsequently, they will be validated against two commercial receivers to evaluate the receiver's robustness and the efficiency of the different jamming signals.

The simulator already possesses the ability to simulate asynchronous spoofing signals for the GPS and Galileo constellations. The difference between synchronous and asynchronous spoofing is explained in Section 6.3 and 6.4. Synchronous spoofing is a form of spoofing that can take over GNSS receivers without triggering detection methods. This makes synchronous spoofing a problematic type to mitigate. The goal is to generate a synchronous spoofing signal such that the observed correlation peak of this signal is within the tracking window of the receiver. For synchronous spoofing to work, the delay between the authentic GNSS signals and the simulated signals should be at most 500 ns. In Chapter 6, it is explained why this is the case.

1.2 Research questions

The research questions of this thesis are given below.

- How can jamming and synchronous spoofing be implemented in a software-based signal simulator?
- What is the impact of jamming and synchronous spoofing on the usage of GNSS by commercial receivers?

In the simulator a total of seven jamming signals have been implemented. Under these signals was a Continuous Wave (CW) signal, a pulsed sine wave and a broadband jamming signal. Furthermore, four types of chirp jamming signals have been implemented. The jamming signals have been evaluated against two GPS receivers and had a significant impact on the receivers' operation. The jamming signals were able to make the receivers lose track of the GPS signals.

Synchronous spoofing was not achieved due to time restrictions, though the principles of multiple types of spoofing are discussed. Moreover, spoofing concepts for future work are elaborated on.

1.3 Outline

The organization of this thesis is as follows. In Chapter 2 background information on GNSS is given. It starts with explaining the fundamental principles of GNSS. Furthermore, the techniques that make location estimation work using a GNSS are discussed. Subsequently, it is described which part of the spectrum is used. Following is a description of the navigation message that is encoded on the GPS signal. Hereafter the general operation of a GPS receiver is discussed. The chapter concludes with accuracy issues and vulnerabilities of a GNSS.

Chapter 3 is about state-of-the-art jamming methods. Two types of Continuous Wave (CW) jamming are discussed. Those jamming types often only jam on one specified frequency. Moreover, a broadband signal and four chirp jamming types will be elaborated on. These types of signals use jamming on more than one frequency, often covering a broader bandwidth of the spectrum.

In Chapter 4, it is discussed how the jamming signals that have been created within the simulator are validated in Matlab, as well as what the expected impact is of those jamming signals on a GPS receiver. Then in Chapter 5 the practical evaluation of the jamming signals is done on two types of GPS receivers; an older time-worn receiver and a newer cutting-edge receiver.

Chapter 6 describes state-of-the-art spoofing methods. Types of spoofing elaborated on are software-only spoofing, asynchronous spoofing, synchronous spoofing and time delay spoofing.

This thesis is concluded with Chapter 7, in which the conclusion, discussion and suggestions for future work are given.

Background 2

GPS was the first GNSS system that was developed. The system's main purpose was for a user to determine its location. Besides GPS other positioning systems have been developed, like Galileo (European Union), Glonass (Russia), BeiDou (China) and QZSS (Japan). The latter only provides coverage of part of Japan and Oceania.

This chapter will first explain the fundamentals of GNSS in Section 2.1. Following is Section 2.2 about the basic principles of a GNSS receiver. This chapter concludes with Sections 2.3 and 2.4, about accuracy issues of a GNSS and the vulnerabilities of a GNSS respectively.

2.1 Fundamentals GNSS

A GNSS constellation consists of three segments: space, ground and the user segment. The space segment consists of the satellites that have been brought into orbit around Earth, at an altitude of 23.222 km. These are broadcasting navigation messages. The ground segment comprises of multiple ground stations responsible for keeping each satellite's clock up to date by using an even more accurate clock placed on Earth. The ground stations also update parameters of the navigation message, oversee the satellites' health and adjust their orbits if needed. The user devices are the receivers that make use of the GNSS' PNT application.

A GNSS constellation is continuously sending messages to Earth. The propagation delay of these signals is used to determine the distance from a user to the satellite. The signals transmitted from the satellite travel approximately at the speed of light (c= $2.998 \cdot 10^8 m/s$). Since the speed of the wave and the time it takes for the message to reach Earth are known, the range from the satellite can be computed using Eqn. (2.1). Here, R_k is the pseudorange from the k^{th} satellite to the user, where k is the satellite number, and dt_k the pseudo propagation time it takes the signal to reach the user from the satellite.

$$R_k = c \cdot dt_k \tag{2.1}$$

In Figure 2.1 two satellites, S1 and S2, are shown. Both are transmitting navigation messages. These include the satellite's location at the moment of transmission and the time of transmission (t_1), among other information. Say a user receives the navigation message from S1 at t_u. From the pseudo propagation time, $dt_1 = t_u - t_1$, the receiver is able to compute its pseudorange to the satellite, $R_1 = c \cdot dt_1$. This, however, is not the true range of the satellite. This range has an offset, for instance due to the clock offset. Therefore it is called the pseudorange and the pseudo propagation time.

Since the satellite's location is known, the user's distance to the satellite can be deduced. The receiver could be at any point of the continuous blue circle. However, since most locations are in space and some underground, it can be assumed that the receiver is on either of the intersections of the blue line with Earth. Note that this figure is shown in 2D. However, the receiver's location can be thought of as if it is connected to the satellite by a wire of length R_1. This wire can move freely around the satellite in 3D. All of these points can be the current location of the receiver.

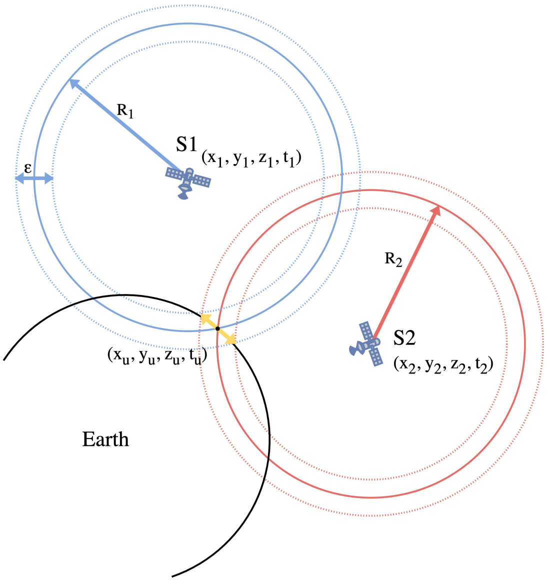


Figure 2.1: A 2D GNSS location estimation

By introducing more satellites, the receiver's location can be estimated more accurately as the measurement error becomes smaller. By intercepting a navigation message from S2, the receiver can also determine its distance to this satellite. The receiver's location should thus be where the blue and red lines cross with Earth. Though it seems that the location solving problem has now been solved, the location that is to be estimated has the form of (x_u, y_u, z_u), which consists of three unknowns. Thus, three equations (three satellites) are needed to solve the problem. This example is shown in 2D. However, a fourth satellite is required to pinpoint the receiver's location because of the receiver's time offset, more on this later this section.

Estimating the location depends on the propagation time of the messages transmitted from the satellites. To estimate the range correctly, the transmission and reception time should be known very accurately. If not, this quickly leads to large errors in the location estimation. The satellites are equipped with atomic clocks that have an error of about 0.1 ns, δt^{sat}. The receivers often have a cheaper clock that has poorer stability. These clock errors can lead to large errors in the location estimation. The error is given in Eqn. (2.2). An estimate of the duration of the propagation delay is 77.46 ms. This can be deduced from the Eqn. (2.1), given the satellites' altitude and the speed of light. A timing error (δt) of 1 μs leads to an error in the range of $\epsilon = 2.998 * 10^8 \cdot 1 * 10^{-6} = 299.8$ m, which is too large for navigational purposes.

$$\epsilon = c \cdot \delta t^{sat} \tag{2.2}$$

These timing errors come from multiple sources, a few of which will be mentioned here. One of these is the ionospheric delay, which causes the radio waves to slow down. This delay is caused by the ionosphere that resides from 60 km to 2000 km. The ionosphere contains a partly ionized medium that affects the speed at which electromagnetic radio waves can propagate through the ionosphere. The size of the ionospheric delay depends on the electron density in the ionosphere. The electron density in the ionosphere is affected mainly by the radiation of the Sun. During the day, there is a more significant activity since the radiation of the Sun causes a split of neutral atoms in free electrons and ions. The troposphere causes delays to the radio waves too. It is the layer between the Earth's surface to about 60 km. The delay caused by the troposphere depends on temperature, humidity and air pressure.

Fortunately, these errors can be modeled. However, when these errors are not corrected for, the location in the previous example is no longer estimated at the point of intersection of the blue and red circles with Earth, see Fig. 2.1. Instead, the user's location could be anywhere in the area indicated by the yellow arrows.

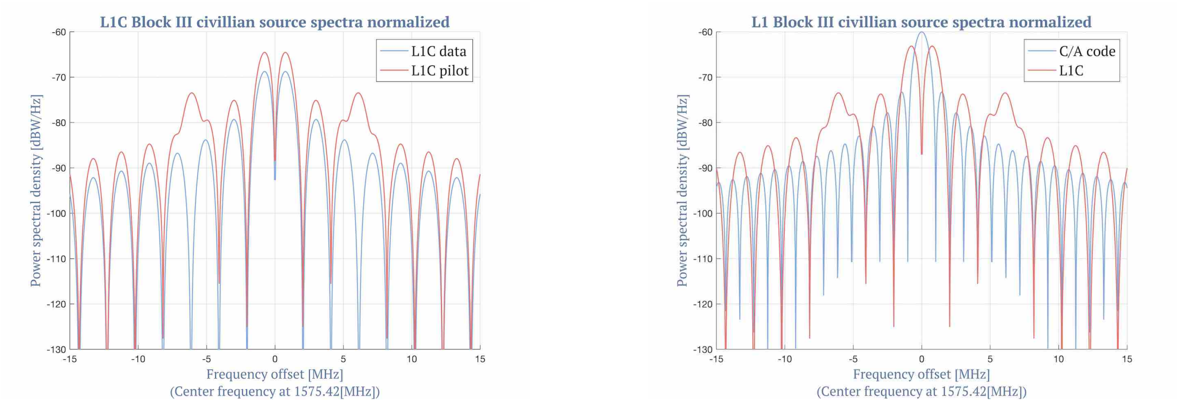
As mentioned, the clocks used in the receivers are not stable enough to be used directly for the position estimation because these have an offset, δt_u, relative to the atomic clocks in the satellites. Since this clock offset is the same relative to every satellite, a fourth unknown that needs to be solved can be introduced in the 3D localization problem. This results in the unknowns (x_u, y_u, z_u, δt_u). The equation that needs to be solved to obtain the location and the correct receiver time is given in Eqn. (2.3). At least four satellites are needed to solve this equation. If the receiver finds more satellites, the least-squares approach can be used to solve the problem.

$$\sqrt{(x_k - \mathbf{x_u})^2 + (y_k - \mathbf{y_u})^2 + (z_k - \mathbf{z_u})^2} = c \cdot (\mathbf{t_u} - t_k) \tag{2.3}$$

2.1.1 GPS signal

The GNSS satellites transmit at frequencies that reside in the L-band. The L-band frequency range goes from 1 GHz to 2 GHz. This frequency band is chosen because of the ionosphere and troposphere properties in this band. The Global Positioning System (GPS) occupies the upper L-band with its L1 signal from 1,559 to 1,610 MHz. GPS uses the lower L-band too for its L5 and L2 signals. These frequency bands are from 1,151 to 1,214 MHz and 1,215.6 to 1,350.0 MHz, respectively. The ionosphere reflects electromagnetic signals with frequencies less than 1 MHz [7]. Signals with a higher frequency will pass through the ionosphere, although they are delayed by the ionospheric activity. Radio frequencies below 1 GHz and above 10 GHz will suffer massive delays. Therefore these frequencies were not chosen for GNSSs. Also, if higher frequencies would have been chosen, the signals would be more sensitive to the electrons and ions freed from ionospheric gasses. Using high frequencies also makes it easier to use small user antennas.

Besides the L1 C/A code signal, a new signal has recently been added to GPS, the L1 Civil (L1C) signal. It is compatible with the L1 legacy signal. The L1C signal has a



(a) L1C

(b) L1 C/A and L1C

Figure 2.2: GPS signal spectrum

more advanced design that can offer higher performance than the L1 C/A signal does. The L1C signal consists of a data and a pilot component. Both components are built from Binary Offset Carrier (BOC) modulated signals. This is a signal form in which most of the signal's power is not located on one frequency but on two instead. The L1C data channel consists of a BOC(1,1) and the pilot channel of a time-multiplexing of BOC(1,1) and BOC(6,1). Both of the signal's power spectral densities can be seen in Fig. 2.2a. The pilot channel does not contain any data. However, an extra encrypted watermark code is overlayed on the pilot signal, making it possible to let users know when the signal is being spoofed. Besides, it is possible to use this watermark code to authenticate the receiver's location to another party. In Fig. 2.2b the combined L1C signal is shown next to the L1 C/A legacy signal.

Although these new signals have been implemented in GPS, many receivers that have been built are still based on the L1 C/A code. The L1C signal was first broadcasted on January 13 2020 [8].

2.1.2 Navigation message

Each GNSS uses a different layout for its navigation message. In this section, the navigation message of GPS will be discussed. For the navigation messages of other GNSS constellations refer to [9].

GPS sends its navigation data at a relatively low bit rate of 50 bps. This data rate is high enough for GPS to transmit all the necessary data within a reasonable time. With a low data rate, the reliability is higher, which is needed since the data is transmitted over a very noisy channel.

The entire GPS navigation message consists of 25 frames, which takes 12.5 minutes to be transmitted. In Fig. 2.3 an overview is given of the GPS navigation message. Each frame consists of five sub-frames, of which the first three sub-frames are satellite specific. Sub-frames 4 and 5 are common to all satellites. Each of the sub-frames start with the same two words. These are the Telemetry (TLM) word which is needed for synchronization, and the Transference (HOW) word that provides information about

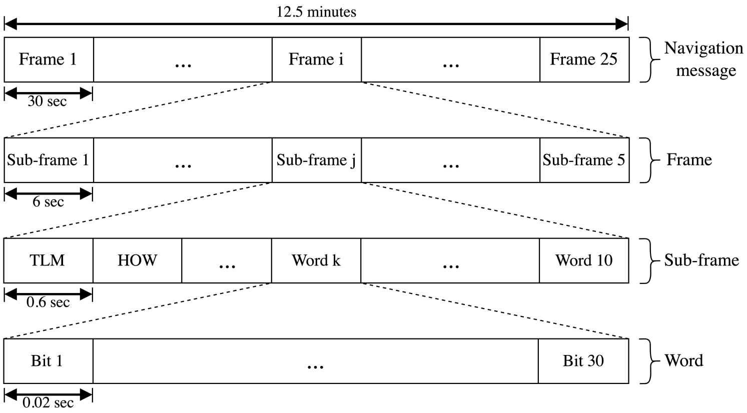


Figure 2.3: Overview GPS navigation message

the time. The following type of information is contained in each sub-frame.

- *Sub-frame 1* contains information about the satellite's clock offset and health information.
- *Sub-frame 2 & 3* contain the ephemerides[i] of the satellite.
- *Sub-frame 4* provides information about the ionosphere, part of the almanac and UTC information.
- *Sub-frame 5* contains data from the almanac and information about the constellation status.

The almanac, which is transmitted in sub-frames 4 and 5, contains the orbital data of all the satellites that are in orbit. This data is less accurate than the ephemerides. The data in the almanac is valid for up to 90 days. If the almanac is stored on a receiver, it can locate the satellites in view faster. The entire almanac is too large to send in one sub-frame. Therefore, sub-frames 4 and 5 are made up of 25 pages each.

In each frame, the sub-frames 1 to 3 are repeated. For sub-frames 4 and 5, the next page is selected. Thus for the entire navigation message to send, 25 frames need to be transmitted. A receiver does not require the entire almanac to find its location. After a satellite has sent the navigation message, it starts over again.

When the receiver wants to download the entire almanac and misses one bit, it has to wait for 12.5 minutes before that bit is sent again.

The navigation message is modulated onto a carrier wave using Binary Phase-Shift Keying (BPSK). Besides the navigation message, a Coarse/Acquisition (C/A) code is modulated in the same way onto the carrier wave. In Fig. 2.4 a simplified version of the modulation can be seen. More information about the navigation message and the modulation can be found in [9] or [10]. This C/A code is a Pseudorandom Noise (PRN) code of 1023 chips. Each of the satellites has a unique PRN gold code [11]. This PRN code is used to distinguish the different satellites from each other. Gold codes

[i] This is information about the orbit the satellite is following and its speed.

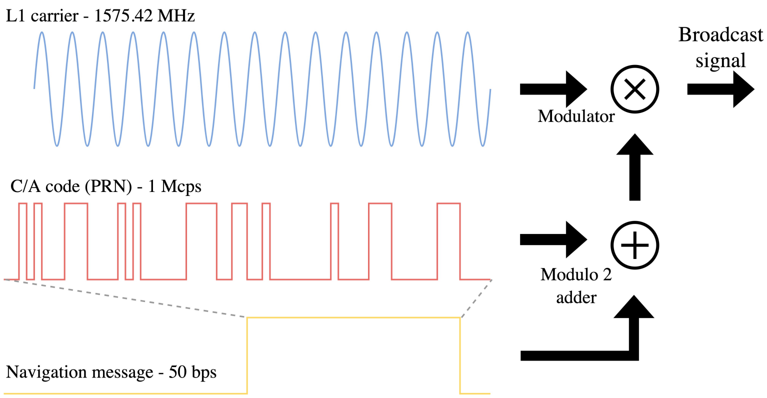


Figure 2.4: GPS signal format (figure not to scale)

correlate very strongly with themselves, but they have small cross-correlations. They are constructed by XOR'ing two maximum length codes. These codes are created using Linear Feedback Shift Registers (LFSRs).

The GNSSs GPS and Galileo make use of the Code Division Multiple Access (CDMA) technique. This technique allows multiple transmitters to send information on the same frequency, sharing the available bandwidth. CDMA uses a spread spectrum technology. In case of GPS each satellite encodes its signal with its own unique PRN code, as shown in Fig. 2.4. By making use of correlation, it is possible to recover the GNSS signals from each satellite from below the thermal noise floor. This is explained in Section 2.2.

The C/A code is transmitted at 1.023 Mchips/s and will transmit a new chip every 1 μs. The chipping sequence is periodic, repeating itself after every 1023 chips. The C/A code is used by the receiver to track the GPS signal, see Section 2.2.

When GNSS signals reach the Earth, their signal power is around -130 dBm. This is about 19 dB below the thermal noise floor. Thus the signal is entirely embedded in noise on reception. The thermal noise power density is -174 dBm/Hz at 290 Kevlin. Assuming a 2 MHz bandwidth (for the C/A code of GPS) of GNSS signals this gives a noise power of $-174 + 10 \cdot log_{10}(2 \cdot 10^6) = -111$ dBm.

2.2 Basic principles of a GNSS receiver

The processing of GNSS receivers are subdivided into three phases. These are called acquisition, tracking & correlation, and decoding the navigation message, which helps determine the receiver's location. During the acquisition phase the receiver searches for satellites that might be in view. For each of the satellites in view, the receiver tries to estimate the time delay and generate an estimation of the carrier phase. The Doppler shift estimation is used to find the correlation peak more quickly. This information helps to speed up the acquisition process. The following paragraphs explain this in more detail.

Once the signals from the satellites reach the receiver down on Earth, they are embedded in a white-noise-like signal. A large part of this noise is filtered in the

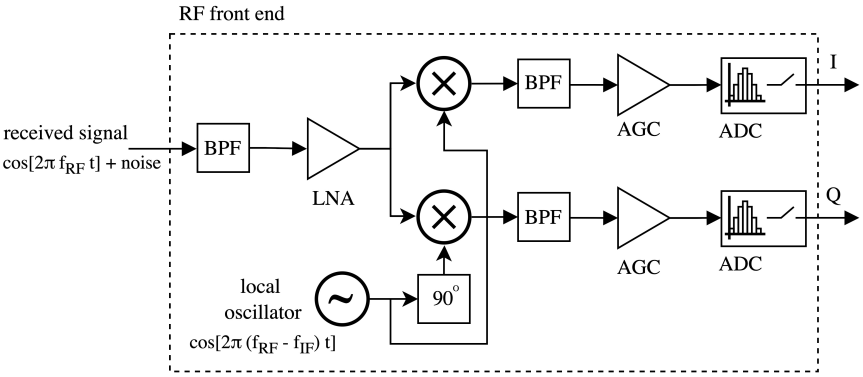


Figure 2.5: Block diagram front-end GPS receiver

receiver front-end, see Fig. 2.5. A more high level overview of a GPS receiver is shown in Fig. 2.7. There are two main types of front-end receiver implementations: heterodyne and homodyne. The latter approach is often referred to as direct conversion. However, this approach will not be explained in this thesis, as it is less commonly used in GNSS receivers. The front-end shown in Fig. 2.5 is a heterodyne mixer. In Fig. 2.5 a difference is made between the in-phase and quadrature-phase of the signal, to keep Fig. 2.7 uncluttered this difference has been left out.

The heterodyne mixer is more commonly used in GPS receivers than the homodyne mixer. Since the output of the homodyne mixer is a baseband signal, this can greatly be affected by flicker noise of the mixer. This effect is greater when the incoming signal is weak, as is the case with GPS signals. More information about flicker noise can be found in [12].

By down-converting to an Intermediate Frequency (IF) in a heterodyne mixer, the Automatic Gain Controller (AGC), Analog to Digital Converter (ADC) and amplifiers do not have to operate on the received frequency (f_{RF}), which is in the range from 1.2 GHz to 1.6 GHz. By doing so, the circuitry can be optimized for the frequency range of the IF that is used.

$$f_{IM} = \begin{cases} f_{RF} + 2f_{IF}, & \text{if } f_{LO} > f_{RF} \text{(high-side injection)} \\ f_{RF} - 2f_{IF}, & \text{if } f_{LO} < f_{RF} \text{(low-side injection)} \end{cases} \tag{2.4}$$

A heterodyne mixer shifts the received signal frequency down so that the spectrum is centered around the IF (f_{IF}). This operation is done by performing a multiplication of the received signal by $cos[2\pi(f_{RF} + f_{IF})t]$. This operation does, however, not only shift the desired spectrum to the IF, but an image spectrum (f_{IM}) will be shifted to the IF too. The image frequencies are located at the f_{RF} plus or minus two times f_{IF} depending on whether the frequency of the local oscilator is higher or lower than f_{RF}, see Eqn. (2.4) (where $f_{LO} = f_{RF} \pm f_{IF}$). That would result in both spectra being present at the IF. The image spectrum is not under control of the GNSS and could be a much stronger signal, therefore interfering with the weak GNSS signal, see Fig. 2.6. This phenomenom can be mitigated by placing a Band Pass Filter (BPF) around the desired GNSS spectrum, hereby filtering out the image spectrum. This is the first BPF the received signal passes through in Fig. 2.5.

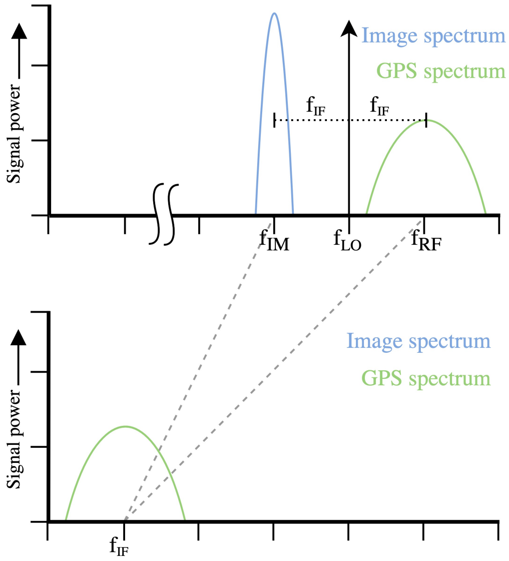


Figure 2.6: Interference image frequency without filtering

The BPF works best if the image spectrum and the GPS spectrum lay far apart. This is beneficial because the transition from pass to stopband is often not shaped as an ideal block filter, but the filter requires some bandwidth to decrease the contribution of adjacent frequencies. This means a high IF should be chosen. However, if the IF is chosen to be too high, the problems created by a high IF might outweigh the benefits of the improved image removal. This is a tradeoff than can be optimized for by receiver manufacturers.

Assuming that the mixer is a low-side downconverter, f_{LO} will be equal to f_{RF} - f_{IF}. However, when looking at the equations, it can be seen that the spectrum is shifted to other frequencies. This can be seen from Eqn. (2.5), using trigonometric identities (where $\omega_{RF} = 2\pi f_{RF}$ and $\omega_{LO} = 2\pi f_{LO}$). The first term in Eqn. (2.5) results in the desired IF. The second term, however, results in the unwanted frequency $2f_{RF}$ - f_{IF}=f_{RF} + f_{LO}. These unwanted frequencies will be removed by another BPF after which the signal is fed through an AGC. The full derivation using Euler's formula is shown in A.3. The derivation shown is only carried out for the in-phase part of the signal. The derivation is roughly the same for the quadrature-phase, apart from the 90-degree phase shift.

$$\begin{aligned} cos(\omega_{RF}t) \cdot cos(\omega_{LO}t) &= \frac{1}{2}[cos((\omega_{RF} - \omega_{LO})t) + cos((\omega_{RF} + \omega_{LO})t)] \\ &= \frac{1}{2}[cos(2\pi(f_{RF} - (f_{RF} - f_{IF}))t) + cos(2\pi(f_{RF} + (f_{RF} - f_{IF}))t)] \\ &= \frac{1}{2}[cos(2\pi f_{IF}t) + cos(2\pi(2f_{RF} - f_{IF})t)] \end{aligned} \tag{2.5}$$

The AGC is responsible for adjusting the gain of the front-end so that the receiver can benefit from the full dynamic range. When multiple antennas are used and a powerful interference source is present from a specific direction, the AGC can reduce the gain in that direction, making the receiver less sensitive to the interference source. At last, the signal is quantized by an ADC.

Following the receiver's front-end, the signal is multiplied by a locally generated signal from a local oscillator. By doing so, the IF is removed from the signal. What remains is the PRN code combined with the data and some noise. The following paragraph will explain this in more detail.

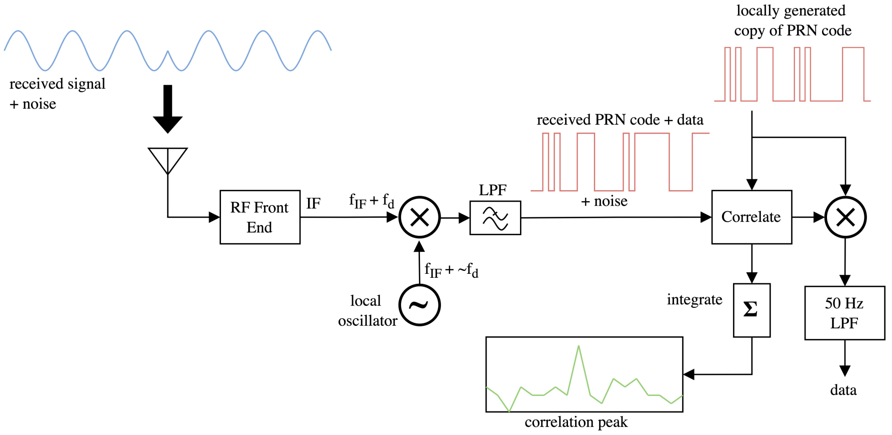


Figure 2.7: Block diagram of GPS receiver

The received signal is transmitted at the L1 frequency. This frequency is known by the receiver and is down-converted to an IF (f_{IF}). Since the satellite is moving and the receiver might be too, a Doppler offset (f_d) changes the signal's frequency. The signal thus consists of the frequency $f_{IF} + f_d$. The local oscillator tries to recreate this frequency exactly. However, since the Doppler frequency cannot precisely be known, a residual frequency offset will remain present on the signal after the multiplication.

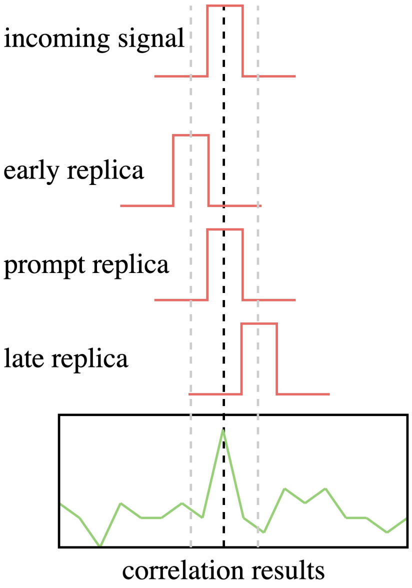


Figure 2.8: Example output of correlators

The residual frequency can be removed from the signal using in-phase and quadrature sampling. Instead of multiplying the signal by one replicated carrier frequency, the signal will be multiplied by an in-phase and a quadrature component. The mixer uses the same principle as used in the front-end mixer. After passing through the mixer, the signal is fed through a Low Pass Filter (LPF) since there should only be low frequencies left on the signal. The signal that remains contains the PRN code of the satellite, the data modulated on top, and some residual noise.

