University Institute of Lisbon

Department of Information Science and Technology

Wireless UAV Restraining System

Diogo Dias dos Santos

A Dissertation presented in partial fulfillment of the Requirements
for the Degree of
Master in Telecommunications and Computer Engineering

Supervisor
Doctor Nuno Manuel Branco Souto, Assistant Professor
ISCTE-IUL
Co-Supervisor
Doctor Pedro Joaquim Amaro Sebastião, Assistant Professor
ISCTE-UL

October 2019

Resumo

O grande crescimento do número de UAVs para uso civil tem sido acompanhado por um aumento do número de incidentes com esses dispositivos, causando vários problemas às autoridades, já que por vezes estes são usados em áreas restritas, o que pode levar à interrupção do tráfego aéreo e possivelmente pôr em perigo vidas humanas. Isto aumenta a necessidade de sistemas de detecção, como os radares, que por vezes têm um custo elevado. Nesta tese, é desenvolvido um sistema de Radar Contínuo Modulado na Frequência baseado em plataformas de Rádio Definido por Software (SDR) de baixo Custo, para detecção de alvos em tempo real. Para o desenvolvimento de software, o GNU Radio é usado e para a componente de hardware são estudadas duas plataformas SDR diferentes, USRP N210 e LimeSDR mini. O software desenvolvido no GNU Radio é testado através de uma série de simulações, a fim de verificar a sua capacidade de detecção da distância a que o alvo se encontra e da velocidade a que o alvo se desloca, e os resultados da detecção de alvos em tempo real para as duas diferentes plataformas SDR são apresentados e comparados. Os resultados obtidos demonstram a capacidade do software desenvolvido ser usado como parte de um sistema de Radar Contínuo Modulado na Frequência baseado em plataformas de Rádio Definido por Software de baixo custo, e também o potencial deste sistema ser usado para a detecção de alvos, desde que seja resolvido o problema do atraso adicional criado pelas plataformas SDR usadas.

Palavras-chave: Rádio definido por software, Modulação em Frequência, Radar, veículos aéreos não tripulados, Frequência de batimento

Abstract

The large growth in the number of UAVs for civil use has been accompanied by an increase in the number of incidents with these devices, causing several problems to the authorities, since sometimes they are used in restricted areas, which may lead to disruption to air traffic and possibly endanger human lives. That increases the need of detection systems, such as radars, which sometimes have a high cost. In this thesis, a Low-cost Software Defined Radio based Frequency Modulated Continuous Wave (FMCW) Radar is developed, for real time target detection. For the software part, GNU Radio is used and for the hardware part, two different SDR platforms are studied, USRP N210 and LimeSDR mini. The software developed on GNU Radio is tested through a series of simulations in order to verify its capacity to obtain the range of the target and also the speed of the target and, real time target detection results for the two different SDR Platforms are presented and compared. The results obtained demonstrate the ability of the developed software to be used as part of a Low-cost Software Defined Radio based FMCW Radar, and also the potential of this system to be used for target detection, as long as the problem of the additional delay created by the SDR platforms used is solved.

Keywords: Software Defined Radio, Frequency modulation, Radar, Unmanned Aerial Vehicles, Beat Frequency

Agradecimentos

Em primeiro lugar gostaria de agradecer ao orientador e ao co-orientador desta dissertação, o Professor Doutor Nuno Manuel Branco Souto e o Professor Doutor Pedro Joaquim Amaro Sebastião, pelo enorme apoio que me deram e pela disponibilidade, tendo sido um grande auxílio no desenvolvimento desta dissertação.

Gostaria também de agradecer à minha família por todo o apoio e paciência ao longo de todos estes meses dedicados ao desenvolvimento desta dissertação.

Queria também deixar o meu agradecimento a todos os meus amigos e colegas, especialmente ao Rui e ao João, que partilharam comigo o espaço de desenvolvimento desta dissertação e foram uma grande ajuda.

Contents

List of Figures

List of Tables

Abbreviations

ADC	Analog to Digital Converter
AGC	Automatic Gain Control
BB	Baseband
CW	Continuous Wave
DAC	Digital to Analog Converter
DDC	Digital Down Converter
DSP	Digital Signal Processor
DUC	Digital Up Converter
FFT	Fast Fourier Transform
FMCW	Frequency Modulated Continuous Wave
FPGA	Field-programmable gate array
FSPL	Free Space Path Loss
IF	Intermediate Frequency
LNA	Low Noise Amplifier
LO	Local Oscillator
LPF	Low Pass Filter
RF	Radio Frequency
RX	Reception
SDR	Software Defined Radio
SNR	Signal to Noise Ratio
TX	Transmission
UAV	Unmanned Aerial Vehicle
UHD	USRP Hardware Driver
USB	Universal Serial Bus

USRP	Universal Software Radio Peripheral
UWB	Ultra Wide Band
VCO	Voltage Controlled Oscillator

Chapter 1

Introduction

This chapter presents the motivation and context, the objectives and the contributions of this thesis.

1.1 Motivation and context

Currently, Unmanned Aerial Vehicles (UAV's) are used in various areas and for different purposes. Their applications range from defence, namely in conducting surveillance in military operations, to emergency response and disaster assessment, for example assessing damage and locating victims, among many others [1]. However, in recent years, the marketing of these devices has begun to emerge, many with very affordable prices, and it has become possible for anyone to own a drone, which is verified by the large growth in the number of UAV's for civil use [2]. This proliferation has been accompanied by an increase in the number of incidents with these devices, which has created several problems for the authorities. The use of these devices in restricted areas, including airports, is only one example of such dangers, as they disrupt air traffic and possibly endanger human lives [3]. This type of situation has led to solutions being sought all over the world to prevent these devices from circulating in certain areas. In some countries, for example, the rules on the use of drones have changed and fines have been imposed on those who do not comply with them [4]. However, despite these new rules, there are still those who insist on using the devices in restricted areas, creating several problems. For this reason, some ways have been found to prevent drones from entering these restricted areas. One example was the use of eagles by the Dutch police, who were trained to capture UAV's. However, in addition to causing injuries to the animal, this type of solution is too rudimentary and limited [5]. The ideal solution for restricting these objects should be based on radio signals, using jamming and/or spoofing techniques. However, before these restriction measures are carried out, it is extremely important to have efficient measures for locating and monitoring UAV's. One of the most appropriate measures for this task is the creation of a radar. A radar consists of the transmission of a signal in the microwave band. This signal is reflected on reaching the target. Part of this reflection, called echo, is received by the radar's receiving antenna and then sent to the receiver to obtain the direction and distance of the target. The distance is calculated using the following equation:

$$R = \frac{c\tau}{2}, \tag{1.1}$$

where τ represents the time delay between the transmitted and the received signal.

However, many of the existing radars are expensive and complex, which can make them unfeasible for many applications. The solution studied on this thesis is simpler and less costly, since it is based on Software Defined Radio (SDR) platforms. This type of solution for the development of radars as already been studied with interesting results [6] [7].

1.2 Objectives of this thesis

Taking into account the previously presented motivation, it is proposed with this work to study if an SDR based Frequency Modulation Continuous (FMCW) Radar can be implemented with GNU Radio and an SDR platform, with reduced cost and being able to perform the same functions of a regular FMCW radar. To achieve that, several steps were planned for this project: First, a theoretical study about radar systems was made. After that, several projects that developed radar systems with SDR were studied, in order to have an idea of different solutions that could be adopted. Next, a GNU Radio flowgraph was developed and studied through a set of simulations in order to verify if it was capable of performing target detection. The developed system was then used together with LimeSDR mini and with USRP N210 SDR Platform to perform real time detection of targets.

1.3 Contributions

- One paper accepted for the Wireless Personal Multimedia Communications 2019 (WPMC 2019), November 24-27 November, ISCTE-IUL, Lisboa, Portugal

Chapter 2

Basic Principles of Radar Systems and SDR platforms

This chapter contains a description of the FMCW Radar and an explanation of the SDR technology.

2.1 Radar

Radio Detection and Ranging System (RADAR) is an electronic device that allows for the detection of targets through their reflected electromagnetic energy. From that it becomes possible, in certain conditions, to calculate the distance, the direction and the speed of the target. The targets that the radar detects depend on the type of radar, but they can be cars, planes, buildings, clouds, UAV's, etc [8].

2.1.1 Radar Basic Principle

In a very simple way, a radar transmits an electromagnetic wave (of the radio type). When the signal hits a target, it gets reflected. Part of the reflected waves, also known as echo waves, are received by the RX antenna radar and sent to the radar receiver. In the radar receiver the distance and the speed of the target are calculated. To calculate the distance, since electromagnetic waves travel a distance R from the transmitter to the target and that same distance from the target to the receiver, the total distance is $2R$. As the electromagnetic waves travel at the speed of light, the distance R can be obtained with the following equation.

$$R = \frac{ct}{2}, \tag{2.1}$$

where c represents the speed of light an t represents the time it takes for the transmission of the signal and the reception of its echo [9]. Figure 2.1 represents the radar basic principle.

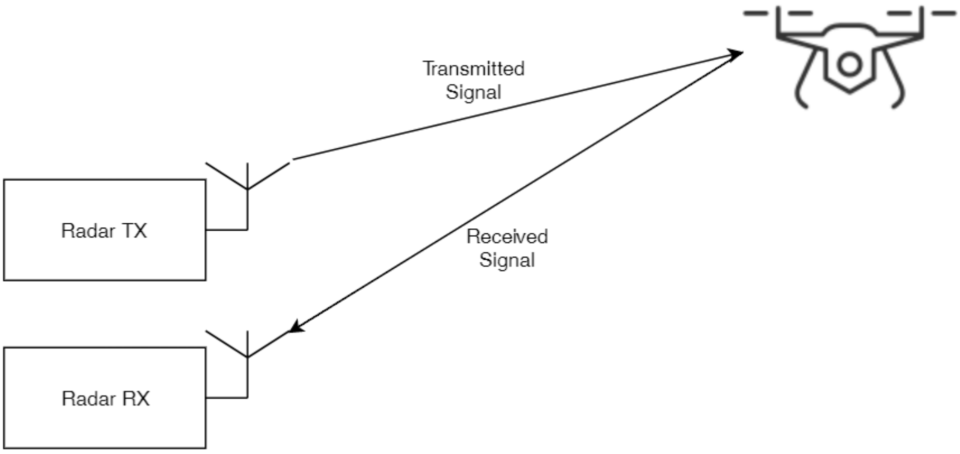


Figure 2.1: Basic detection radar principle of operation

2.1.2 Radar Basic Design

A radar usually consists of:

-Transmitter: Responsible for the creation of a RF signal that is transmitted by the TX antenna. The transmitter should have high power.

-Receiver: It is responsible for amplifying and demodulating the signal that is received by the RX antenna.

- Antennas: The antennas are responsible for the transmission and the reception of the signals. In some radars two antennas are used (one for the transmission and the other for the reception), but some radars use only one antenna for both transmission and reception. In the second one, a duplexer is used in order to alternate between the transmission and the reception [8].

2.1.3 Radar Classification

Radar Systems can be categorized according to the transmitted waveform as pulse or continuous wave radars.

A Pulse Radar transmits a short duration pulse and receive the echoes of the reflected target on the intervals of the transmissions. With the delay between the transmited and the received signal, the range of the target is obtained.

A Continuous Wave (CW) Radar, on the other hand, transmits a continuous signal and receives the reflected signal simultaneously. This makes this radar capable of obtaining the speed of the target, through Doppler effect. However, it is not capable of obtaining the range of the target, since it cannot obtain the delay of the target [8].

In order to have a continuous wave radar capable of measuring the distance of a target, a special type of CW Radar can be used, which is called Frequency Modulated Continuous Wave (FMCW) Radar.

2.1.4 Radar Definitions

- **Sample Rate**

  Since the transmitted signal has a finite bandwidth, the sample rate must comply with the Nyquist-Shannon sampling theorem, which says that the sample rate must be equal or bigger than two times the signal bandwidth.

  $$f_s \geqslant 2B \tag{2.2}$$

- **Range Resolution**

  Range resolution is defined as the minimum distance between two objects that allows to distinguish them as two different targets [10]. This value is determined by the bandwidth of the transmitted signal, through equation 2.3. Figure 2.2 demonstrates this effect, where τ represents the pulse width time.

  $$\Delta R \geqslant \frac{c}{2B} \tag{2.3}$$

- **Maximum Unambiguous Range**

  The maximum unambiguous range refers to the maximum range of a target.

  $$R_M = \frac{cT_c}{2}, \tag{2.4}$$

  where T_c refers to the chirp time.

  This value is usually much bigger than the maximum that can really be achieved, due to power limitations, and so the maximum range of a target is usually obtained from equation 2.5 [8].

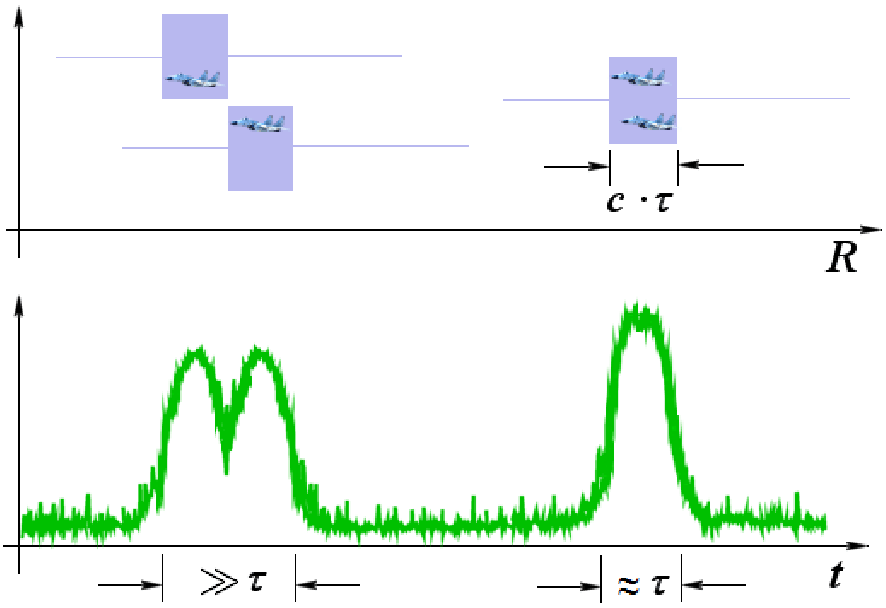


FIGURE 2.2: Range Resolution effect. Image taken from [11]

- **Radar range equation**

  The radar range equation defines the maximum achievable range of a radar, and has the following equation:

$$R_{max} = \sqrt[4]{\frac{P_t G_t G_R \lambda^2 \sigma}{(4\pi)^3 S_{min}}} \tag{2.5}$$

  , where P_t is the power transmitted, G_t and G_r are respectively transmitter and receiver antenna gain, λ is the radar operating wavelength, σ is the Radar Cross Section (RCS), and S_{min} is the minimum power detectable by the radar.

2.1.5 FMCW Radar

The FMCW radar is a particular type of a continuous wave (CW) radar in which the transmitted signal is frequency modulated. That modulation allows, unlike the regular CW radar, to measure, not only the speed of a target, but also the distance of a target [12]. This type of radar brings a series of advantages such as:

-The frequency modulation spreads the transmitted energy over a large modulation bandwidth B, providing good range resolution, critical for target discrimination in the presence of clutter.

- The power spectrum of the FMCW signal is nearly rectangular over the modulation bandwidth, making noncooperative interception difficult.

-Since the transmitted waveform is deterministic, the form of the return signals can be predicted. This makes it resistant to jamming, since any signal not matching this form can be suppressed.

-The signal processing required to obtain range information from the digitized intermediate frequency (IF) signals can be done very quickly with fast Fourier transforms (FFTs) [10].

