University of Tartu
Faculty of Science and Technology
Institute of Technology

Rein Jaks

UAV Direction Finding Using Phase Difference

Master's thesis

Supervisor
MSc Jaanus Kalde

Tartu 2020

Resümee/Abstract

Mehitamata õhusõiduki suuna määramine faasivahe meetodit kasutades

Mehitamata õhusõidukid on viimase aastakümne jooksul muutunud järjest tavapärasevamaks nähtuseks. Kuigi neid kasutatakse nii meelelahutuseks kui fotograafiaks, on mehitamata õhusõidukid suureks ohuks julgeolekule. Selle kaitseks on võimalik avastada raadio suunamäärajaga raadiosaatjaga varustatud õhusõidukeid. Kuigi raadio suunamäärajaid on juba olemas, on need sageli rasked, kallid ja vajavad keerukaid riistvaralisi lisasid. Antud omaduste vältimiseks koostatakse töös automaatselt signaali tuvastav raadio suunamääraja Adcock meetodil, kasutades odava klassi tarkvararaadiot ja kahte monopol antenni.

CERCS: T125 Automatiseerimine, robootika, juhtimistehnika, P140 Jadad, Fourier analüüs, funktsionaalanalüüs, T121 Signallitöötlus, T330 Sõjandus ja militaartehnoloogia

Märksõnad: Mehitamata Õhusõiduk, Tarkvararaadio, Suuna Määramine, Faasivahe, Raadiolaine, Signaali Tuvastus, Adcock

UAV Direction Finding Using Phase Difference

UAVs have become more common over the course of last decade. While these can be used for entertainment purposes and capture stunning photography these can cause great threat to national security when deployed in the wrong location. Radio Direction finder can be used for detecting and finding an UAV with a transmitter on board. While there are multiple direction finders available, these are heavy, expensive and need complicated technology to work. For this thesis using an inexpensive SDR and two monopole antennas a direction finder with automatic signal detection is constructed implementing Adcock phase difference method.

CERCS: T125 Automation, robotics, control engineering, P140 Series, Fourier analysis, functional analysis, T121 Signal Processing, T330 Military science and technology

Keywords: UAV, Software Defined Radio, SDR, Direction finding, Phase Difference, Radio Wave, Signal Detection, Adcock

Table of contents

Resümee/Abstract 1

Table of contents 2

List of Figures 4

List of Used Abbreviations and Constants 5

List of Tables 7

1. Introduction 8

1.1 Problem Statement 8

1.2 Objectives 8

2. State of the Art 9

2.1 Direction finding methods 9

2.2.1 Location finding 9

2.2.2 Direction Finding Algorithms 10

2.2 Software Defined Radios 13

2.3 Currently available technologies 14

3. Methodology 15

3.1 Radio wave theory 15

3.1.1 Phase measurement 15

3.1.2 Distance 17

3.2 Hardware 18

3.2.1 Software Defined Radio 18

3.2.2 Antennas 19

3.2.3 Testing Devices 20

3.3 Development 20

3.3.1 Phase measurement 21

3.3.2 Signal detection 24

3.3.3 Analysis and visualisation....26
4. Testing....29
4.1 Minimum pulse duration....29
4.2 Maximum range....31
4.3 Stability and Accuracy....31
5. Limitations and Further Improvements....36
5.1 Limitations....36
5.2 Improvements....36
6. Conclusions....37
Acknowledgement....38
Bibliography....39
Appendix I – Flow graph of the GNU Radio program....42
Appendix II – Codes for custom blocks....43
Signal converter....43
Pin to frequency....43
If threshold....44
Appendix III – Signal visualisation....45
Appendix IV – Error Handling....47
Non-exclusive licence to reproduce thesis and make thesis public....49

List of Figures

Figure 1: Difference of trilateration and triangulation [6] 10
Figure 2: Illustration of the Adcock method using cathode array tubes [7] 11
Figure 3: VDF antenna in EETU (Tartu Airport) 12
Figure 4: Block diagram of an SDR [11] 13
Figure 5: Signal with 45-degree phase difference simulated in MATLAB 16
Figure 6: Testing objects - from the left video transmitter (1), two-way radio (2), remote (3). 20
Figure 7: Simplified flow graph of the software. Graph made in code2flow environment. 21
Figure 8: Phase difference calculation in GNU Radio 22
Figure 9: Phase signals (blue phase 1, red phase 2 and green is subtraction) in time domain with delay of 60 samples 22
Figure 10: Phase subtraction process with two custom blocks 23
Figure 11: Phase signals (blue phase 1, red phase 2 and green is subtraction) in time domain with delay of 60 samples with added custom if block in GNU Radio QTGUI. 24
Figure 12: Signal detection part in GNU Radio 25
Figure 13: Visualisation of an incoming signal with 109-degree azimuth towards the receiver 27
Figure 14: Phase measurements over iterations 27
Figure 15: Measurements in the former Soviet Raadi military airfield, picture taken from the Republic of Estonia Land Boards Geoportals orthophoto. 29
Figure 16: Signal hopping measurements for the Eachine i6 remote 30
Figure 17: Measurement methodology for stability and accuracy 31
Figure 18: Indoor measurements on 0 and 90 degrees with and without error handling 32
Figure 19: Field test with 0, 45, 90- and 135-degree azimuth angles 34

List of Used Abbreviations and Constants

c	Speed of Light $3*10^8$ m/s
dB	Decibel
dBm	Decibel-milliwatts
dBW	Decibel-watts
EETU	Tartu Airport
FFT	Fast Fourier Transform
FSPL	Free Space Path Loss
GHz	Gigahertz
GRCon17	GNU Radio Conference 2017
k	Boltzmann constant $1.380649*10^{-23}$ J/K
MHz	Megahertz
MLAT	Multilateration
NDB	Non-directional Beacon
PWM	Pulse Width Modulation
SAR	Search And Rescue
SDR	Software Defined Radio
SIGINT	Signals Intelligence
SR	Software Radio
UAVs	Unmanned Aerial Vehicles
UDP	User Datagram Protocol

USRP	Universal Software Radio Peripheral
VDF	VHF Direction Finder
VHF	Very High Frequency
VOR	VHF Omnidirectional Range

List of Tables

Table 1: Results from the testing 35

1. Introduction

During the last decade UAVs (Unmanned Aerial Vehicles) have become more common among hobbyists, photographers, and other enthusiasts. Higher end models can often be equipped with professional cameras and other specialist equipment can be installed on board, which makes them useful for SAR (Search And Rescue) missions, different professional uses or even in military surveillance [1]. Since it can be a threat on national security, counter measures must be available. One of them can be finding the direction from the observer.

1.1 Problem Statement

This problem was brought up by a member of the Estonian Defence League's Cyber Defence Unit. The problem was that existing direction finders for determining UAVs large, expensive, and hard to carry for patrol units without vehicles.

A solution using SDR (Software Defined Radio) and open source programming language was requested. Also, it needed to be built with a possibility to be further developed by the Cyber Defence Unit members.

For this thesis, an SDR based direction finding solution using phase difference method was created. It is able to find a UAVs video transmitter in a 180-degree range using two antennas, an SDR and a computer.

1.2 Objectives

The main objective for this thesis is to construct 180-degree direction finder with SDR by implementing phase difference method, also known as Adcock method, with automatic signal detection. Next, suggestions are created for a 360-degree direction finder, which could be used for location finding. The device should cover most of the 83,5MHz bandwidth in the frequency range of 2400-2483,5MHz. This is due to the regulations of radio frequency plan, which states these frequencies for usage for short-range devices. [2]

2. State of the Art

2.1 Direction finding methods

Different direction-finding methods are being used as early as beginning of 20th century. Currently the technology has had advancements, but the main idea is still same. This chapter describes methods used previously, which are currently available, and technology used in current thesis.

2.2.1 Location finding

There are multiple uses for radio direction finding – these can be navigational aids, SAR (Search And Rescue), data collection, or even radio intelligence. One of the first uses for direction finding technology was marking the air passages for aircraft with the NDB (Non-directional Beacon), which transmitted a Morse code on a certain frequency without any directional properties in very low frequency range.

The pilot marked down the frequencies and the Morse codes in pre-flight briefing and flew in the direction of the frequency. This method in aviation was later replaced by VOR (VHF (Very High Frequency) Omnidirectional Range) and with GPS in the newer aircraft.

In the radio signal intelligence field (later known as SIGINT – Signals Intelligence) the need for searching for a location of the incoming signal was introduced during the First World War, where knowing the location of enemy vessels gave an advantage. During the second World War the usage of radio direction finding was more common, as German submarines used to have direction finders on board to locate British submarines by their radar. [3] In modern warfare, the Early Warning Threat detection is the most common use for passive direction finders. Also targeting, homing, and jamming are common uses for this technology. [3]

Since most of the UAVs in use today have an on-board transmitter, which either transmits video feed or other information to the pilot the UAVs can be interpreted as transmitters.

There are multiple ways to discover a transmitter location. With multiple antennas the most used system is trilateration, illustrated on the left of Figure 1. It measures the time and calculates the time difference, which the signal spends to reach three receivers. The distance calculated using this method is called pseudo range. If pseudo range is acquired the location can be calculated by observing at the common part of the areas [4]. In Estonian aviation this method is still in use under the name of Multilateration (MLAT), which uses four or more antennas in a field to determine the exact point in three-dimensional space for an aircraft. [5]

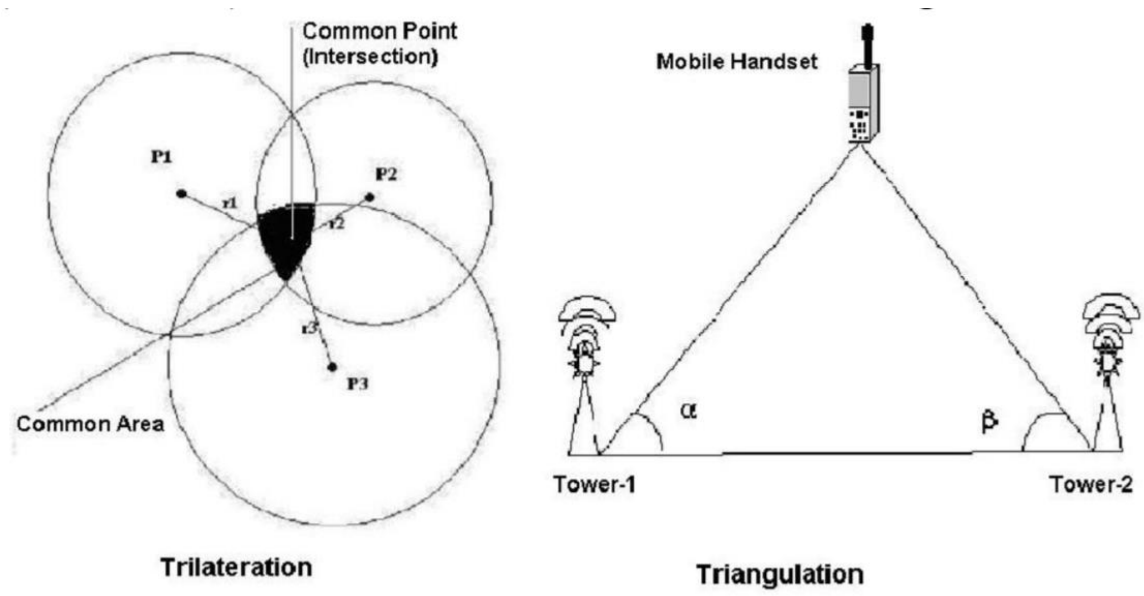


Figure 1: Difference of trilateration and triangulation [6]

Similarly, to trilateration, triangulation uses multiple antennas, from which each determines the direction of the signal, with difference seen on Figure 1. Each direction creates a mathematical line on the map, and the theoretical location of the device is where the lines meet. [4]

However, finding the exact location of the transmitter requires a lot of preparation, sophisticated technology and sometimes only the direction of the transmitter is needed. This can be done with one or multiple receivers in one location, which can reduce the preparation time and complexity of the system.