Unfortunately, the remaining signal cannot be used yet to process the navigation data since the signal is still embedded in too much noise. The signal, therefore, is correlated by locally generated copies of the satellites' PRN codes. The signal is correlated with multiple locally generated copies, some earlier and some later than the received signal. The correlator will search over the 1023 possible different chips and the possible Doppler frequencies the signal could be on. This gives a broad area that has to be searched.

The locally generated copy that is exactly in sync with the received signal will have a correlation peak. The other locally generated copies will give noise. For this purpose, gold codes are used for the PRN codes. They strongly correlate when the signal is precisely the same, but they have a weak correlation when the signal is different. This is shown in Fig. 2.8.

The PRN code combined with the navigation data is used to estimate the propagation time of the signal. The travel time is required to derive a pseudorange. This has been explained in Section 2.1. The PRN code is repeated every 1 ms, and the signal's travel time is in the range of 70 ms. Thus between the satellite and the receiver reside 70 fully transmitted PRN codes. By correlating the PRN code with a replica, the receiver can derive the remaining fraction of the PRN code. However, the receiver can not obtain the total amount of PRN codes that have been transmitted between the satellite and the receiver. Thus the PRN code can not be used as a standalone estimator for dt_k. Therefore, the navigation message is used for further estimating the propagation time.

Once the correlation has been completed, the PRN code is stripped off and a 50 bps data stream remains. The data stream is passed through a LPF of 50 Hz before further processing of the navigation data is done.

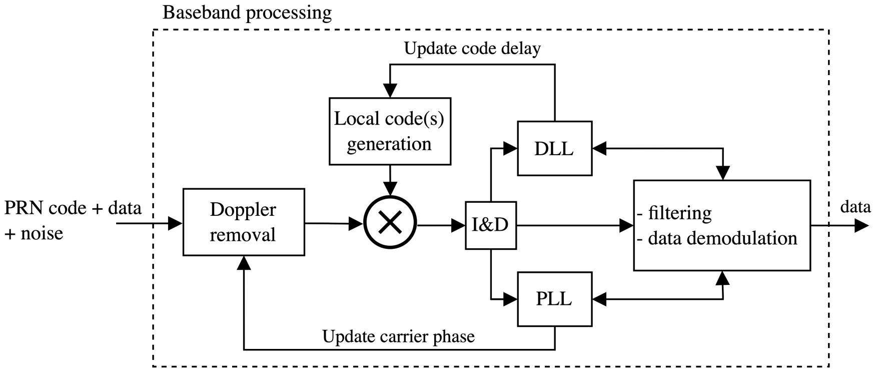


Figure 2.9: Block diagram baseband processing GPS receiver

In Fig. 2.9 a more detailed representation of the final stages is shown, from the GPS receiver shown in Fig. 2.7. The first block shows that the Doppler shift, f_d in Fig. 2.7, that the local oscillator uses to remove the remaining Doppler shift is kept up to date by the Phase Lock Loop (PLL). The locally generated copy of the PRN code is updated by the Delay Lock Loop (DLL).

2.2.1 Parallel Code Phase Search

The blocks that follow are the Integrate & Dump (I&D) block, the DLL and the PLL. These blocks are combined to implement the parallel code phase search algorithm [13], which will be explained in this section. It is used to calculate the correlation of the incoming GPS signal with the local PRN replicas.

Following the Doppler removal block is the Integrate & Dump (I&D) block, here the signal is integrated using coherent, see Section 2.2.2, or non-coherent integration, see

Section 2.2.3. Often a combination of coherent and non-coherent integration is used.

The output of the I&D block from Fig. 2.9 is given in Eqn. (2.6). Here A is the amplitude, D_i the navigation data on the signal, $R_c(\epsilon_\tau)$ is the autocorrelation function of the received PRN code and its replica. The terms $cos(\epsilon_\theta)$ and $sinc(\pi\epsilon_{f_D}T_I)$ are used in the PLL. Here ϵ_{f_D} represents the remaining Doppler shift that is still present on the signal after the first Doppler estimation that had been done by the local oscillator and ϵ_θ is the estimated carrier phase of the incoming signal. In some receivers there is another loop, called the Frequency Lock Loop (FLL). Sometimes the FLL is combined with the PLL.

The final term, $n(m)$, represents the remaining noise present on the signal, where m is the sample number.

The goal is to make the three error terms, ϵ_τ, ϵ_{f_D}, and ϵ_θ, go to zero. If this is achieved by the tracking loops the terms $R_c(\epsilon_\tau)$, $cos(\epsilon_\theta)$ and $sinc(\pi\epsilon_{f_D}T_I)$ go to 1. Then Eqn. (2.6) simplifies to Eqn. (2.7).

$$I(m) = A \cdot D_i(m) \cdot \underbrace{R_c(\epsilon_\tau) \cdot cos(\epsilon_\theta) \cdot sinc(\pi\epsilon_{f_D}T_I)}_{=1} + n(m) \tag{2.6}$$

$$I(m) = A \cdot D_i(m) + n(m) \tag{2.7}$$

In Eqn. (2.8) the autocorrelation function is shown. Here C_i is the spreading code on the received signal and $\tilde{C_i}$ is the local replica of the PRN with a time delay offset of $\tilde{\tau}$.

The GPS signal may be considered wide-sense stationary [14] when the multiplicative noise is extremely low. This makes it possible to use the Wiener–Khinchin theorem [15] since this theorem only holds for wide-sense stationary processes. The theorem states that instead of performing the autocorrelation function in the time domain as a circular convolution, it can be performed in the frequency domain using multiplication. Circular convolution in the time domain is multiplication in the frequency domain. By making this domain shift, the operation becomes less computationally expensive.

A representation of computing the autocorrelation using the Wiener-Khinchin theorem is shown in Eqn. (2.9). It shifts the computation from the time domain to the frequency domain using a Fast Fourier Transform (FFT). From the PRN code of the local replica, the complex conjugate is taken. After the computation has been done in the frequency domain, the result is converted back to the time domain using the Inverse Fast Fourier Transform (IFFT).

The autocorrelation function is used to deduce the time delay offset, ϵ_τ, which is used in the DLL for further processing.

$$R_c(\epsilon_\tau) = \int_{T_0+(m-1)T_I}^{T_0+mT_I} C_i(t+\tau) \cdot \tilde{C_i}(t+\tilde{\tau})dt \tag{2.8}$$

$$R_c(\epsilon_\tau) = IFFT[FFT[C_i(t)] \cdot FFT[\tilde{C_i}(t)]^*] \tag{2.9}$$

After the correlation, the code and phase delay can be derived by the DLL and PLL. Now that the Doppler shift and PRN code have been stripped of the signal it can be used for further processing such as filtering and demodulation of the data encoded on the signal.

2.2.2 Coherent integration

Data bits from the navigation message are sent every 20 ms, which can change the phase of the PRN code by 180 degrees, depending on the sign of the data bit that has been sent. In coherent integration, the sign of the received signal integrates with it.

The phase of the received PRN code can therefore change every 20 ms. If the PRN code changes of phase it correlates in the opposite sign, as before the phase shift, with the local replicas generated. If the correlation was positive at first, the phase shift will decrease the correlation peak. When the integration of the signal is performed across multiple phase shifts of the signal, the resulting correlation will likely lie around zero since multiple correlation peaks have averaged out each other. The phase changes that occur due to the data bits cannot be corrected since the signal has not yet been found. The data that is sent is still unknown to the receiver.

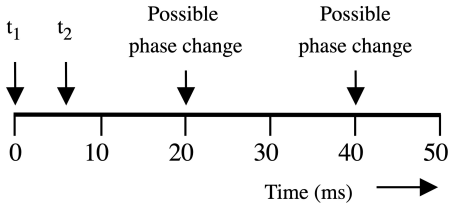


Figure 2.10: Example phase change during correlation

Using coherent integration, the maximum time that can be correlated is 20 ms since a phase shift might happen after this time. See Fig. 2.10, assuming that at 0 ms a phase change happend and the correlator starts correlating the signal at t_1, it can correlate for 20 ms before the next possible phase change. However, if it starts correlating at t_2, it can only correlate for 14 ms. The phase changes happen due to the navigation bits encoded on the signal. The structure of the navigation message and the data rates have been discussed in Section 2.1.2.

2.2.3 Non-coherent integration

When using non-coherent integration, two channels are created: an in-phase and a quadrature-phase channel. The carrier frequency of the quadrature channel is shifted by 90 degrees. The in-phase channel is modulated on a cosine and the quadrature channel on a sine wave. The correlation results of both channels are squared and added. After the square root is taken, this operation is called Root Squared Squared (RSS). By doing so, the correlation peak is no longer dependent on the phase changes created by the data bits of the navigation message.

Using non-coherent integration makes it possible to correlate the signal for a longer time period than 20 ms. Although, by using non-coherent integration, there is a loss of about 1 dB caused by squaring the signal.

In Fig. 2.11 the origin of the squaring error is shown. The top two figures are the correlation of the signal's in-phase and quadrature-phase components. This is how the result of coherent integration would be. For this example, all the energy is in the in-phase channel. Both the in-phase and quadrature-phase components have a mean noise of zero.

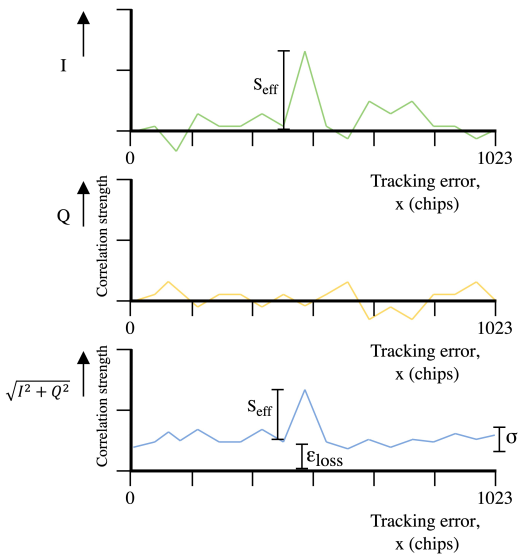


Figure 2.11: Origin squaring error non-coherent integration

The result of non-coherent integration is shown in the bottom plot of Fig. 2.11. As a result of the RSS operation, the effective magnitude (S_{eff}) of the correlation peak changes, as well as the mean noise value, which is now above zero. The effective peak magnitude decreases as a result of the squaring loss (ϵ_{loss}). This is what causes the loss of roughly 1 dB. An auxiliary commodity as a result of the RSS operation is that the standard deviation (σ) of the correlation decreases.

2.2.4 Location estimation

When estimating the position of a GPS receiver, it is not the location of the receiver that is calculated, but instead, an assumption of the receiver's location is made and the error from the assumed location to the real receiver's location is calculated. The reason this approach is used has to do with the pseudorange equations. In Eqn. (2.10) the generalized pseudorange equation is given. The derivation of this equation is given in A.1. This equation only considers the offset of the satellite and receiver clocks and neglects the contribution of other error sources like the atmosphere. The variables shown in bold are the ones that need to be estimated.

$$R_k = \underbrace{\sqrt{(x_k - \mathbf{x_u})^2 + (y_k - \mathbf{y_u})^2 + (z_k - \mathbf{z_u})^2}}_{\mathrm{g}} + (\boldsymbol{\delta}\mathbf{t^u} - \delta t^{sat}) \cdot c \tag{2.10}$$

The geometric distance, g, needs to be linearized in order to solve the equation. This can be done by using a first order Taylor expansion [16]. To this end an estimation of the receiver's location (x_0, y_0, z_0) is made. The geometric distance can then be estimated by Eqn. (2.11), and can be rewritten as Eqn. (2.12) (where $g_0 = \sqrt{(x_k - x_0)^2 + (y_k - y_0)^2 + (z_k - z_0)^2)}$. The derivation of this can be found in A.2. When g is replaced in Eqn. (2.10) and the known terms are moved to the left side this gives Eqn. (2.13).

$$g \approx g_0 + g\frac{\partial}{\partial x_u}|_{x_0,y_0,z_0}\underbrace{(x_u - x_0)}_{\Delta x} + g\frac{\partial}{\partial y_u}|_{x_0,y_0,z_0}\underbrace{(y_u - y_0)}_{\Delta y} + g\frac{\partial}{\partial z_u}|_{x_0,y_0,z_0}\underbrace{(z_u - z_0)}_{\Delta z} \tag{2.11}$$

$$g \approx g_0 - \underbrace{\frac{x_k - x_0}{g_0}}_{\overline{a_k}} \Delta x - \underbrace{\frac{y_k - y_0}{g_0}}_{\overline{b_k}} \Delta y - \underbrace{\frac{z_k - z_0}{g_0}}_{\overline{c_k}} \Delta z \tag{2.12}$$

$$\begin{aligned} R_k &= g_0 - \overline{a_k}\Delta x - \overline{b_k}\Delta y - \overline{c_k}\Delta z + (\boldsymbol{\delta}\mathbf{t^u} - \delta t^{sat}) \cdot c \\ \underbrace{R_k - g_0 + \delta t^{sat} \cdot c}_{\gamma_k} &= -\overline{a_k}\Delta x - \overline{b_k}\Delta y - \overline{c_k}\Delta z + \boldsymbol{\delta}\mathbf{t^u} \cdot c \end{aligned} \tag{2.13}$$

When a GPS receiver has four or more satellites in view, the linear system is as in Eqn. (2.14), which can be solved using least squares estimation [17], shown in Eqn. (2.15). The U matrix then gives the distance between the estimated position (x_0, y_0, z_0) and the real receiver's location (x_u, y_u, z_u). By iterating this process the estimated position can be calculated more accurately every round.

$$\begin{gathered} \underbrace{\begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \end{bmatrix}}_{\Gamma} = \underbrace{\begin{bmatrix} -\overline{a_1} & -\overline{b_1} & -\overline{c_1} & 1 \\ -\overline{a_2} & -\overline{b_2} & -\overline{c_2} & 1 \\ -\overline{a_3} & -\overline{b_3} & -\overline{c_3} & 1 \\ -\overline{a_4} & -\overline{b_4} & -\overline{c_4} & 1 \end{bmatrix}}_{H} \underbrace{\begin{bmatrix} \mathbf{\Delta x} \\ \mathbf{\Delta y} \\ \mathbf{\Delta z} \\ \boldsymbol{\delta}\mathbf{t^u} \cdot c \end{bmatrix}}_{U} \\ \Gamma = H \cdot U \end{gathered} \tag{2.14}$$

$$U = (H^T \cdot H)^{-1} \cdot H^T \cdot \Gamma \tag{2.15}$$

2.3 Accuracy issues with GNSS

The gravitational force on the satellites is smaller than on Earth since they are in an orbit 23.222 km above Earth. Therefore, by Einstein's law, time goes faster onboard the satellites. However, due to the speed at which the satellite is traveling time goes slower aboard the satellite, which means time is going faster on Earth. This time offset is corrected by base stations on Earth that correct the time on the satellite.

In dense urban environments, buildings can severely impact the accuracy of the GNSS's PNT. Here three distinct scenarios can be made [18]. In the first scenario, the GNSS signals are either blocked or unavailable. In the second scenario, the signals are received via a Non-Line-of-Sight (NLoS) path because the direct Line-of-Sight (LoS) path is blocked by a building. This can introduce positioning errors in the range of tens of meters [19]. In the third scenario, signals arrive at the receiver via LoS and NLoS paths. The error in this scenario is dependent on the strength of the reflected signals, path delays, phase differences and the design of the receiver [20]. Receiving these multipath signals can affect the outcome of the correlation function. The multipath propagation can either constructively or destructively influence the correlation peak. Constructive multipath propagation will influence the correlation peak of the prompt replica of the PRN code to increase. The peak of the correlation function increases, gets broader and slightly shifts to the right, which results in a positive timing error. In contrast, destructive multipath propagation will decrease the correlation peak, resulting in a negative timing error [see 21, pp. 420-424]. In Fig. 2.12 the effect of multipath propagation on the correlation peak of a GPS signal is shown. The correlation function is essential in finding the correct time delay of the signal. This can lead to large localization errors if not dealt with properly. This is an issue since the correlator in the receiver does not search for the exact peak of the correlation function. It takes a range and assumes the peak lies in the middle of this range. Thus, if the correlation function's peak gets broader or smaller, this affects the location at which the correlator expects the peak to be. The LoS signals of GNSSs have a Right-Hand Circular Polarization (RHCP), whereas most NLoS signals have a Left-Hand Circular Polarization (LHCP). The changes in polarization are often the result of reflections. By using a dual-polarization antenna, techniques can be applied that reduce multipath effects. The cost for this is more expensive hardware and an increase in computational complexity as a distinction between the signals that are RHCP and LHCP needs to be made by the receiver [22].

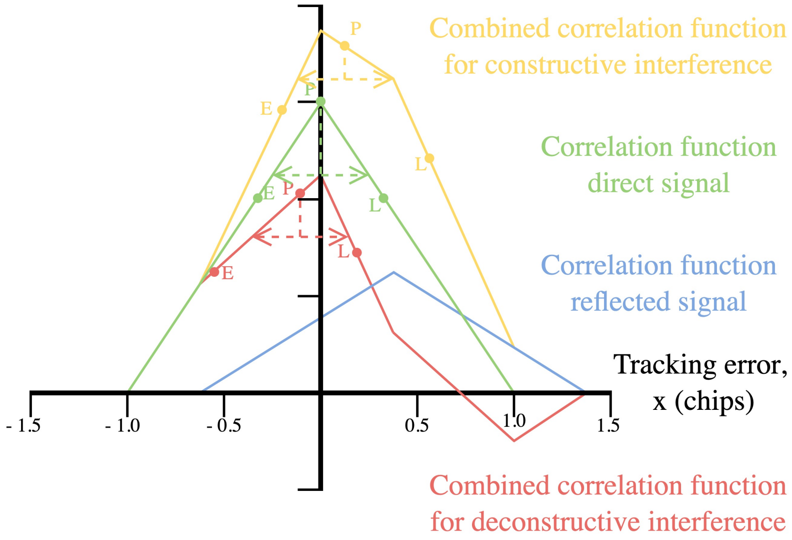


Figure 2.12: Effect of multipath propagation on correlation function

2.4 Vulnerabilities of GNSS

In Section 2.1.2 it is mentioned that GNSS signals are received on Earth below the thermal noise floor. It is not difficult for an adversary to broadcast a signal with a signal strength above the thermal noise floor, which makes it difficult for receivers to recover the GNSS signals if a strong jamming signal is interfering with the GNSS signals.

Matched power attacks can remain undetected by the AGC of the receiver since the power is not substantially higher than the original GNSS signal. However, this is often not the case with jamming attacks. These typically transmit a much stronger signal than the AGC observes.

Consumer receivers frequently only have one antenna implemented in the device.

This makes the receivers more prone to jamming attacks. Since the receivers cannot tell from which direction the interfering signal is coming, the AGC cannot adjust the gain properly to reduce the impact of the jamming attack.

The civil signals of GPS and Galileo are not encrypted, which makes them susceptible to spoofing as well. The radio signals of GNSS for civil use are openly available and can be recreated by using a simulator and a SDR. An adversary can recreate the GNSS radio signals and change some of its data to confuse the receiver. This can have a severe impact on applications that rely on GNSSs.

2.4.1 Vulnerable components of a GNSS receiver

In civil GNSS receivers, the Integrated Front-End (IFE) is the weak spot concerning interference or jamming. If the IFE, which includes the AGC, does not instantaneously recover from interference or jamming, then the GNSS signal cannot be tracked properly by the downstream digital signal processing [23]. Whenever the IFE has no built-in mitigation techniques, then narrowband interference and jamming are generally two times more effective for the same amount of in-band power than band limited white noise [23].

When a pulsed jamming signal is designed to match the time constant of the AGC, this can give the jammer 10's of dBs of advantage [24]. The time constant of an AGC is specific for every GNSS receiver. In some receivers, it can be around 10 ms, while in moving cars, it is around 1-2 ms due to the fast-changing environment. A faster time constant makes the AGC more sensitive to changing GNSS signal strength. Although a control loop with a short time constant (<<1 ms) can produce an unstable output gain [25]. Since AGCs react relatively slow, incorrect quantization levels will be selected. This problem can be solved by using a faster AGC and more quantization levels. Another technique that can be applied is to use blanking. Every time a pulse is detected, the ADC outputs a zero, thereby removing the distorted signal and lowering the impact of the pulse jammer [24]. The samples that contain the pulsed interference are then considered useless. However, as soon as the pulse vanishes, the receiver's front-end can return to normal operation. The downside is that signal loss is introduced in the samples for which pulse interference was detected, as these samples were removed from the incoming signal.

In the tracking loops of the receiver, the DLL and PLL, the interference or jamming will introduce jitter. The noise that is present on the received GPS signal increases as the received signal is a combination of the authentic GPS signal and the jamming signal, which is transmitted in the same frequency band as the GPS signal. This will have as a result that the correlation of the incoming GPS signal with the local replica has a higher noise floor, making it more complex for the tracking loops to track the GPS signal. If the interference or jamming is too strong, the introduced jitter will increase up to the point that the tracking loops are no longer able to track the PRN code of the signal properly. This can be seen when comparing Fig. 4.2b, where the receiver is not being jammed and the correlation peak of the GPS signal is visible, Fig. 4.8c, where jamming has raised the noise floor within the correlator and Fig. 4.8d, where the receiver is jammed with too much power for it to be able to distinguish the authentic correlation peak from the noise.

When a GNSS receiver is undergoing a synchronous spoofing attack, the adversary attemps to have the spoofed signal arrive at the receiver with just a little higher power than the authentic GPS signal. This is needed since the receiver often tracks the highest correlation peak it finds and the correlation peak becomes higher if the imcoming signal power is higher. The synchronous spoofing attack often does not set off any alarm bells within the IFE or the AGC specifically. A synchronous spoofing attack exploits the vulnerabilities of the tracking loops of a receiver, the DLL and the PLL. How this works is explained in more detail in Chapter 6.

State-of-the-art jamming methods of GNSS signals 3

This chapter will discuss some state-of-the-art jamming methods. Modern GNSS receivers are getting more resilient to jamming methods by using mitigation techniques to counter jamming attacks. Therefore, jamming techniques should not solely focus on the potential of adding interference to the receiver but should exploit the weaknesses of the receivers' front-ends. This makes it harder for receivers to mitigate the effects of the jamming attack.

In Fig. 3.1 an illustration is provided that shows how a jamming attack looks like. An adversary is broadcasting jamming signals, thereby introducing too much interference in a certain region, making it very hard or impossible for a user receiver to determine its position and time derivations.

This chapter is structured as follows. It starts with Section 3.1 which describes with what power levels GPS signals are received on Earth and what power levels should be used when wanting to jam a GPS receiver. Hereafter, Continuous Wave (CW) and pulsed sine wave jamming are discussed in Sections 3.2 and 3.3 respectively. Following is a description of broadband jamming in Section 3.4. Section 3.5 elaborates on different types of chirp jamming.

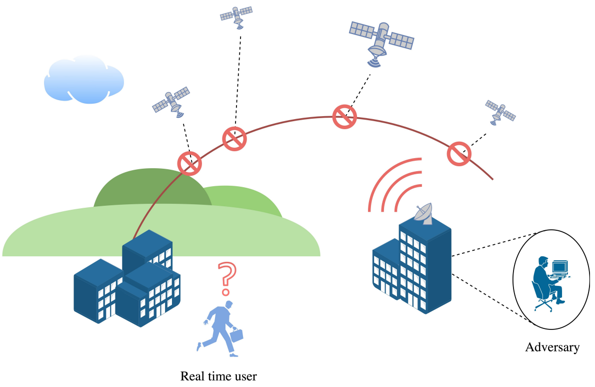


Figure 3.1: Illustration of jamming

3.1 Power levels of GPS and jamming signals

GNSS signals are transmitted from satellites in a medium Earth orbit. These signals are weak when they reach the Earth's surface, even below the thermal noise level. Using Friis formula, given in Eqn. (A.8), it is possible to calculate an upperbound of the received signal power. This gives a received power of -118.1 dBm. (the derivation is given in A.4) The minimum guaranteed received signal power of the transmitted signals is given in table 3.1[i].

When a jammer is transmitting a signal 10,000 times stronger than the received GNSS signals, the signal used for jamming is still very weak. Jamming levels are referred to in terms of a power ratio, called the Jamming/Signal (J/S) ratio, expressed in units of decibels, as shown Eqn. (3.1), where j_{rc} is the received jammer power in watts and s_{rc} is the received signal power in watts.

Table 3.1: Received minimum and maximum terrestrial RF signal strength [26], [27], [28].

GNSS	**Signal**	**Isotropic signal level (dBW)**
GPS	L1 C/A	-158.50
	L1C Pilot	-158.25
	L1C Data	-163.00
Galileo	E1	-157.25

$$J/S = 10 \cdot log_{10}(j_{rc}/s_{rc}) \tag{3.1}$$

In a situation where the received jamming power is 10,000 stronger than the received signal power, this corresponds to a Jamming/Signal (J/S) of 40 dB. If an adversary wants to jam the L1 C/A signal with a jammer that is 10,000 stronger, the power of the jammer that should be received at the target's receiver is -158.50 dBW + 40 dB = -118.5 dBW, which is not difficult to transmit from a simplistic antenna.

Earlier research has already shown that with broadband Additive White Gaussian Noise (AWGN), high power narrowband and pulsed signals, it is possible to fully saturate the front-end of GNSS receivers causing the receiver to lose track of the GNSS signal [29].

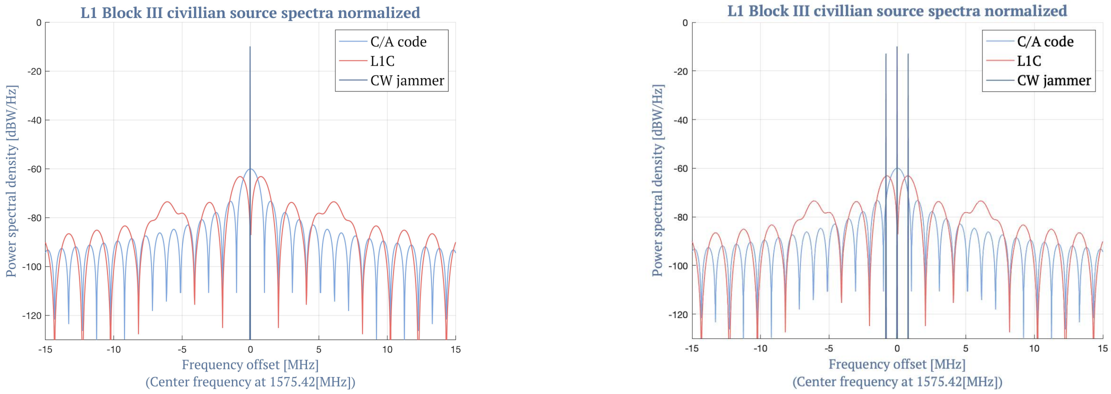
3.2 Continuous wave tone jamming L1

The most simple form of jamming is a tone jammer broadcasting the L1 frequency on a Continuous Wave (CW). CW jamming is a jamming form that transmits a cosine on the carrier frequency of GPS. The signal's form is shown in Eqn. (3.2), where A, f_c and θ are the amplitude, the L1 frequency and a random initial phase respectively. The spectrum of the signal consists of a single spike at 1,575.42 MHz. This spectrum combined with the L1 C/A and L1C spectra can be seen in Fig. 3.2a. In the plot shown in Fig. 3.2a, the signal spectra of GPS and the jamming signal appear to be separate. However, for the receiver these signals are seen as a summation. The receiver multiplies the signal by a local reference PRN code. This has been described in more detail in Section 2.2. This operation makes the satellite's L1 C/A signal bandwidth collapse from 1.023 MHz to 50 Hz as the PRN code is stripped off. The jammer signal goes from a negligible bandwidth to a bandwidth of 1.023 MHz. At the end of the receiver, the signal is passed through a Band Pass Filter (BPF), see Fig. 2.7. The

[i] Only GNSS signals relevant to this thesis are shown

50 Hz signal of the original L1 C/A signal will remain almost un-attenuated when passed through this filter. However, most interference will be filtered out when the jammer signal is passed through this filter. The fraction of energy of the jammer that gets through the filter is $\frac{50}{1{,}023{,}000} = \frac{1}{20{,}460}$. For a receiver to be able to track a legacy signal, the signal strength needs to be around ten times greater than the noise [30]. The receiver will lose track of the authentic GPS signal if the J/S exceeds 33.1 dB $(20,460/10 = 2,046 = 10 \cdot log_{10}(2,046) = 33.1\text{dB})$. In Eqn. (3.2) and all following equations regarding jamming forms, $j_m(t)$ is used to express the jamming signal.

$$j_m(t) = A \cdot cos(2\pi f_c t + \theta) \tag{3.2}$$



(a) Single tone CW jamming

(b) Multiple tone CW jamming

Figure 3.2: GPS signal spectrum with CW jamming

3.2.1 Multiple tone continuous wave jamming

As mentioned in Section 2.1.1, the signal spectrum of GPS has been modernized. The BOC L1C signal has been added to the spectrum. For more modern receivers to lose track of the signal, it no longer suffices to only jam the L1 C/A signal. It would very likely lead to distortion at the receiver. However, to be certain that the receiver will lose track of the GPS signal, the L1C signal should be jammed too. The main lobes of this signal reside on ± 1,023 MHz of the center frequency of GPS [27]. The signal that will be transmitted has the form as shown in Eqn. (3.3). The frequencies used here are $f_1 = 1,575.42$ Mhz, $f_2 = \frac{1{,}023}{2} = 511.5$ kHz and $f_3 = \frac{1{,}023}{2} = 511.5$ kHz. The frequencies that are present on the signal after modulation are $f_1 = 1,575.42$ MHz, $f_1 - f_2 - f_3 = 1,575,420 - 2 \cdot 511.5 = 1,574.397$ MHz and $f_1 + f_2 + f_3 = 1,575,420 + 2 \cdot 511.5 = 1,576.443$ MHz. The peaks of the BOC are 1,574.397 and 1,576.443 MHz.

$$j_m(t) = A \cdot \prod_{n=0}^{2} cos(2\pi f_n t + \theta_n) \tag{3.3}$$

By jamming on the center frequency of GPS's L1 signal, 1,575.42 MHz and on the side lobes of the L1C signal, more noise will be introduced in the receiver.

