



User Equipment Tracking Beamforming Using Software-Defined Radio

Candidate Number: 218831

Advisor: Falah H. Ali

Submitted for the degree of MSc 5G Mobile Communications and Intelligent
Embedded Systems
University of Sussex
School of Engineering and Informatics
September 2020
Word count: 13194

Abstract

The fifth generation of mobile communication systems introduces several enhancements in comparison to the previous generations. These improvements are achieved by using new technologies and techniques to tackle the existing problems such as the interference caused by multiple users and transmitting signals thorough the millimetre wave band. This work presents the implementation of transmitting and receiving beamformer system for the sub 6 GHz band by using a low-cost Software-Defined Radio unit. The sections in this report include the background of the beamforming and software-defined radio concepts, the design of multiple components and simulations, and the testing and verification of a real-life prototype performing an Over-the-Air test.

Acknowledgements

Thank you to the University of Sussex for the opportunity to study here, as well as the staff from the 5G Mobile Communications and Intelligent Embedded Systems course; especially Dr. Falah Ali, whose guidance and support made this work possible.

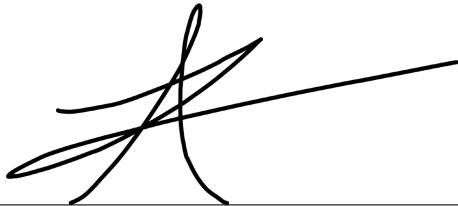
I want to thank the Science and Technology Council of Mexico (Consejo Nacional de Ciencia y Tecnología, CONACYT) that has sponsored my studies, and all the people there that helped me with all my academic and financial issues.

I also thank Mr John Kinghorn, whose generous sponsorship allowed me to achieve this milestone in my life.

I want to thank my parents and siblings, whose unconditional support inspires me to keep going. Finally, all my thanks and love to my dear wanguita, because all the work done here would have been easier without you; nonetheless, boring :).

Statement of Originality

This report is submitted as part requirement for the degree of MSc 5G Mobile Communications and Intelligent Embedded Systems at the University of Sussex. It is the product of my own labour except where indicated in the text. The report may be freely copied and distributed provided the source is acknowledged. I hereby give permission for a copy of this report to be loaned out to students in future years.

A handwritten signature consisting of several black ink strokes forming a stylized, abstract shape.

Candidate Number: 218831

List of Figures

2.1	Combiner beamformer antenna array.	9
2.2	Two-Element beamformer with different element distancing.	10
2.3	Ideal SDR block diagram.	13
2.4	Receiver architectures.	14
2.5	AD9361 RF Transceiver block diagram.	15
3.1	Simulink model for the UE tracking simulation.	20
3.2	Example of the coordinate system.	21
3.3	Simulink subsystem for the UE's movement.	21
3.4	Simulink subsystem for the Modulator	22
3.5	Simulink subsystem for the Receiver.	24
3.6	Simulink subsystem for the valid frame.	24
3.7	Simulink subsystem representing the wireless channel.	25
3.8	Simulink subsystem for the RX beamformer at the UE	27
3.9	Simulink subsystem for the TX beamformer at the BS	27
3.10	Simulink subsystem for the DOA estimation.	27
3.11	BladeRF 2.0 A4 block diagram. [1]	29
3.12	Device setup and utilisation. [2]	30
3.13	Antenna holder example structures.	33
3.14	Handmade antenna structure with antennas and SMA cables.	34
3.15	Technical drawing of the antenna holder.	34
3.16	Simulink model for the HIL demonstration.	35
4.1	Simulink model for testing the MIMO interface of the BladeRF.	37
4.2	Frequency spectrum of the signals coming from the radio station and the SDR.	38
4.3	IQ signal from both RX channels in the SDR.	38
4.4	Simulink model for testing the Modulator and Receiver blocks.	39
4.5	Constellation diagrams showing different modulated signals.	40
4.6	Results of the receiver test.	41
4.7	BER of the receiver test.	42
4.8	Example layout for the TX beamformer test.	42
4.9	Normalised radiation pattern for the theoretical and real beamformers.	43
4.10	Example layout for the RX beamformer test.	44
4.11	Simulink model for representing the RX beamformer test scenario.	45
4.12	Theoretical radiation pattern for the beamformer at different steering angles.	46
4.13	Measured constellation diagram after the RX beamforming.	47
4.14	Theoretical phase pattern for the beamformer at different steering angles.	47
4.15	Constellation diagram of the signal sensed by the RX antenna array.	48
4.16	BER performance across experiments for the UE tracking simulation test	50