- **Architecture Overview**

  The FMCW Radar produces an FMCW waveform, also known as chirp signal, that is a result of a carrier signal that is frequency modulated by a periodic linear signal. This periodic linear signal can either have a sawtooth or a triangular waveform. That signal is transmited by a TX antenna and the electromagnetic waves will be reflected by a target. The reflected signal is received by the RX antenna and the signal is processed by a low-noise amplifier before being mixed with a copy of the transmit FMCW waveform. After the signal being mixed, it is processed by a low-pass filter and FFT is performed in order to get range information about the target [12]. Figure 2.3 represents the basic structure of FMCW Radar.

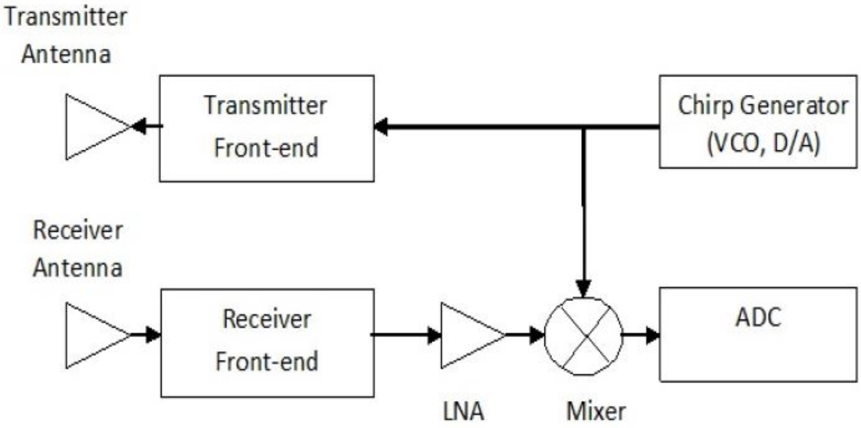


FIGURE 2.3: Block Diagram of FMCW Radar. Image taken from [7]

- **FMCW Waveform Generation**

  As described in the Architecture Overview, the FMCW waveform is a result of a carrier signal that is frequency modulated by a periodic linear signal. To achieve that, the FMCW waveform is generated with the periodic linear signal, with a sawtooth or a triangular waveform, as an input signal for a Voltage Control Oscillator (VCO), which will generate a signal whose frequency value is controlled by the amplitude of the periodic linear signal. That signal, also known as chirp is represented by Figures 2.4 and 2.5 and can be expressed by the following equation:

  $$x_t(t) = A_t \cos(2\pi(f_0 + \frac{B}{2T_c}t)t) \tag{2.6}$$

  , where A_t represents the amplitude of the transmit signal, f_0 the initial frequency, T_c the chirp period, and t the time inside a single sweep/chirp period.

- **Received Signal**

  The chirp signal is transmitted, reflected on a target, and received by the front end of the receiver. The signal that is received by the RX antenna will have a frequency lag in relation to the transmited signal, which is proportional to the time delay. The following equation represents this signal:

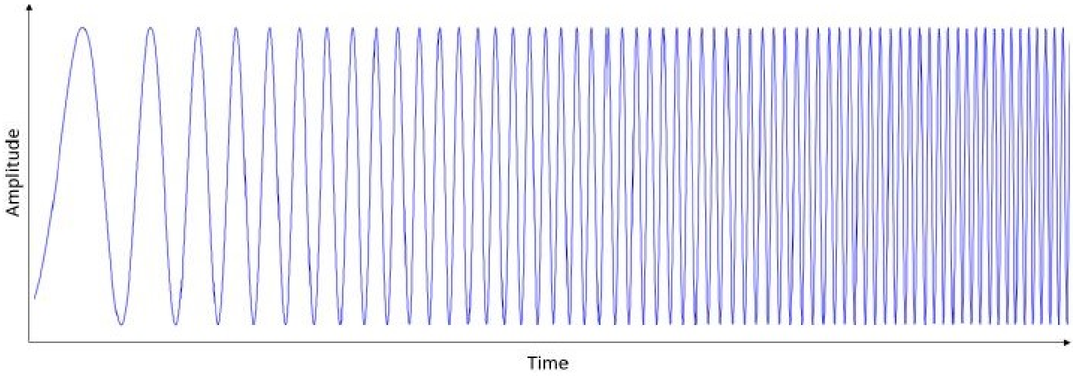


Figure 2.4: Amplitude/Time plot of the FMCW waveform. Image taken from [12]

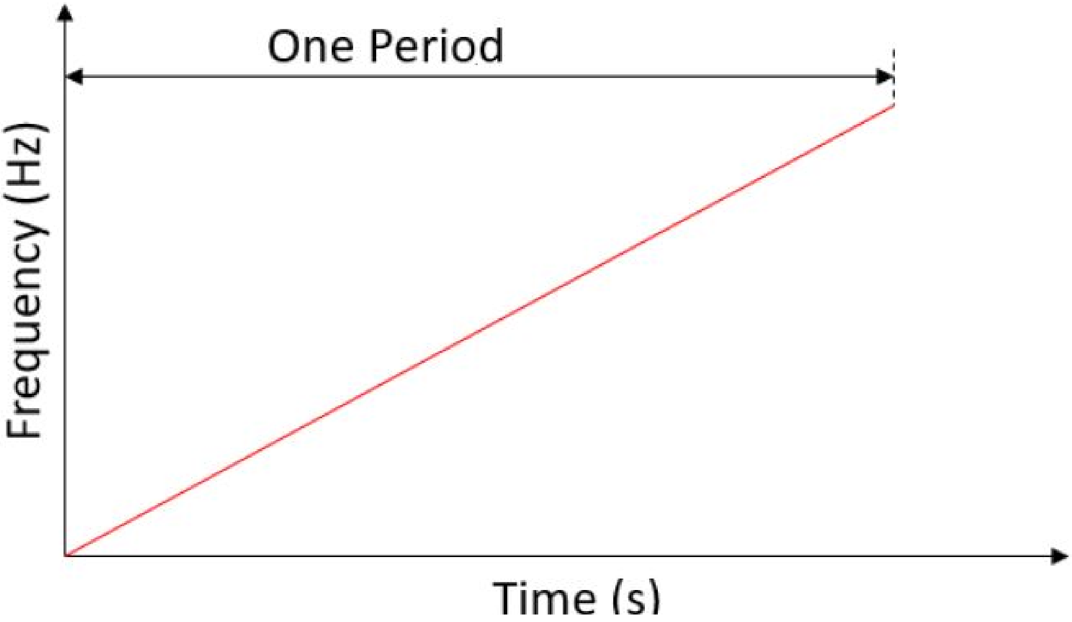


Figure 2.5: Frequency/Time plot of the FMCW waveform. Image taken from [12]

$$x_r(t) = A_r \cos[2\pi(f_0 + \frac{B}{2T_s}(t-\tau)](t-\tau), \tag{2.7}$$

where A_r represents the amplitude of the received signal, T_s the chirp period and τ the time delay between the transmission and the reception of the signal.

Figure 2.6 represents the frequency/time plot of the FMCW TX and RX signal, where τ corresponds to the time delay and the f_b value represents the

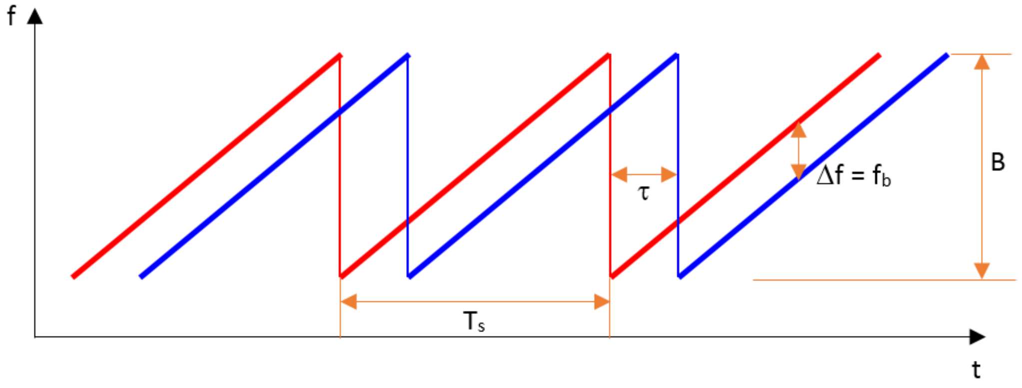


FIGURE 2.6: Frequency/Time plot of the FMCW TX and RX signal. Image taken from [13]

frequency lag, known as beat frequency, which is defined as the frequency difference between the transmitted and the received signal.

$$f_b = f_{received} - f_{transmited} \tag{2.8}$$

- **Mixer**

  The beat frequency is obtained with a mixer or a multiplier between the transmitted and the received signal. The resulting signal of the mixture of the two signals is equal to:

$$x_t(t).x_r(t) = A_t\cos(\Omega_t).A_r\cos(\Omega_r) = \frac{1}{2}A_tA_r[\cos(\Omega_t + \Omega_r) + \cos(\Omega_t - \Omega_r)] \tag{2.9}$$

$$\Omega_t + \Omega_r = 2\pi(2(f_c + \frac{B}{2T_c}t)t - \frac{B}{2T_c}\tau t - f_c\tau - \frac{B\tau}{2T_c}t + \frac{B\tau^2}{2T_c}) \tag{2.10}$$

$$\Omega_t - \Omega_r = 2\pi(\frac{B}{T_c}\tau t + f_c\tau - \frac{B\tau^2}{2T_c}), \tag{2.11}$$

where the phase-sum term represents an oscillation at twice the carrier frequency and is generally filtered out, and the phase-difference term contains the beat frequency.

$$f_b = \frac{B}{T_c}\tau \tag{2.12}$$

- **Low-pass filter**

  In order to remove the phase-sum term and only keeping the chirp signal, which contains the beat frequency, the mixed signal passes through a Low-pass filter. This Low-pass filter has a cutoff frequency inferior to the phase-sum term frequency in order to eliminate that term.

- **Target Range Value**

  The range is obtained from the beat frequency value. The beat frequency value is obtained by performing an FFT on the chirp signal and finding its maximum.

  The time delay is equal to:

$$\tau = \frac{2R}{c} \tag{2.13}$$

by replacing τ in equation 2.12:

$$f_b = \frac{B2R}{T_c c} \tag{2.14}$$

and from this equation the range of the target is obtained.

$$R = \frac{T_c c f_b}{2B} \tag{2.15}$$

- **Range and Radial velocity calculation for a moving target**

  In order to be able to detect the range and speed of a moving target, triangular waveform should be used instead of sawtooth, since with sawtooth waveform it is not possible to observe the Doppler Shift Effect and it makes radial velocity calculation impossible [10].

  When a target moves, the beat frequency will suffer a Doppler Frequency Shift, as represented in the following equation.

  $$f_d = \frac{2v}{\lambda}, \tag{2.16}$$

  where v is the speed of the target and λ is the wavelength of the radar. This will generate two different frequencies: a summed part and a subtracted part, as represented in equations 2.17 and 2.18.

  $$f_{bu} = f_b - f_d \tag{2.17}$$

  $$f_{bd} = f_b + f_d \tag{2.18}$$

  From these two frequencies, the range and the speed of the target can be obtained as

  $$v = \frac{\lambda}{4}(f_{bd} - f_{bu}) \tag{2.19}$$

  $$R = \frac{T_c c}{4B}(f_{bd} + f_{bu}) \tag{2.20}$$

  The Doppler Shift Effect is represented by Figure 2.7

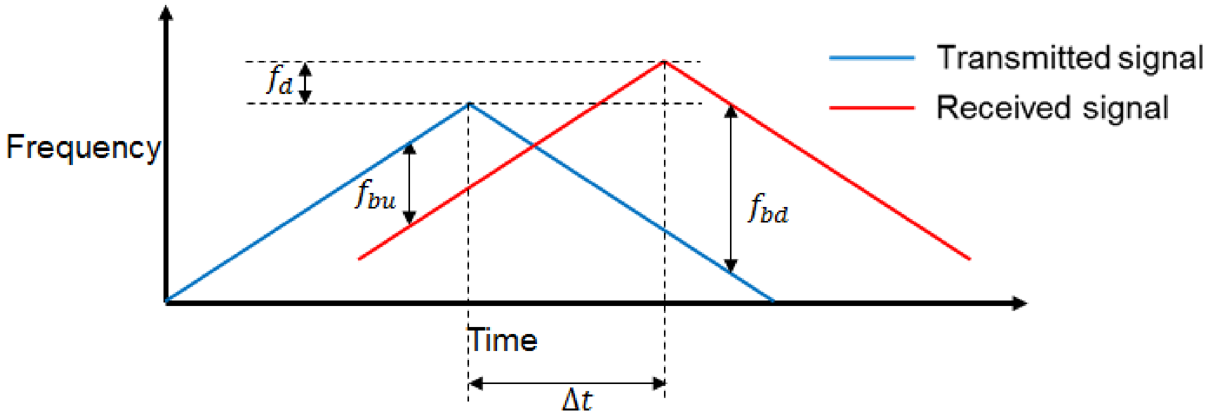


FIGURE 2.7: Doppler Shift Effect on FMCW Radar . Image taken from [14]

2.2 Software Defined Radio

Software Defined Radio is a radio communication system where signal processing components normally implemented in hardware, such as modulators, demodulators, filters, amplifiers, etc., are implemented using software on a computer, or other programming device, like a field-programmable gate array (FPGA). The main advantages of using SDR are:

- Flexibility: Since most of the operations of an SDR are defined by the software on the system, it is very easy to modify or upgrade the software, to provide a different behaviour to the system.

- Common hardware that implements SDR and a Digital Signal Processor (typically a host computer) are used to develop the system.

- Low cost: The systems have simple architectures which result in smaller costs.

However SDR still have some limitations such as:

- ADC Sampling Rate: Most RF applications work in the GHz region, which means that the required sample rate cannot be achieved by most analog-digital converters.

- Antennas are designed to operate in a specified frequency band, limiting the operating frequency of SDR.

- The processor speed of some computers might not be sufficient to perform realtime processing of signals [15].

2.2.1 SDR Architecture

The design of a receiving and transmitting SDR consists of an antenna, an SDR Platform that contains a wideband RF front-end, a down respectively up-converter, an ADC/DAC, and an FPGA, and a host computer connected by a USB or a Gigabit Ethernet Connection [15].

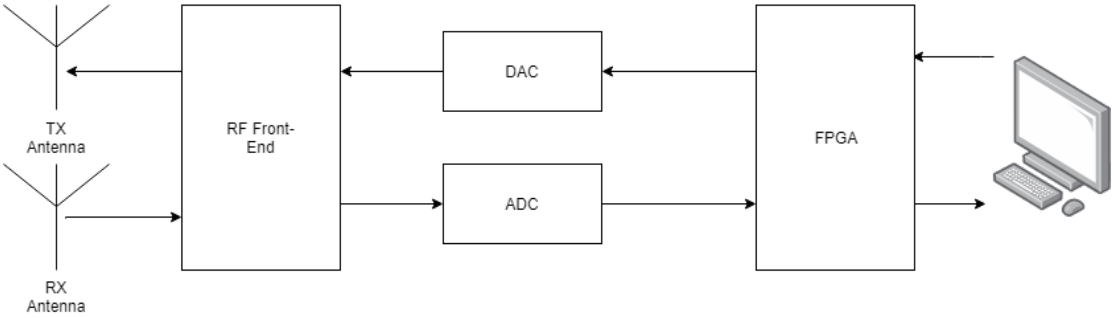


Figure 2.8: SDR generic block diagram

- **Host Computer**

  The host computer is responsible for sending and receiving signals to/from the SDR platforms and performing signal processing, like modulation, filtering and FFT. It also controls basic hardware parameters such as sample rate, operating frequency, and gain of the transmitter/receiver. Normally, it uses software tools, such as Matlab and GNU Radio.

- **FPGA**

  The implementation of an SDR can either have most processing made on the FPGA, or most processing taking place on a computer, using GNU Radio, for example, with the FGPA only performing basic operations. In this dissertation, most of the processing is made on a computer with GNU Radio, meaning that the FPGA only works as a Digital Front End, having two main functions:

- Sample Rate Conversion, which is a functionality to convert the sampling rate from one rate to another, which is necessary because the host computer and the SDR Platform must be synchronized.