This method is reasonably simple to implement since only three sine waves should be modulated on top of each other, using Amplitude Modulation (AM). This uses the same mathematical principle as shown in Eqn. (2.5). However, it is a modulation with three cosines instead of two cosines. The short derivation is shown in Eqn. (3.4), where $\omega_n = 2\pi f_n$. The full derivation using Euler's identity can be found in A.5.

The BOC signal of GPS is received with roughly 3 dB less power than the L1 C/A, see Fig. 3.2a. Therefore a jamming signal was created that also transmitted with 3 dB less power on the side frequencies, f_2 and f_3, which could be seen from the derivation shown in Eqn. (3.4).

The expectation is that modern receivers can filter most of the interference generated by this jammer using, for example, notch filters. These are effective filters against CW interference [31]. Due to time restrictions, this jamming form will not be implemented in the simulator.

$$\begin{aligned}
cos(\omega_1 t) \cdot cos(\omega_2 t) \cdot cos(\omega_3 t) &= \frac{1}{2}[cos((\omega_1 - \omega_2)t) + cos((\omega_1 + \omega_2)t)] \cdot cos(\omega_3 t) \\
&= \frac{1}{2}[cos((\omega_1 - \omega_2)t) \cdot cos(\omega_3 t) + cos((\omega_1 + \omega_2)t) \cdot cos(\omega_3 t)] \\
&= \frac{1}{4}[cos((\omega_1 - \omega_2 - \omega_3)t) + cos((\omega_1 - \omega_2 + \omega_3)t) \\
&\quad + cos((\omega_1 + \omega_2 - \omega_3)t) + cos((\omega_1 + \omega_2 - \omega_3)t)] \\
&= \frac{1}{4}[cos((\omega_1 - \omega_2 - \omega_3)t) + 2 \cdot cos(\omega_1 t) + cos((\omega_1 + \omega_2 + \omega_3)t)]
\end{aligned} \tag{3.4}$$

3.3 Pulsed sine wave jamming

With this type of jamming, pulses of a sine wave are transmitted instead of an uninterrupted broadcasted signal. The pulsed sine wave has the same waveform as the CW, see Eqn. (3.2), but periods of silence are introduced. These silence periods are defined by the duty cycle (d) of the signal. The duty cycle is the fraction of the time that the signal is broadcasted. A pulse signal is mainly characterized by its duration and duty cycle. See Eqn. (3.5) for the duty cycle. In this equation τ is the pulse width of the signal and T_s is the signal's period. In Fig. 3.3 an example is given of how a duty-cycled signal is built. A pulse jammer requires less power than required for a CW since it is not continuously broadcasting, making it more difficult to mitigate properly. For example, a notch filter is tuned to the frequency of the pulsed sine wave and this notch filter is continuously turned on. This will make the receiver filter the signal also when no interference is present on the signal, thus removing information and degrading the quality.

A disadvantage of this jamming method is that it requires knowledge of the receiver's design to affect the AGC in an optimum way [32]. The pulse jammer can then keep the receiver constantly in acquisition mode even though the jammer is not continuously on. The jammer will continue to disrupt the receiver in the off period since the components need a recovery period before the distortion that was caused is entirely removed.

In the simulator, it is possible to modify the center frequency of the pulsed sine wave and its period and duty cycle.

$$d = \frac{\tau}{T_s} \tag{3.5}$$

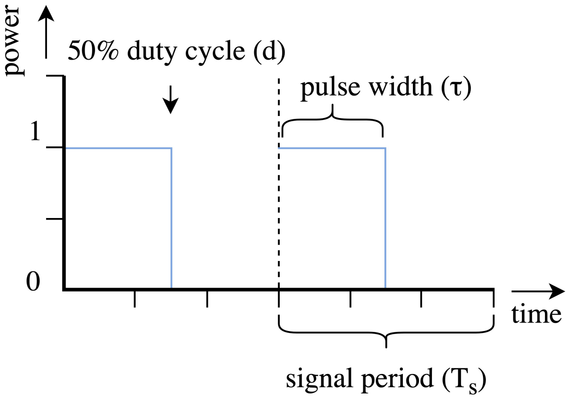


Figure 3.3: Example duty cycle

3.4 Broadband jamming

Instead of jamming specific frequencies as in the CW tone jamming of Section 3.2 or with pulsed sine wave jamming in Section 3.3, another possibility is to jam an entire frequency band. This makes it more difficult for the receiver to estimate at which frequency the noise resides and remove the noise from the received signal. In [33], it was shown that broadband jamming is the second worst type of interference for a receiver of the types that were researched. Broadband noise is an interference source that is Gaussian in nature.

Gaussian white noise cannot be predicted as it is random, making it hard to filter from an incoming signal on a receiver. Gaussian white noise sources can also be found in nature, such as thermal noise. In nature Gaussian white noise often has a relatively low but constant PSD. However, such a signal can also be built and by transmitting this signal with a high power, the noise floor rises.

For creating Gaussian white noise, a random number generator is necessary. Unfortunately, it is impossible to create a true random number on a computer. However, it is possible to create pseudorandom numbers. The broadband jamming signal will therefore be pseudorandom.

Linear Feedback Shift Register

The Gaussian white noise signal will be generated by creating a pseudorandom frequency. This is done by multiplying 1 MHz by a normalized pseudorandom number. The pseudorandom number is generated by a Linear Feedback Shift Register (LFSR) of 32-bits, also called taps.

An LFSR is a shift register that uses a combination of two or more of its current states as input. The 32 states of the LFSR combined are the output of the LFSR. This is a binary number that can be converted to a decimal, done by multiplying the value of each tap by $2^{(\text{tap }\#)}$, in table 3.2 an example is given for the 4-bit binary number 0110.

Table 3.2: Conversion binary to decimal

Tap #	**Value**	**Factor**	**Sum**	
0	0	$0 \cdot 2^0$	0	
1	1	$1 \cdot 2^1$	2	
2	1	$1 \cdot 2^2$	4	
3	0	$0 \cdot 2^3$	0	+
			6	

An LFSR of 32-bits can generate a maximum of length of $2^{32} - 1 = 4,294,967,295$ different numbers. As the sample rate of the HackRF is 20 MHz the signal will repeat itself after $\frac{4{,}294{,}967{,}295}{20 \cdot 10^6} = 214.7$ seconds. This maximum length can only be reached if the LFSR uses the right taps as input. For an 32-bit LFSR these taps are 1, 5, 6 and 31 [34]. The 32-bit maximum length LFSR is shown in Fig. 3.4. The start state of the LFSR can be any except a scenario in which all states are zero, if this is chosen the LFSR will always remain zero.

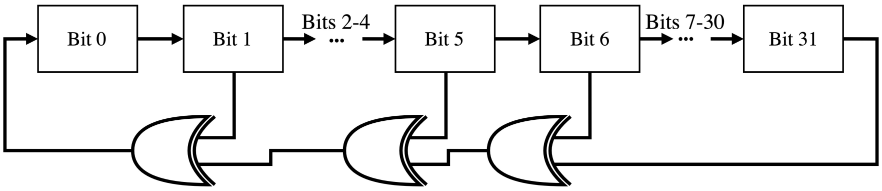


Figure 3.4: 32-bit LFSR with taps

Bandpass filtering

When the 1 MHz frequency is multiplied with the normalized output of the LFSR the signal, the frequency ranges from 0 to 1 MHz, called f_{rand}. However, when creating the broadband jamming signal, $j_m(t)$ see Eqn. (3.6), the signal spreads to a bandwidth of 10 MHz wide[ii] as this is the Nyquist frequency, $\frac{20 \cdot 10^6}{2} = 10$ MHz. The signal has, at this point, a normal distribution. However, to in order to only jam a specific bandwidth,

[ii] No explanation could be found why the bandwidth spreads to 10 MHz

the signal has to be filtered by a bandpass filter. This is done by the bandpass filter of the HackRF. After filtering the signal by the bandpass filter, the signal has a Gaussian distribution.

$$j_m(t) = A \cdot cos(2\pi f_{rand} t + \theta) \tag{3.6}$$

3.5 Chirp jamming

Chirp jamming is a more complex type of jamming to filter for a receiver, as well as the broadband jamming signal, compared to CW jamming. A chirp jammer changes the frequency of the signal it transmits. The frequency is varied between a start and stop frequency, f_{start} and f_{stop}, respectively. Once the frequency reaches the stop frequency, the chirp often repeats itself. However, this is not the case with symmetric chirp signals. Besides the start and stop frequencies, the chirp repetition period, T, is an important parameter in defining a chirp. This is the rate at which the chirp is repeated.

$$k = \frac{f_{stop} - f_{start}}{T} \tag{3.7}$$

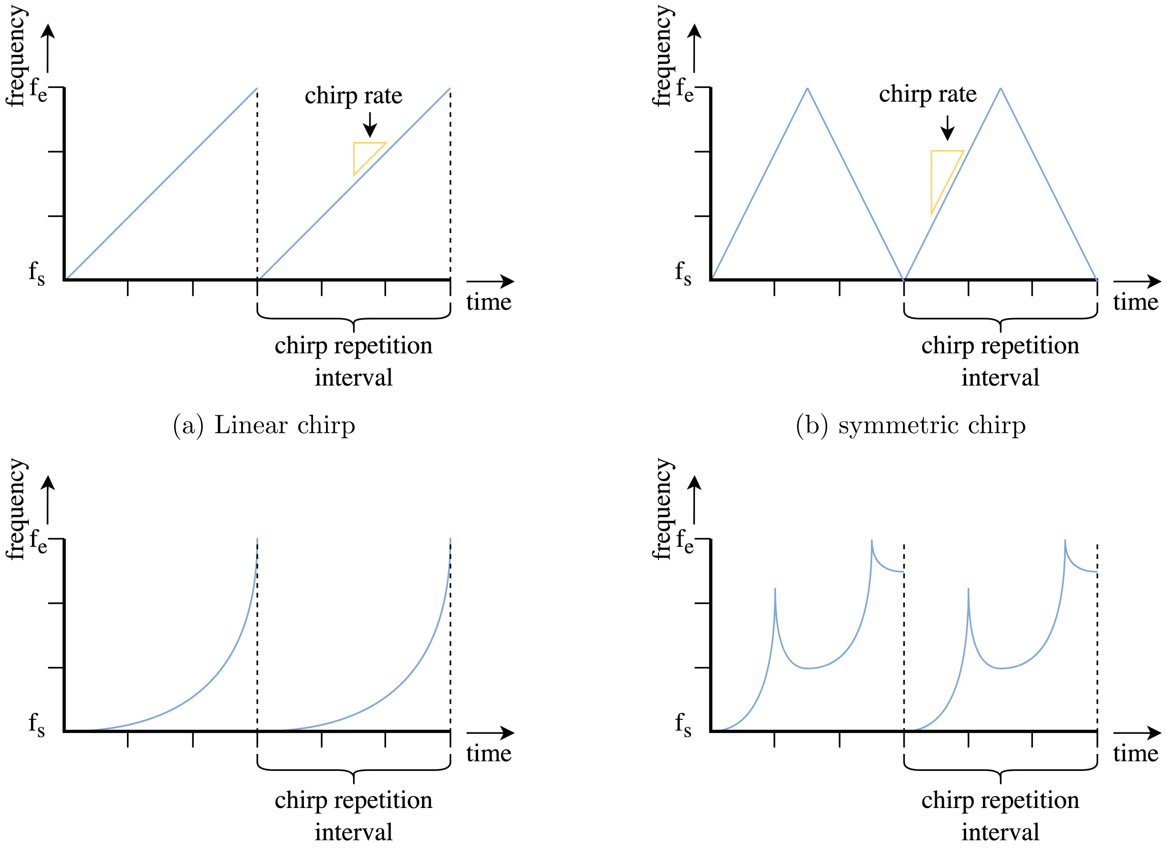
The chirp rate, k, the rate at which the frequency increases over time, can be computed from the start and stop frequency, together with the chirp repetition rate. This formula can be seen in Eqn. (3.7). Techniques have been developed that can filter out linear chirps [35]. The frequency of chirps can be changed in multiple ways. The most common are.

- *Linear.* The frequency of the chirp increases linearly in one direction. In a spectrogram, this chirp type looks like a saw-tooth wave.
- *Symmetric.* The chirp frequency changes linearly in two directions. The frequency first increases to the stop frequency like the linear chirp and then the frequency is linearly reduced back to the start frequency. In a spectrogram, this looks like a triangular wave.
- *Non-linear.* A non-linear chirp could, for example, be a chirp that exponentially increases its frequency between the start and stop frequency.
- *Unpredictable.* These chirps have a behavior that is hard to characterize and are often referred to as noise signals [36].

While there are numerous ways to implement chirps, these are the most well-known forms. In Fig. 3.5 a visualisation of the chirps is shown.

3.5.1 Mathematical representation of jamming signals

In total, four chirp forms have been implemented in the simulator. All of the hereafter discussed chirp jamming wave forms had been created in Matlab before their implementation in the simulator was done. The first chirp that was implemented was the linear chirp, see Section 3.5.2. However, since this waveform will introduce frequency jumps in



(a) Linear chirp

(b) symmetric chirp

(c) Non-linear chirp

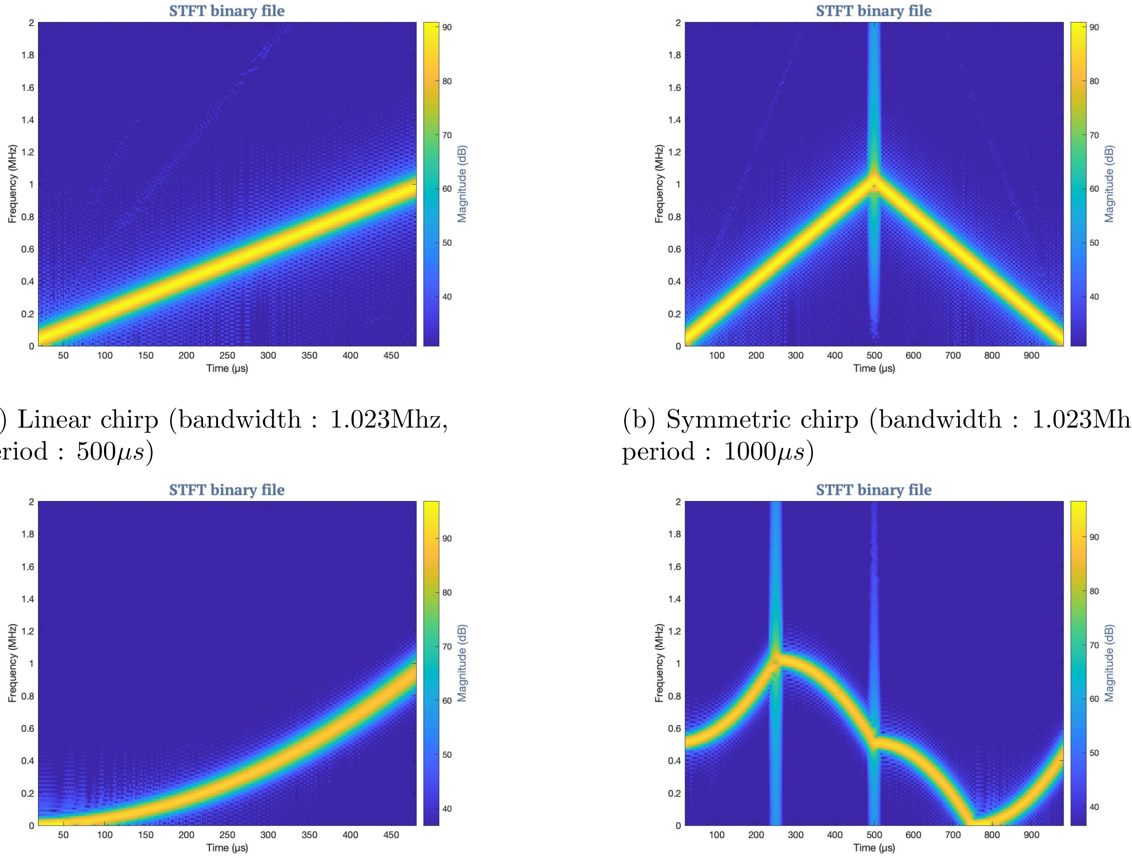
(d) Unpredictable chirp

Figure 3.5: Chirp types

the spectrum when multiple chirps are transmitted in succession, creating a sawtooth waveform. Therefore another chirp type was implemented, called the symmetric chirp, see Section 3.5.3. This chirp does not contain frequency jumps.

Interference Mitigation (IM) in GNSS receivers is getting better, and literature shows, [35], that removing linear chirps from the spectrum is possible by parameter estimation of the chirp. According to [35] the parameters of a non-linear chirp are hard to predict. Thus another chirp called the quadratic chirp was implemented, see Section 3.5.4. However, the quadratic chirp concentrates a relatively large amount of power at one side of GPS's main lobe. This can be seen in the Power Spectral Density (PSD) plots in Section 4.7 Fig. 4.10a. The PSD is explained in Section 4.1.1. Therefore, a design for a chirp was made that was non-linear and concentrates more of its power at the center frequency of GPS. The result of this was the double quadratic chirp, see Section 3.5.5. From Fig. 4.11a in Section 4.8, it can be seen that the power distribution of the double quadratic chirp concentrates more power around the center frequency of GPS than the quadratic chirp does. The expectation is that this will result in more distortion being introduced in the correlator of GNSS receivers. More on the correlator in Section 2.2.

In Fig. 3.6 the four chirp forms that have been implemented are shown. The chirps shown in this figure have a bandwidth of 1.023 MHz. This is the single-sided spectrum of the chirps.



(a) Linear chirp (bandwidth : 1.023Mhz, period : 500μs)

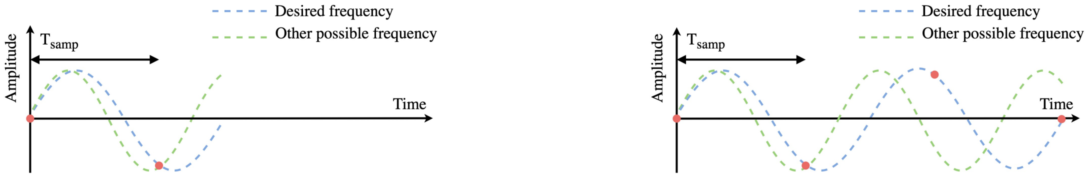
(b) Symmetric chirp (bandwidth : 1.023Mhz, period : 1000μs)

(c) Quadratic chirp (bandwidth : 1.023MHz, period : 500μs)

(d) Double quadratic chirp (bandwidth : 1.023Mhz, period : 1000μs)

Figure 3.6: Chirp forms

The period is chosen to be 500 μs. This reduces the creation of residual frequencies and results in a cleaner chirp. If the period is chosen to be smaller, this results in 'broader' chirp signals because the sampling frequency remains the same, which is 20 MHz. This is because fewer sample points are used to create a chirp with a shorter period than those used to create the signal with a period of 500 μs. This makes it harder to create a neat interpolation between the sample points. Because of the interpolation, the created signal is not the exact frequency but could be a bit higher or lower. In Fig. 3.7 this principle is shown. The red dots in this figure represent the sample points. In the top Fig. 3.7a the period of the signal is shorter and it can be seen that the green signal matches the sampling points too, say this is the signal with a period of 100 μs, shown in Fig. 3.8a. In the second Fig. 3.7b the signal's period is longer and therefore has more sampling points. Say this is the signal with a period of 500 μs, shown in Fig. 3.8b. This figure shows that the green signal does not match all sampling points and thus is not the correct frequency of the signal.

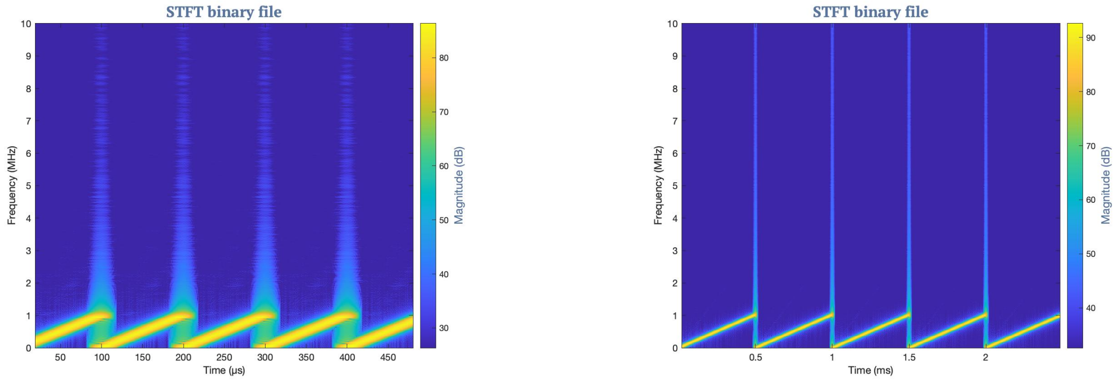


(a) Example short signal period

(b) Example long signal period

Figure 3.7: Different amount of sample points

In Fig. 3.8 the difference between a linear chirp with a period of 100 μs and 500 μs is shown. From this figure, it can be seen that the chirp with a period of 100 μs has a 'broader' chirp, as was just explained. It also has a cluttered spectrum at the transition points of the chirps, resulting in more replicas arising in the spectrum when these signals are transmitted. This can be deduced from the transition between the chirp periods and these are less clean in Fig. 3.8a than they are in Fig. 3.8b. These frequencies will cause spikes to be created in the spectrum. It is not necessarily bad that the chirp of the 100 μs period has a broader chirp, as the power transmitted is on the desired frequencies in the start to stop frequency range. The problem, however, is that in the transitions of the chirps a significant part of the power is lost to other frequencies than those between the start and stop frequency.



(a) Linear chirp (bandwidth : 1.023Mhz, period : 100μs)

(b) Linear chirp (bandwidth : 1.023Mhz, period : 500μs)

Figure 3.8: Different chirp periods

3.5.2 Linear chirp

The mathematical representation of the linear chirp that has been implemented is shown in Eqn. (3.8). Here A is the amplitude, f_{start} is the start frequency, k is the chirp rate (see Eqn. (3.7)) and θ the starting phase of the signal. The shape of the chirp can be modulated by changing these parameters. The most important parameters are the start & stop frequencies and the chirp rate. The chirp amplitude is not of interest here because this can be tuned by setting the gain in the simulator. The starting phase is not important since it does not contribute to the distortion of the GNSS signals. The start frequency, stop frequency and the chirp period can be set in the simulator. The chirp rate, k, is derived from the start and stop frequency in the simulator. This holds for all chirp jamming signals.

$$j_m(t) = A \cdot cos(2\pi(f_{start} + \frac{k}{2}t)t + \theta) \tag{3.8}$$

3.5.3 Symmetric chirp

The symmetric chirp is like the linear chirp. However, when the chirp has reached its stop frequency, it decreases its frequency until it reaches the start frequency and then repeats itself. There is no sudden frequency jump present in the spectrum by sweeping over the frequencies in this way when the jamming signal is transmitted. In Eqn. (3.9) the mathematical representation of the symmetric chirp is given. The first half of the chirp, $j_{m1}(t)$, has the same mathematical representation as the linear chirp. The difference is that the chirp rate, k, is not divided by two. This is because the chirp period is twice that of the linear chirp. Therefore the chirp rate is already twice as low. This is explained using Eqn. (3.7) in Section 3.5.1. The second half of the symmetric chirp, $j_{m2}(t)$, is a bit different from the first half. In the second half of the chirp, the frequency has to start at f_{stop} and the chirp rate is subtracted.

$$\begin{aligned} j_{m1}(t) &= A \cdot cos(2\pi(f_{start} + k \cdot t)t + \theta) \\ j_{m2}(t) &= A \cdot cos(2\pi(f_{stop} - k \cdot t)t + \theta) \end{aligned} \tag{3.9}$$

3.5.4 Quadratic chirp

The chirp rate for the quadratic chirp has been modified compared to the linear and symmetric chirp rate. The chirp rate used for the quadratic chirp can be seen in Eqn. (3.10). This modification is needed for the chirp to reach the stop frequency in time before the end of the chirp repetition interval, see Fig. 3.5. In Eqn. (3.11) the mathematical representation of the chirp is given.

$$k = \frac{f_{stop} - f_{start}}{3 \cdot T^2} \tag{3.10}$$

$$j_m(t) = A \cdot cos(2\pi(f_{start} + k \cdot t^2)t + \theta) \tag{3.11}$$

3.5.5 Double quadratic chirp

The equation of the double quadratic chirp is similar to the one for the quadratic chirp, albeit with a factor 8 in front of the chirp rate. This is required since the chirp repition period, T, of the double quadratic chirp is twice as large as that of the quadratic chirp, see Fig. 3.6. Since it is quadratic the chirp rate, k, is four times smaller. This can be deduced from Eqn. (3.10), if T is two times larger, k, becomes four times smaller, $k = \frac{f_{stop}-f_{start}}{3 \cdot (2T)^2} = \frac{1}{4} \cdot \frac{f_{stop}-f_{start}}{3 \cdot T^2}$. The last factor two comes from the fact that the double quadratic chirp is composed of double the number of chirp sections per period used in the quadratic chirp. In 500 μs, two chirp sections are used, whereas only one chirp section is used with the quadratic chirp in 500 μs. Multiplying these two factors gives a factor of $4 \cdot 2 = 8$ in front of the chirp rate.

As described in Eqn. (3.12) four separate equations where used to describe the different sections of the chirp. This is required since the double quadratic chirp is built out of four separate phases, see Fig. 3.6d. The chirp rate is the same as for the quadratic chirp, see Eqn. (3.10). The bandwidth of the chirp is represented by $f_{bw}= f_{stop} - f_{start}$ and the chirp is built around a center frequency, $f_{cent} = f_{stop} - \frac{f_{bw}}{2}$. The first section of the chirp is $j_{m1}(t)$. It starts at the center frequency, f_{cent}, of the chirp and stops at frequency, f_{stop}. The second section of the chirp, $j_{m2}(t)$ then goes from the stop frequency, where the first section just ended, back to the center frequency. Since the frequency should decrease, the chirp rate, k, is substracted in this chirp section. The third section, $j_{m3}(t)$, is continuous from the center frequency down to the minimum frequency of the chirp, f_{start}. The fourth section, $j_{m4}(t)$, concludes the chirp. It goes from the minimum chirp frequency, f_{start}, to the center frequency of the chirp, f_{cent}. This ensures that when multiple chirps are placed in accession, there are no large frequency jumps in the signal.

The chirp that is shown in Fig. 3.6d has its center frequency at 511.5 kHz and a bandwidth of 511.5 kHz. When the chirp is transmitted and jams a GPS signal, the center frequency is shifted to the center frequency of GPS (1,575.42 MHz) by the simulator.

$$\begin{aligned}
j_{m1}(t) &= A \cdot cos(2\pi(f_{cent} + 8 \cdot k \cdot t^2)t + \theta) \\
j_{m2}(t) &= A \cdot cos(2\pi(f_{stop} - 8 \cdot k \cdot t^2)t + \theta) \\
j_{m3}(t) &= A \cdot cos(2\pi(f_{cent} - 8 \cdot k \cdot t^2)t + \theta) \\
j_{m4}(t) &= A \cdot cos(2\pi(f_{start} + 8 \cdot k \cdot t^2)t + \theta)
\end{aligned} \tag{3.12}$$

4 Validation & simulation of jamming signals

In this chapter, the validation and the Matlab simulations of jamming signals that have been created within the simulator will be discussed. First, Section 4.1 will explain how the validation of the jamming signals is done in Matlab. The different jamming types are then discussed in the following order: CW in Section 4.2, pulsed sine wave in Section 4.3, broadband jamming in Section 4.4 ,the linear chirp in Section 4.5, the symmetric chirp in Section 4.6, the quadratic chirp in Section 4.7 and at last the double quadratic chirp in Section 4.8. After which the comparison of the jamming signals is done in Section 4.9.

