

**University Institute Of Lisbon**

Department Of ISTA

# GPS Jamming and Spoofing using Software Defined Radio

Diogo Alexandre Martins da Silva

A Dissertation presented in partial fulfillment of the Requirements for the Degree of

#### Master in Telecommunications and Computer Engineering

**Supervisor**

Prof. Pedro Sebastião ISCTE-IUL

October 2017

## Resumo

O Sistema de Posicionamento Global (GPS) é actualmente o melhor exemplo e o mais usado Sistema de Navegação Global por Satélite (GNSS) e como maior parte dos veículos atuais dependem do posicionamento do GPS para terem autonomia, é importante perceber as vulnerabilidades do GPS e reconhecer as ameaças como o bloqueio e o jamming.

Equipamentos para bloquear e fazer spoofing podem ser comprados online a um preço acessível. A tecnologia Software Defined Radio (SDR) traz a flexibilidade e provoca uma ainda maior diminuição de custo.

Esta investigação tem como objectivo explorar esta tecnologia respondendo às seguintes questões:

* Existe uma combinação de forma de onda, potência e amplitude adequada para perturbar o sinal GPS?
* O bloqueio do GPS é igualmente fácil de realizar em software como é em hardware?
* Fazer spoofing ao GPS é mais fácil de realizar em software?

O desenvolvimento deste trabalho consiste em dois testes: um teste de bloqueio ao GPS onde o foco é perceber qual a combinação de forma de onda, potência e amplitude que consegue bloquear o sinal GPS e um teste de spoofing ao GPS onde o HackRF One é usado para "enganar" um telemóvel.

Os resultados do teste de bloqueio mostraram que o bloqueio ao GPS é fácil de realizar usando diferentes combinações, devido à baixa potência de sinal recebido dos receptores de GPS. O teste do spoofing provou que é possível fazer spoof a um telemóvel com uma configuração SDR com um custo acessível.

**Palavras-chave:** GPS, Jamming, Spoofing, Software Defined Radio, HackRF One

## Abstract

The GPS is currently the most widely used and best known example of a Global Navigation Satellite System (GNSS) and as a system relies on GPS positioning to make its next move, it is important to understand GPS vulnerabilities and acknowledge the threats like jamming and spoofing.

Jamming and spoofing equipment can be openly purchased online at a low cost. The Software Defined Radio (SDR) technology is also bringing the flexibility and cost efficiency to a whole new level.

This research aims to explore this technology by asking the following research questions:

* Is there a suitable combination of waveform, power and amplitude to disrupt the GPS signal?
* Is GPS jamming equally easy to accomplish in software as it is in hardware?
* Is GPS spoofing easier to accomplish in software?

The study consists of two tests: a GPS jamming test where focus is to realize what combination of waveform, power and amplitude is able to jam the GPS signal and a GPS spoofing test where a HackRF One is used to fool a mobile.

The results from the jamming test have shown that GPS jamming is easy to accomplish using different combinations as GPS signals have low received signal power. The spoofing test proved that it is capable to spoof a mobile with a fairly inexpensive SDR setup and freely available software.

**Keywords:** GPS, Jamming, Spoofing, Software Defined Radio, HackRF One

## Acknowledgements

I would like to thank everyone who followed me in my academic career, from friends to family. Everyone played an important part in helping to fulfill this achievement.

I would like to thank, in particular, to my grandfather who provided for me and acted like a father in these last five years. I also would like to thank my parents for their comprehension and caring.

Finally, a special thanks to my supervisor, professor Pedro Sebastião who pro- vided guidance and supervised the project and was always available to clarify the doubts arisen throughout this dissertation. Thanks for all the help and feedback.

Lisbon, 30th October 2017, Diogo Silva

# Contents

[Resumo](#_bookmark0) iii

[Abstract](#_bookmark1) v

[Acknowledgements](#_bookmark2) vii

[List Of Figures](#_bookmark3) xi

[List Of Tables](#_bookmark4) xiii

[Abbreviations](#_bookmark5) xv

1. [Introduction](#_bookmark6) 1
   1. [Scope And Ambition](#_bookmark7) 1
   2. [Objectives And Main Contributions](#_bookmark8) 2
   3. [Thesis Outline](#_bookmark9) 3
2. [Literature Review](#_bookmark10) 5
   1. [Overview Of GNSS](#_bookmark11) 5
      1. [GPS Ranging Codes](#_bookmark13) 6
      2. [GPS Navigation Message](#_bookmark14) 7
      3. [GPS Frequency Information](#_bookmark15) 8
      4. [GPS Signal Power](#_bookmark18) 9
   2. [Radio Frequency Interference](#_bookmark19) 9
      1. [GPS Jamming](#_bookmark20) 10
      2. [GPS Spoofing](#_bookmark22) 12
   3. [Jamming Impact On GNSS Receivers](#_bookmark23) 13
      1. [Impact On The Front-End Stage](#_bookmark24) 14
      2. [Impact On The Acquisition Stage](#_bookmark25) 14
      3. [Impact On The Tracking Stage](#_bookmark26) 14
      4. [Impact On The Position](#_bookmark27) 15
   4. [Research On GPS Jamming Using Hardware](#_bookmark28) 15
      1. [Receivers Ability To Determine Position](#_bookmark29) 15
      2. [Jamming-to-Signal Ratio](#_bookmark31) 17
      3. [Carrier-to-Noise Density Ratio](#_bookmark33) 18
      4. [Conclusions](#_bookmark37) 20

Contents

* 1. [Research On GPS Spoofing Using Hardware](#_bookmark38) 21
     1. [Spoofing A Truck](#_bookmark39) 21
     2. [Spoofing A Drone](#_bookmark40) 22
     3. [Conclusions](#_bookmark42) 22
  2. [GPS Jamming And Spoofing On A Software View](#_bookmark43) 23

1. [HackRF One And SDR In GNU Radio](#_bookmark48) 27
   1. [HackRF One Main Description](#_bookmark49) 27
   2. [SDR In GNU Radio](#_bookmark55) 30
   3. [Case Study](#_bookmark57) 31
2. [Experiments And Results](#_bookmark69) 39
   1. [GPS Jamming](#_bookmark70) 39
      1. [Flow Graph](#_bookmark71) 40
      2. [Cosine Waveform](#_bookmark73) 41
      3. [Square Waveform](#_bookmark77) 41
      4. [Triangle Waveform](#_bookmark81) 42
      5. [Saw Tooth Waveform](#_bookmark85) 43
      6. [Additive White Gaussian Noise](#_bookmark89) 43
      7. [Results](#_bookmark94) 44
   2. [GPS Spoofing](#_bookmark99) 46
      1. [Generating The GPS Signal](#_bookmark100) 46
      2. [Transmitting The GPS Signal](#_bookmark105) 48
      3. [Results](#_bookmark109) 49
3. [Conclusions And Future Work](#_bookmark116) 51
   1. [Conclusions](#_bookmark117) 51
   2. [Future Work](#_bookmark118) 52

[Appendices](#_bookmark119) 57

1. [GNSS Acquisition In The Absence Of Interference](#_bookmark119) 57
2. [GNSS Acquisition In The Presence Of Interference](#_bookmark124) 61

[Bibliography](#_bookmark126) 65

**List of Figures**

* 1. [GPS Triangulation](#_bookmark12) 6
  2. [Power Spectra Of GPS Signals On L2 And L1](#_bookmark17) 9
  3. [Jammers Classification](#_bookmark21) 12
  4. [The Ability Of GPS Receivers To Resist Jamming](#_bookmark30) 16
  5. [The Effect Of Various Jammers On GPS Receivers](#_bookmark32) 17
  6. [C/N0 For Ipex SW Receiver And The Theoretical Curve](#_bookmark34) 18
  7. [Accuracy For Ipex SW Receiver](#_bookmark35) 19
  8. [C/N0 As Function Of Interference Power Level](#_bookmark36) 20
  9. [The Hornet Mini UAV](#_bookmark41) 22
  10. [SDR Equipment](#_bookmark44) 24

[3.1 HackRF One View](#_bookmark51) 28

[3.2 ANT 500](#_bookmark54) 29

* 1. [GNU Radio Flow Graph](#_bookmark56) 31
  2. [Receiving A Signal From The Remote Control](#_bookmark58) 32
  3. [QT GUI Frequency Sink](#_bookmark59) 32
  4. [Saving The Waveform To A File](#_bookmark60) 33
  5. [Replaying The Saved Waveform](#_bookmark61) 33
  6. [Throttle Block Disabled](#_bookmark62) 34
  7. [QT GUI Time Sink](#_bookmark63) 35
  8. [Background Noise](#_bookmark64) 36
  9. [Burst Of Activity](#_bookmark65) 36
  10. [Multiply Const](#_bookmark66) 37
  11. [Amplified Signal](#_bookmark67) 37
  12. [Transmitting The Amplified Signal](#_bookmark68) 38
  13. [Signal Transmission Flow Graph](#_bookmark72) 40
  14. [Cosine Waveform](#_bookmark74) 41
  15. [Square Waveform](#_bookmark78) 42
  16. [Triangle Waveform](#_bookmark82) 42
  17. [Saw Tooth Waveform](#_bookmark86) 43
  18. [Noise Source Block](#_bookmark90) 43
  19. [Gaussian Noise](#_bookmark91) 44
  20. [Loss Of Signal](#_bookmark98) 45
  21. [Compiling GPS-SDR-SIM](#_bookmark101) 46
  22. [Generating GPS Signal](#_bookmark102) 46

*List Of Figures*

* 1. [GPS Signal Generated](#_bookmark104) 48
  2. [Transmission Command](#_bookmark106) 48
  3. [Transmitting GPS Signal](#_bookmark108) 49
  4. [Initial Position](#_bookmark110) 50
  5. [Final Position](#_bookmark113) 50

[A.1 BPSK](#_bookmark121) 58

# List of Tables

[2.1 GPS Frequencies](#_bookmark16) 8

[3.1 HackRF One Features](#_bookmark50) 28

* 1. [Different Waveforms And Amplitudes](#_bookmark95) 44
  2. [Total Transmitted Power](#_bookmark97) 45
  3. [GPS-SDR-SIM Options](#_bookmark103) 47
  4. [Hackrf\_Transfer Options](#_bookmark107) 49

# Abbreviations

**AWGN A**dditive **W**hite **G**aussian **N**oise **A/D A**nalog to **D**igital

**ADC A**nalog to **D**igital **C**onverter **AGC A**utomatic **G**ain **C**ontrol **BB B**ase **B**and

**BPSK B**inary **P**hase for **S**hift **K**ey **BER B**it **E**rror **R**ate

**C/***N*0 **C**arrier-to-**N**oise Density Ratio

**C/A C**oarse/ **A**cquisition

**CDMA C**ode **D**ivision **M**ultiple **A**ccess **CW C**ontinuous **W**ave

**CWI C**ontinuous **W**ave **I**nterference **CAF C**ross-**A**mbiguity **F**unction

**DECT D**igital **E**nhanced **C**ordless **T**elecommunications **DSP D**igital **S**ignal **P**rocessing

**DS-SS D**irect **S**equence **S**pread **S**pectrum **GPS G**lobal **P**ositioning **S**ystem

**GNSS G**lobal **N**avigation **S**atellite **S**ystem

**GSM G**lobal **S**ystem for **M**obile **C**ommunications **I/Q I**n-Phase **Q**uadrature

**I.I.D I**ndependent and **I**dentically **D**istributed **ILS I**nstrument **L**anding **S**ystem

**IF i**ntermediate **f** requency **J/S J**amming-to-**S**ignal Ratio

*Abbreviations*

**LTE L**ong **T**erm **E**volution

**PVT P**osition, **V**elocity and **T**ime **PSD P**ower **S**pectral **D**ensity **PPS P**recise **P**ositioning **S**ervice **P(Y) P**recision **C**ode

**PDF P**robability **D**ensity **F**unction **PRN P**seudo**R**andom **N**oise

**R/C R**adio-**C**ontrolled **RF R**adio **F**requency

**RFI R**adio **F**requency **I**nterference **SDR S**oftware **D**efined **R**adio

**SPS S**tandard **P**ositioning **S**ervice **SMA S**ub**M**iniature Version **A**

**TF T**ime-**F**requency **UHF U**ltra **H**igh **F**requency

**UMTS U**niversal **M**obile **T**elecommunication **S**ystem **VHF V**ery **H**igh **F**requency

**VOR V**HF **O**mni-directional **R**adio-range **WSS W**ide **S**ense **S**tationary

# Chapter 1 Introduction

In this first chapter, it is going to be explained the ambit and motivations that led to the development of this dissertation as well as the objectives that are to be accomplished and lastly the way the dissertation is structured.

### Scope And Ambition

Nowadays most of the systems rely on Global Navigation Satellite System (GNSS) to navigate. The Global Positioning System (GPS) is currently the most widely used and best-known example of GNSS.

One main concern is that many of these systems have a shared dependency with the GPS, which means that any attempt to sabotage its service may lead to a simultaneous failure on these systems, which are expected to be independent [[1].](#_bookmark127) Missiles, ships, cars, aircrafts and drones all rely on the positioning provided by the GPS to make their next move.

In a world where GPS is vital, we must understand the GPS vulnerabilities and acknowledge the threats. GPS signals can be easily disrupted because they have a low received signal power. It is also important to acknowledge that the

1

only solution is to develop counter-mechanisms and technologies to detect and eliminate these threats.

Regarding the threats that expose the GPS vulnerabilities, two of them stand out: jamming and spoofing. Jamming is a real threat as it is very easy to deny GPS positioning. But it is no match to the effect and destruction that spoofing can cause. Spoofing however is harder to accomplish as it is needed a fully recreation of the GPS signal.

Since the cost and size of jamming and spoofing equipment has been falling and nowadays the technology can be openly purchased online, these threats are no longer a military capability only. For those with certain technical skills, there are several websites which describe in detail how to build this equipment and provide detailed designs [[2].](#_bookmark128)

The frequent advances in technology are also bringing hardware equipment to a software level and the Software Defined Radio (SDR) technology is gaining more recognition as it brings the flexibility and cost efficiency to a whole new level.

### Objectives And Main Contributions

After acknowledging and reporting the existence of various tests regarding GPS jamming and spoofing using hardware equipment, the main goal is to explore it on a software level.

This dissertation takes in consideration a SDR equipment, that can be used to transmit or receive signals, the SDR platform in GNU Radio, who allows to transmit in a frequency of choice and a test equipment, so it is capable to check if the jamming and spoofing are being successful.

On an early stage, there is an analysis of the chosen SDR equipment and its features and an understanding of how it works and how it can be used together with the SDR, to basically create a jammer controlled by software only.

Chapter 1. *Introduction*

They are two main tests whose evaluation methods are based on analyzing the test equipment behavior regarding the power, amplitude and waveform of the signal, by checking if the equipment affected by the SDR equipment works or not. One refers to GPS jamming and the other to GPS spoofing. The main objective is to expose that the GPS is a weak signal who is easily disrupted in a software environment, which makes it practically doable for everyone who owns a SDR equipment.

The tests will seek to obtain answers to certain questions like:

* + - Is there a suitable combination of waveform, power and amplitude to disrupt the GPS signal?
    - Is GPS jamming equally easy to accomplish in software as it is in hardware?
    - Is GPS spoofing easier to accomplish in software?

With the use of the SDR equipment and platform, the intention is to pro- vide a contribution, regarding the SDR technology, in the configuration of radio equipment in software, in particular, equipment able to disrupt the GPS signal. An article, which describes part of the work done in this dissertation, is being developed with the objective to be published in a future conference.

### Thesis Outline

Chapter 2 is the literature review and overviews the GNSS, focusing on the GPS. Characterizes radio frequency interference, focusing on jamming and spoofing. Reports relevant research on GPS jamming and spoofing on a hardware view. Introduces GPS jamming and spoofing on a software view and presents the available SDR equipment in the market choosing the one used for the software tests.

Chapter 3 analyses the GNSS signal model in a more theoretical way, with and without interference, focusing also on different types of interference. Jamming impact on the various GNSS receiver stages is also briefly analyzed.

3

Chapter 4 introduces jamming and spoofing in a software view by presenting the HackRF One, the chosen SDR equipment, which is capable of receiving and transmitting radio signals created in software. The concepts SDR and GNU Radio are also introduced and interconnected with the device. A case study is presented with the purpose of getting to know how the HackRF One works.

Chapter 5 consists of own experiments and results. The first test refers to GPS jamming and consists on the HackRF One transmitting different waveforms with different amplitudes aimed at the GPS signal in order to jam it. The second test refers to GPS spoofing where the HackRF One is used to transmit an incorrect GPS signal near a mobile in order to deceive it and get a fix on a wrong location.

Chapter 6 presents the main conclusions to the GPS signal jamming and the GPS spoofing tests’ results. There is a acknowledgement that the transition from hardware to software makes the GPS even more susceptible to threats that assuredly can be created by everyone. Recommendations for future works are also addressed.