- Channelization, which includes up/down conversion in the transmitter and the receiver and channel filtering.

This process is done by:

- On the transmitting side, the Digital Up Converter (DUC) translates the baseband signal, sampled by f_s, to the sample rate of the hardware and sends the signal to the DAC.

- On the receiving side, the Digital Down Converter (DDC) receives the signal from the ADC and extracts the baseband signal [16].

- **Analog-to-Digital and Digital-to-Analog Converter**

  The DAC converts the digital signal, received from the DUC, to an analog signal. The ADC receives the analog signal from the RF Front End and converts it to a digital signal.

- **RF front-end**

  The RF front-end main functions are to transmit and receive the signals at various operating frequencies and to change the signal to/from the Intermediate Frequency (IF) or baseband signal (BB), to/from the RF operating frequency. On the transmission part, digital samples are converted into an analog signal by the Digital-to-Analog Converter, which in turn feeds the RF front end. This analog signal is mixed with the RF carrier, modulated and then transmited. On the receiving path, the antenna captures the RF signal. The signal passes through a Low Noise Amplifier, to amplify weak signals and reduce the noise. The signal is mixed with a signal from the

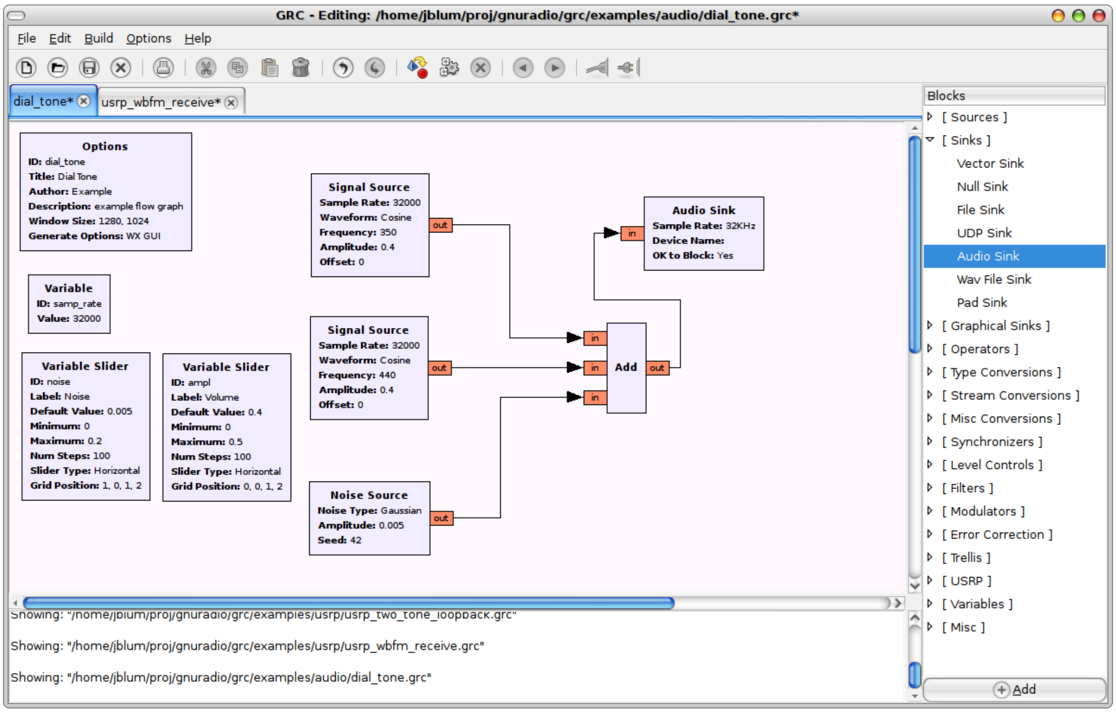
Local Oscillator (LO) in order to down convert it to the IF, in cases where the SDR board is not zero-IF [16].

Zero-IF boards

Zero-IF, sometimes called a direct conversion or homodyne architecture, uses a LO which is set equal to the desired carrier frequency being tuned, or very close to it. This reduces the IF frequency to zero or near zero, rather than standard values such as 455 kHz. This kind of technique is used by some SDR boards, such as LimeSDR mini and BladeRF. The advantage is that it eliminates some of the IF-related circuitry, and also avoids the problem of "image frequencies" which are an inherent part of the mixer output in the heterodyne architecture. The zero-IF receiver architecture is also attractive because it can ease demodulation of very wideband RF signals [17]. Despite these advantages there are also some well known disadvantages, such as self-detection due to LO-RF leakage and DC offset. The first problem is caused by the fact that the local oscillator in a direct-conversion receiver is at the exact same frequency as the desired signal. Hence, any leakage of the LO signal to the RF input or the antenna will pass through the entire signal chain, which appears as a DC offset. The second problem is the inherent DC offset in the baseband amplifier stage, and its drift over temperature [18].

- **Software Tool**

  In this dissertation, the software development tool used is GNU Radio. GNU Radio is a software development tool that provides signal processing functions to implement SDR. The digital processing blocks are developed in C++ and the link between the various blocks is developed in Python. However, an integrated software environment, such as GNU Radio Companion, can be used to realize the applications through a graphical interface [19]. The software developed on GNU Radio can be used without any SDR board in order to perform signal processing in a simulation environment, or in conjunction with a board, such as the USRP N210 or the Lime SDR Mini, both used in this dissertation. For this purpose, the hardware drivers are used. These drivers are responsible for accessing the SDR platforms with the software. On GNU Radio, there are specific blocks for these drivers.



GRC - Editing: /home/jblum/proj/gnuradio/grc/examples/audio/dial_tone.grc*

Options
ID: dial_tone
Title: Dial Tone
Author: Example
Description: example flow graph
Window Size: 1280, 1024
Generate Options: WX GUI

Variable
ID: samp_rate
Value: 32000

Variable Slider
ID: noise
Label: Noise
Default Value: 0.005
Minimum: 0
Maximum: 0.2
Num Steps: 100
Slider Type: Horizontal
Grid Position: 1, 0, 1, 2

Variable Slider
ID: ampl
Label: Volume
Default Value: 0.4
Minimum: 0
Maximum: 0.5
Num Steps: 100
Slider Type: Horizontal
Grid Position: 0, 0, 1, 2

Signal Source
Sample Rate: 32000
Waveform: Cosine
Frequency: 350
Amplitude: 0.4
Offset: 0

Signal Source
Sample Rate: 32000
Waveform: Cosine
Frequency: 440
Amplitude: 0.4
Offset: 0

Noise Source
Noise Type: Gaussian
Amplitude: 0.005
Seed: 42

Audio Sink
Sample Rate: 32KHz
Device Name:
OK to Block: Yes

```
Showing: "/home/jblum/proj/gnuradio/grc/examples/usrp/usrp_two_tone_loopback.grc"

Showing: "/home/jblum/proj/gnuradio/grc/examples/usrp/usrp_wbfm_receive.grc"

Showing: "/home/jblum/proj/gnuradio/grc/examples/audio/dial_tone.grc"
```

FIGURE 2.9: GNU Radio interface

- **SDR Platforms**

  The SDR Platforms are the hardware components of the SDR. These systems consists of an FPGA with components to provide the ADC and DAC functionality and an RF front end. The platforms can be connected with GNU Radio (as in this case) with an USB cable or a Gigabit Ethernet cable to create an SDR system.

2.3 Software Defined Radar

Software Defined Radar is a radar system that applies the same principles of Software Defined Radio in order to create a system where most of the processing, like signal generation, filtering, up and down conversion, etc, is performed by software, generally on a host computer. This brings many advantages such as:

- the possibility to create a multipurpose radar;

- the possibility to re-use the same hardware;

- easier implementation of advanced signal processing algorithm;

- faster development and reduced price. [20]

2.3.1 Software Defined Radar Architecture

On the transmitting side, the radar waveform generator produces a complex type radar waveform. The in-phase and quadrature components are then sent to the SDR Platform. After passing DUC, DAC and LPF, the in-phase component of the signal is mixed with the in-phase component of the local oscillator (LO), and the same process happens for the quadrature components. The signal that is obtained in the output of the RF module corresponds to the FMCW waveform and that signal is the one send to the air by the SDR Platform. On the receiving side, the signal is amplified by the LNA, mixed with the radar operating frequency and extracted into in-phase and quadrature components. The signal then passes on LPF to eliminate high frequency components, passes on the ADC to be digitized and is sampled down to the original sample rate with DDC. That signal is then sent to the signal processor, in order to obtain the target range.

2.3.2 Literature Review

Software Defined Radio has been used to implement radar systems, with GNU Radio being used to develop the software and with USRP as front end.

[6] presents a Software defined FMCW Radar with GNU Radio and the USRP N210 for weather surveillance. The prototype of this system is realized by use of GNU Radio and USRP N210. GNU Radio is used to generate the FMCW waveform and to mix the transmitted and received signal components in order to obtain the beat signal. To perform low pass filtering and FFT, in order to obtain the range of the target, Matlab is used. To transmit and receive the reflected signals, USRP N210 is connected with the host computer and with two UWB antennas (one for the transmitted signal and the other for the received signal). The studied system has a working frequency of 2.1 GHz, and uses a sawtooth waveform, with a bandwidth of 0.75 MHz, chirp period of 1 ms and a sampling rate of 6 Ms/s.

In [12], a series of SDR hardware platforms are investigated and their performance is compared. Of the studied platforms it is concluded that both USRP N210 and the QM-RDKIT can be used, since they have suitable operating frequency, bandwidth, sampling rate, uncomplicated programming task, and friendly software support. After this study, two FMCW Radar systems are developed and they performance is compared. One of the systems used the QM-RDKIT, operating in the 2.4 GHz frequency band, with a pair of HyperLink HG2418P antennas, which are suitable for that band. The other system used USRP N210, operating at a 2.1 GHz frequency band, with a pair of UL-235A-498 horn antennas, which are suitable for 2.1 GHz band, and GNU Radio for the software part of the FMCW radar. Both systems used the Mini-Circuits ZVE-2W-272+ as a power amplifier, since its operation covers both the 2.1 GHz and the 2.4 GHz band. The developed systems are then used in a series of outdoor experiments and their behaviour is analyzed.

In [21], USRP N210 and GNU Radio are used to obtain the radiation pattern of a log periodic antenna. The setup developed uses two USRP N210 with WBX Daughterboards. One USRP connected to an Omni directional antenna serves as the transmitter and the other USRP with the log-periodic antenna is at the reception part. The antennas are kept in parallel at a distance of 6m. At the transmitting end, a single tone of frequency 100 KHz is mixed with a signal of central frequency 500 MHz. On the receiving end, the received signal from the log periodic antenna passes through a Low Pass Filter and the received signal spectrum is observed. By changing the angle of the log-periodic antenna and manually noting down the power obtained at the FFT plot, the radiation pattern of the log-periodic antenna is created.

Chapter 3

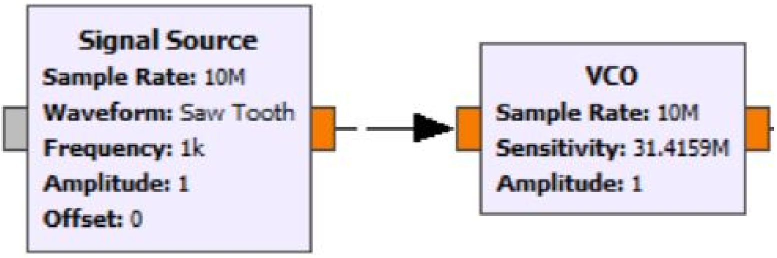
Implementation and simulation of the proposed FMCW radar

This chapter presents the software developed with GNU Radio for an FMCW Radar and a series of simulations to verify its ability to detect targets. This software will then be used in Chapter 4 for the experimental results, with different SDR Platforms.

3.1 Developed Software

3.1.1 FMCW Waveform Generation

The Signal Source and the VCO blocks are responsible for generating the transmitting signal. The Signal Source is responsible for the production of the modulating signal, and the VCO generates the chirp signal through the modulating signal created by the signal source. For the chirp signal, the frequency varies according to the bandwidth defined in this block, during the chirp time defined in the signal source. The bandwidth value is defined by the sensitivity parameter of the VCO. The sensitivity value is an angular frequency, so the desired bandwidth must be multiplied by 2π. Figure 3.1 represents the FMCW waveform generation on GNU Radio.


Signal Source
Sample Rate: 10M
Waveform: Saw Tooth
Frequency: 1k
Amplitude: 1
Offset: 0

VCO
Sample Rate: 10M
Sensitivity: 31.4159M
Amplitude: 1

Figure 3.1: FMCW Waveform generation on GNU Radio

3.1.2 Digital Signal Processing

- **Mixer**

  The transmited signal is reflected on a target and the received signal is multiplied with the chirp signal, with the Multiply Block. The Multiply block serves, as the name indicates, to multiply the received and transmitted signals. From that multiplication, two different terms will appear on the signal, as shown on equations 2.10 and 2.11, and from the term represented in equation 2.11 the beat frequency is obtained. Figure 3.2 represents the mixing of the transmitted signal with the simulated received signal, created through the delay block.

- **Low-pass Filter**

  As previously mentioned, the multiplication of the transmitted and received signals generates a component in which there is a sum of frequencies of the two signals, as shown in equation 2.10, and these frequencies do not represent the beat signal. The beat frequency is represented by the component in which the frequencies of the signals are subtracted, as shown in equation 2.11. Thus, the use of the low-pass filter aims to eliminate the component of high frequencies.

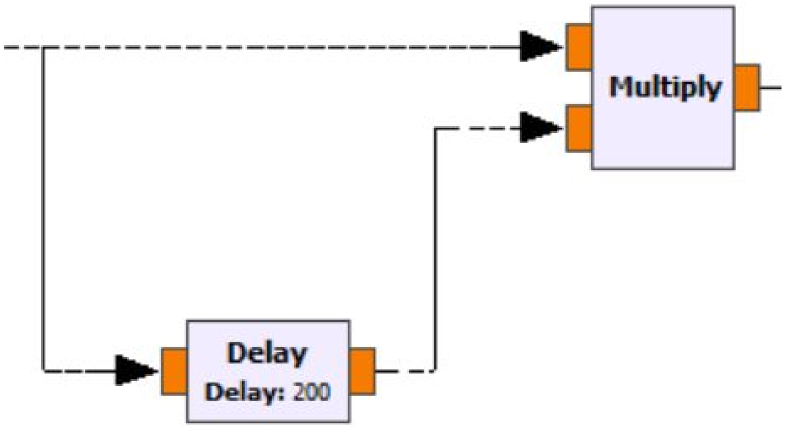


Figure 3.2: Mixing of transmitted and received signals

- **Target Range Calculation**

  By performing an FFT over the signal resulting from the multiplication between the transmitted signal and the received signal, it is then possible to obtain the beat frequency. Through the following equation:

  $$R = \frac{cfT}{2B} \tag{3.1}$$

  it is then possible to obtain the distance at which the simulated target is located.

3.2 Simulation Parameters

3.2.1 Sample Rate Value

The sample rate value will influence the bandwidth and subsequently the range resolution. In this system the chosen value for the sample rate was 10 Msps, since according to the Nyquist theorem, the maximum bandwidth must be equal to 5 MHz, which translates in a range resolution of 30 meters, and because, even though a higher value would increase the range resolution, it would require more computational load.

3.2.2 Bandwidth Value

For the bandwidth, the value to be used must comply with the Nyquist theorem, so the maximum bandwidth of the signal is equal to 5 Mhz. The value of the bandwidth translates in a range resolution of 30 meters, which can be to high to distinguish targets, but as said before, to achieve a bigger bandwidth, a bigger value for the sample rate was needed, and that would require a bigger computational load.

3.2.3 Waveform

For the static target simulations, the sawtooth waveform was used since it is simpler, but for the moving target simulations, the triangular waveform was then used.

