University of Tartu

Faculty of Science and Technology

Institute of Technology

Erik Amor

Design, Testing and Analysis of an Affordable Direction of Arrival System Using Off-The-Shelf SDR Hardware and Software

Master's thesis (30 EAP)
Robotics and Computer Engineering

Supervisor:

MSc Jaanus Kalde

Tartu 2022

Abstract/Resüumee

Design, Testing and Analysis of an Affordable Direction of Arrival System Using Off-The-Shelf SDR Hardware and Software

The growing use of wireless communication increases the need for devices to determine the location of the signal source. Affordable, accessible and universal devices can have a great role in addressing this aside precision tools. Combining popular off-the-shelf radio receivers with direction of arrival algorithms could provide a convenient tool. The setup is proven to give sufficient results for several location applications using HackRF radio and signal processing in software. The accuracy of the system can be improved by characterising the system for each use case. After the setup is completed, such system can still be convenient to use for non-technical end user.

CERCS: T120 Systems engineering, computer technology; T121 signal processing; T125 Automation, robotics, control engineering.

Keywords: Software-defined radio, SDR, Direction of arrival, DOA

Univresaalset ja vabalt kättesaadavat riistvara ja tarkvara kasutava signaali suunamääramise süsteemi disain, testimine ja analüüs.

Juhtmevabade edastussüsteemide kasvav kasutus tõstab nõudlust ka lihtsasti ligipääsetavate signaali allikate lokaliseerimist võimaldavate seadmete järele. Lisaks täpsetele ja kallitele seadmetele pakub head alternatiivi universaalset vastuvõtjat kasutav tarkvaralise raadio baasil loodud süsteem. Kasutades HackRF platvormi signaali vastuvõtjana ja olemasolevaid signaali suuna hindamise algoritme on võimalik saada hinnang signaali suunale, mille täpsus on piisav enamikes rakendustes. Hinnangut on võimalik parandada viies läbi süsteemi kalibreerimine vastavalt kasutusjuhule. Peale süsteemi üles seadmist ja kalibreerimist on sellise süsteemi kasutamine vaatamata universaalsusele lõpp-kasutajale endiselt mugav ja ei vaja erialasi teadmisi.

CERCS: T120 Süsteemitehnoloogia, arvutitehnoloogia; T121 signaalitöötlus; T125 Automatiseerimine, robootika, control engineering

Märksõnad: arvutid, tarkvaraline raadio, suunamääramine

Contents

List of figures

List of tables

Acronyms

SDR - Software-Defined Radio;

GRC - GNU Radio Companion;

DOA - Direction Of Arrival;

POC - Proof-Of-Concept;

UI - User Interface;

RF - Radio Frequency, used to describe type of hardware, signal and software tools;

FFT - Fast Fourier Transform;

MUSIC - Multiple Signal Classification;

JS - JavaScript;

1 Introduction

With the increasing development of technology, wireless communication has become a more important part of everyday life. The electromagnetic spectrum all around every one of us is crowded with all kinds of signal with most of us barely being aware of that. Still, with the increasing number of devices using this common resource, regulating the use and ensuring safe operation become more important. For both of those reasons, gathering information about the current situation in the electromagnetic domain is necessary.

Several applications and services rely on certain parts of the spectrum to be available for them to transmit a signal when needed. Therefore, it is crucial to make sure that the required frequency range is free and not used by non-authorised users. It is especially important as there are no physical restrictions for anyone to transmit a signal of their choice. Enforcing the correct use of spectrum and to held any violators accountable requires authorities first to recognise the signal and then locate the source.

Knowing the location of an RF signal source can also be helpful when searching for lost people and tracking vehicles. In any case, a special device for carrying out the operation is necessary. Although these systems exist in most facilities that require those, their amount is often limited, therefore increasing the reaction time in cases they are needed. The limitation can be because of the high cost or complexity of the system.

At the same time, software-defined radio (SDR) systems have gained popularity [2]. These systems allow using common computers together with universal RF receivers and transmitters. By separating the hardware from the specific application, they give the end-user freedom to do more data processing using the same hardware [3].

This concept simplifies the hardware design of necessary radio devices since less processing is done in hardware, bringing down the cost. Not being application-specific allows selling more of the same design, reducing the end cost even more. This results in low-cost and widely available radio hardware for receiving and analysing the majority of RF signals. It also allows development of peripheral-independent tools to work on the signals, resulting in many open-source applications and libraries being developed and available for everyone to be used.

Although the direction of arrival (DOA) estimation systems make use of SDR features, there are not a lot of universal solutions available which make use of the platform-independent development and low-cost peripherals. As these devices and tools are not developed for this specific purpose, there are several extra steps to consider when using them in this application. Making use of the potential of these universal and widely available tools to create a low-cost and widely usable DOA estimation system is the source of motivation for the thesis.

1.1 Related work

With the DOA estimation being an increasingly important field, there is much research done on this topic. A lot of this research considers more precise DOA detection of certain signals by finding a more customised algorithm or antenna array setup to provide the best estimate for the frequency, modulation and environment parameters of the specific application [4] [5]. Another branch of research concentrates on making the DOA devices smaller for any portable and hand-held devices [6]. In recent years, also machine learning based DOA estimation has gained popularity, being analysed both as a separate method or as an addition to existing hardware [7] [8].

All this research gives a lot of insight into how to optimize algorithms and setup for specific signals or how to avoid effects caused by a noisy environment, antenna placement and coupling and get more precise results. To the authors' best knowledge, there is less research on making the DOA more universal in both physical antenna placement setup and detectable signals.

1.2 Goals

The goal of this project is to make an affordable DOA system based on software-defined radio. The project consists of finding the best possible hardware setup, considering cost and availability as well as figuring out the algorithm that works the best with the chosen hardware solution. To evaluate the outcome of the project, some requirements for the desired system were set, that are covered in **1.1**.

Table 1.1: Set requirements for the DOA system

Requirement	Priority	Reasoning
Low hardware cost	HIGH	The idea of the system is to reduce the requirements on accuracy and cost of single system by allowing more nodes of the system to be distributed and work together.
Hardware available and open-source	HIGH	There are several DOA systems available that are achieved using specially made hardware that results in costly systems. Those systems are also harder to modify to specific needs. The system being developed should be as versatile and customisable as possible.
Output certainty information	HIGH	The system should not only output the estimated DOA, but it should also convey some information about how certain the offered value is. This would give the user of the system chance to use a method of analyzing the results depending on the application.
Universal application	HIGH	The system should work with different antenna configurations and signal frequencies.
Real-time capability	MEDIUM	For most applications having constant real-time output with no intervals when the signal is not scanned is beneficial. For this to be possible, the algorithm has to work at least as fast as the data is acquired.
Low user input on regular use	MEDIUM	For the end system it is beneficial if, once set up, the use of the system would be as user-independent as possible. This would allow for better system distribution as the workers do not have to be each experts in the field.

2 Problem overview

2.1 Physical principle

When having multiple antennas located in some area in a way that there is some spacing between them, the signal reaches the antennas at a different time as the speed of radio waves in air has a constant and known speed. The difference of the time when the same part of signal reaches two of such placed antennas depends on the direction the signal comes from, relative to the antennas. The basic principle is covered in figure **2.1**.

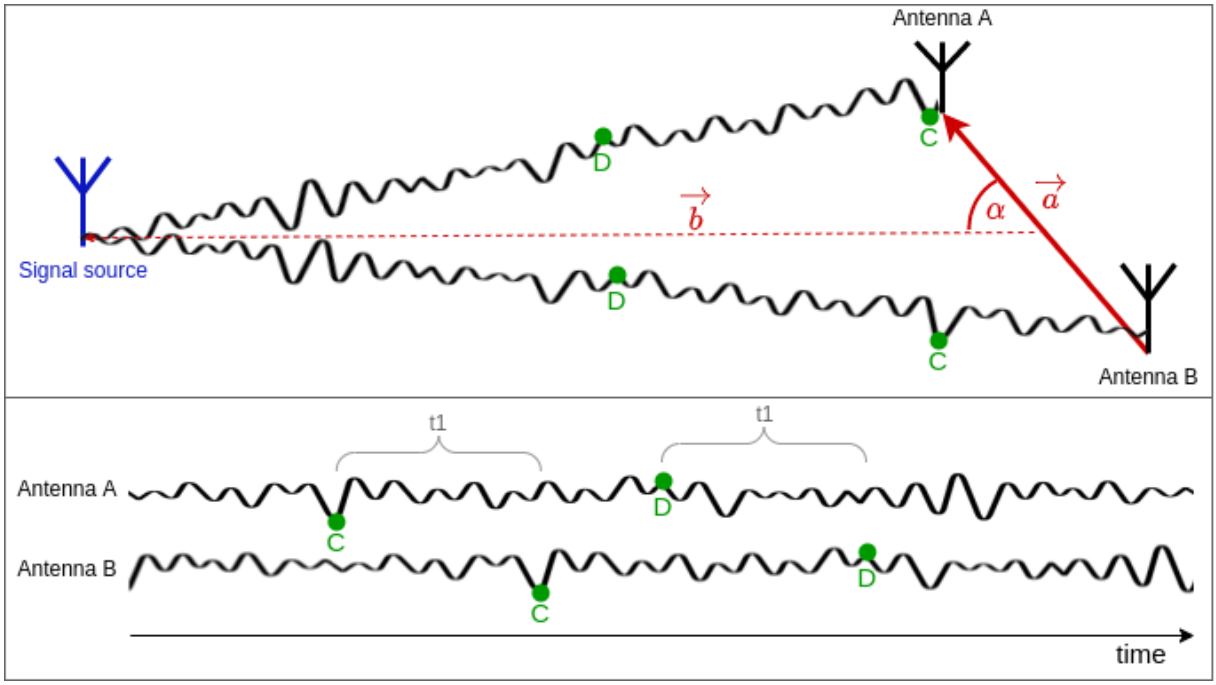

Figure 2.1: The basic principle of finding signal direction using stationary antennas. The example (1) shows how a signal from a source reaches two antennas at a certain time. Vector $\vec{a}$ connects the bases of antennas used and vector $\vec{b}$ connects the base of the coordinate system (mid point of antennas) to the signal source. The angle between those vectors can be calculated using the delay of signal arrival time between the antennas (2). It can be seen, that each point on the wave reaches one of the antennas certain time $t1$ earlier than another

If this data can be gathered in a synchronised way and it is possible to determine from the received signal, how much earlier the signal reached one of the antennas, it is possible to calculate the angle that the vector towards the source of the signal makes with the vector from one antenna to the other. As the delay would be exactly same from the symmetrical point from the vector connecting the antennas, the calculation would yield two possible results. This effect

is illustrated in figure 2.2. To overcome this ambiguity, another antenna could be added to the system such that it is not on line with the first two antennas. As shown in figure 2.3 in such case the false directions do not match, but the correct direction is the same for each pair of antennas.

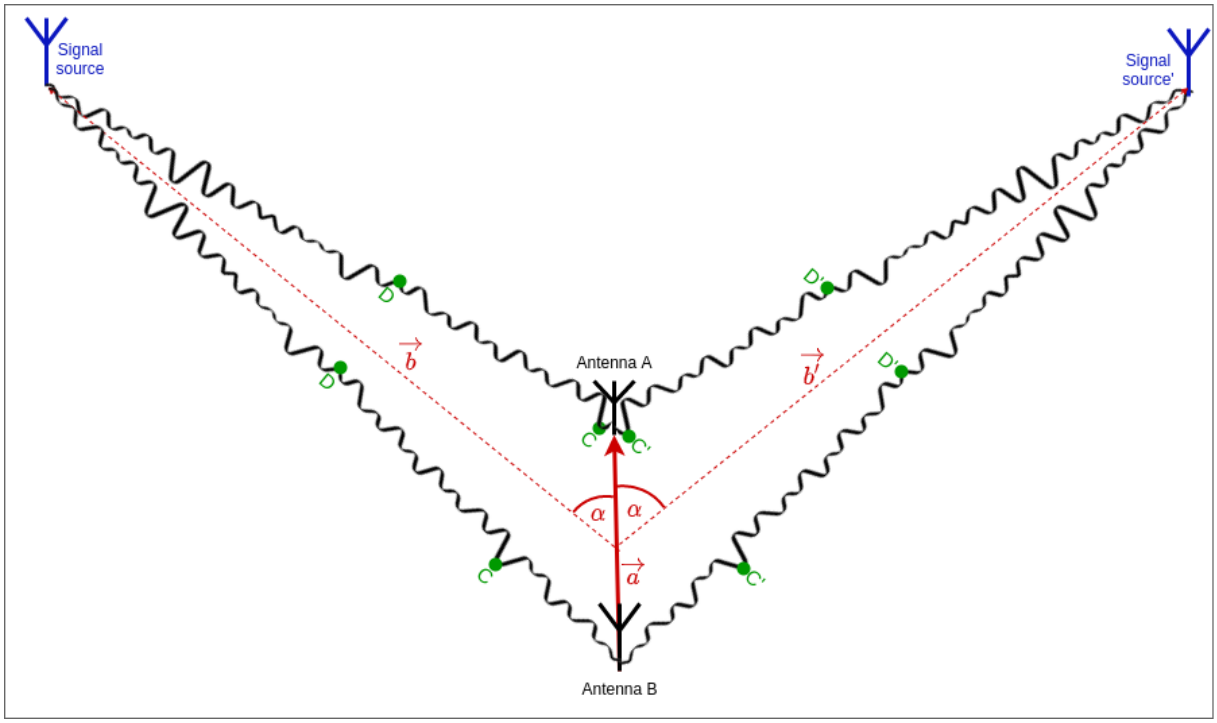

Figure 2.2: Two signal sources with different locations creating signals that reach the two antennas with exactly the same delay because of symmetry.

2.2 Available information

Using the off-the-shelf SDR hardware while considering the cost sets it's limitations on the amount and quality of the available data. There is a HackRF SDR used in the setup of this thesis. Although some of the available platforms can offer somewhat better performance in terms of getting more data, the limitations still conceptually remain.
The most important parameters of HackRF board for this application are the maximum baud rate, sample resolution and operating frequency. The values for the parameters are covered in table 2.1.