# Chapter 2 Literature Review

In this chapter, there is a brief overview of the GNSS systems with the main focus on the GPS. Radio frequency interference is characterized, focusing on jamming and spoofing. A brief review of how jamming can affect the GNSS receiver stages is also addressed. Relevant jamming and spoofing research are reported and finally, the SDR technology is presented with a choice being made regarding the SDR equipment that is going to be used for the tests.

### 2.1 Overview Of GNSS

Global Navigation Satellite System is the standard generic term for satellite navigation systems that provide signals from space transmitting positioning and timing data, with global coverage. Nowadays there are two main GNSSs: GPS from United States and Glonass from Russia. China is expanding their regional Beidou and the European Union has Galileo in development. They all work almost the same way [[3].](#_bookmark129) This dissertation focuses on the GPS signal and so, the other GNSSs are not going to be described in detail. Also the GNSS acquisition in the absence and in the presence of interference is analyzed in detail in [[4]](#_bookmark130) and presented in Appendix [A](#_bookmark119) and [B,](#_bookmark124) respectively.

5

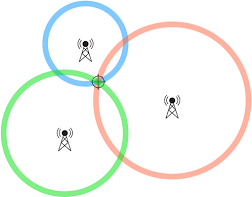
The GPS signal reaches a GPS receiver from a series of earth orbiting satellites to determine positioning via triangulation, which is a method in which three separate points are measured to calculate a location with an accuracy of only a few meters, as shown in Figure [2.1.](#_bookmark12)

Figure 2.1: GPS Triangulation

#### 2.1.1 GPS Ranging Codes

There are two ranging codes that are used in the original GPS: The Coarse/Acquisition (C/A) and the Precision (P)-code. The first is used by civilian users and refers to the Standard Positioning Service (SPS) and the second by users authorized by the US Department of Defense and refers to the Precise Positioning Service (PPS). Being a military signal, to prevent unauthorized users from spoofing the signal, the P-code was encrypted by being modulated with a special encryption secret sequence named W-code, generating the Y-code. The encrypted signal is referred to as the P(Y) code [[5].](#_bookmark131)

The C/A code is a 1*.*023 bit long pseudorandom noise (PRN) code which repeats every millisecond and is transmitted at 1*.*023 Mbit/s. The P(Y) code is a 6*.*1871 × 1012 bit long PRN code which repeats once a week and is transmitted at

10*.*23 Mbit/s. Although the C/A’s PRN is particular for each satellite, the P(Y)

code’s PRN is a small segment of the master P(Y) code approximately 2*.*35 × 1014 bits in length and each satellite repeatedly transmits its assigned segment of the master code.

In order to allow the receiver to recognize various satellites using the same frequency without mutual interference, the GPS signal uses a Code Division Multiple Access (CDMA) spread-spectrum technique. Each satellite transmits a unique PRN code, which does not correlate well with any other satellite’s PRN code, spreading the signal over a wide bandwidth (2 MHz for the C/A code and 20MHz for the P(Y) code). The use of different PRN sequences enables multiple satellites to transmit signals simultaneously at the same frequency. A receiver can distinguish among these signals based on their different codes [[6].](#_bookmark132) This process is called Direct Sequence Spread Spectrum (DS-SS). Also, it is argued in [[7]](#_bookmark133) that the reason to use DS-SS is its ability to combat radio frequency interference.

#### GPS Navigation Message

The receiver also needs to know detailed information regarding the satellites’ position and network and this information is modulated at 50 bit/s on top of both the C/A and P(Y) codes and it is called the Navigation message. It is constructed from a 1*.*500-bit frame, divided in 5 subframes of 300 bits who require 6 seconds to transmit each.

There are three major components in the navigation message. The first contains the satellite’s status and health and the GPS data and time and are located in the subframe 1. The second refers to the subframe 2 and 3 which together contain the ephemeris data which allows the receiver to calculate the position of the satellite. The third contains the information and status concerning all satellites in the constellation, their locations and PRN numbers and is called the almanac. Subframes 4 and 5 only contain 1/25th of the total almanac message each, meaning the receiver must process 25 whole frames worth of data to retrieve the entire 15000-bit almanac message, requiring 12.5 minutes to receive it from a single satellite [[5].](#_bookmark131)

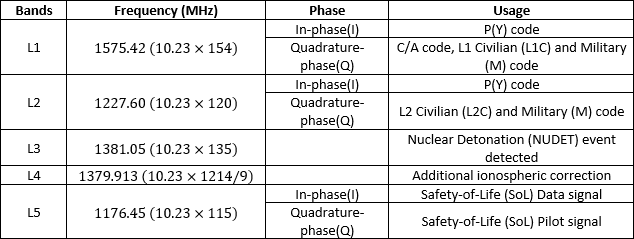
Although the ephemeris information is detailed and considered valid for no more than four hours, the almanac information is more general and valid for up

to 180 days. In order to calculate a position fix using any satellite, the receiver needs to have an accurate and complete copy of the ephemeris data from that satellite. The ephemeris data can be downloaded directly after the receiver picks up that satellite’s signal turn. The almanac assists the receiver in determining which satellites to search for.

#### GPS Frequency Information

The GPS signal must be modulated onto a carrier frequency to travel from the satellite to the receiver. The original GPS design utilizes two frequencies: one at 1575*.*42 MHz (10*.*23 MHz ×154) called L1 band and a second at 1227*.*60 MHz (10*.*23 MHz ×120), called L2 band. Additional advances in technology and new demands on the existent system led to the modernization of the GPS system and more frequencies are being utilized forming new bands. Table 2.1 lists the bands and frequencies of the recent GPS system as well as their use.

Table 2.1: GPS Frequencies



The C/A code is transmitted on the L1 band as a 1*.*023 MHz signal using a Bi- Phase Shift Key (BPSK) modulation technique. The P(Y) code is transmitted on both the L1 and L2 bands as a 10*.*23 MHz signal using the same BPSK modulation, however the P(Y) code carrier is in quadrature with the C/A carrier; meaning it is 90o out of phase.

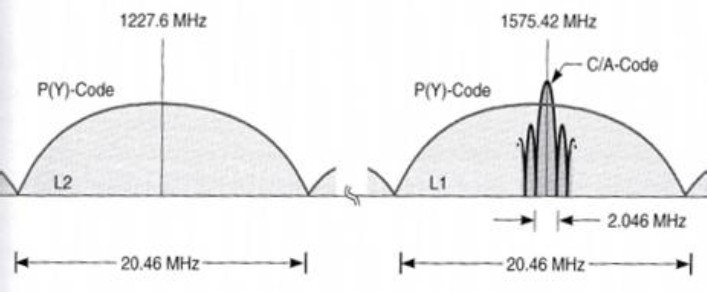
Figure [2.2](#_bookmark17) shows the C/A code and the P(Y) code both being spread centered at the carrier frequency. This allows the power spectral density to be reduced while the signal power is unchanged.

Figure 2.2: Power Spectra Of GPS Signals On L2 And L1 [[7]](#_bookmark133)

#### GPS Signal Power

When comparing to the signals generated on the surface, the GPS signals are very weak. Looking at the antenna input port of a satellite, its radio frequency (RF) power is about 50 W and, as the satellite antenna spreads the RF signal evenly over the surface of the earth, the transmitted power is attenuated. This happens mainly due to the signal transmission path loss, because the transmitted power decays with the distance squared as it travels from its orbit to the user [[7].](#_bookmark133)

The minimum received power level for the users on the earth is −158*.*5 dBW for the C/A code on L1 and −160 dBW for the P(Y) code on L2 according to the GPS specifications [[7].](#_bookmark133) It is equivalent to 10*−*6 which is well below the background RF noise level sensed by an antenna on the receiver. It is stated in [[7]](#_bookmark133) and [[8]](#_bookmark134) that "the extremely low signal power that reaches the receiver is the Achilles’ heel of GPS".

### 2.2 Radio Frequency Interference

It is considered interference any radio frequency from any undesired source that is received by a GNSS receiver. Such interference can result in degraded navigation accuracy or complete loss of receiver tracking [[6].](#_bookmark132)

According to [[3]](#_bookmark129) and [[9]](#_bookmark135) there is unintentional and intentional interference. Unintentional interference is referred to interference coming from other broadcast televisions, Very High Frequency (VHF) transmitters and personal electronic de- vices. GPS is susceptible to this kind of interference and to ionospheric effects and signal blockage. Its effects were most noticeable to SPS users who use single frequency. Intentional interference, on the other hand, can be classified as jamming, spoofing and meaconing. This dissertation does not approach meaconing.

Radio frequency interference (RFI) can be pulsed or continuous. GPS tolerates pulsed RFI better than continuous RFI because the pulses are shorter than the duration of a GPS data bit. According to [[10],](#_bookmark136) continuous RFI can be classified and differentiated into broadband and narrowband bandwidth. A broadband RFI has a bandwidth equal or greater than the GNSS bandwidth, which is 2 MHz for the GPS C/A code, while the narrowband RFI has lesser. A continuous wave (CW) signal consists of a single tone which is concentrated in a very narrow band around the center frequencies and it is the simplest form of interference.

#### GPS Jamming

GPS jamming is defined in [[10](#_bookmark136)] as “the emission of radio frequency energy of sufficient power and with the proper characteristics to prevent receivers in the target area from tracking the GPS signals”. Jamming signals can be characterized by its center frequency and by its power described as jamming-to-signal ratio (*J/S*) in dB. The *J/S* decreases with the distance from the jammer to the receiver.

Jamming will result in a decreased measured signal strength for the receivers. The signal strength is described by the carrier-to-noise density ratio (*C/N*0) which, according to [[3],](#_bookmark129) is the fundamental navigation signal quality parameter at the receiver. The equation *C/N*0 is defined and analyzed with detail in [[6]](#_bookmark132) and shows that: the received jammer power increases as the distance between the jammer and the receiver gets smaller; the signal strength will decline as the jamming occurs

and a higher PRN code rate, in theory, will lead to a lower decrease in the signal strength as the jamming is occurring.

They are different classifications suggested for civilian jammers. In [[11]](#_bookmark137) jammers are divided according to their jamming signal type:

* + - * class I: CW signals; the jammer transmits a CW signal;
      * class II: single saw-tooth chirp signals; the jammer transmits a frequency- modulated signal with a saw-tooth time-frequency (TF) evolution;
      * class III: multi-saw-tooth chirp signals; the device transmits a frequency- modulated signal but its TF evolution is more complex, and it is determined

by the combination of several saw-tooth functions;

* + - * class IV: chirp with signal frequency bursts; the device transmits a frequency- modulated signal and frequency bursts are used to enlarge the frequency band affected by the disturbing signal.

Experiments made in [[12]](#_bookmark138) and [[13]](#_bookmark139) group jammers based on power source and antenna type:

* + - * Group 1 - Jammers designed to plug into a 12 V car cigarette lighter socket. These jammers usually have low transmitting power (below 100 mW) and possibility to connect an external antenna.
      * Group 2 - Jammers powered by battery and equipped by an external antenna

connected via an SubMiniature version A (SMA) connector. Some of the jammers can transmit on both the L1 and L2 frequency bands, and the transmit power is up to 1 W.

* + - * Group 3 - Jammers disguised as a harmless electronic device, for example

cell phones. They have no external antenna and these jammers normally use saw-tooth frequency modulation.

Figure [2.3](#_bookmark21) shows the combination of both classifications of the main civilian jammers in the market.

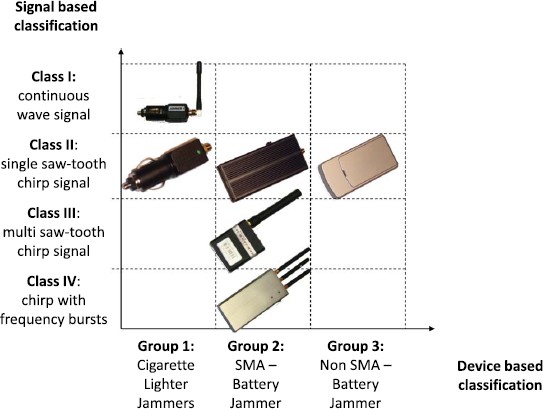


Figure 2.3: Jammers Classification [[14]](#_bookmark140)

A study done in [[12]](#_bookmark138) together with [[13]](#_bookmark139) shows that most jamming devices generate broadband interference. It is stated in [[15]](#_bookmark141) while referring to [[9]](#_bookmark135) and [[12]](#_bookmark138) that more advanced jammers than those described above are about to appear on the market.

#### GPS Spoofing

Spoofing is defined in [[10]](#_bookmark136) as "a technique that has long been used to deceive a radar’s target-ranging operation. In the case of GPS, the intent is to cause an active GPS receiver (whether or not presently tracking GPS signals) to lock onto legitimate-appearing false signals". Although the GPS’s P(Y) code is encrypted and thus hard to spoof, the signal structure, the codes and modulation of the C/A code are open to public making it easier to spoof.

According to [[16],](#_bookmark142) spoofing attacks can be grouped into: simplistic, intermediate and sophisticated attacks.

In the simplistic attack, there is no need to know the victim’s original Position, Velocity and Time (PVT) as the GPS simulator is used to broadcast GPS signals for the spoofed position to the GPS receiver under attack. This form of attack

can be easily detected if pseudorange, *C/N*0 and Doppler jumps, which will occur, are monitored.

In the intermediate attack, the spoofer gains information of the victims original PVT and aims to replicate and broadcast the GPS signal while moving towards the victim. The receiver will eventually lock on to the spoofed signal’s higher power without realizing and then the spoofer can go back to the victim’s PVT arbitrary value. One way to detect it is to monitor the Doppler and pseudo range variations when moving the victim’s receiver antenna.

The same process as before happens in the sophisticated attack but now several coordinated spoofers are used to attempt to emulate the spatial signal domain requiring multiple transmitting antennas to successfully accomplish it. This proves very hard to detect for a conventional single antenna receiver [[17].](#_bookmark143)

Most of the commercial spoofers do not attack the GNSS signal itself, they insert the spoofed information directly at the receiver’s output and that requires a physical or a software access to the victim’s receiver. RF-signal spoofers are much less common and commercial GNSS RF-signal generators cost minimum 100*.*000 euros. Although they are not capable of performing a intermediate spoofing at- tack they can disturb a standard GNSS receiver significantly. SimSAFE software together with GNSS RF-signal generator is probably the first ready-to-buy commercial GNSS spoofer [[18]](#_bookmark144) which supports spoofing attacks on the GPS L1 and the Galileo E1 bands. A commercial GNSS receiver tracks the current signals from the sky and uses this information to generate an appropriate spoofing signal out of the GNSS RF simulator.

### 2.3 Jamming Impact On GNSS Receivers

Jammers broadcasting a strong power are easily detectable. The ones broadcasting an intermediate power are the more dangerous, as they can decrease the receiver’s performance without making it lose lock or prevent the acquisition of the satellite

signals. The jamming can affect different stages of a GNSS receiver which are briefly described below. A more detailed analysis of this impact, complemented with experiences and tests is presented in [[1](#_bookmark140)4].

#### Impact On The Front-End Stage

The first receiver stage that can be affected is the front-end, which has the goal to filter the incoming signal in the desired bandwidth and downconverting it to the chosen IF before performing the A/D conversion.

Modern receivers have an Automatic gain control (AGC) between the analog portion of the front-end and the ADC and jamming impacts the AGC values and modifies the distribution of the samples at the output of the ADC. This leads to several elements of the front-end to work outside their nominal regions [[14].](#_bookmark140)

#### Impact On The Acquisition Stage

The acquisition block is the first digital signal processing stage of a GNSS receiver and has to determine the signal presence and to provide a rough estimate of the signal code delay and Doppler frequency [[6].](#_bookmark132)

In the presence of interference, the probability of erroneously declaring the signal presence increases and the acquisition block may provide erroneous Doppler and delay estimates. The effects of CWI on the acquisition blocks are analyzed in detail in [[19]](#_bookmark145) and an extensive study on the effects of several kinds of interference on the acquisition probabilities can be found in [[20].](#_bookmark146)

#### Impact On The Tracking Stage

The tracking block is responsible for presenting fine estimates of the signal parameters. GNSS measurements such as pseudoranges, carrier phases and Doppler shifts are generated by these estimates. Jamming will affect the measurements’

quality causing increased measurement variances, biases, and measurement outliers [[20],](#_bookmark146) [[21].](#_bookmark147)

Different effects can happen depending on the power received and on the type of jamming signal. In most cases, an increased bit error rate (BER) can occur and in the worst cases, the receiver is unable to decode the navigation message [[14].](#_bookmark140)

#### Impact On The Position

If the interfered signal is processed by both acquisition and tracking stages, the GNSS receiver is still able to output an estimate of the position, except this esti- mate will be degraded as based on pseudoranges affected by the interference.