4.17	BER performance for systems with different antenna array elements for the UE tracking simulation test.	50
4.18	Measured DOD for the UE tracking simulation test.	51
4.19	Real test environments for the UE tracking test	52
4.20	Errors across time for each environment in the UE tracking test.	53
4.21	DOA measurements for the UE tracking tests.	54

List of Tables

2.1	SDR comparison table.	15
3.1	BladeRF's RF specifications. [1]	29
3.2	SC16Q11 layout example for 2 channels.	31
3.3	Formulas for the parametric design of the antenna holder.	33
4.1	System parameters for the test scenarios.	36
4.2	Testing parameters for the modulator block.	39
4.3	Testing parameters for the receiver block.	40
4.4	Results of the receiver test scenario.	41
4.5	Results of the TX beamforming test.	43
4.6	Simulated baseline results for the RX beamformer test.	45
4.7	Real baseline results of the RX beamformer test.	45
4.8	Results of the simulated scenario for the RX beamformer.	46
4.9	Results of the real scenario for the RX beamformer.	46
4.10	Input parameters for the UE tracking simulation test.	49
4.11	Component combination for each experiment in the UE tracking simulation test.	49
4.12	Resulting BER for the UE tracking test.	52

Contents

1	Introduction	6
1.1	Project Description	6
1.2	Justification	7
1.3	Objectives	7
2	Background	8
2.1	Beamforming	8
2.1.1	Introduction	8
2.1.2	Theoretical Background	8
2.1.3	Implementing Beamforming	11
2.1.4	Adaptive Beamforming	11
2.2	Software-Defined Radio	13
2.2.1	Introduction	13
2.2.2	Hardware and Flexibility	13
2.2.3	Available Software-Defined Radio Units	14
2.3	Beamforming in Software-Defined Radio	16
2.3.1	Coherence and Synchronisation	16
2.3.2	Interfacing	16
2.3.3	Transceiver Architecture	16
3	Methodology	18
3.1	Introduction	18
3.2	Simulation	19
3.2.1	Problem Statement and Initial Assumptions	19
3.2.2	Modulator and Receiver	21
3.2.3	Wireless Channel	25
3.2.4	Beamformers and Direction of Arrival Estimation	26
3.3	SDR Selection	28
3.3.1	Introduction	28
3.3.2	Requirements	28
3.3.3	Selected Hardware	28
3.4	Hardware Interface for MATLAB/Simulink	30
3.4.1	libbladeRF	30
3.4.2	MATLAB System Object	31
3.5	Antenna Mounting	33
3.6	Hardware-in-the-Loop Demonstration	35

4 Tests and Validation	36
4.1 Introduction	36
4.2 MIMO Interface for the BladeRF	37
4.3 Modulator and Receiver	39
4.3.1 Modulator	39
4.3.2 Receiver	40
4.4 Beamformers	42
4.4.1 TX Beamformer	42
4.4.2 RX Beamformer	43
4.4.3 Direction of Arrival	47
4.5 User Equipment Tracking	49
4.5.1 Simulation	49
4.5.2 Real Implementation	50
5 Conclusions	55
5.1 Future Work	55
A Repository	57

Chapter 1

Introduction

1.1 Project Description

This report outlines the work done for UE tracking beamforming using Software-Defined Radio, and describes the different stages of development from the background and state-of-the-art to testing and validation of a real-life prototype. The final product of this project is a beamformer system capable of steering the outgoing signal's direction and filtering the incoming signals using a software algorithm and a software-defined radio unit.