Parameter	Numerical value
Operating frequency	1MHz to 6GHz
Sample resolution	8 bits
Sample rate	2 Msps to 20 Msps
Interface	USB 2.0 Hi-speed

Table 2.1: HackRF SDR parameters [1]

The available frequency range is enough to cover any signals that might be interesting for such applications as any higher frequencies can not cover distances that would make localisation of

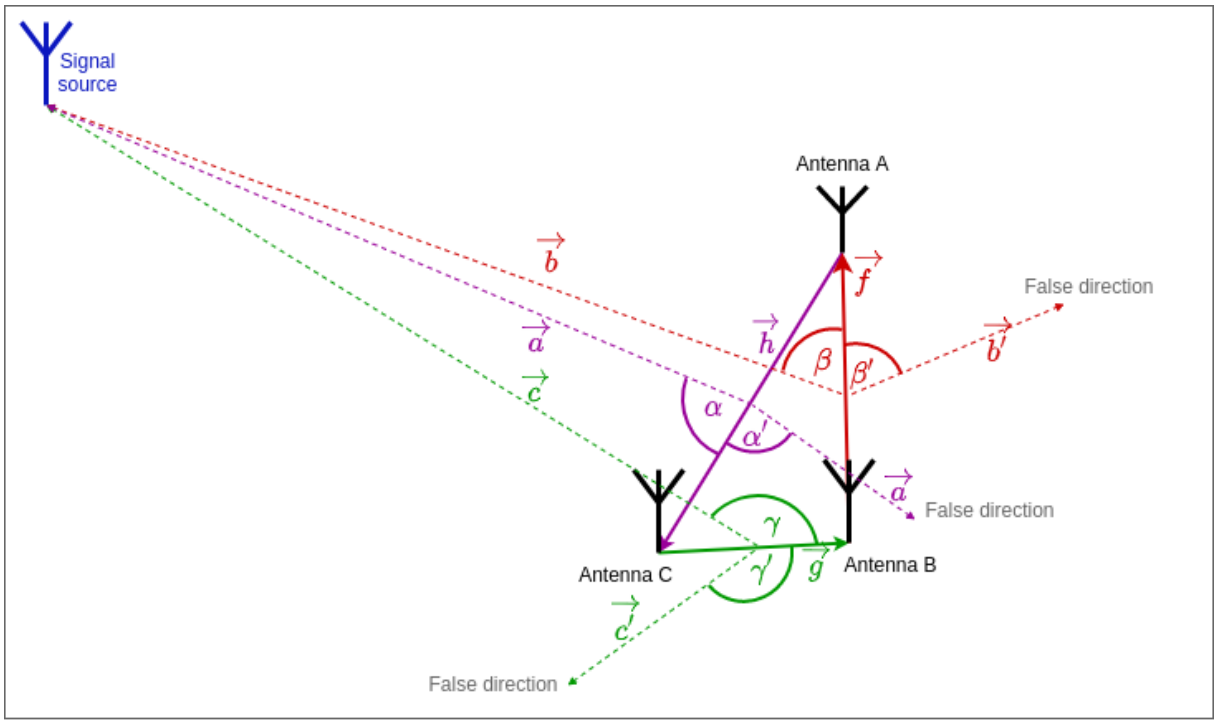

Figure 2.3: Using three antennas to get extra information and combining the information to disambiguate the direction.

the signal source using the measurement method covered here reasonable nor possible because of limitations discussed later. The sampling rate is limited by the used USB 2.0 protocol. This means, that all of the devices connected, will use this protocol to communicate, consequently when adding 3 of such devices to a same USB controller that need to work in parallel, the theoretical maximum sample rate for each device will be 6 2/3 Msps. To give a buffer for control data, 6 Msps was used throughout this thesis. The received samples are 8bit IQ data, which is received by mixing down the signal within the hardware of the SDR board using a generated LO signal with determined frequency.

2.3 Limitations

Having sample rate limited to 6 Msps sets some limitations on what information can be gathered from the sampled signal. In this case the delay between signals reaching the antennas is usually less than the time between the taken samples. For example it means that if finding the delay in full samples, the smallest detectable time difference would be $1/(6Msps) = 0.17\mu s$. With this time the radio waves cover the distance of $0.17\mu s * c \approx 51m$. However, antenna spacing over this value is not reasonable for any connected system. For this reason a purely real-numbered correlation based approach would not work.

2.4 Available tools

During the project, multiple programming environments and languages were used that are standard when controlling and working with data from SDRs. Therefore the work was mainly done in Python and C++ programming languages and using Gnu Radio Companion (GRC) as the

environment. Additionally, NodeJS and plain JavaScript (JS) with some visualisation libraries were used for a demonstrative user interface development.

GRC is an open-source toolkit that allows to quickly implement SDR projects by providing a visual programming interface as well as implementing common signal processing and visualisation blocks [9]. A block is the main component of the software and several blocks can be arranged on the signal path to make up a flow diagram. The blocks are implemented in either C++ or Python and follow a specific architecture, which can be followed to create custom blocks. During the project covered in this thesis, both existing blocks and self-written or modified blocks implemented in C++ and Python were used. The main architecture of the toolkit is based on a continuous linear data flow, which sets some limitations to what can be implemented easily using this tool. It also has built-in buffer management, which should increase efficiency and therefore provide bigger throughput. On the downside, it makes the flow harder to control and creates bigger delays for more complex flow diagrams as each block has its own input and an output buffer.

The GRC software comes with convenience tools to make custom blocks in C++ and in Python. The tools create a set of template files with some pre-filled options that can be set with the generation tool. Further changes and features can be done manually, but the official documentation does not cover all possible options. However, during this thesis, the architecture of the automatically generated blocks was learned and the necessary files for custom blocks were mostly created manually, bypassing the generation tool. This was mainly because the project required several non-standard blocks. The blocks written entirely by the author were mainly composed in Python and the blocks that were closer to the hardware and were mainly modified used C++ as the programming language.

While the GRC comes with blocks that handle receiving the data into the flow diagram from common RF boards, including HackRF, the lower-level tools that provide the actual communication with the board are needed. For HackRF, these tools are combined into a hackrf-tools package that is also open-source and freely available for use. The package provides tools for receiving and transmitting data, updating the firmware and getting system information from the board. There are also some tools that allow using more complex features that make use of the hardware and firmware capabilities on the HackRF board. During this project, several of those tools were used and some modified tools based on the existing ones were created.

Since the system should be capable of having multiple devices deployed and gathering the data, a server to handle the connections should exist. For proof of concept, a simple server and web application were also written in this thesis. For this NodeJS was used as an increasingly popular framework for back-end development [10] [11]. And to make the received information usable for the operator of the system, a simple graphical user interface (GUI) was made using JS with a couple of libraries. Leaflet.js is a widespread open-source library for displaying data on a map within a web browser [12]. Chart.js allows to display different types of graphs and was also used in this project [13].

3 Proof-of-concept Implementation

3.1 Reasoning

As the final implementation of the system has high complexity, the first step of the project was to build and implement a preliminary hardware and software setup to find out if the chosen hardware and algorithms are capable of acquiring data in a form that can be used for determining the direction. It can also help to figure out which parts are most critical and make any changes more easily than in a finalized system setup that would have different ease-of-use layers added. For this concept, this part can significantly reduce unnecessary work in cases where parts of the system that were expected to work, work differently or do not fit for the direction estimation purpose.

In the proof of concept phase, different parts of the system are tested and best option for each subsection is chosen. These parts of the project include hardware setup, data preprocessing before applying the DOA algorithm, DOA algorithm itself and conversion and presentation of the output direction estimate. In addition it can give an idea of what kind of hardware is needed to do the required data processing or how much optimisation needs to be done to run the system on a predetermined hardware.

3.2 Approach

For fast testing of the effects of each of the building blocks of the solution, GRC graphical IDE is used. This allows to test different architectures quickly and switch between real and generated data as needed. The ease of creating user interfaces for changing parameters during run-time helps to find out how different coefficients and types change the results.

The testing was started with lowest possible complexity and set of algorithms. Features were added if the lower level solution showed that the output is feasible.

3.3 Delay effect with two antennas

The implementation started by testing out if the effect described in the physical principle section is detectable using the best hardware chosen by theoretical capabilities and price. The test setup consisted of two antennas that were measurable distance from each other and a single off-the-shelf handheld radio as a signal source. For this test to be successful, the following conditions have to be true:

- If the signal source is stationary, then the output should stay constant with a reasonable tolerance.

- Output angle of a simplest algorithm should show some change when the angle of the signal source relative to the antennas changes.
- If the signal source is returned to the original position after being moved away, the output should be similar to the initial result.
- There should be correspondence in at least some ranges between the direction of the source movement and the output.

The chosen hardware setup had a separate hackRF SDR for each of the antennas. The ADCs of the two devices were synchronised according to a manual [14]. This synchronization process included making physical connections of the trigger input and flashing a special hardware to the hackRF boards that was not yet included in the official release as of the time writing this thesis. The devices were connected to a computer where the data was processed in real time. The simplest algorithm that was chosen for this step was phase-based calculation to find out one of the possible angles. Choosing the correct angle from possible results or having all the possibilities calculated were not necessary in this implementation step. As the algorithm is quite simple there was no preprocessing needed and all of the incoming data could be processed in real time. The results were shown on a histogram graph with 360 bins, which will give a reasonable way of visualizing both the most common angle with requested accuracy as well as the deviation of the results. This program was achieved by using only the built in blocks of GRC.

The sub-result and output of the implemented program were first tested using two Agilent (Keysight) 33250A signal generators to provide the best-case signal of realistic frequency. The signal generators feature synchronisation option between multiple devices, allowing to create multiple signals with constant and controlled phase shift between them [15]. The setup for the system is shown in figure 3.1. This test allowed to check if mixing down the signal by the built-in RF mixer of the SDR platform causes any distortion to the phase and if the algorithms were implemented without mistakes [16]. The results of this test confirmed that the phase difference information received as the intermediate output from the system was in correspondence with the original RF phase difference between the two input signals. The initial test also showed that despite having the hackRF boards synchronized according to the manual, there were some distortions from the expected synchronization.

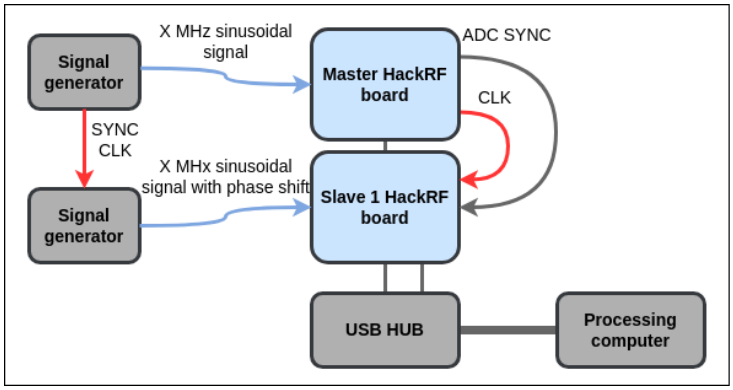

Figure 3.1: Test setup for verifying the correct delay between two signals can be measured using two HackRF boards.

Firstly, the LO frequency, that was common for all of the synchronized devices drifted over time. This does not cause severe distortions in comparing the signals, but can become relevant when requesting data for longer periods as the intermediate frequency of the signal can change over time and that in turn can decrease the performance of the algorithm if constant signal frequency is assumed. The problem was however not considered as severe and it was possible to compensate in software so no hardware modifications or additions were necessary.

The second and more severe issue was that the LO signals of the individual devices were synchronized within one run of the program but between different launches the phase relations changed. While the PLL implemented on the LO generation chip kept the phase differences that were initially present on startup until the program was running with same center frequency, whenever the LO generation was cancelled or different frequency was requested the phase differences were randomized again. As proven in the first test, the effect of phase difference in one input signal to the RF mixer is directly carried over to the IF signal. Therefore on each launch of the DOA program, the phase difference measured by the system consists of the actual difference and some random difference. This effect significantly complicates the use of the system as the methodology is based on finding the exact absolute delay in signals and addition of random value loses the base on which the absolute difference is based. As no reliable way of getting over this problem purely in software was found, the chosen method was to add additional simple hardware to create a way of determining the phase shifts in the system. Overview of the added hardware and necessary steps for making use of it are covered in section 3.4.

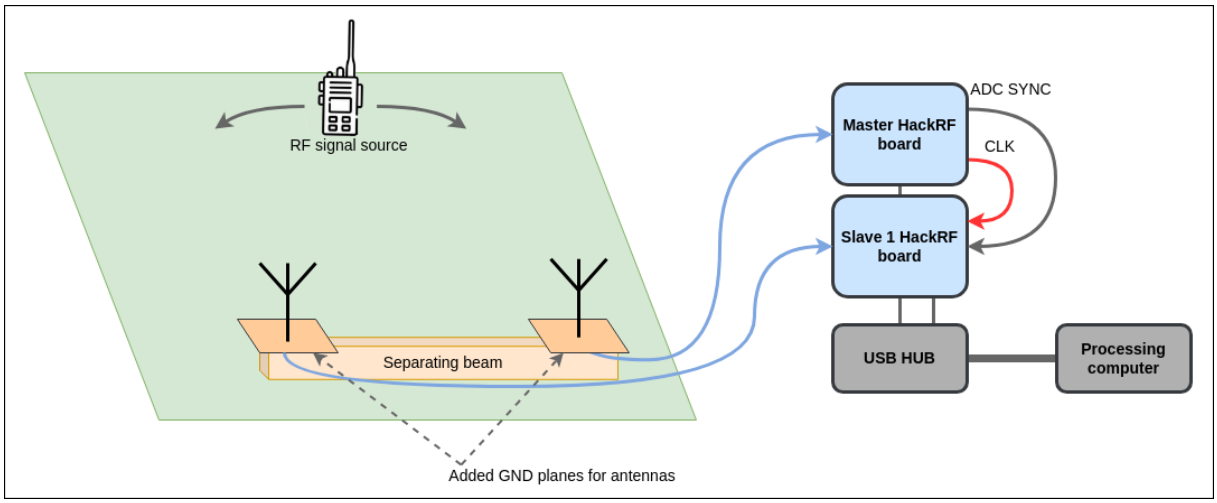

Figure 3.2: Test setup with 2 antennas on a field with fixed gap between them. Simple handheld walkie-talkie is used as a movable signals source.