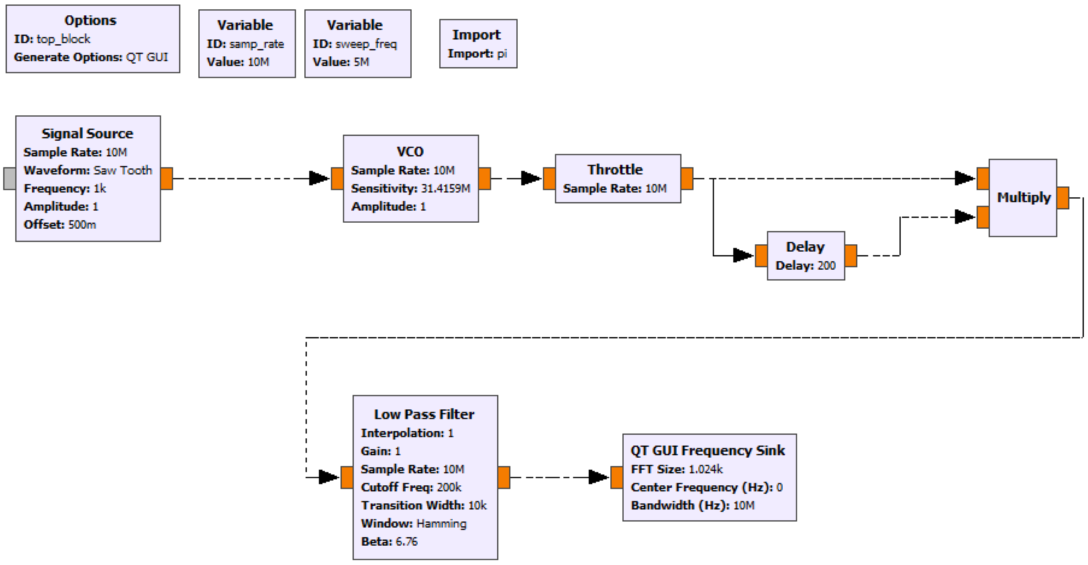
The following table summarize the specifications for the FMCW Radar simulations.

Table 3.1: FMCW Radar Simulation Parameters

Parameter	**Value**
Sample Rate	10MS/s
Waveform	Sawtooth and Triangular
Chirp Period	1ms
Bandwidth	5MHz
Range Resolution	30m
Maximum Unambiguous Range	150km

3.3 Simulation for a static target

The first simulation was developed to test the flow graph behavior in the detection of a static target. The signal source and VCO are responsible for the FMCW waveform generation. The throttle is used to prevent the average rate to exceed the specific rate. The delay block is used to simulate the received signal. After the mixture, the signal passes trough the low-pass filter in order to eliminate the phase-sum term of the mixed signal. The FFT of this signal is then represented trough QT GUI Frequency Sink block, in order to obtain the beat frequency value. Figure 3.3 represents the flowgraph for this simulation.



Options
ID: top_block
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 10M

Variable
ID: sweep_freq
Value: 5M

Import
Import: pi

Signal Source
Sample Rate: 10M
Waveform: Saw Tooth
Frequency: 1k
Amplitude: 1
Offset: 500m

VCO
Sample Rate: 10M
Sensitivity: 31.4159M
Amplitude: 1

Throttle
Sample Rate: 10M

Delay
Delay: 200

Multiply

Low Pass Filter
Interpolation: 1
Gain: 1
Sample Rate: 10M
Cutoff Freq: 200k
Transition Width: 10k
Window: Hamming
Beta: 6.76

QT GUI Frequency Sink
FFT Size: 1.024k
Center Frequency (Hz): 0
Bandwidth (Hz): 10M

FIGURE 3.3: Simulation of static target detection

For this simulation, a delay of 200 samples was used, which means that the delay is 20 microseconds, from equation 3.2, which corresponds to a range of 2998 meters, from equation 1.1. According to equation 2.14 the expected beat frequency is 100 kHz. For this reason the cutoff frequency of the low-pass filter is 200 kHz, because the phase-sum term represents an oscillation at twice the carrier frequency.

$$\tau = \frac{N_s}{f_s} = 20\mu s, \tag{3.2}$$

where N_s represents the number of samples, and f_s the sample rate.

The observation of the graph shows that the beat frequency obtained with the simulation is in agreement with the theoretical value, which means that this flowgraph is capable of obtaining the range of the target.

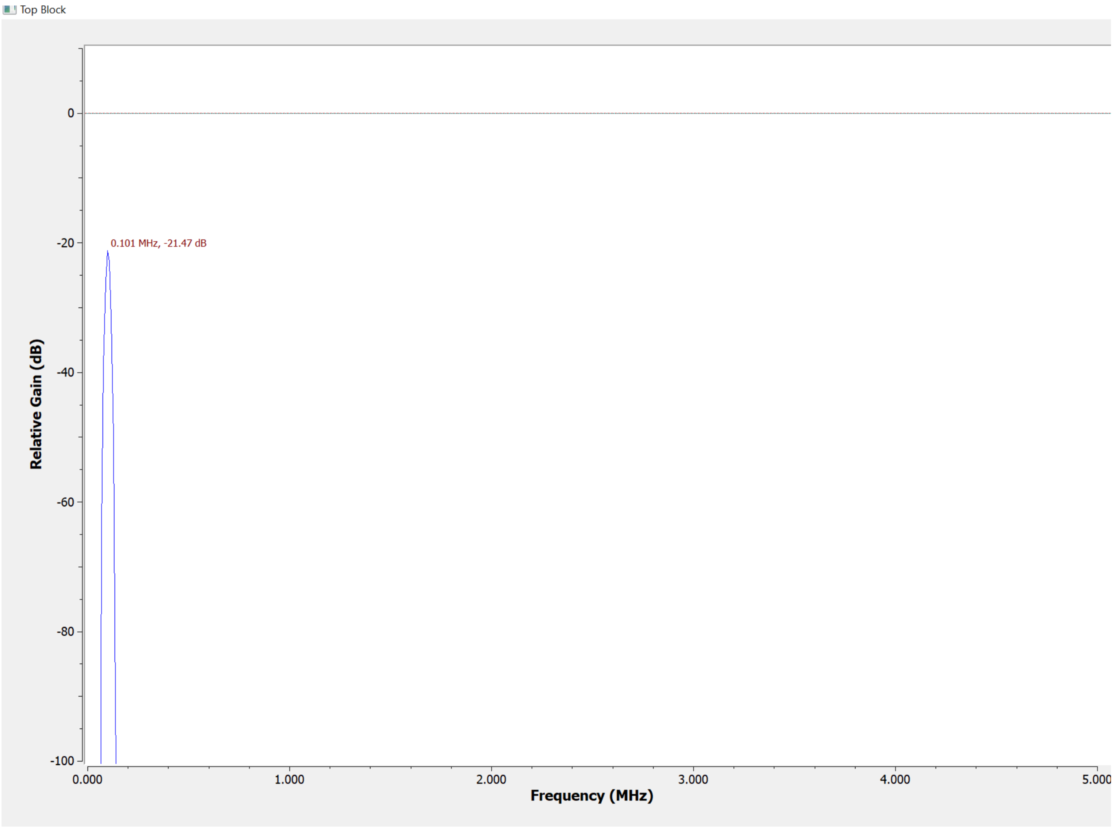


FIGURE 3.4: Static target beat frequency

3.4 Simulation for three static targets

This simulation aimed to verify the behaviour of the flow graph when it has to detect more than one target. To simulate that, two more delay blocks were added to the previous flowgraph, one of 400 samples and the other of 600 samples. Figure 3.5 represents this flowgraph.

For this simulation the expected values for the beat frequencies are 100, 200 and 300 kHz.

Fig. 3.6 shows that this flowgraph is capable of detecting various targets, as long as they have a separation bigger than the range resolution.

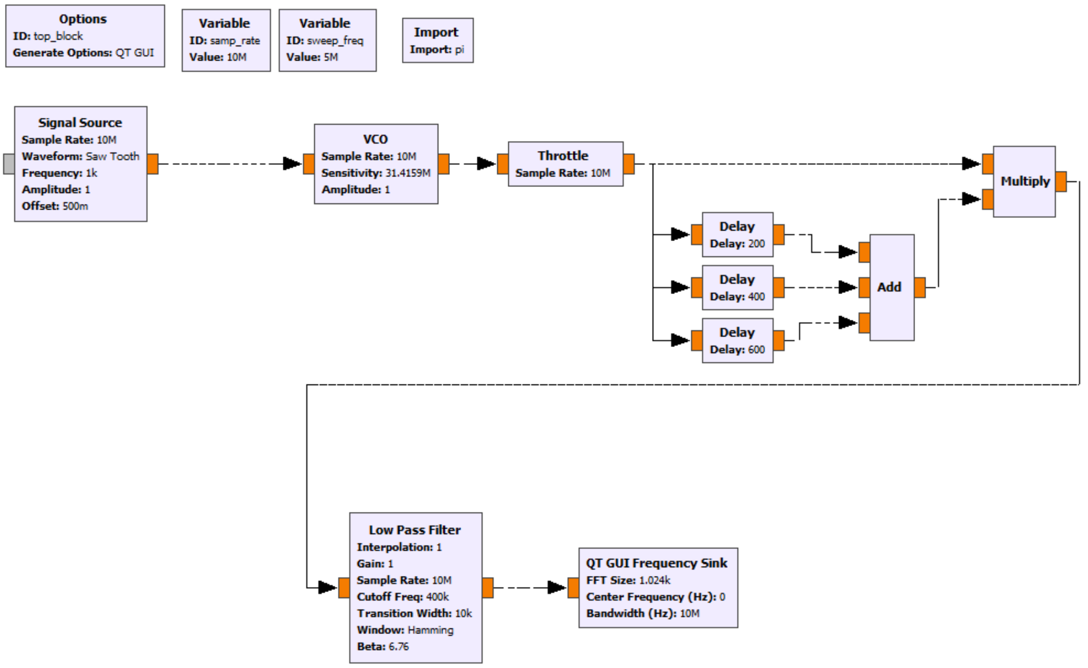


FIGURE 3.5: Three static target flowgraph

3.5 Simulation of a moving target

This simulation was made to verify if the proposed implementation was capable of detecting a slowly moving target. To achieve that, the delay block has a variable value. That variation was made with a signal that varies the delay between a set of two values, and that value is being saved in a variable within the probe function block. The target was simulated to move at a speed of 15 m/s by varying the delay between 100 samples and 200 samples during approximately 100 seconds. Figure 3.7 represents this simulation.

It was observed that the beat frequency changes between 50 kHz (for 100 samples), as shown on Figure 3.8 and 100 kHz (for 200 samples), as shown on Figure 3.9. For this flowgraph, the Doppler shift was not taken under consideration.

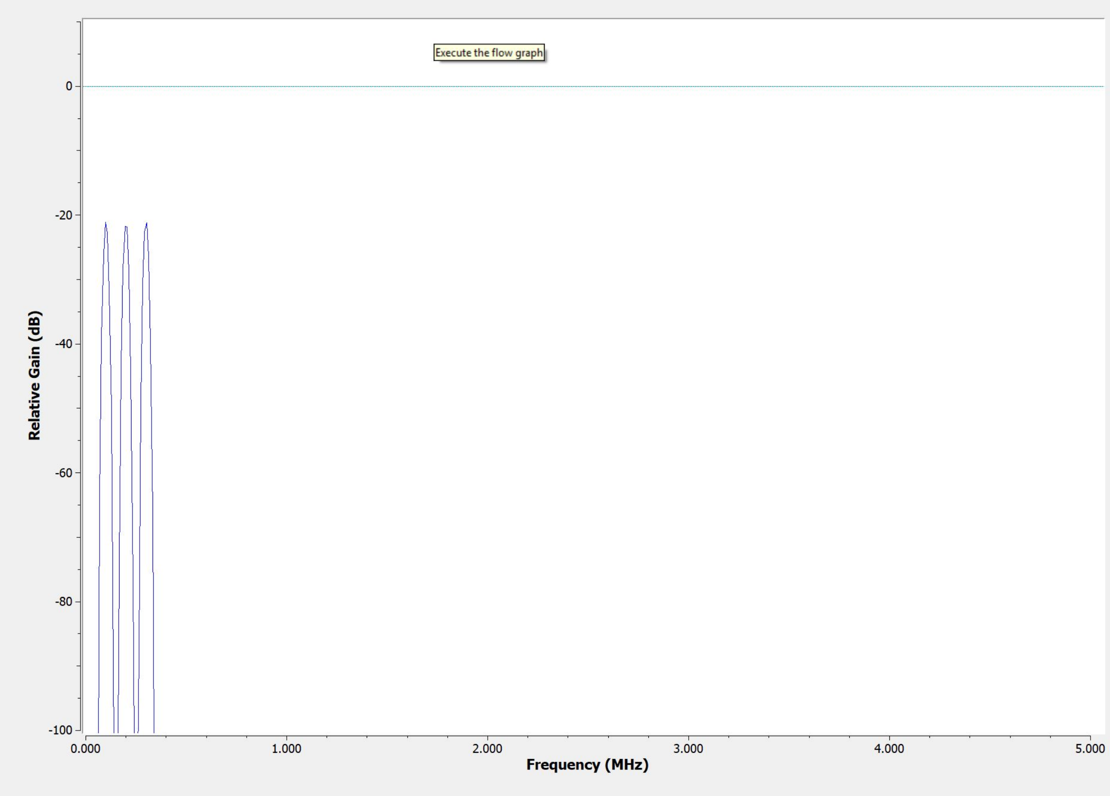
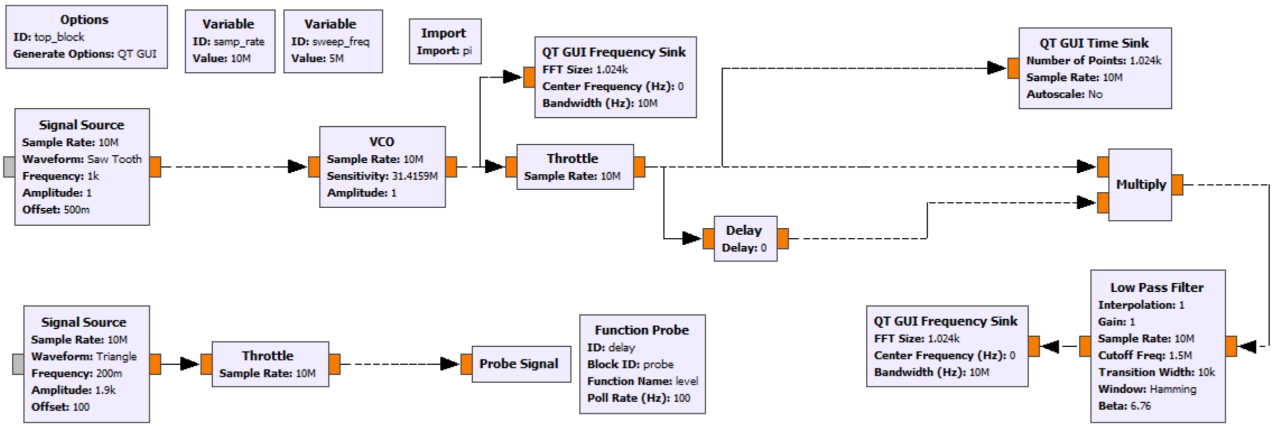


FIGURE 3.6: Beat Frequency for three targets



Options
ID: top_block
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 10M

Variable
ID: sweep_freq
Value: 5M

Import
Import: pi

QT GUI Frequency Sink
FFT Size: 1.024k
Center Frequency (Hz): 0
Bandwidth (Hz): 10M

QT GUI Time Sink
Number of Points: 1.024k
Sample Rate: 10M
Autoscale: No

Signal Source
Sample Rate: 10M
Waveform: Saw Tooth
Frequency: 1k
Amplitude: 1
Offset: 500m