The background section in chapter 2 refers to the research done for the different components used for this project, such as beamforming and software-defined radio. The beamforming background in section 2.1 defines the concept of beamforming, its mathematical models, the different implementation methods and techniques for angle estimation. The software-defined radio background in section 2.2 describes the concept of SDR, why is it used, a rough explanation of how some of its components work and the products available at the moment of writing this report. Finally, the final section (2.3) outlines the requirements beamforming introduce, which are particularly important when implementing it using a software-defined radio.

Chapter 3 refers to the methodology and thought process for developing the different components used for the project. The simulation model in section 3.2 presents the intended scenario and the software model representing its principal components; the goal is to present the developed blocks for beamforming and the end-to-end communications system which are also used for other sections besides simulation. Section 3.3 describes the selection process of a software-defined radio unit based on the requirements introduced in section 2.3. After selecting the unit, section 3.4 describes the development of a software interface for the selected software-defined radio so that it can be used inside the model environment. Section 3.5 describes the design and manufacturing of a parametric antenna holder for antenna arrays. Finally, section 3.6 presents the combination of all the components featured in chapter 3 as a Hardware-in-the-Loop model capable of testing the simulation environment in real-life while using the same algorithms and validation metrics.

Chapter 4 outlines the test and validation process for most of the products developed in chapter 3. Standard metrics as the BER and EVM are used to evaluate the communication link capabilities, while other non-standard tests are used to evaluate the beamforming algorithm implemented in the software-defined radio. A real-life test scenario is described and compared to its simulation counterpart in section 4.5.

Finally, some conclusions, overall comments and further work is discussed in chapter 5.

1.2 Justification

This project originated from the new generation of mobile communications systems, commonly called 5G. This generation introduces several new technologies and improvements in comparison to the previous generations. As 5G intends to improve data throughput, user capacity and latency, designers must introduce new technologies to tackle the inherited constraints. Beamforming is one of these technologies and is capable of transmitting signals through the millimetre wave band while reducing the produced and received interference. Beamforming is a mature technology, commonly used in radar, medical and acoustic applications and its use in digital communications are becoming more common, as the newest WIFI standard, WIFI6, includes beamforming as one of its main features. Also, the software-defined radio part of this project shows that complex algorithms used in 5G can still be exploited using low-cost hardware as the unit used in section 3.3.

1.3 Objectives

The goals for this project are divided into two categories, primary and secondary goals. Main goals refer to the minimum expected work, but also the highest valued, requiring the most effort and research. Secondary goals are details that would make the primary goal's result more accessible, useful and accurate for future work to be more accessible.

The primary goals are the following. A beamforming antenna array prototype will be designed, modelled using a Hardware-in-the-Loop software tool to allow simulating and deploying the model in a fast and convenient way. As of the model, the antenna array will be suitable to act as a beamforming system consisting of a Software-Defined Radio unit, based and designed using the design constraints referred in sections 2.1 and 2.3. This system will hold a transmitter and receiver beamforming algorithm capable of being executed in Software-Defined Radio devices; this algorithm will implement the theoretical background of Beamforming, as shown in section 2.1.2. Lastly, the system will be able to detect users and then track them in order to maintain a stable communication channel; this feature is covered in section 2.1.4. A practical implementation of the designed Beamforming system will be done by building a test bench and use it on an Over-the-Air test with another SDR platform.

The secondary goals are mentioned as optional features, such as the system being flexible and accurate for natural expansion in the future, to create a simulation model in order to verify and deploy the system's program using Hardware-in-the-Loop and to test different adaptive beamforming algorithms. The work plan includes time and resources needed to achieve these goals, but will not be obligatory to implement in the final version.

Chapter 2

Background

2.1 Beamforming

2.1.1 Introduction

Beamforming is a spatial diversity technique to narrow the directivity of an antenna array to focus only in a specific direction. Compared to omnidirectional antennas which radiation pattern is constant in every direction and often unitary, a beamformer's radiation pattern is only unitary or larger at a few directions. One of the main characteristics of beamformers is to be able to steer its main lobe to any desired direction within a valid range. The advantages of using beamforming are the reduced interference compared to omnidirectional antennas and the BER improvement caused by the inherited gain of the antenna array.