The system was also tested using real RF signal over air, that was sent using simple transmitter and captured using two antennas that were fixed to set distance from each other using non-conductive base. The test was carried out outside with no reflective objects nearby. The setup is also shown in figure 3.2. It was seen, that a signal from a very well defined spot stayed constant, filling the first requirement. It was also shown, that within a consistent distance and in certain sectors, the peak of phase shift on the histogram graph moved in one direction. The sectors, where the correlation between signal source movement and the histogram peak movement were the best were the ones where the source was located to the side of the antennas. The output in

these areas could jump to a random angle even on the smallest change of signal source position. This showed that the system was very sensitive to the effects that the antennas had for each other. Section 3.5 covers some approaches taken to reduce these effects.

3.4 LO synchronization using calibration signal

For the system to be able to tell the exact delay of the signal reaching each of the antennas, it was necessary to know or consider the LO signals precisely. And since the LO signals were generated within the hardware of the HackRF, there was no simple way of making them to have pre-determined phase relations with each other that would stay constant over several launches of the program without modifying the hardware. The other option was to figure out the phase relations on each launch and use that information to analytically remove that effect from the signal or results. Since no other options were available with the chosen hardware, this option was chosen.

To implement the solution, a calibration signal that is the same for all of the devices was needed. This would allow to get the inputs from all of the devices and with the phase of the LO signals being the only difference between the sampled signals, it would be possible to figure out the phase shifts. To generate the signal, simple external hardware was used, which served two tasks. First, it was able to generate random noise and match it to 3 different outputs, that could be connected to the signal inputs of the SDRs. Secondly, it provided a way to switch whether the inputs were connected to the noise generator or to the receiving antennas of the system. The hardware solution for the module was not created by the author of this thesis, but the author contributed to creating the requirements and carrying out the testing. The principle of the module is shown on figure 3.3.

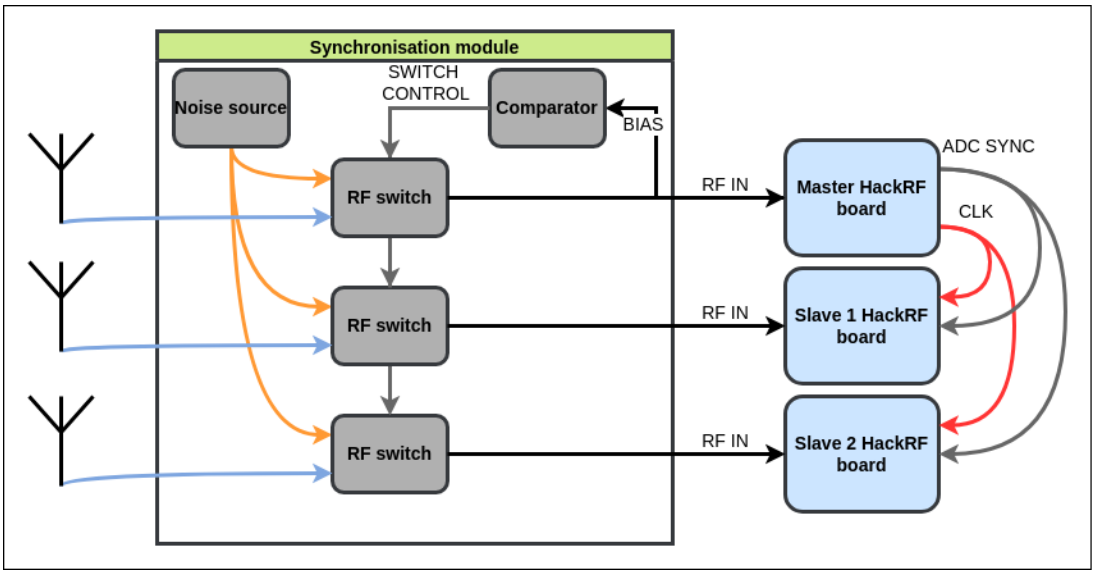

Figure 3.3: Extra connections between used hackRF boards and added LO synchronisation module for synchronous operation of three HackRF boards.

Using the signal generator setup once again, it was determined, that the LO phase shifts were

successfully determined, using the module. The test showed that when considering the phase shift that was measured based on the input from the noise generator, the phase difference stayed the same over different launches, when the phase difference of the input signals was constant. The correspondence that was present in the previous setup also remained.

There are also some drawbacks to using this methodology for determining the LO phase shifts. First, it takes extra time to acquire the required data in the beginning of each measurement. This reduces the ability of making measurements with the system, where quick reaction to detected signal or fast changes of detectable frequency are needed. Secondly, since it is another separately conducted measurement, it introduces another source of error, that will add up with the measurement error of the actual signal. The more calibration data acquired, the smaller the error will be, but this would increase the time needed, which increases the previously discussed speed restriction. Finally, it increases the system cost and complexity, which is one of very important considerations for the project.

3.5 Physical setup improvement

As discussed in section 3.3, the output from the simple test system was correct when the input signal to the devices was clean and without distortions (like signal from signal generator), but the quality of the physical setup of the antennas had a very important role in this assumption being true. Therefore it was important to determine, which factors of the setup had the biggest effect on the quality of the important parameters of input signal.

The first test where the issue was encountered included 2 antennas that were fixed on a wooden frame. It is known that two antennas that are close together can affect the received signal of one another depending on the possible paths relative to the receivers. The induced current by one receiving antenna effects the parameters of others nearby. As a general rule, this effect, called mutual coupling is recognisable when antennas are closer to each other than the signals wavelength or the length of antennas. Without any proper counterpoise this effect is magnified and other distortion effects are present. [17] [18]

For this reason, a low-resistance connection to ground or counterpoise extending at least half of the received signal wavelength should be present for monopole antennas [19]. In this work a copper ground plane (can be seen in figure 3.2) was used in initial testing setup for each antenna and for final tests the antennas were fastened to a single large metal object instead of separate smaller ones. These improvements reduced the unwanted effects significantly.

3.6 Combining the results of 3 antennas

To combine the results of several pairs of antennas there are two general approaches. It is possible to calculate the directions or distributions separately for each of the pair and use methods to combine them or to use calculations that based on the locations of all antennas give a single output. During this thesis, mainly the first approach was used as it provided more intermediate data for analysing and troubleshooting. The algorithms for direct combined result were mainly tested in the finalising phase of the development and as a direct comparison in the analysis.

Combining the results can also be done differently, depending on the desired output. It is pos-

sible to separate the peak angles and their certainties from each of the pairs separately or to combine the distributions into a single output distribution in a way that the peak of the resulting distribution is highest where the individual distributions all have high values.

During the thesis both methods of combining the results were tested, but method combining the probability distributions was finally used. For this, all of the separate distributions from all of the 3 pairs of antennas were normalised to have a sum of 1. Then all of the probability distributions were element-wise multiplied to have high values only where all of the distributions have a high value. This method allowed the overall estimate to also have a reasonable probability rating to it, that is theoretically limited to be between 0 and 1.

3.7 Final POC software setup

The proof-of-concept testing resulted in a program implemented in GRC with a user interface (UI) to configure the system and see the results. The data flow consisted of two paths with one for synchronisation purposes and one for the DOA calculation. Both of the signal paths passed a frequency translator and a decimating filter to reduce the amount of data to process only the necessary band of the captured spectrum. The synchronisation path had a complex correlation applied between each two pairs of signals out of which only the resulting phase was used for the reasons provided earlier. The phase shift was then shown to the end user on a histogram graph with 360 bins of size 1 degree. The DOA calculation path had a 2-antenna based DOA algorithm applied to each pair of antennas. Applying the algorithm separately was necessary as the user was then able to customize all of the parameters and see the results. The results of those algorithms were then combined and shown on a line graph. The simplified flow diagram is shown on Figure 3.4 and the full diagram can be seen in Appendix A.

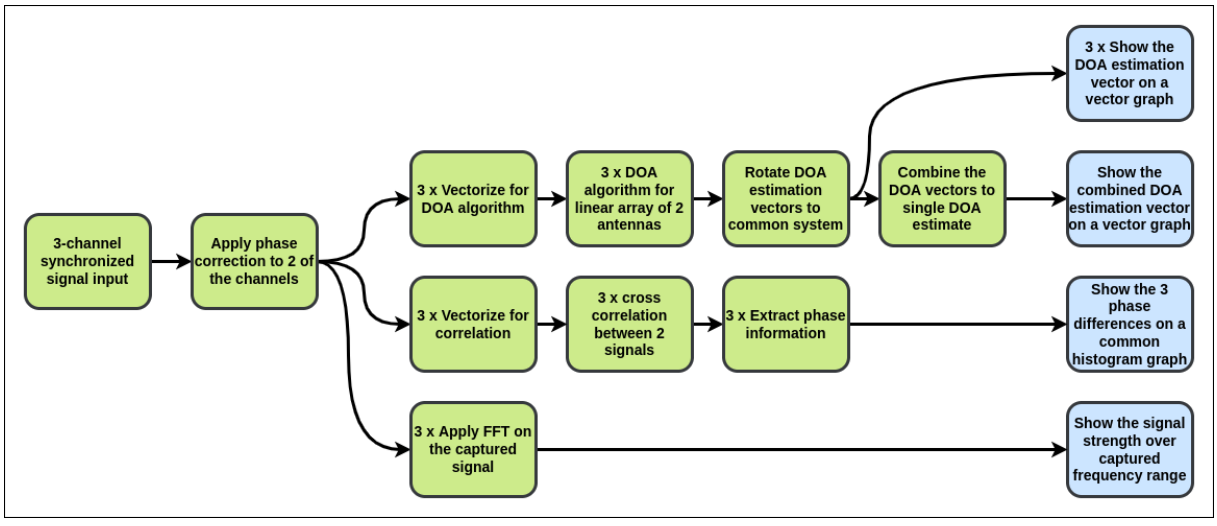

Figure 3.4: General data flow of POC DOA software. Steps taken to show user the necessary information to operate and get a reading from the program.

The UI allowed the user to configure a set of different parameters during run-time. There were also many configuration variables that had to be set in the flow diagram prior to executing it.

Table 3.7 explains all of the used parameters. For the ease of use, the UI was separated into 4 different tabs for phase calibration, signal view with gain setup and two tabs for DOA estimation with antenna array parameter configuration. The 4 tabs are shown in Appendix B.
During implementing the prototype software, several custom blocks were written for GRC. The following gives an overview of these blocks.

- Phase Correction block. Used to apply a user-specified phase shift to data samples that pass through the block. In this project the main use case for the block was removing any system-introduced phase shifts from the signals. Internally uses mathematical operation of multiplying complex number with a complex number with a magnitude of 1 and phase of the required phase shift.

- DOA block for 2 signals from 2 antennas. Wraps the DOA algorithms from Pyargus library to be used in GRC [20]. Internally uses the methods provided by the library to create a linear array of two antennas with the given spacing parameter. The spacing is given as a multiple of wavelength. Additionally the block takes the length of vectors used, the algorithm that is used and the update rate as arguments. The inputs to the block are vectors of received data of given length. The output is a vector of 360 points representing the DOA probability distribution relative to the given pair of antennas. The output is recalculated only after certain amount of inputs that can be determined by the user, to allow real-time processing.

- Complex cross-correlation block for 2 signals. This block takes two vectors of requested size and performs a complex-numbered cross-correlation on those with a given window size. If the window size is smaller than the full vector, then the output consists also values sample-wise. If the two sizes match the output is a single complex correlation value, that has a phase corresponding to the phase difference between these two signals.

- DOA estimation array rotation block. This block takes all 3 DOA estimation vectors as input signals and has a rotation parameter for each. The block rotates each of the arrays by the amount of elements specified by the parameter and gives the resulting 3 vectors as output. It is used to get the vectors from a direction that is relative to the two antennas making up each pair to a direction relative to the common coordinate system.

There were also several blocks developed during the prototype development that were not used in the final setup as they did not provide required performance. Some of those blocks are described in the following list.

- As an addition to the phase correction block, a delay block was first tested. The idea of the delay lock was to keep samples one signal in a buffer of determined size to delay it relative to another signal. As the block was 1 to 1 block in GRC, the first output samples for the delayed signal were zeros. The benefit of such method would be that it would work even with signal that have more than one full period of delay, without creating false effects on the signal. Since phase correction only modifies the phase, it does not account for any other changes in signal that are also delayed. But since the effects other than phase were determined to be unusable, then there was no need for this block. The block is also limited to delaying the signal by full samples, which is not the case for the actual delay, which for reasonable frequency and antenna distance ratio does not occur.

- DOA block that allowed 3 input signals with any placement of 3 antennas. This block was built on top of the corresponding method in the Pyargus library, which allowed to input

the antenna positions as list of X and Y coordinates. The method was not used in the final POC software as it was hard to configure all of the antenna distances and angles at once, without receiving separate feedback from each pair. In the end result this operation can be added as it provides better estimate, since the combination step is included in the likelihood based estimate.

- DOA estimation based on the phase difference. This block allowed to calculate the direction based on pure phase difference information. The block considered all of the possibilities with given frequencies by adding all of the possible unwrapped options of the measured phase difference. The result was given as a probability vector with added circular normal distributions for each possibility. The deviation for all of the possible angles was considered constant. The negative side of the block compared to later used DOA algorithms was that it did not consider that the certainty of a measured angle can be different depending on the DOA angle itself or the quality of the signal.

- FFT based phase finder. This algorithm used the phase information of a frequency domain bin that corresponded to the required frequency. For reducing the complexity, only the required frequency component was calculated. Although this block did output a correct and also well averaged phase, it provided similar amount of information and had similar complexity with the cross-correlation block, the cross-correlation block was used in this case.