An universal rule to quantify the performance degradation according to positioning error is not defined as it depends on the operation position algorithm. Depending on the *J/S*, the jamming signal can either degrade the position solution or generate total loss of lock of the GNSS signals [[14].](#_bookmark140)

### Research On GPS Jamming Using Hardware

The possibility to jam the GPS signals has been an awareness for many years for the military. The C/A code can be disrupted by low powered jammers over a great area. In 1994, a study done in [[22]](#_bookmark148) showed that a simple 1 W airborne jammer could prevent a receiver from tracking a GPS signal, which already had been locked-on at a distance of 10 km, for a distance of 85 km. The research on jamming civilian GPS signals has increased in recent years as low powered jammers have become easily available.

#### Receivers Ability To Determine Position

A study regarding the susceptibility of four different GPS receivers to jamming has been conducted in [[23]](#_bookmark149) to find how easy it is to jam the GPS signal. A standard

commercial RF signal generator generating a frequency modulation signal on the L1 band and passive GPS antennas are used. The signal output strength varies between −3 dBm (0*.*5 mW) and 17 dBm (50 mW).

An initial test shows that a jammer power of 13 dBm (20 mW) is able to disrupt all GPS receivers being moved away from the jammer till a distance of 2 km. Starting from that distance, the main test consists in having all receivers on the ground level, where they can establish a GPS position, and start increasing the jamming power until all receivers couldn’t establish a position anymore.

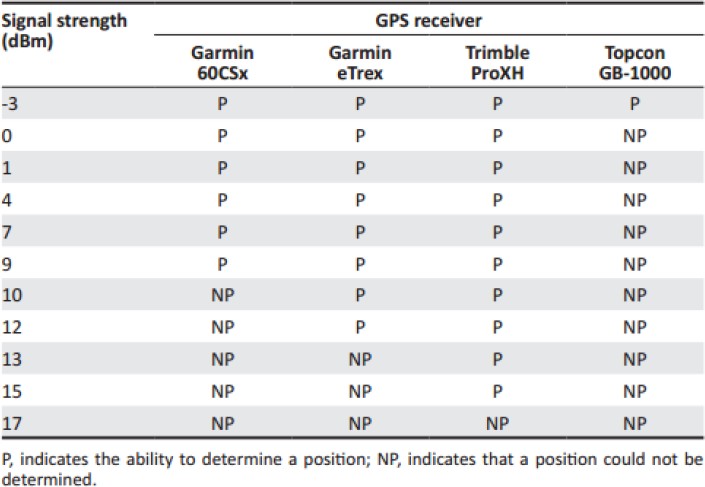


Figure 2.4: The Ability Of GPS Receivers To Resist Jamming [[23]](#_bookmark149)

Figure [2.4](#_bookmark30) shows that while the Trimble receiver has the most resistance to jamming, the Topcon receiver, which is the most sophisticated receiver of the four, has the least. Between the Garmin receivers the eTrex, the less sophisticated, has better results. The authors conclude that it is easy indeed to jam the GPS receivers and that "jamming remains a serious threat to the integrity of navigation that needs further investigation" [[23].](#_bookmark149)

One weakness that can be pointed out to this study is not analyzing the quality of the position determination throughout the different jamming signal strengths. Studying the performance when the jammer strength is close to the threshold before losing track to the satellites would be interesting.

#### Jamming-to-Signal Ratio

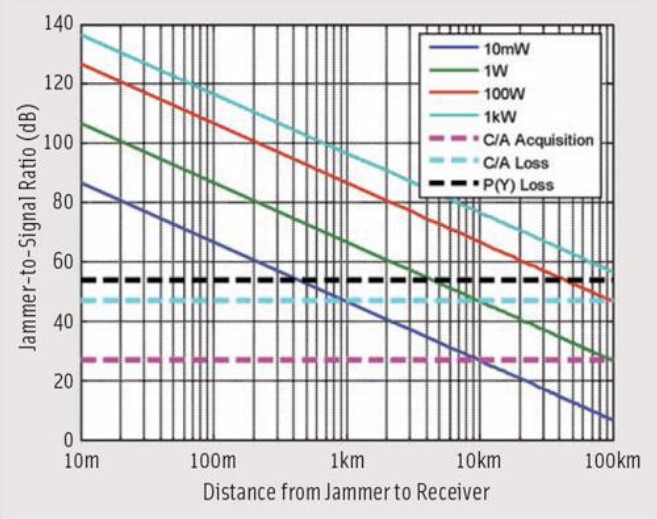
Studies have been conducted to prove how vulnerable consumer a grade GPS receiver is when analyzing the *J/S*. The maximum *J/S* refers to the amount of non GNSS interference the receivers can handle when they are still acquiring or tracking the desired signal. Figure [2.5](#_bookmark32) shows theoretically values plotted in [[24]](#_bookmark150) for different CW broadband jammers with power from 10 mW to 1 kW, where the horizontal dashed lines represent some typical receiver thresholds.

Figure 2.5: The Effect Of Various Jammers On GPS Receivers [[24]](#_bookmark150)

This study concludes that the more powerful the jammer, the bigger the distance able to prevent acquisition to the C/A code. A 10 mW jammer would cause the receiver to lose the lock up to 1 km whereas a 1 W jammer would cause it to lose up to 10 km. The findings regarding an airborne jammer in [[22]](#_bookmark148) support these values.

Other studies have been conducted to prove how vulnerable consumer grade GPS receivers are. A GPS L1 Jammer with the output power of 13 dBm has been used in [[25]](#_bookmark151) to jam 6 different receivers. The jammer transmits a chirp signal with multi saw tooth functions at 1577 MHz with a spectrum bandwidth of 16*.*3 MHz. The test has been conducted with three different cases: no jamming, maximum J/S set at 15 dB and set at 25 dB.

The study concludes, complementing the results from [[23],](#_bookmark149) that receivers react differently to jamming due to their internal processes and filtering techniques. With the J/S of 25 dB, the worst receiver gets a position fix 16 % of the time with a maximum error of 129 meters, while the best receiver gets a position fix all the time with a maximum error of 16 meters.

#### Carrier-to-Noise Density Ratio

Being a quality parameter at the receiver, various studies have been conducted focusing on the *C/N*0.

An open-field test in Germany has been conducted in [[26]](#_bookmark152) using a multi- frequency Ipex software GNSS receiver and a 0*.*1 mW (−40 dBW) jammer trans- mitting a chirp signal with a bandwidth of 11*.*8 MHz in the L1 band. The jammer belongs to the category broadband interference and starts approaching the receiver from a starting distance of 1200 meters. Figure [2.6](#_bookmark34) shows the *C/N*0 values over time comparing them to the theoretical effective *C/N*0.

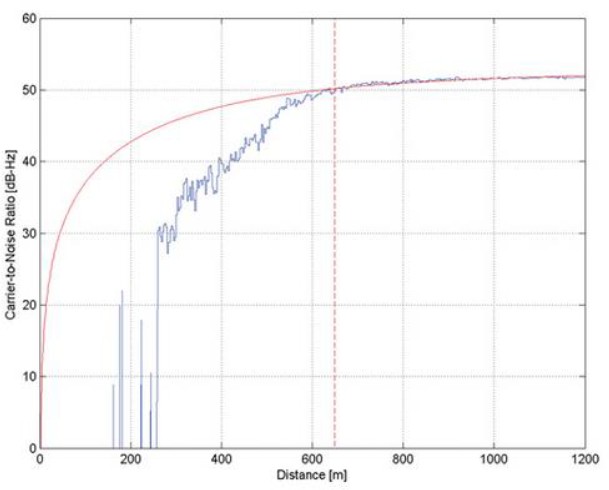


Figure 2.6: *C/N*0 For Ipex SW Receiver And The Theoretical Curve [[26]](#_bookmark152)

The theoretical and measured values match until the front end becomes saturated, which happens around 650 meters. Before the saturation, the jammer

degrades the correlation process by raising the noise floor causing positions error, as shown in Figure [2.7,](#_bookmark35) of more than 50 meters just before the receiver lost track.

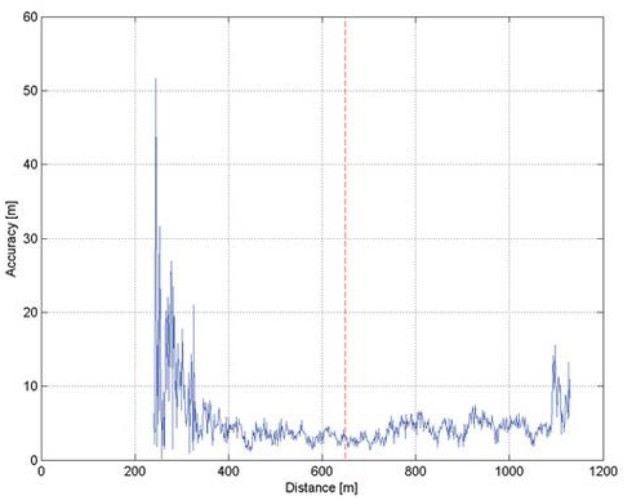


Figure 2.7: Accuracy For Ipex SW Receiver [[26]](#_bookmark152)

Authors conclude that "when the front-end analogue to digital converter (ADC) is saturated it causes heavily degradation of the signal which exceeds the pure degradation caused by the increased jammer power until loss of lock of signal” [[26].](#_bookmark152)

Other tests have been also conducted to survey grade and mass-market receivers and when compared with the professional receiver, it is found that the professional one is interfered at a shorter distance but loses lock on the signal earlier, concluding once again that interference range of a jammer is very dependent on the receiver architecture.

Another study has been done in [[11]](#_bookmark137) focusing on the *C/N*0 and also analyzing interference outside the L1 GPS band. The *C/N*0 is measured for the GPS L1 C/A signal to examine the commercially GPS receiver’s immunity to different interference sources. The interference sources used are: a CW signal and a broad- band Additive White Gaussian Noise (AWGN) signal (48 MHz bandwidth) both centered on the GPS L1 frequency; a Global System for Mobile Communications (GSM) centered at 900 MHz, a Digital Enhanced Cordless Telecommunications (DECT) and a Long Term Evolution (LTE) signal both centered at 1900 MHz, all outside the L1 frequency.

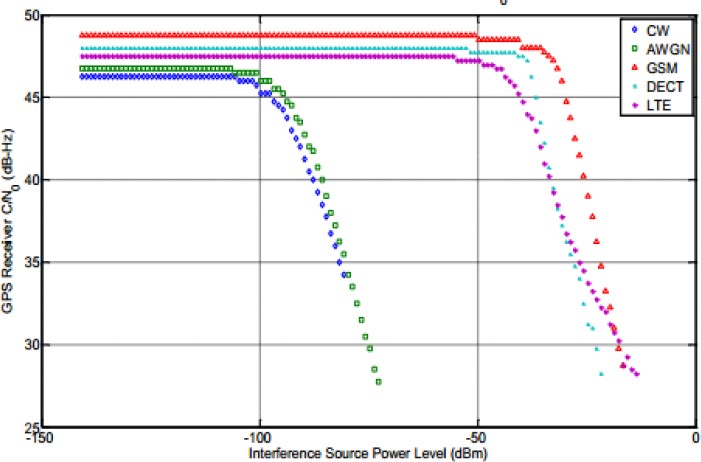


Figure 2.8: *C/N*0 As Function Of Interference Power Level [[11]](#_bookmark137)

Figure [2.8](#_bookmark36) shows how the receiver’s *C/N*0 varies according to each interference source. Like it is expected the in-band jamming sources cause more disruption than the out-band ones. The out-band sources need to be significantly higher to affect the receiver’s performance. Between the in-band sources, the CW interference causes the receiver to lose lock first than the AWGN broadband noise with the same *C/N*0.

It is concluded that the use of multiple GNSS frequencies provides jamming immunity because, with adjacent GNSS frequency bands, the intentional source which is targeted at a particular band will likely be less pronounced.

#### Conclusions

The reported research on GPS jamming concludes that it is easy to jam a civilian GNSS receiver on quite large distances. Jammers are easily available and most of them generate broadband interference [[12]](#_bookmark138) [[13].](#_bookmark139)

Several studies have been made where the signal properties of different GNSS jammers have been surveyed and their effect on receivers has been measured by use of GNSS simulators [[25]](#_bookmark151) [[27].](#_bookmark153) Real outdoor GPS jamming tests have also been conducted [[23]](#_bookmark149) [[26].](#_bookmark152) The findings from these studies are generally that the combination of high sensitivity GPS receivers and the low signal strength from the satellites make GPS receivers vulnerable to jamming.

It is also concluded that makes sense to study *C/N*0 under jamming conditions as it is an important quality indicator. The studies referred focus on the *C/N*0 ratios when the receiver is in tracking phase [[11]](#_bookmark137) [[26].](#_bookmark152) It would have been interesting to study the *C/N*0 ratio during the acquisition phase.

### Research On GPS Spoofing Using Hardware

Spoofing is a much more elegant attack than jamming because if multiple receivers cannot get a signal lock suddenly, suspicions are raised and counter measures can be applied before more damage is done. Spoofing however can go undetected, as the navigation system will not find anything faulty until the fatal moment.

#### Spoofing A Truck

A simple demonstration that GPS is vulnerable to spoofing has been conducted in [[28].](#_bookmark154) A GPS Satellite Simulator, which can overpower the GPS signal from space, is rented for about 1000 dollars a week in an attempt to create a hijacking scenario were an attack truck containing the GPS simulator attempts to spoof and redirect a target cargo truck containing a GPS receiver.

The attack sumps up in three main steps: first of all, the GPS receiver signal lock must be broken, which can be achieved with a GPS jammer or waiting for the truck to drive under some type of obstruction like a bridge or a tunnel. Secondly, the target truck must lock into the spoofed signal and this is achieved by approaching the truck and waiting for the GPS tracking device in the truck to lock onto the counterfeit signal. Finally, the attack truck needs to continue broadcasting the fake GPS signal and needs to stay close to the target truck in order to maintain the lock.

Authors conclude that with a stronger signal or signal amplifiers, the spoofing can happen over a greater distance, and that civilian GPS is indeed vulnerable to

simple spoofing attacks that almost everyone can exploit. They also suggest one inexpensive countermeasure is improving the clocks in the GPS receivers.

#### Spoofing A Drone

Another experiment has been conducted in [[29]](#_bookmark155) but this time demonstrating an effective over-the-air spoofing attack targeting a unmanned aerial vehicle (UAV), shown in Figure [2.9,](#_bookmark41) using a sophisticated GPS spoofer. The spoofer transmits simulated GPS signals from approximately 620 meters and capture the drone’s navigation system. It then induces the captured GPS receiver to mislead the UAV with false position and velocity solutions which are only stopped when a safety pilot takes over manual control of the aircraft.

Figure 2.9: The Hornet Mini UAV

The main conclusions from this experiment are that a GPS spoofer can alter a UAV’s perception of its location from a considerable distance away and the GPS receivers cannot acknowledge that the PVT solution is being overwritten. One downside from the experiment is that a long-term control of the UAV was not achieved, since the spoofer does not know the UAV’s real-time current position and velocity.

#### Conclusions

The reported research on GPS spoofing concludes that although it is an expensive and detailed process to obtain or build a GPS simulator to use in a spoofing attack, when successful, there is no real defense mechanisms implemented on most GPS receivers to prevent or even detect these attacks.

Studies creating a hijacking scenario using GPS dependent trucks and drones have been conducted in [[28]](#_bookmark154) and [[29]](#_bookmark155) which further highlight GPS spoofing as a dangerous threat.

It is imperative that more research and funds are devoted to develop and test practical and effective countermeasures.

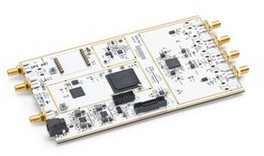
### 2.6 GPS Jamming And Spoofing On A Software View

On a hardware level, the expensive jammers available in the market only allow to jam a specific configured frequency. It is not possible to analyze that specific frequency neither to configure the equipment to a different frequency, which makes its use limited, not only by its configuration but also by its only frequency of operation. Having a jamming equipment that can be studied, analyzed and implemented to adapt to various frequencies brings a new world of possibilities where the GPS and all the other main technologies such as GSM, Universal Mo- bile Telecommunication System (UMTS) and LTE can be attacked using the same piece of equipment.

Spoofing often has require a sophisticated and expensive GPS simulator that costs thousands of dollars. Recreating the fully GPS signal has been a complex process only within range of professionals and experienced users. Nowadays the GPS signal can be emulated in open source projects and produced through a SDR equipment making the attack’s cost very low and much easier to accomplish.

A SDR equipment is defined, in general, as one with a wide frequency range and with no focus on any specific frequency. With all the SDR products popping up in the recent years a roundup is needed in order to decide the best SDR equipment suitable for the experiments that are going to be made. From analyzing various SDRs according to cost, frequency range, ADC resolution and capability of transmitting or receiving signals the choice falls between three products: the

HackRF One, the BladeRF and the USRP B200/B210, which are represented in Figure [2.10a,](#_bookmark45) Figure [2.10b](#_bookmark46) and Figure [2.10c](#_bookmark47) respectively.