A total of seven jamming signals have been implemented. The first implemented jamming signal was the Continuous Wave (CW), which is a jamming signal that is transmitted as a sine wave with a fixed frequency. A variation on this jamming signal is a pulsed sine wave signal, read more on this in Section 3.3. Another jamming type is the broadband signal which is a Gaussian white noise signal, that can be created within a specified bandwidth. In the simulator, four chirp signals have been integrated: linear, symmetric, quadratic and double quadratic chirp signals.

The seven jamming profiles integrated into the simulator can be tuned modularly to the user's preference. This allows for in-depth research to the impact of the jamming signals with different characteristics. All of the characteristics of the jamming profiles are tunable, for the CW this is the center frequency. For the pulsed sine wave the center frequency and duty cycle can be tuned. The characteristic that can be tuned for the broadband and chirp signal are the start and stop frequency and the period of the signal. Apart from the waveform, the duration and gain of the transmission can be selected for all jamming types. The jamming signals will be discussed in more depth in the following sections.

As mentioned in Chapter 1 there are a lot of in-car chirp jammers sold and used, even though they are illegal in many countries. These often are built from relatively simple analog hardware: a 555-timer that feeds a sawtooth to a Voltage-Controlled Oscillator (VCO). Often those VCOs are not very accurate and therefore need a large bandwidth, around 20 MHz [4], to cause enough distortions to the GPS band. The chirps created by these devices often have periods of 10 μs.

The chirps used to evaluate the performance of the two GPS receivers had a bandwidth of 1.023 MHz. This choice was made because from the PSD of the GPS signal it could be seen that most power of the mainlobe is within 1 MHz of the center frequency of GPS. Thus most interference could be introduced if this part of the GPS band would be jammed. The chirp period of the jamming signals was chosen to be 500 μs, in Section 3.5 is explained why this choice was made.

4.1 Simulation setup

The jamming signals created in the simulator, both the in-phase and quadrature-phase, are written to a binary file. This is a file format that can be used to transmit radio signals. The in-phase and quadrature-phase components are represented by an 8-bit number and they are saved by alternately storing an in-phase sample followed by a quadrature sample (I, Q, I, Q, etc.). This file is used for the creation and transmission of radio signals with an SDR. In Matlab, the binary files' validation is done to check whether the binary file contains the correct jamming types. Additionally, the characteristics of the jamming type were checked. For this inspection both the Short-Time Fourier Transform (STFT) and the PSD where used. The STFT was used to check if the chirp rate was right, if the start and stop frequencies were accurate and whether the period of the chirp matched with the desired period. The STFT is explained indepth in Section 4.1.3. The FFT was used to calculate the PSD. From the PSD it was confirmed whether the power in the signal was distributed to the correct frequencies and that no power was assigned to unwanted frequencies since this would not contribute to an effective jamming signal.

Moreover, the chirps' effect on the correlation of the GPS L1 C/A signal was tested. This is done by superimposing the jamming signal onto a GPS signal. Before this is done the output of the binary files, both the jamming files and the simulated GPS files are normalized using peak normalization. The GPS files are binary files too that contain a short scenario with GPS signals of multiple satellites. In order to evaluate the effect of the jamming signals on the GPS signal, a parallel code phase search algorithm [13] was built in Matlab as a correlator for these simulations. The correlation is done with 1 coherent round and 25 non-coherent rounds. The parallel code phase search algorithm, as well as coherent and non-coherent integration are explained in, Section 2.2.1, Section 2.2.2 and Section 2.2.3.

Only one coherent integration round is used to avoid phase changes affecting the integration. In Section 2.2.2, it is explained how these phase changes of the signal could affect the integration. For the non-coherent integration 25 integration rounds were chosen as this provides a good balance between the accuracy of the results and computation power.

The GPS signal that is used in the simulations is a simulated GPS signal from the GPS simulator. The PSD of this signal can be seen in Fig. 4.2a. The signal strength of the GPS signal is -41 dBm. Usually, the GPS signal would be modulated onto a carrier wave of 1,575.42 MHz before it is transmitted. Here the baseband signal modulated around 0 Hz is used.

4.1.1 Power spectral density

The PSD figures have been created by averaging segments of the signal. This method makes it possible to assign power to the correct frequency more accurately and can reduce noise fluctuations. This comes at the cost of having a lower frequency resolution since fewer data points are available for each FFT calculation. The difference between using averaging and not averaging can be seen in A.6 in Fig. A.1.

The algorithm is visualized in Fig. 4.1. The first step is to divide the signal, $s(m)$,

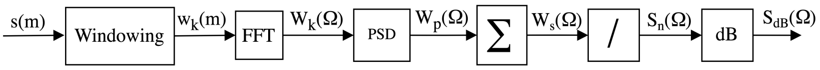


Figure 4.1: Block diagram PSD

into n windows [37]. For this algorithm a rectangular window is used. When using a rectangular window, the contribution of the signal at the start and end of each window is not eliminated. This negative side effect that normally arises when using windowing functions (e.g., Bartlett, Hann, Welch) does not occur now. Thus the windows do not need to overlap, which would normally be required to improve the accuracy of the PSD due to the side effect of windowing [38], see Fig. A.2a in A.7.

$$\begin{aligned}
W_k(\Omega) &= \sum_{m=1}^{N_{samp}} w_k(m) \cdot e^{-j2\pi \cdot (\Omega-1) \cdot \frac{m-1}{N_{samp}}}, 1 \leq \Omega \leq N_{samp} \\
W_p(\Omega) &= \frac{|W_k(\Omega)|^2}{f_{samp} \cdot N_{samps}} \\
W_s(\Omega) &= \sum_{k=0}^{n} W_p(\Omega) \\
S_n(\Omega) &= \frac{W_s(\Omega)}{n} \\
S_{dB}(\Omega) &= 10 \cdot log_{10}(W_n(\Omega))
\end{aligned} \tag{4.1}$$

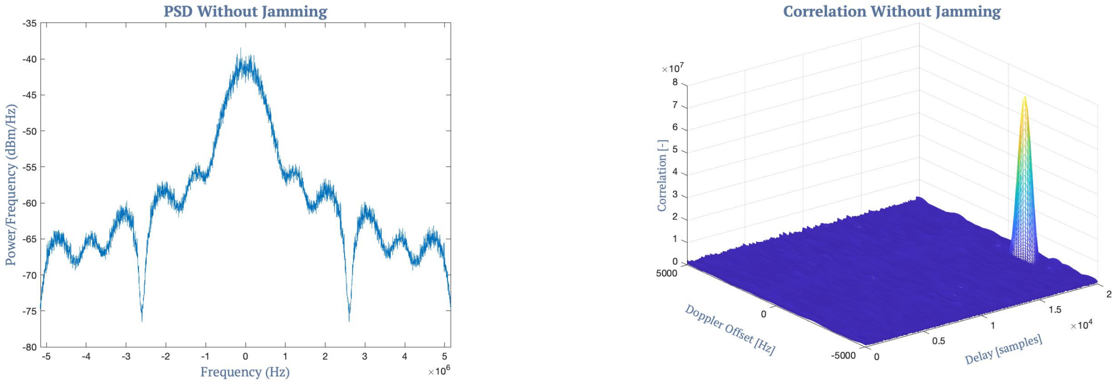
After deviding the signal into n windows, the discrete FFT of each window is computed, see Eqn. (4.1). Each window then consists of N_{samp}= $\frac{N_{s(m)}}{n}$ samples (where $N_{s(m)}$ is the vector length of $s(m)$). After this the PSD, $W_p(\Omega)$, of each FFT is computed. The sampling frequency, f_{samp}, is 20 MHz (the frequency that is used on the HackRF to transmit the signals, see Section 5.1). These PSDs, $W_p(\Omega)$, are then summed, giving $W_s(\Omega)$. The result is divided by n. This division is necessary since summing the PSDs of each window caused the overall gain to be multiplied by the number of windows that were used. The last step is to convert the result to dBs.

4.1.2 Signal-to-Noise Ratio

The distortion effect of the generated jamming signals is evaluated by comparing their effect on the correlation of the GPS signal. The noise introduced in the correlator is measured by using the Signal-to-Noise Ratio (SNR). The correlation peak of the undistorted GPS signal can be seen in Fig. 4.2b. These correlation plots are created by the GPS receiver in order to find the tracking and Doppler offset, this is done by the DLL and the PLL respectively. This has been explained in Section 2.2.1. Once these are found, they are used to track the GPS signal, how the peak is formed can be read in Section 2.2.

This analysis assumes that the receiver will track the largest peak found from the correlator based on the largest correlation value. In order to find the correct location of the correlation peak of the undistorted GPS signal, the simulation is first done without jamming. When jamming is enabled, the signal is assumed to be lost if the correlator no longer finds the authentic correlation peak. This could be because another peak has become higher than the authentic correlation peak or that the correlation peak has been dissolved in the correlation noise generated by the jamming signal. In the latter situation, the correlation peak is accepted as the correct correlation peak if this peak is within half the peak width of the original peak location, i.e., within 19.55 samples. The half-peak width of the correlation peak can be calculated with Eqn. (4.2). In Fig. 4.3 an example is given of how the correlation peak becomes dissolved in the correlation noise. This theory is also briefly discussed in Section 2.3. In Figure 4.3a the correlation peak of the undistorted GPS signal is shown, together with the half peak width as calculated in Eqn. (4.2). From Fig. 4.3b, it can be seen that the correlation noise is increased due to the quadratic chirp jamming. The dashed line shows the authentic correlation peak and the half peak width on both sides. In Fig. 4.3c the correlation noise has become larger than the authentic correlation peak. However, because the peak lies within the half-peak width of the authentic correlation peak, the receiver will accept this peak as the correct correlation peak because the receiver will not be significantly impacted if the "accepted" peak is still within this window. Therefore the receiver is still able to acquire the true correlation peak and therefore track the GPS signal.

Because the GPS signal is known for the simulations, the correct location of the correlation peak is known too. In the subsiding sections, a threshold level is used to determine whether the GPS receiver would still be able to acquire the GPS signal. This threshold is reached when the correlator can no longer find the correlation peak in the correct range within the half-peak width of the authentic correlation peak.



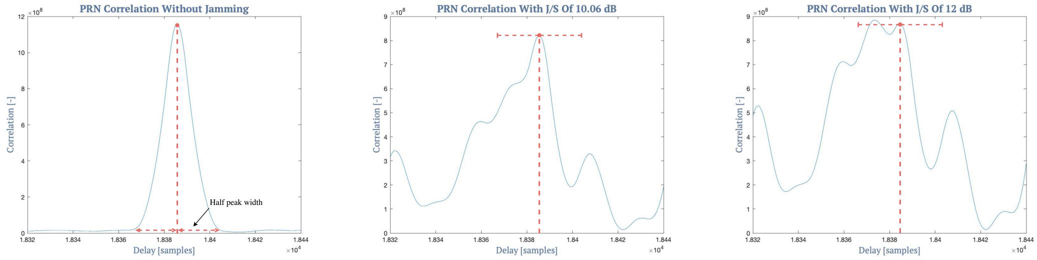
(a) Power spectral density GPS signal

(b) Correlation peak GPS signal

Figure 4.2: Results simulation GPS binary file

$$\text{half peak width} = \frac{\text{sampling frequency}}{\text{PRN chip frequency}} = \frac{20 \cdot 10^6}{1.023 \cdot 10^6} = 19.55 \text{ samples} \tag{4.2}$$

The SNR is calculated using Eqn. (4.3) [39]. The correlator searches for the residual Doppler shift on the GPS signal using frequency bins. Using this approach, the correlator does not have to check for each frequency individually whether it is the correct Doppler shift. Read more about the GPS receiver in Section 2.2.



(a) Undistorted correlation (b) Distorted correlation (c) Fully distorted correlation

Figure 4.3: Behaviour PRN correlation peak under jamming circumstances

The possible Doppler frequencies that the correlator is searching over, in the example shown in Fig. 4.2b this is from -5000 Hz to 5000 Hz, are subdivided into N bins. In Eqn. (4.3), $max(bin_N)$ corresponds to the maximum correlation value in frequency bin N, $mean(bin_N)$ is the mean of that bin and $var(bin_N)$ is the variance in correlation values. The SNR is calculated for every bin and of all SNR_N, the highest value is the SNR value of the signal[i].

$$SNR_N = 10 \cdot log_{10}(\frac{(max(bin_N) - mean(bin_N))^2}{var(bin_N)}) \tag{4.3}$$

4.1.3 Short-Time Fourier Transform

The chirps signals have a high rate of change in frequency. This makes it difficult to use the standard Fourier analysis, see Eqn. (4.4). The Fourier transform decomposes a signal in frequency components and the relative strength is computed of each frequency component. However, this approach does not provide information about the timing of the particular frequency characteristics since the Dirichlet kernel, $e^{j\Omega t}$, is evenly distributed in time.

$$F(\Omega) = \int_{-\infty}^{\infty} f(t)e^{-j\Omega t}dt \quad \leftrightarrow \quad f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\Omega)e^{j\Omega t}d\Omega \tag{4.4}$$

The STFT, see Eqn. (4.5), moves a fixed window size over the time function, $f(t)$. From each window interval, the frequency content is extracted. The STFT can be used very well for nonstationary signals (e.g., speech or music signals) since the STFT can map the frequency accurately to the time for signals which have frequency components that vary over time. The FFT is not suited for this as it sweeps evenly over the time axis, not taking short-duration bursts of the signal into account.

$$F(\Omega, t) = \int_{-\infty}^{\infty} f(t)g^*(t-\tau)e^{-j\Omega t}dt \tag{4.5}$$

The STFT slides a window, $g(t)$, over the time axis, calculating the Fourier transform within the given window, thereby allocating the frequencies present in that window

[i] This only holds if the GPS signal is not dissolved under noise

to a time point, τ. Therefore the STFT is very useful to check whether the created chirp signals have the right start and stop frequency, as well as checking if the chirps have the correct chirp period.

However, the STFT is only able to provide a suboptimal tradeoff between the time and frequency resolution. This is due to the frequency resolution being the same for all locations of the spectrogram.

The Kaiser window [40] is used as it performs best in terms of main-lobe width [41]. In Eqn. (4.6), the equation used for computing the components of the Kaiser window is given. Here I_0 is the zeroth order modified Bessel function of the first kind [40]. There is a tradeoff between the width of the main lobe and its sidelobes. When β is one the main lobe of the Kaiser window is relatively small and has a sharp transition between its side lobes. However, when β is increased the width of the main lobe grows and the sidelobes gradually disappear.

For the analysis, the values used for the window length was, $L = N + 1 = 256$ sample points and $\beta = 5$. This window size was optimal since a larger window size would have led to a lower resolution and a smaller window size would require too much computational power. A lower value for β would have led to higher amplitudes of the sidelobes, resulting in power not being allocated to the correct frequencies.

$$g(n) = \frac{I_0(\beta\sqrt{1-(\frac{n-N/2}{N/2})^2})}{I_0(\beta)}, 0 \leq n \leq N \tag{4.6}$$

For the STFT, the total number of discrete Fourier transform points per window was 512. As explained in Section 4.1, windowing eliminates the contribution of the signal at the start and end of each window. This negative side effect can be resolved by having subsequent windows overlap [38], see Fig. A.2b in A.7. The optimum found for the overlap length of each window was 220 data points. There was not enough compensation for the windowing effect with a lower overlap value and with a higher value the required computation power became too high.

4.2 Continuous wave single tone

The continuous wave jamming form only has one parameter that can be changed to alter its waveform, namely its center frequency. For the simulations and the practical evaluation, this frequency was set on the center frequency of GPS, 1,575.42 MHz. Since most of the power of the GPS signal resides on this frequency, jamming on this frequency will introduce the most interference in the correlator of the GPS receiver. For the validation of the CW jamming signal, a CW signal with a frequency of 1 MHz was written to a binary file. This file was analyzed in Matlab and from the STFT shown in Fig. 4.4 it can be seen that the signal indeed contains the 1 MHz signal. Using a correlator in Matlab a simulation was done to see at which J/S ratio the CW can make the correlator lose track of a GPS signal. This resulted in a J/S ratio of 29.65 dB, which is relatively high. This will be elaborated on in Section 4.9.

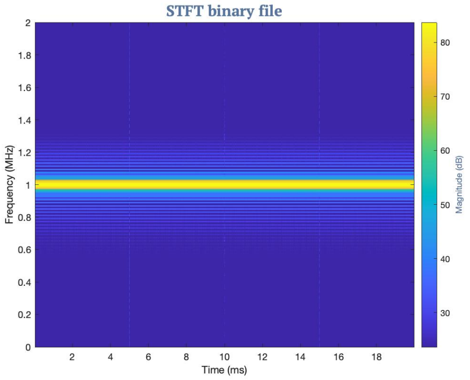


Figure 4.4: Continuous wave

4.3 Pulsed sine wave

For the simulations and the evaluation a duty cycle and period of the pulsed sine wave were chosen that have the highest chance of matching the time constant of the AGC. This is since the value of the time constant varies for different receiver designs, ranging from around 1-2 to 10 ms. A period for the pulsed sine wave signal was chosen in the middle with a period duration of 5 ms and a duty cycle of 50%. This signal was created within the simulator at a frequency of 1 MHz and written to a binary file. This binary file was validated in Matlab by use of the STFT. From Fig. 4.5 it can be seen that the signal indeed has a frequency of 1 MHz, a period of 5 ms and a duty cyle of 50 %. The pulsed sine wave was able to make the correlator lose track of the GPS signal at a J/S ratio of 12.08 dB.

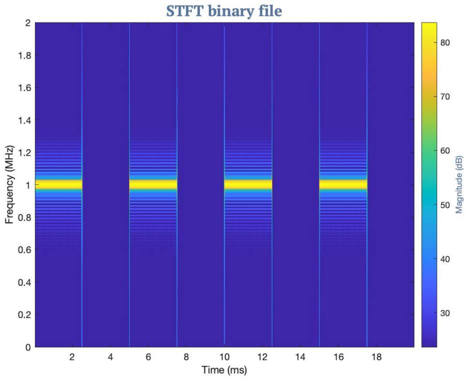


Figure 4.5: Pulsed sine wave (period: 5 ms, duty cycle: 50%)

4.4 Broadband jamming

The parameters of the broadband jamming signal that can be modified are the bandwidth and the carrier frequency of the signal. The broadband jamming signal was chosen to have a bandwidth of 1.023 MHz and a carrier frequency of 1,575.42 MHz. This bandwidth was chosen because most of the power of the simulated GPS signal resides in this area, see Fig. 4.2a. A recording of the signal, created by the simulator, was made and validated in Matlab by using the STFT [42]. As can be seen in Fig. 4.6 the stopband starts at 1 MHz.

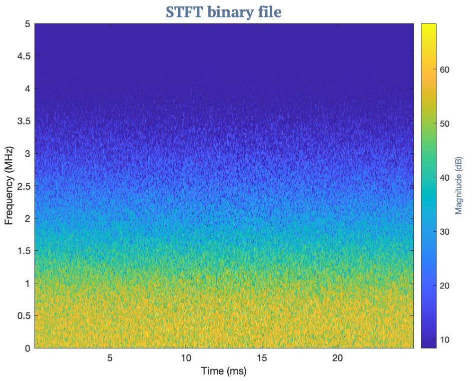


Figure 4.6: Broadband jamming (bandwidth : 1.023 MHz)

In Fig. 4.7 the result of superposing the broadband jamming signal onto the GPS signal can be seen. In Fig. 4.7a the combined PSD of the GPS and broadband jamming signal is shown. From this figure can be seen that the signal does not have a roll-off that is as sharp as the chirp signals have, see the PSD of the linear chirp in Fig. 4.8a. The

The SNR of the GPS signal is shown against the J/S ratio in Fig. 4.7b. The dashed line shows the tracking threshold. From this figure can be seen that the receiver starts to lose track of the GPS signal at a J/S ratio of 8.23 dB. The correlation plot in Fig. 4.7c shows the correlation at threshold jamming. This is the level from which the correlation peak of the GPS signal is no longer distinctive. The correlation figure shows that the entire noise floor of the correlation plot rises. This is caused by the RSS operation of the non-coherent integration, see Section 2.2.3.

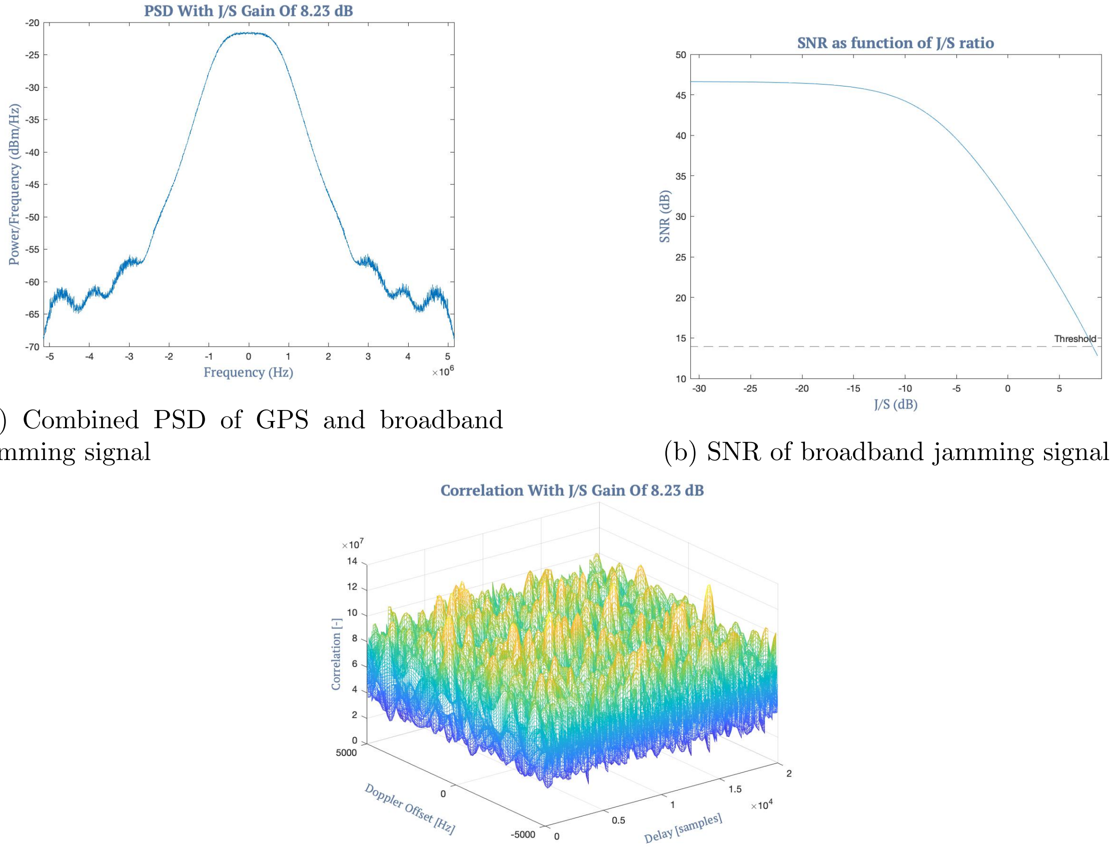
4.5 Linear chirp

The bandwidth and period of the chirps created by the simulator can be chosen arbitrarily by the user. All chirp signals were validated with a bandwidth of 1.023 MHz, a period of 500 μs and a carrier frequency of 1,575.42 MHz. The chirp was validated in Matlab by using a STFT. The STFT showed that the simulator made the chirps with the characteristics that were given as input, see Fig. 3.6a. As can be seen from the figure, the chirp indeed has a bandwidth of 1.023 MHz and a period of 500 μs.

In Fig. 3.6a, a replica is visible that lies just above the linear chirp. This replica could be the result of from having a repetition period of the chirp that is too high. If the chirp period is increased, this replica's presence becomes less noticeable.

The result of superposing the linear chirp on the simulated GPS signal can be seen in Fig. 4.8.The presence of the linear jamming signal can be seen in the PSD shown in Fig. 4.8a.

In Fig. 4.8b, the SNR is shown against the J/S ratio. The figure shows that a J/S of 0.93 dB is the threshold at which the receiver starts to lose track of the GPS signal. The undistorted GPS signal's maximum power was -41 dBm, see Fig. 4.2a. The linear chirp has a power here of -40.07 dB, which is 0.93 dB stronger than the simulated GPS signal, thus resulting in a J/S of 0.93 (see Eqn. (3.1)). However, before this threshold



(a) Combined PSD of GPS and broadband jamming signal

(b) SNR of broadband jamming signal

(c) Threshold J/S ratio

Figure 4.7: Simulation broadband jamming signal

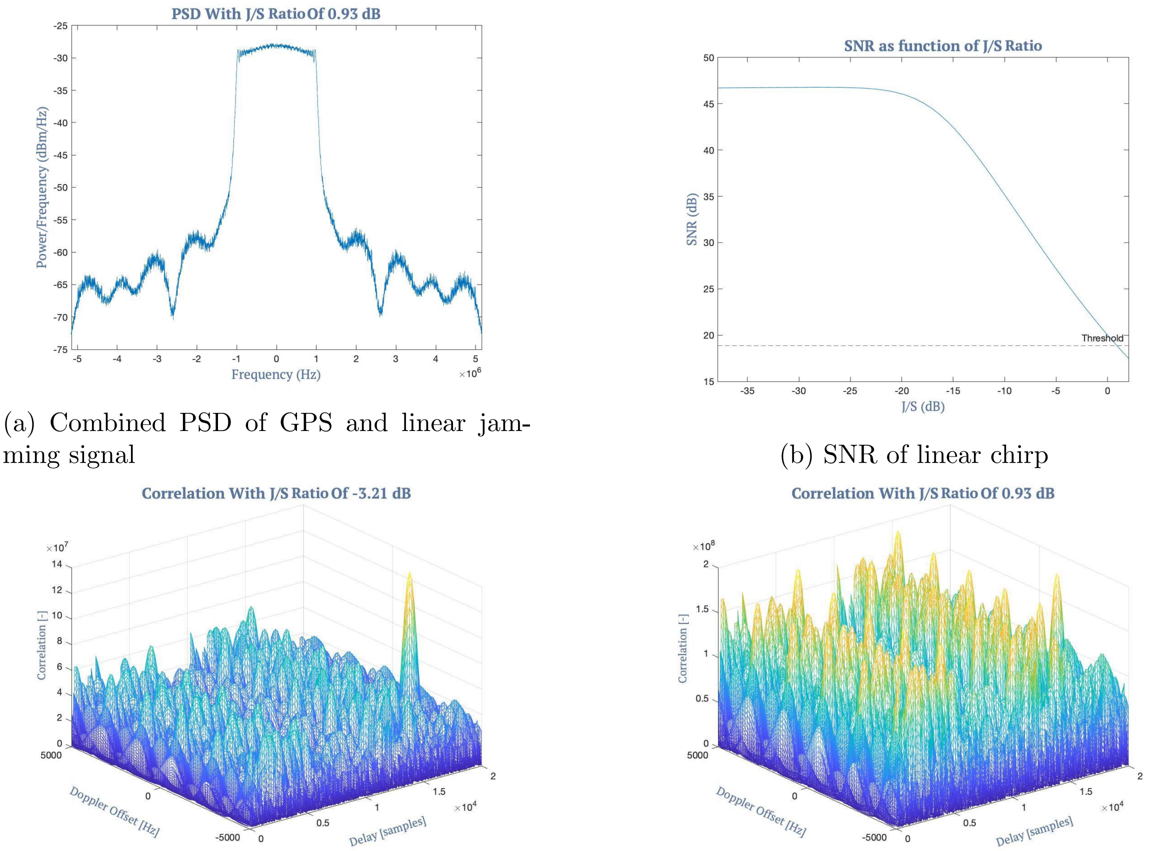
is reached, distortions are already present within the correlator of the receiver, as can be seen in Fig. 4.8c. Though these distortions are not enough for the receiver to lose track of the GPS signal, they can reduce the accuracy of the location estimation due to distortions in the correlation peak [43], as shown in Fig. 4.3. When the threshold jamming level is reached, the authentic correlation peak is no longer distinctive in the correlation plot, see Fig. 4.8d.

4.6 Symmetric chirp

The symmetric chirp is validated using the same procedure as for the linear chirp. From Fig. 3.6b it can be seen that the chirp has the desired behavior and was created correctly within the simulator.

The superposition of the symmetric chirp onto the GPS signal can be seen in Fig. 4.9. In Fig. 4.9a and Fig. 4.9c the effect of the symmetric jamming signal on the GPS signal is shown.