Parameter	Configure	Example value	Explanation
Device ID	Before execution	1	Determines the set of IDs of the used HackRF boards. It is necessary to know the correct IDs as the devices have different roles in the synchronization setup.
Sample Rate	Before execution	2 MHz	The sample rate of the ADC on the HackRF board. The sample rate is limited by the throughput of the USB 2 protocol.
Signal Frequency	Before execution	51.025 MHz	The frequency of the signal that is to be measured. The center frequency (LO frequency of HackRF) is determined by this value and the frequency shift.
Frequency shift	Before execution	450 kHz	The offset between the signal frequency and center frequency. It is necessary to offset the signal to reduce the effect of the LO leakage on the important signal. The center frequency is chosen to be lower than the frequency of signal of interest by the amount of this parameter.
Decimation factor	Before execution	10	The factor by which the amount of samples is reduced after the signal is translated to 0 frequency and filtered. Bigger factor will reduce the need for processing power but provide less quality.
Nr of samples for DOA	Before execution	1024	The amount of samples that are used to provide one DOA angle estimate. Bigger amount will provide better accuracy but will require more processing power.
Nr of samples for correlation	Before execution	256	The amount of samples that are used to provide one phase shift estimate. Bigger amount will provide better accuracy but will require more processing power.
Phase fix	Run-time	0.5 rad	This parameter is applied for each signal separately. The phase that is added to the signals before any other parts of the processing take place. It is used as a counter effect the phase change caused by LO phase differences and by slight differences in the lengths of the cables.
Distance between antennas	Run-time	1.25 m	This parameter is applied to each pair of signals separately. Determines the assumed distance between antennas used in the calculations.
Angle of antennas	Run-time	0.5 rad	This parameter is applied to each pair of signals separately. Determines the assumed distance angle of the pair of antennas in the used coordinate system that is used in the calculations.
HackRF gains	Run-time		Allows to set the RF, IF and BB gains of all of the HackRF devices. The gains can be set to make the maximum use of available range without saturation. Allows to use the system with input signals of different power.

Table 3.1: Parameters configurable in the POC version of the implemented DOA system.

4 Analysis

4.1 Analysis overview

The prototype system that was created and proved to give some estimation about the direction had shown to work differently with different setups and parameters that was not fully understood by just seeing the output. To figure out exactly, a system analysis were carried out, paying more attention to the aspects that seemed to affect the system output in unexpected manner. The analysis were done both on simulated data and on real-life data to separate the effects introduced by the system from the effects that were the results of bad data. New steps of analysis were added as the outputs of a step required further investigation. In the following sections the main components of the analysis that were carried out, are discussed.

4.2 Simulation of data

Using the real data from sensors can introduce error into the system from several sources during the data acquisition steps. Also, the expected result is not so well defined as it depends on the accuracy of the physical setup and environmental details that can affect that. This is a reason, why in the analysis step there was also simulated data used to verify the results in the case of ideal data input. This allows to find out the importance of any mathematical errors during the approximated calculations and filtering. Those errors can then be compared between different algorithms and parameters. When modifying the algorithm in any ways, the ability to generate and use simulated data also allows to confirm that there were no errors made, in a more simple way.

To make the simulated data as useful as possible, the form of the data has to as precisely as possible resemble the real data. In the current case the simulation should be capable of calculating the signals reaching the antennas depending on the input antenna configuration and signal source location. In addition it has to allow imitating the random nature of LO phase differences between the 3 nodes and generation of calibration signal based on these differences. Following signal reception, mixing down the data should be imitated in IQ format. Following that, the sampling as 8-bit format with requested rate should be possible. As an extra feature the simulation can have the option to input different kinds of noise into the system in different steps of the process. [16]

When simulating the data, the steps until sampling are done with significantly higher sample rate to as closely as possible resemble the continuous process of the RF circuitry on the actual hardware. The results are later in the sampling step downsampled to get only the values on necessary time points.

The inputs to the simulator program are:

- coordinates for all of the antennas in a 2-dimensional coordinate system (x_{a1}, y_{a1}), (x_{a2}, y_{a2}), (x_{a3}, y_{a3}).
- coordinates of the signal source in the same coordinate system as the antennas (x_s, y_s). Alternatively the angle of the signal source in the coordinate system as absolute angle α. In this case the distance is assumed to be constant 10km from the origin of the coordinate system. This also means that for the simulation to be accurate, the antennas should be located around the origin.
- LO frequency f_lo. This corresponds to the center frequency set on the SDR.
- input signal parameters. This can be a single frequency for constant sinusoidal signal or list of frequencies along with their duration $(f_1, f_2, \dots)$, $(t_1, t_2, \dots)$. This approach allowed to simulate digital frequency modulation schemes. In this case the function can be summarized as:

$$f(t) = \begin{cases} sin(f_1 t) & t \mod (t_1 + t_2 + \dots) < t_1 \\ sin(f_2 t) & t_1 < t \mod (t_1 + t_2 + \dots) < t_2 \\ \dots \end{cases} \tag{4.1}$$

- sample rate f_s
- number of samples for calibration signal n.
- number of samples for the signal to find the DOA of N.
- multiplier of how many times bigger the sample rate is before downsampling K.

The program outputs a file with 8-bit IQ data for all of the 3 antennas in interleaved format. Saving to a file is done instead of processing it immediately mainly for two reasons. It allows to more easily use the same data while configuring the processing parameters and it makes possible to use the simulated data by the previously implemented POC software in GRC. The architecture of simulation compared to real setup scenario is shown in figure 4.1.

As the first operation, the simulation program calculates the distances for each antenna to the signal source. As only the differences of the distances are important then these are found relative to the first antenna $a1$. The corresponding differences are marked as d_2 and d_3. If the signal on the antenna a_1 is represented as $y_{a1} = f(t)$, then the signal for antenna A_2 is $y_{a1} = f(t - d_2)$ and for antenna A_3 is $y_{a1} = f(t - d_3)$.

Next, the signal is mixed down using the LO frequency and only the lower frequency signal is let through a filter in the actual hardware setup. Mixing the signal with LO is also the step, where the data becomes complex, by being mixed with LO and 90 degree phase shifted LO. Those two signals are calculated using sine and cosine functions in the simulation program. The mixing step itself is done using its mathematical representation of multiplying two signals. In the simulated discrete domain it is done as an element-wise operation. The mixed down IQ signal then is calculated in the following way: $I_{an} = y_{an} * cos(f_{LO} t)$, $Q_{an} = y_{an} * sin(f_{LO} t)$. The result can be represented as a complex number of $I_{an} + j * Q_{an}$. The filtering is done using a Butterworth filter as there is no great way of simulating an analog filter in discrete calculations and it gave a reasonably good result on simulated data as there are only two frequency

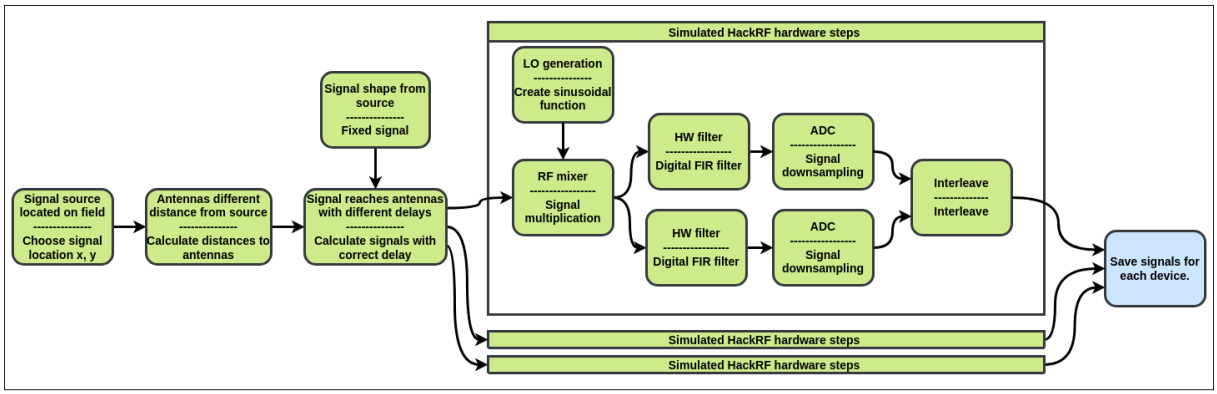

Figure 4.1: The steps taking place for signal acquisition in real (shown above separation lines) and simulated (shown below separation lines) scenario.

components in the signal at that moment out of which one has to be suppressed.

In the sampling step, every K-th element is kept, since the initial sampling rate was chosen to be K times greater than the desired output sample rate. The results are converted to 8 bit integers, which is the bit depth of the ADC on the chosen SDR HackRF2. The values of the 3 antennas are then interleaved and written to corresponding file.

4.3 Analysis on simulated data

Simulating the data can give information how different steps in the data manipulation process effect the quality of the data by allowing to monitor the results with different system configurations. However, for this method needs a way of evaluating the correctness and quality of the results. In this analysis two measures were used to measure the accuracy of the result in different processing phases.

First measurement used was the deviation in the signal phase differences. When creating the simulated data, the exact phase difference of the generated RF signals can be calculated and this can be compared with corresponding phase differences of the resulting signals between all of the preprocessing steps. The benefits of this evaluation method include the ability to monitor the accumulative error between each step and ability to see the possible error of using pure phase-shift based approach for finding the required phase correction. Disadvantages of the system are that it is sensitive to the effects of modulation and does not directly reflect the error of the end result. Also, this result does not allow to compare the error from the simulated results with the error from real data, as in real data, the true phase difference between input signals cannot be measured as precisely as it can be calculated on the simulated data.

Another evaluation metric that was used, was calculated based on the final output of the MUSIC algorithm, that was seen to perform best in the prototyping step. MUSIC is an algorithm which has DOA estimation as one of its use cases. It is based on eigen decomposition of the received signals' correlation matrix, assuming presence of uncorrelated noise [21]. The better performance compared to other tested maximum likelihood methods comes from the predetermined amount of signal sources, which reduces the effect of spectral noise in the simple DOA estimation [22]. The DOA use case of MUSIC algorithm in the Pyargus library calculates the

DOA distribution for given number of bins. From the normalized version of the distribution, it is possible to see the probability, that the correct angle has and also what is the probability of the most probable angle according to the algorithm. The latter can be useful in real signal cases, where the true angle is not precisely measurable, by still providing the certainty information. Although this method does not allow to monitor which processing step introduced the biggest error, it can still be used to compare the end result by seeing the effect, when using different parameters. It can be also directly compared with the results achieved from real-life data.

In this section those two methods are used side by side to visualize different effects the system setup, data and taken preprocessing steps practically and theoretically have on the result.

4.3.1 Probability from different angles

As discussed in the physical concept section, the change in the signal delay between two antennas is based on the different distance from the signal source to those two antennas. The change in this distance however is not constant for all signal angles around the system. For the same change in DOA angle, the change in signal delay is larger when the signal source is to the side of the antennas and smaller when the signal source is closer to being in line with the two antennas. The change in the delay for 1 degree of change in DOA angle is shown on figure **4.2**.

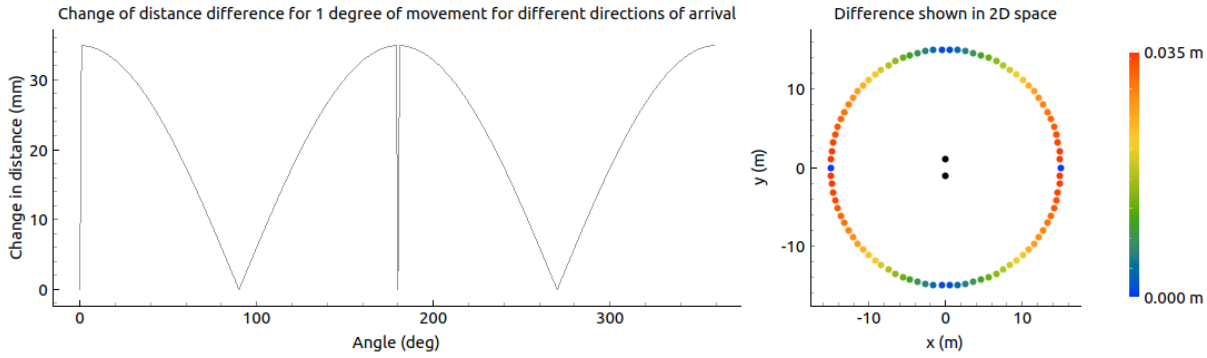

Figure 4.2: The change of the distance from the signal source to two antennas when the signal source moves by one degree, for different locations relative to the antennas. The graphs show the absolute change in distance when the signal source moves from x-0.5 degree to x+0.5 degree spot with antenna spacing of 1.5 meters and signal source located 100m from the origin. On the right the same information is represented in 2D space with antennas marked as black dots.

This concept also carries over to the phase difference, which changes faster when the delay changes faster with linear relation to the distance change. For this reason, the phase difference can be compared to the ideal sinusoidal function that can be calculated from the setup parameters. However the phase difference can be wrapped and therefore has to be unwrapped before the analysis can be done.

To see how this concept effects the results, when using the MUSIC algorithm, the algorithm was first tested with the same setup. The probabilities of the correct angle were gathered with a step of one degree. Results can be seen on figure **4.3**. It can be seen, that in general the certainty

of the correct angle is lower in the areas where the change in distance was also lower. This is because the larger deviation caused by the similarity of results from close angles.

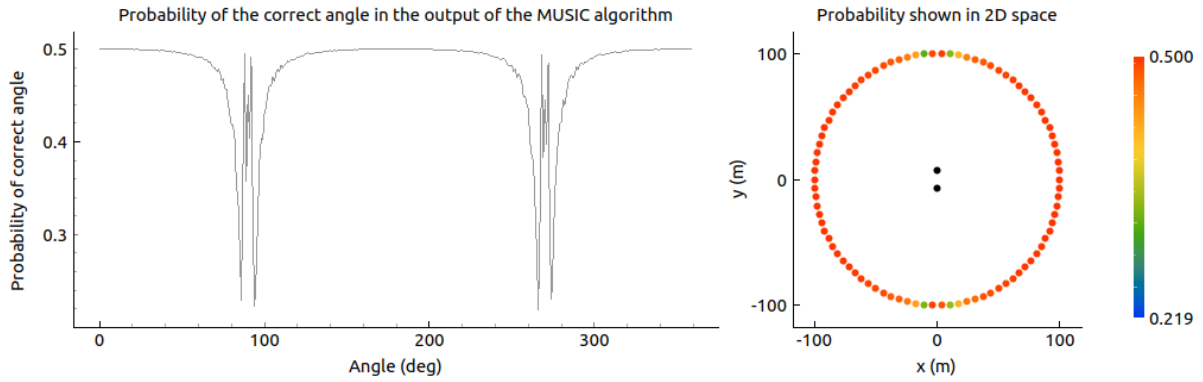

Figure 4.3: The probability of the correct angle in the output of the MUSIC algorithm with two antennas spaced 2 meters from each other and signal source located 100m from the origin. On the right the same information is represented in 2D space with antennas marked as black dots.