(a) HackRF One (b) BladeRF (c) USRP B200/B210

Figure 2.10: SDR Equipment

The HackRF One is the cheapest of the three and it is one of the first low cost SDRs that appeared in the market, whose price is around 250 euros. It is capable of receiving and also transmitting. It is a half duplex transceiver which means it is necessary to switch between modes by command. Its main advantages are its transmit capabilities, wide bandwidth and massive frequency range (1 MHz to 6 GHz). The only downside is its small 8 bit resolution and poor RF design which affects signal-to-noise (SNR) performance.

The BladeRF is another capable transceiver SDR. There are two versions which cost around 355 euros and 550 euros and so, it is more expensive than the HackRF, but it can operate in full duplex which means it can receive and transmit simultaneously. It has a smaller frequency range (300 MHz to 3.8 GHz) compared to the HackRF, but has a greater ADC resolution. The 12 bit ADC makes it a better receiver than HackRF but misses out on the frequencies below 300 MHz, which can be received with a transverter for an extra cost.

The USRP B200/B210 is the most expensive of the three, as there are also two versions which cost around 570 euros and 930 euros. It is considered an advanced SDR aimed more towards the professional and research market. It has the possibility of transmitting and receiving in full duplex with two signals at the same time and has the same ADC resolution than the BladeRF. It has better frequency range than the BladeRF (70 MHz to 6 GHz) but it also misses out on the frequencies below 70 MHz.

Focusing mainly on the cost and the information available on the internet, the HackRF One is the chosen SDR equipment as it is the cheapest, offers the wider frequency range and has better support for starters. The HackRF One is described and analyzed in detail in Chapter [3.](#_bookmark48)

# Chapter 3

**HackRF One And SDR In GNU Radio**

This chapter presents in detail the HackRF One, its specifications and how it can be used as a test equipment for RF systems when integrated with SDR in GNU Radio. Finally, a case study is presented showing the HackRF One being used.

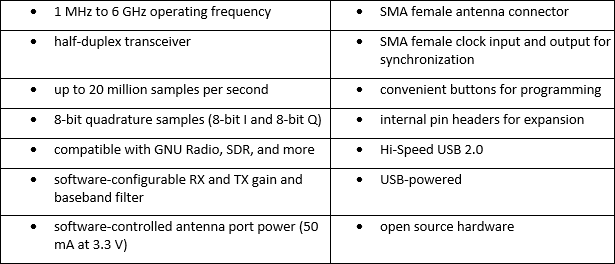
### HackRF One Main Description

"The HackRF One from Great Scott Gadgets is a Software Defined Radio peripheral capable of transmission or reception of radio signals from 1 MHz to 6 GHz. Designed to enable test and development of modern and next generation radio technologies, HackRF One is an open source hardware platform that can be used as a USB peripheral or programmed for stand-alone operation" [[30].](#_bookmark156)

To understand the features of the HackRF One, which are listed in Table [3.1,](#_bookmark50) and how to use them, first a overview of its buttons, LEDs and ports is needed. Figure [3.1a](#_bookmark52) and Figure [3.1b](#_bookmark53) show the HackRF One front and back view, respectively.

27

Table 3.1: HackRF One Features





* + 1. Front View (b) Back View

Figure 3.1: HackRF One View

The 3V3, 1V8 and RF LEDs are three different power supplies within the HackRF. They should all be on when receiving or transmitting a signal. If they are not all on, then that indicates a problem.

The USB LED indicates that the host computer has configured the HackRF One as a USB device.

As soon as the HackRF One is plugged in via USB the first three LEDs should turn on very quickly followed by the USB LED very shortly.

The final two LEDs are the RX and the TX which stand for receive operation and transmit operation respectively. One of them should be on whether the HackRF One is receiving or transmitting a radio signal.

The RESET button reboots the micro-controller within the HackRF One. This can also be done by unplugging and plugging back the USB power supply. It is just more quicker and easier to press the button.

The DFU button it is used for a firmware update mode, but it is not used that often as HackRF is able to update its own firmware without having to go into DFU mode. The reason there is a DFU mode is to be able to recover the HackRF One if a firmware update went wrong and it starts not working correctly. To enter DFU mode, hold down the DFU button as the HackRF One is plugged in or as the RESET button is pressed and released.

The ANTENNA port, the CLKIN and CLKOUT ports are all SMA connectors. The antenna port can be known as a RF port, as either an antenna or a cable going to some other RF equipment can be connected.

The antenna chosen to use with the HackRF One is the ANT 500, as shown in Figure [3.2,](#_bookmark54) which is just a simple telescopic antenna that can operate over a pretty wide range of frequencies and it has a SMA male connector that allows it to connect directly to the HackRF One without any adapters.



Figure 3.2: ANT 500

The clock in and clock out ports are for clock synchronization between multiple HackRF Ones or between one HackRF One and an external clock source. A SMA cable can be connected to one HackRF One’s clock out port and another HackRF One’s clock in port and both clocks will be synchronized, or a 10 MHz square wave signal is connected to the clock in and the HackRF One will automatically synchronize to that external clock. The signal should vary between 0 V and 3*.*3 V.

These ports have not been used as only one HackRF One is used in the experiments.

### SDR In GNU Radio

SDR is essentially a radio communication system where components that have been typically implemented in hardware are instead implemented by means of software.

It is the application of Digital Signal Processing (DSP) to radio waveforms. An analog signal is a continuous variant function over time like radio waves and sound waves for example. A digital signal is made up of discrete values at discrete points in a horizontal axis. In SDR that horizontal axis is often a time domain.

Just as a sound card in a computer digitizes audio waveforms, a software radio peripheral digitizes radio waveforms. It is like a very fast sound card with the speaker and microphone replaced by an antenna. The HackRF One acts in just this way, an all-in-one SDR in a small enclosure a little bigger than a cell phone.

In HackRF One there is a sampler or a ADC that takes an analog waveform, samples it or measures it at discrete period in time, then give us the value at each sample point. Every sample is a number so a digital signal is a sequence of numbers.

GNU Radio is a free and open-source software development toolkit that pro- vides signal processing blocks to implement software radios. It can be used with readily-available low-cost external RF hardware to create software-defined radios, or without hardware in a simulation-like environment.

The GNU Radio software [[31]](#_bookmark157) provides the framework and tools to build and run software radio or just general signal-processing applications. As with all SDR systems, reconfigurability is a key feature. Instead of using different radios de- signed for specific purposes, a single, general-purpose, radio can be used as the

radio front-end, and the signal-processing software handles the processing specific to the radio application.

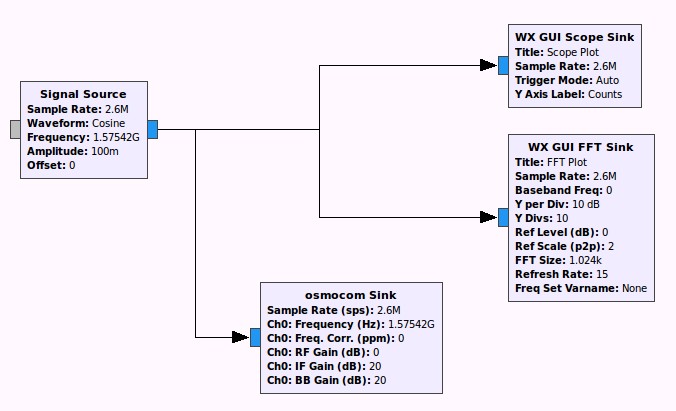
GNU Radio Companion is a front-end graphical interface to GNU Radio. It is a tool that actually automatically creates python programs that are software radio programs. The GNU Radio applications themselves are generally known as flow graphs, which are a series of signal processing blocks connected together.

Figure 3.3: GNU Radio Flow Graph

These flow graphs can be written in either C++ or the Python programming language. Figure [3.3](#_bookmark56) shows a mere visual example of such flow graphs. The GNU Radio infrastructure is written entirely in C++, and many of the user tools are written in Python.

### Case Study

This is a demonstration of a capture and replay using the HackRF One with the target device being a radio-controlled car. This experiment has been made by Michael Ossmann, the creator of the HackRF One [[32].](#_bookmark158) A signal is transmitted from its original transmitter and captured with the purpose of being replayed and see if the car responds.

To replay a signal, first it needs to be captured. A flow graph is created to receive a signal from the remote control, as shown in Figure [3.4](#_bookmark58) Sample rate is set

at 2 million samples per second, which is the lowest recommended when using the HackRF One.

On the osmocom source, the frequency is set at 27 MHz which is the frequency of operation present in a sticker of the R/C car. The RF gain is changed to 0 dB and the IF band and BB (baseband) gain to 16 dB which is a good starting point. The remaining parameters are kept at the default values.

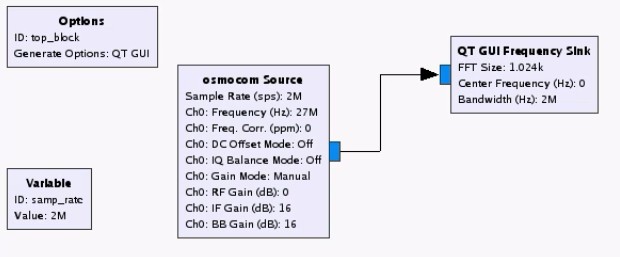
The QT GUI Frequency Sink shows the frequency domain view and allows to find the remote control’s frequency of operation.

Figure 3.4: Receiving A Signal From The Remote Control

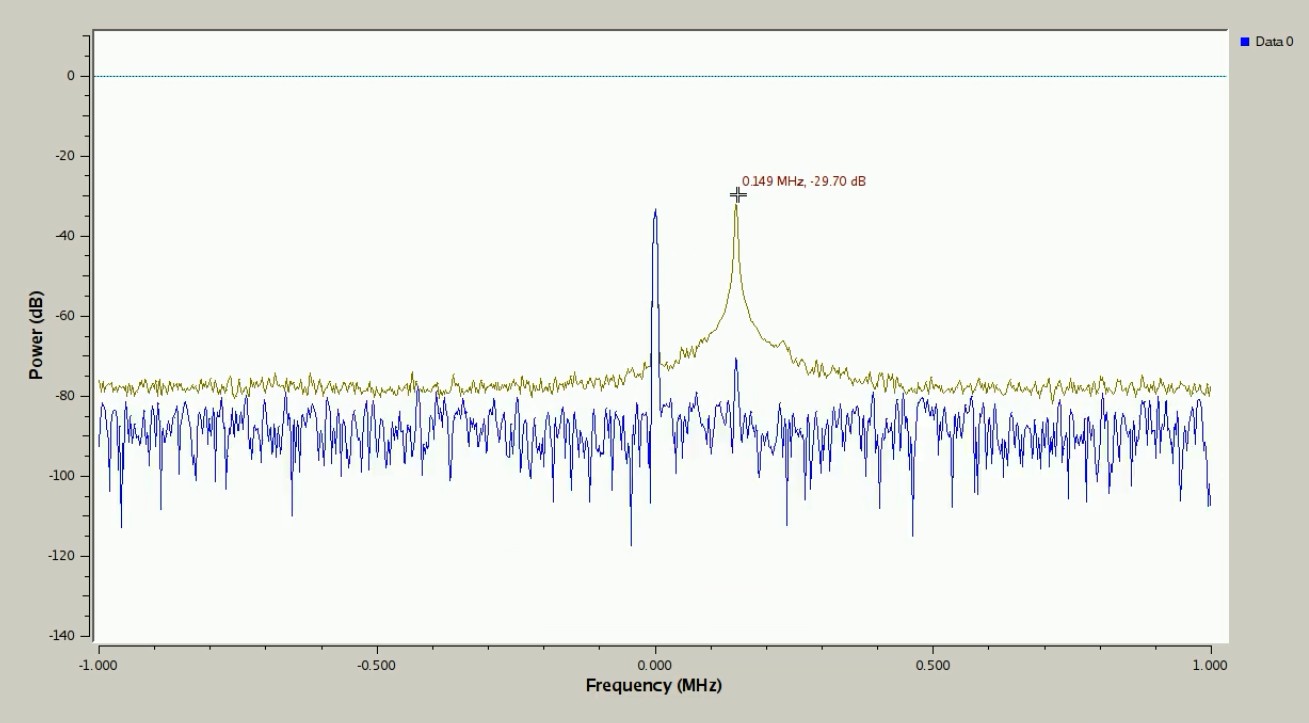
With max hold enabled, the QT GUI Frequency Sink shows a peak around 150 MHz above the 27 MHz when the remote control is used to move the car, as shown in Figure [3.5.](#_bookmark59) The radio-controlled (R/C) car’s frequency of operation has been found.

Figure 3.5: QT GUI Frequency Sink

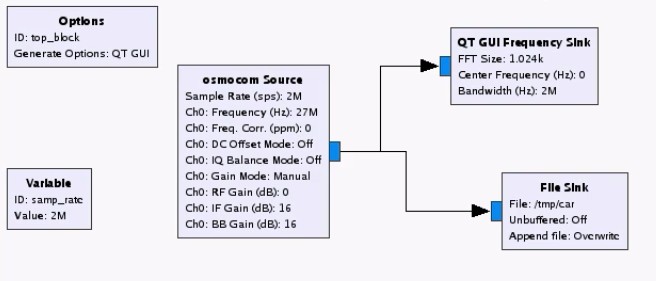
Now in order to save this waveform to be replayed in the future, a file sink is connected to the osmocom source. This is a raw digital waveform, it is the information coming from the HackRF One being saved directly to a file. The flow graph is running but now as the remote control is used to move the car, the waveform is saved to a file, as shown in Figure [3.6.](#_bookmark60)

Figure 3.6: Saving The Waveform To A File

In order to replay this waveform, a new flow graph must be created. Now instead of a file sink, a file source is used pointing to the waveform file created earlier, as shown in Figure [3.7](#_bookmark61).

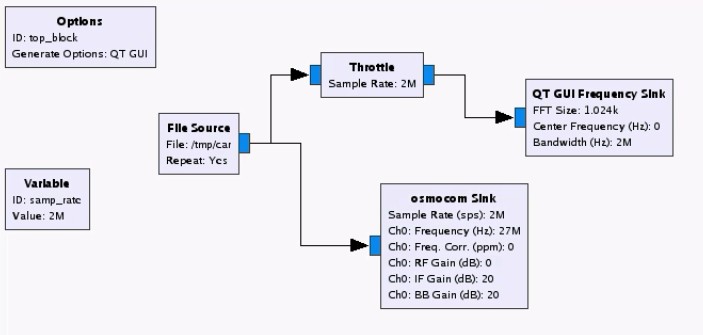
For the rules to be correct on the frequency and for it to perform in the same speed as real-time the flow graph’s sample rate should be set to the same sample rate used for the capture, which is 2 million samples per second.

Figure 3.7: Replaying The Saved Waveform

A QT GUI Frequency Sink is connected to the file source. Ideally a throttle block would be connected in between. A throttle block limits the data throughput

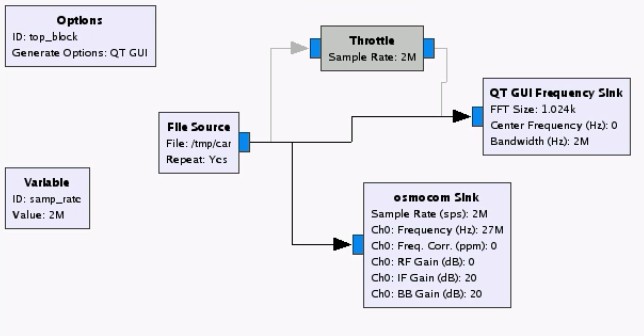
to the specified sampling rate. This prevents GNU Radio from consuming all CPU resources when the flow graph is not being regulated by external hardware. Since a osmocom sink, which refers to the HackRF One, is needed to replay this file, the throttle block can be disabled, as shown in Figure [3.8](#_bookmark62) If there is as a throttle in between, a two clock problem will exist.

Figure 3.8: Throttle Block Disabled

There is no possibility of the CPU crashing because the Frequency Sink isn’t going to go any faster than the osmocom sink, as they will both going to be pulling from the same file source, so the data will flow at the same rate to both of them.

If the flow graph is run like this, it will be the same as Figure [3.5](#_bookmark59) as expected but the file source keeps being repeated as it is the default setting for a file source.

On the osmocom sink, the frequency is set at 27 MHz which is the same frequency used to capture the signal. The RF gain is changed to 0 dB. The IF gain and BB gain are actually a little bit different when in transmit mode. The BB gain actually makes no difference whatsoever because there is no baseband amplifier on the transmit path of the HackRF One, only on the receive path. The IF gain can be left on the default value of 20 dB which is roughly in the middle point of the gain range.