From Fig. 4.9c it can be seen that the distortion effect of the symmetric jamming signal is not as evenly spread as for the linear chirp. There are peaks in the correlation



(a) Combined PSD of GPS and linear jamming signal

(b) SNR of linear chirp

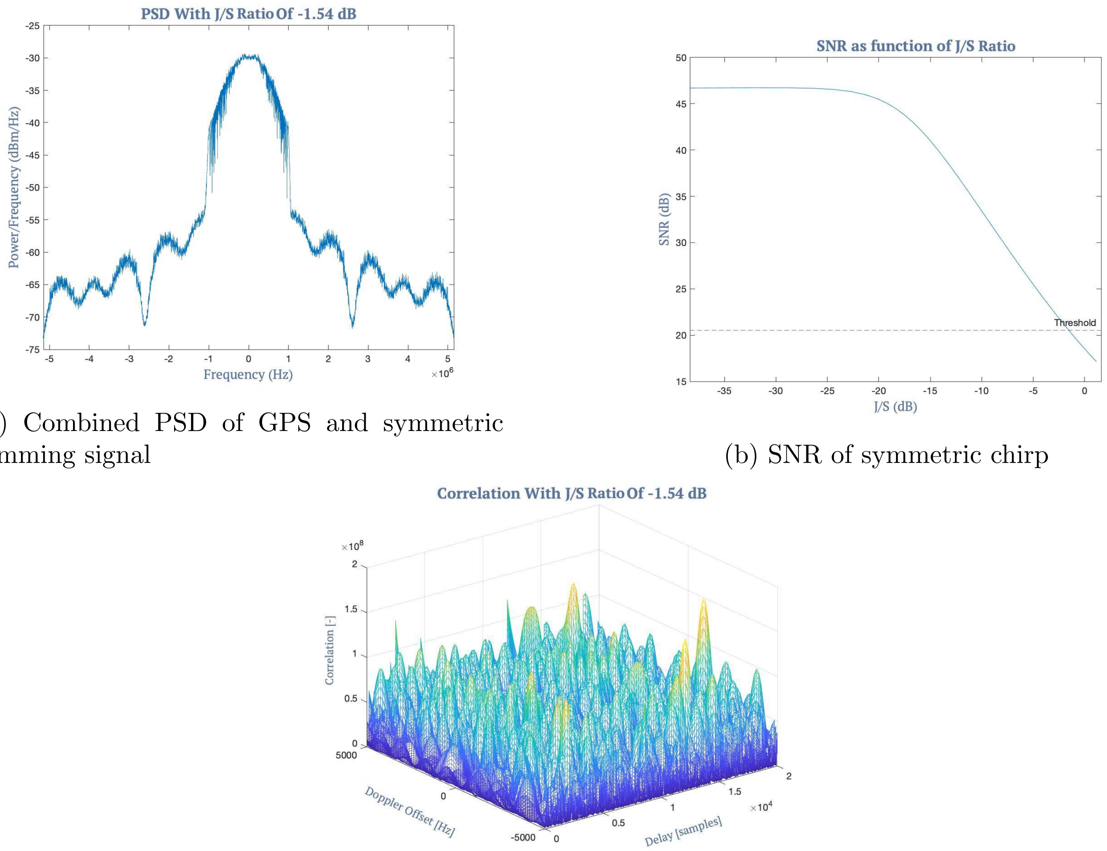
(c) Distortion correlation

(d) Threshold J/S ratio

Figure 4.8: Simulation linear chirp

figure that have higher correlation levels than the average noise introduced in the correlator. From the PSD of the symmetric chirp in Fig. 4.9a it can be seen that the power of the chirp has more outliers than the linear chirp has in its PSD. The outliers in the PSD of the symmetric chirp support the presence of the spikes that arise within the correlator due to the symmetric chirp. This has to do with the fact that on specific frequencies there is relatively more power present than on others, which makes the correlation of the GPS signal more random. Therefore, these frequencies will have a higher correlation value, creating more significant peaks within the correlator. The PSD of the linear chirp, see Fig. 4.8a, has its power distributed more evenly around the center frequency, which could explain why the noise created by the linear chirp within the correlator is distributed more evenly.

In Fig. 4.9b it can be seen that the threshold J/S ratio for the symmetric chirp is -1.54 dB, which is just a bit lower than for the linear chirp. This means that a little less power is needed to cause the correlation of the receiver to drop below the tracking threshold as is needed when the linear chirp is used.



(a) Combined PSD of GPS and symmetric jamming signal

(b) SNR of symmetric chirp

(c) Threshold J/S ratio

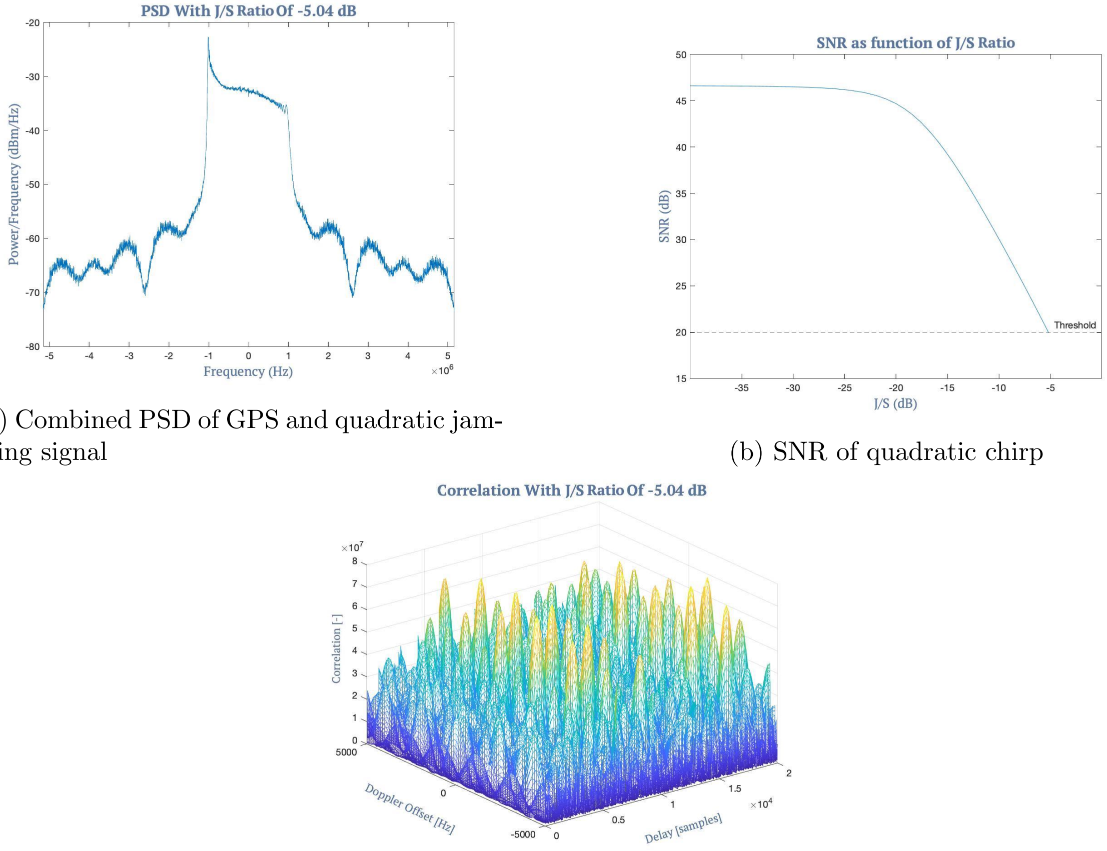
Figure 4.9: Simulation symmetric chirp

4.7 Quadratic chirp

From Fig. 3.6c it can be seen that the chirp has the desired behavior. The signal starts at frequency, f_{start} : 0 Hz, and stops at frequency, f_{stop} : 1.023 MHz.

The power in the quadratic chirp is not as evenly spread as in the linear and symmetric chirps. This can be seen in Fig. 4.10a. This is because the signal is longer present at the low frequencies of the chirp, which also results in more power being transmitted on these frequencies. The correlation plot of the quadratic chirp can be seen in Fig. 4.10c. The quadratic chirp introduces a couple of large spikes within the correlator. Most likely, this is because the chirp transmits more power on the lower frequencies. This power then transfers into the correlator.

According to the simulations done in Matlab, the quadratic chirp can cause more distortions within the correlator than the linear and symmetric chirp do. The threshold J/S ratio is reached at -5.04 dB. This can be seen in Fig. 4.10b.



(a) Combined PSD of GPS and quadratic jamming signal

(b) SNR of quadratic chirp

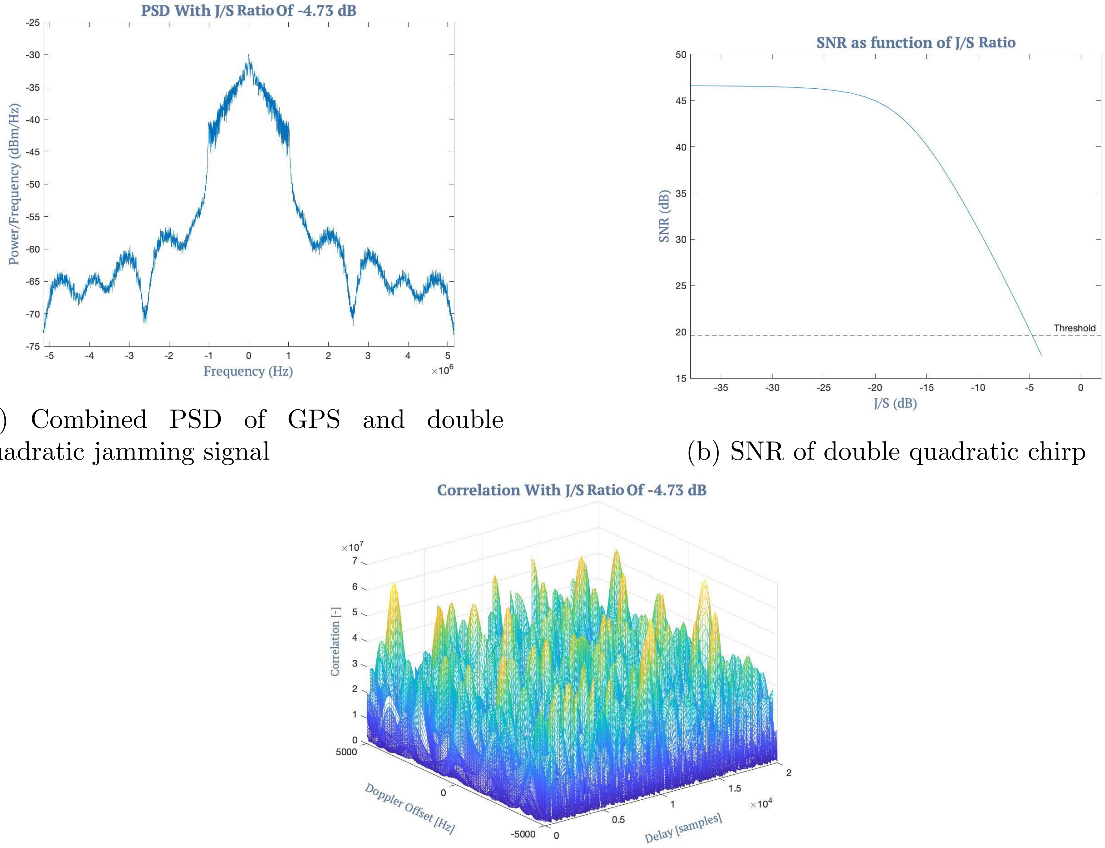
(c) Threshold J/S ratio

Figure 4.10: Simulation quadratic chirp

4.8 Double quadratic chirp

The double quadratic chirp was created because the expectation was that the quadratic chirp would not be very efficient at introducing noise into GPS receivers. This was based on the PSD figures that were created in Matlab. The expectation is that the double quadratic chirp will distort the GPS signal the best. The signal transmits longer on the center frequency of GPS, as the quadratic chirp does, which distorts the GPS signal more. The form of the double quadratic chirp can be seen in Fig. 3.6d. The signal starts at the center frequency, here 511.5 Hz, and has a bandwidth of 1.023 MHz.

The PSD, the SNR and the correlation plot (at threshold jamming power) of the jamming signal can be seen in Fig. 4.11a, Fig. 4.11b and Fig. 4.11c respectively. In Fig. 4.11a it can be seen that most of the jamming power is indeed transmitted around the center frequency. In Fig. 4.11b the SNR is plotted against the J/S ratio. At a J/S ratio of -4.73 dB the receiver is likely unable to track the GPS signal.



(a) Combined PSD of GPS and double quadratic jamming signal

(b) SNR of double quadratic chirp

(c) Threshold J/S ratio

Figure 4.11: Simulation double quadratic chirp

4.9 Comparison of jamming signals

The J/S threshold values at which the GPS receiver loses track with each of the discussed jamming signals are shown in table 4.1. The simulations show that the difference between the chirp signals and the CW is significant. In comparison, the difference between the chirp signals is relatively small.

The jamming signal that requires the most power to make the receiver lose track of the GPS signal is the CW transmitted on the center frequency of GPS. This requires a jamming to signal ratio of 29.65 dB. This power is significantly higher than that of the other signals. The expectation is that the correlator that was build in Matlab is not able to handle this jamming well and it will most likely have a better performance on jamming the two receivers.

The pulsed sine wave signal requires considerably less power than the CW signal. The pulsed jamming signal can distort the correlation of the GPS signal already at a J/S ratio of 12.08 dB. This jamming signal transmits power only 50% of the time while introducing noise into the correlation figure and saving on transmission power as the pulsed jamming signal can distort the AGC when the time constants are matched.

The chirp jamming signals show relatively minor differences between the threshold J/S ratios. The linear chirp performs worst with a J/S threshold of 0.93 dB, after which the symmetric chirp follows with a J/S ratio of -1.54 dB. There is no clear explanation for why the symmetric chirp performs better than the linear one. The double quadratic chirp performs best. This is most likely because this chirp transmits more power near the center frequency of GPS, causing more distortion to the GPS signal and hence the correlation figure. The CW too transmits on the center frequency of GPS. However, the CW does not jam the frequencies arround the center frequency. Therefore it causes less interference to the GPS signal.

The performance of the broadband jamming signal lays in between that of the chirp signals and the continuous wave. It needs a higher jamming power to make the correlator lose track of the GPS signal than the chirp signals do. This is most likely because the chirp signals can make the correlator lose track of the GPS signal by creating a couple of large peaks within the correlation figure. Whereas, the broadband jamming signal needs to raise the entire noise floor to the same level as the height of the correlation peak of the GPS signal has.

Remarkable is that the performance of the quadratic chirp is relatively high. Since much of the power transmitted by the quadratic chirp is transmitted besides the center frequency, the expectation would be that this signal has a smaller interference with the GPS signal. In Fig. 4.12 the SNR level of the four chirps is plotted against the J/S ratio. The CW and pulsed sine wave signal have not been included in this figure.

Table 4.1: J/S ratios jamming signals.

Jamming signal	**Threshold J/S ratio (dB)**
Continuous wave	29.65
Pulsed sine wave	12.08
Broadband signal	8.23
Linear chirp	0.93
Symmetric chirp	-1.54
Quadratic chirp	-5.04
Double quadratic chirp	-4.73

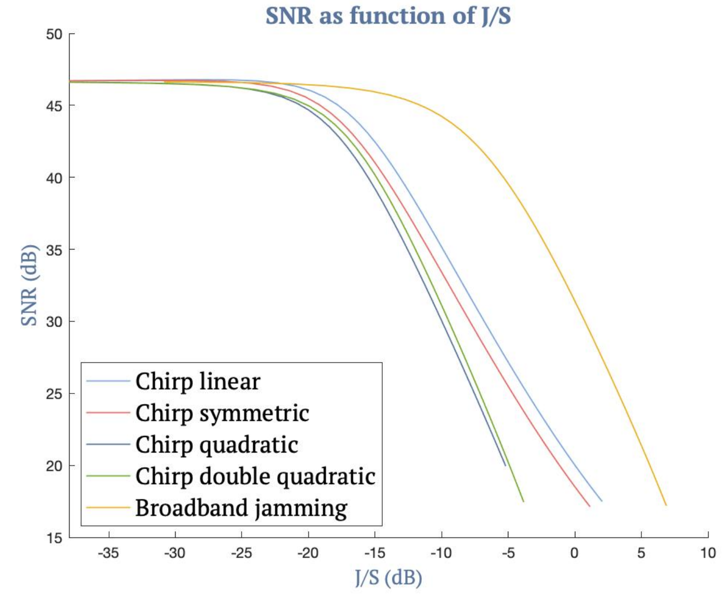


Figure 4.12: Combined SNRs of chirp signals

Practical evaluation of jamming signals 5

In this chapter the effectiveness of the jamming signals will be evaluated on the performance of two different GNSS receivers. To measure the impact of the jamming signal on a receiver the Carrier-to-Noise ratio (C/N0) is tracked while the J/S ratio is gradually increased, by transmitting the jamming signals with a higher gain. First, the test setup will be explained in Section 5.1. Hereafter the found results are discussed in Section 5.2 and in the final Section 5.3 the difference in performance between both receivers is discussed.

5.1 Test setup

Two different types of receivers are used. The first one is a time-worn GNSS receiver that is about a decade old with a cost of around €50. The second receiver is a cutting-edge GNSS receiver of around €1200. The cutting-edge receiver has a built-in Interference Mitigation (IM) option. The evaluation was done with and without IM turned on.

All of the seven jamming profiles that where discussed in Section 4 have been evaluated for both receivers.

The setup of the test can be seen in Fig. 5.1. The GPS signal is transmitted by a LabSat 3 Wideband and the jamming signal is transmitted by a HackRF One. These two signals are combined by a GPS210 signal splitter and then fed into the receiver. This splitter can also be used as a combiner. It has a 200 Ohm internal load on one of the input ports, providing a simulated antenna current drain preventing receiver faulting. This does not affect the jamming signal passed through this port as it is used to filter DC components from the signal and the jamming signal does not contain a DC component.

The GPS signal used in the tests is the L1 C/A signal. A simulation file of the constellation of 1 January 2022 at 0:00 has been made. Therefore, for every one of the tests, the same scenario can be used. This scenario is transmitted by the LabSat 3 at -98 dBm. The signal strength of the signal is adapted in the LabSat by raising the noise floor and not by adding attenuation to the signal [44].

On the HackRF, the highest sampling frequency is used. This is 20 MHz so that the highest possible accuracy of the jamming signals can be obtained. The HackRF has an amplifier with a gain that can be set from 0 to 47 dB in steps of 1 dB. The lowest output power that the HackRF can generate is -32 dBm. This is with an amplification gain of 0 dB. When the jamming signal is transmitted from the HackRF with an output power of -32 dBm, the jamming signal is too strong to be fed directly into the combiner. With this gain, the distortion introduced in the receiver is already too high, making it unable to track the GPS signal and the decay of the C/N0. Therefore, attenuators are placed between the HackRF and the combiner so that the course of the C/N0 ratio can be followed. While running tests for the final performance evaluation, it was found that to track the decay of the C/N0 ratio, at least 59 dB of attenuation of the jamming signal was required since the jamming power would otherwise be too strong. An attenuation of 63 dB was chosen to show that the C/N0 remained unchanged when the gain with which it is jammed was too low.

$$\begin{aligned} J/S_{low} &= (-32-63) - -98 = 3\text{dB} \\ J/S_{high} &= (15-33) - -98 = 80\text{dB} \end{aligned} \tag{5.1}$$

The amplification range, of 47 dB, of the HackRF's amplifier, is not wide enough to evaluate the receivers. Therefore, the choice was made to remove one attenuator of 30 dB halfway during the test to create a more extensive reach with which the receivers could be jammed. The first half of the test is performed by increasing the HackRF's amplifier gain from 0 to 29 dB with an attenuator of 63 dB. Then the attenuator is changed to 33 dB and the HackRF passes through the gains 0 to 47 dB. This gives a total range of 77 dB, which is enough to track the changing C/N0 level of the GNSS receivers. The minimum and maximum achievable J/S ratio with this setup is shown in Eqn. (5.1). Here, for the low J/S_{low} ratio calculation, the -32 dBm is the minimum transmit power of the HackRF, the -63 dB results from the attenuator and -98 dBm is the set power of the LabSat. For the calculation of the high J/S_{high} ratio, the maximum transmit power of the HackRF is 15 dBm, an attenuator of 33 dB is used and the transmit power of the LabSat remains -98 dBm.

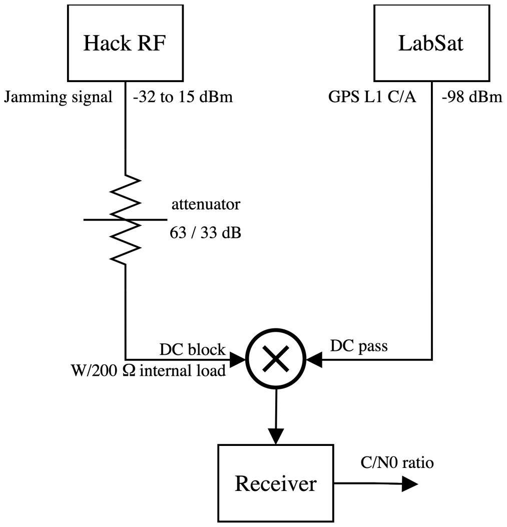


Figure 5.1: Test setup evaluation jamming types

The cutting-edge receiver calculates the C/N0 as shown in Eqn. (5.2), where P_{rec} is the received GNSS signal power in dBW, T_{ant} is the antenna noise temperature in Kelvin (typically 130 K), k_b is Boltzmann's constant and NF_{sys} is the noise figure of the system, shown in Eqn. (5.3). In this equation, NF_{amp} is the noise figure of the low noise amplifier, NF_{rx} is the noise figure of the receiver module and G_{preamp} is the netto pre-amplification, all in dB.

It is not precisely known how the time-worn receiver calculates the C/N0 ratio.

For logging the C/N0 level for the evaluation, NMEA files are used [45]. A GNSS receiver needs a C/N0 ratio of about 20 dB to be able to track a GNSS signal [30]. A receiver may lose track of the GNSS signal before reaching this threshold.

$$C/N0 = P_{rec} - 10 \cdot log_{10}(T_{ant} + 290 \cdot (10^{NF_{sys}/10} - 1)) - 10 \cdot log_{10}(k_b) \tag{5.2}$$

$$NF_{sys} = 10 \cdot log_{10}(10^{NF_{amp}/10} + \frac{10^{NF_{rx}/10} - 1}{10G_{preamp}/10}) \tag{5.3}$$

5.2 Results

In this section, the effects of the different jamming profiles on the C/N0 level are discussed for the two GNSS receivers. First, the results of the time-worn receiver will be shown in Section 5.2.1, followed by the results of the cutting-edge receiver, in Section 5.2.2.

5.2.1 Time-worn GNSS receiver

The findings of the jamming profiles on the time-worn GNSS receiver are shown in Fig. 5.2. In Table 5.1 per jamming profile the J/S ratios are shown at which the receiver lost track of the GPS signal. A clear distinction can be made between the course of the CW, pulsed sine wave and the broadband jamming signal oposite to the chirp signals from the figure. However, this does not seem to affect at which J/S ratio the receiver loses track of the GPS signal, except for the broadband jamming signal.

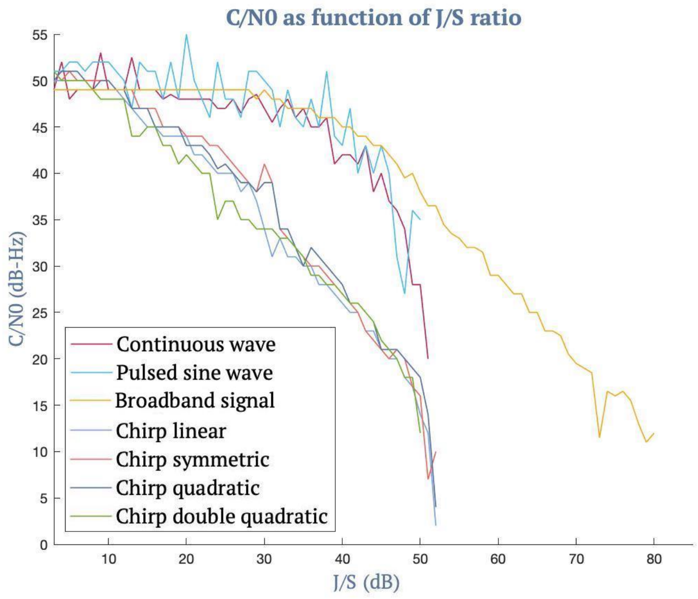


Figure 5.2: Results time-worn receiver

The reason that the CW and the pulsed sine wave jamming profiles have higher C/N0 values than the chirp signals do, is because the time-worn receiver has a mitigation technique that filters significant spikes from the spectrum. This, however, does not seem to affect the J/S ratio at which the receiver loses track of the GPS signal.

Another observation that can be made from Fig. 5.2 is that the C/N0 course of the pulsed sine wave profile is much more spiked than the other profiles. This could be the result of the AGC trying to compensate for the pulsed jamming. However, by doing so, it sometimes adjusts the gain too much and it is not fast enough to correct this in time. Therefore the gain for the incoming signal is sometimes too high, resulting in a spiked C/N0 plot.

Table 5.1: J/S ratios jamming profiles at loss of lock on time-worn receiver.

Jamming signal	**J/S ratio (dB)**
Continuous wave	51
Pulsed sine wave	50
Broadband signal	80
Linear chirp	52
Symmetric chirp	52
Quadratic chirp	52
Double quadratic chirp	50

According to [46] the AGC can help the receiver to obtain better results when it is distorted by pulsed jamming and this effect decreases as the pulsed jamming becomes CW jamming (i.e., the duty cycle is increased). The results in Fig. 5.2 show that for this receiver, the AGC can help the receiver obtain a better C/N0 ratio for the pulsed profile than it is for a CW jamming profile.

The CW and the pulsed jamming profiles do not reach the low C/N0 ratios that the broadband and chirp signals attain. However, it can be seen that the profiles make a sharp dive around a J/S ratio of 51 dB. It could be that the lower C/N0 values cannot be calculated for the CW since the step size of 1 dB, with which the jamming gain is increased, is too large. The jump in C/N0 ratio around a J/S ratio of 48 dB that the pulsed profile makes, is most likely caused by the compensation of the AGC. Even though the calculated C/N0 ratio of the CW and pulsed sine wave are still relatively high, the caused distortion is too high and the GPS signal is lost.

The broadband jamming signal needs a significant amount of power more than the other jamming signals do to make the receiver lose track of the GPS signal. It requires a J/S ratio of 80 dB to make the time-worn receiver lose track of the GPS signal. This can be caused by two reasons, or a combination of both. As broadband jamming signal is first created within the simulator and covers a bandwidth of 10 MHz, it is filtered by a bandpass filter to target the desired bandwidth. However, because only 1 MHz of the bandwidth is transmitted a large part of the signal is filtered. Therefore, the broadband signal that is outputted from the HackRF does not contain as much power as the other jamming signals do. Another reason the broadband jamming signal requires a higher J/S ratio before the GPS receiver loses track of the GPS signal is that where the other jam signals need to introduce a couple large correlation peaks within the correlator to cause distortions, the broadband jamming signal needs to raise the entire noise floor to the same height that the correlation peak of the GPS signal has.

The chirp jamming profiles all follow the same course and obtain the same results. However, the profile that performs best is the double quadratic chirp, which can make the receiver lose track of the GPS signal with the lowest J/S ratio, which is 50 dB. This is in agreement with the results found in the Matlab simulations as here the double quadratic chirp made the receiver lose track of the GPS signal with the lowest J/S ratio too, see Section 4.9.

5.2.2 Cutting-edge GNSS receiver

In this section, the results of the cutting-edge GNSS receiver are discussed. The C/N0 ratios of the receiver against the J/S ratio are shown in Fig. 5.3. One clear distinction that stands out is that the course of almost all the jamming profiles is about the same when IM is turned off, except the broadband jamming signal. The J/S ratios at which the receiver loses track of the GPS signal are shown in Table 5.2a. From this table it becomes evident that the CW and the pulsed sine wave are the most effective. This could have to do with the fact that these signals transmit all of their power on the center frequency of GPS, at which the most information resides. In contrast, the chirp profiles have a more spread distribution of their jamming power.

All of the jamming profiles make the receiver lose track of the GPS signal at a C/N0 ratio of about 20 dB, as discussed in Section 5.1 and in [30].