For this reason, the placement of the additional antenna is important. The ideal placement would create a uniform detection certainty across all possible angles by having the separate pairs of antennas have their directions of better detection supplement each other. Although an ideal setup is hard to achieve and is not the goal of this thesis, some simple placements were tested. The first tested setup had the antennas in an L-shape with unequal sides. The L-shape creates two main pairs with the strong part of one pair directly covering the weak part of another. The third pair adds some certainty to everything. The results can be seen on figure 4.4.

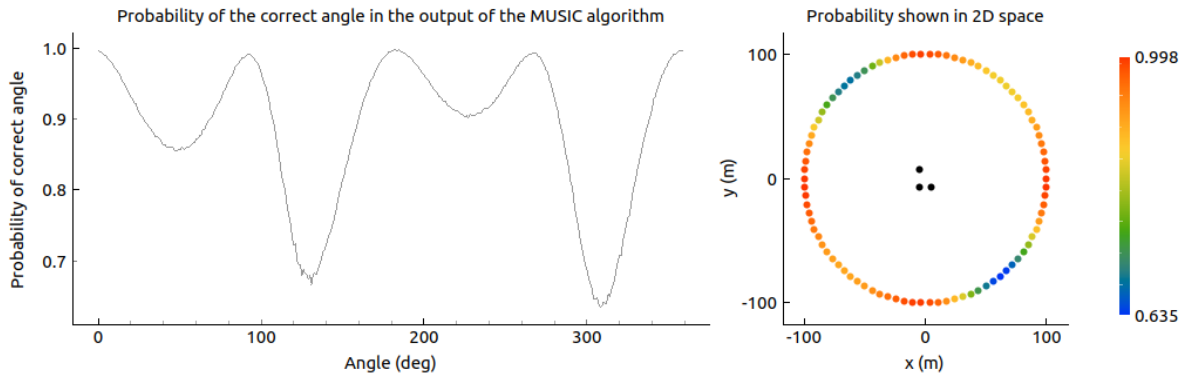

Figure 4.4: The probability of the correct angle in the output of the MUSIC algorithm with three antennas spaced in the vertices of a right angle triangle and signal source located 100m from the origin. On the right the same information is represented in 2D space with antennas marked as black dots.

It can be seen, that adding the third antenna increased the average certainty significantly, as the probability is no more equally distributed between two possible symmetrically equivalent di-

rections. However, there distribution is not very uniform, with some areas having significantly lower certainty values than others. These areas are located in places that are least covered by the sides of antenna pairs. In the current case the difference in the probabilities is around 36 %. Approach, which should create the best coverage is to put all of the antenna pairs at equal angle spacing. That is achieved by having the antennas located at the vertices of equilateral triangle as the sides all have 60 degree difference between them. The results of this placement can be seen on figure 4.5.

Since there is still limited amount of 3 antennas, the probability distribution is not ideal, but the range of change in probabilities for different areas has decreased significantly, being now only about 10 %. The lowest probability has also increased by 18 %.

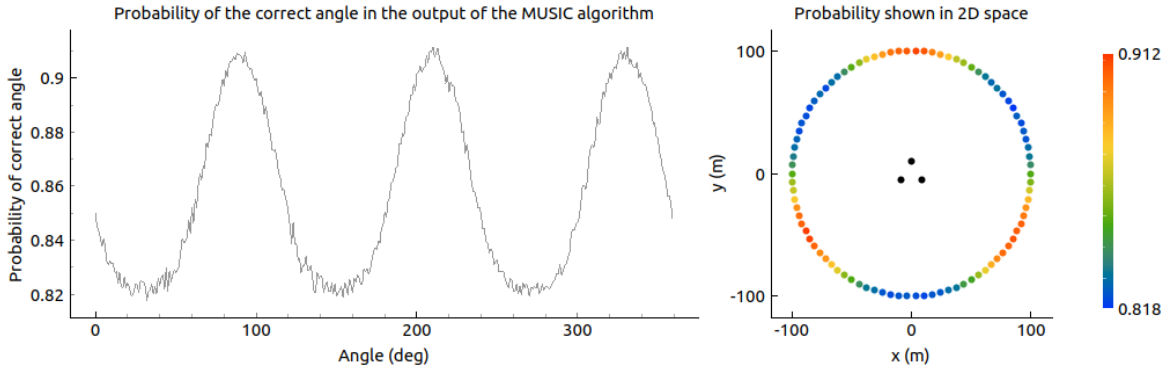

Figure 4.5: The probability of the correct angle in the output of the MUSIC algorithm with three antennas spaced in the vertices of a equilateral triangle and signal source located 100m from the origin. On the right the same information is represented in 2D space with antennas marked as black dots.

4.3.2 Other tests

As mentioned, the simulation program was also used for other tests. For example, the effect of the distance the signal source is from the receiving DOA estimation setup was tested. The test verified, that from any reasonable distance for which such system can be used (more than 100 m away) and for antenna spacing of less than 3 m apart the effect of the signal source not being infinitely far away was smaller or in same scale than other noise introduced by the system. This was confirmed to be true for both the effect on the phase difference and the output of DOA algorithm.

4.4 Acquiring real-life data for analysis

To analyze the quality and errors of the developed system as thoroughly as possible, a decent dataset needed to be collected using a setup that would imitate the system use. To gather as much output data as possible while still keeping it in as raw format as possible for various analyzes, a special program was developed to save the required data with minimal user input. This kind of program setup would allow different setups to be tested when using this kind of DOA approach in the future as well.

The main part of the acquisition program consisted of the system finding the signal phase difference in the correlation method as described in the X section. The additions include an added user interface for choosing the ground truth for the signal source angle and a button to save the program output for a determined time period from the time of the button press. This kind of approach allowed to be sure that the signal was available from the desired angle before doing the saving operation. For this a separate custom block was also written for the GRC that managed the user input and saved the correct amount of samples to a file with necessary meta-information to read the data later for analysis.

During the work of this thesis, one dataset for 30 MHz to 88 MHz was collected using this acquisition method. The setup consisted of 3 antennas located in 3 corners of a common metal object to simulate one possible realistic setup. The origin of the coordinate system was chosen between all of the antennas. The physical distances between the antennas were: 3.3 m, 3.5m and 1.2m, forming approximately a right angle triangle. There were in total 26 measurements made for each frequency: 2 calibration signals and all possible angles with a step size of 15 degrees. The distance from the origin to the signal source was approximately 25 meters. This distance was chosen as the angle caused by the differences in antenna positions is negligible compared to other noise present in the system. Taking the signal source even further would also possibly increase the error because of a weaker signal. However, it can also make the angle more accurate and reduce signal interference between antennas. Since previous tests of just taking signal source further away showed those effects cancelling out each other, the smaller distance was chosen for convenience, allowing to gather more data points with the same use of resources. The output consists of 32 files and the analyzes based on those are described in the next section.

4.5 Analysis of errors in real-life data

As the saved data from the acquisition program considers phase differences, while the antenna spacing allowed phase differences to overflow for higher frequencies, the data had to be preprocessed to get the absolute delays. This step was partially automated, with user interface to visualize and handle the results, while manually determining the places where the phase difference had been wrapped. The program allowed to choose ranges where the phase had wrapped 1 time, 2 time and so on. The output was then modified by adding corresponding amount of full periods in radians to the input. A sample unwrapping, done using the created interface can be seen on figure 4.6. The user interface can be seen in appendix D.

The processed data then characterizes the delays for each frequency from all directions for each pair of antennas. This allowed to evaluate the error compared to the expected delays function and errors compared to systematic changes, where system description parameters are modified so that the average error would be minimal. The latter was important because it allowed to estimate any systematic deviations from physical parameters, that depend on the frequency. For this analysis Python program was written to extract the necessary information from the data files.

One approach was to visualize the collected results with the expected results as comparison and calculate maximum and average errors that the system could make by having the measured errors in the detected delays. Other approach finds the best system parameters to correspond to

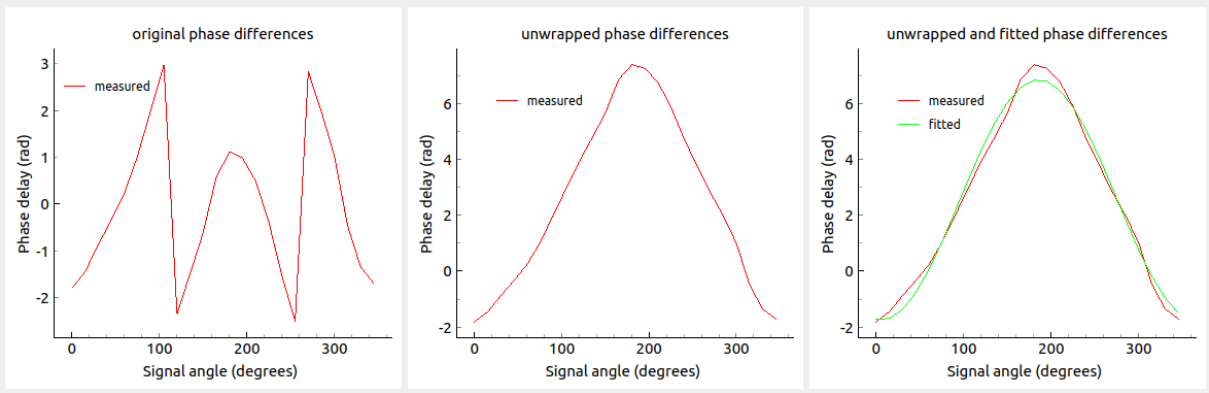

Figure 4.6: The unwrapping and fitting steps of analysing the measured real-life phase differences. The original phase differences of one pair of antennas over different DOA angles (left graph), the unwrapped version of the same differences (center graph) and best fit of sinusoidal function for the unwrapped signal (right graph).

the measured delays and compares them to the physical measurements over frequencies. The best parameters were selected using least-square method on the unwrapped phase differences. The fitted data can be seen in the rightmost graph on figure 4.6 The full amplitude of the approximated signal gives the information about antenna spacing as bigger spacing would give greater change in phase difference when signal source moves from a position that is on line with the antennas to a position next to the antennas. The phase of the approximated sinusoidal function gives the angle of the vector that connects the two antenna locations relative to the coordinate system. Finally, the difference between the calibration signal and the signal from a remote source gives the delay in cables. The correspondence between the parameters of the fitted sinusoidal function parameters and the antenna placement options are shown on figure 4.7.

For the data that was gathered during this thesis, the results are as shown on figure **4.8**, figure **4.9** and figure **4.10**.
The analysis shows that the virtual distance between the antennas that best corresponds to the received data is decreasing when the frequency of the signal of interest increases. With the setup for this measurement, the change in distance reached up to 0.7 m for antenna spacing of 3.5 m in the tested frequency range of 60MHz. This is a significant change and can introduce big error into the system. Therefore, as a minimum option, a multiplier for the antenna distances should be added to the setup, based on the current frequency. From the results here, it seems that a constant assuming linear change would reduce most of the error. The corresponding linear approximations are also shown on figure **4.8** with dotted lines. For a better result, the physical interaction between the antennas should be researched and the resulting model function used for the distance approximation.
Another part of the antenna locations is the angle they are located to each other in the used coordinate system. The angles were also calculated from the best fit to the measured data. The results showed that the antenna angles that would give the best result stayed rather constant over the entire tested frequency range. They also matched closely with the angles that were measured in the physical setup. Those results suggest that there is no need to use any additional frequency based configuration for modifying the angles of the antennas. Combining this knowledge with how the distances change with increasing signal frequencies, the effective locations of the an-

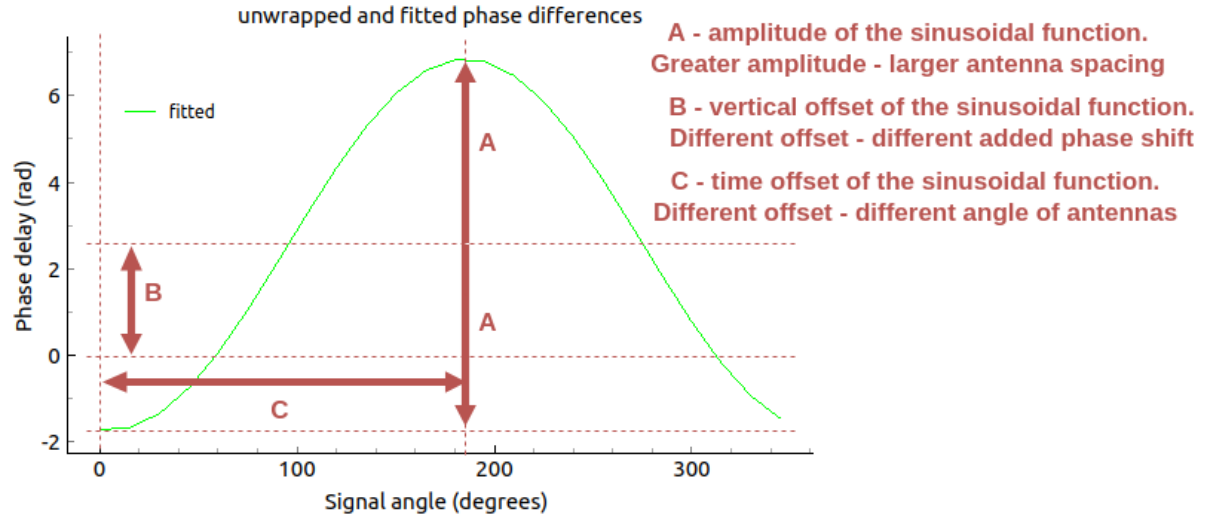

Figure 4.7: The properties of fitted sinusoidal function that are used to determine the antenna spacing parameters which would cause smallest average error.