At this point, if the flow graph is run it should replay the signal, transmit over the air to the R/C car and see the car actually move, although this does not

happen. The first suspicion is that the signal is not being transmitted at high enough power.

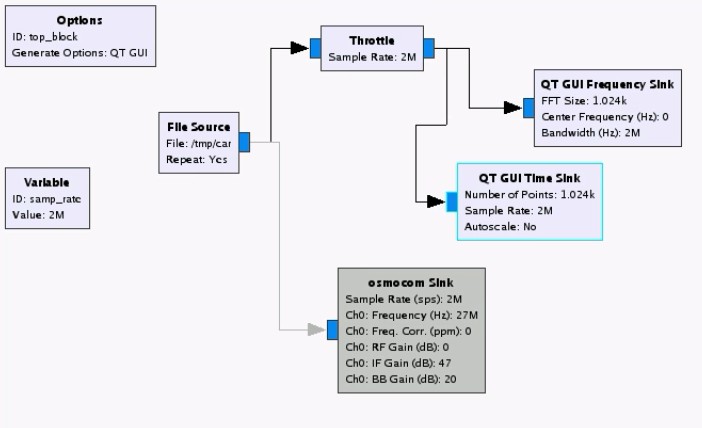
The IF gain is then changed to 47 dB which is the maximum value that be used, 27 dB higher than the previous value. The same result happens, the car is not moving.

Figure 3.9: QT GUI Time Sink

One of the benefits of having a saved waveform is that we have a consistent test vector. Everything should be the same every time the data is read out of this file. To test what might be wrong, a QT GUI Time Sink, which gives a time domain view, is connected. The throttle is enabled again and connects the file source to both the frequency and time sinks. The osmocom sink is turned off for now, as shown in Figure [3.9.](#_bookmark63)

Figure [3.10](#_bookmark64) shows that when zooming the time view and pausing when there are no bursts of activity, there is a visible offset from 0 to −0*.*1 and that is an explanation for the 0 Hz spike in the middle of the frequency domain view. There is some portion of the signal all the time that is a little bit off from 0 and that is perfectly normal in a receiver like the HackRF One to produce that type of a small defect.

When pausing during burst of activities, the visible signal is also fairly weak, not that much bigger than the background noise between bursts of activity. This

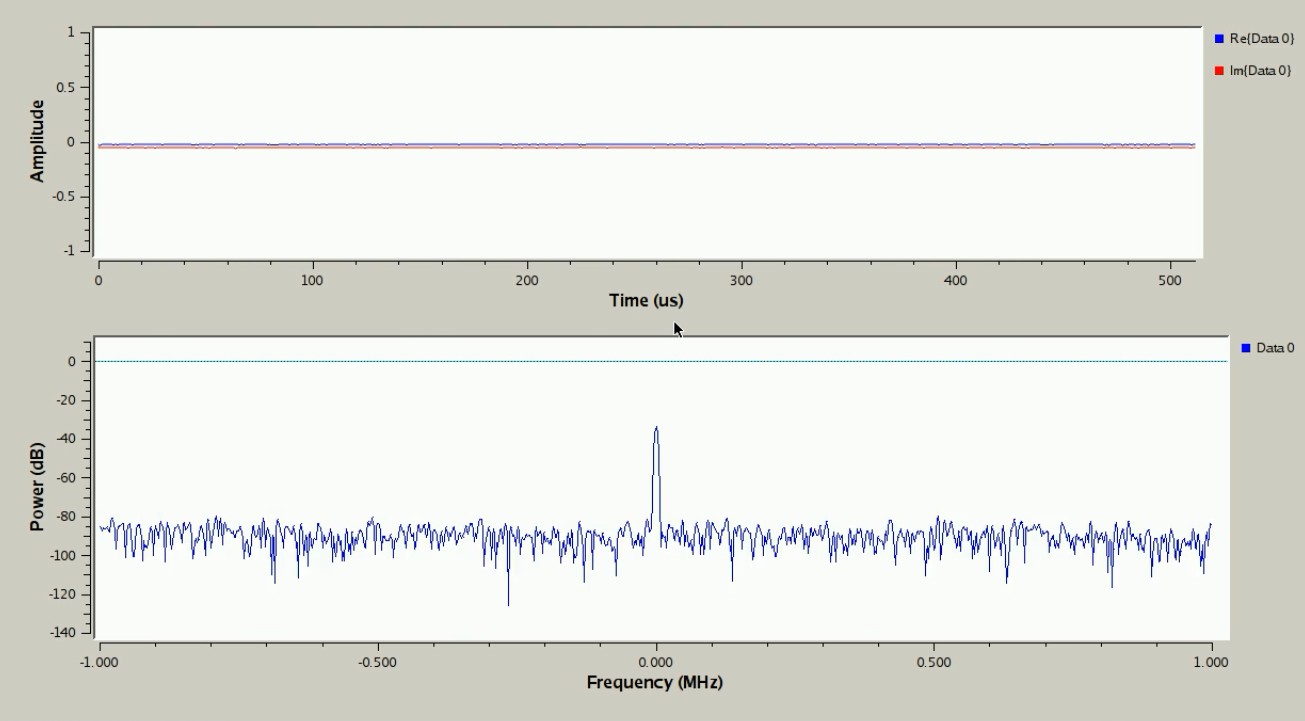


Figure 3.10: Background Noise

transmission is only ranging between about 0*.*1 and −0*.*15 of amplitude, as shown in Figure [3.11.](#_bookmark65) This is fairly small maximum amplitude.

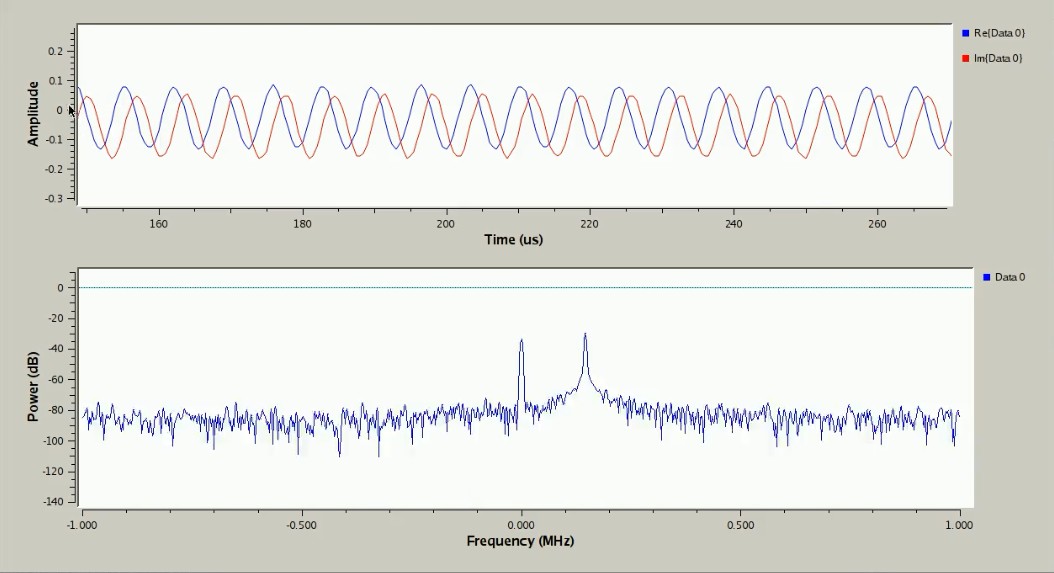


Figure 3.11: Burst Of Activity

The expected scale of the amplitude is between −1 and 1, because when the osmocom source receives information from the HackRF One, it takes 8-bit values and scales them and assigns them as floating point values that are between −1 and 1. So only a very small portion of the dynamic range is being used.

The solution could be to actually amplify the signal in the digital domain, before the waveform is given to the HackRF One. Samples can be scaled by multiplying them. A math operator named Multiply Const is connected between the file source and the throttle block, with the value of 6, as shown in Figure [3.12.](#_bookmark66)

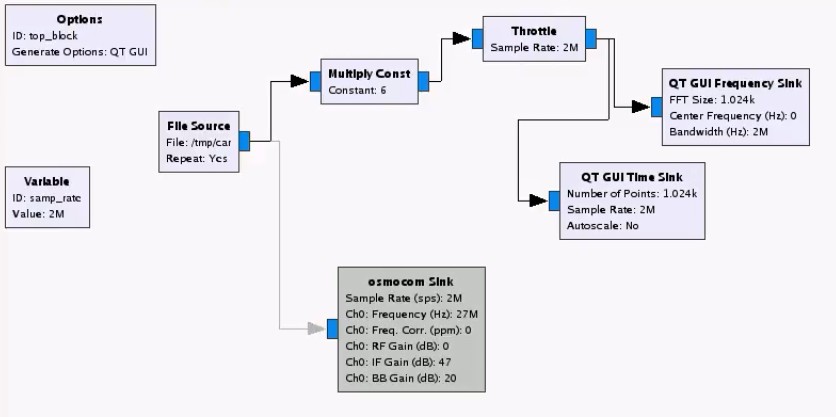


Figure 3.12: Multiply Const

The flow graph is run again, as shown in Figure [3.13](#_bookmark67) and now there is a difference in the transmission signal. Its amplitude extends from about 0*.*5 and

−1. This is a way to amplify a signal in the digital domain before converting it to

the analog domain.

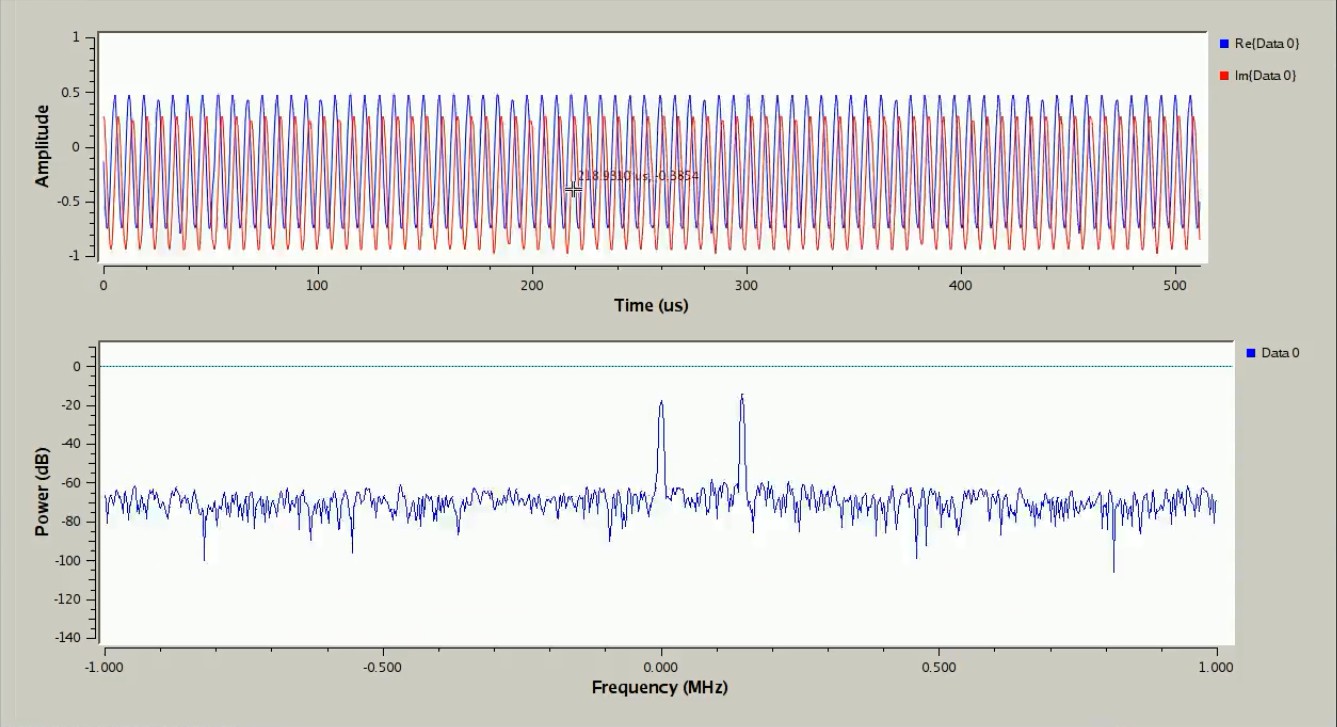


Figure 3.13: Amplified Signal

Finally, the osmocom sink is turned back on and the waveform file is trans- mitted with this new flow graph with the Multiply Const connecting also to the osmocom sink, as shown in Figure [3.14.](#_bookmark68)

Now the HackRF One mimics the remote control and R/C car moves the exact same way as recorded when creating the file, repeating the exact same set of movements while the flow graph is running.

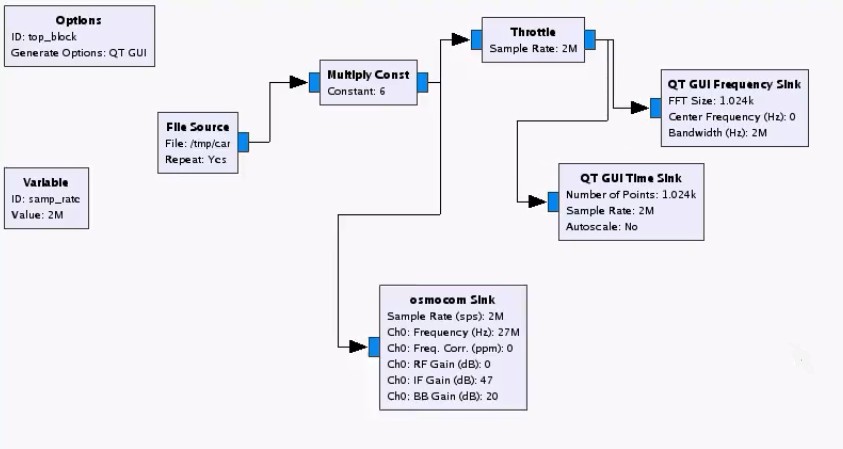


Figure 3.14: Transmitting The Amplified Signal

This case study concludes that by using a SDR equipment, in this case a HackRF One, it is possible to capture any radio signal transmitted by a device, like a remote control or a car key, and replay that signal causing the HackRF One to pose as the device itself and deceive the receiving equipment.

# Chapter 4

**Experiments And Results**

This chapter contains all the experiments made using the HackRF One together with SDR in GNU Radio and a mobile as test subject with the objective of disrupting the GPS signal. It is divided in two parts: the first part aimed to check what signal waveform affected the GPS signal the most. The second part aimed to do GPS spoofing, an attempt to deceive the mobile’s GPS by broadcasting incorrect GPS signals. Results were then, analyzed and reported.

### 4.1 GPS Jamming

In this experiment, the HackRF One has been used to transmit a signal created in GNU Radio Companion, in order to jam the GPS signal in a mobile. The waveforms tested are: cosine, square, triangle and saw-tooth. An additive white gaussian noise (AWGN) is also tested. For every waveform the initial amplitude value is 0*.*1 and starts being incremented 0*.*1, which also increases the power gain of the signal, until the GPS signal is jammed.

39

#### Flow Graph

The flow graph created to emulate a signal being transmitted through the HackRF One is displayed in Figure [4.1.](#_bookmark72)

The variable block defines the sample rate which is and should be the same in all the flow graph. It is set at 2*.*6 million samples per second. This block is created by default whenever a flow graph is created.

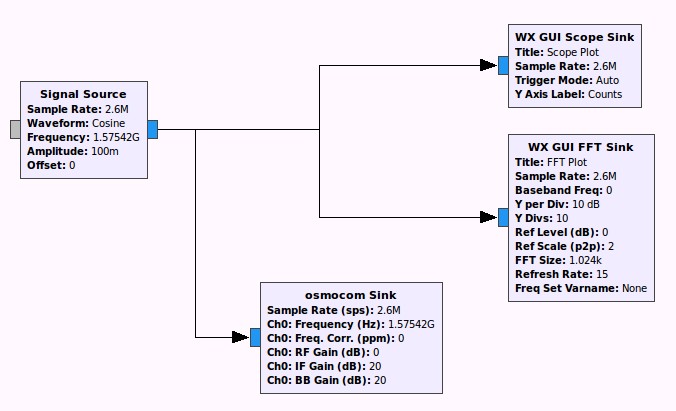
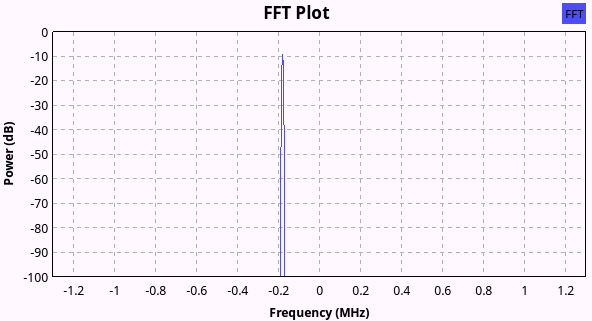
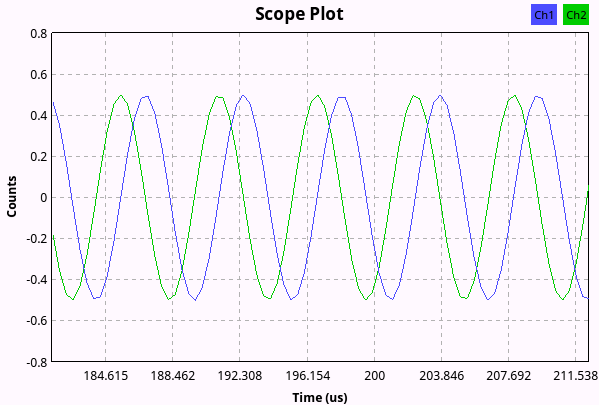
The signal source block acts like a waveform generator. The waveform parameter takes different forms as different waveforms are tested. The frequency is set at 1*.*57542 GHz, which is the L1 GPS frequency. The amplitude also takes different values during the tests. The offset is kept at the default value of 0.