VCO
Sample Rate: 10M
Sensitivity: 31.4159M
Amplitude: 1

Throttle
Sample Rate: 10M

Multiply

Delay
Delay: 0

Signal Source
Sample Rate: 10M
Waveform: Triangle
Frequency: 200m
Amplitude: 1.9k
Offset: 100

Throttle
Sample Rate: 10M

Probe Signal

Function Probe
ID: delay
Block ID: probe
Function Name: level
Poll Rate (Hz): 100

QT GUI Frequency Sink
FFT Size: 1.024k
Center Frequency (Hz): 0
Bandwidth (Hz): 10M

Low Pass Filter
Interpolation: 1
Gain: 1
Sample Rate: 10M
Cutoff Freq: 1.5M
Transition Width: 10k
Window: Hamming
Beta: 6.76

FIGURE 3.7: Flow graph of moving target simulation

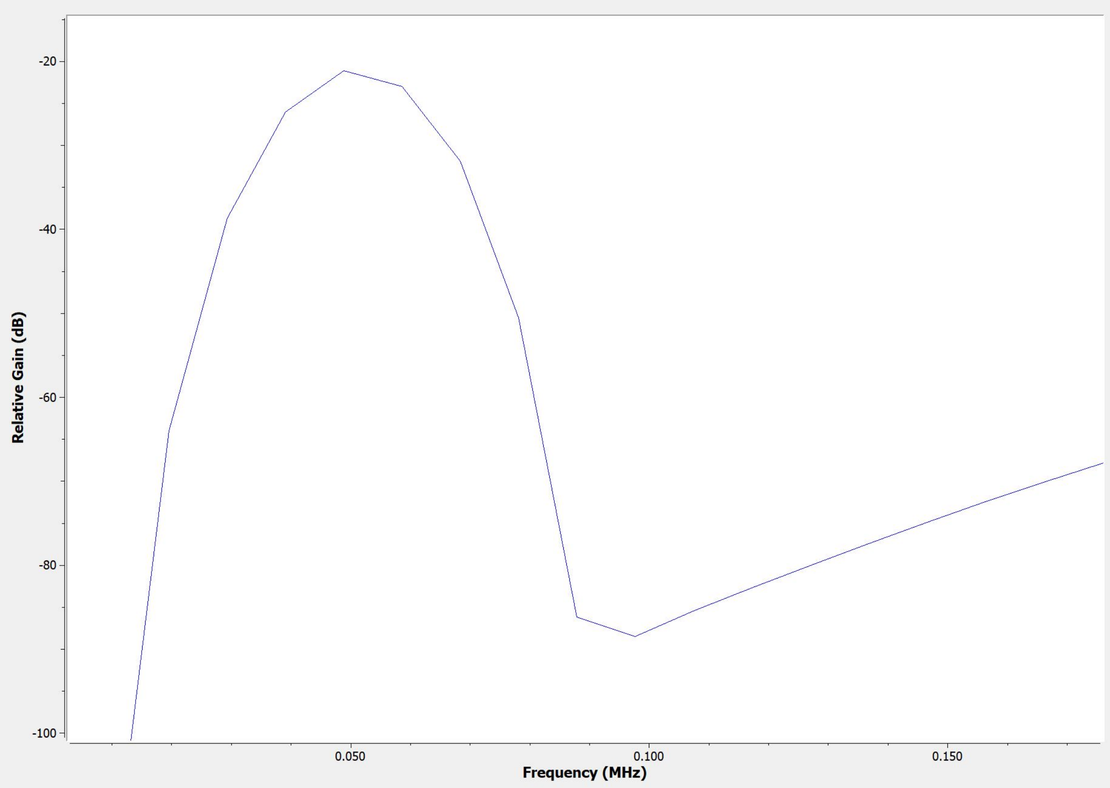


FIGURE 3.8: Minimum frequency

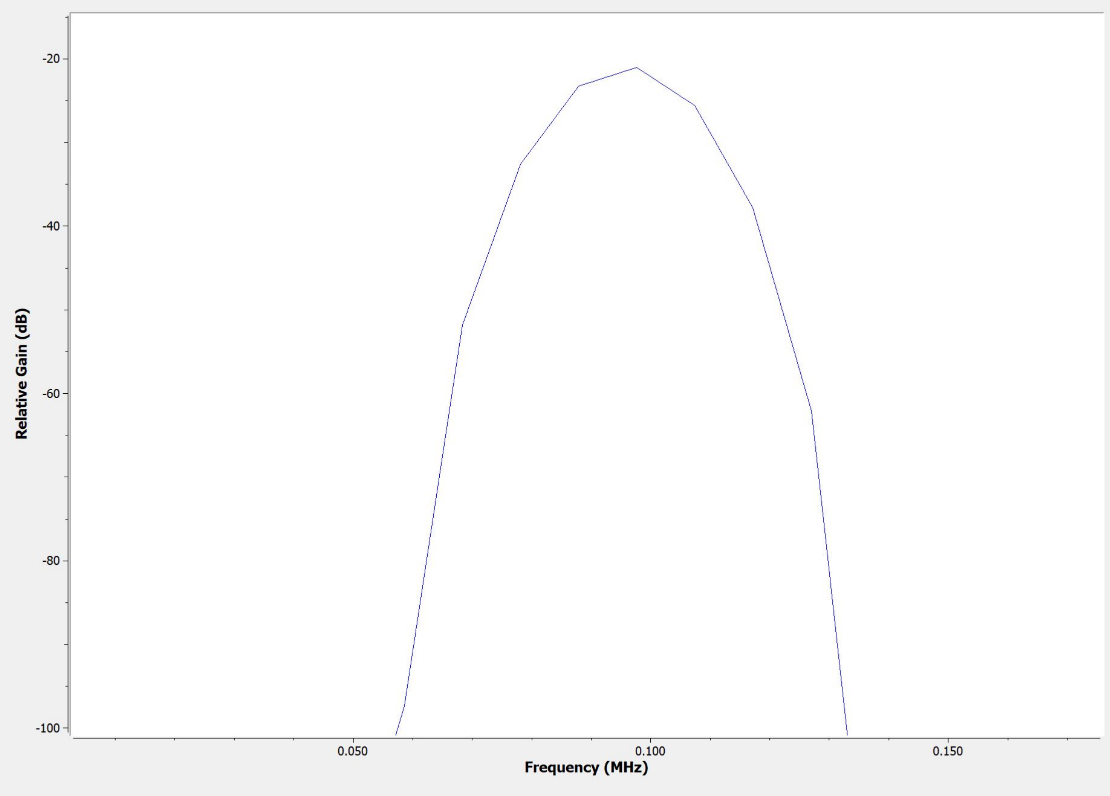
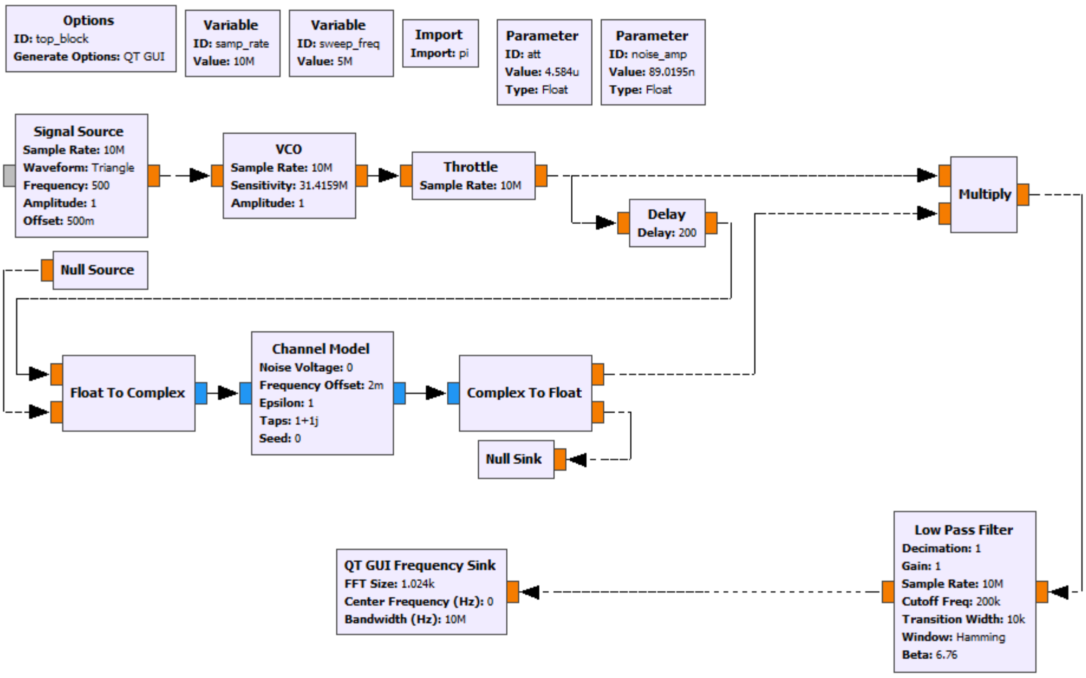


FIGURE 3.9: Maximum frequency

3.6 Simulation of a moving target with Doppler Shift Effect

For this simulation, a frequency offset (f_d) of 20kHz was added to the received signal in order to simulate the Doppler frequency from a moving target. The frequency offset was added through the Channel Model block. For this simulation, a triangular waveform was used since it makes it easier to detect the two beat frequencies. Figure 3.10 shows the designed flowgraph of this simulation.



Options
ID: top_block
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 10M

Variable
ID: sweep_freq
Value: 5M

Import
Import: pi

Parameter
ID: att
Value: 4.584u
Type: Float

Parameter
ID: noise_amp
Value: 89.0195n
Type: Float

Signal Source
Sample Rate: 10M
Waveform: Triangle
Frequency: 500
Amplitude: 1
Offset: 500m

VCO
Sample Rate: 10M
Sensitivity: 31.4159M
Amplitude: 1

Throttle
Sample Rate: 10M

Delay
Delay: 200

Channel Model
Noise Voltage: 0
Frequency Offset: 2m
Epsilon: 1
Taps: 1+1j
Seed: 0

Low Pass Filter
Decimation: 1
Gain: 1
Sample Rate: 10M
Cutoff Freq: 200k
Transition Width: 10k
Window: Hamming
Beta: 6.76

QT GUI Frequency Sink
FFT Size: 1.024k
Center Frequency (Hz): 0
Bandwidth (Hz): 10M

FIGURE 3.10: Flow graph of Doppler frequency effect simulation

Fig. 3.11 shows that two frequencies appear, instead of only the beat frequency. One of the frequencies is equal to $f_b - fd = 80kHz$ and the other is equal to $f_b + fd = 120kHz$.

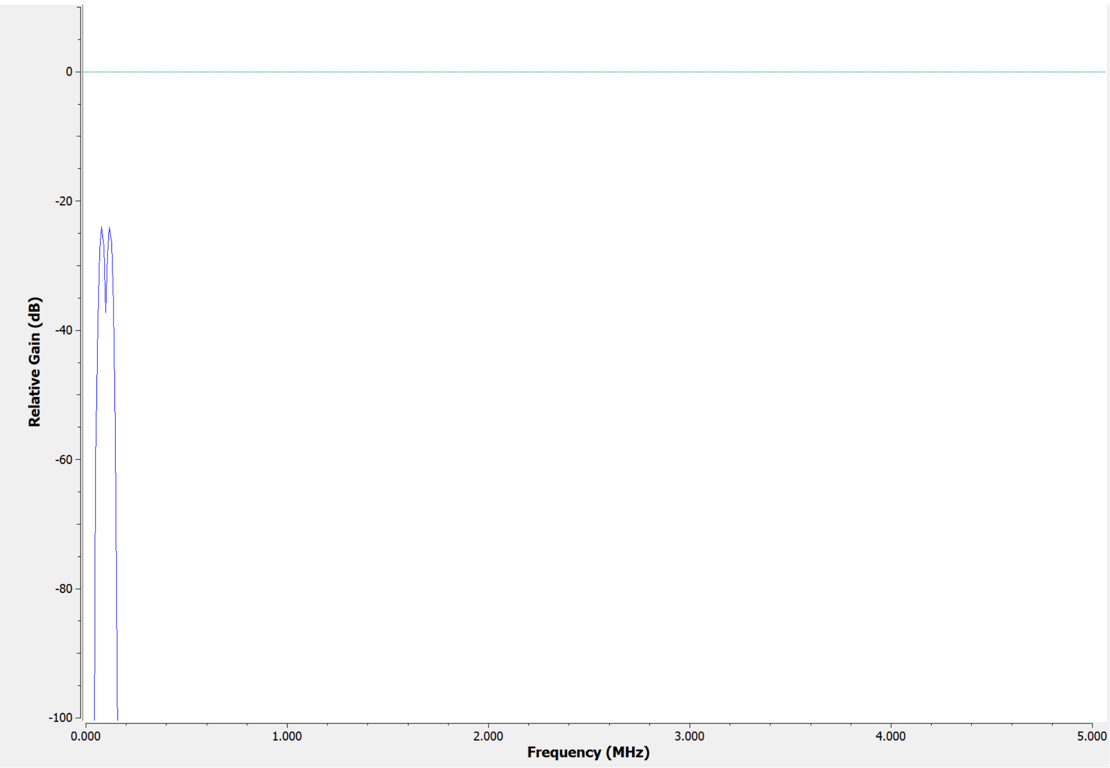


FIGURE 3.11: Doppler Shift effect on the beat frequency

3.7 Effect of Noise and Signal Propagation

When a signal is received in real time, it suffers attenuation due to propagation through the channel and corruption effects due to noise, which can make it difficult to detect a target. This simulation aims to verify if those effects still make it possible for the developed FMCW Radar to detect targets. The noise effect has the following equation for the noise power:

$$N = 10^{\frac{-178+10\log(B)-30}{10}} [W] \tag{3.3}$$

The attenuation is calculated with the following equation assuming free-space path loss:

$$FSPL = \sqrt{(\frac{\lambda}{4\pi 2R})^2}, \tag{3.4}$$

where R represents the range of the target and λ the wavelength of the radar.

For this simulation it was used a range value for an object delayed 2998m, which is about 200 samples, from equation 3.2 and a bandwidth of 868 MHz. That gives a noise value with a power of $8.9 \cdot 10^{-8}$ W and an attenuation of $4.584 \cdot 10^{-6}$. To simulate this, the channel model block was used with the noise voltage parameter equal to $N \cdot \frac{1}{FSPL}$. Figure 3.12 represents this simulation.

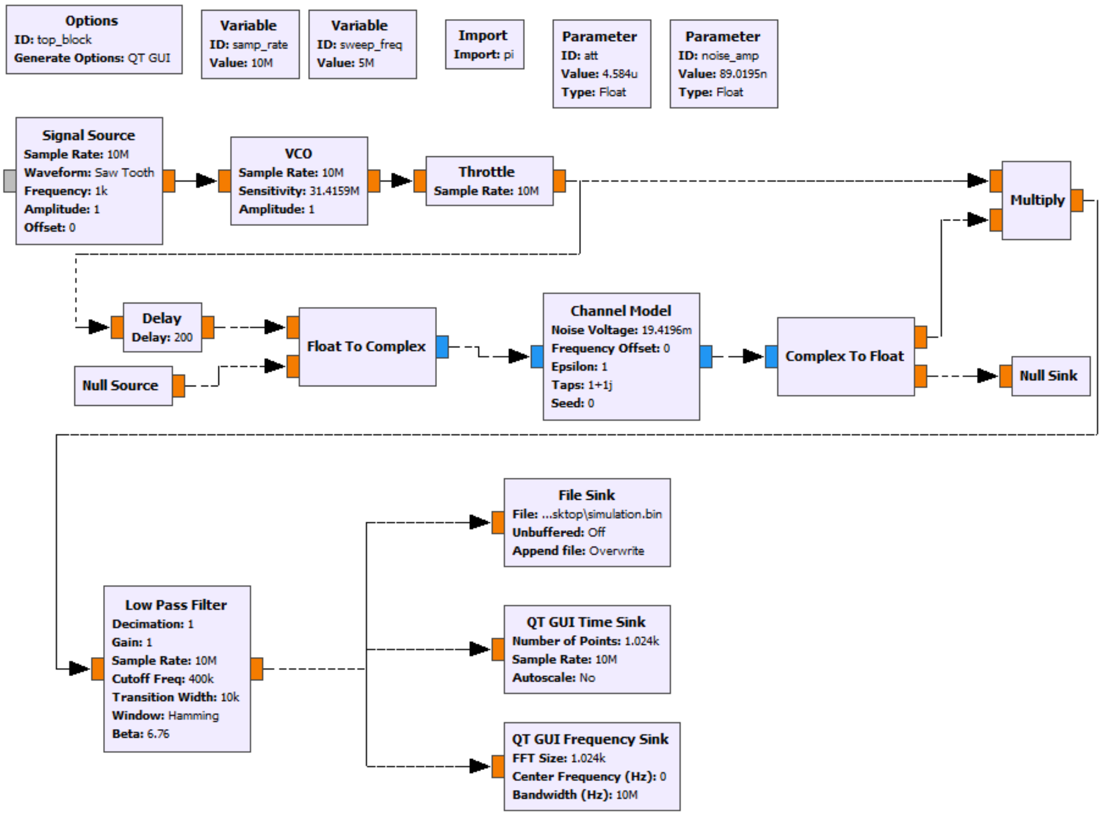


FIGURE 3.12: Flow graph of static target with the effects of noise and attenuation

The observation of Figure 3.13 shows that a noise floor appears inside the LPF band, when compared to the previous simulations.