Beamforming has been present in several different industries for a few decades, with applications like sonar radar, medical ultrasound and acoustics and, of course, mobile communications already exploiting the benefits this technique offers.

2.1.2 Theoretical Background

The mathematical analysis of transmission (Tx) and receiving (Rx) beamformers, often called precoders and combiner beamformers, respectively, is reasonably straightforward. The derivation of the Rx beamforming, the combiner, assumes a Uniform Linear Array (ULA) composed of M antenna elements and placed at a constant distance d . As a signal approaches the antenna array at an angle θ , it is first sensed by one of the edges of the array; element 0 in the case of the ULA in figure 2.1, is considered the reference element. Then, the signal reaches the following element after a lapse, followed by the next. [3, 4]

The lapse in which one element receives the incoming signal to the following is given by the propagation speed of electromagnetic waves, the distance and the signal's direction of arrival, and is described by the equation 2.1, where c is the speed of light. The consequence of this phenomena is that each antenna element in the array will get the same signal, but with a time delay (τ) between adjacent elements. Each element's signal $x_m(t)$ is then multiplied by the complex conjugate of the constant w_m , which modifies the signal's amplitude and phase before adding them at the output stage. The mathematical model for ULA combiners is described by equation 2.2. [3]

$$\tau = \frac{d \sin(\theta)}{c} \quad (2.1)$$

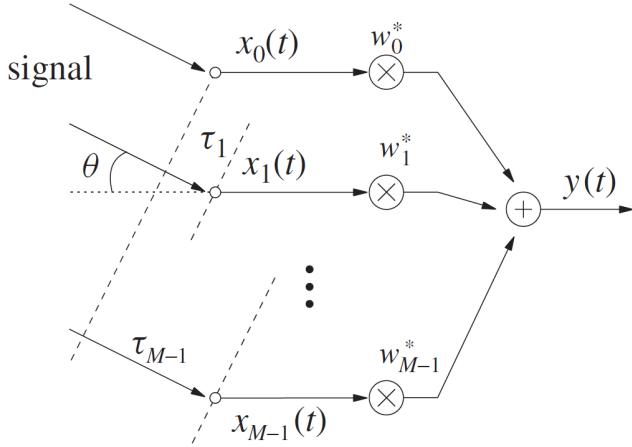


Figure 2.1: Combiner beamformer antenna array. [4]

$$y(t) = \sum_{m=0}^{M-1} x_m(t) w_m^* \quad (2.2)$$

Assuming $x_m(t)$ is a complex wave, it can be replaced by its phasor form $x_m(t) = \exp(j\omega t)$. Hence, all the elements of the array can be replaced their phasor form and then add the lapse $\tau_m = m\tau$ as a phase shift. Also, the Direction of Arrival (DOA) θ is constrained to the range $[-\frac{\pi}{2}, \frac{\pi}{2}]$, and the phase shift of the first element of the array is $\tau_0 = 0$. The output of the combiner beamformer is given by equation 2.3, while its frequency response is given by equation 2.4

$$y(t, \theta) = \sum_{m=0}^{M-1} \exp(j\omega(t - \tau_m)) w_m^* = \exp(j\omega t) \sum_{m=0}^{M-1} \exp(-j\omega\tau_m) w_m^* = \exp(j\omega t) \sum_{m=0}^{M-1} \exp(-j\frac{\omega d \sin(\theta)}{c}) w_m^* \quad (2.3)$$

$$P(\omega, \theta) = \sum_{m=0}^{M-1} \exp(-j\omega\tau_m) w_m^* = \sum_{m=0}^{M-1} \exp(-j\frac{\omega d \sin(\theta)}{c}) w_m^* \quad (2.4)$$