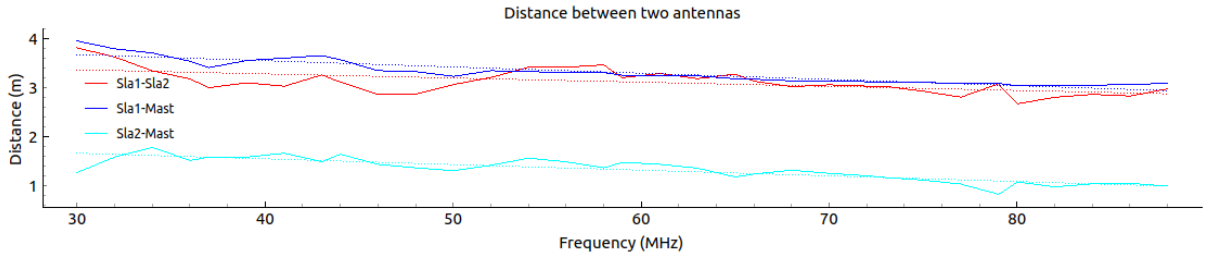

Figure 4.8: The calculated distances between the antennas that would give the smallest average error across all directions for different frequencies. The calculation is based on the measured results.

tennas come closer to the origin, when the origin is located at the center of the antennas. In other words, the whole setup shrinks to a smaller area, while maintaining its shape.
The third parameter that was possible to estimate from the best fit on the measured results, was the calibration offset in phase differences. This is the offset caused by different paths from the antennas to the measuring system. A linear change would be expected here as the time delay difference in different cables should remain constant while the wavelength changes causing different phases. The change would be faster for bigger difference in the length of the considered signal path. The analysis done on the measured data confirmed this relation.

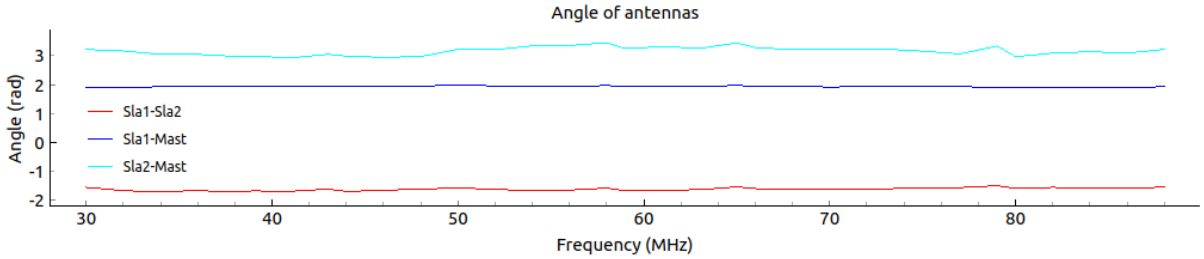

Figure 4.9: The calculated angles of the antennas that would give the smallest average error across all directions for different frequencies. The calculation is based on the measured results.

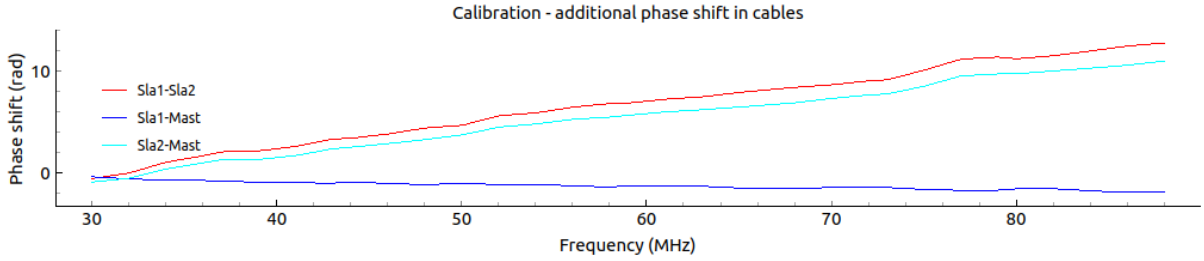

Figure 4.10: The calculated delays in cables for different frequencies. The calculation is based on the measured results.

5 Finalising

The final step of the project included making a simple demo application to showcase a possible use of the developed system. It included tools for visualisation and some complementing elements to show a possible way to use the system in a final setup. In this case the final development included mainly three features. Firstly, a simple server was developed, that allowed multiple devices to send DOA data to it. Secondly, a script was created to automatically start the HackRF boards in the correct mode and with correct parameters to save the data, process it afterwards to find the DOA estimate and send to the server. Lastly an UI was created that allowed user to initiate the script to detect DOA of any wanted signal and to show the results on a real map.

5.1 Automation architecture

The automation process that was developed during this thesis, included a command-line tool from hackrf-tools package for scanning the requested areas of the spectrum, by switching through all of the necessary frequencies. The tool was run by the server and the data was gathered and shown in the UI using graphing library ChartJS. This part of the system allows user to see any signals appearing in the spectrum and based on that information the user could decide to determine the DOA of any of those. The UI allowed the user to initiate the set of DOA scripts by clicking on the desired frequency bin.

When the user selected the frequency, the server gathered all of the necessary parameters for that frequency. The parameters were found based on the results of the analysis to provide the least average error for the used system setup and stored in a dictionary. Then a modified hackrf command-line tool for signal reception from all 3 devices was called with these parameters and option to save the received data to a file. This script captured a constant amount of data, after which a Python script took over. The python script calculated a single DOA estimate based on the received data using also parameters that correspond to the frequency. The resulting DOA probability distribution was shown to the user on a map along with the most probable direction vector and the device position as a reference. The probability distribution was shown as a polar graph around the device location with points further away indicating bigger probability. The UI for this can be seen in Appendix E.

5.2 Automation result

The automation result proved that it is possible to make this kind of system in the current form usable by end user, if all of the necessary setup and measurements have been done for the system at hand. However, the current architecture did not allow to create a setup and simply input its

parameters, which was the main goal of this thesis. This was mainly because of the frequency-dependant effective distance between antennas being different from physical distances. The current automation architecture also did one measurement at a time to prove the concept, however in the final use it is probable that the DOA subprogram should run and update the DOA estimate for the selected frequency until user chooses to end it.

6 Results and future work

The implemented POC software and analysis carried out, well visualised what system designed in this thesis is capable of and also pinpointed biggest causes for errors in such system. The simple system was proven to output reliable data that could be usable on its own in multiple different applications. It was also seen that for system implemented like this some theoretical analysis can be done, which can give valuable input to find the best configuration.

Additionally, several options were discovered which could further improve the developed system, that were described in this thesis, but implementing of which did not fit into the scope of this work. To make the system more universal and end-user friendly, these aspects could be researched more closely.

This work showed, that having the antennas close to each other, has effect on the received signal. This effected the reception delay, that was the basis for the calculation of DOA estimate. In order to better understand the effect and the resulting change in effective antenna locations to use in the algorithms, a simulation or a thorough measurement could be carried out. The simulation could include a way to construct a computer model for each new system to calculate the parameters for all possible frequencies and directions while considering any electromagnetic effects between the antennas. The alternative would be to make measurements using more setups and several frequencies to get an N-dimensional matrix with all of the changeable parameters and use fitting functions to learn the best parameters to be used.

Another more significant issue with the current system was that several programs were run as whole individual programs for each task. Although for the purposes of this work, this was sufficient, it is not the most effective way as lot of data has to be exchanged between those programs and parsed to different formats. It also introduces several sources for asynchronous operations that are harder to control between fully separate programs. Therefore, another feature that should be developed, before the system could be used is to overview the full system architecture and tie the programs that communicate with the HackRF hardware with the rest of the system back-end.

7 Conclusions

Over the two years, while the work covered in this thesis has been carried out, system requirements for a simple and affordable SDR-based direction estimation system were stated. Next, the work on a prototype system to test the feasibility of such a system and find the best hardware and software solutions was started. The initial tests were carried out in steps, beginning with two antennas and a well-controlled signal from a signal generator. The tests continued with real data received from the environment and finally used an additional antenna to improve the result and eliminate symmetric ambiguity. The tests gave reason to believe that such a system could be implemented. However, some unexpected drawbacks with necessary synchronisation required additional hardware and software components to be implemented. Finally, a method using a phase output of a complex-numbered correlation algorithm for system calibration and a mathematical MUSIC algorithm, which was implemented in an external Python library, was chosen for the rest of the analysis and development.

The analysis step consisted of analysing the system behaviour on both simulated and real-world data. A special Python program was created for the simulated data, which allowed to create input signal without any other information in the spectrum affecting the result, but still having all of the hardware processing steps applied to the signal mathematically. For the real-life data, a special program was also written using GRC that allowed saving necessary data and a Python program to process this data. Several aspects of the system were analysed, including the effect of antenna placement and signal source distance. As a result, it was concluded that parameters that were previously proven to work in real-time were sufficient to keep the error negligible compared to other unavoidable deviations. The most significant cause for problems was seen to be the nearby antennas influencing each other and the effective distance between antennas changing as a result. A usable but not proven to be a fully universal solution of using a linear relationship specific to a system setup was proposed.

Finally, a software stack was written to demonstrate a possible use for the system. For this, a simple server with a web interface was created to provide a user interface and automatic initialisation of the DOA estimation process. The visual aid provided by the software can be used to develop the system further, either to implement the future improvements listed in the thesis or to modify the project for specific purposes.

The work in the thesis satisfied the goals set for this thesis. The final setup was composed of low-cost and widely available radio peripherals and open-source tools and libraries, except for using extra hardware for synchronisation. However, this capability is relatively simple to add and can be achieved in multiple ways depending on the application. The process provided a probability distribution over all possible directions. The result was allowed to provide visual information in real-time without processing all of the data. This speed can be considered sufficient as no more data can be obtained by the user in real-time and if more information is needed,

the necessary data can be saved and analysed later using the same method. The final software stack also confirmed that such system can be used by less skilled operator.

Acknowledgement

I am thankful to my instructor J. Kalde for giving feedback on the practical and theoretical part of the thesis while allowing freedom in choosing the methodology and content. I would also like to thank the group of graduating friends for providing motivational weekly events for writing thesis together.

Erik Amor

References

[1] GreatScottGadgets, HackRF One https://greatscottgadgets.com/hackrf/one/ 16.05.2022, 15:14 (UTC).

[2] ElectronicsNotes. Understanding the Software Defined Radio Receiver SDR. https://www.electronics-notes.com/articles/radio/sdr-software-defined-radio-receiver/sdr-basics.php 16.05.2022, 16:20 (UTC).

[3] National Instruments Corp. Software-defined Radio: Past, Present, and Future https://www.ni.com/en-us/innovations/white-papers/17/software-defined-radio–past–present–and-future.html 16.05.2022, 16:20 (UTC).

[4] D. M. Vijayan and S. K. Menon, Direction of arrival estimation in smart antenna for marine communication, *2016 International Conference on Communication and Signal Processing (ICCSP)*, 2016, pp. 1535-1540, DOI: 10.1109/ICCSP.2016.7754416.

[5] R.Jaks, *UAV Direction Finding Using Phase Difference*, 2020, https://dspace.ut.ee/bitstream/handle/10062/72110/Jaks$_M Sc2020.pdf?sequence = 1isAllowed = y$.

[6] S. Sun, H. Li and J. Xiong, Direction of arrival estimation using compact MIMO array for portbale devices, *2017 IEEE International Symposium on Antennas and Propagation USNC/URSI National Radio Science Meeting*, 2017, pp. 1433-1434, doi:10.1109/APUSNCURSINRSM.2017.8072759.

[7] N. B. Kanters, Direction-of-arrival estimation of an unknown number of signals using a machine learning framework, 2020, DOI:https://essay.utwente.nl/80493/1/Kanters$_M A_E EMCS.pdf$.

[8] A. Rajani and P. Kora, Direction of Arrival Estimation by Using Artificial Neural Networks, 2021 *Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV)*, 2021, pp., DOI: .1109/ICICV50876.2021.9388514.

[9] Gnu Radio Wiki https://wiki.gnuradio.org/index.php/Main$_Page$ 19.05.2022, 21:50 (UTC).

[10] NodeJS https://nodejs.org/ 19.05.2022, 10:20 (UTC).

[11] W3Techs, Usage statistics of Node.js https://w3techs.com/technologies/details/ws-nodejs 18.05.2022, 10:00 (UTC).

[12] LeafletJS https://leafletjs.com/ 19.05.2022, 10:20 (UTC).

[13] ChartJS https://www.chartjs.org/ 19.05.2022, 10:20 (UTC).

[14] HackRF Documentation, Multiple Device Hardware Level Synchronization https://hackrf.readthedocs.io/en/latest/multiple$_device_hardware_synch.html$ 16.05.2022, 15:10 (UTC).

[15] Keysight Technologies, 33250A Function / Arbitrary Waveform Generator, 80 MHz https://www.keysight.com/zz/en/product/33250A/function–arbitrary-waveform-generator-80-mhz.html 18.05.2022, 10:00 (UTC).

[16] J. R. Machado-Fernández *Software Defined Radio: Basic Principles and Applications*, November 2014, https://revistas.uptc.edu.co/index.php/ingenieria/article/view/3160/4347

[17] M. K. Ozdemir, H. Arslan and E. Arvas, Mutual coupling effect in multiantenna wireless communication systems, *GLOBECOM '03. IEEE Global Telecommunications Conference (IEEE Cat. No.03CH37489)*, 2003, pp. 829-833 Vol.2, DOI: 10.1109/GLOCOM.2003.1258355.

[18] L. Ying, X. Chi and Z. Yourun, Direction of arrivals estimation for antenna arrays consider mutual coupling effect, *2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN)*, 2017, pp. 708-712, DOI: 10.1109/ICCSN.2017.8230203.

[19] R. D. Straw, *Arrl Antenna Book*, ARRL, 2003.

[20] Pyargus GitHub page https://github.com/petotamas/pyArgus 19.05.2022, 10:20 (UTC).

[21] R. Joshi, A. Dhande *UAV Direction Finding Using Phase Difference*, *International Journal of Research in Engineering and Technology (IJRET)* 2014, pp. 633-636, 10.1.1.686.2682.

[22] R. Schmidt, Multiple emitter location and signal parameter estimation, *in IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276-280, March 1986, DOI:10.1109/TAP.1986.1143830.

Appendix A. DOA flowchart in GRC.