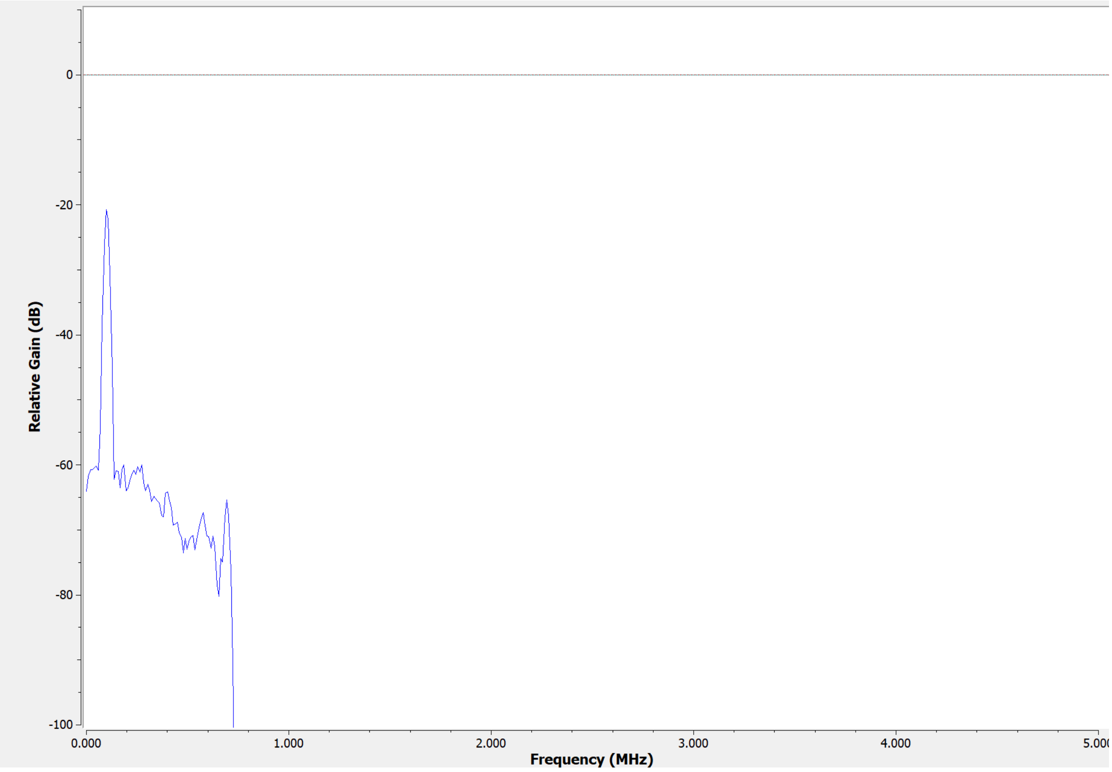


FIGURE 3.13: Effect of noise attenuation on the beat frequency

3.8 Conclusion of the simulations

The simulations that were performed in this chapter aimed to verify GNU Radio capabilities to be used together with an SDR Platform in order to develop an FMCW Radar capable of detecting single and moving targets. At first, simulations were made to verify the capability of detecting single and multiple static targets. Then, the capability to detect the movement of a target was tested. Finally, tests were made to verify the system capability to detect Doppler Shift Effect when a target is moving and to verify if the effect of noise and signal propagation still made it possible for the developed FMCW Radar to detect targets. The conclusion is that there is a potential for the developed software to be used together with an SDR Platform in order to develop an FMCW Radar capable of detecting single and moving targets.

Chapter 4

Experimental tests of the implemented FMCW Radar

This chapter presents the developed radar system and the system results for different SDR platforms, in order to compare the radar behaviour with different boards.

4.1 Hardware used

4.1.1 SDR Platforms

- **USRP N210**

  The USRP N210 is an hardware platform that is used with GNU Radio, to develop low cost software defined radios. The system consists of:

  -Two 14 bit, 100 MSample/s ADCs;

  -Two 16 bit, 400 MSample/s DACs;

  -A Xilinx Spartan 3-2000 FPGA;

  -A Gigabit Ethernet interface for the connection with a computer with GNU Radio;

-Two extension sockets for 1 or 2 daughterboards, which works as an RF front-end. In this dissertation, the CBX daughterboard was used. This board is full-duplex (allowing for the transmission and reception of signals at the same time) and covers a frequency range from 1.2 GHz to 6 GHz, with an instantaneous bandwidth of 40 MHz.

Figure 4.1 represents the USRP basic architecture.

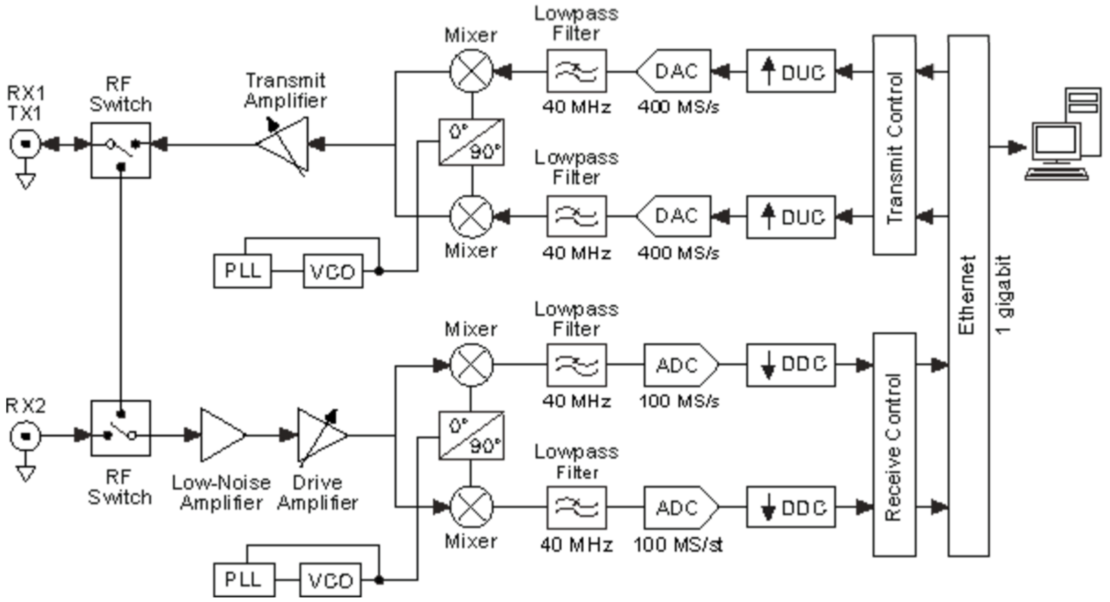


FIGURE 4.1: USRP basic architecture. Image taken from [22]

The main factors that limit USRP N210 performance are the processor speed, the host connection bandwidth, and the antennas used.

Processor speed

The processor speed of the host computer must be sufficient to maintain the throughput rate from the host computer to the USRP. If the processor capacity is not enough, it will result in an underflow, which means that the host computer is not producing data fast enough.

Host connection bandwidth

The host connection bandwidth refers to the connection between the USRP and the host computer. Even though the connection between the URSP

N210 and the host computer is Gigabit Ethernet, it still limits the maximum sample rate, the modulation bandwidth, and the samples per period.

Antennas

The antennas that are used will limit the performance of the system, since they work in a specific frequency band, which means the system will only work in that frequency band. There are antennas that offer wideband or multiband capacity, and that are more directional, however that will increase the cost of a system that aims to be low-cost.

Radar Transmit Power

When SDR is used to create a radar system, the transmit power affects the signal to noise ratio (SNR), which will influence the maximum detection Range. The transmit power value depends on the RF daughterboard model.

- **LimeSDR mini**

  LimeSDR mini is a low-cost software defined radio board that has as main features:

  - Intel's MAX 10 FPGA that has as main function to transfer digital data between the computer.

  - Lime Microsystems LMS7002M RF transceiver

  - USB 3.0 interface to connect the board with a computer.

  The LimeSDR-Mini development board block diagram is represented by Figure 4.2.

  This board has an RF frequency range from 10 MHz to 3.5 GHz and a maximum bandwidth of 30.72 MHz. The biggest advantage of this board when compared to others is its low price, coming in at 139 USD, when the USRP N210 comes in at 2200 USD.

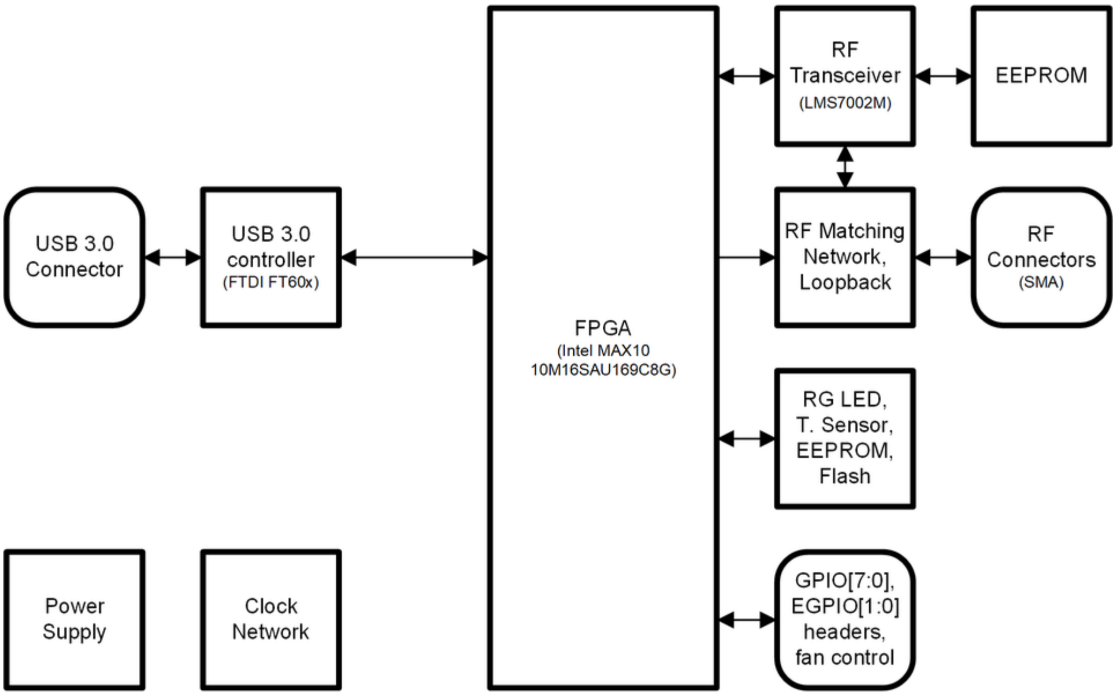


FIGURE 4.2: LimeSDR-Mini Development Board Block Diagram. Image taken from [23]

4.1.2 Antennas

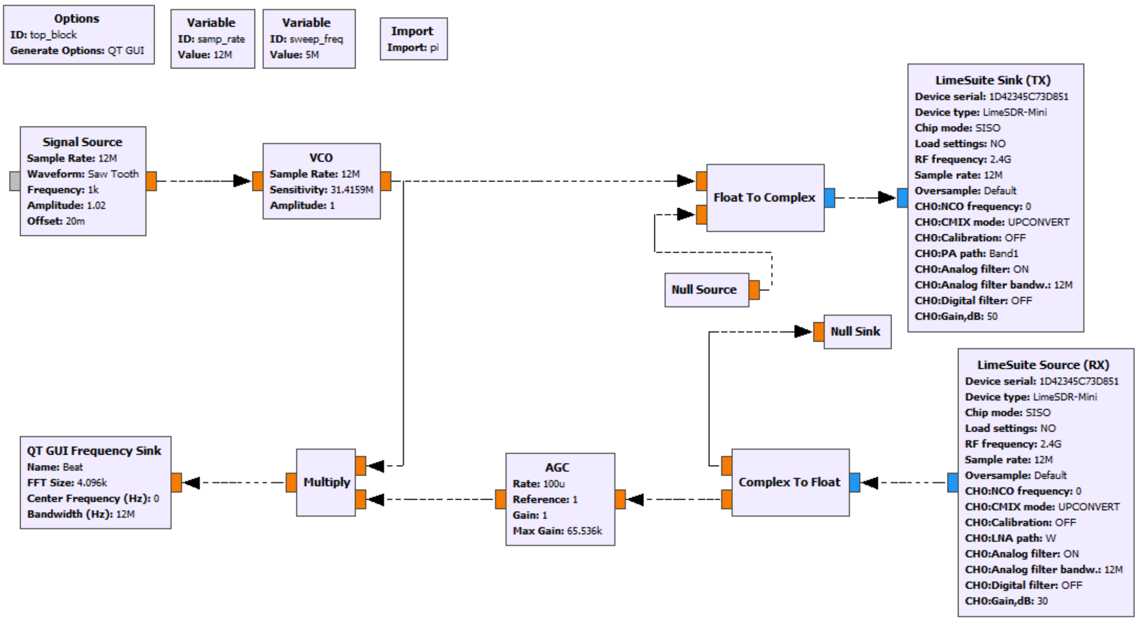
Besides the GNU Radio Software and the SDR Platform, the choice of antennas is also important. For this system the antennas used are omnidirectional antennas for 2.4 GHz band.

4.2 Experimental configuration and results

The GNU Radio software used for this system is adapted from the one that was developed on the previous chapter. The signal source and VCO generate the chirp signal. The signal is then streamed to the SDR board and radiated by the transmitted antenna. At the same time, the board receives the echo from targets captured by the RX antenna and streams the signal into GNU Radio. The RX signal passes on an Automatic Gain Control (AGC) block for maintaining the amplitude of the received signal higher. The TX and RX signal are then mixed to obtain the beat frequency. For LimeSDR mini, LimeSuite Sink and LimeSuite

Source blocks are used to stream the signal to the LimeSDR mini and to stream the signal to GNU Radio, respectively, as shown on Figure 4.3. For USRP, USRP Sink and USRP Source blocks are used to stream the signal to the USRP and to stream the signal to GNU Radio, respectively, as shown on Figure 4.4.

On these blocks, the most important settings to be configured are the center frequency, which corresponds to the RF frequency, which in this case is equal to 2.4GHz since it is the ideal frequency for the antennas, the sample rate, which is equal to 12 MS/s, and the gain value, which is equal to 50 dB (for TX) and 30 dB (for RX), since they increase the power of the signal.