Similar to digital signal processing, the antenna elements must be placed within a maximum distance from each other to avoid temporal aliasing. This constraint guarantees that the recovered signal by each antenna element to have a unique and ascending phase shift ($\angle \exp(j\omega\tau_{m+1}) > \angle \exp(j\omega\tau_m)$). The signal's wavelength gives the constraint, and its described by the equation 2.5. Replacing the constraint in equation 2.5 in equation 2.4 results in the frequency response for an optimal combiner. The beamformer's radiation pattern can be computed using equation 2.6, which describes the array's magnitude at different directions of arrival.

$$d \leq \frac{\lambda}{2} = \frac{\pi c}{\omega} = \frac{c}{2f} \quad (2.5)$$

$$P(\theta) = \sum_{m=0}^{M-1} \exp(-jm\pi \sin(\theta)) w_m^* \quad (2.6)$$

A beamformer's radiation pattern consists of two components: the main lobe and the side lobes. The main lobe is the most prominent and widest spike in the beamformer's magnitude; by default, the

main lobe is centred at 0° . The side lobes are narrower and smaller lobes located near the main lobe, and these are often not desired. Finally, the area between lobes is called the null, as it has the lowest magnitude.

As previously stated, w_m is a complex weight that describes the element's gain and phase shift. These parameters are used to steer the beamformer's main lobe and nulls to filter signals from the desired DOA deliberately and to attenuate signals from a different direction. Weights are also used to suppress the side lobes by using techniques as the ones used in DSP (FIR filter design and window functions). Each of the previous features is linked to a component of complex numbers; altering the magnitude may help reducing the side lobes while increasing the phase steers the main lobe. The complex weight is defined by equation 2.7, where A_m is an array for suppressing the side lobes and ϕ is the desired steering angle. Replacing equation 2.7 into equation 2.6 results in the frequency response of a optimal beamformer ($d = \lambda/2$) with an steerable main lobe presented in equation 2.8.

$$w_m = A_m \exp(jm\pi \sin(\phi)) \quad (2.7)$$

$$P(\theta, \phi) = \sum_{m=0}^{M-1} A_m \exp(-jm\pi(\sin(\theta) - \sin(\phi))) \quad (2.8)$$

Steering the main beam may cause an undesired mirrored version of the same if the steering angle is above a certain value. This impairment is called Grafting lobe, and the steering angle range for avoiding it is given by equation 2.9. The maximum steering angle for optimal beamformers (where $d = \lambda/2$) is 45° . Although this limitation can be solved by placing the antenna elements closer to each other, the array directivity is sacrificed, and more antenna elements are required to maintain the original gain pattern; resulting in a trade-off between maximum steering angle and directivity.

$$\phi \leq \arcsin\left(\frac{\lambda}{d} - 1\right) \quad (2.9)$$

Beamformers can be modelled as $(M - 1)^{th}$ order Low-pass FIR filter. The filter's normalised

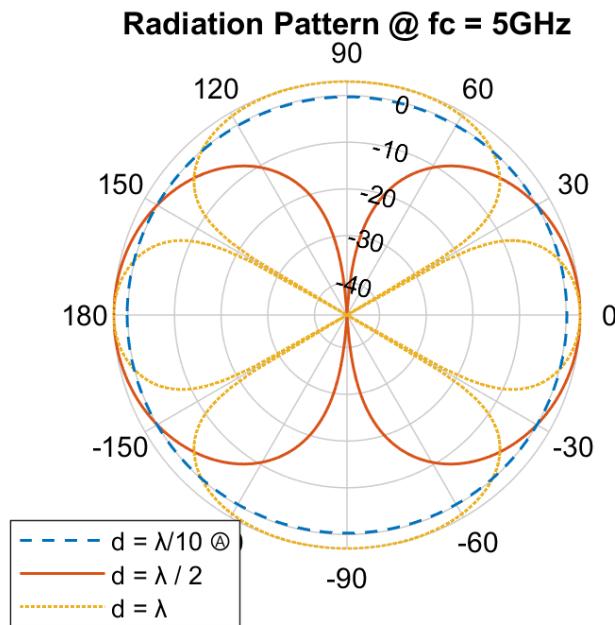


Figure 2.2: Two-Element beamformer with different element distancing.

pass-band and stop-band frequencies can be input as the main lobe's pass-band direction and stop-band direction. This property makes designing weights for sidelobe suppression simple using tools like MATLAB. The only concern the designer needs to consider is that beamformers have the same limitations as FIR filters and sharp responses require arrays with many elements [4]. If the designer does not desire to implement side lobe suppression, then A_m may be defined as a unitary array.