2.2.2 Direction Finding Algorithms

The direction of a radio signal can be found either by measuring the amplitude, phase, or both. The simplest way for measuring the direction with passive single channel receiver is to use a rotating antenna, more known as radar systems. [3, p. 12] With this method a highly directional antenna can be used. This way the antenna is physically rotating around its axis and listening to incoming signals. When a signal is detected, the direction of the antenna is registered. If the signal strength is also registered, the approximate range can be calculated [3, pp. 12-13]. This method, however, requires accurate mechanical moving parts and its direction-finding capabilities are dependent on the speed of the rotation.

One of the more complicated, but widely used, methods is using the Adcock/Watson-Watt method. With this method two dipole antenna pairs, located in a 90-degree shift from the centre, are placed as an array. While the Adcock method uses at least four antennas in a circular pattern the Watson-Watt method adds one antenna to the centre point for reference to decrease ambiguity caused by cathode ray tubes, which were used when this method was created [7]. When signal is intercepted the phase difference between each pair is registered and from there the direction of the incoming signal is calculated [8] [9]. Although the original method has four antenna array (shown on Figure 2) there are multiple different variations available. The Adcock method is also used as the main method for the current thesis.

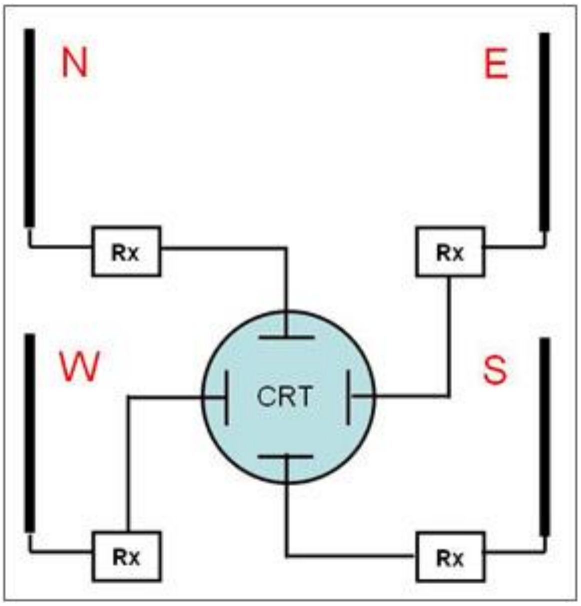


Figure 2: Illustration of the Adcock method using cathode array tubes [7]

The main concern in the Adcock method is the reliability of the measurements and the distance between antennas. It has been calculated that the error can reach up to 7° if the antennas are not in the perfect distance and approximations are used. Furthermore, the antennas must be moved when the frequency changes. [9] In the modern systems this can be solved with post-processing of the information. This method is still used in aviation in VDF (VHF (Very High Frequency) Direction Finder) devices, also seen on Figure 3.

Figure 3: VDF antenna in EETU (Tartu Airport)

Pseudo-doppler direction finding method is another widely used direction finding methods. Initially it consisted of an antenna rotating with constant angular velocity and one stationary antenna, usually in the middle. With these antennas the doppler shift between the rotating and stationary antenna is measured and direction of the transmitter determined. Similarly, to the Adcock method pseudo doppler has downsides: antenna misplacement, approximation and interaction errors can all produce significant faults of the measurement. [9]

The correlative interferometer is a more advanced direction-finding solution than the previous systems as it needs digital systems to work. It works with the Adcock method and uses previously known values from a known direction-finding system. Then it calculates the quadric error, also known as correlation coefficient, by comparing it with the measured phase difference to previously determined values. By observing the maximum coefficient, the angle of the signal can be omitted. [10]

In conclusion the direction finders can be simply one of the mentioned or they can be a mixture of different technologies. However, the issues with all the direction finders are similar – reflections, interferences, geometrical, delay, hardware, and modulation. [9]

Reflections are caused by reflecting surfaces around the measurement area. This can cause false readings as well as signal interferences, which can make the reading of the signal parameters difficult. Geometrical errors, such as inexact instalment of antennas, will also produce errors in signal reading as well as wrong cable management, which can cause signal interferences, delay, and polarisation of the signal, when not correctly shielded or placed. Amplitude modulation introduced magnitude mismatches in the measurement systems. However, using multiple receivers, the errors caused by amplitude-, frequency- and phase modulation can be avoided. [9]

2.2 Software Defined Radios

In this thesis SDRs are being described and used. Since these devices are more specialist orientated for a narrow field of users it can be hard to understand the importance of this technology.

The first software radios were used already during the 80s for military applications. For the wider audience, a software radio was introduced in 1995 in IEEE Communications Magazine. SDR can be multiband, supporting multiple standards or services or have multiple channels or all of them at once.

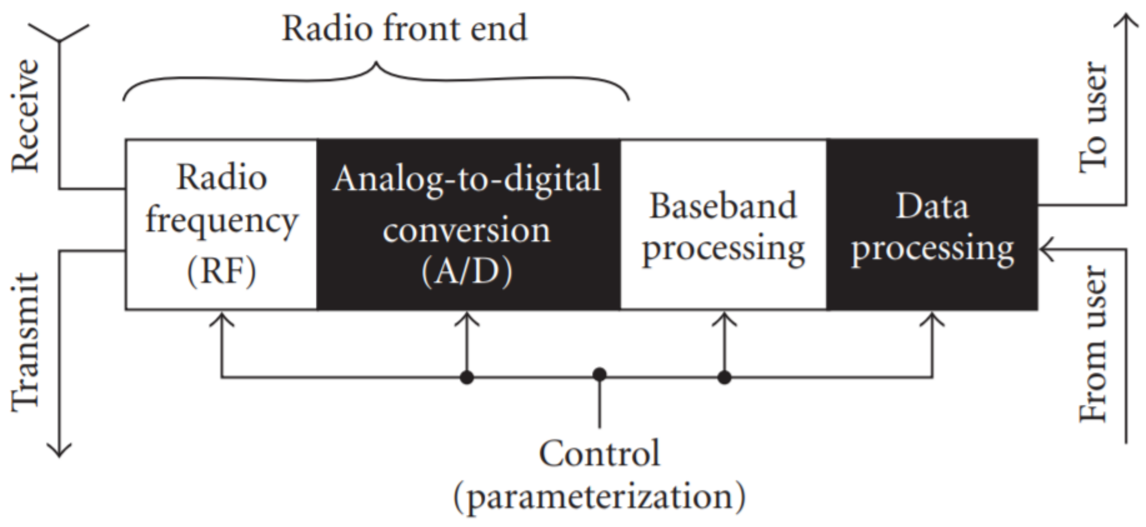


Figure 4: Block diagram of an SDR [11]

Figure 4 shows the block diagram of an SDR. There has been uses of SR (Software Radio) with a difference of additional control bus on SDR, which can be used to reconfigure parameters instantly [11]. If compared to a regular radio, where time consuming hardware adjustments would be needed to change the parameters, using an SDR gives an advantage on cost effectiveness and flexibility. For example, instead of single use items built during prototyping SDRs can be used for multiple purposes [12].

This technology has risen lately since chips used in SDRs are often also used in everyday radio communications devices – starting from chips in the TV remote up to mobile devices and specially fabricated chips for this use. In correlation with advancements in the mobile communications field SDRs have become more capable of different tasks.

2.3 Currently available technologies

The most similar professional product to the current thesis project is Rode & Schwarz’s DD007 direction finder, which is portable, and the I/Q data can be externally analysed. This device works in the range of 20MHz to 6GHz with optional extras and the display bandwidth is 10MHz. To capture the bursting signal, it needs to last for 10ms. [13]

KerberosSDR is a 4-channel coherent SDR build specifically for direction finding. It has open source programming and is based on the most common R820T2 and RTL2832U chips. There is an open source demo software available for Linux, which can be run on most native Linux devices. However, since the frequency range is 24MHz to 1,7GHz it is not usable on the current problem. [14]

There have been also some methods using older RTL-SDR models. One of them was presented on GRCon17 (GNU Radio Conference 2017) by Sam Whiting and Todd Moon. This method uses two coherent RTL-SDR models, which needed cross correlation to synchronise these devices. It could locate a handheld transmitter and worked on Android phone. [15]

3. Methodology

Most UAVs that are in use today carry a transmitter on board, which transmits video or other data to the pilot. To determine the direction of the drone in relation to the user this radio wave can be observed. To understand this method radio wave theoretical background is introduced as well as choices made for hardware and development process.

3.1 Radio wave theory

3.1.1 Phase measurement

The basis of this thesis is to observe and measure the radio wave of the transmitter. Radio wave is a type of electromagnetic radiation, which is usually modulated to carry information, usually wirelessly, over a distance. When measuring the radio wave, the frequency is most often observed. From the frequency the wavelength can be calculated using the following formula:

(1) $$\lambda = \frac{c}{f}$$ [16]

, where λ is the wavelength in meters,
c is the speed of light in the current enviroment,
f is the frequency in *Hz*.

To simulate a radio wave the unmodulated carrier wave is presented as a mathematical sine wave. The formula for the sine wave is the following:

(2) $$y(t) = Asin(2\pi ft + \varphi)$$

, where y is the value at the given time,
A is amplitude,
t is time,
f is frequency in *Hz*,
φ is phase.

However since in signal processing complex I/Q data is being used, where I represents the real part of the complex signal and Q represents imaginary part, then the phase can be calculated using only real part [17]. It can be done using the following method:

(3) $$y(t) = Asin(2\pi ft + \varphi) = Icos(2\pi ft) - Qsin(2\pi ft)$$ [17]

, where

(4) $$I = Acos(\varphi)$$ [17]

(5) $$Q = \mathrm{Asin}(\varphi)$$ [17]

If the amplitude is assumed to be 1, since it is no relevant for current applications, then the phase difference between two signals can be using only the real parts of the signal:

(6) $$\Delta\varphi = cos^{-1}(I_1) - cos^{-1}(I_2)$$

For this equation to work the antennas need to be exactly half a wavelength apart. Otherwise the antenna system needs to be calibrated for the frequency in use. For example, if the frequency in observation would be 2,4 GHz, then the distance would be:

(7) $$\lambda = \frac{\frac{c}{f} * 1}{2} = \frac{3 * 10^8}{2,4 * 10^9} * 0,5 = 0,0625m = 6,5cm$$

With this condition the signal with a transmitter on a 45-degree azimuth on the centre point would be as following:

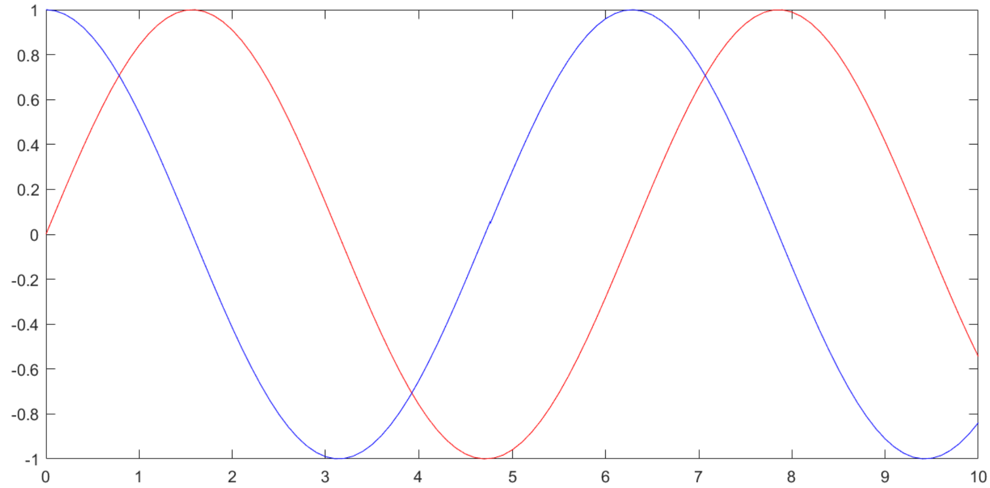


Figure 5: Signal with 45-degree phase difference simulated in MATLAB

The method illustrated on Figure 5 is the output of a two-antenna array and with this method only a 0-180-degree azimuth could be calculated. While it still could show the direction of the incoming radio signal on a 2D plane each measurement would still have two possible outcomes mirroring from the centre line in a 360-degree plot. If two more antennas were added perpendicularly with the same centre point azimuth on a 360-degree scale could be calculated between two antenna arrays using the Pythagoras theorem.