Figure 4.1: Signal Transmission Flow Graph

The osmocom sink enables the HackRF to connect with the remaining flow graph. The channel frequency is set the same as the signal source frequency. The RF gain is set at 0 dB. Everything else is kept at its default values.

The WX GUI Scope Sink and WX FFT Sink are connected so it is possible to visualize what it is being transmitted. The WX GUI Scope Sink allows a time domain view of the signal and the WX FFT Sink allows a frequency domain view. Their parameters are kept all at its default values.

#### Cosine Waveform

By selecting the cosine waveform on the signal source block, the output is a cosine wave with peak amplitude configured by the amplitude parameter and average value set by the offset parameter.

* + - 1. Scope Plot (b) FFT Plot

Figure 4.2: Cosine Waveform

Figure [4.2a](#_bookmark75) shows the Scope Plot of the cosine waveform and Figure [4.2b](#_bookmark76) the FFT Plot. The FFT Plot shows a single spike of nearly −9 dB when using amplitude of 0*.*5.

#### Square Waveform

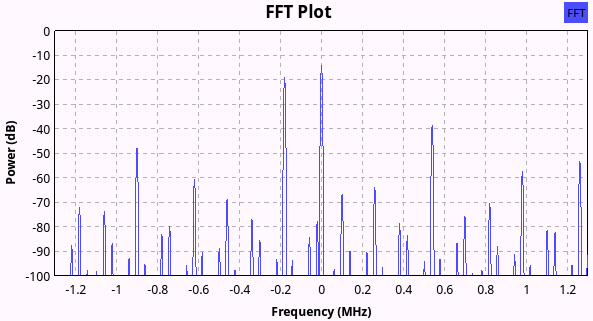
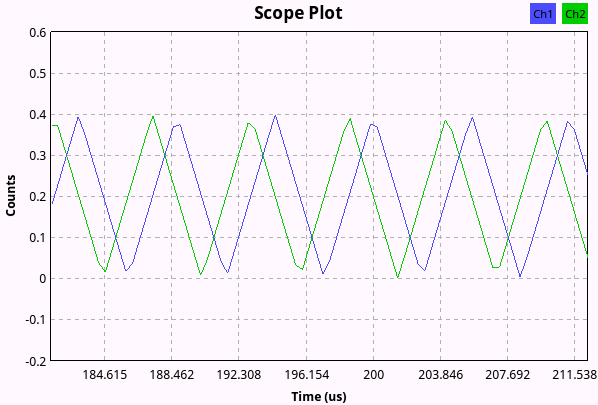
By selecting the square waveform on the signal source block, the output is a square wave with peak-to-peak amplitude configured by the amplitude parameter and the average value set by the offset plus amplitude/2. In the complex case, the imaginary signal is simply another square wave that has been shifted by ninety degrees.

Figure [4.3a](#_bookmark79) shows the Scope Plot of the square waveform and Figure [4.3b](#_bookmark80) the FFT Plot. There are multiple spikes but only two of them are significant with powers of nearly −16 dB with an amplitude of 0*.*3.

* + - 1. Scope Plot (b) FFT Plot

Figure 4.3: Square Waveform

#### Triangle Waveform

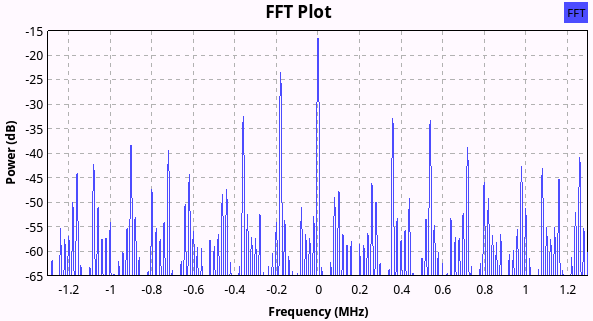
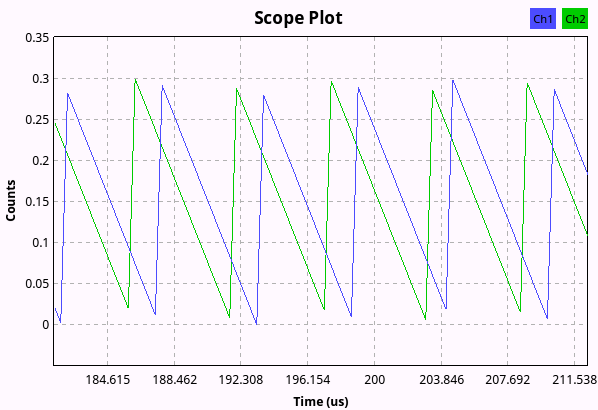
By selecting the triangle waveform on the signal source block, the output is a triangle wave with peak-to-peak amplitude configured by the amplitude parameter and the average value set by the offset plus amplitude/2. In the complex case, the imaginary signal is simply another triangle wave that has been shifted by ninety degrees.

* + - 1. Scope Plot (b) FFT Plot

Figure 4.4: Triangle Waveform

Figure [4.4a](#_bookmark83) shows the Scope Plot of the triangle waveform and Figure [4.4b](#_bookmark84) the FFT Plot. The higher spike has power of nearly −14 dB when using an amplitude of 0*.*4.

#### Saw Tooth Waveform

By selecting the saw tooth waveform on the signal source block, the output is a positive-going saw tooth wave with peak-to-peak amplitude configured by the amplitude parameter and the average value set by the offset plus amplitude/2. In the complex case, the imaginary signal is simply another saw tooth wave that has been shifted by ninety degrees.

* + - 1. Scope Plot (b) FFT Plot

Figure 4.5: Saw Tooth Waveform

Figure [4.5a](#_bookmark87) shows the Scope Plot of the saw tooth waveform and Figure [4.5b](#_bookmark88) the FFT Plot. One spike highlights with the power of −16 dB when using an amplitude of 0*.*3.

#### Additive White Gaussian Noise

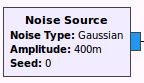
The signal source block is replaced by a noise source block with the noise type set at gaussian, as shown in Figure [4.6,](#_bookmark90) which presents a gaussian distribution.

Figure 4.6: Noise Source Block

The Scope Plot and the FFT Plot of the AWGN are shown in Figure [4.7a](#_bookmark92) and Figure [4.7b](#_bookmark93), respectively. The average power gain value is around −45 dB when using an amplitude of 0*.*3.

* + - 1. Scope Plot (b) FFT Plot

Figure 4.7: Gaussian Noise

#### Results

Table [4.1](#_bookmark95) shows the different waveforms with different amplitudes tested. The check mark indicates that the GPS signal was jammed and the cross that is was not. The maximum transmit power of the HackRF One is 15 dBm when the frequency ranges from 10 MHz to 2150 MHz. The total transmitted power adds up the gain set by the signal source or the AWGN.

The cosine waveform is only able to jam the mobile’s GPS when the power gain goes up to −9 dB. The square and saw tooth waveform are able to jam it with a power gain of −16 dB, while the triangle waveform is able to jam it with a power gain of −14 dB. The AWGN is able to jam the mobile’s GPS with a power gain of −45 dB and thus, is the one which needs less power gain to accomplish the intended.

Table 4.1: Different Waveforms And Amplitudes



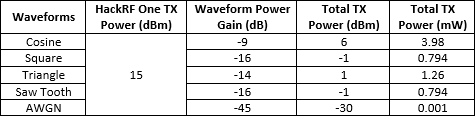
Table [4.1](#_bookmark96) shows the sum between the HackRF One transmit power and the power gain and also the conversion from dBm to watts of the total transmitted

power for every signal source and the AWGN. The power conversion from dBm to watts is given by the equation [4.1.](#_bookmark96)

*P* (*W* ) = 1*W* × 10(*P*(*dBm*)*/*10)*/*1000 (4.1)

The cosine waveform needs the highest power to be able to jam the GPS signal in the mobile while the AWGN needs the lowest.

Table 4.2: Total Transmitted Power



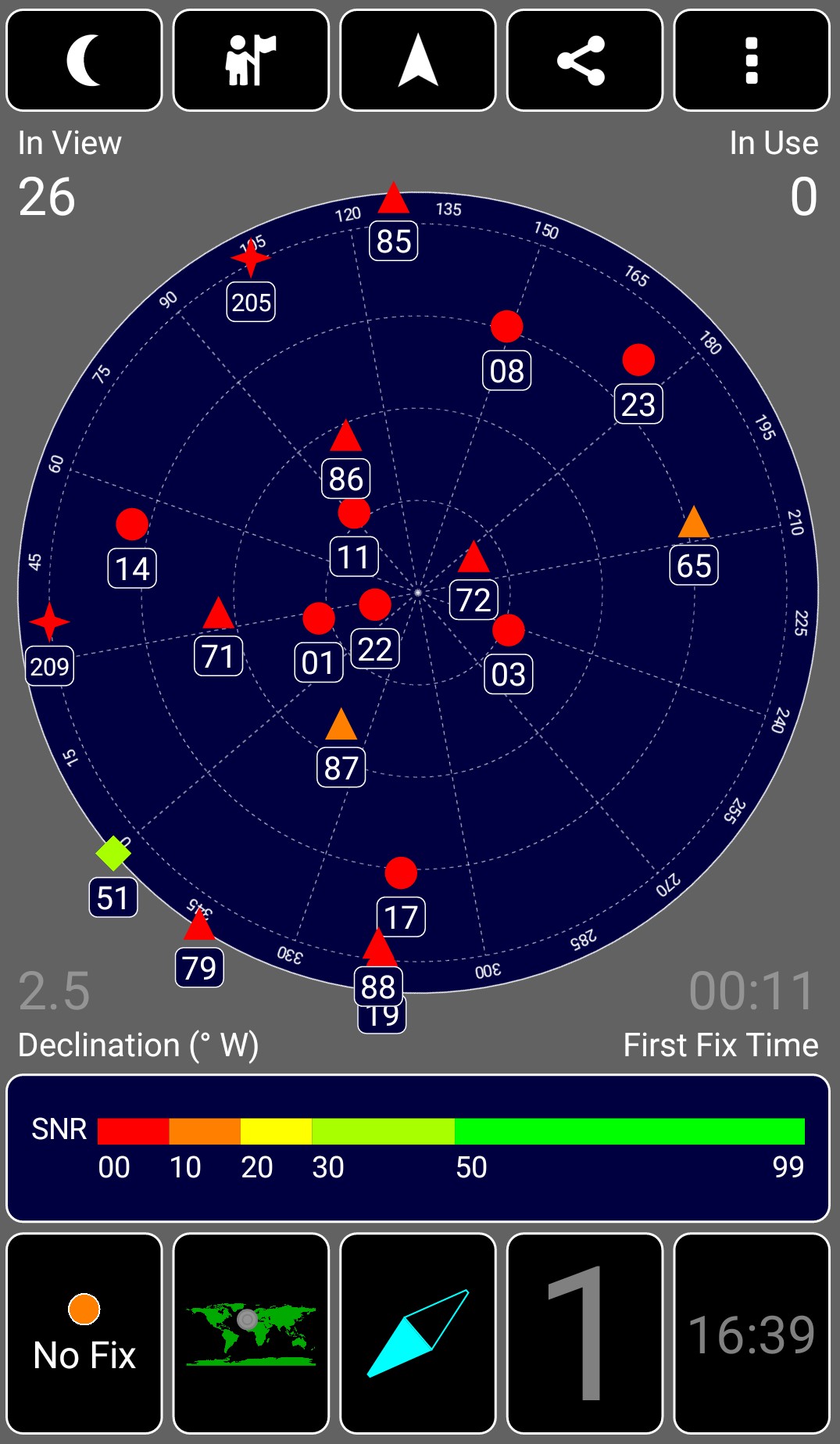
The app used to test the GPS location in the mobile is called GPS Test and it is available on Google Play [[33].](#_bookmark159) Figure [4.8](#_bookmark98) shows when the app isn’t able to get a 3D fix location and thus, isn’t able to report the mobile’s location.

Figure 4.8: Loss Of Signal

### GPS Spoofing

In this experiment, the HackRF One has been used to deceive a mobile’s GPS location by transmitting a simulated GPS signal with false coordinates. This process is called GPS Spoofing.

The software used to generate the GPS signal is GPS-SDR-SIM, a GPS L1 C/A signal generator [[34].](#_bookmark160) This software generates a file with In-phase/Quadrature (I/Q) samples of the L1 C/A signal complex envelope that is ready to be trans- mitted via any SDR offering quadrature modulation in L1 band, like the HackRF One. The file generation runs offline so it is not time critical.

#### Generating The GPS Signal

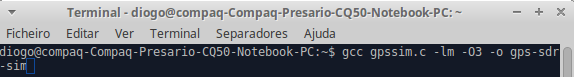
In order to use the software to generate the GPS signal, first it needs to be compiled into the Linux operating system. The program can be compiled with one simple command shown in Figure [4.9.](#_bookmark101)

Figure 4.9: Compiling GPS-SDR-SIM

After the compilation, the GPS signal file can be generated using the command shown in Figure [4.10.](#_bookmark102) The options used are the following:

* + - * -b 8, as HackRF uses 8-bit I/Q sampling
      * -e, followed by a brdc file
      * -l, followed by coordinates of the desired location

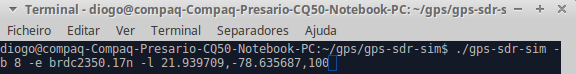
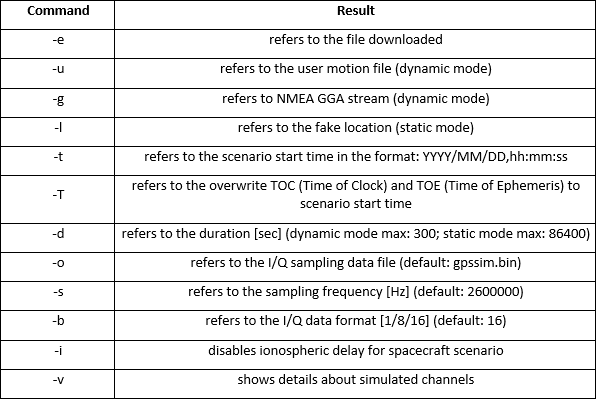


Figure 4.10: Generating GPS Signal

Table [4.3](#_bookmark103) describes all the options available when generating a file using the GPS-SDR-SIM.

Table 4.3: GPS-SDR-SIM Options



The brdc file is a daily GPS broadcast ephemeris file which can be accessed in the ephemerides archive maintained by NASA [35]. These files contain the coordinates of the satellites, in the worst scenario, with a 24 hour lag. They are then used to generate the simulated pseudorange and Doppler for the GPS satellites in view. This simulated range data is then used to generate the digitized I/Q samples for the GPS signal.

The GPS signal file takes 300 seconds, which is the default duration, to be generated, as shown in Figure [4.11.](#_bookmark104) A file, named "gpssim.bin", is created which is the simulated GPS signal file.