2.1.3 Implementing Beamforming

Although Beamforming's only requirement is to satisfy the equation 2.8, real implementation of beamformers is usually categorised into three types. The main difference between the three types is where the precoder and combining stages are performed.

Analogue Beamforming

Analogue Beamforming performs precoding and combining using specialised circuitry hardware; this makes an exceedingly power-efficient and cost-effective beamformer; since the incoming or transmitted signals can be processed without even considering Beamforming is applied in the system, excluding the steering algorithm.

It also saves on the use of RF Front Ends, since one of these can drive multiple antennas while keeping a hardware phase shifter and amplifier to each antenna. Having this is much better in terms of power and cost-efficiency.

The disadvantage of Analogue Beamforming is the reduced flexibility and added hardware complexity, which also makes steering the main lobe more difficult.

Digital Beamforming

Digital Beamforming main feature is the ability of precoding or combining using DSP algorithms; making this type of beamformer very flexible and configurable by only modifying the DSP algorithm in a processor.

It is also the simplest one to implement using off-the-shelf components since each antenna requires much hardware to receive or transmit signals. This detail makes Beamforming possible using existing solutions or Software Defined Radio units.

The most significant disadvantages of Digital Beamforming is the low scalability efficiency, due to the cost and power requirements of having an RF Front End to drive a single antenna.

Hybrid Beamforming

This implementation combines the attributes of both Analogue and Digital Beamforming by allowing the system to perform the precoding and combining in both baseband and broadband domains.

This type of beamformer combines the benefits and only a few disadvantages of both Analog and Digital Beamforming, by having flexibility and power and cost-efficiency. This benefit allows Hybrid Beamforming to be implemented on a large scale.

Some disadvantages of Hybrid Beamforming are the increased hardware and software complexity to be able to perform combining and precoding efficiently in both domains.

2.1.4 Adaptive Beamforming

Adaptive Beamforming is a technique to update the beamformer's weight according to the incoming signal; mostly to detect the direction of the incoming signal and steer the main lobe accordingly, while

suppressing interference. Adaptive Beamforming is only a category of a total of three; other techniques such as scanning and switched Beamforming are mature technologies, but the most flexible and efficient beamformers are adaptive.

The Spatial Covariance Matrix

Most of the adaptive beamformers require to compute channel estimation techniques, but instead of using pilots, they use something called the Spatial Co-variance Matrix. This matrix, often referred to as the R_u , matrix, while its subscript represents the device that computes the matrix; in this case, the Uplink (The User Equipment). The Spatial Covariance Matrix is given by the Expected Value ($E[X] = \int_0^\infty xf(x)$) of the auto-correlation of the received signal [3].

$$R_u = E[uu^H] \quad (2.10)$$

Temporal Reference Algorithms

Many algorithms are based on the Wiener Solution, which consists in optimising the error signal $J = E[|e(k)|^2]$ to obtain the optimised weights $W_{opt} = R_u^{-1}P_u$, where R_u is the spatial correlation matrix and P_u is the cross-correlation between the input and desired signals. The computational requirement of this algorithm is significant since a matrix must be inverted. Because of this, other algorithms such as the Method of Steepest Descent, Least Mean-Squares, Direct Matrix Inversion and Recursive Least-Squares are used to lower the computational requirement by using estimations and recursive solutions. [3]

Source Location Algorithms

Other algorithms called Source Location Techniques, these allow the UE to find the direction of the base station and then to update the steering angle ϕ while keeping the previously designed weights. Two of the most popular methods are the MUSIC and ESPRIT algorithms. Is fair to say that these algorithms do not directly output the estimated DOA; instead, they return the spatial spectrum, and the peaks of this spectrum are interpreted as the DOA. [3]