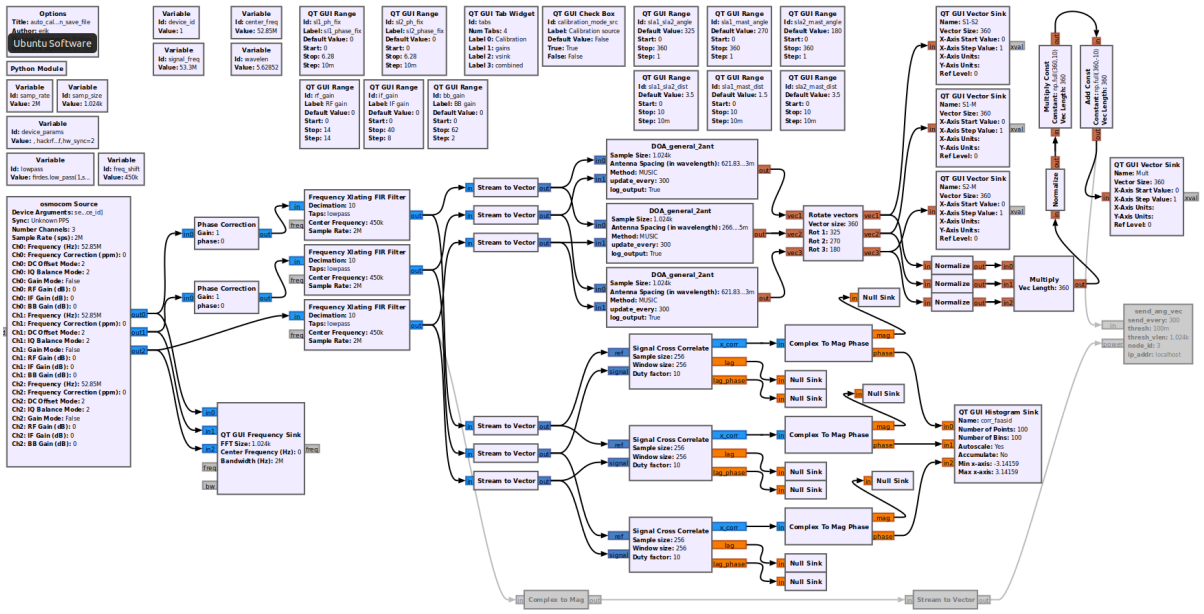

Figure 7.1: The flowchart for the DOA algorithm, developed in GRC as a POC.

Appendix B. DOA prototype UI.

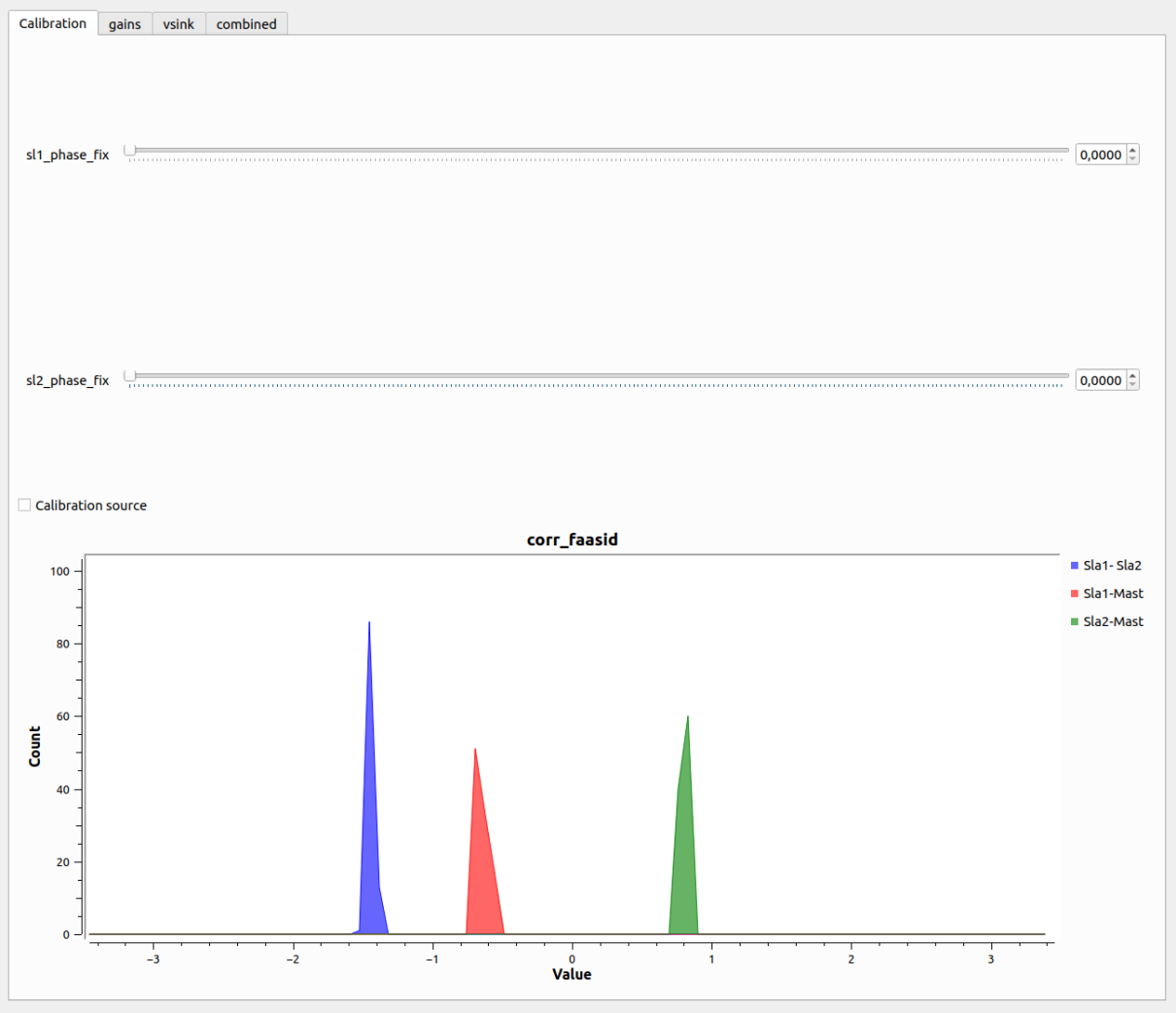

Figure 7.2: The tab in the prototype DOA UI showing the phase difference of between each pair of signals on histogram graph and sliders for choosing phase corrections to be applied to two of the signals.

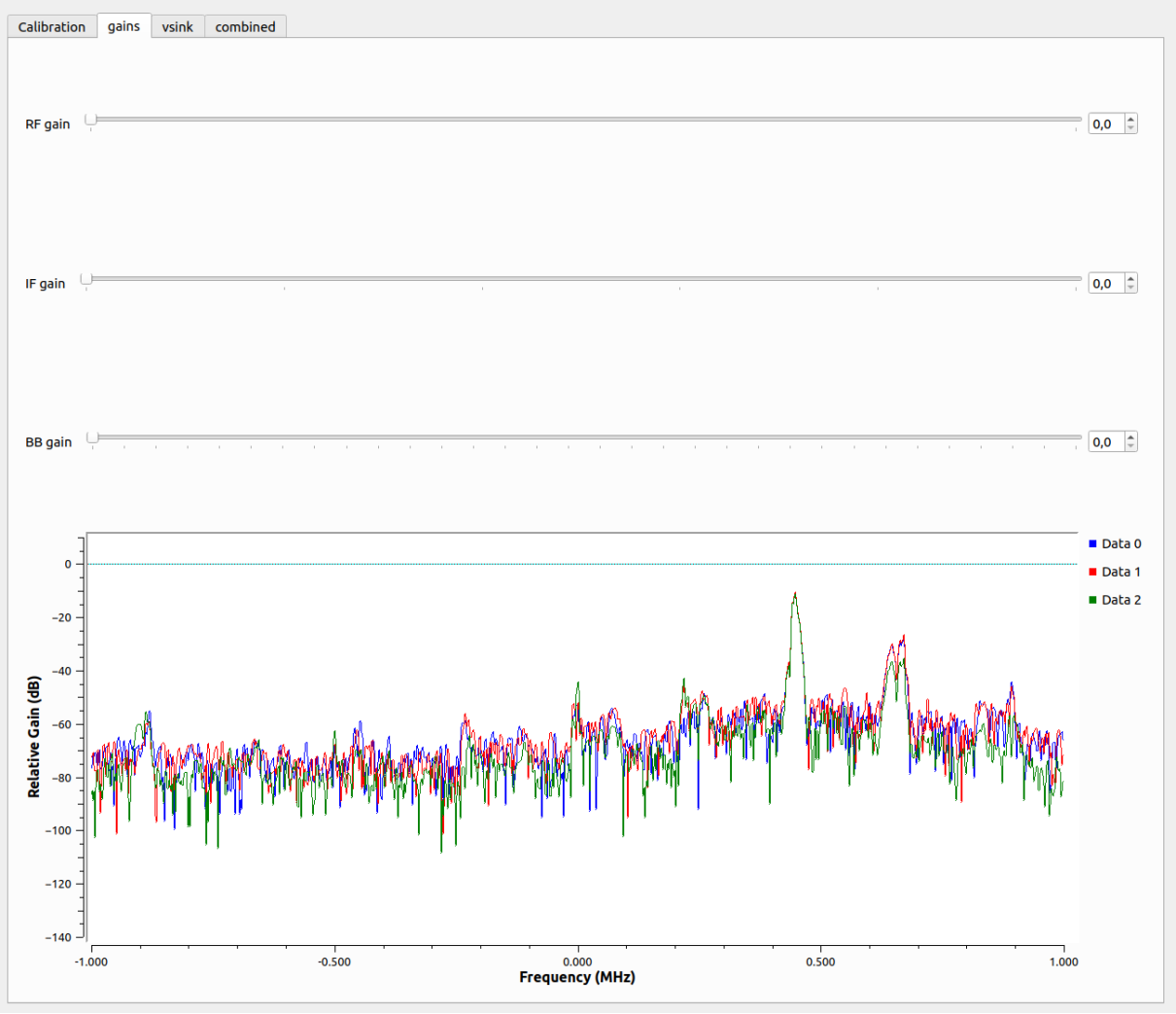

Figure 7.3: The tab in the prototype DOA UI showing the FFT result of the captured band of the signal. The tab also features sliders to set gains of the signals in different processing steps

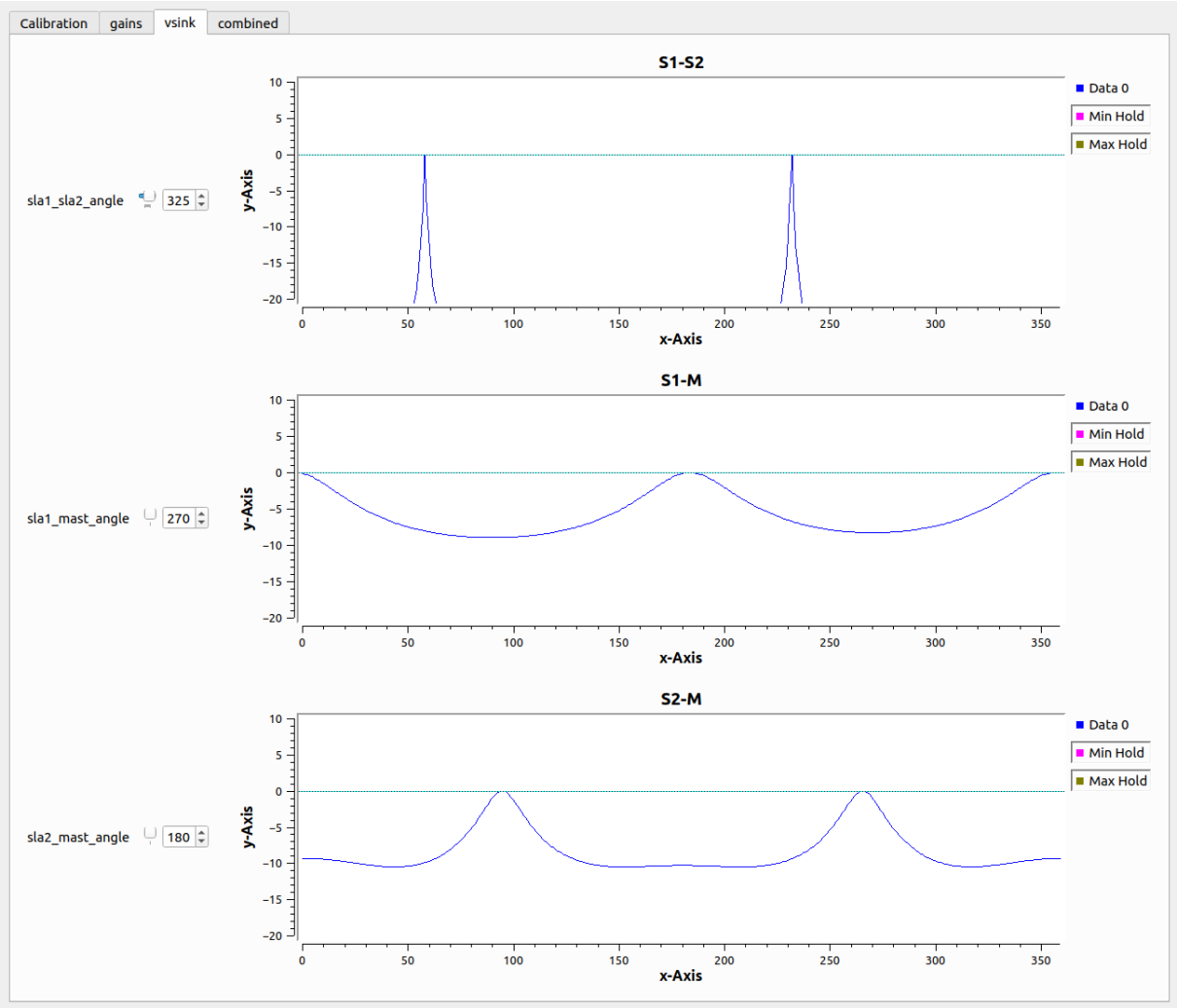

Figure 7.4: The tab in the prototype DOA UI showing the separate DOA estimations for each pair of antennas with current distance and angle parameters. The tab also features sliders for adjusting the angle that the pairs of antennas are located in the coordinate system.