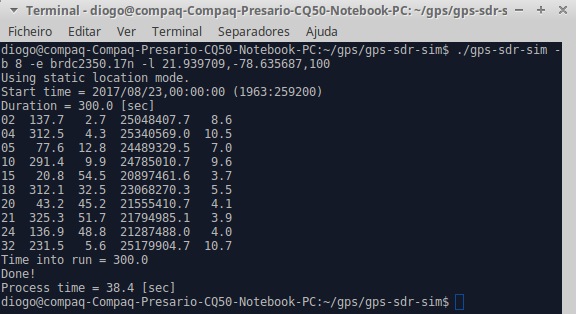


Figure 4.11: GPS Signal Generated

#### Transmitting The GPS Signal

After generating the file, the signal can now be loaded into the HackRF One for playback using the command shown in Figure [4.12.](#_bookmark106) The hackrf\_transfer tool allows the HackRF One to transfer or receive a signal straight from the Linux’s terminal command, without the need of a SDR platform, like the GNU Radio. The options used are the following:

* + - * -t, followed by the "gpssim.bin" file
      * -f, followed by the GPS L1 frequency band: 1*.*57542 GHz
      * -s, followed by the set sample rate of 2*.*6 million
      * -a 1, which enables the amplitude
      * -x 0, which sets the transmission variable gain amplitude gain to 0 dB

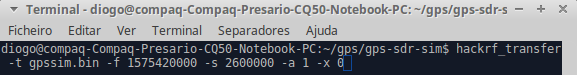
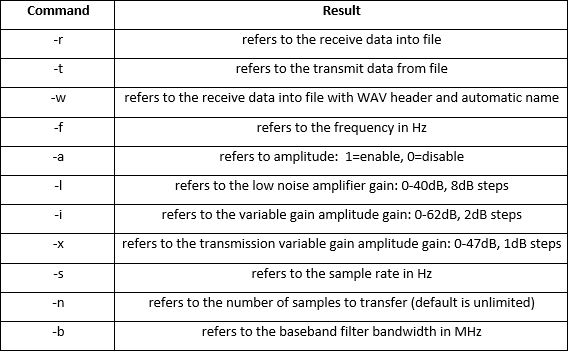


Figure 4.12: Transmission Command

The full commands available for the hackrf\_transfer tool are described in Table

[4.4.](#_bookmark107) Figure [4.13](#_bookmark108) shows the HackRF transmitting the simulated GPS signal.

Table 4.4: Hackrf\_Transfer Options



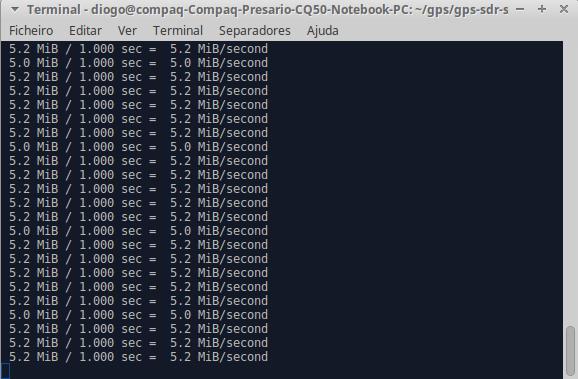


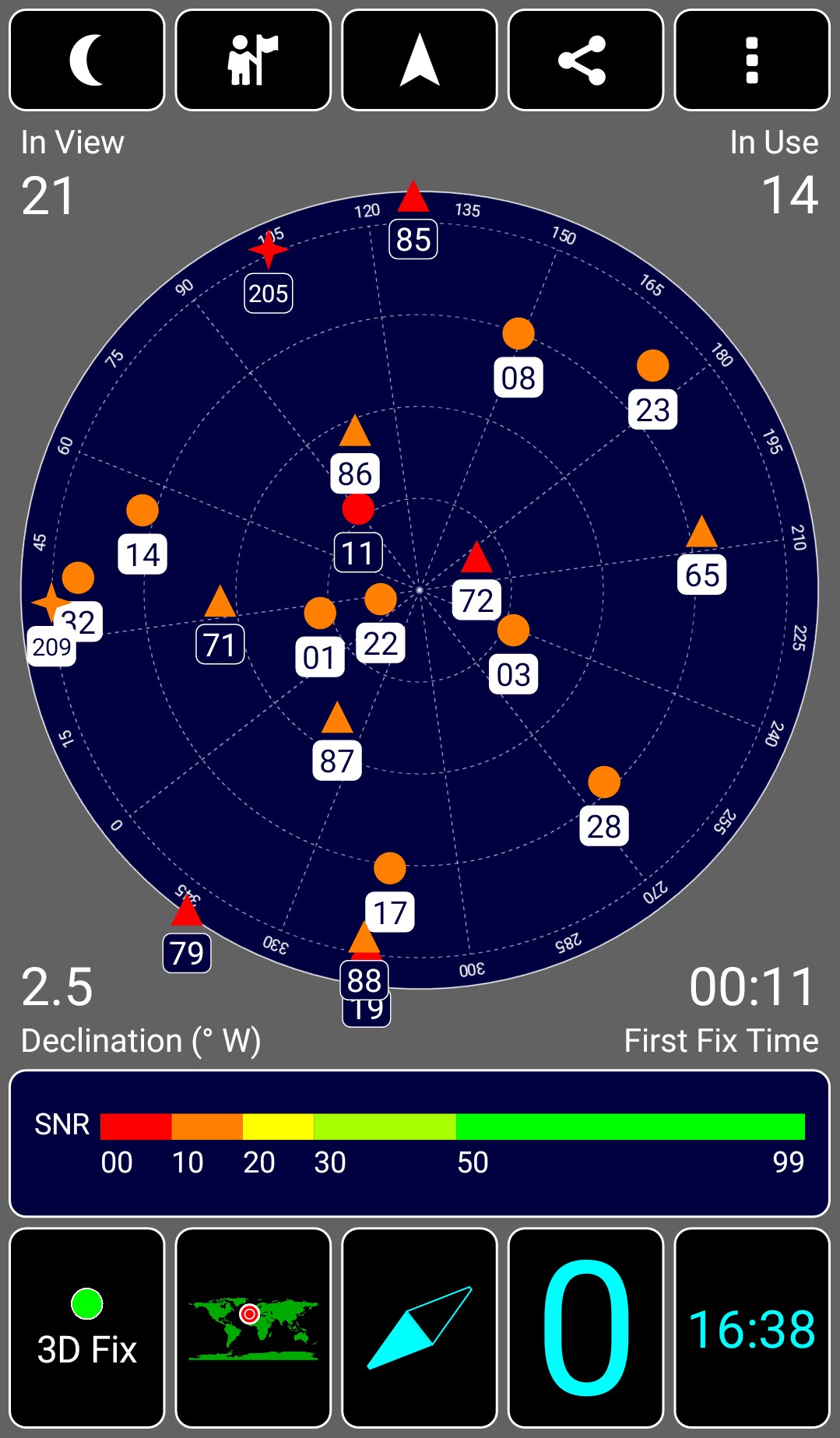
Figure 4.13: Transmitting GPS Signal

#### Results

The GPS Test app is opened on the mobile. Figure [4.14a](#_bookmark111) shows the initial position. The mobile is able to get a 3D fix and the position corresponds to Portugal as seen in Figure [4.14b.](#_bookmark112)

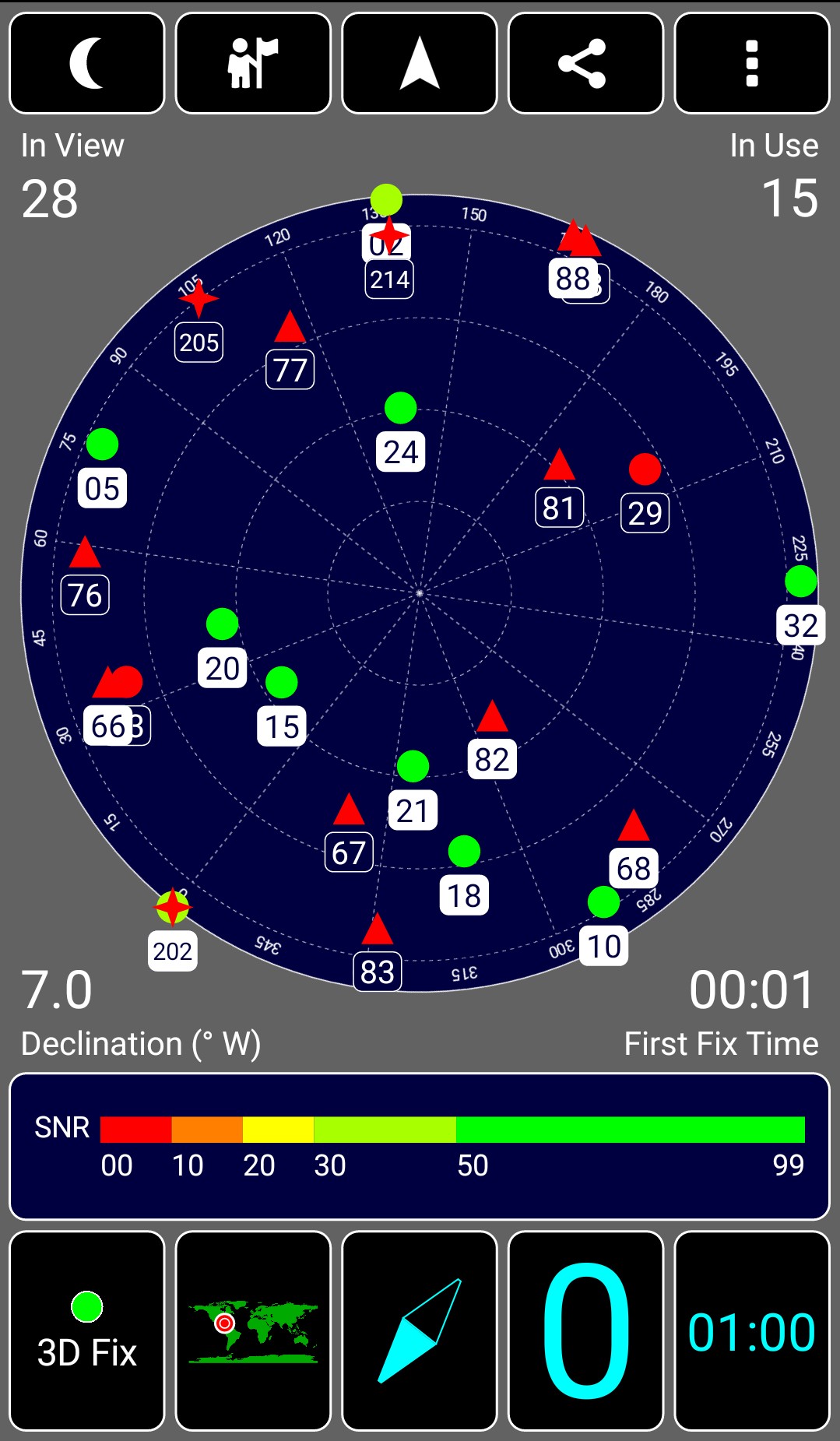
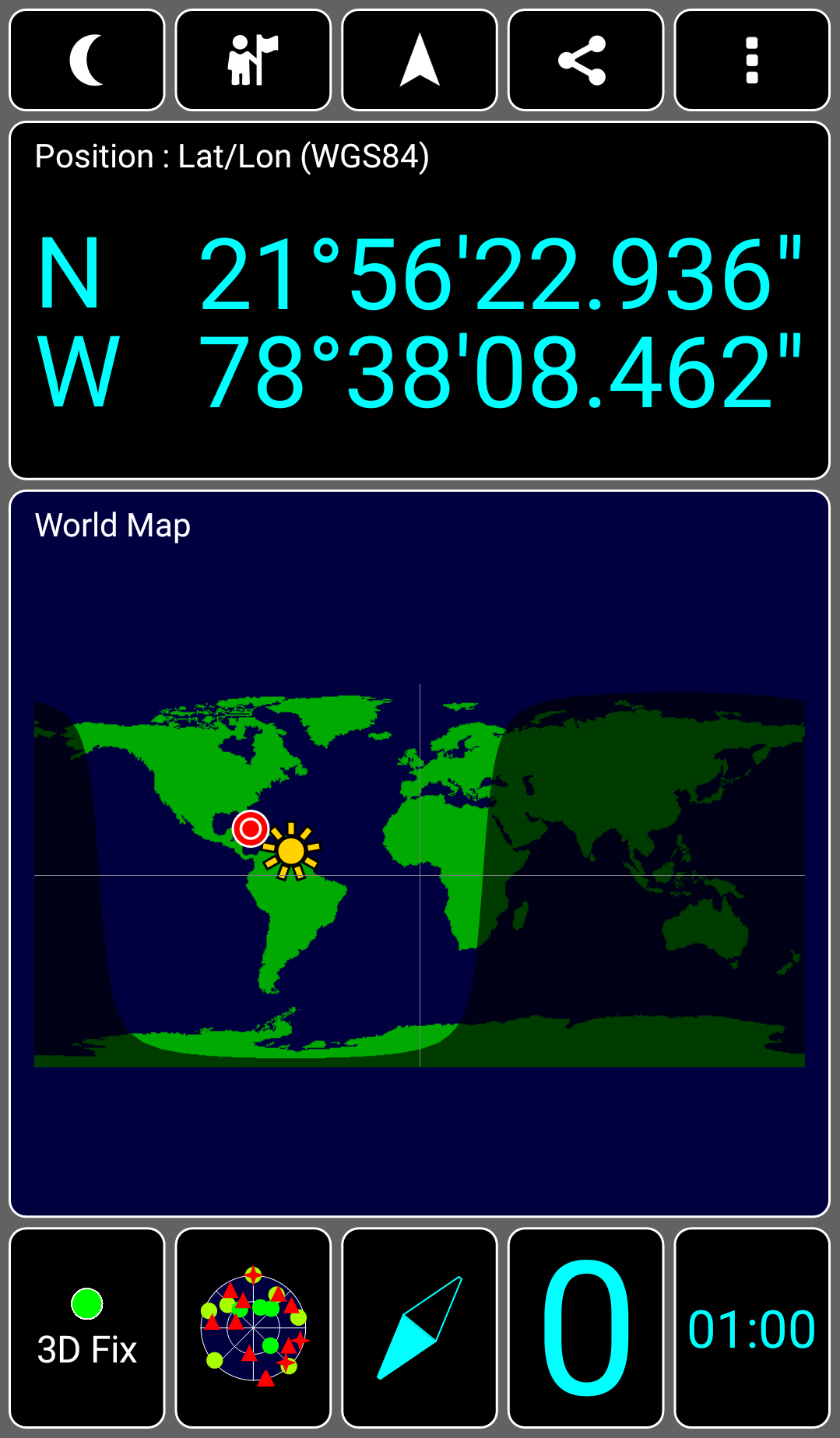
When the HackRF starts transmitting, the mobile loses the signal for a moment and then locks to a 3D fix but now the position corresponds to Cuba, the selected

set of coordinates, as shown in Figure [4.15a](#_bookmark114) and [4.15b](#_bookmark115). It is also visible that the timezone changed.

1. 3D Fix Ini- tial Position
2. Map Final Position

Figure 4.14: Initial Position

* 1. 3D Fix Fi- nal Position
  2. Map Final Position

Figure 4.15: Final Position

# Chapter 5

**Conclusions And Future Work**

### Conclusions

This research has revealed that it is possible to use the HackRF One or any SDR equipment capable of transmitting and receiving signals to capture and replay a radio signal. This technology allows malicious users to exploit the many RF applications. Such attacks like unlocking a car or disarming an alarm system become easy to accomplish as the SDR equipment can copy the same radio signal that is being transmitted through the air to the receiving equipment. Countermeasures need to be implemented to abolish these threats.

Software Defined Radio has come to break the barrier between the so called hackers and normal users with malicious intentions. What used to be a question of complex radio understanding and hardware skills, is now a question of a not so difficult research process, an understanding of Digital Signal Processing and software frameworks for SDR.

This research has further revealed that GPS is even a more vulnerable signal when SDR is involved, successfully answering all the questions raised in Chapter [1.](#_bookmark6)

51

Chapter 5. *Conclusions And Future Work*

GPS jamming has always been a more or less easy attack to accomplish and now with the use of SDR it is proven that it is equally easy to accomplish and moreover it is cheaper. The presented GPS jamming experiment shows that the white gaussian noise is the best waveform to jam the GPS signal as it requires the less power and amplitude to achieve it. The specifications of the antenna harmed the effectiveness of the attack as the range between the HackRF One and the mobile needed to be short.

GPS spoofing without SDR is difficult as there are many challenges in generating a GPS stream that will pass a signal to a GPS receiver. It is also expensive as it requires high precision equipment. The presented GPS spoofing experiment has proven that it can be easily accomplished using fairly inexpensive SDR setup and freely available software. The setup of one HackRF One and the GPS-SDR- SIM software allows the achievement of GPS signal spoofing without the high-end hardware.

Both these attacks have not faced resistance from the mobile and so, it is imperative that detection and mitigation systems are developed and implemented in order to prevent or at least difficult these attacks.

### Future Work

Since this study has only focused on the GPS technology on a mobile, further research should focus on using Software Defined Radio to expose the vulnerabilities of other technologies and equipment. On the other hand, research using SDR to develop detection and mitigation systems is starting to appear.

If a more capable antenna or external amplifiers had been used in these experiments, the distance the attacking equipment can disrupt the receiver would have increased, increasing the effectiveness of the GPS jamming and spoofing.

The software used in the GPS spoofing has only allowed to re-transmit satellite positions from the ephemeris archive. Achieving a real-time simulation would make

Chapter 5. *Conclusions And Future Work*

it possible to gain navigational control of the affected devices, such as drones or self-driving vehicles. If more research is put into it there will be a public safety concern as mid-air collisions with other aerial vehicles or building could happen as well driving accidents.