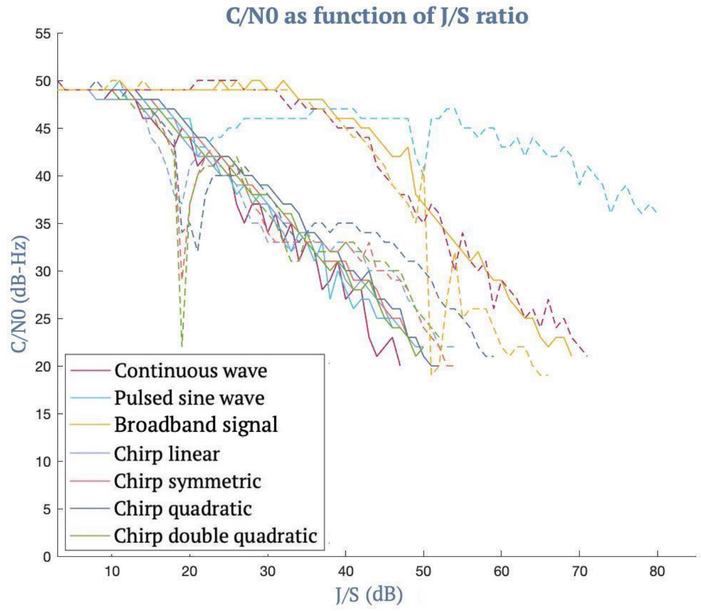


Figure 5.3: Combined results cutting-edge receiver (continuous line: without IM, dashed line: with IM)

The results of the cutting-edge receiver with IM turned on showed more interesting results, see the dashed lines in Fig. 5.3. The chirp profiles have a steep decrease in the C/N0 level around a J/S ratio of 20 dB. At this point, the jamming is detected by the receiver, and the IM is activated. The same holds for the broadband jamming signal at a J/S ratio of 50 dB. In the case of the chirp signals, the distortion is filtered by using wideband interference mitigation. This sharp decrease could be caused by the fact that too much of the spectrum is removed, the interference as well as the GPS signal, causing the C/N0 to fall. The pulsed profile too has this dive, although it is less visible. As for the CW profile, this dive is nonexistent. The CW jamming profile is filtered by a notch filter, which is a different filter than the chirp profiles are filtered with. This could explain the absence of the dive in the C/N0 ratio of the CW jamming profile.

Table 5.2: J/S ratios jamming profiles at loss of lock on cutting-edge receiver.

(a) Without interference mitigation.

Jamming signal	**J/S ratio (dB)**
Continuous wave	47
Pulsed sine wave	47
Broadband signal	69
Linear chirp	50
Symmetric chirp	50
Quadratic chirp	52
Double quadratic chirp	51

(b) With interference mitigation.

Jamming signal	**J/S ratio (dB)**
Continuous wave	71
Pulsed sine wave	n.a.
Broadband signal	66
Linear chirp	54
Symmetric chirp	54
Quadratic chirp	61
Double quadratic chirp	53

The IM of the receiver is very effective against the CW and the pulsed profiles. The notch filter can stall the tracking loss of the GPS signal to a J/S ratio of 70 dB, which means that 24 dB more jamming power is needed to make the receiver lose track of the GPS signal. The IM is most effective against the pulsed profile. The range used for the tests was not large enough to make the receiver lose track of the GPS signal in pulsed sine wave jamming. The jamming level could be increased by another 33 dB by removing the last attenuators, giving a maximum output power of 15 dBm. However, this is beyond the scope of this thesis, and due to timing restrictions, this test was not performed.

From Fig. 5.3 it can be seen that the broadband jamming signal requires a lower J/S ratio when the IM is turned on. The IM will most likely use wideband interference mitigation to try and remove the broadband jamming from the incoming signal. However, because the broadband jamming is pseudorandom and covers most of the GPS spectrum it cannot be filtered well. The IM might be able to remove a part of the jamming signal but at the same time it also removes part of the GPS signal. The removal of part of the broadband jamming and the GPS signal does not improve the C/N0.

The jamming profile that performs best under the circumstance with IM turned on is the double quadratic chirp, see Table 5.2b. This is most likely because the form of this chirp can inflict the most distortion to the GPS signal due to its chirp form. Making it transmit power primarily on and around the center frequency of GPS, see Fig. 4.11a.

When comparing the results of the chirp signals shown in Table 5.2a and 5.2b. It can be seen that the wideband IM of the receiver is quite effective too. With IM turned on, the chirp signals need 4,75 dB more power on average in order to make the receiver lose track of the GPS signal.

Fig. 5.3, clearly showed the difference in gain between the IM turned on or off. It shows it remains tough to filter broadband and chirp signals from the spectrum [35], as opposed to the obtained gain in the case with the CW profiles.

5.3 Receiver comparison

When the results of both receivers are compared, it is interesting to see that the time-worn receiver is more resilient to the jamming than the cutting-edge receiver without IM being turned on. Sometime after the tests were performed, it was found that the cutting-edge receiver was running on old firmware. This could be a reason why the time-worn receiver has a better performance. Unfortunately, the tests could not be repeated with updated firmware on the cutting-edge receiver due to the restricted time.

It could be that the time-worn receiver performs worse for other jamming bandwidths or periods since only one type of bandwidth and period was tested for this thesis. The effect of turning on the IM in the cutting-edge receiver is visible in the results. In that case, the cutting-edge receiver undoubtedly outperforms the time-worn receiver, especially for the CW profiles.

The results differ from the simulations. This is most likely because, in Matlab, the simulations were only performed on the correlator block of a GPS receiver, whereas these tests were performed on the complete GPS receiver's design. Here other components than the correlator influence the results. Such as the AGC, and the quantization by the ADC. Since if the gain is selected wrong by the AGC or the quantization is affected by the distortions of the jamming signal, further baseband processing becomes harder. Another aspect that differed from the simulations is that here the C/N0 ratios were used while the simulations used the SNR.

6 State-of-the-art spoofing methods for GNSS signals

This chapter will discuss some state-of-the-art spoofing methods. A GNSS spoofing attack is the deliberate transmission of manipulated GNSS signals that may lead to an incorrect position, velocity or time derivations in a receiver. An illustration of what a spoofing attack looks like is given in Fig. 6.1. Here the black dotted lines represent the authentic GNSS signals and the red dotted lines represent the spoofed signals. It is also called a spoofing attack if the outputs, position, velocity and time derivations are derived correctly but based on signals that are not from the authentic GNSS satellites [47]. The latter addition is needed to incorporate the attack mode in which the spoofer moves the authentic correlation peak of its location in a continuous manner. This is what happens in a synchronous spoofing attack, more on this in Section 6.4.

Spoofing of civilian receivers is relatively easy compared to the spoofing of military GNSS receivers. The military GNSS signals are protected by private encryption keys, which define their spreading codes. In contrast, the civilian spreading codes are publicly known, making them more vulnerable to spoofing attacks.

This chapter starts in Section 6.1 with an introduction to spoofing. Following are four sections on different types of spoofing: software-only spoofing in Section 6.2, asynchronous spoofing in Section 6.3, synchronous spoofing in Section 6.4. Section 6.5, is about time delay spoofing. In Section 6.6 it is explained how the implementation of synchronous spoofing was approached. Section 6.7, explores techniques that can be used to implement synchronous spoofing for future work. This chapter concludes with Section 6.8.

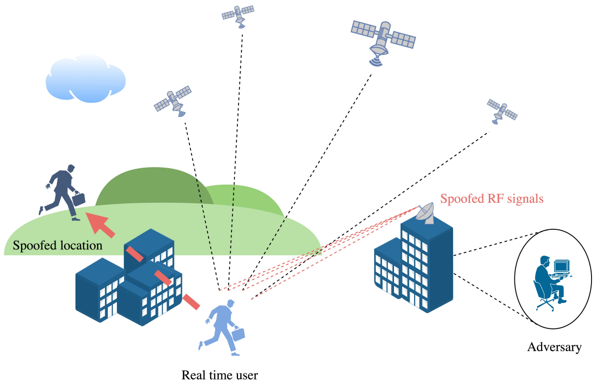


Figure 6.1: Illustration of spoofing

6.1 An introduction to spoofing

In the past two decades, the number of reported spoofing events has risen. Spoofing demonstrations on civilian receivers go back to 2008 [48, 49, 50, 51]. The first attack that was executed dates from 2011, when Iran allegedly hacked a drone [52]. However, it remains up to debate whether GPS spoofing was involved in the hack. In 2017, GPS interference in the Black Sea area was reported that affected the navigation of multiple ships [53, 54].

In the previously named occurences of spoofing, the victim receiver was attacked by a malicious entity. There are also cases in which the victim receiver is owned by the spoofer. People with motives for this are persons with an electronic ankle bracelet, a ship's captain that wants to fish in forbidden waters or a Pokémon Go player who wants to catch Pokémon from home. For the latter, a software-only version of GNSS spoofing is used, more on this in Section 6.2. This type of spoofing uses a downloadable application that can fool the GNSS module of a phone. Android has patched this way of spoofing the GNSS module of its phones [55]. However, it seems that new methods of spoofing the GNSS module have surfaced [56].

Spoofing has been seen as a threat to civilian GNSS receivers for a long time [57]. The Volpe Report, in 2001, created awareness about the fact that a sophisticated spoofer could be able to generate a set of self-consistent pseudo ranges [58]. The FAA responded to the Volpe Report that 'spoofing requires a technologically sophisticated adversary' [59], among other exclamations that indicated the FAA was not likely to undertake action against spoofing. More warnings followed, [48, 60]. These messages were ignored by civil receiver manufacturers up until a working receiver-spoofer was developed [49, 61], see Section 6.4.1.

Spoofing risks

The hazards of spoofing can be very diverse. It is relatively innocent when a person uses software-only spoofing when playing Pokémon Go. In contrast, spoofing a superyacht, thereby steering it off course without having the bridge crew noticing it [51], could endanger the passengers and their surroundings. When spoofing is used to falsify the receiver's position, it can make ships run aground or force an airplane to crash, resulting in very dangerous or even deadly situations.

There are also cases in which spoofing is used to confuse self-driving vehicles that depend on the GNSS signals for their control systems. Spoofing of such systems could cause the car to take the wrong exit, stop or even crash.

Furthermore, spoofing has been used to skip virtual airport queues by Uber drivers [62, 63], which is affecting the customers, other drivers and Uber itself.

Another form of a spoofing attack is when the time within the GNSS data message is changed. This would fool GNSS receivers used for timing applications such as cell phone towers, the power grid and financial transactions. When these applications fail due to timing errors, this could result in communication malfunctions, power outages or disrupting automated financial trading.

6.2 Software-only spoofing

A software-only spoofer is a type that does not transmit any radio waves that interfere with GNSS receivers. Therefore, it is an economical form of spoofing, and it is, for the time being, legal since it is not violating any FCC or international telecommunications regulations [47]. Instead of spoofing the GNSS receiver, it is an application that runs alongside a real GNSS receiving application. This could run on a mobile phone, as with the Pokémon Go and the Uber examples, or any other electronic device.

The additional application that poses as the actual GNSS application cuts off the communication of the GNSS receiving application to the GNSS dependent application, as shown in Fig. 6.2 . It then connects itself to the GNSS dependent application and feeds false position and timing information as input.

Software-only spoofing is a type of spoofing that was not yet foreseen in the Volpe report of 2001 [58].

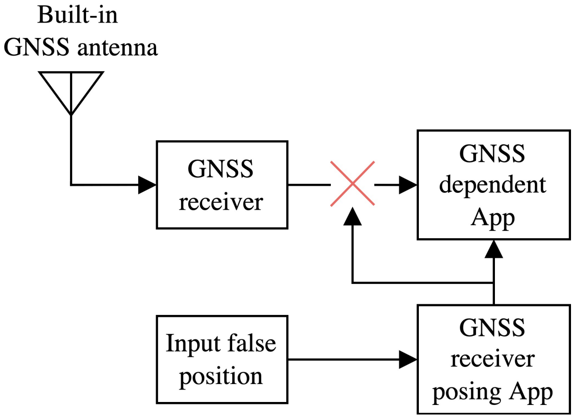


Figure 6.2: Architecture software-only spoofing

6.3 Asynchronous spoofing

Asynchronous spoofing is the transmission of false GNSS signals, which appear to be genuine GNSS signals for the receiver. The signals have the correct frequency, civilian spreading codes and plausible navigation messages. The spoofer must be located in the range of the victim receiver so that it can receive the false GNSS signals. Most correlators of GNSS receivers lock on the signal that has the highest correlation value. Thus, if the false signals' correlation value is slightly higher than those of the authentic GNSS signals, the receiver locks on the false GNSS signals. For most receivers, this is only the case during the acquisition stage. Once a receiver has locked on a GNSS signal, it will keep tracking this signal and will not start to track the false GNSS signals. The receiver can be brought back into acquisition mode if it is first jammed. More on this in Section 6.3.1.

Some receivers contain mitigation techniques that are able to observe when two correlation peaks arise instead of one, see Fig. 6.3. When the receiver notices this, it knows that it is likely being spoofed. However, this could also originate from a multipath propagation of the authentic GPS signal, which can later be filtered from the incoming signal.

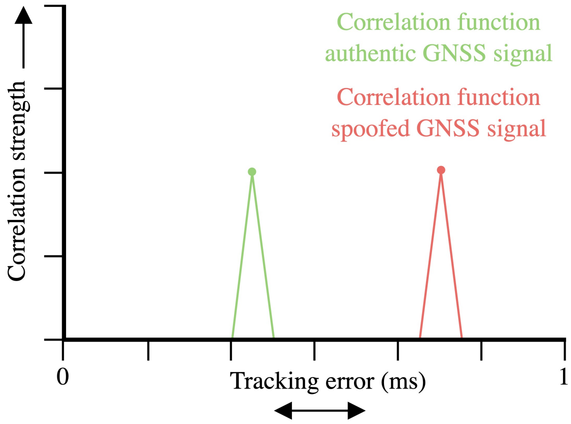


Figure 6.3: Asynchronous spoofing correlation plot

The mathematical representation of the signal that is visible for a receiver when it is spoofed is shown in Eqn. (6.1) [47].

$$\begin{aligned} y(t) = Re\Big\{ &\sum_{i=1}^{N} A_i D_i[t-\tau_i(t)]C_i[t-\tau_i(t)]e^{[j2\pi\cdot f_c t-\phi_i(t)]}+ \\ &\sum_{l=1}^{N_s} A_{sl} D_{sl}[t-\tau_{sl}(t)]C_{sl}[t-\tau_{sl}(t)]e^{[j2\pi\cdot f_c t-\phi_{sl}(t)]}\Big\} \end{aligned} \tag{6.1}$$

The signal that is received by the victim receiver is the superposition of the true GNSS signals and the spoofed GNSS signals. In Eqn. (6.1) N represents the number of visible satellites and N_s the number of spoofed satellites. The carrier amplitude of the signals is A_i and A_{sl} for the true and spoofed signals respectively. The navigation data is represented by D_i and D_{sl}, for the true and spoofed signals. The respective i-th true and l-th spoofed spreading codes are C_i and C_{sl}. The code phases and carrier phases for the true and spoofed signals are, τ_i, ϕ_i, τ_{sl} and ϕ_{sl} respectively. The center frequency is f_c.

To take over the receivers' lock of the true signals, it is often required that at least the same number of satellites are spoofed. This means $N_s \geq N$. For the receiver to believe a spoofed satellite could be a real one, it is required that the spoofed spreading codes are the same as the real ones. Otherwise the satellites will not show up in the correlator of the receiver, thus $C_{si}(t) = C_i(t)$ for all $i = 1, ..., N$. As mentioned before, the signals of the spoofer need to arrive at the receiver with a higher power than the true signals have. This makes sure that the receiver's correlator locks on the spoofed signals, thus $A_{si} > A_i$ for all $i = 1, ..., N$. For more information about the correlator of the receiver, see Section 2.2.

With asynchronous spoofing, the code and the carrier phases are almost always out of sync with the authentic signals. This often is the case because of a lack of information or knowledge on the adversary's side. This will result in two correlation peaks being visible for the victim receiver, see Fig. 6.3.

When spoofing, the adversary can change one or more parameters of the spoofed GNSS signals to confuse the victim receiver. A strategy could be to only adapt the navigation data bits and keep the code and carrier phases the same, $\tau_{si}(t)$ and $\phi_{si}(t)$ for $i = 1, ..., N$. Another approach could be to adapt the code and carrier phases to induce false timing and positioning.

6.3.1 Jam then spoof

Receivers are more vulnerable to spoofing attacks when an adversary first jams the signal, after which the receiver should restart the acquisition process. This makes it easier to use unsophisticated spoofing attacks.

6.4 Synchronous spoofing

When spoofing synchronously, the correlation peak of the spoofed signals should be located at the exact location as the authentic GNSS signals. This only happens if the spoofed signals contain the same data as the authentic signals and therefore have the same waveform. However, this property alone is not enough to be able to spoof a GNSS receiver. The spoofed signals should be synchronized accurately with the authentic signals in both frequency and time.

An example is given in Fig. 6.4. In this example, the assumption is made that the frequency of the spoofed signal is the same as the original signal at the receiver's location, the Doppler shifts of both signals are aligned. In the figure, the blue line is the combined correlation of the original signal (green) and the spoofed signal (red), with the black dots being the early, prompt and late replicas as tracking dots. The receiver will only be able to see the combined correlation. The spoofer takes over the receiver's Delay Lock Loop (DLL) as follows :

1. At first, the spoofed signal is absent.

2. Then, the gain of the spoofed signal is slowly increased so that it is not immediately noticed by the receiver.

3. This continues until the spoofed signal has a higher gain than the authentic signal.

4-5. After this, the delay of the spoofed signal is changed so that the tracking loop of the receiver follows the spoofed signal's correlation peak.

6. When the correlation peak has been moved away from the original correlation peak, the spoofed signal has taken over the receiver's DLL. The receiver is now tracking the spoofed signal.

The spoofed signal should be aligned very accurately to the authentic signal for this takeover to work. The correlation peak of an authentic signal often has a width of around 1955 ns.

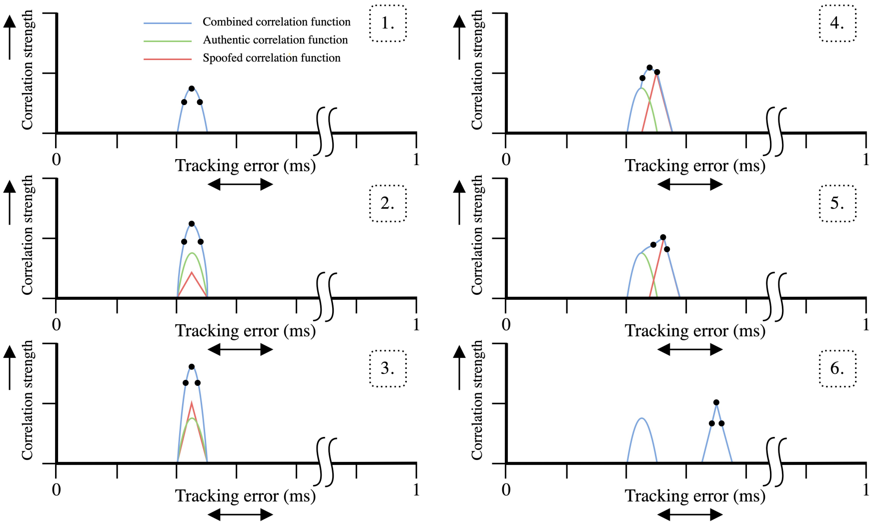


Figure 6.4: Evolution correlation peak synchronous spoofing

This can be derived in the following way. The total width of the correlation peak is equal to two PRN chips. This is because the correlation of a rectangular pulse signal with itself is a triangle that has twice the width of the rectangule and since the rectangular pulse signal, in this case, is one PRN chip wide [10], the correlation peak should be two PRN chips wide. For GPS the PRN frequency is 1.023 MHz. This results in two chips being equal to $\frac{2}{1.023 \cdot 10^6} = 1995$ ns, which is the width of the correlation peak.

Another approach for deriving the width, using the correlation peak of a captured GPS signal, is as follows. The sampling frequency was 53 Mhz. The width in samples of the correlation peak is $6115 - 6030 = 85$ samples, see the red lines in Fig. 6.5. In order to derive the width in seconds, this has to be divided by the sampling frequency, $\frac{85}{53 \cdot 10^6} = 1.604 \cdot 10^{-6}$ s, which is equal to 1604 ns. This is $\frac{1995-1604}{1995} \cdot 100 = 19.6\%$ less wide than the theoretical width of the correlation peak, which is too large a difference for it to be correct.

However, the previous approach of calculating the width of the correlation peak in Fig. 6.5 only takes the tip of the iceberg of the correlation peak into account. The remainder of the correlation peak's width is burried underneath the noise floor that has rissen due to the squaring operation in non-coherent integartion, see Section 2.2.3. When the correlation peak is extrapolated it has a range of $6123 - 6019 = 105$ samples, see the dotted blue lines in Fig. 6.5. This gives a width of $\frac{105}{53 \cdot 10^6} = 1981$ ns, which is a much more plausible width for the correlation peak.

The remaining difference in width between the theoretical and the measured case can be caused by the atmosphere that changes the GPS signal, multipath propagations, measurement errors, errors in the extrapolation or processing errors within the receiver.

For the spoofed signal to be considered synchronous, its correlation peak should be located within 500 ns of the authentic signal's peak since otherwise two separate correlation peaks become visible to the receiver. If that is the case, it no longer is synchronous spoofing but asynchronous spoofing.

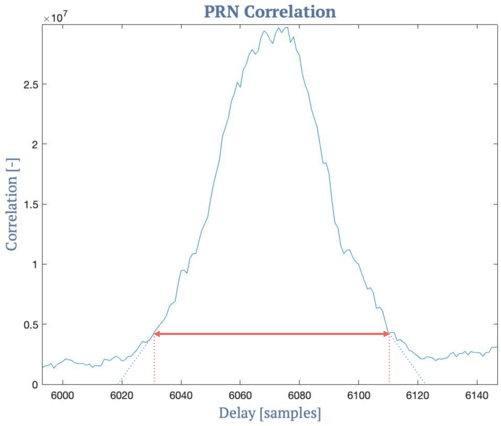


Figure 6.5: Width correlation peak

Matching the Doppler shift of the spoofed signals to the authentic signals is easier than it is to align the signals in time. The Doppler shift on the authentic signals can be calculated accurately. Using Rinex files[i], the location and speed of the satellites and the ionospheric delay can be computed. This information is enough for the spoofed signals to have a matching Doppler shift with the authentic signals.

When the location is spoofed inaccurate this will results in a timing error. However, this error is relatively small. A location error of 5 meter results in a 16.68 ns timing error, $\delta t = \frac{\delta x}{c} = \frac{5}{2{,}9979 \cdot 10^8} = 16.68$ ns. For synchronous spoofing to work, the maximum timing error should be at most 500 ns. A small simulation error of the position thus has no significant impact on the synchronous spoofing. However, when an error in the timing occurs, this leads to a larger error in the correlation peak's location.

In the example of Fig. 6.4 the gain of the spoofed signal is adequate, as the spoofed correlation peak is not much higher than the original correlation peak and also not lower. An adversary does not know the height of the correlation peaks of the authentic signals at the location of the victim receiver. The adversary has to guess what this value will be. However, he can make an educated guess by deploying a receiver himself to measure the signal strength of the GPS signals at his location and estimate which gain is necessary to spoof the victim receiver. The adversary does not want to transmit the spoofed signals at a much higher level than the authentic signals since the victim receiver is more likely to detect this.

6.4.1 Receiver spoofer

By connecting a GNSS receiver to the simulator, it can see and track the visible GNSS signals. Although the simulator would also be able to find out which satellites could be visible for the victim receiver by using Rinex files, this offers the advantage of obtaining insight into the strength of the signals[ii]. Moreover, the simulator can derive its clock offset from the GPS time.

[i] these are files that contain satellite navigation data relative to a specific time interval

[ii] given that the spoofer is in the neighborhood of the victim receiver

6.4.2 Nulling spoofing

In the case of synchronous spoofing, the authentic correlation peak stays visible for the receiver once the spoofing signal is done with the drag-off. There are two correlation peaks visible. Although the receiver is tracking the spoofed signal, this could alarm a sophisticated GNSS receiver that it is being spoofed. A solution to this problem could be to use nulling to remove the authentic correlation peak. Nulling can be done by transmitting the counter phase of the authentic GPS signal.

An example is given in Fig. 6.6. The colours in the figure have the same meaning as in Fig. 6.4. The different stages shown are as follows:

1. The spoof signal is absent.
2. The gains of the spoofed signal and the counter phased signal are increased. For the receiver, nothing changes since both simulated signals cancel each other.
3. The gain of the simulated signals is increased until it has the same level as the original signal.

4-5. The spoofed signal is now dragged away from the authentic correlation location. The nulling signal can remove a large part of the authentic correlation peak. However, a small ripple will remain visible at the receiver.

6. When the correlation peak has been moved away from the original correlation peak, the spoofed signal has taken over the receiver's DLL. The receiver is now tracking the spoofed signal.

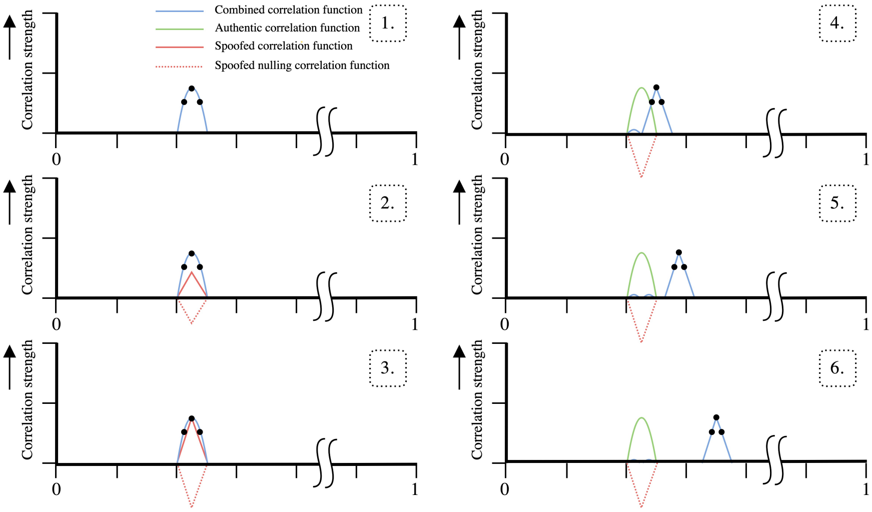


Figure 6.6: Evolution correlation peak time delay spoofing

In the previous example, the assumption is made that the gain of the spoofed and nulling signals match the authentic GPS signal very accurately. However, if the gain of the nulling signal does not match the true signal's gain, the remaining ripple will be more significant.

If the receiver's noise floor is raised, the remaining ripple could be masked. This could make the remaining ripple vanish among the noise floor.

If a satellite is simulated but not visible to the victim receiver, this could result in irregularities in the correlation figure of the receiver. As the correlation cannot become negative, for non-coherent integration[iii] see Section 2.2.3, the effect of the nulling signal will not cause a large negative peak in the correlation figure. However, at the location of the nulling signal, a large positive correlation peak will become visible, making the spoofing detectable to the receiver.

6.5 Time delay spoofing

The idea of implementing this type of spoofing arose during the implementation of synchronous spoofing. Nothing could be found in literature regarding this type of spoofing. Instead of introducing a second correlation peak directly underneath the authentic correlation peak into the receiver, the spoofed correlation peak will be introduced with a different tracking error than the authentic GPS signal. The Doppler shift of the spoofed signal should be simulated correctly and be the same as on the authentic GPS signal. This simulator is able to estimate the Doppler shift correctly.

As can be seen in Fig. 6.7, the Doppler shifts of both correlation peaks are equal. However, the time delays differ.

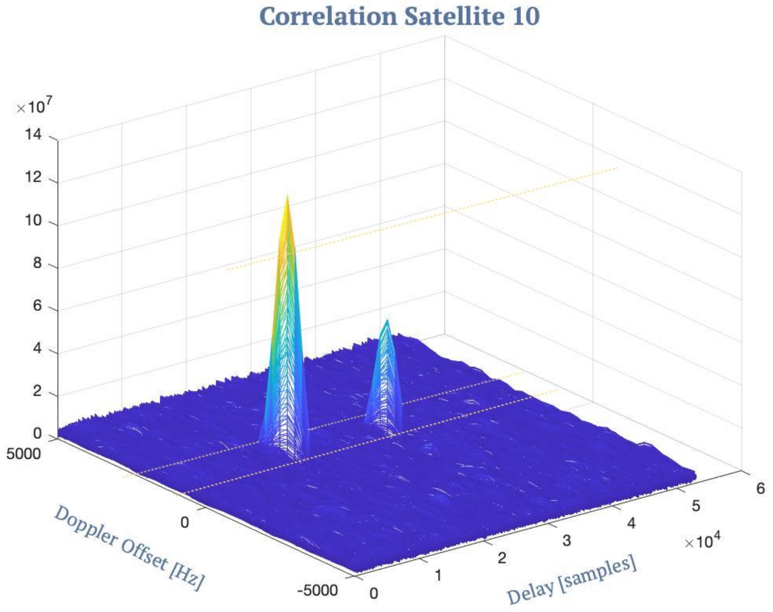


Figure 6.7: Correlation figure of satellite 10

[iii] This type of integration is mostly used within GNSS receivers

The take over of the receiver's lock on the authentic GPS signal by the spoofed GPS signal, shown in Fig. 6.8 is as follows :

1. The spoofing signal is absent.

2. The spoofed GPS signal is introduced with a power level slightly higher than that of the authentic GPS signal.

3. The time delay of the spoofed GPS signal is changed so that the spoofed correlation peak will slowly sweep over the entire range of possible tracking error values.

4-5. The spoofed signal passes over the correlation peak of the authentic GPS signal. Thereby it traps the tracking loops of the GPS receiver. The spoofed correlation peak is dragged away from the authentic correlation location.

6. When the correlation peak has been moved away from the original correlation peak, the spoofed signal has taken over the receiver's DLL. The receiver is now tracking the spoofed signal.