3.1.2 Distance

To find out the minimum signal strength the noise floor needs to be calculated. The example is done using the 61,44MHz bandwidth, which is the maximum bandwidth of some devices considered for this thesis. The theoretical noise floor in decibels is calculated on a room temperature of 20°C, or 290°K, using the following method:

(6) $$P = 10 * log_{10}(1000 * kTB) = 10 * log_{10}(1000 * 290k * 61{,}44 * 10^6) = \sim -96dBm$$ [18]

, where k is Boltzmann's constant of $1{,}38 * 10^{-23}\ J/K$,

T is the room temperature in Kelvin,

and B is the bandwidth of the signal in Hz.

The minimum theoretical signal strength is -96dB. FSPL (*Free Space Path Loss*) defines how much energy gets lost in the environment. This way the theoretical maximum distance can be calculated if the power of transmitter is known. FSPL formula is following:

(6) $$FSPL(W) = \left(\frac{4\pi df}{c}\right)^2$$ [19]

, where FSPL is in watts,

d is distance in km,

f is frequency in Hz,

c is speed of light in m/s.

For example, transmitter is transmitting with the frequency of 2,4GHz with the power of 25dBm and rx and tx antenna gains are about 2dBi, which is an acceptable gain for omnidirectional antennas. Receiver noise floor is calculated to be -96dBm, however a signal needs to be at least 3dB over the noise floor making the minimum required signal strength to be -93dB. If link budget is added up the maximum FSPL can be 122dB (25+2+2-(-93)). From this the maximum distance can be calculated:

$$d = \sqrt{\frac{FSPL * c^2}{(4\pi f)^2}} = \sqrt{\frac{10^{\frac{112}{10}} * (3 * 10^8)^2}{(4\pi * 2{,}4 * 10^9)^2}} = 0{,}229\ km \quad (7)$$

From this calculation it can be said, that theoretically if the receiver is closer than 0,229 kilometres the signal should be discoverable. However, it must be noted, that the signal needs to be high enough from the noise floor to be discoverable. Also, this equation works only with straight line of sight, which is clear of obstacles. In other words, the Fresnel zone must be clear. In most situations natural or artificial objects can cause interference, which leads to signal disruption and causes signal weakening.

3.2 Hardware

For the hardware part an SDR, antennas and testing object are needed. The selection of the hardware parts are explained in the following paragraphs.

3.2.1 Software Defined Radio

The most important hardware part is the signal interpreter. Since the SDR technology has become more relevant during the past decade due to the advancements of the consumer electronics, it was decided to use SDR to complete this task.

In the selection there are multiple SDRs available for this task. In the following paragraph three of the most prominent are discussed.

The first selection was HackRF One by Great Scott Gadgets for its wide 20MHz bandwidth, which can be clocked up to 40MHz and wide frequency range from 1MHz to 6GHz. Also, it has a proven compatibility with GNU Radio software and good online support community. While it only has one channel, multiple devices can be connected in series to match the reference clock to do the signal phase measurements. [20]

The second option is to use LimeSDR by Lime Microsystems. Like HackRF One this is also a low cost SDR meant for developers, students, and inventors. This devices frequency range is narrower than that of HackRF – from 100kHz to 3,8GHz. However, it has a bandwidth of

61,44MHz as standard, which is far more suitable for current operations. Furthermore, it has 2x2 MIMO (Multiple Input Multiple Output) already integrated into the system, which allows to connect two receiving channels without any external hardware. [21] This is useful for the current thesis because the two channels are already synchronised, and no further correlation is needed.

Similarly to LimeSDR BladeRF 2.0 micro xA4 also offers a bandwidth of 61,44MHz, however the frequency range is wider – starting from 70MHz to 6GHz. It is also compatible with most used programming environments and 2x2 MIMO capabilities. [22]

National Instruments brand called Ettus Research produces a series of high-quality software defined radios, which are being used by universities. Options are offered starting from a simple 2x2 MIMO SDR named USRP B210 with 56MHz of bandwidth. Higher end models can be used for wider bandwidth, for example model USRP X300, which has up to 160MHz of bandwidth on each channel [23]. Ettus Research products were not considered for this project due to the high price range.

There were also multiple other SDRs which were taken into consideration but were not mentioned. These devices lacked the minimum frequency requirements for the problem, could not be used as a multi-channel system or cost and availability factors were not acceptable for the project.

The chosen SDR for the project is LimeSDR for its availability at the current time and low purchase cost of 299 dollars [21]. Furthermore, while using an SDR with MIMO already built in further correlation to match the signals can be dismissed in the processing in the software part.

3.2.2 Antennas

To propagate the signal from the environment to the SDR two antennas are needed. Antennas cannot have high directional characteristics in a narrow band area. With higher directional capabilities in a certain direction, the method under current observation could not be constructed effectively.

Most common antennas used are regular monopole or dipole antennas (similar to Figure 3), which consists of a certain length wire. There are also many other shapes and mixtures from different types of usable antennas.

Since frequency range, which is used for this thesis, is also used for wide band of operations, including Wi-Fi, then regular 2,4GHz Wi-Fi antennas can be used for current project.

3.2.3 Testing Devices

To test the device, different transmitters for direction finding are being used which are also shown on Figure 6. For development, a two-way radio MT 975 by Cobra is being used for its ease of use. To test accuracy and stability, a 2,4GHz 8 Channel 500mW video transmitter based on a TS321 amplifier with a monopole antenna is being used. For testing frequency hopping signals, an Eachine i6 remote is being used.

Figure 6: Testing objects - from the left video transmitter (1), two-way radio (2), remote (3).

3.3 Development

Development consisted mainly of three categories: phase measurement, signal detection and analysis and visualisation.

The reason for phase measurement is to calculate the phase difference between two signals to calculate the azimuth of the transmitter. Since the signal for the phase measurement needs to be in narrow bandwidth to eliminate as much noise as possible signal detection is needed. Signal detection finds the frequency and decides how much the original signal is needed to be shifted for it to fit inside the filter pass area. It is also used as a trigger for visualisation part, which is used also for result analysis.

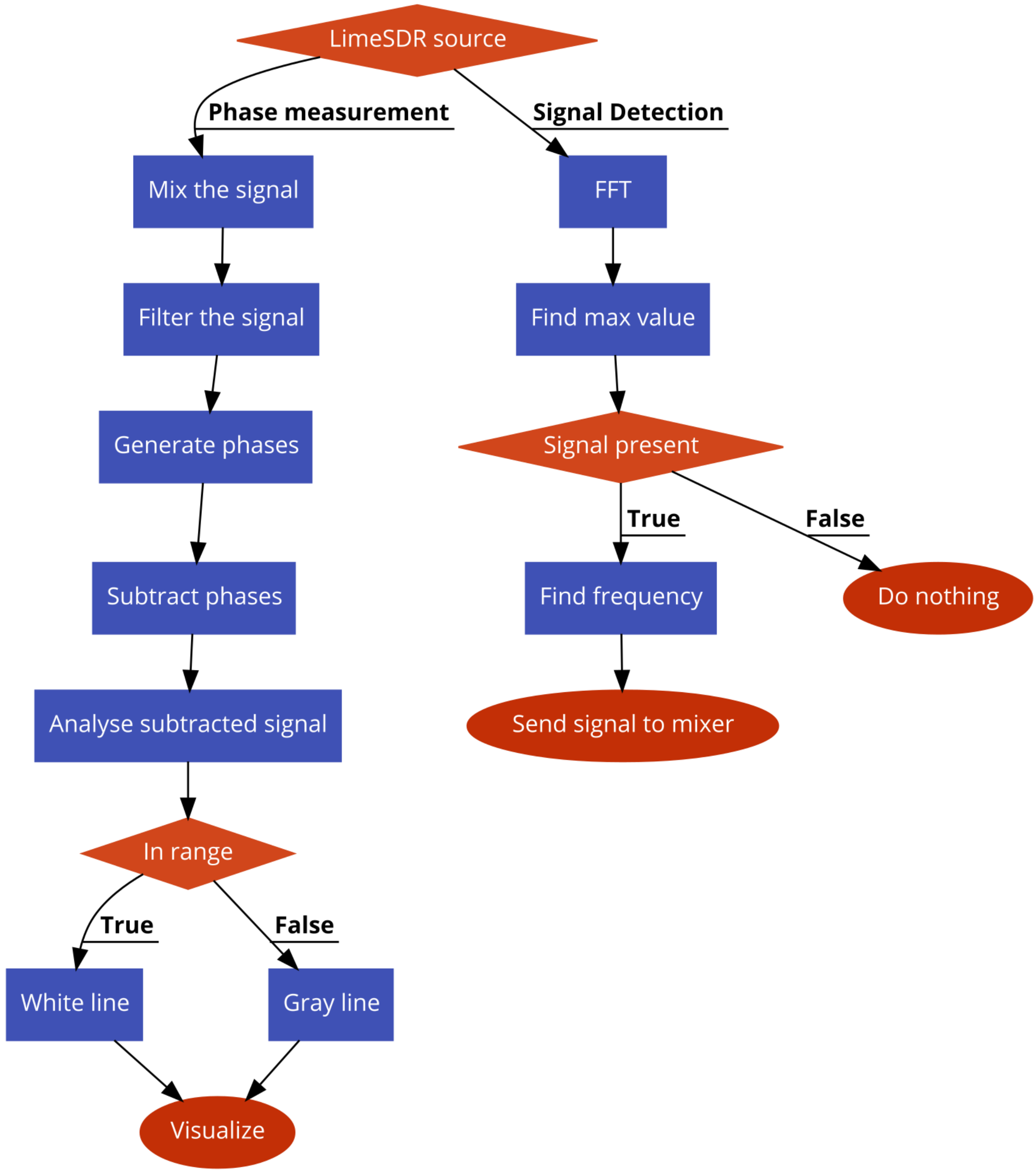


Figure 7: Simplified flow graph of the software. Graph made in code2flow environment.

3.3.1 Phase measurement

Development is done in the graphical programming environment GNU Radio 3.8, which uses Python programming language to do signal processing. Similarly to LabView, it uses units called blocks, which can be connected using arrows.

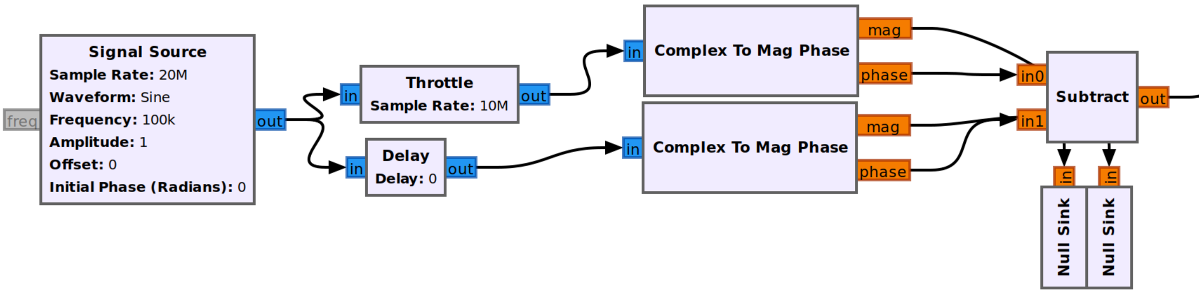


Figure 8: Phase difference calculation in GNU Radio

Firstly, a phase difference method was built using a simulated signal source made by built in waveform generator. Delay block is used with variable slider to simulate the time delay between the signal arrivals. Then the phases are extracted from complex signals and subtracted from each other. This is visualized on Figure 8.