Options
ID: top_block
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 12M

Variable
ID: sweep_freq
Value: 5M

Import
Import: pi

Signal Source
Sample Rate: 12M
Waveform: Saw Tooth
Frequency: 1k
Amplitude: 1.02
Offset: 20m

VCO
Sample Rate: 12M
Sensitivity: 31.4159M
Amplitude: 1

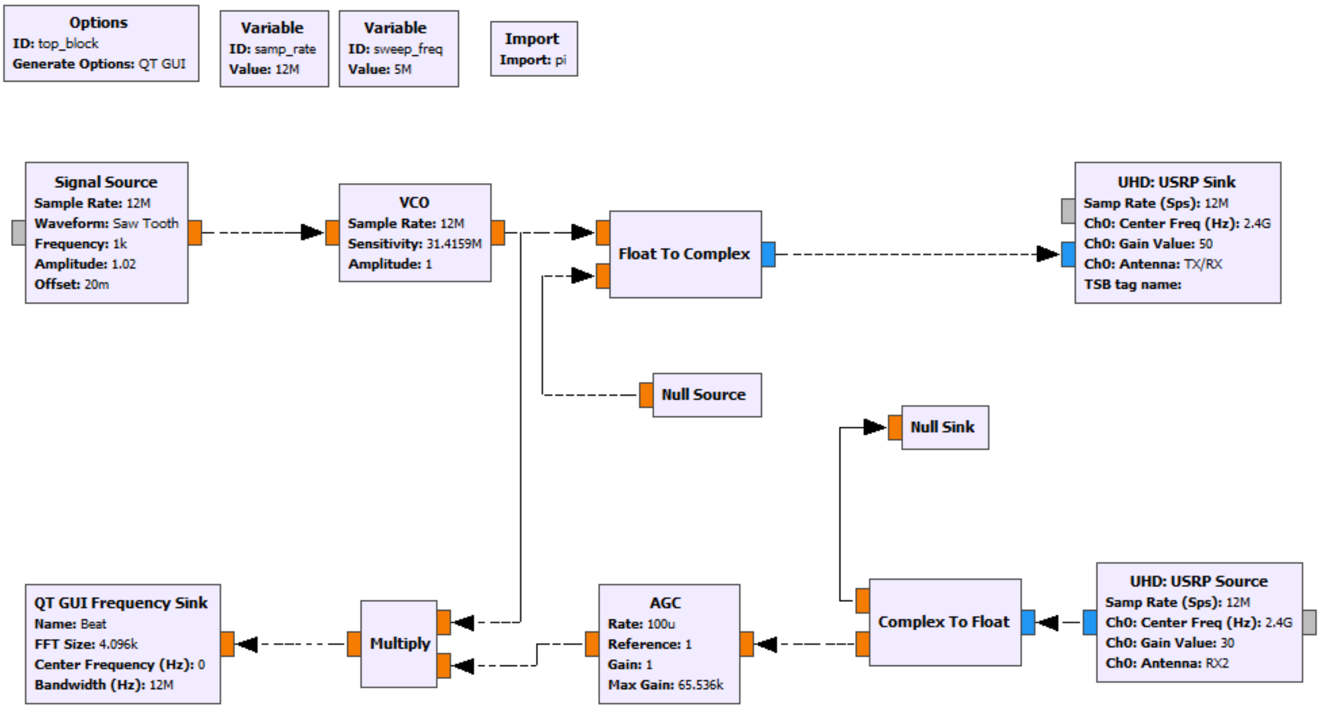
LimeSuite Sink (TX)
Device serial: 1D42345C73D851
Device type: LimeSDR-Mini
Chip mode: SISO
Load settings: NO
RF frequency: 2.4G
Sample rate: 12M
Oversample: Default
CH0:NCO frequency: 0
CH0:CMIX mode: UPCONVERT
CH0:Calibration: OFF
CH0:PA path: Band1
CH0:Analog filter: ON
CH0:Analog filter bandw.: 12M
CH0:Digital filter: OFF
CH0:Gain,dB: 50

LimeSuite Source (RX)
Device serial: 1D42345C73D851
Device type: LimeSDR-Mini
Chip mode: SISO
Load settings: NO
RF frequency: 2.4G
Sample rate: 12M
Oversample: Default
CH0:NCO frequency: 0
CH0:CMIX mode: UPCONVERT
CH0:Calibration: OFF
CH0:LNA path: W
CH0:Analog filter: ON
CH0:Analog filter bandw.: 12M
CH0:Digital filter: OFF
CH0:Gain,dB: 30

QT GUI Frequency Sink
Name: Beat
FFT Size: 4.096k
Center Frequency (Hz): 0
Bandwidth (Hz): 12M

AGC
Rate: 100u
Reference: 1
Gain: 1
Max Gain: 65.536k

FIGURE 4.3: FMCW Radar software for LimeSDR mini



Options
ID: top_block
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 12M

Variable
ID: sweep_freq
Value: 5M

Import
Import: pi

Signal Source
Sample Rate: 12M
Waveform: Saw Tooth
Frequency: 1k
Amplitude: 1.02
Offset: 20m

VCO
Sample Rate: 12M
Sensitivity: 31.4159M
Amplitude: 1

Float To Complex

UHD: USRP Sink
Samp Rate (Sps): 12M
Ch0: Center Freq (Hz): 2.4G
Ch0: Gain Value: 50
Ch0: Antenna: TX/RX
TSB tag name:

Null Source

Null Sink

QT GUI Frequency Sink
Name: Beat
FFT Size: 4.096k
Center Frequency (Hz): 0
Bandwidth (Hz): 12M

Multiply

AGC
Rate: 100u
Reference: 1
Gain: 1
Max Gain: 65.536k

Complex To Float

UHD: USRP Source
Samp Rate (Sps): 12M
Ch0: Center Freq (Hz): 2.4G
Ch0: Gain Value: 30
Ch0: Antenna: RX2

Figure 4.4: FMCW Radar software for USRP N210

4.2.1 Experimental Parameters

The following table presents all the important parameters used for this system. The sample rate value used is limited by the computer capacity, since a higher value would require a bigger computational load. The range resolution value is limited by the bandwidth value, which must comply with the Nyquist Theorem. In order to have a more effective radar system, this range resolution should be higher, however that would require a bigger computational load.

Table 4.1: Experimental Parameters Setting

Parameter	**Value**
Sample Rate	12MS/s
Waveform	Sawtooth
Chirp Period	1ms
Bandwidth	5MHz
Range Resolution	30m
Maximum Unambiguous Range	150km
Frequency	2.4GHz
Antennas Used	Omnidirectional in 2.4GHz

4.2.2 LimeSDR mini and USRP N210 results

To simplify the experimental tests, the antennas used on this system are omnidirectional and are side by side. For this reason, the transmitted signal should be instantaneously received by the reception part, which means that the transmited and received signals should be synchronized. However, due to delay caused by both the LimeSDR mini and USRP N210 components and the operating system the target delay is disturbed and the SDR becomes out of calibration, making real time target detection impossible. On figure 4.5 and figure 4.6, this effect can be seen. In an ideal system, the beat frequency value should be near 0 MHz, since the signals should be synchronized. However the delay caused by both boards hardware and by the operating system, stagger the signals, which increases the beat frequency value. This delay is difficult to predict, since its value is variable on the same run, which changes the beat frequency value. Figures 4.7 and 4.8 represent this variation of delay on the same run, with the beat frequency value changing when compared with figures 4.5 and 4.6.

A different configuration was also tested, reducing the bandwidth from 5 Mhz to 1 MHz, in order to verify how this delay varies when some condition changes. The beat frequency value shows that the delay also varied when compared to the previous configuration, as shown on figures 4.9 and 4.10.

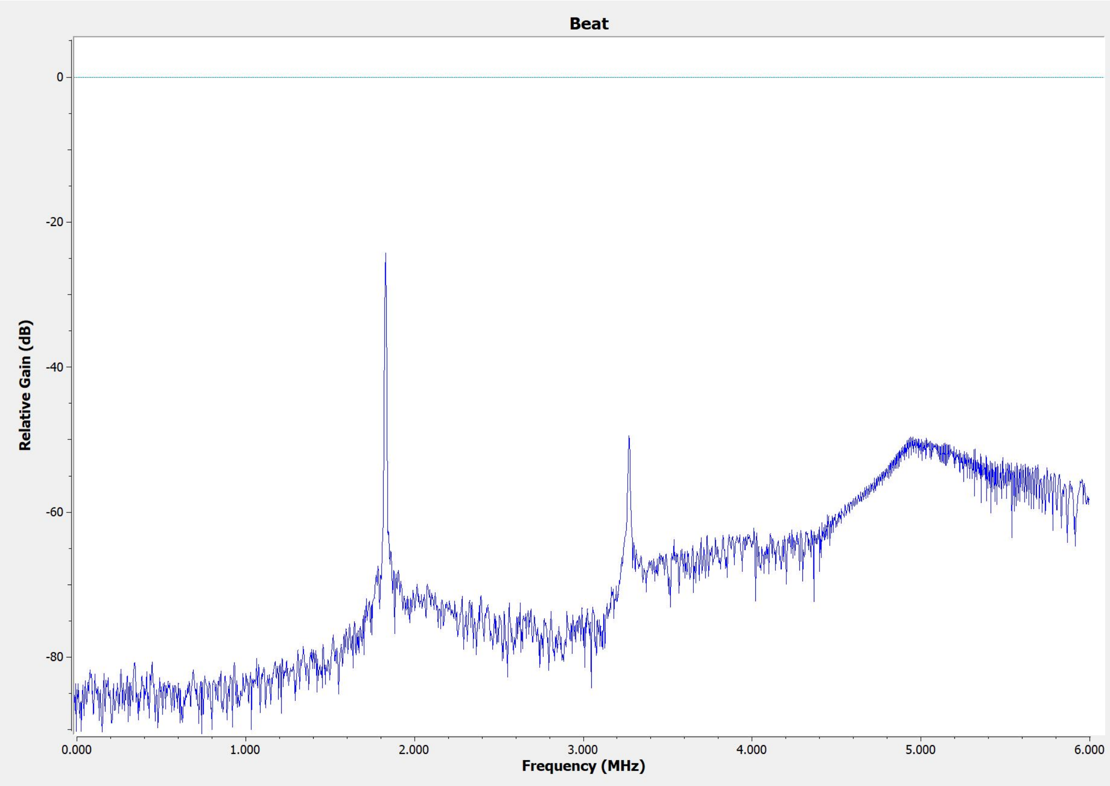


FIGURE 4.5: Beat Frequency results for LimeSDR mini

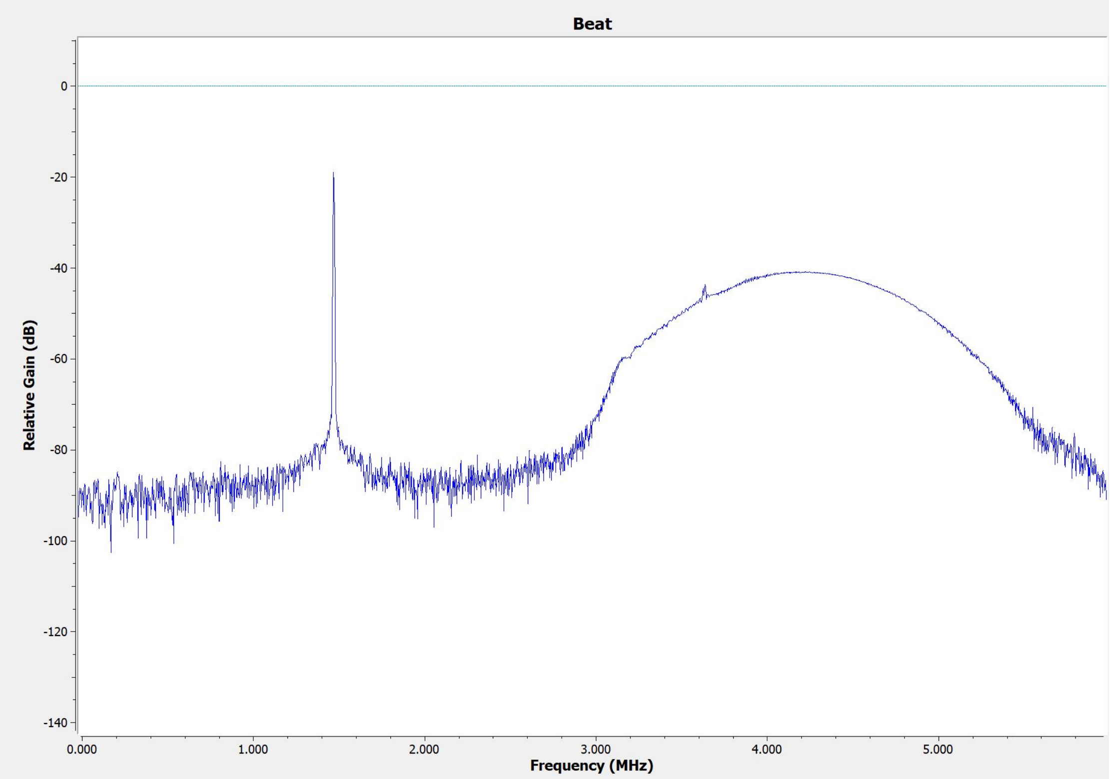


FIGURE 4.6: Beat Frequency results for USRP N210

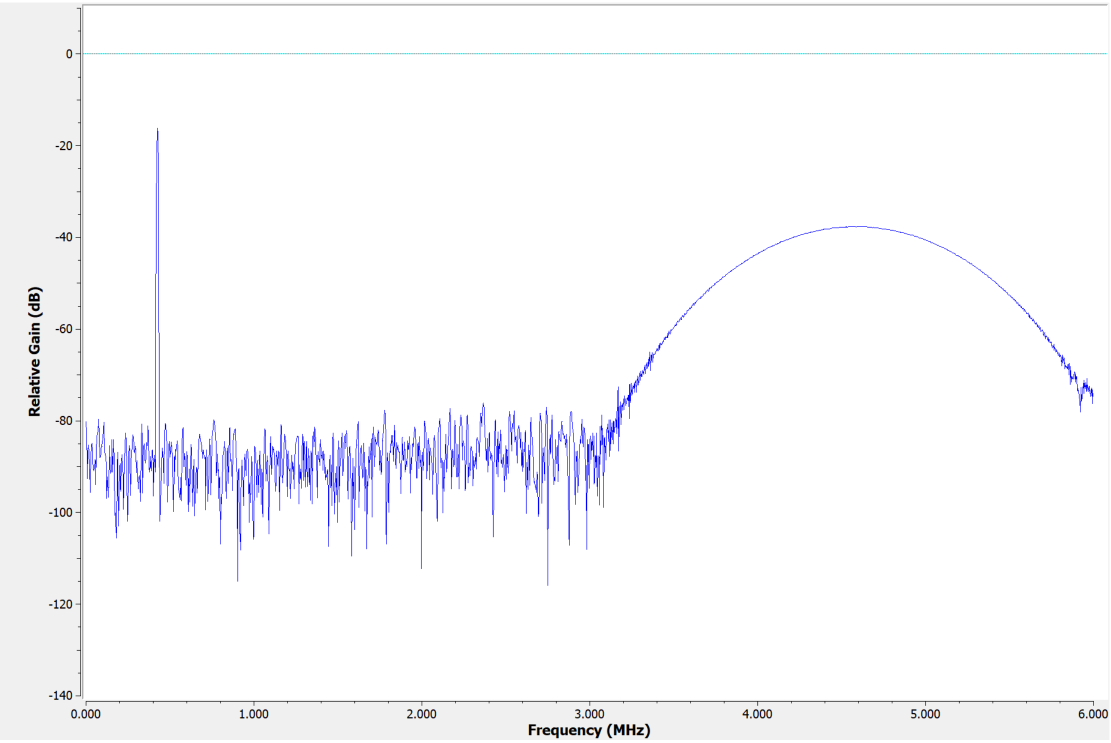


Figure 4.7: Beat Frequency variation due to change of delay value for LimeSDR mini