Also by acknowledging that nowadays everything is becoming interconnected through the internet in a process called Internet of Things, these security threats are the main topic to take into account. Everything from houses to phones becomes a target for a possible malicious attack. It is imperative that further research is conducted in order to understand how to detect and fight these threats.

53

# Appendices

**Appendix A**

**GNSS Acquisition In The Absence Of Interference**

In the absence of interference the input signal received at the GNSS receiver, can be represented by [[4]:](#_bookmark130)

*Ns*

Σ

*yRF* (*t*) = *rRF,i*(*t*) + *ηRF* (*t*) (A.1)

*i*=1

where the term *rRF,i*(*t*) is the sum of *Ns* useful signals, from *Ns* different satellites, and *ηRF* (*t*) is a noise term.

*rRF,i*(*t*) assumes the following structure:

*rRF,i*(*t*) = *Aici*(*t* − *τi*)*di*(*t* − *τi*) cos[2*π*(*fRF* + *fd,i*)*t* + *ϕRF,i*] (A.2)

where

* + - *Ai* is the amplitude of the *ith*;
    - *τi* is the code phase delay introduced by the transmission channel;
* *ci*(*t* − *τi*) is the spreading sequence given by the product of several terms and assumes values between -1,1;
* *di*(*t* − *τi*) is the navigation message, BPSK modulated, containing satellite data, each binary unit is called bit;
* *fd,i* is the Doppler frequency shift affecting the *ith* and *ϕRF,i* is the initial carrier phase offset;
* *fRF* is the carrier frequency and depends on the GNSS signal band under analysis. For GPS L1 band it is 1575*.*42 MHz.

*ci*(*t*) can be expressed as:

*ci*(*t*) = *c*1*,i*(*t*)*c*2*,i*(*t*)*sb,i*(*t*) (A.3)

where

* *c*1*,i*(*t*) is the periodic repetition of the primary spreading code
* *c*2*,i*(*t*) is the secondary code
* *sb,i*(*t*) is the subcarrier signal which is the periodic repetition of a basic wave that determines the spectral characteristics of *rRF,i*(*t*). The GPS C/A code adopts the BPSK signal as shown in [A.1.](#_bookmark121)

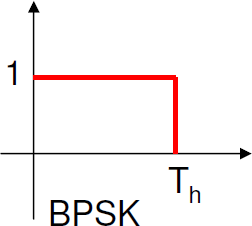


Figure A.1: BPSK

*ηRF* (*t*) is assumed to be AWGN with power spectral density (*P SD*) = *N*0*/*2.

Each useful signal is characterized by power:

Appendix A. *GNSS Acquisition In The Absence Of Interference*

*A*2

*i*

*Ci* =

2

(A.4)

and the overall signal quality is quantified by the carrier-to-noise-power-density ratio (*Ci/N*0).

The input signal (Equation [A.1)](#_bookmark120) is received by the receiver antenna, down- converted and filtered by the receiver front-end. So, before the Analog to Digital (A/D) conversion, the received signal is given by:

*Ns Ns*

*y*(*t*) = Σ *ri*(*t*)+*η*(*t*) = Σ *Aic*˜*i*(*t*−*τi*)*di*(*t*−*τi*) cos[2*π*(*fRF* +*fd,i*)*t*+*ϕi*]+*η*(*t*) (A.5)

*i*=1

*i*=1

where *fIF* is the receiver intermediate frequency (IF). The impact of the front- end is neglected and so, *c*˜*i*(*t* − *τi*), which represents the spreading sequence after the front-end filtering assumes the following condition:

*c*˜*i*(*t*) ≈ *ci*(*t*) (A.6)

*η*(*t*) is the down-converted and filtered noise component.

When a generic sequence *x*[*n*] is processed in any digital platform, the notation *x*[*n*] = *x*[*nTs*] is adopted, after sampling *y*(*t*) at the sampling frequency *fs* = 1*/Ts*. In a digital receiver the IF signal is sampled through an ADC, which generates a sampled sequence *y*(*nTs*). After Equation [A.5](#_bookmark122) is sampled and digitized, when neglecting the quantization impact, the following signal model is obtained:

*Ns*

Σ

*y*[*n*] = *Aic*˜*i*[*n* − *τi/Ts*]*di*[*n* − *τi/Ts*] cos(2*πFD,in* + *ϕi*) + *η*[*n*] (A.7)

*i*=1

where *FD,i* = (*fIF* + *fd,i*)*Ts*

59

The spectral characteristics of the discrete-time random process *η*[*n*] depends on the filtering type same as the sampling and the decimation strategy adopted in the front-end. When the IF signal and noise are sampled at Nyquist rate, as the sampling frequency becomes 2*BIF* , where *BIF* is the front-end bandwidth, the noise variance becomes easy to know:

*σ*2 = *E*{*η*2[*t*]} = *E*{*η*2[*nT* ]} = *N*0 *f*

*IF*

*s*

2

*s*

= *N B*

(A.8)

The auto-correlation function is another important parameter for the noise characterization which states that *η*[*n*] is a classical independent and identically distributed (i.i.d) wide sense stationary (WSS) random process, or a white sequence. *δ*[*m*] is the Kronecker delta function.

0

*IF*

*RIF* [*m*] = *E*{*η*[*n*]*η*[*n* + *m*]} = *σ*2

*IF*

*δ*[*m*] (A.9)

The different GNSS signals are analyzed separately by the receiver, as the spreading code sequence is orthogonal. The index *i* in Equation [A.7](#_bookmark123) is dropped as only a single satellite is considered and so, the resulting signal becomes:

*y*[*n*] = *r*[*n*] + *η*[*n*] = *Ac*[*n* − *τ /Ts*]*d*[*n* − *τ /Ts*] cos(2*πFDn* + *ϕ*) + *η*[*n*] (A.10)

# Appendix B

**GNSS Acquisition In The Presence Of Interference**

The problem of characterizing the jamming signal has been addressed in several papers [[9],](#_bookmark135) [[1](#_bookmark138)2], [[13],](#_bookmark139) [[25],](#_bookmark151) [[36],](#_bookmark162) [[37].](#_bookmark163) Most jammers used in a civil context broadcast frequency modulated signals with an almost periodic behavior. There can be deviations due to drifts in the local oscillators used for the signal generation. The signal center frequency varies according to a periodic pattern.

In the presence of interference, the disturbing signal is accounted in an additional term which is added to Equation 10, and so this equation becomes [[4]:](#_bookmark130)

*y*[*n*] = *r*[*n*] + *η*[*n*] + *i*[*n*] (B.1)

where *i*[*n*] models the disturbing signal. Depending on the time/frequency and statistical characteristics of the disturbing signal *i*[*n*] can assume different expressions.

##### Saw-tooth interference

When the periodic pattern corresponds to a saw-tooth function, before the A/D conversion *i*(*t*) can be modeled as:

61

Appendix B. *GNSS Acquisition In The Presence Of Interference*

*i*(*t*) = √2 cos(2*π*(*fRF* + *fi*(*t*))*t* + *ϕi* (B.2)

where

* *fi*(*t*) is the instantaneous frequency of the disturbing signal
* *fRF* denotes the RF
* *ϕi* models the signal phase

The amplitude variations of the disturbing signal are usually less than 0.5 dB and so, are neglected in Equation [B.2.](#_bookmark125) *fi*(*t*) defines a practically periodic frequency pattern which is characterized by a sweep range, the frequency interval affected by the jammer signal, and a sweep period which is the time required to span the sweep range. Sweep periods are typically around 10 *µ*s and sweep ranges are usually in the 10-40 MHz interval [[12](#_bookmark138)], [[37].](#_bookmark163)

The shorter the sweep period, the more difficult it is to mitigate the impact of the jammer. For example, a notch filter will have more difficulties to estimate *fi*(*t*) and remove the disturbing signal, as fast frequency varying signals are more difficult to track [[27].](#_bookmark153)

##### Gaussian narrow-band interference

This interference type includes all the signals that can be effectively characterized by a Gaussian probability density function (pdf). The spectrum of these signals only occupies a portion of the GNSS signal band. The disturbing signal can have a relatively wide band that is, however, narrow with respect to the GNSS signal under consideration. *i*[*n*] can be modeled as a colored Gaussian process when the interference is assumed to be zero mean and WSS:

*i*[*n*] ∼ *N* (0*, σ*2

*IN T*

) (B.3)

Appendix B. *GNSS Acquisition In The Presence Of Interference*

characterized by a PSD *Gl*(*f* ) and an auto-correlation *Rl*[*n*] which allow to characterize the time/frequency characteristics of *i*[*n*]. This characterization allows to describe a wide variety of interference.

##### Continuous Wave Interference

Continuous Wave Interference (CWI) include all narrow-band signals that can be reasonably represented as pure sinusoids with respect to the GNSS bands. They can be generated by Ultra High Frequency (UHF) and VHF TV, VHF Omni- directional Radio-range (VOR) and Instrument Landing System (ILS) harmonics, by spurious signals caused by power amplifiers working in non-linearity regions or by oscillators present in many electronic devices [[38].](#_bookmark164)

When a real CWI is present, *i*[*n*] assumes the expression:

*i*[*n*] = *AINT* cos(2*πfinTs* + *θint*) = *AINT* cos(2*πFintn* + *θint*) (B.4) where *AINT* is the interference amplitude and *fi* is the interference frequency.

*θint* is a uniformly distributed random variable of the form :

*θint* ∼ U (−*π, π*) (B.5)

The Jammer to Noise ratio defines the power of the interference with respect to the noise variance as follows:

*J A*2 1 *A*2

= *IN T* = *IN T*

(B.6)

2 2*N*0*BIF*

*σ*

*N* 2

*IF*

63

# Bibliography

[1]M. Thomas, J. Norton, A. Jones, A. Hopper, N. Ward, P. Cannon, N. Ack- royd, P. Cruddace, and M. Unwin, *Global Navigation Space Systems : reliance and vulnerabilities*, 2011.

[2]S. Yarwood, “Jamming & radio interference: understanding the impact,” vol.

IET Sector Insights, pp. 1–6, 2012.

[3]B. Hofmann-Wellenhof, H. Lichtenegger, and E. Wasle, *GNSS - global nav- igation satellite systems: GPS, Glonass and more*, 1st ed. Springer-Verlag Wien, 2008.

[4]D. Borio, “A Statistical Theory for GNSS Signal Acquisition,” *Dipartimento di Elettronica*, vol. Doctoral T, p. 293, 2008.

[5]R. Bingley, *Handouts Satellite Based Positioning (H24VST)*. University of Nottingham, 2013.

[6]E. Kaplan and C. Hegarty, *Understanding GPS. Principles and applications.*

Norwood, Massachusetts: Artech House, 2006.

[7]P. Misra and P. Enge, *Global Positioning System: Signals, Measurements and Performance*. Lincoln, Massachusetts: Ganga-Jamuna Press, 2010.

[8]D. Last, *GNSS: The Present Imperfect*. InsideGNSS, 2010.

[9]S. Pullen and G. X. Gao, *GNSS Jamming in the Name of Privacy*. InsideG- NSS, 2012.

65

[10]J. Volpe, *Vulnerability assessment of the transportation infrastructure relying on the global positioning system*. National Transportation Systems Center, 2001.

[11]P. Craven, R. Wong, N. Fedora, and P. Crampton, *Studying the Effects of Interference on GNSS Signals*. San Diego, California: International Technical Meeting of The Institute of Navigation, 2013.

[12]R. Mitch, R. Dougherty, M. Psiaki, S. Powell, B. O’Hanlon, J. Bhatti, and

T. Humphreys, *Signal characteristics of Civil GPS Jammers*. ION GNSS 2011, 2011.

[13]T. Kraus, R. Bauernfeind, and B. Eisfeller, *Survey of In-Car Jammers – Analysis and Modelling of the RF Signals and IF Samples*. ION GNSS 2010, 2011.

[14]D. Borio, F. Dovis, H. Kuusniemi, and L. Lo Presti, “Impact and Detec- tion of GNSS Jammers on Consumer Grade Satellite Navigation Receivers,” *Proceedings of the IEEE*, vol. 104, pp. 1233–1245, 2016.

[15]E. Axell, F. Eklöf, M. Alexandersson, and P. Johansson, *Jamming detection in GNSS receivers: Performance evaluation of field trials*. Nashville, Tennessee: ION GNSS 2013, 2013.

[16]T. Humphreys, B. Ledvina, M. Piaski, B. O’Hanlon, and P. Kintner, *Assessing the Spoofing Threat: Development of a Portable GPS Civilian Spoofer*. ION GNSS 2008, 2008.

[17]M. Psiaki, S. Powell, and B. O’Hanlon, *GNSS Spoofing Detection Using High- Frequency Antenna Motion and Carrier-Phase Data*. ION GNSS 2013, 2013.

[18]K. von Hunerbein and W. Lange, *A New Solution of Generation of Spoofing Signals for GNSS Receivers*. In Proceedings of International Symposium on Certification of Gnss Systems and Services (CERGAL), 2014.

*References*

[19]D. Borio, “GNSS acquisition in the presence of continuous wave interference,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 1, pp. 47–60, 2010.

[20]M. Wildemeersch, E. C. Pons, A. Rabbachin, and J. F. Guasch, *Impact Study of Unintentional Interference on GNSS Receivers*, 2010.

[21]F. Dovis, *GNSS Interference Threats and Countermeasures*. Norwood, Mas- sachusetts: Artech House, 2015.

[22]P. Ward, *GPS Receiver RF Interference Monitoring, Mitigation, and Analysis Techniques*. Navigation-Journal of the Institute of Navigation, 1994.

[23]A. Niekerk and L. Combrinck, *The use of civilian type GPS receivers by the military and their vulnerability to jamming*. South African Journal of Science, 2012.

[24]M. Jones, *The Civilian Battlefield. Protecting GNSS Receivers from Interfer- ence and Jamming*. InsideGNSS, 2011.

[25]H. Kuusniemi, E. Airos, M. Bhuiyan, and T. Kröger, *GNSS Jammers: how vulnerable are Consumer grade Satellite Navigation Receivers?* European Journal of Navigation, 2012.

[26]R. Bauernfeind, T. Kraus, D. Dötterböck, and B. Eisfeller, *Car Jammers: Interference Analysis*. GPS World, 2011.

[27]D. Borio, C. O’Driscoll, and J. Fortuny, *GNSS jammers: Effects and counter- measures*. Proc. 6th ESA Workshop Satellite Navig. Technol./Eur. Workshop GNSS Signals Signal Process, 2012.

[28]J. Warner and R. Johnston, “A Simple Demonstration that the Global Po- sitioning System (GPS) is Vulnerable to Spoofing,” *The Journal of Security Administration*, pp. 19–28, 2002.

[29]D. Shepard, J. Bhatti, T. Humphreys, and A. Fansler, “Evaluation of Smart Grid and Civilian UAV Vulnerability to GPS Spoofing Attacks,” *ION GNSS*, pp. 3591–3605, 2012.

67

[30]“HackRF One,” (Accessed: April 2017). [Online]. Available: h[ttps:](https://greatscottgadgets.com/hackrf/)

[//greatscottgadgets.com/hackrf/](https://greatscottgadgets.com/hackrf/)

[31]“GNU Radio, the free and open software radio ecosystem,” (Accessed: April 2017). [Online]. Available: <http://gnuradio.org/>

[32]“Case Study - Replay,” (Accessed: June 2017). [Online]. Available: <https://greatscottgadgets.com/sdr/11/>

[33]“GPS Test App,” (Accessed: August 2017). [Online]. Available: h[ttps:](https://play.google.com/store/apps/details?id=com.chartcross.gpstest)

[//play.google.com/store/apps/details?id=com.chartcross.gpstest](https://play.google.com/store/apps/details?id=com.chartcross.gpstest)

[34]“GPS-SDR-SIM,” (Accessed: August 2017). [Online]. Available: h[ttps:](https://github.com/osqzss/gps-sdr-sim)

[//github.com/osqzss/gps-sdr-sim](https://github.com/osqzss/gps-sdr-sim)

[35]“BRDC File Archive,” (Accessed: August 2017). [Online]. Available: <ftp://cddis.gsfc.nasa.gov/gnss/data/daily>

[36]R. H. Mitch, M. L. Psiaki, B. O’Hanlon, S. P. Powell, and J. Bhatti, *Civilian GPS jammer signal tracking and geolocation*. Nashville, Tennessee: Proc. 25th Int. Tech. Meeting Satellite Div. Inst. Navig., 2012.

[37]D. Borio, J. Fortuny, and C. O’Driscoll, *Spectral and spatial characterization of GNSS jammers*. Baska, Croatia: Proc. 7th GNSS Vulnerabilities Solutions Conf., 2013.

[38]R. J. Landry and A. Renard, *Analysis of potential interference sources and as- sessment of present solutions for GPS/GNSS receivers*. 4th Saint-Petersburg on INS, 1997.