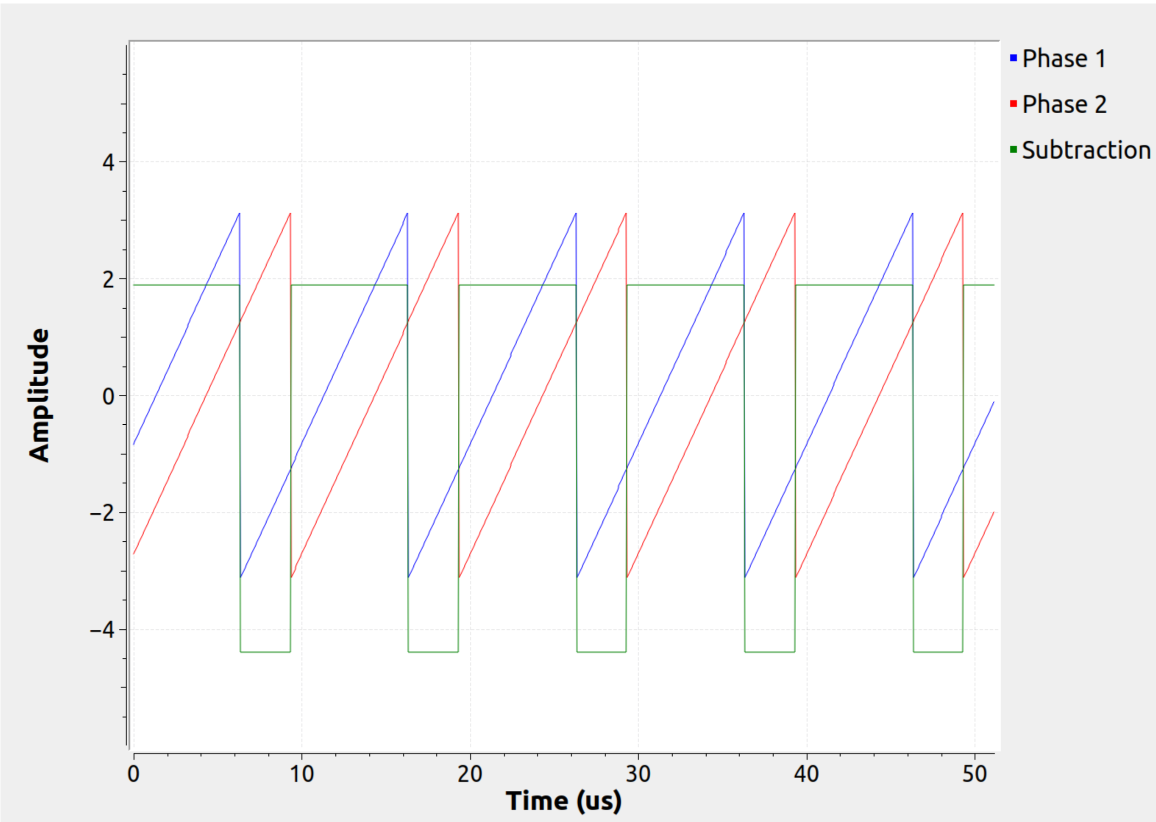


Figure 9: Phase signals (blue phase 1, red phase 2 and green is subtraction) in time domain with delay of 60 samples.

This method gave output, where two sawtooth phase signals were subtracted from each other, resulting in a PWM (Pulse Width Modulation) type signal, where the signal changes amplitude between -2 pi to 2 pi and width is the time difference between peaks of the two signals. This is illustrated on Figure 9 with a green line.

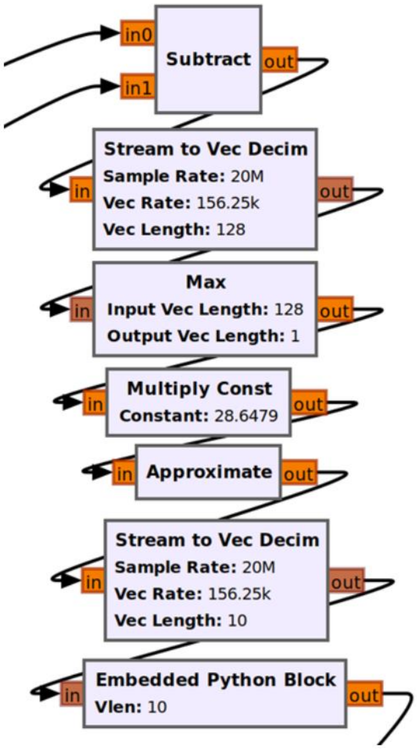


Figure 10: Phase subtraction process with two custom blocks

The positive part of the signal moves from 0 to 2pi, negative part relatively from -2pi to 0. To extract continuous phase initially stream was converted into a vector consisting of 128 points and the maximum value was taken. This value then was converted to degrees from radians and converted back to vector and mode was taken from that vector to statistically choose the most occurring value as is seen on Figure 10.

This method proved to be functional with long lasting signals, however with bursts and short signals this method does not work. Since short signals last only a short time, vectorising the signals causes a problem, where additionally to the signal data noise is added into the vector. This causes false readings and is not reliable.

To eliminate the output value intermittent changing, a custom block containing an if clause was introduced to the scheme. If the blue signal on figure 3 is less than the signal marked in red, then 2pi is added on the blue signal. When using this method phase signal is extracted purely from simulated signal as seen from figure below with a green line.

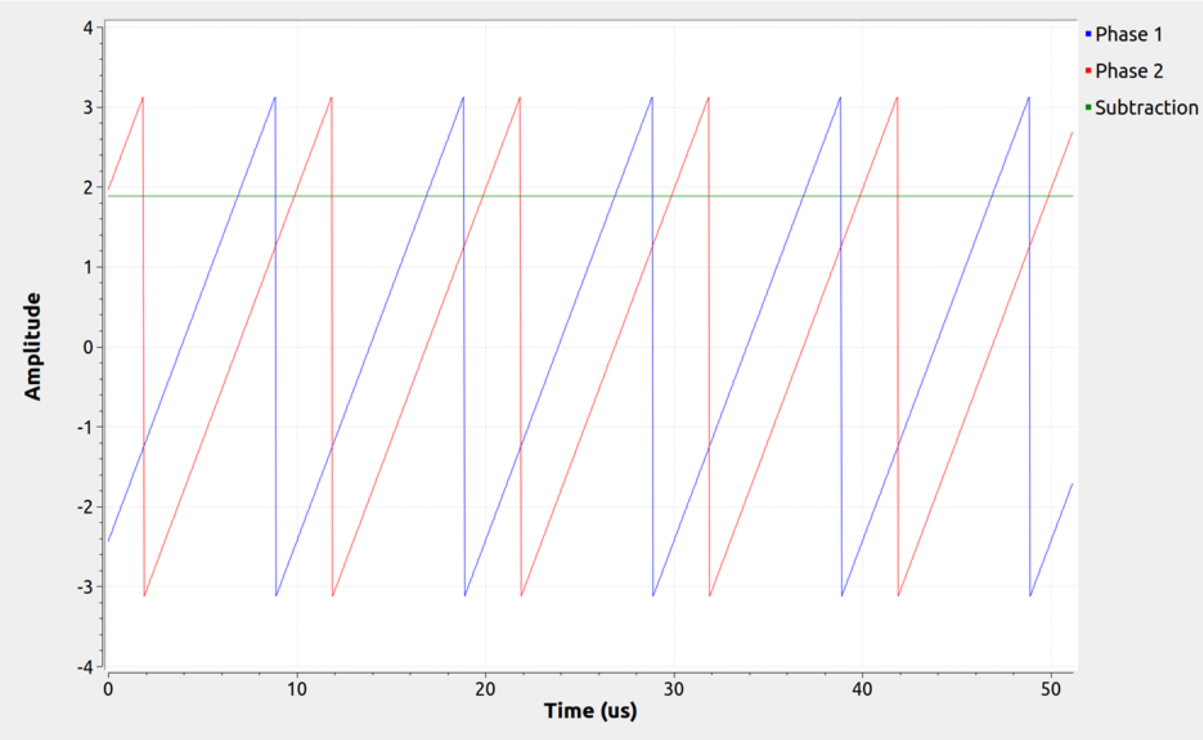


Figure 11: Phase signals (blue phase 1, red phase 2 and green is subtraction) in time domain with delay of 60 samples with added custom if block in GNU Radio QTGUI.

This method produces a solid stream of phase measurements as output. This is streamed out from the GNU Radio environment to local UDP (User Datagram Protocol) port, where it is further examined and displayed.

3.3.2 Signal detection

To detect the signal FFT (Fast Fourier Transform) with 1024 points is done to analyse the signal in frequency domain. Using 1024 points helps to down sample the signal from bandwidth of 61,44 mega samples and fasten the computational speed while still getting enough information for this purpose. From the frequency domain the data is converted to magnitude and the maximum value and the location from the frequency domain is extracted for the further analysis.

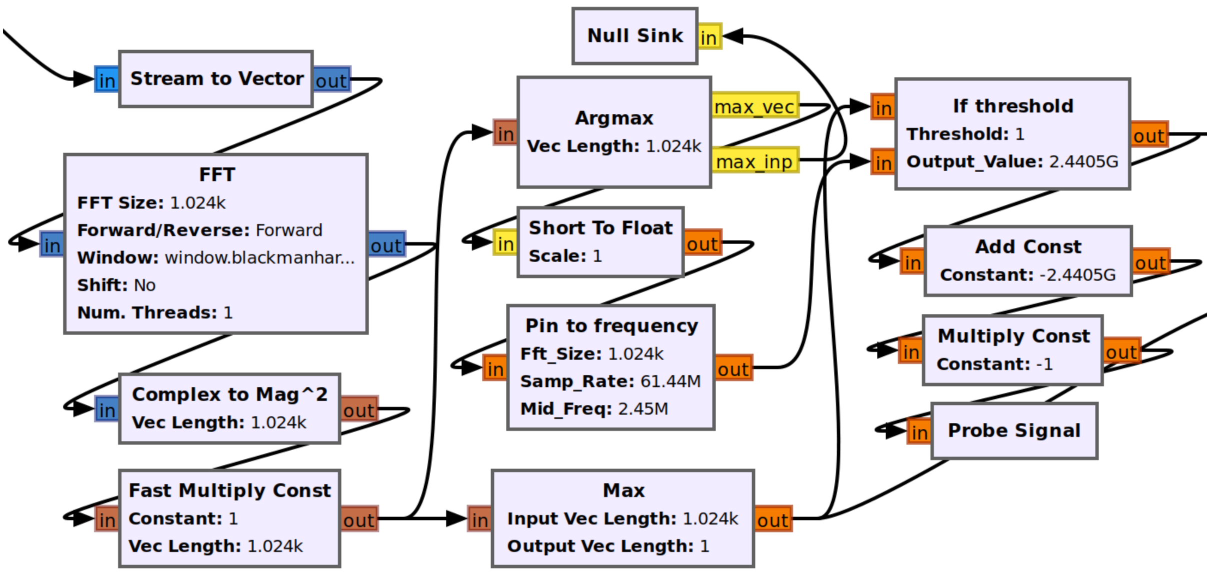


Figure 12: Signal detection part in GNU Radio

In Figure 12 two custom blocks are introduced – bin to frequency block and if threshold block. Bin to frequency block introduces a method, where the frequency is found by observing the bin in the location of the maximum value. Since the bins are in frequency order with the width of the bandwidth, the frequency according to the bin can be calculated using following method:

(8)
$$f_p = loc * \frac{sample\ rate}{fft\ width} + f_c$$

, where f_p is frequency according to the bin number,

loc is the location of the bin in the array,

$fft\ width$ is the width of the FFT,

f_c is the center frequency.