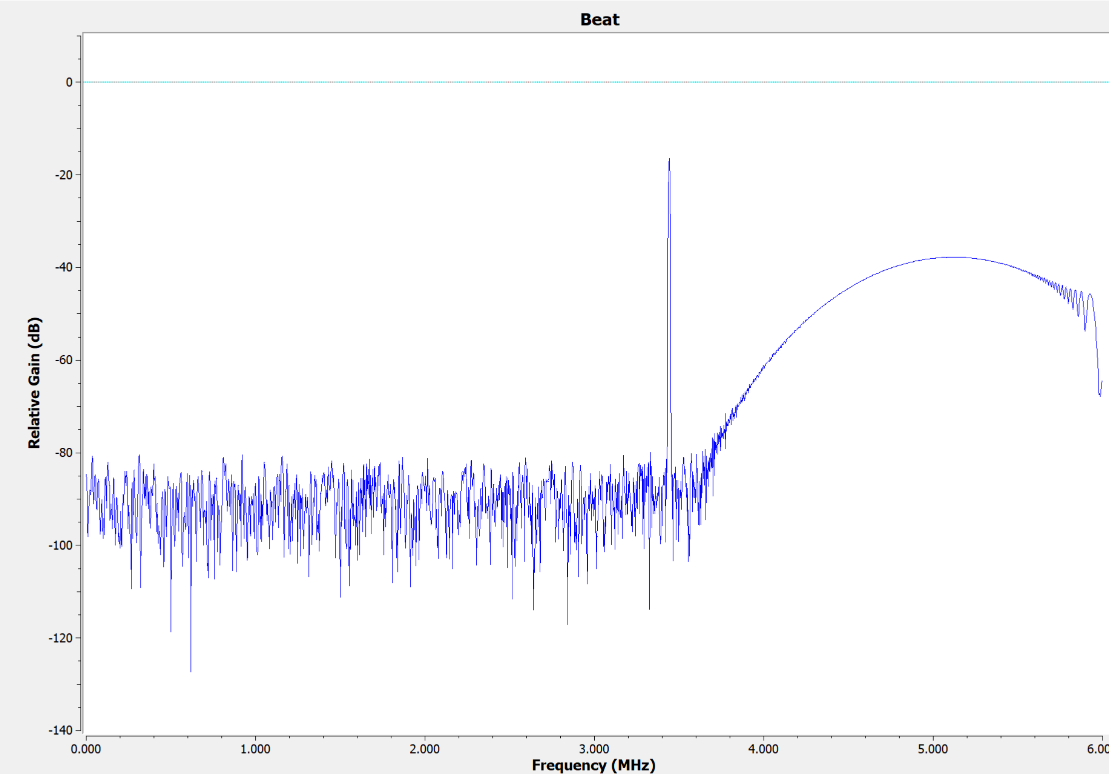


Figure 4.8: Beat Frequency variation due to change of delay value for USRP N210

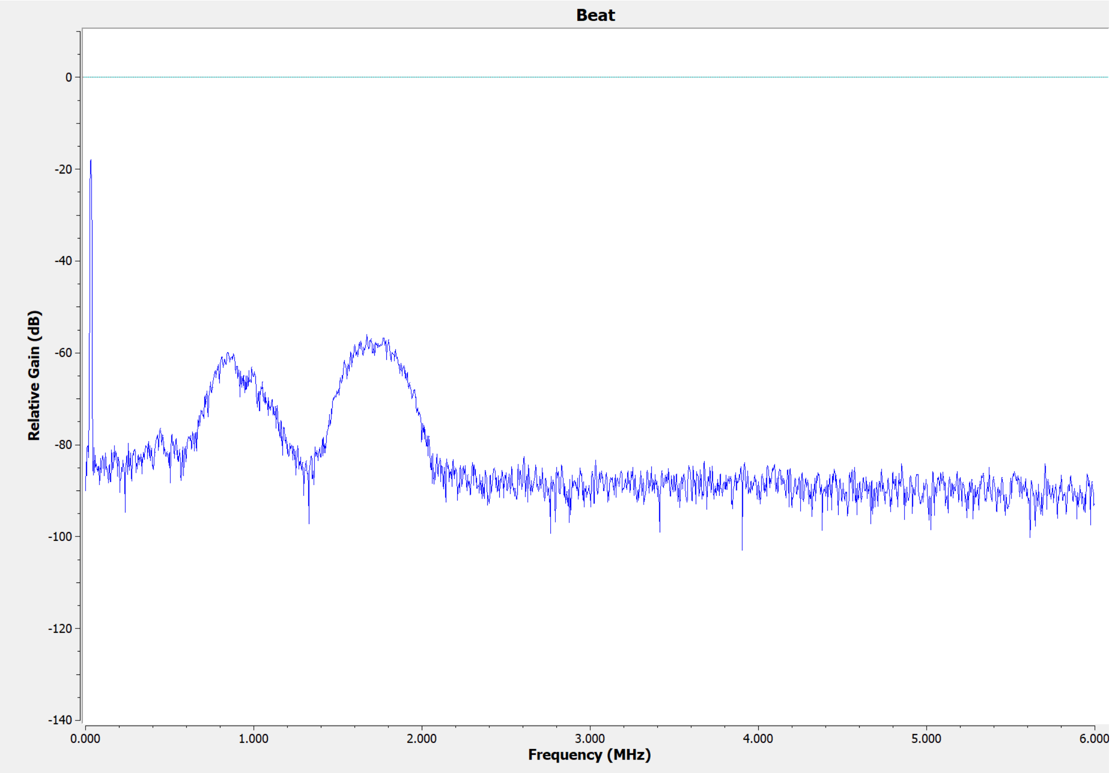


FIGURE 4.9: Beat Frequency results for LimeSDR mini for a bandwidth of 1 MHz

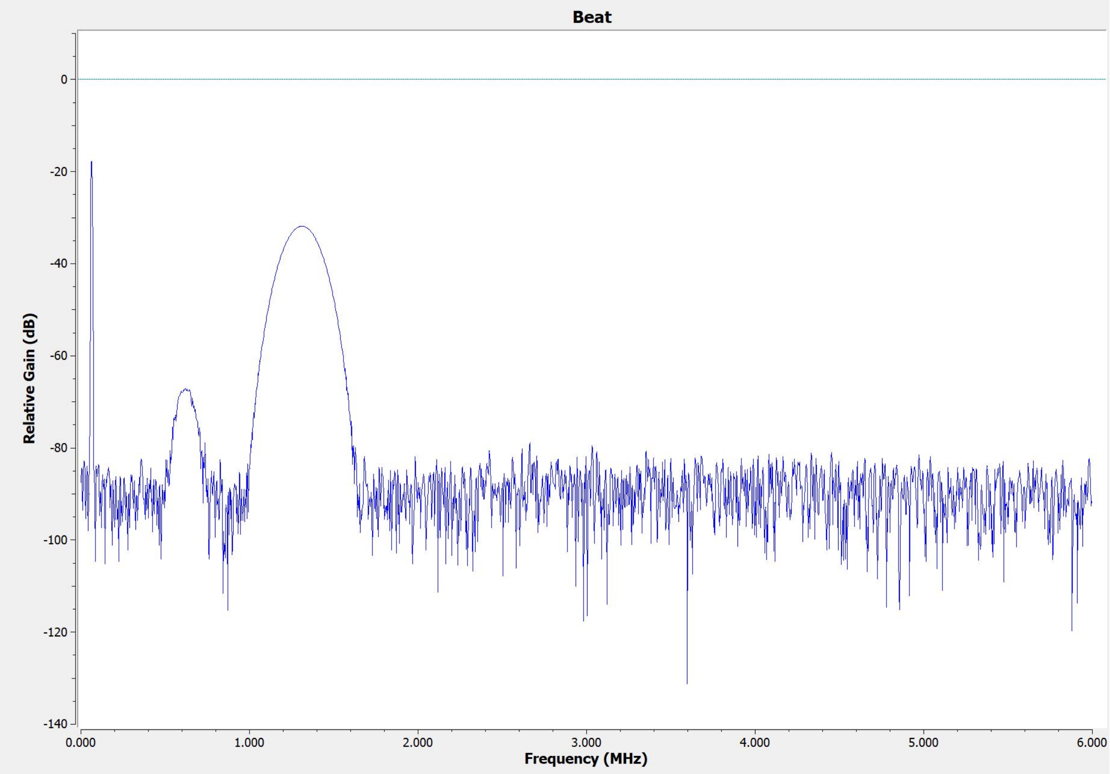


FIGURE 4.10: Beat Frequency results for USRP N210 for a bandwidth of 1 MHz

In order to solve this issue, a calibration can be made to remove this additional latency caused by the boards. On [24], an automatic calibration is proposed. About half second worth of samples from the beginning are dumped to remove the start-up noise from the SDR board. A set of samples equal to the size of one chirp period are then collected from the transmitter and receiver. By taking FFT, performing a complex conjugate multiplication and then inverse FFT, the index of the maximum point is found in the magnitude result. This index, read by the Function Probe block at every second, indicates the real-time additional delay between the TX and RX streams and drives the Variable Delay block in the receive chain. Thus, the TX and RX streams are always aligned by removing the additional delay frequently enough. However the objective of this thesis was to test the potential of both boards to obtain real time target detection in a direct way. When comparing to the simulation results, besides these effects, it is also more difficult to obtain the beat frequency value, due to the effects of noise and signal propagation.

Chapter 5

Conclusion and Future work

5.1 Conclusion

This dissertation aimed to test the ability to use Software Defined Radio platforms in order to develop a Frequency Modulation Continuous (FMCW) Radar, with reduced cost and being able to perform the same functions of a regular FMCW radar. In order to test those capabilities, the software part of the system, developed with GNU Radio, was tested through a series of simulations. The developed simulations tested the ability to perform single static target detection, multiple static targets detection, to detect a slowly moving target, to study the Doppler shift effect and the last simulation aimed to verify if attenuation due to propagation through the channel and corruption effects due to noise still made it possible for the developed FMCW Radar to detect targets. These simulations proved that the developed software is capable of being used as part of an SDR based Frequency Modulation Continuous (FMCW) Radar. Then, the software was integrated with the LimeSDR and with USRP N210, and with two antennas, to create the FMCW Radar system. The results showed that this system has a potential of performing target detection. However in order for this FMCW Radar to be capable of performing target detection, a solution for the time delay between transmitter and receiver that is caused by both boards main components and by the operating system must be found. That solution might be a calibration on the system that removes the delay.

5.2 Future work

The main issue of this FMCW Radar is the non deterministic delay between the transmitter and receiver chains, that is caused by both boards main components and by the operating system. So, in the future there must be a solution for this issue, since this delay will make target detection impossible. Another major difficulty in the development of the system was the fact that the antennas that were used, were not ideal for a radar. The antennas used on this dissertation are omnidirectional antennas, which limits the capability of this system to perform target detection. For future work, the antennas must be more directional, so that the system is capable of detecting real targets on real time. Another limitation of this system is its range resolution. The range resolution is limited by the bandwidth value, and for this case, since the bandwidth value is 5 MHz, the range resolution is 30 meters, too small when we want to detect closer targets. In order to have a bigger bandwidth value, the sample rate should also have a bigger value. However, a bigger value demands a bigger processing capacity of the host PC used to run the software part of the system, which was not the case. So, for future work, a host PC with bigger processing capacity, should also be used. Also on future work, there should be an automatic way to obtain the range of the detected target, since right now the beat frequency value is being observed on real time, and the range that corresponds to that beat frequency is calculated offline. In the future it should be possible to observe, right at the moment the distance of the target. Another improvement that could be made on this system is adding the capacity of detecting targets that are at different directions. After these issues and limitations are solved, this system could be used together with a spoofing system and a jammer in order to, not only detect the presence of the target, but also prevent them from accessing certain regions.

Bibliography

[1] "Have a look at 10 major application areas of drones." `https://www.geospatialworld.net/blogs/10-major-application-areas-of-drone/`, June 2018.

[2] "Commercial drones are here: The future of unmanned aerial systems." `https://www.mckinsey.com/industries/capital-projects-and-infrastructure/our-insights/commercial-drones-are-here-the-future-of-unmanned-aerial-systems`.

[3] "Drone causes gatwick airport disruption." `https://www.bbc.com/news/uk-40476264`, Jul 2017.

[4] "Portugal: Há novas regras para a utilização de drones." `https://pplware.sapo.pt/informacao/portugal-ha-novas-regras-para-drones/.`, Jan 2019.

[5] E. Vattapparamban, İ. Güvenç, A. İ. Yurekli, K. Akkaya, and S. Uluağaç, "Drones for smart cities: Issues in cybersecurity, privacy, and public safety," in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 216–221, IEEE, 2016.

[6] A. Prabaswara, A. Munir, and A. B. Suksmono, "Gnu radio based software-defined fmcw radar for weather surveillance application," in *2011 6th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*, pp. 227–230, IEEE, 2011.

[7] S. Sundaresan, C. Anjana, T. Zacharia, and R. Gandhiraj, "Real time implementation of fmcw radar for target detection using gnu radio and usrp," in *2015 International Conference on Communications and Signal Processing (ICCSP)*, pp. 1530–1534, IEEE, 2015.

[8] Tutorialspoint.com, "Radar systems overview." `https://www.tutorialspoint.com/radar_systems/radar_systems_overview`, journal=www.tutorialspoint.com.

[9] M. I. Skolnik, *Introduction to radar systems.* McGraw-Hill, 1980.

[10] M. Jankiraman, *Design of multi-frequency CW radars.* SciTech Publishing, Inc., 2007.

[11] C. Wolff, "Radar basics." `http://www.radartutorial.eu/01.basics/Range%20Resolution.en.html`.

[12] J. Zhu, "Low-cost, software defined fmcw radar for observations of drones," 2017.

[13] "Rf tutorial lesson 17: Simulating a frequency-modulated continuous-wave (fmcw) radar system." `http://www.emagtech.com/wiki/index.php/RF_Tutorial_Lesson_17:_Simulating_a_Frequency-Modulated_Continuous-Wave_(FMCW)_Radar_System`.

[14] `https://www.mathworks.com/help/phased/examples/automotive-adaptive-cruise-control-using-fmcw-technology.html`.

[15] M. M. Hill, *Developing a generic software-defined radar transmitter using GNU Radio.* PhD thesis, 2012.

[16] D. Sinha, A. K. Verma, and S. Kumar, "Software defined radio: Operation, challenges and possible solutions," in *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, pp. 1–5, IEEE, 2016.

[17] C. de la Morena-Álvarez-Palencia and M. Burgos-Garcia, "Experimental performance comparison of six-port and conventional zero-if/low-if receivers

for software defined radio," *Progress In Electromagnetics Research*, vol. 42, pp. 311–333, 2012.

[18] J. Strange and D. Grant, "Direct conversion: No pain, no gain." `https://www.eetimes.com/document.asp?doc_id=1205337`, Apr 2002.

[19] J. P. Oliveira, J. B. N. Leite, F. C. B. F. Müller, and A. B. da Rocha Klautau Jr, "Uso de software livre no ensino de telecomunicações: Estudo de caso com gnuradio e usrp,"

[20] T. Debatty, "Software defined radar a state of the art," in *2010 2nd International Workshop on Cognitive Information Processing*, pp. 253–257, IEEE, 2010.

[21] T. Ajitha, E. Joy, A. AnishJoyce, and R. Gandhiraj, "Radiation pattern measurement of log-periodic antenna on gnu radio platform," in *2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE)*, pp. 1–4, IEEE, 2014.

[22] `https://kb.ettus.com/About_USRP_Bandwidths_and_Sampling_Rates`.

[23] "Limesdr-mini v1.1 hardware description." `https://wiki.myriadrf.org/LimeSDR-Mini_v1.1_hardware_description`.

[24] B. Liu and R. Chen, "Software-defined radar and waveforms for studying micro-doppler signatures," in *Radar Sensor Technology XVIII*, vol. 9077, p. 907718, International Society for Optics and Photonics, 2014.