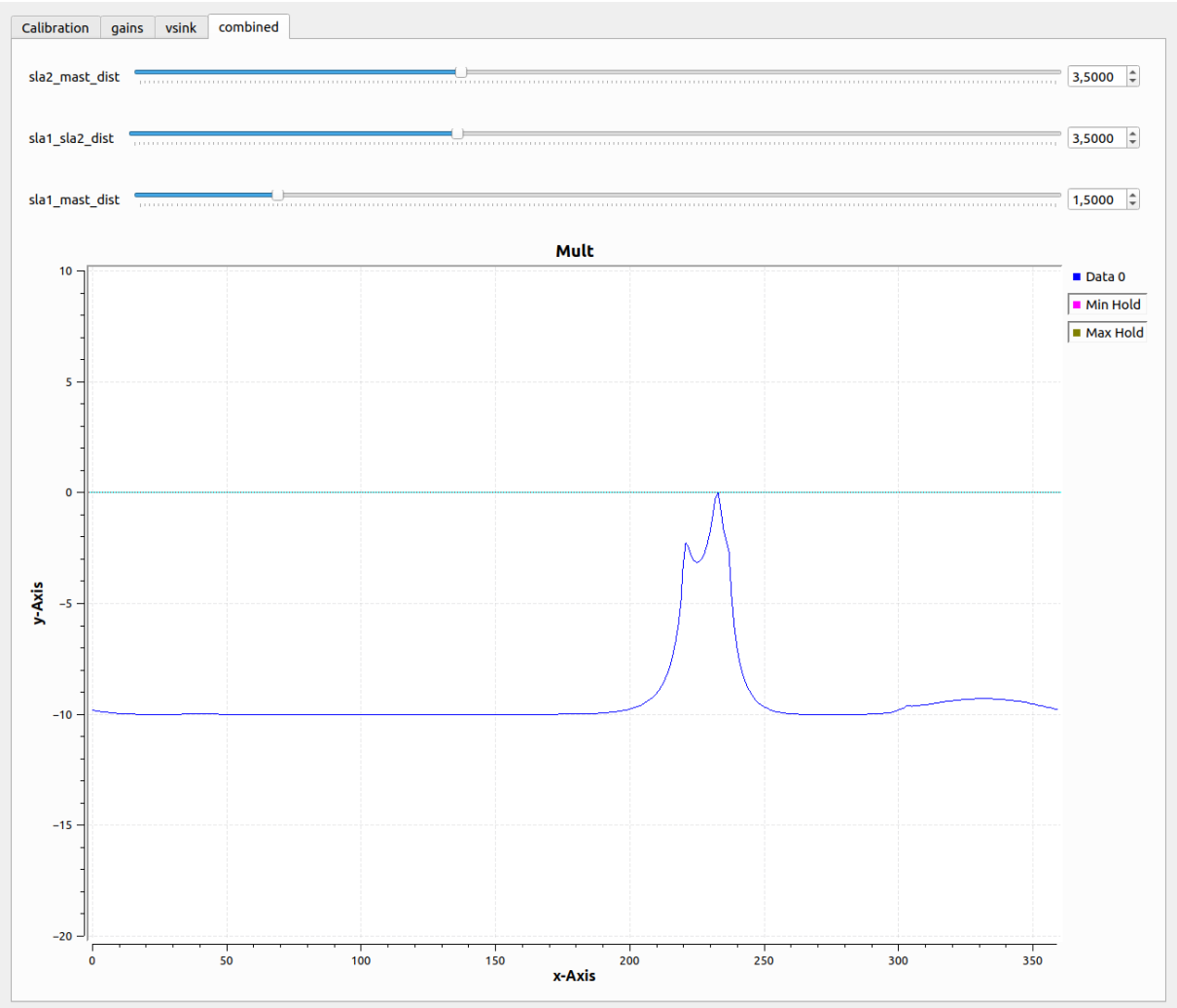

Figure 7.5: The tab in the prototype DOA UI showing the final DOA estimate combined from the 3 separate estimates based on the rotation of each pair of antennas. The tab also feature sliders for adjusting the distance between each pair of antennas that is used in the calculations.

Appendix C. Analysis capture flowchart.

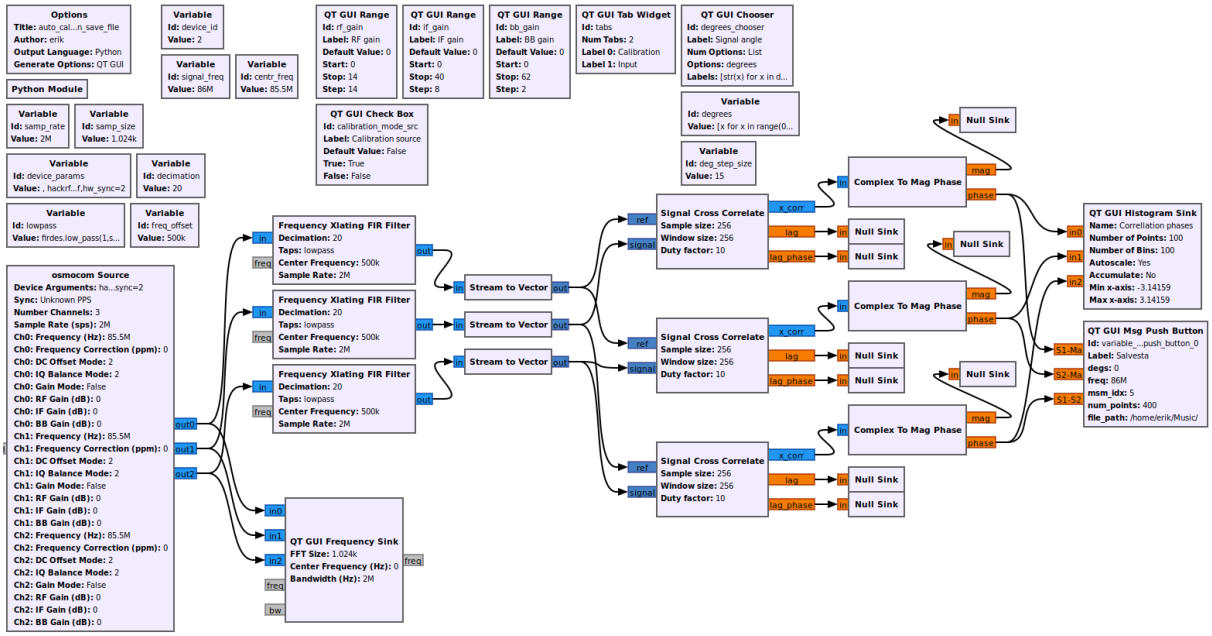

Figure 7.6: The GRC flowchart for the program to gather phase difference data from different angles for analysis.

Appendix D. Signal unwrapping UI.

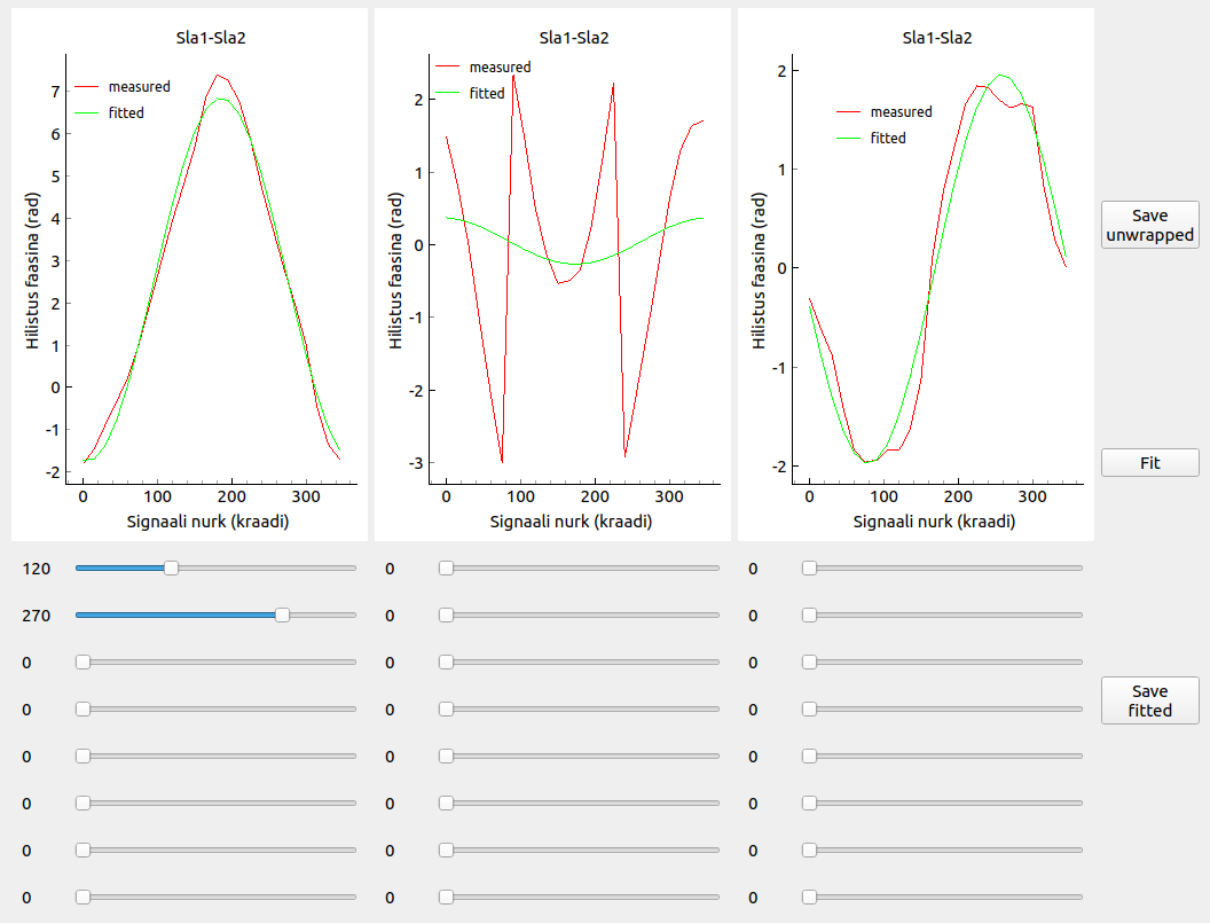

Figure 7.7: The UI for preprocessing and analysing real life data.

Appendix E. DOA final demo UI.

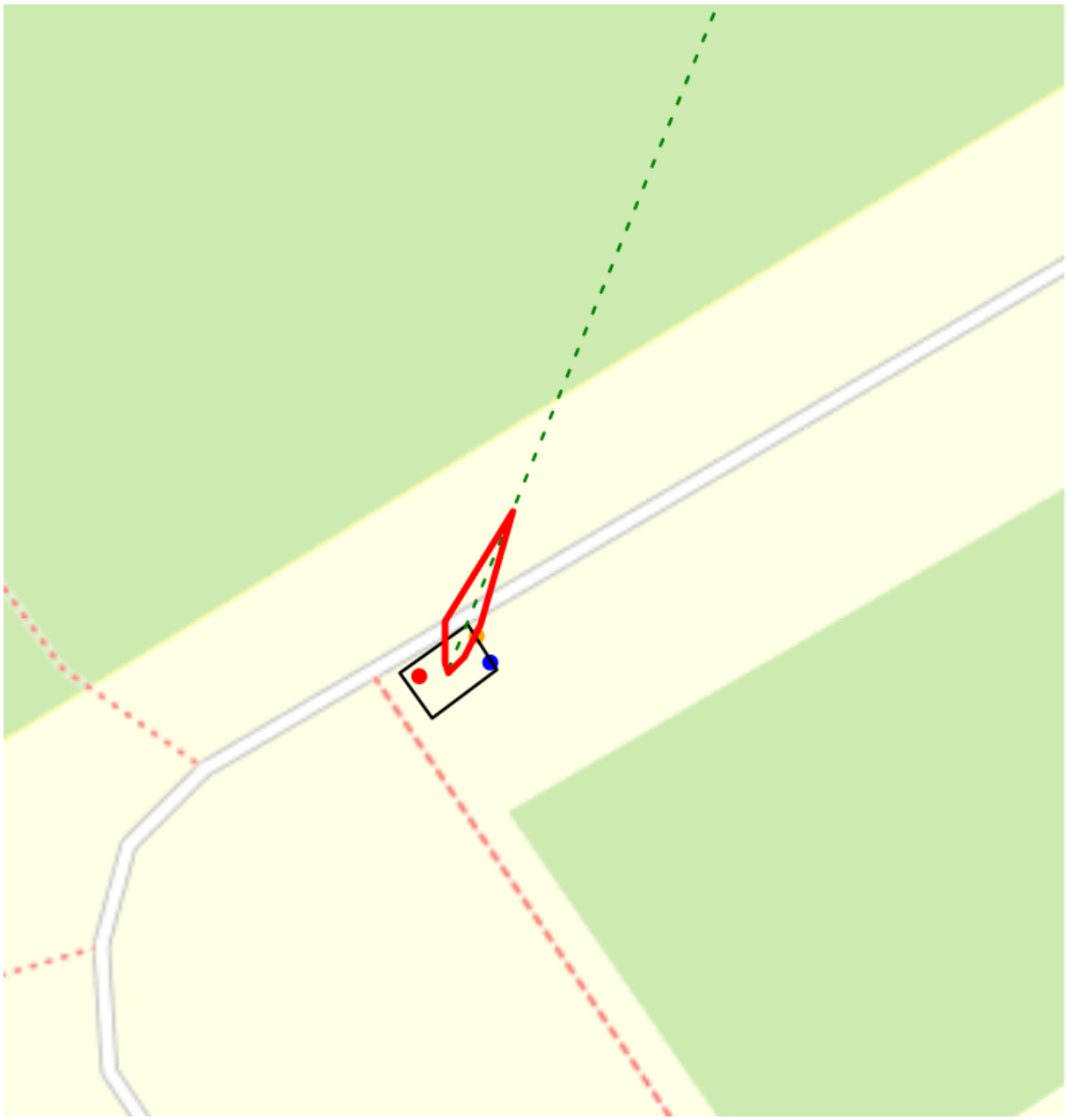

Figure 7.8: The tab in the final demo program showing the estimated DOA (red line) on a map with a added vector for most probable direction (green dotted line) and antenna location information (red, yellow and blue dots.

Non-exclusive licence to reproduce thesis and make thesis public

I, Erik Amor

1. 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 until the expiry of the term of copyright, my thesis

   ”Design, Testing and Analysis of an Affordable Direction of Arrival System Using Off-The-Shelf SDR Hardware and Software”

   supervised by Jaanus Kalde.

2. I grant the University of Tartu a permit to make the thesis specified in point 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 4.0, 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 commercial use 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 points 1 and 2.

4. I confirm that granting the non-exclusive licence does not infringe other persons’ intellectual property rights or rights arising from the personal data protection legislation.

Erik Amor
24.05.2022