The next custom block is the if threshold, which decides if the maximum output value is above the threshold or not. If the decision is true then the output is the frequency, which it was assigned to, else it outputs the centre frequency, which is already observed.

To get the frequency in observation to the centre the centre frequency is being subtracted and it is multiplied by -1. This is done since the signal generated to the frequency mixer needs to be in the opposite direction of the signal to place the signal in observation in the middle. If everything is calculated the corrugation frequency is sent to function probe.

From the function probe the frequency is given as a frequency to generate a waveform with the signal generator. Then the signal is multiplied to simulate a frequency mixer and is forwarded to a band pass filter and from there the phase is extracted. With this loop the phase measurement is done.

3.3.3 Analysis and visualisation

The signal is streamed out of the GNU Radio programming environment by using UDP ports. This is used to analyse the signal on a different node or programming language if necessary. However, in this thesis, a single node is used and frequency, magnitude, and direction data are received and observed by using Python. For data visualisation purposes matplotlib library is used.

Data is visualised on a 180-degree polar plot, with a white line showing the direction of the signal. While no signal is detected the line is not visible. Also, in the upper left corner of the polar graph is a rectangle, in which the direction is written while being rounded to the nearest integer. The background of the rectangle is black while no signal is detected and red while there is a signal illustrated in Figure 13.

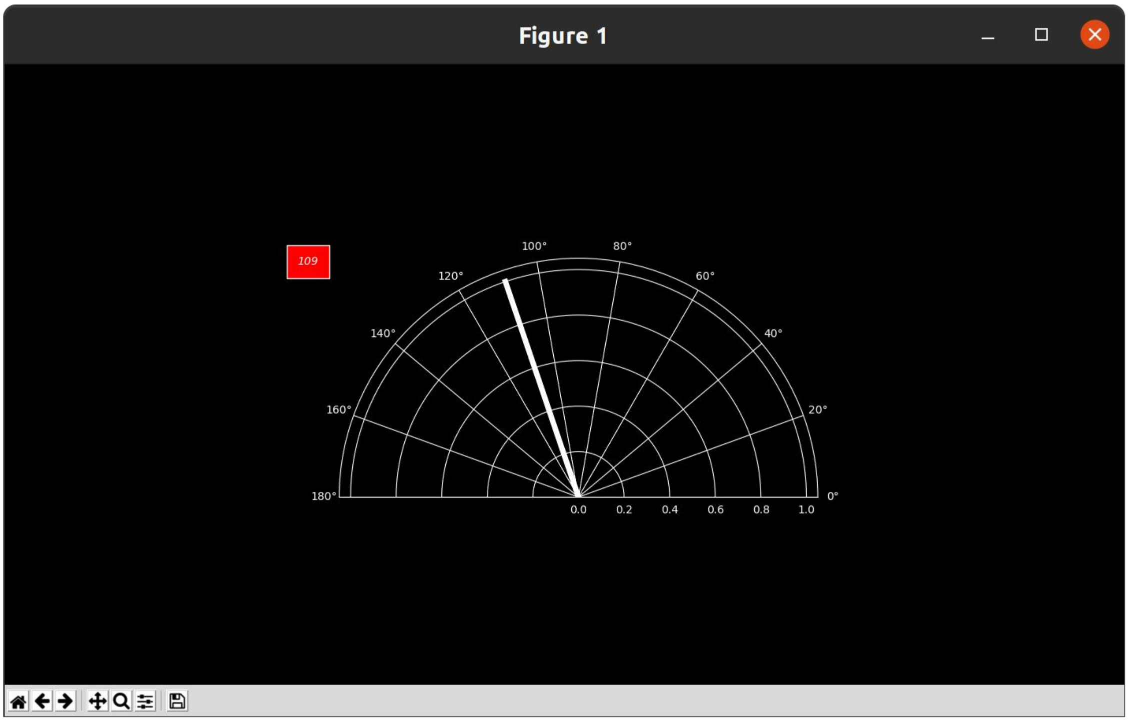


Figure 13: Visualisation of an incoming signal with 109-degree azimuth towards the receiver

When a still standing incoming signal is observed, it is noted, that most of the time signal direction is around same value. However, it is seen that errors come in and produce anomalies, visualised on Figure 14.

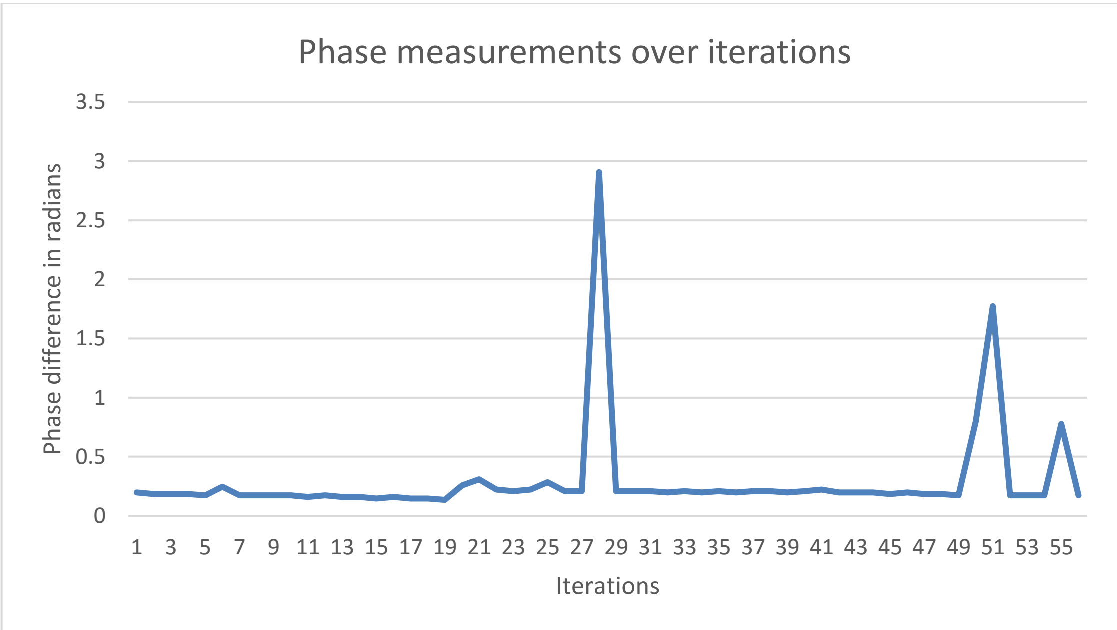


Figure 14: Phase measurements over iterations

It is seen, that while the transmitter is still and most of the measurements are in between 0 and 0,5 radians, there are spikes, that are significantly larger than the rest of the measured phase differences. These false positive measurements can be caused by reflections of the walls and other objects, fault in the subtraction process or some other disruption.

To eliminate false readings an error handling system is created. It saves all the values during continuous measuring and cross-correlates the array from the last 30 measurements with itself, which is then divided by the sum of the measurements. When the measured value range is over 10 degrees of the acquired measurement value the line which is pointing the direction is coloured grey to flag this reading as a possible error.

During further development process the out-of-bounds value was eliminated and the mode of the last 30 measurements is displayed. If the detected signal has less than 30 measurements done it is considered unreliable.

Data transfer between python scripts is done using UDP. Visualising data with matplotlib reduces the computational speed in the computer. By using distributed method results are more reliable and accurate.

4. Testing

Testing was conducted to find out the reliability and usability of the built device. To test the device minimum pulse duration, distance, accuracy, and stability were measured. While the minimum pulse duration was measured in a laboratory in University of Tartu, minimum distance, stability, and accuracy were measured in the former Soviet Raadi military airfield and in Fraternity Rotalia.

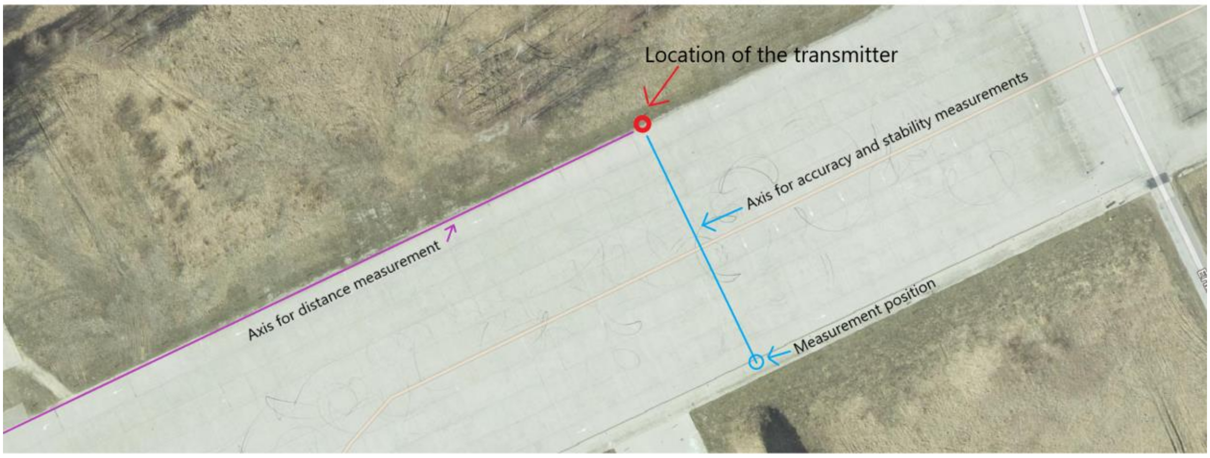


Figure 15: Measurements in the former Soviet Raadi military airfield, picture taken from the Republic of Estonia Land Boards Geoportals orthophoto library.

On the figure above the transmitter's location is shown with a red circle as well as axes for the measurements of the maximum distance with purple line and accuracy and stability with blue line. Also, location of the measurements is shown with a blue circle.

4.1 Minimum pulse duration

Transmitters on remote UAVs often use frequency hopping to have better signal quality by avoiding interference. For this purpose, pulse duration is observed to find out the minimum signal length which the direction finder can detect. Then a frequency hopping remote meant for controlling unmanned vehicles is tested and the values are compared.

To find out the minimum pulse duration the built direction finder was set up and Rigol DSG3060 Signal Generator was set on the frequency of 2,43 GHz with a pulse modulation with a period of 1s and pulse width of 1ms. The built direction finder was set on the centre frequency of 2,44 GHz, bandwidth of 61,44 MHz and a gain of 10 dB. 850-6500 MHz PCB LP WA5VJB Log-periodic antenna by Kent Electronics was connected to the output of the signal generator and placed in proximity of the device. Then the pulse width was increased by 1 ms until signals were detected.

While increasing the pulse duration first pulses were detected with pulse duration of 15ms, however these pulses were always detected at random and not at all times. The device started detecting all the signals with a pulse duration of 83ms.

To find out frequency hopping signals from the remote Rhode & Schwartz HMS 3000 spectrum analyser was used. Centre frequency was set on 2,443 GHz and divider width was set on $200\mu s$.