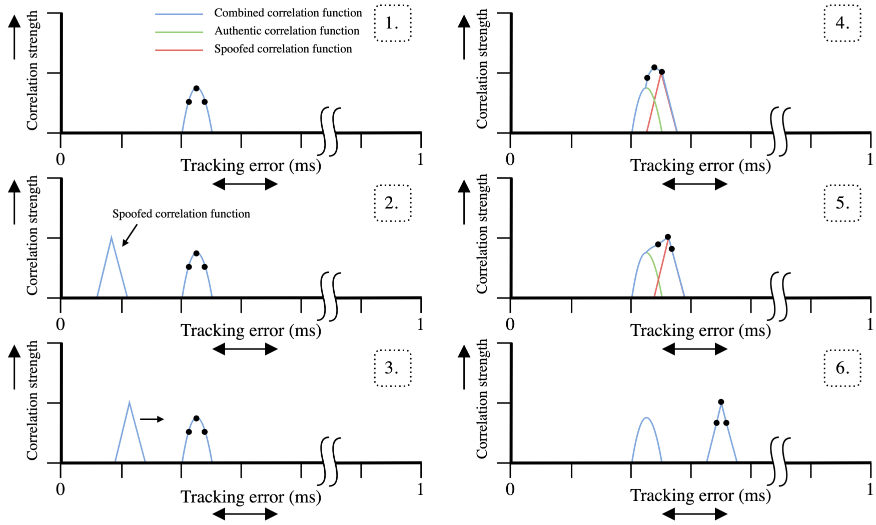

Combined correlation function
Authentic correlation function
Spoofed correlation function
1.
2.
3.
4.
5.
6.
Spoofed correlation function
Correlation strength
Tracking error (ms)
0
1

Figure 6.8: Process of time delay spoofing (for clarity the x-axis title has been left out)

If this type of spoofing can take over the lock of a GPS receiver by trapping its tracking loops (DLL, PLL and FLL), it is a more feasible type of spoofing than synchronous spoofing. This is because the time synchronization of the spoofed signal does not have to be as accurate as is the case with synchronous spoofing.

6.6 Research path spoofing

Two types of spoofing were attempted in this thesis. These were synchronous spoofing, as discussed in Section 6.4 and time delay spoofing, as discussed in Section 6.5. This section will describe the findings that were discovered during the implementation. The receiver used for spoofing is the same as the time-worn receiver used with the jamming experiments. This receiver was used as it was the least advanced receiver that was available and therefore the chances were highest that the receiver could be spoofed.

6.6.1 Implementation synchronous spoofing

In this section, the findings and steps taken during the implementation of synchronous spoofing are discussed. For the implementation of synchronous spoofing, the asynchronous spoofing simulator of CGI was used. In Section 6.4 it is explained how a synchronous spoofing attack is carried out.

Simulator inconsistencies

At first, the code was checked for inconsistencies. The observation was that the altitude was not correctly implemented in the simulator. It was set to mean sea level altitude instead of the GPS altitude, which is the distance/height from the ellipsoid. Hereafter, it was verified whether the simulator took the leap seconds since 1 January, 1980 into account. Up until now[iv], the number of leap seconds is 18. As the number of leap seconds used in the simulator was correct, this led to the conclusion that the remaining offset might not come from errors within the code of the simulator.

Another inconsistency that was found was that the spoofing simulator had an offset of 4 seconds to the authentic GPS signals. This offset is too large to spoof the GPS signals synchronously. The delay was found by comparing the timing information of the simulator against a GPS receiver that was capturing real-time GPS signals. For synchronous spoofing to work, the offset of the spoofed signals should be smaller than 500 ns, as explained in Section 6.4.

The simulator uses the computer time as synchronization. It was found that the time on the computer was not synchronized correctly and thus neither was the time of the simulator. In order to correct the time offset, the computer was synchronized with Network Time Protocol (NTP) servers from the NTP pool project [64]. These servers have an accuracy of stratum two or three, depending on the available servers at the moment of synchronization. A stratum level of zero is a device directly connected to a GPS antenna and has little or no delay. Stratum level zero devices cannot be directly connected to the internet. Instead, they first need to be connected to a computer, which operates as a stratum one server. A larger delay and inaccuracy are introduced when moving up the stratum scale. A stratum two server is connected to a stratum one server and receives its time via a network path. The NTP provides good accuracy in the range of 1 to 50 ms [65]. Although it was already known that by synchronizing the computer with a NTP server the maximum time delay of 500 ns would be exceeded. However, it was done since this would decrease the time offset from 4 seconds to a couple of milliseconds.

Once the computer was synchronized with a stratum two server, the time was accurate within 3 ms. For the simulator to obtain the highest possible time accuracy, a

[iv] 12-05-2022

GPS receiver should be connected to the computer to directly feed the current GPS time to the simulator. This was not implemented directly since this implementation would take a considerable amount of time. Therefore it was checked first whether other delays or inconsistencies would make synchronous spoofing infeasible.

Delays between simulator and HackRF

An aspect that could prevent synchronous spoofing from being possible is the delays that are present between the simulator and the transmission of the Radio Frequency (RF) signal from the HackRF. The spoofed signals are generated in real-time by the simulator and sent to the HackRF via a pipeline. It was checked whether the spoofed signals could be generated fast enough by the simulator. This was done by measuring the time from the start of the signal's generation until the first data was sent over the pipeline. In Section A.8 the function used for timing these delays is discussed.

The delay between generating the spoofed signals and the transmission over the pipeline had a mean of 19.34 ms and a standard deviation of 2.28 ms. Because it is a variable delay, it cannot be added to the time that the spoofed signals are created for. However, it is possible to pre-build the spoofed signals, save them to a binary file and transmit them precisely at the moment in time they are built for. In the first implementation of this approach, there was no pipeline that was already openend between the simulator and the HackRF. Instead, the hackrf_transfer program was used. This program directly transfers data from a binary file to the HackRF. Unfortunately, the delays measured using this approach were even more considerable. It had a mean delay of 36.86 ms and a standard deviation of 7.27 ms. Again this delay is not fixed and thus cannot be added to the time the spoofed signals are created for.

The increase in delay between the generate and transmit and the build and transmit approach is most likely because the hackrf_transfer program still needs to boot when called. Therefore a combination of the two approaches was implemented. The spoofed signal was pre-created and saved in a binary file and a pipeline already opened between the simulator and the HackRF was used. This approach gave a mean delay of 482.90 us and a standard deviation of 85.41 us. By adding the smallest found delay to the time of transmission, the mean delay was made 73.20 us. Although the delay has been decreased significantly, it is still too large to spoof a GPS receiver synchronously.

Delays of SDR processing

So far only delays that originate from the simulator and the laptop have been discussed. The signal processing of the SDR (i.e., HackRF) adds an unknown but significant amount of latency that is difficult to account for. The most significant delays are of DAC buffering, τ_{DAC}, FPGA buffering, τ_{FPGA}, USB output buffering (delay depends on the USB transfer block size), $\tau_{usb_{out}}$, USB driver input buffering, $\tau_{usb_{in}}$, and Digital Signal Processing (DSP), τ_{DSP}, such as SDR software filtering buffering and SDR demodulation buffering. The sum of this delay is given in Eqn. (6.2).

For the DAC buffering delay a deterministic latency can be given, this is a guaranteed upper limit on the latency. The estimated transmission latency, $\tau_{lat,Tx}$, ranges from

10 to 30 ms [66]. The variation in these latencies are too large, making it impossible to synchronously spoof GPS, without the reduction of this variation in latency.

$$\tau_{lat,Tx} = \tau_{DAC} + \tau_{FPGA} + \tau_{usb_{out}} + \tau_{usb_{in}} + \tau_{DSP} \tag{6.2}$$

It is not possible to make the HackRF buffer the spoofed signal as it does not have a buffer that is suited for this. Even if the HackRF had such a buffer, it would not be achievable to start transmitting this buffer at an exact time instance as the HackRF does not contain a clock for counting time.

This section described the findings during the implementation of a synchronous spoofing simulator. Even if the computer clock would be synchronized directly to GPS time via a receiver, the delays between the simulator and the HackRF, as well as the delays from the input of the HackRF to the output of RF signals, are too large. In Section 6.7 two other ways that could be used to implement synchronous spoofing are explained.

6.6.2 Implementation time delay spoofing

This section will describe how the implementation of time delay spoofing was approached, along with the findings. In Section 6.5 the basic principles of time delay spoofing are explained.

The setup that was used for time delay spoofing is shown in Fig. 6.9. The HackRF transmits the spoofed GPS signal over an SMA coax cable which is fed into the same signal combiner used for the practical evaluation of the jamming signals, see Chapter 5. To the other input of the combiner a GPS antenna is connected. The output of the combiner is fed to the time-worn receiver.

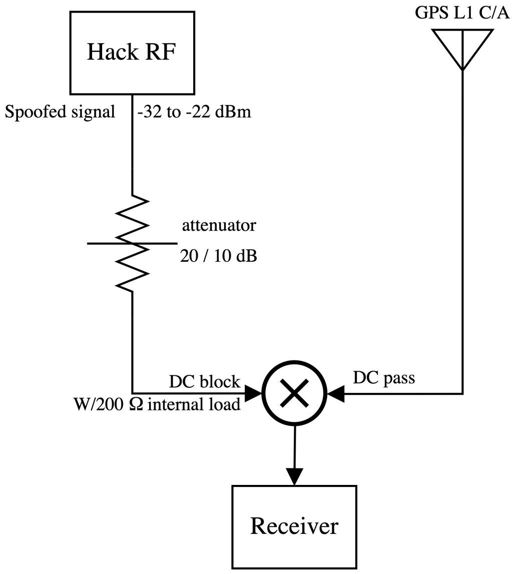


Figure 6.9: Test setup time delay spoofing

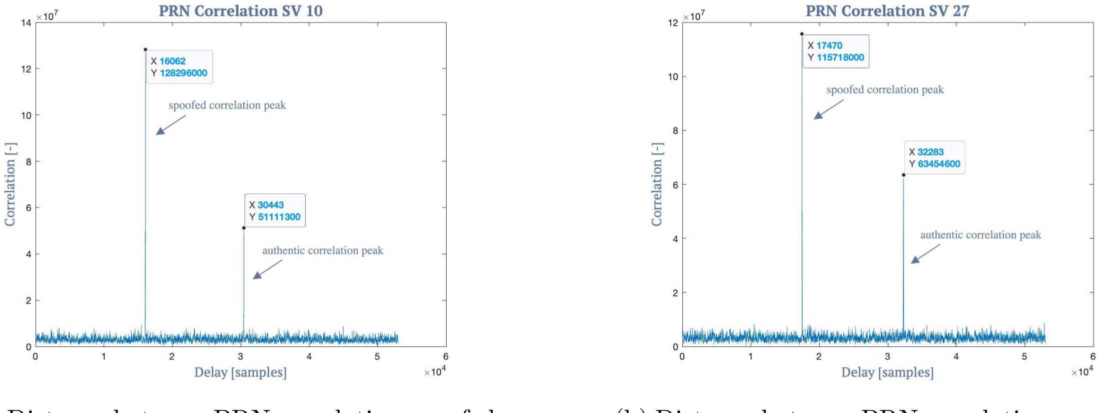
In the simulator, the time delay of the spoofed signal is changed. This has been implemented in two ways. The first method was to save the time delay of every individual satellite at the start of the simulation. Every interval the time delay of the corresponding satellite was increased by one chip. This was done for three types of intervals, 0.1, 0.5 and 1 second. The speed with which the time delay was changed was varied as it is not known how the receiver would react to the changing time delay and whether it would be able to track the signal. The tests executed using this method of varying the time delay showed that it was changed in the wrong way. This was found as the receiver was unable to track the spoofed GPS signals even if only the spoofed signals were fed into the receiver without the authentic GPS signals.

Therefore, a second method was implemented. In this method, the additional time delay shift was added to the time delay shift that is calculated continuously throughout the simulation by the simulator based on the satellite's location. Since the satellites are moving, the time delay is constantly changing.

For time delay spoofing, the power of the spoofed GPS signals should be a little higher than those of the authentic GPS signals at the input of a GPS receiver. As there was no equipment available to measure the incoming power levels, the tests were carried out with different power levels for the spoofed signal to see whether the spoofed signals could capture the lock of the receiver. For this, power levels from -52 dBm to -42 dBm were used, as it was found that this range was large enough. When a power level of less than -52 dBm is used the spoofed signal is no longer stronger than the GPS signal, which is required inorder to be able to spoof a GPS receiver.

By time delay shifting the spoofed signal, the receiver's lock could not be captured. At first only the GPS signal is visible for the receiver. Once the test starts the spoofed signal, which is changing its time delay, is presented to the GPS receiver too.

The spoofer was able to capture the lock of the receiver by making the receiver lose track of the authentic GPS signals by transmitting the spoofed GPS signals, this was with an output of -42 dBm from the HackRF.



(a) Distance between PRN correlation spoofed and authentic signal of SV 10

(b) Distance between PRN correlation spoofed and authentic signal of SV 27

Figure 6.10: PRN correlation of two SVs

One of the reasons that the time-worn receiver does not switch its lock from the authentic GPS signal to the spoofed GPS signal could be that the time delays of simulated signals are not consistent among different Satellite Vehicles (SVs). To illustrate this a capture[v] of the combined signal, the authentic GPS signal and the spoofed signal were taken. In Matlab, the correlation of this signal was calculated using the parallel code phase search algorithm, which is described in Section 2.2.1. In Fig. 6.10a the PRN correlation is shown of SV 10 and in Fig. 6.10b the PRN correlation of SV 27 is shown. The Doppler shift of the spoofed signal is equal to that of the authentic GPS signal. Therefore, only the axis of the time delay is shown.

By showing the difference in distance between the correlation peaks of the authentic and spoofed GPS for SV 10 and 27 it can be shown that the time delays are not simulated consistently by the simulator.

$$\begin{aligned}\text{Difference correlation peaks SV } 10 &= 30,443 - 16,062 = 14,381 \text{ samples}\\ \text{Difference correlation peaks SV } 27 &= 32,283 - 17,470 = 14,813 \text{ samples}\end{aligned} \tag{6.3}$$

In Eqn. (6.3) the difference between the authentic correlation peak and the spoofed correlation peak of SVs 10 and 27 are calculated. From this calculation, it can be seen that the time delay difference between the authentic and spoofed GPS signal of the SVs are not the same. The difference between the two SVs is 8.34 code chips. The calculation for this is given in A.9.

Since the time delay differences between SVs are not equal, the spoofed GPS signals cannot take over all the SVs of the receiver simultaneously. Therefore, Receiver Autonomous Integrity Monitoring (RAIM) is activated at the moment one of the spoofed satellites' correlation peaks is in sync with the correlation peak of the authentic signal. RAIM is an integrity check that is executed by receivers and it checks whether the location and timing information of a satellite is consistent with itself and the other satellites. If this is not the case, it will filter out the inconsistent SV in the calculation of the receiver's location.

6.7 Future work spoofing

This section explores techniques to implement synchronous spoofing for future work. First, some recommendations for the simulator and techniques for synchronous spoofing are made. Following are two sections that elaborate on two techniques for implementing synchronous spoofing.

An irregularity that should be improved in the simulator is the time shift delay between different satellites. This is necessary for synchronous spoofing as otherwise, it is impossible to take over the receiver's lock of multiple satellite signals.

The delays caused by the computer and the simulator can most likely be reduced by using a real-time operating system. Such a system can schedule the tasks of the simulator, giving them a higher priority over the other tasks that the computer executes. This can render pre-building the spoofed GPS signals redundant and enables the creation of the spoofed signals in real-time. Using a Field-Programmable Gate Array (FPGA)

[v] this capture was taken with a sampling frequency of 53 MHz at 17:53 09-05-2022 in Delft, The Netherlands

can also reduce the delays that are currently present in the real-time simulation of the spoofed GPS signals. Because the FPGA will only execute the simulator program, no other programs are running in the background, as is the case on a computer. However, when implementing either of those concepts, the delay caused by the SDR remains. Therefore, the following two sections expand on techniques that offer solutions to this problem. In Section 6.7.1 a technique to synchronously spoof using an SDR that is equipped with a buffer is discussed. After that Section 6.7.2 explores an approach that uses a buffer and a correlator to correct for the time offset between the spoofed GPS signal and the authentic GPS signal.

6.7.1 SDR with buffer

As briefly discussed in previous sections, it is possible to reduce the delay caused by an SDR with the use of a buffer in the SDR. The HackRF is not suited for this. A block diagram of the implementation is given in Fig. 6.11. The spoofed GPS signals are created by the simulator for a transmission time that lies in the future and are stored in the SDR's buffer. The RF signal can then be transmitted with a smaller delay, as the pre-processing has already been done. The steps that are taken in the pre-processing were discussed in Section 6.6.1. A GPS receiver will be connected to the computer in order to guarantee accurate timing information. The SDR should possess the option to track time. The SDR should also possess the option to receive Pulse Per Second (PPS) signals to keep its time synchronized. A PPS signal is used by GPS receivers to mark the start of the next second. Along with the spoofed signals, the simulator will provide the time of transmission to the SDR to ensure transmission of the signals at the right moment.

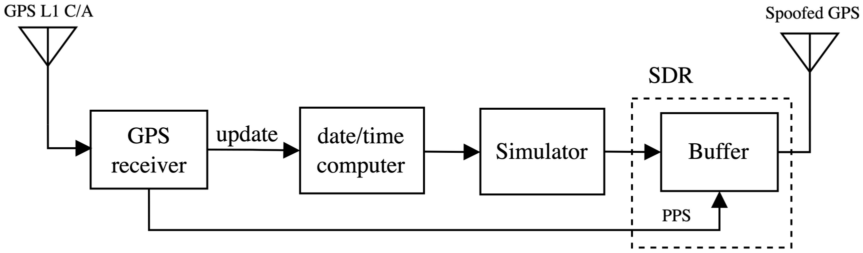


Figure 6.11: Setup synchronous spoofing using an SDR buffer

Using this method, the delays discussed in Section 6.6.1 are removed. The remaining offset of 3 ms on the computer is removed by using a GPS receiver as a time source. The delay caused by the SDR is removed since the signal is pre-processed by the SDR and can be transmitted directly from the buffer. However, it should be tested what the delay is between the buffer and transmission, as the signal still needs to be processed by a Digital-to-Analog Converter (DAC).

6.7.2 Shift through buffer

A method that requires less complex hardware than the previously suggested implementation, is to create a buffer within the simulator and shift the signal over time based on the difference between the correlation peaks of the authentic and spoofed signals. A visualisation of this implementation is given in Fig. 6.12.

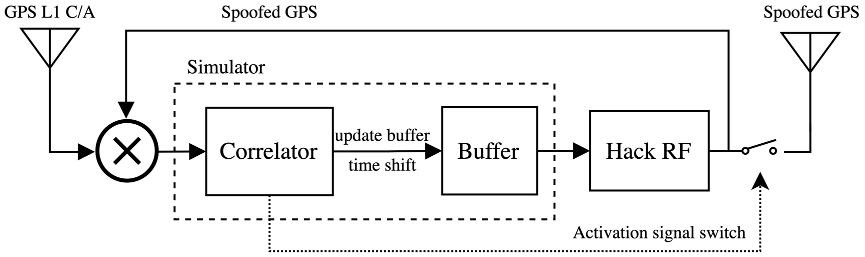


Figure 6.12: Setup synchronous spoofing by shifting through a buffer

In the simulator, a correlator and a buffer are added. These two components can be used to shift the correlation peak of the spoofed GPS signals to the correct position before the spoofed GPS signals are transmitted. The process consists of two steps. In the first step, spoofed GPS signals are created and saved in the buffer. The signals are not yet transmitted as RF signals. Instead, they are transmitted over an SMA coax cable and fed in a signal combiner along with real-time GPS signals captured by an antenna. In the simulator, the correlator will then calculate the correlation figure of the signal, see Fig. 6.13. The correlator will pass the time delay difference on to the buffer that will shift the signals within the buffers accordingly. Only when the correlation peak of the spoofed GPS signals matches the correlation peak of the authentic GPS signals the simulator will send a signal to the switch, starting the transmission of RF signals.

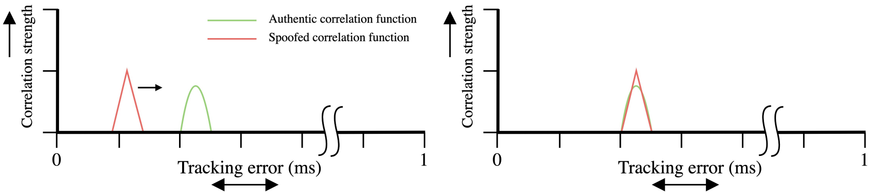


Figure 6.13: Shifting correlation peak of spoofed GPS signal

This approach does not require a GPS receiver to be connected to the computer for providing timing information. This method can deal with the variable delays that are present in the system as the correlator can constantly compare the correlation peak of the spoofed signals with the authentic GPS signals and update the spoofed signals accordingly.

6.8 Spoofing conclusion

This chapter discussed several types of spoofing, as well as a couple approaches used to try and achieve synchronous spoofing. During the process of trying to implement synchronous spoofing it was found that the delays within system were too large to attain synchronous spoofing. Therefore, the attempt as made to implement synchronous spoofing using another approach, which was given the name 'time delay spoofing'. Unfortunately this approach did not work due to time delay offsets between SVs.

The final Section 6.7 described two other approaches for synchronous spoofing. One that used an SDR with a buffer and another technique that used a buffer within the simulator. These methods could not be tested due to time contraints.

7 Conclusion

The goal of this thesis was to research the vulnerabilities of GNSSs to jamming and spoofing. The research consisted of two parts. The first concerned the impact jamming signals have on GNSS receivers. The second part investigated whether spoof signals could be created to synchronously spoof and capture the lock of a GNSS receiver without triggering any alarms.

Jamming

A total of seven jamming signals have been created and tested on two GPS receivers, a time-worn and a cutting-edge receiver. Two of the jamming signals were created for jamming the center frequency of GPS. These are the CW and the pulsed sine wave. These two jamming types can also be modulated on other frequencies. Furthermore, a broadband jamming and four chirp jamming signals have been built. These signals have a changing frequency over time. The time-worn receiver showed a similar performance as the cutting-edge receiver when its Interference Mitigation (IM) was turned off. In that case, the CW and pulsed sine wave introduced the most interference into the cutting-edge receiver, making it lose track of the GPS signal at a J/S level of 47 dB. When the IM was turned on, the cutting-edge receiver outperformed the time-worn receiver. The CW and pulsed sine wave were filtered best by the IM, by using a notch filter. The chirp signals were not filtered equally, but the receiver's performance increased. These jamming signals were filtered using wideband IM. The quadratic chirp was filtered best from the incoming signal. This is most likely because most of its jamming power lies further away from the center frequency of GPS. This makes it easier for the receiver to decrease the contribution of this part of the spectrum, thereby achieving a better performance. Remarkable was that the required J/S ratio decreased for the broadband jamming signal on the cutting-edge receiver with IM turned on. This was caused because the GPS signal was also partially removed by the IM when it tried to remove the broadband jamming signal. The best performing jamming signal with IM turned on was the double quadratic chirp. It was able to make the cutting-edge receiver lose lock of the GPS signal at a J/S level of 53 dB.

Synchronous spoofing

For the implementation of synchronous spoofing, the maximum delay between the spoofed and authentic GPS signals should be at most 500 ns. Moreover the Doppler shift of the spoofed signals should match the authentic GPS signal's. At first, the simulator was searched for inconsistencies. Hereafter, delays in the setup were mapped to generate an overview of how those could best be reduced. It was only possible to make the date and time accurate to 3 ms, which is insufficient. However, the decision was made not to increase the accuracy before it was sure that other delays would be within the maximum range of 500 ns. The delay for transmitting the signals from the simulator to the HackRF had a mean of 482.90 us, which is almost 1000 times larger than the maximum allowed delay. Moreover, a SDR has a processing delay ranging from 10 to 30 ms. Compensating for this delay is difficult. Hereafter, another implementation of spoofing was done called time delay spoofing. However, due to offsets between the time delays of different SVs the GPS receiver did not lock onto these signals.

In the end, synchronous spoofing was not achieved. However, some inconsistencies in the simulator were corrected and ideas for future development are provided in Section 6.7.

7.1 Future work

For future work, it would be interesting to investigate how the broadband jamming signal performs if it is amplified after the bandpass filter and redo the experiments for this jamming signal on both receivers.

Due to time restrictions, there was no time to test and implement the three spike CW signal. In future work, this can be easily added to the simulator using the same approach that was used for the CW.

Moreover, for the pulsed sine wave signal, the simulation and test have only been executed with a duty cycle of 50% and a period of 5 ms due to time restrictions. If more time would have been available other duty cycles and periods could have been evaluated and tested. If the period and duty cycle are to be matched to the time constant of the AGC of a receiver, the expectation is that this will significantly affect the Carrier-to-Noise ratio (C/N0) levels of the receiver [32]. This is because the quantization of the signal can no longer be done properly if the AGC is unable to select a proper gain. Without an adequate quantization of the signal, further processing of the signal becomes very difficult. This eventualy results in significantly affected C/N0 levels.

The simulations and evaluations of the other jamming profiles have not been performed with other chirp periods or bandwidths since this is beyond the scope of this thesis. However, it would be interesting to see the effect of the chirp signals with a shorter period since still much of the transmitted power is within the given bandwidth, 0 to 1.023 MHz.

This thesis only focussed on the creation and addition of jamming signals to GPS signals to distort the operation of GPS receivers. However, it might be very interesting to research how well the GPS signals can be recovered from this interference which was done for linear chirp signals in the research of Daniël Kappelle [35].

For spoofing, it might be interesting to implement the approaches provided in Section 6.7 to verify whether these are feasible methods that can lead to synchronous spoofing.

Appendix A

A.1 Derivation navigation equations

The pseudorange from a satellite to a GPS receiver can be deduced from the known equation *distance* = *time* · *speed*. This is shown in the first line of Eqn. (A.1) for the GNSS application. The propagation time from satellite to the receiver, dt_p, is multiplied by the speed of light, c, giving the pseudorange from the satellite to the receiver.