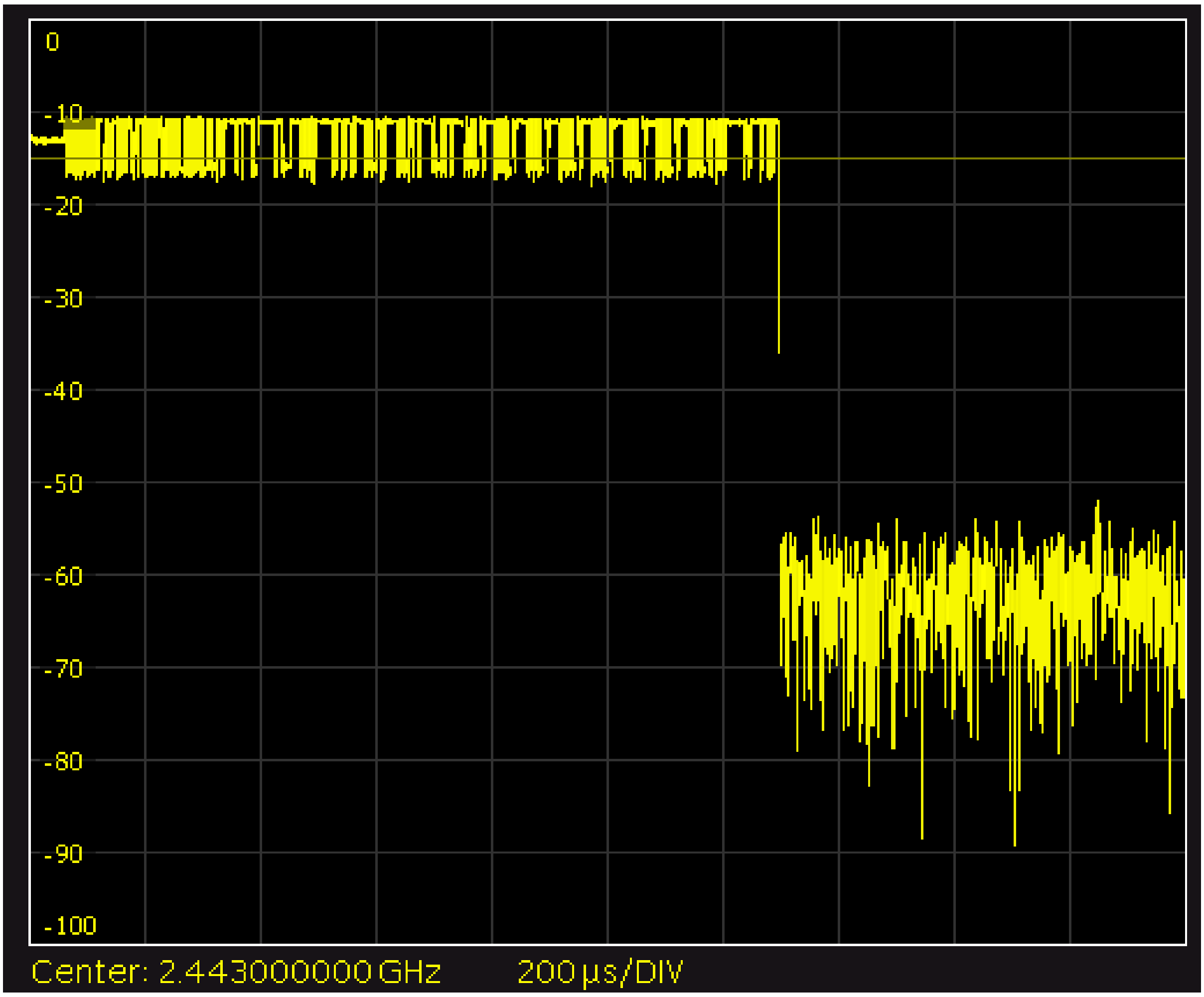


Figure 16: Signal hopping measurements for the Eachine i6 remote

From the Figure 16 it is seen that pulse width is about $60\mu s$. Since the algorithm detected pulses starting from 15ms, which is about 250 times higher, it is not possible to effectively distinguish frequency hopping signals.

4.2 Maximum range

To find out the maximum range, the receiver was moved away from the transmitter along the purple axis illustrated on Figure 15. When the signal detection could not detect the signal anymore then the distance in meters was written down. The gain was set on 60 dB and the bandwidth was 61,44 MHz. The transmitter was located 96cm from the ground and the receiver was set on 1,5m from the ground.

It was found that the signal detection started to lose signal at 140 m. By moving forward, the signal could not be detected at all after 196m. This is close to the theoretical maximum, which was calculated in chapter 3.1.2.

4.3 Stability and Accuracy

To test the stability of the direction finder, two static directions were tested over the course of 30 seconds. Tests were conducted indoors to simulate an environment with multiple reflecting surfaces and outdoors, on a field to minimize reflections. Two antennas with the distance of 6,5 cm were put on an axis, which was rotated from the middle point of antennas towards the transmitter. During testing raw data was collected and put through error handling algorithm as a whole. To reduce the processing power needed for visualisation, the first 1000 measurements were taken from the array.

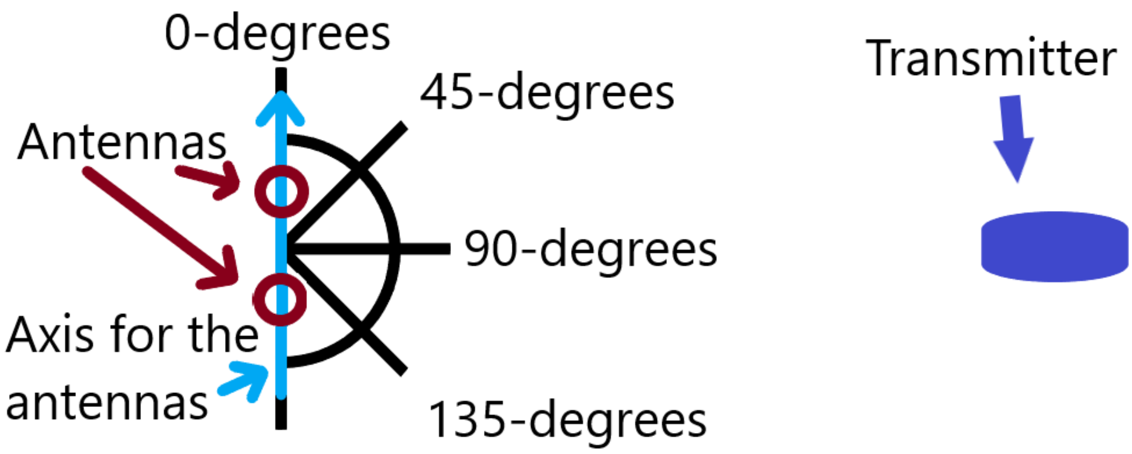


Figure 17: Measurement methodology for stability and accuracy

To find out the stability factor, the mean displacement from the data correlated output is being observed using the following formula:

(9)
$$g = \frac{1}{n}\sum_{i=0}^{n}|d_i - c|$$

, where g is the stability factor,

n is the number of samples in the data,

d_i is one sample in the location of i

c is the correlated output.

Firstly, indoor testing was conducted. The location was the great hall of Fraternity Rotalia, located in Tähe 3, Tartu, Estonia, where the distance between the receiver and transmitter could be set on 20 meters. Direction finder was set to a bandwidth of 61,44 MHz and a gain of 35 dB. The 2,4GHz 8 Channel 500mW video transmitter based on TS321 amplifier was used for testing on channel 1. Antennas were set perpendicularly towards the receiver and data was saved into a file.

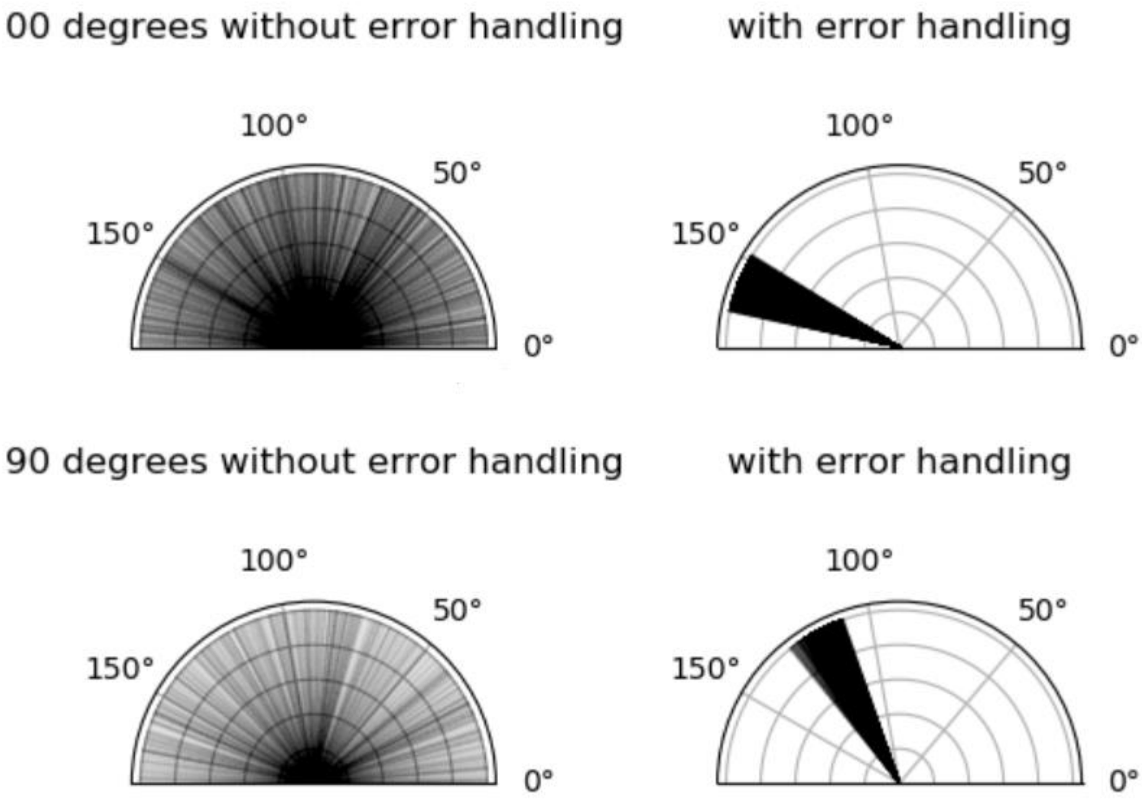


Figure 18: Indoor measurements on 0 and 90 degrees with and without error handling

From the indoor measurements signal appears to come from all the directions and the error handling algorithm gives out a false positive, which does not correspond to the physical direction of the antenna. This direction is most probably the most frequently occurring reflection in the room. Also, the stability factor on 0-degree measurement is 54,6 degrees. However, 90-degree measurement produced the stability factor of 11,6 degrees, which is quite stable. This can show strong reflections coming from one direction, or some hardware error such as displacement.

Outdoor testing was conducted with the same hardware as indoor testing. The distance between transmitter and receiver was set on 70 meters (+- 1 meters as measurement error) on the blue axis on Figure 15. The transmitter was set on 96 cm from the ground and the receiver was set on 1,5m from the ground located on the position on Figure 15. Bandwidth was set on 61,44 MHz and gain on 60dB. Possible nearby reflective objects are concrete ground, trees and two dunes filled with earth.

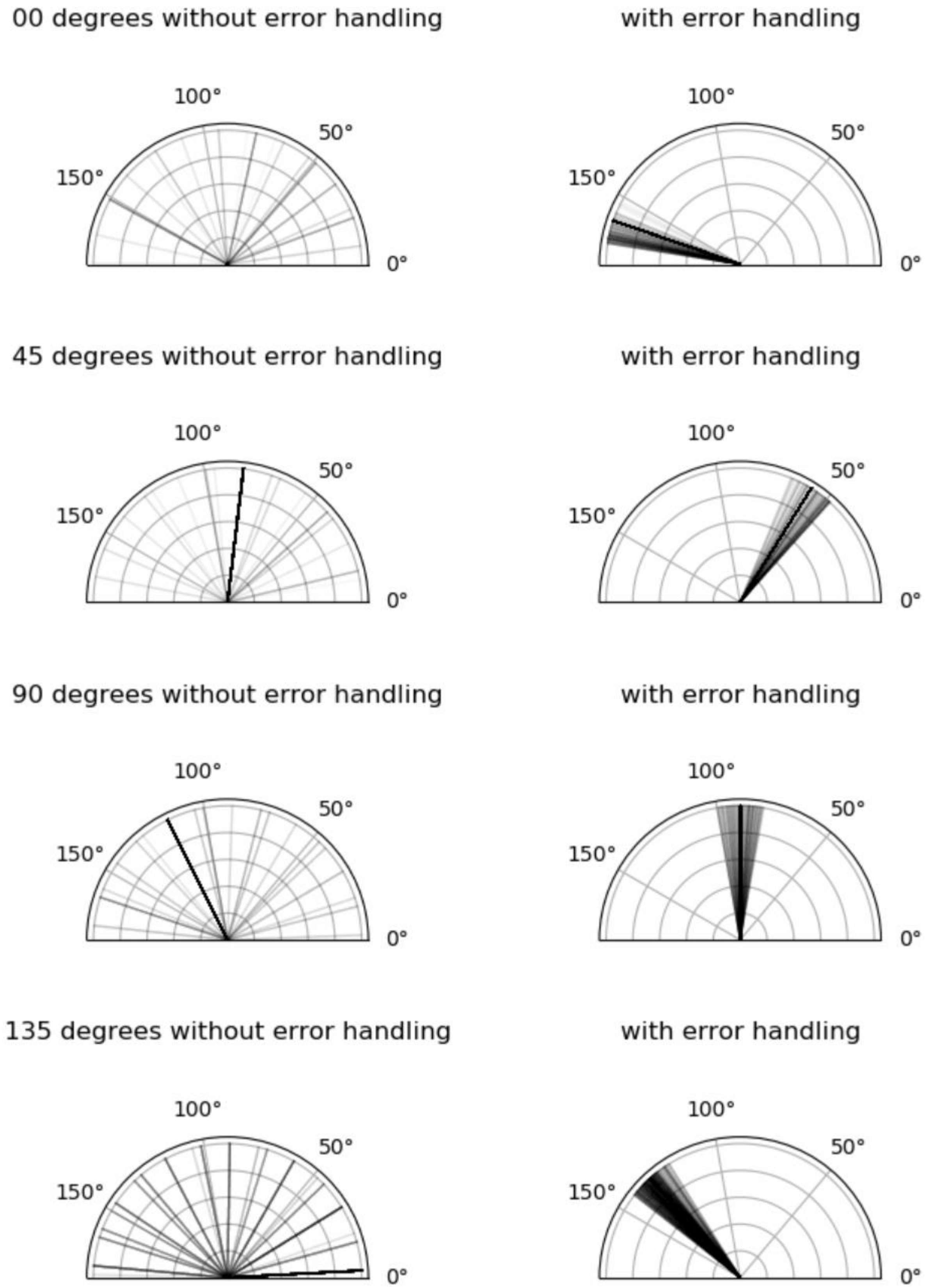


Figure 19: Field test with 0, 45, 90- and 135-degree azimuth angles

Unlike indoor measurements, which had lots of noise, field tests, visualised on Figure 19, show good correlation between the direction of the antenna and the result. In terms of stability it can be seen that raw data without error handling is prone to discover many unwanted signals. The data from both indoor and outdoor measurements is shown in the table below.

Table 1: Results from the testing

Measurement	Indoor 00	Indoor 90	Field 00	Field 45	Field 90	Field 135
Direction	158°	119°	161°	58°	90°	132°
Stability	54,6°	11,6°	52,8°	27,3°	25,4°	10,2°

If the data is looked at separately, the 0-degree measurement produces a correlated output of 161 degrees, which seems far off. However, it must be noted that this is on radial axis where the signal under 0 is considered to be 180 degrees. This noted that the difference is 19 degrees from the calculated signal. Since the error margin on current measurements is set on 20 degrees, it is possible to say that this measurement is acceptable, however by real world standards not very accurate. That is why the stability factor is close to the indoor measurements of 52,8 degrees.

The 45-degree measurement produced more stable signal with correlative output closer to the antenna direction towards the receiver, 58 degrees, which is 13 degrees off the real azimuth. Stability factor for this azimuth was 27,3 degrees.

90-degree measurement was the most accurate, with correlative output of exactly 90 degrees. However, the stability factor was still relatively high – 25,4 degrees.

The most stable measurement was the one with 135 degrees with a stability factor of 10,2 degrees. This had also correlative factor of 132 degrees, which made it quite accurate.

Overall, the accuracy on the field in a range of 45-135 degrees was up to 13 degrees after the corrective measures. Under 45 or over 135 degrees the accuracy rose to the 20-degree range. The stability was not so dependent on the direction.

5. Limitations and Further Improvements

This implemented method has its limitations, which possibly can be eliminated with further improvements.

5.1 Limitations

Main limitations are the vulnerability to reflections, slow reaction time to signals and low range. It is seen from chapter 4.3 that reflected signals and phase errors cause lots of accuracy problems. Therefore, a 10- to 20-degree buffer zone is needed to be taken into consideration as a measurement error to estimate the signal direction from the user.

The range was close to the theoretical maximum that is available by bandwidth used for measurements. However, it is still limited by the omnidirectional characteristic's requirement for the antennas. Since regular monopole antennas were used for these experiments, which do not have good directional characteristics. Improvement to direction finding oriented antennas would have benefits to the range as well to other areas.

The pulse width limitations are mostly hardware oriented. The currently used LimeSDR has no capability to catch fast frequency hopping signals. Also, software improvements could have an impact on signal detection speed. The currently used software measures the signal and then on next iteration it tunes itself onto it. If the signal detection and tuning could be performed on a single iteration, the signal detection time could be made twice shorter.

5.2 Improvements

Firstly, for further improvements a more capable SDR is suggested, which would have faster refresh rate to identify frequency hopping signals. Also, a 4-channel synchronized receiver is suggested to build a direction finder with 360-degree azimuth finding capabilities. The suggested model would be using two synchronized Ettus USRP N300 series with networking support to use the devices for triangulation location finding.

To implement more accuracy and stability, adding a correlative signal processing method to the algorithm is suggested. Also, frequency dependent signal correlation would have benefits for accuracy and stability. For additional range, antennas with more directional capabilities could be used, as well as adding signal amplifiers for additional gain.

6. Conclusions

UAVs have become more threatening to national security than before due to the widespread accessibility and affordable price. This has caused a reason to have more advanced counter measures against devices that may be a threat. While direction-finding methods were available since the beginning of 20^{th} century, hardware has not caught up to being able to effectively observe the 2,4 GHz band until recently. Also, more user-friendly approach to user interface has made it more appealing for hobbyist and enthusiasts to use the hardware and implement methods of interest.

The main reason for this thesis was to implement a phase difference method on a low cost SDR without any complicated hardware while making the finished device still usable as a direction finder for UAVs. While the tests showed that it is not able to recognise frequency hopping signals, it is able to identify and find out the direction of with a 20-degree accuracy from the distance of 70 meters. The maximum range it can identify UAV video transmitters signal without interruptions is 140 meters while it can identify glimpses of a signal from a distance of 196 meters.

Overall, it can be said that the method works, however it needs some improvements to be deployable. Improvements include hardware and software updates as well as well considering additional amplifiers, signal propagating additions and an improved signal processing method.

Acknowledgement

The student has received grant for this thesis from Ministry of Defence to generate the growth in the research of national defence topics.

Also, the student is very grateful to all the help from his supervisor Jaanus Kalde as well as Kalev Märtens from Rhode & Schwartz, Andres Moks from Estonian Aviation Academy and Liis Balogh from Clarified Security.

Signed,

Rein Jaks

Bibliography

[1] M. A. Ma'sum, “Simulation of intelligent Unmanned Aerial Vehicle (UAV) For military surveillance,” in *International Conference on Advanced Computer Science and Information Systems (ICACSIS*, Bali, 2013.

[2] E.-. j. i. “Eesti raadiosageduste plaan,” Riigi Teataja, 26 10 2017. [Online]. Available: https://www.riigiteataja.ee/akt/125012019006. [Accessed 03 05 2020].

[3] S. E. Lipsky, Microwave Passive Direction Finding, Raleigh: SciTech Publishing Inc., 2004.

[4] P. Handley, “Radio Direction Finding Techniques,” Hackaday, 23 07 2017. [Online]. Available: https://hackaday.io/project/25995-bloodhound-autonomous-radiolocation-drone/log/63866-radio-direction-finding-techniques. [Accessed 25 04 2020].

[5] E. A. N. Services, “Seire,” Estonian Aviation Navigation Services, [Online]. Available: http://www.eans.ee/seire. [Accessed 27 04 2020].

[6] M. &. K. M. &. A. M. &. A. S. Akram, “ecision Tree for Selection Appropriate Location Estimation Technique of GSM Cellular Network,” in *International Conference on Engineering & Emerging Technology, At Superior University*, Lahore, Pakistan, 2014.

[7] I. Pellejero, “Adcock/Watson-Watt Radio Direction Finding,” Ismael Pellejero Ibáñez, [Online]. Available: http://www.ipellejero.es/tecnico/adcock/english.php. [Accessed 16 05 2020].

[8] B. Donath, “Direction finder operating according to the Watson-Watt principle”. Munich, DE Patent 4754281, 29 09 1986.

[9] R. Hammerle, “Factors limiting the accuracy of Doppler and Adcock direction finding systems,” in *IEE Colloquium on Passive Direction Finding*, London, UK, 1989.

[10] “Radiomonitoring & Radiolocation,” Rohde&Schwarz, 2011.

[11] F. K. Jondral, "Software-Defined Radio—Basics and Evolution to Cognitive Radio," *EURASIP Journal on Wireless Communications and Networking,* p. 9, 2005.

[12] "The Wireless Innovation Forum," [Online]. Available: https://www.wirelessinnovation.org/assets/documents/SoftwareDefinedRadio.pdf. [Accessed 05 16 2020].

[13] Rhode & Schwarts, "R&S®DDF007 Portable Direction Finder Product Brochure," 2016. [Online]. Available: https://scdn.rohde-schwarz.com/ur/pws/dl_downloads/dl_common_library/dl_brochures_and_datasheets/pdf_1/DDF007_bro_en_5214-3700-12_0400.pdf. [Accessed 03 05 2020].

[14] KerberosSDR, "About," KerberosSDR, 2020. [Online]. Available: http://kerberossdr.com/about/. [Accessed 03 05 2020].

[15] D. S. T. M. Sam Whiting, "Direction of Arrival Analysis on a Mobile Platform," in *GNU Radio Conference 2017,* San Diego, 2017.

[16] H. V. K. Tarkpea, "Elektromagnetism," in *Elektromagnetlained,* Tartu, MAURUS KIRJASTUS, 2019, p. 152.

[17] "What is I/Q Data?," National Instruments, 16 10 2019. [Online]. Available: http://www.ni.com/tutorial/4805/en/. [Accessed 28 04 2020].

[18] "What is Noise Floor for radio receivers," Electronics Notes, [Online]. Available: https://www.electronics-notes.com/articles/radio/radio-receiver-sensitivity/what-is-noise-floor.php. [Accessed 16 05 2020].

[19] "Free Space Path Loss: details & calculator," ElectronicsNotes, [Online]. Available: https://www.electronics-notes.com/articles/antennas-propagation/propagation-overview/free-space-path-loss.php. [Accessed 12 05 2020].

[20] M. Ossmann, "GitHUB," Great Scott Gadgets, 12 09 2017. [Online]. Available: https://github.com/mossmann/hackrf/wiki/HackRF-One. [Accessed 05 05 2020].