The propagation time is composed of the difference between the time of reception at the user receiver, t_u, and the time of transmission, t_1. However, as explained in Section 2.1 the satellite and receiver have time offsets, δt^{sat} and δt_u respectively, which are not yet taken into account in these equations. This can be rewritten into the fourth line in the equation. Now using Eqn. (2.3) the geometric distance, g= $(t_u{-}t_1){\cdot}c$, can be rewritten giving the final result.

$$
\begin{aligned}
R_k &= dt_p \cdot c \\
R_k &= (t_u - t_1) \cdot c \\
R_k &= ((t_u + \boldsymbol{\delta}\mathbf{t^u}) - (t_1 + \delta t^{sat})) \cdot c \\
R_k &= \underbrace{(t_u - t_1) \cdot c}_{\text{g}} + (\boldsymbol{\delta}\mathbf{t^u} - \delta t^{sat}) \cdot c \qquad\qquad \text{(A.1)} \\
R_k &= \underbrace{\sqrt{(x_k - \mathbf{x_u})^2 + (y_k - \mathbf{y_u})^2 + (z_k - \mathbf{z_u})^2}}_{\text{g}} + (\boldsymbol{\delta}\mathbf{t^u} - \delta t^{sat}) \cdot c
\end{aligned}
$$

A.2 Derivation linearisation geometric distance

In order to simplify Eqn. (A.2) the partial derivatives need to be calculated. In Eqn. (A.3) the partial derivative for x_u is calculated, the same steps can be taken to derive the partial derivatives for y_u, and z_u. As shown in Eqn. (2.10) $g = \sqrt{(x_k - \mathbf{x_u})^2 + (y_k - \mathbf{y_u})^2 + (z_k - \mathbf{z_u})^2}$. Using the following derivation rule, $\sqrt{u}\frac{\partial}{\partial u} = \frac{u\frac{\partial}{\partial u}}{2\cdot\sqrt{u}}$, it is possible to rewrite Eqn. (A.3) into Eqn. (A.4) (where $g_0 = \sqrt{(x_k - x_0)^2 + (y_k - y_0)^2 + (z_k - z_0)^2)}$. The last partial derivative can be rewritten using the derivation rule, $u^2\frac{\partial}{\partial u} = 2u \cdot u\frac{\partial}{\partial u}$, giving the final result in the last line of Eqn. (A.4).

$$
g \approx g_0 + g\frac{\partial}{\partial x_u}|_{x_0,y_0,z_0}\underbrace{(x_u - x_0)}_{\Delta x} + g\frac{\partial}{\partial y_u}|_{x_0,y_0,z_0}\underbrace{(y_u - y_0)}_{\Delta y} + g\frac{\partial}{\partial z_u}|_{x_0,y_0,z_0}\underbrace{(z_u - z_0)}_{\Delta z} \quad \text{(A.2)}
$$

$$g\frac{\partial}{\partial x_u}|_{x_0,y_0,z_0} = \sqrt{(x_k - \mathbf{x_u})^2 + (y_k - \mathbf{y_u})^2 + (z_k - \mathbf{z_u})^2}\frac{\partial}{\partial x_u}|_{x_0,y_0,z_0} \tag{A.3}$$

$$\begin{aligned} g\frac{\partial}{\partial x_u}|_{x_0,y_0,z_0} &= \frac{((x_k - \mathbf{x_u})^2 + (y_k - \mathbf{y_u})^2 + (z_k - \mathbf{z_u})^2)\frac{\partial}{\partial x_u}|_{x_0,y_0,z_0}}{2 \cdot g_0} \\ &= \frac{2(x_k - x_0) \cdot -1}{2 \cdot g_0} \\ &= -\frac{x_k - x_0}{g_0} \end{aligned} \tag{A.4}$$

A.3 Derivation IF

In Eqn. (A.7) the full derivation of Eqn. (2.5) is given. This derivation is using the Euler identity given in Eqn. (A.5). In the first two sentences the identities are fully rewritten. This is similar to the derivation given in Eqn. (A.6), the e-powers can be replaced with $2cos((\omega_{RF} - \omega_{LO})t) + 2cos((\omega_{RF} + \omega_{LO})t)$. If the equation is furter rewritten this gives the result shown too in Eqn. (2.5).

$$cos(\omega t) = \frac{e^{j\omega t} + e^{-j\omega t}}{2} \tag{A.5}$$

$$\begin{aligned} 2[cos((\omega_{RF} - \omega_{LO})t) + cos((\omega_{RF} + \omega_{LO})t)] &= 2 \cdot \Big[\frac{e^{j(\omega_{RF}-\omega_{LO})t} + e^{-j(\omega_{RF}-\omega_{LO})t}}{2} \\ &\quad + \frac{e^{j(\omega_{RF}+\omega_{LO})t} + e^{-j(\omega_{RF}+\omega_{LO})t}}{2}\Big] \\ &= e^{j(\omega_{RF}-\omega_{LO})t} + e^{j(\omega_{LO}-\omega_{RF})t} + e^{j(\omega_{RF}+\omega_{LO})t} + e^{-j(\omega_{RF}+\omega_{LO})} \end{aligned} \tag{A.6}$$

$$\begin{aligned} cos(\omega_{RF}t) \cdot cos(\omega_{LO}t) &= \frac{1}{2}[e^{j\omega_{RF}t} + e^{-j\omega_{RF}t}] \cdot \frac{1}{2}[e^{j\omega_{LO}t} + e^{-j\omega_{LO}t}] \\ &= \frac{1}{4}[e^{j(\omega_{RF}+\omega_{LO})t} + e^{j(\omega_{RF}-\omega_{LO})t} + e^{j(\omega_{LO}-\omega_{RF})t} + e^{-j(\omega_{RF}+\omega_{LO})t}] \\ &= \frac{1}{4}[2 \cdot [cos((\omega_{RF} - \omega_{LO})t) + cos((\omega_{RF} + \omega_{LO})t)]] \\ &= \frac{1}{2}[cos(2\pi(f_{RF} - (f_{RF} - f_{IF}))t) + cos(2\pi(f_{RF} + (f_{RF} - f_{IF}))t)] \\ &= \frac{1}{2}[cos(2\pi f_{IF}t) + cos(2\pi(2f_{RF} - f_{IF})t)] \end{aligned} \tag{A.7}$$

A.4 Received signal power using Friis formula

In this section is explained what the received power at a GNSS receiver is of the GPS signals that are transmitted from the satellites in space. The transmission power differs per satellite. Satellites of type, IIR-M, have an transmission power of 145 W and the satellites that are of type, IIF, have a transmission power of 240 W [67]. For the following calculation an transmission power of 240 W will be assumed.

$$P_r = \frac{P_t \cdot G_t \cdot G_r \lambda^2}{(4\pi R)^2} \tag{A.8}$$

For calculating the received signal power Friis formula is used, given in Eqn. (A.8). Where:

- P_r is the received power
- P_t is the output power of the transmitting antenna
- G_r is the gain of the receiving antenna
- G_t it the gain of the transmitting antenna
- λ is the wavelength of the signal ($\lambda = \frac{c}{f}$)
- R is the distance between the antennas

The gain of the receiving antenna is neglected in this calculation as it differs per receiving antenna, an antenna without gain is assumed (i.e., 0 dB). The gain of a satellite antenna, G_t, is around 16 dBi [68]. The wavelength of the GPS signal is $\lambda = \frac{2.998 \cdot 10^8}{1{,}575.42 \cdot 10^6} = 0.1903$ m. The satellites have an orbit altitude of 20,200 km [69]. The calculation of the received power can be seen in Eqn. (A.9). This gives a received power of $10 \cdot log_{10}(5.3699 \cdot 10^{-15}) + 30 = -112.7$ dBm. However, this calculation does not take losses, other than the free space loss, into account. Taking additional losses, such as atmospherically and depolirization losses, into account the received power decays. The atmospheric attenuation loss is 2.0 dB and the depolirization loss is 3.4 dB [70]. This makes the received signal power $-112.7 - 2.0 - 3.4 = 118.1$ dBm.

$$\begin{aligned} P_r &= \frac{240 \cdot 10^1.6 \cdot 10^0 \cdot 0.1903^2}{(4\pi 20,200 \cdot 10^3)^2} \\ &= 5.3699 \cdot 10^{-15} \end{aligned} \tag{A.9}$$

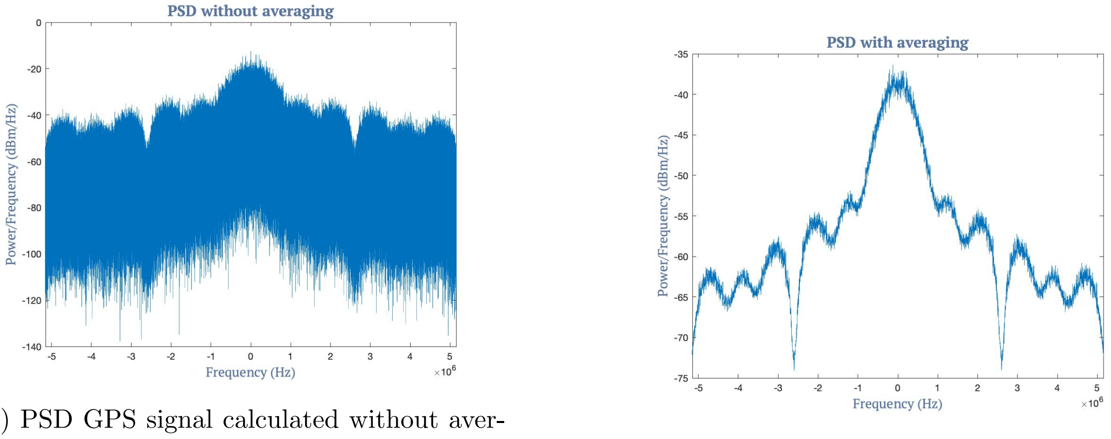
A.5 Derivation AM

The full derivation of Eqn. (3.4) is given in Eqn. (A.11). This derivation is done using Euler's identity, given in Eqn. (A.5). The final part of Eqn. (A.11) is rewritten using the derivation given in Eqn. (A.10). In the final sentence of Eqn. (A.11) $cos((\omega_1 - \omega_2 + \omega_3)t) + cos((\omega_1 + \omega_2 - \omega_3)t)$ is rewritten into $2 \cdot cos(\omega_1 t)$, as f_2 and f_3 are equal in the case shown in Section 3.2.1.

$$\begin{aligned}
&2[cos((\omega_1-\omega_2-\omega_3)t)+cos((\omega_1-\omega_2+\omega_3)t)+cos((\omega_1+\omega_2-\omega_3)t)+cos((\omega_1+\omega_2+\omega_3)t)]\\
&\quad=2\cdot\Big[\frac{e^{j(\omega_1+\omega_2+\omega_3)t}+e^{j(\omega_1-\omega_2+\omega_3)t}+e^{j(-\omega_1+\omega_2+\omega_3)t}+e^{j(-\omega_1-\omega_2+\omega_3)t}}{2}\\
&\qquad+\frac{e^{j(\omega_1+\omega_2-\omega_3)t}+e^{j(\omega_1-\omega_2-\omega_3)t}+e^{j(-\omega_1+\omega_2-\omega_3)t}+e^{-j(\omega_1+\omega_2+\omega_3)t}}{2}\Big]\\
&\quad=e^{j(\omega_1+\omega_2+\omega_3)t}+e^{j(\omega_1-\omega_2+\omega_3)t}+e^{j(-\omega_1+\omega_2+\omega_3)t}+e^{j(-\omega_1-\omega_2+\omega_3)t}\\
&\quad+e^{j(\omega_1+\omega_2-\omega_3)t}+e^{j(\omega_1-\omega_2-\omega_3)t}+e^{j(-\omega_1+\omega_2-\omega_3)t}+e^{-j(\omega_1+\omega_2+\omega_3)t}
\end{aligned}\tag{A.10}$$

$$\begin{aligned}
cos(\omega_1t)\cdot cos(\omega_2t)\cdot cos(\omega_3t)&=\frac{1}{8}[e^{j\omega_1t}+e^{-j\omega_1t}][e^{j\omega_2t}+e^{-j\omega_2t}][e^{j\omega_3t}+e^{-j\omega_3t}]\\
&=\frac{1}{8}[e^{j(\omega_1+\omega_2)t}+e^{j(\omega_1-\omega_2)t}+e^{j(-\omega_1+\omega_2)t}+e^{-j(\omega_1+\omega_2)t}][e^{j\omega_3t}+e^{-j\omega_3t}]\\
&=\frac{1}{8}[e^{j(\omega_1+\omega_2+\omega_3)t}+e^{j(\omega_1-\omega_2+\omega_3)t}+e^{j(-\omega_1+\omega_2+\omega_3)t}+e^{j(-\omega_1-\omega_2+\omega_3)t}\\
&\quad+e^{j(\omega_1+\omega_2-\omega_3)t}+e^{j(\omega_1-\omega_2-\omega_3)t}+e^{j(-\omega_1+\omega_2-\omega_3)t}+e^{-j(\omega_1+\omega_2+\omega_3)t}]\\
&=\frac{1}{4}[cos((\omega_1-\omega_2-\omega_3)t)+cos((\omega_1-\omega_2+\omega_3)t)+cos((\omega_1+\omega_2-\omega_3)t)\\
&\quad+cos((\omega_1+\omega_2+\omega_3)t)]\\
&=\frac{1}{4}[cos((\omega_1-\omega_2-\omega_3)t)+2\cdot cos(\omega_1t)+cos((\omega_1+\omega_2+\omega_3)t)]
\end{aligned}\tag{A.11}$$

A.6 Power spectral density figures

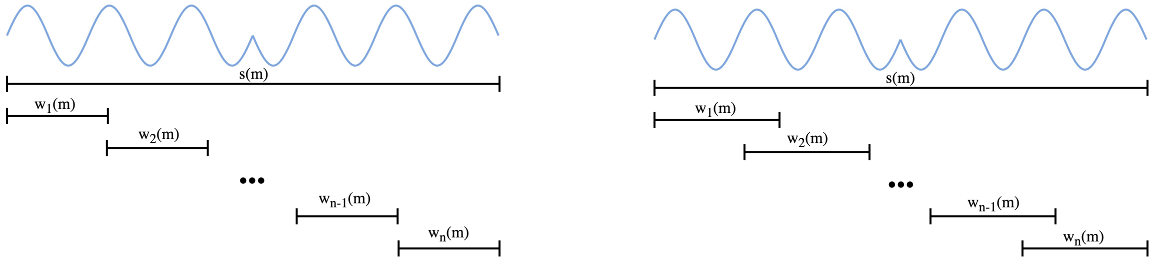


(a) PSD GPS signal calculated without averaging

(b) PSD GPS signal calculated with averaging

Figure A.1: Calculation PSD of a GPS signal

A.7 Windowing



(a) Windowing without overlap

(b) Windowing with overlap

Figure A.2: Windowing

A.8 Timing function Windows

This measurement was done using the Windows function timespec_get(), because it is the most accurate timing function available. The accuracy of this function differs per computer. Therefore, it was checked whether this function would be able to measure nanoseconds, incase the delay could be decreased to such a level later on in the research. The accuracy of this function can be estimated by finding the performance frequency of the computer, which can be found using another Windows function, called QueryPerformanceFrequency(). From this function it was found that the performance frequency of the computer was 10 MHz, the tick interval then is tick interval $= \frac{1}{\text{performance frequency}} = \frac{1}{10 \cdot 10^6} = 100$ ns. This means that the time resolution on the computer, and the timespec_get() function, is 100 ns.

A.9 Difference PRN code chips

In this section the difference in PRN code chips between SVs 10 and 27 is calculated. For this, first the number of samples per chip needs to be calculated, this is done in Eqn. (A.12). The difference in samples between SV 10 and 27 is $14,813 - 14,381 = 432$, see 6.3. Thus the difference in PRN chips can be calculated as given in Eqn. (A.13). Giving a difference of 8.34 chips difference between SV 10 and 27.

$$\begin{aligned} \text{Number of samples per PRN} &= \text{sample frequency} \cdot \text{duration of one PRN code} \\ &= 53 \cdot 10^6 \cdot 0.1 = 53,000 \text{ samples per PRN code} \\ \text{Numer of samples per chip} &= \frac{53,000}{1,023} = 51.8 \text{ samples per chip} \end{aligned} \tag{A.12}$$

$$\text{Differenence in chips} = \frac{432}{51.8} = 8.34 \text{ chips} \tag{A.13}$$

References

[1] *Overview vital processess in the Netherlands.* URL: https://www.nctv.nl/onderwerpen/vitale-infrastructuur/overzicht-vitale-processen. (accessed: 06.06.2022).

[2] J. Bhatti D. Shepard and T. Humphreys. *Drone hack: Spoofing attack demonstration on a civilian unmanned aerial vehicle.* URL: https://gpsworld.com/drone-hack/. (accessed: 16.11.2021).

[3] Rigas Themistoklis Ioannides, Thomas Pany, and Glen Gibbons. "Known Vulnerabilities of Global Navigation Satellite Systems, Status, and Potential Mitigation Techniques". In: *Proceedings of the IEEE* 104.6 (2016), pp. 1174–1194. DOI: 10.1109/JPROC.2016.2535898.

[4] Thomas Kraus, Roland Bauernfeind, and Bernd Eissfeller. "Survey of In-Car Jammers - Analysis and Modeling of the RF Signals and IF Samples (Suitable for Active Signal Cancelation)". In: Sept. 2011.

[5] J. Mitola. "Software radios-survey, critical evaluation and future directions". In: *[Proceedings] NTC-92: National Telesystems Conference.* 1992, pp. 13/15–13/23. DOI: 10.1109/NTC.1992.267870.

[6] José Raúl Machado Fernández. "Software Defined Radio: Basic Principles and Applications". In: 2015.

[7] J.M. Juan Zornoza J. Sanz Subirana and M. Hernández-Pajares. *Ionospheric Delay.* URL: https://gssc.esa.int/navipedia/index.php/Ionospheric_Delay. (accessed: 29.11.2021).

[8] GPS World Staff. "4th GPS civilian signal goes live". In: *GPS World* (Jan. 2020).

[9] J. M. Juan Zornoza J. Sanz Subirana and M. Hernandez-Pajares. *GNSS Data Processing.* Volume I: Fundamentals and Algorithms. ESA Communications, 2013. ISBN: 9789292218867.

[10] URL: https://gssc.esa.int/navipedia/index.php. (accessed: 12.11.2021).

[11] R. Gold. "Optimal binary sequences for spread spectrum multiplexing (Corresp.)" In: *IEEE Transactions on Information Theory* 13.4 (1967), pp. 619–621. DOI: 10.1109/TIT.1967.1054048.

[12] H. Darabi and A.A. Abidi. "A 4.5-mW 900-MHz CMOS receiver for wireless paging". In: *IEEE Journal of Solid-State Circuits* 35.8 (2000), pp. 1085–1096. DOI: 10.1109/4.859497.

[13] Wenxu Zhang et al. "Application of FFT parallel code phase search algorithm in GNSS software". In: *2016 IEEE 13th International Conference on Signal Processing (ICSP).* 2016, pp. 1165–1170. DOI: 10.1109/ICSP.2016.7878011.

[14] Pengda Huang, Y. Pi, and Ilir Progri. "GPS Signal Detection under Multiplicative and Additive Noise". In: *Journal of Navigation* 66 (July 2013). DOI: 10.1017/S0373463312000550.

[15] A. Khintchine. "Korrelationstheorie der stationären stochastischen Prozesse". In: *Mathematische Annalen* 109 (Dec. 1934), pp. 604–612. DOI: 10.1007/BF01449156.

[16] Brook Taylor. "Methodus Incrementorum Directa et Inversa". In: *London* (1715), p. 21–23 (Prop. VII, Thm. 3, Cor. 2). Translated into English in Struik, D. J. (1969).

A Source Book in Mathematics 1200–1800. Cambridge, Massachusetts: Harvard University Press. pp. 329–332.

[17] A. Charnes, E. L. Frome, and P. L. Yu. "The Equivalence of Generalized Least Squares and Maximum Likelihood Estimates in the Exponential Family". In: *Journal of the American Statistical Association* 71.353 (1976), pp. 169–171. DOI: 10.1080/01621459.1976.10481508.

[18] Mounir Adjrad and Paul Groves. "Intelligent Urban Positioning using Shadow Matching and GNSS Ranging Aided by 3D Mapping". In: Sept. 2016. DOI: 10.33012/2016.14845.

[19] Li-Ta Hsu. "Analysis and modeling GPS NLOS effect in highly urbanized area". In: *GPS Solutions* 22 (Nov. 2017). DOI: 10.1007/s10291-017-0667-9.

[20] Z Jiang et al. "Multi-Constellation GNSS Multipath Mitigation Using Consistency Checking". In: (Jan. 2011).

[21] P. Misra and P. Enge. *Global Positioning System: Signals, Measurements, and Performance.* 2nd edition. Ganga-Jamuna Press, Lincoln MA, 2006.

[22] Jasmine Zidan et al. "GNSS Vulnerabilities and Existing Solutions: A Review of the Literature". In: *IEEE Access* (2020), pp. 1–1. DOI: 10.1109/ACCESS.2020.2973759.

[23] "Interference And Jamming:(Un)intended Consequences". In: *Inside GNSS* (Apr. 2012).

[24] Christopher Hegarty et al. "Suppression of pulsed interference through blanking". In: *Proceedings of the IAIN World Congress and the 56th Annual Meeting of The Institute of Navigation (2000).* 2000, pp. 399–408.

[25] Stefan Fulga Dennis Arthur Fielder Anne Fielder. "Gps receiver with improved immunity to burst transmissions". U.S. pat. US20100150284A1. June 27, 2010.

[26] Anthony Flores. "GPS - Interface Control Document (IS-GPS-200)". In: May 2021. URL: https://www.gps.gov/technical/icwg/IS-GPS-200M.pdf.

[27] Anthony Flores. "GPS - Interface Control Document (IS-GPS-800)". In: Apr. 2021. URL: https://www.gps.gov/technical/icwg/IS-GPS-800H.pdf.

[28] EUSPA. "Galileo - Open Service - Signal In Space Interface Control Document (OS SIS ICD v2.0)". In: Jan. 2021.

[29] M. Moussa. "High Resolution Jamming Detection in Global Navigation Satellite System". In: *Electrical and Computer Engineering* (June 2015). Master Thesis.

[30] James J. Spilker Jr. (Editor) Y. Jade Morton (Editor) Frank van Diggelen (Editor) and Bradford W. Parkinson (Editor). *Position, Navigation, and Timing Technologies in the 21st Century: Integrated Satellite Navigation, Sensor Systems, and Civil Applications, Volume 1.* 1st edition. Wiley-IEEE press, Jan. 2021, pp. 619–653. ISBN: 978-1-119-45841-8.

[31] Syed Waqas Arif, Adem Coskun, and Izzet Kale. "A Fully Adaptive Lattice-based Notch Filter for Mitigation of Interference in GPS". In: *2019 15th Conference on Ph.D Research in Microelectronics and Electronics (PRIME).* 2019, pp. 217–220. DOI: 10.1109/PRIME.2019.8787822.

[32] S Heppe and P Ward. "RFI & Jamming and its Effects on GPS Receivers, based on communication theory". In: *Navtech seminars course.* Vol. 452. 2003, pp. 5–6.

[33] S. M. Deshpande. "Study of Interference Effects on GPS Signal Acquisition". In: *Geomatics Engineering* (Jan. 2004). Ph.D Thesis.

[34] Meenakshi Yadav and Shruti Hathwalia. "Design and Analysis of a 32 Bit Linear Feedback Shift Register Using VHDL". In: *Int. Journal of Engineering Research and Applications* 4 (6 June 2014), pp. 99–102. ISSN: 2248-9622.

[35] Daniël Kappelle. "GNSS Chirp Interference Estimation and Mitigation". In: (). URL: http://resolver.tudelft.nl/uuid:0befa060-4a95-4e30-9658-769f20501920.

[36] J. Rossouw van der Merwe et al. "GNSS interference monitoring and characterisation station". In: *2017 European Navigation Conference (ENC)*. 2017, pp. 170–178. DOI: 10.1109/EURONAV.2017.7954206.

[37] Soosan Beheshti and Maryam Ravan. "Adaptive windowing in nonparametric power spectral density estimation". In: *2008 Canadian Conference on Electrical and Computer Engineering*. 2008, pp. 001183–001186. DOI: 10.1109/CCECE.2008.4564725.

[38] William H. [et. al.] *Power Spectrum Estimation Using the FFT*. 2nd edition. Cambridge University Press, 1992, sec. 13.4. ISBN: 978-1-119-45841-8.

[39] Daniele Borio, Cyrille Gernot, and Florence Macchi. "The Output SNR and its Role in Quantifying GNSS Signal Acquisition Performance". In: (Jan. 2008).

[40] J. Kaiser and R. Schafer. "On the use of the I¡inf¿0¡/inf¿-sinh window for spectrum analysis". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28.1 (1980), pp. 105–107. DOI: 10.1109/TASSP.1980.1163349.

[41] Hrishi Rakshit and Muhammad Ullah. "A Comparative Study on Window Functions for Designing Efficient FIR Filter". In: July 2019.

[42] Ali N. Akansu and Richard A. Haddad. "Chapter 5 - Time-Frequency Representations". In: *Multiresolution Signal Decomposition (Second Edition)*. Ed. by Ali N. Akansu and Richard A. Haddad. Second Edition. San Diego: Academic Press, 2001, pp. 331–347. ISBN: 978-0-12-047141-6. DOI: https://doi.org/10.1016/B978-012047141-6/50005-7. URL: https://www.sciencedirect.com/science/article/pii/B9780120471416500057.

[43] Yun Zhao, Xiaonan Xue, and Tingfei Zhang. "Receiver-channel based adaptive blind equalization approach for GPS dynamic multipath mitigation". In: *Chinese Journal of Aeronautics* 26 (Apr. 2013), pp. 378–384. DOI: 10.1016/j.cja.2013.02.015.

[44] *Attenuation LabSat 3*. URL: https://en.racelogic.support/LabSat_GNSS_Simulators/Software_Info/SatGen_V3_Software/SatGen_V3_User_Manual/04_-_SatGen_Dynamic_Scenario_Creation#:~:text=0183%20GGA%20sentence.-,Carrier%20to%20Noise%20Ratio,-SatGen%20can%20reduce. (accessed: 29.03.2022).

[45] NMEA org. *NMEA 0183 standard*. URL: https://www.tronico.fi/OH6NT/docs/NMEA0183.pdf. (accessed: 08.02.2022).

[46] Xinzhi Dai et al. "Performance of GNSS receivers with AGC in noise pulse interference". In: *2016 5th International Conference on Computer Science and Network Technology (ICCSNT)*. 2016, pp. 735–740. DOI: 10.1109/ICCSNT.2016.8070255.

[47] James J. Spilker Jr. (Editor) Y. Jade Morton (Editor) Frank van Diggelen (Editor) and Bradford W. Parkinson (Editor). *Position, Navigation, and Timing Technologies in the 21st Century: Integrated Satellite Navigation, Sensor Systems, and Civil Applications, Volume 1*. 1st edition. Wiley-IEEE press, Jan. 2021, pp. 655–680. ISBN: 978-1-119-45841-8.

[48] J. Sterling Warner, Roger G. Johnston, and Cpp Los Alamos. "A Simple Demonstration that the Global Positioning System (GPS) is Vulnerable to Spoofing". In: 2012.

[49] T. Humphreys et al. "Assessing the Spoofing Threat: Development of a Portable GPS Civilian Spoofer". In: Jan. 2008, pp. 2314–2325.

[50] Andrew Kerns et al. "Unmanned Aircraft Capture and Control Via GPS Spoofing". In: *Journal of Field Robotics* 31 (July 2014). DOI: 10.1002/rob.21513.

[51] Jahshan Bhatti and Todd Humphreys. "Hostile Control of Ships via False GPS Signals: Demonstration and Detection". In: *Navigation* 64 (Mar. 2017), pp. 51–66. DOI: 10.1002/navi.183.

[52] A. Rawnsley. "Iran's Alleged Drone Hack: Tough, but Possible". In: (Dec. 2011). URL: https://www.wired.com/2011/12/iran-drone-hack-gps/.

[53] U.S. DoT Maritime Administration. *Anon.* July 2017. URL: https://www.maritime.dot.gov/msci/2017-005a-black-sea-gps-interference. (accessed: 01.03.2022).

[54] S. Goff. *Mass GPS Spoofing Attack in the Black Sea Strengthen Calls for PNT Backup.* July 2017. URL: https://insidegnss.com/reports-of-mass-gps-spoofing-attack-in-the-black-sea-strengthen-calls-for-pnt-backup/. (accessed: 01.03.2022).

[55] Zeroghan. *Android security update disables GPS spoofing in Pokémon GO.* Mar. 2017. URL: https://pokemongohub.net/post/news/android-7-1-security-update-disables-gps-spoofing-pokemon-go/. (accessed: 01.03.2022).

[56] Ayjinka. *How to play Pokemon GO without moving on Android.* Dec. 2021. URL: https://devsjournal.com/pokemon-go-android-hack.html. (accessed: 01.03.2022).

[57] E. Key. "Techniques to Counter GPS Spoofing. Internal Memorandum". In: *MITRE Corporation, Bedord* (1995).

[58] John A. Volpe National Transportation Systems Center. "Vulnerability Assessment of the Transportation Infrastructure Relying on the Global Positioning Syst". In: Aug. 2001. URL: https://rosap.ntl.bts.gov/view/dot/8435.

[59] Anon. *Volpe GPS Vulnerability Report Recommendations & FAA Response.* URL: https://www.caasd.org/library/presentations/navigation_workshop/FAA_Responses.pdf. (accessed: 01.03.2022).

[60] Leader D. Scott. "Anti-Spoofing & Authenticated Signal Architectures for Civil Navigation Systems". In: Sept. 2003, pp. 1543–1552.

[61] "Assessing the Spoofing Threat". In: *GPS World* 20 (Jan. 2009), pp. 28–38. URL: https://www.gpsworld.com/defensesecurity-surveillanceassessing-spoofing-threat-3171/. (accessed: 01.03.2022).

[62] Spenser Hill. *How Uber Drivers Use Fake GPS Apps for Quicker Ride Requests.* Apr. 2020. URL: https://www.imyfone.com/change-location/fake-gps-on-uber/. (accessed: 01.03.2022).

[63] Spenser Hill. *GPS Spoofing: A Growing Problem for Uber.* June 2017. URL: https://soliddriver.com/GPS-Spoofing-A-Growing-Problem-for-Uber. (accessed: 01.03.2022).

[64] *NTP Pool Project.* URL: https://www.pool.ntp.org/zone/europe. (accessed: 21.03.2022).

[65] Shanta Rangaswamy and Sourabha Murthy. “An Overview of Network Time Protocol”. In: *High Technology Letters* 27 (July 2021). DOI: 10.37896/HTL27.6/3704.

[66] Nguyen Truong and Chansu Yu. “Investigating Latency in GNU Software Radio with USRP Embedded Series SDR Platform”. In: Oct. 2013, pp. 9–14. DOI: 10.1109/BWCCA.2013.11.

[67] Steigenberger P., Thoelert S., and Montenbruck O. “GNSS satellite transmit power and its impact on orbit determination”. In: *Journal of Geodesy* 92.353 (2018), pp. 609–624. DOI: 10.1007/s00190-017-1082-2.

[68] Willard A. Marquis and Daniel L. Reigh. “The GPS Block IIR and IIR-M Broadcast L-band Antenna Panel: Its Pattern and Performance”. In: *NAVIGATION* 62.4 (2015), pp. 329–347. DOI: https://doi.org/10.1002/navi.123. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/navi.123.

[69] *Satellite Navigation - GPS - Space Segment.* URL: https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/gps/spacesegments. (accessed: 09.05.2022).

[70] Rakesh Nayak Glenn MacGougan Gerard Lachapelle. “OVERVIEW OF GNSS SIGNAL DEGRADATION PHENOMENA”. In: *International Symposium on Kinematic Systems in Geodesy, Geomatics And Navigation* (2001), pp. 87–100. DOI: 10.1.1.595.4021.