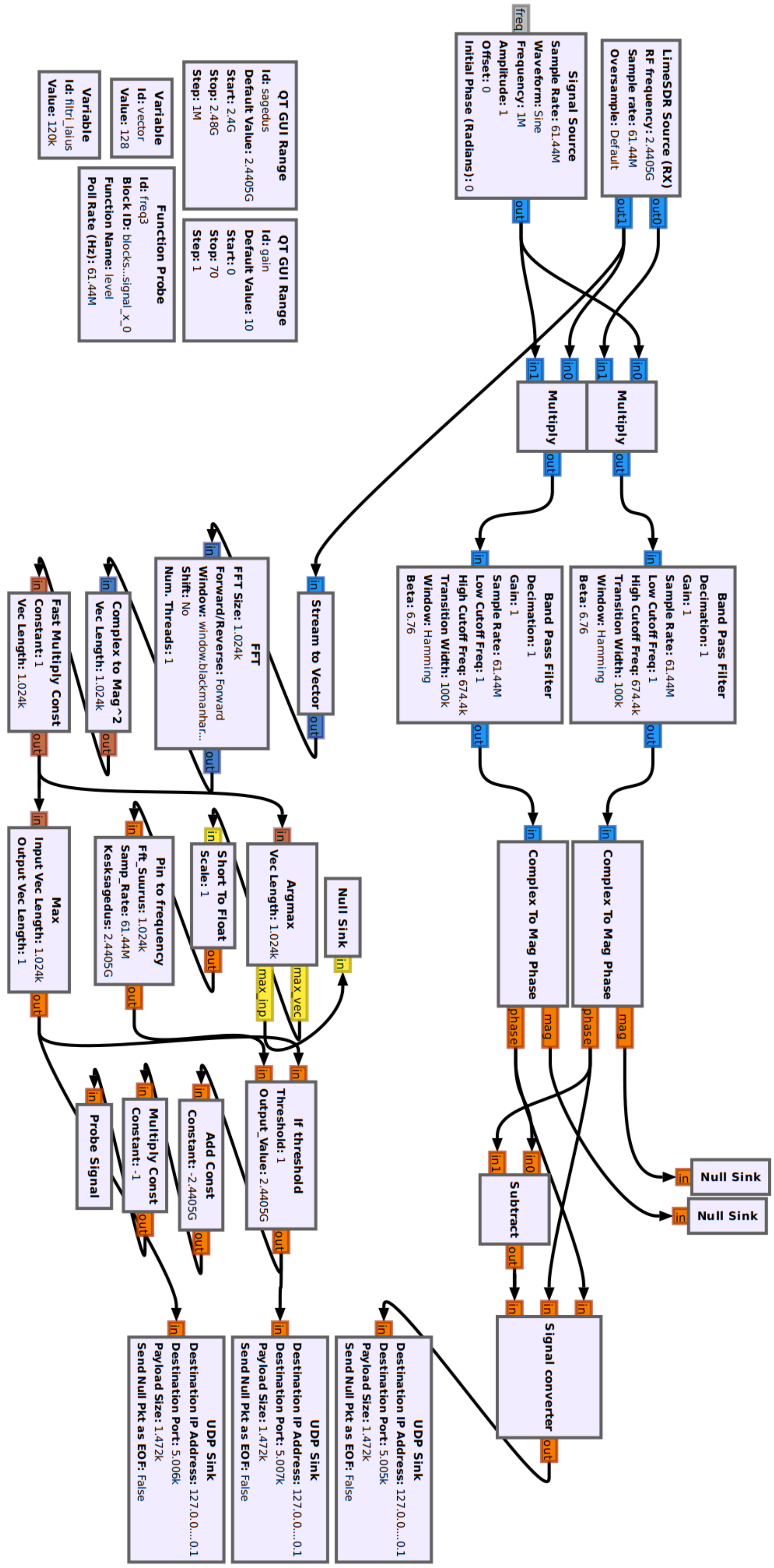
[21] L. Microsysems, "LimeSDR," Lime Microsystems, 2020. [Online]. Available:

https://limemicro.com/products/boards/limesdr/. [Accessed 05 05 2020].

[22] N. LLC, “bladeRF 2.0 micro,” Nuand, [Online]. Available: https://www.nuand.com/bladerf-2-0-micro/. [Accessed 05 05 2020].

[23] E. Research, “USRP Hardware Driver and USRP Manual,” [Online]. Available: https://files.ettus.com/manual/index.html. [Accessed 16 05 2020].

Appendix I – Flow graph of the GNU Radio program



Appendix II – Codes for custom blocks

Codes are available also in electronic source: https://github.com/reinj/Direction-finder

Signal converter

Written inside GNU Radio Embedded Python block

```
class blk(gr.sync_block):  # other base classes are basic_block, decim_bloc
k, interp_block
    """Embedded Python Block example - a simple multiply const"""

    def __init__(self):  # only default arguments here
        """arguments to this function show up as parameters in GRC"""
        gr.sync_block.__init__(
            self,
            name='Signal converter',   # will show up in GRC
            in_sig=[np.float32,np.float32,np.float32],
            out_sig=[np.float32]
        )


    def work(self, input_items, output_items):

        a = 2*np.pi
        if input_items[0][0] < input_items[1][0]:
            output_items[0][:] = input_items[2][0] + a
        else:
          output_items[0][:] = input_items[2][0]

        return len(output_items[0])
```

Pin to frequency

Written inside GNU Radio Embedded Python block

```
class blk(gr.sync_block):  # other base classes are basic_block, decim_bloc
k, interp_block
    """Embedded Python Block example - a simple multiply const"""

    def __init__(self, fft_size=1024.0, samp_rate=20000000.0, centre_freque
ncy=2440000000.0):  # only default arguments here
        """arguments to this function show up as parameters in GRC"""
        gr.sync_block.__init__(
            self,
```

```
            name='Pin to frequency',   # will show up in GRC
            in_sig=[np.float32],
            out_sig=[np.float32]
        )
        # if an attribute with the same name as a parameter is found,
        # a callback is registered (properties work, too).
        self.fft_size = fft_size
        self.samp_rate = samp_rate
        self.centre_frequency = centre_frequency

    def work(self, input_items, output_items):

        out-
put_items[0][:] = input_items[0] * (self.samp_rate/self.fft_size) + self.ce
ntre_frequency
        return len(output_items[0])
```

If threshold

Written inside GNU Radio Embedded Python block

```
class blk(gr.sync_block):  # other base classes are basic_block, decim_bloc
k, interp_block
    """Embedded Python Block example - a simple multiply const"""

    def __init__(self, threshold=1.0, output_value=0.0):  # only default ar
guments here
        """arguments to this function show up as parameters in GRC"""
        gr.sync_block.__init__(
            self,
            name='If threshold',   # will show up in GRC
            in_sig=[np.float32,np.float32],
            out_sig=[np.float32]
        )
        # if an attribute with the same name as a parameter is found,
        # a callback is registered (properties work, too).
        self.threshold = threshold
        self.output_value = output_value

    def work(self, input_items, output_items):
        if input_items[0][0] > self.threshold:
          output_items[0][:] = input_items[1]
        else:
          output_items[0][:] = self.output_value
        return len(output_items[0])
```

Appendix III – Signal visualisation

Codes are available also in electronic source: https://github.com/reinj/Direction-finder

```
def direction():
    # Code copied from: https://wiki.python.org/moin/UdpCommunication
    UDP_IP = "127.0.0.2"
    UDP_PORT = 5000
    sock = socket.socket(socket.AF_INET, # Internet
                         socket.SOCK_DGRAM) # UDP
    sock.bind((UDP_IP, UDP_PORT))

    while True:
        data, addr = sock.recvfrom(1024) # buffer size is 1024 bytes
    # End of copied code
        return float(data)

def frequency():
    # Code copied from: https://wiki.python.org/moin/UdpCommunication
    UDP_IP = "127.0.0.2"
    UDP_PORT = 5000
    sock = socket.socket(socket.AF_INET, # Internet
                         socket.SOCK_DGRAM) # UDP
    sock.bind((UDP_IP, UDP_PORT))

    while True:
        data, addr = sock.recvfrom(1472) # buffer size
    # End of copied code
        a = struct.unpack_from('f', data)
        return a[0]

def main():

    dir1 = direction()
    print("Direction aquired: ", dir1)
    if dir1 <= 181:
        freq = frequency()
        dir_rad = np.radians(dir1)
        print("Direction: ", dir1)
        print('Frequency: ', freq)
        color1 = 'white'
        color2 = 'red'
    else:
        dir_rad = 0
        freq = 0
        color1 = 'black'
        color2 = 'black'
```

```
    matplotlib.rcParams['figure.figsize'] = (8,4.5)
    plt.style.use('dark_background')
    ax1 = plt.subplot(111, projection='polar')
    ax1.set_rmax(1)
    ax1.set_thetamin(0)
    ax1.set_thetamax(180)
    ax1.text(2.45, 1.6, round(dir1), style='italic',
        bbox={'facecolor': color2, 'alpha': 1, 'pad': 10})
    ax1.plot([0,dir_rad], [0,1], color = color1, linewidth=5.0)
    plt.show(block=False)
    plt.pause(0.001)
    plt.clf()

while __name__ == "__main__":
    main()
```

Appendix IV – Error Handling

Codes are available also in electronic source: https://github.com/reinj/Direction-finder

```
def direction_get():
    # Code copied from: https://wiki.python.org/moin/UdpCommunication
    UDP_IP = "127.0.0.1"
    UDP_PORT = 5005
    sock = socket.socket(socket.AF_INET, # Internet
                     socket.SOCK_DGRAM) # UDP
    sock.bind((UDP_IP, UDP_PORT))

    while True:
        data, addr = sock.recvfrom(1472) # buffer size
    # End of copied code
        a = struct.unpack_from('f', data)
        print("Direction data recieved: ", a[0])
        return a[0]

def direction_send(data):
    data1 = str(data)
    # Parts of code copied from: https://wiki.python.org/moin/UdpCommunication
    UDP_IP = "127.0.0.2"
    UDP_PORT = 5000
    MESSAGE = bytes(data1, encoding = 'utf-8')

    sock = socket.socket(socket.AF_INET, # Internet
                        socket.SOCK_DGRAM) # UDP
    sock.sendto(MESSAGE, (UDP_IP, UDP_PORT))
    # End of copied code

def magnitude():
    # Code copied from: https://wiki.python.org/moin/UdpCommunication
    UDP_IP = "127.0.0.1"
    UDP_PORT = 5006
    sock = socket.socket(socket.AF_INET, # Internet
                     socket.SOCK_DGRAM) # UDP
    sock.bind((UDP_IP, UDP_PORT))

    while True:
        data, addr = sock.recvfrom(1472) # buffer size
    # End of copied code
        a = struct.unpack_from('f', data)
        print('Magnitude data: ', a[0])
        return a[0]

def error_handler(temp1):
```

```python
    a = np.correlate(temp1,temp1)
    b = int(a[0]/sum(temp1) )
    tolerance = 10 # How many degrees is the tolerance rate
    b1 = b - tolerance
    b2 = b + tolerance
    n = 0
    for i in temp1:
        if b1 <= i <= b2:
            i = i
        else:
            temp1[n] = b
        n = n + 1
    return np.radians(temp1)


def main():
    dir1 = np.round(np.degrees(direction_get()))/2
    mag = magnitude()
    data = np.loadtxt('directions.txt')
    print("Text loaded")
    no_signal = 255 # random number over 180 is sent to identify that there is
 no signal present
    if mag >= 1:
        while len(data) < 10:
            with open('directions.txt', 'a') as dirfile:
                dirfile.write("{}\n".format(str(dir1)))
        else:
            with open('directions.txt', 'a') as dirfile:
                dirfile.write("{}\n".format(str(dir1)))
            yes_signal = error_handler(data)
            direction_send(yes_signal[0])
        print("Positive packet sent")
    else:
        with open('directions.txt', 'w') as dirfile:
            dirfile.write("{}\n".format(str()))
        direction_send(no_signal)
        print("Negative packet sent")

while __name__ == "__main__":
    main()
```

Non-exclusive licence to reproduce thesis and make thesis public

I, Rein Jaks

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce,for the purpose of preservation, including for adding to the DSpace digital archives untilthe expiry of the term of copyright,

   “UAV Direction Finding Using Phase Difference”

   supervised by Jaanus Kalde
2. I grant the University of Tartu a permit to make the work specified in p. 1 available tothe public via the web environment of the University of Tartu, including via the DSpacedigital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercialuse of the work until the expiry of the term of copyright.
3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.
4. I certify that granting the non-exclusive licence does not infringe other persons’ intellectual property rights or rights arising from the personal data protection legislation.

Rein Jaks
21.05.2020