2.2 Software-Defined Radio

2.2.1 Introduction

Software Defined Radio (SDR) is a term to refer to radio systems with almost all its functionality done in software, instead of hardware as it is commonly found in most of the RF applications in recent times. An ideal SDR, as shown in Figure 2.3, would have an RF Front End only consisting of a power amplifier and high-speed Analogue to Digital converter. Hence, the remaining physical layer functions such as modulation, synchronisation or encoding are done using DSP. [5]

SDR systems are expected to work in a wide range of the frequency spectrum and to perform various operations as a dedicated hardware implementation would have. For example, using a low-cost SDR unit for hearing a local FM radio station and for getting an OFDM signal using QAM modulation in the 2.4 GHz band are both widespread uses.

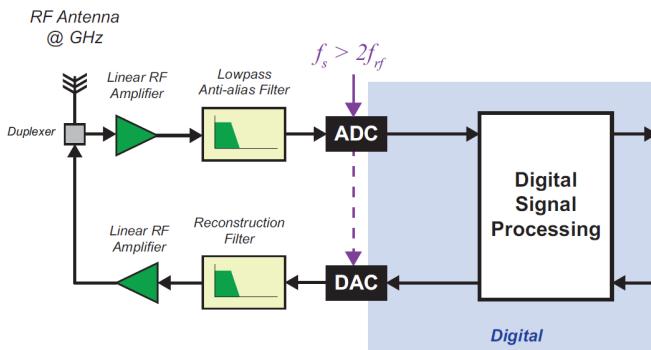


Figure 2.3: Ideal SDR block diagram. [5]

2.2.2 Hardware and Flexibility

Model-Based Design and Hardware in the Loop

Unlike old radio units using specialised hardware to receive and process a particular kind of incoming signal, SDR hardware is generic enough to process data of several frequencies and modulation schemes; a Field Programmable Gate Arrays (FPGA) inside the SDR radio often does this feature. FPGAs in the context of DSP, are faster and more efficient than microcontrollers; because of the dedicated DSP slices inside the silicon, the higher processing speed for RF applications and the available tools to design and simulate RF systems.

High-Level Synthesizer (HLS) systems like MATLAB/Simulink, Python or SystemC are used to model an RF system, simulate it to properly tuning and debugging, and finally generating HDL code to program the real hardware according to the model. This process is called Model-Based design. HLS software may already have specific tools to implement specific hardware or software processes, making the model generation even more comfortable.

Another commonly used design methodology is the Hardware in the Loop (HIL). Similar to Model-Based Design, HIL uses a high-level modelling software such as MATLAB/Simulink or GNU Radio to develop an algorithm of the system which includes the SDR as part of the model. Often, HIL limits the capability of the SDR as no custom HDL is generated to be executed in the FPGA; this method uses the SDR as a signal source and sinks so the computer can execute the intended algorithm. HIL models are faster to test and verify than Model-Based Design and also easier to create as no subsystems are compiled into HDL; however, a performance penalty is present as the algorithm is not being wholly executed in hardware.

Transceiver Architectures

An essential stage in SDRs is the AD/DA conversion architecture. There are three different architectures to get or produce signals. The Heterodyne architecture is the oldest from the three, and consists of a physical downconverter using a mixer and a local oscillator (LO); this method shifts the received broadband signal to baseband for the ADC to sample it, or in the opposite direction if a DAC is generating the signal. The Direct Conversion architecture (Also known as Zero-IF), doubles the bandwidth efficiency by adding a second Heterodyne converter, but the LO signal for the second converter is shifted by 90°; this enables having two independent data streams known as the In-Phase and Quadrature (IQ) signals. Finally, the Direct Sampling architecture converts signals from broadband to baseband and vice versa using software; this is the most demanding method as the DAC/ADC must have a sampling rate in the order of hundreds or even thousands of MHz. However, this method offers the most excellent flexibility as the signal can be easily modified even when it is up-converted.

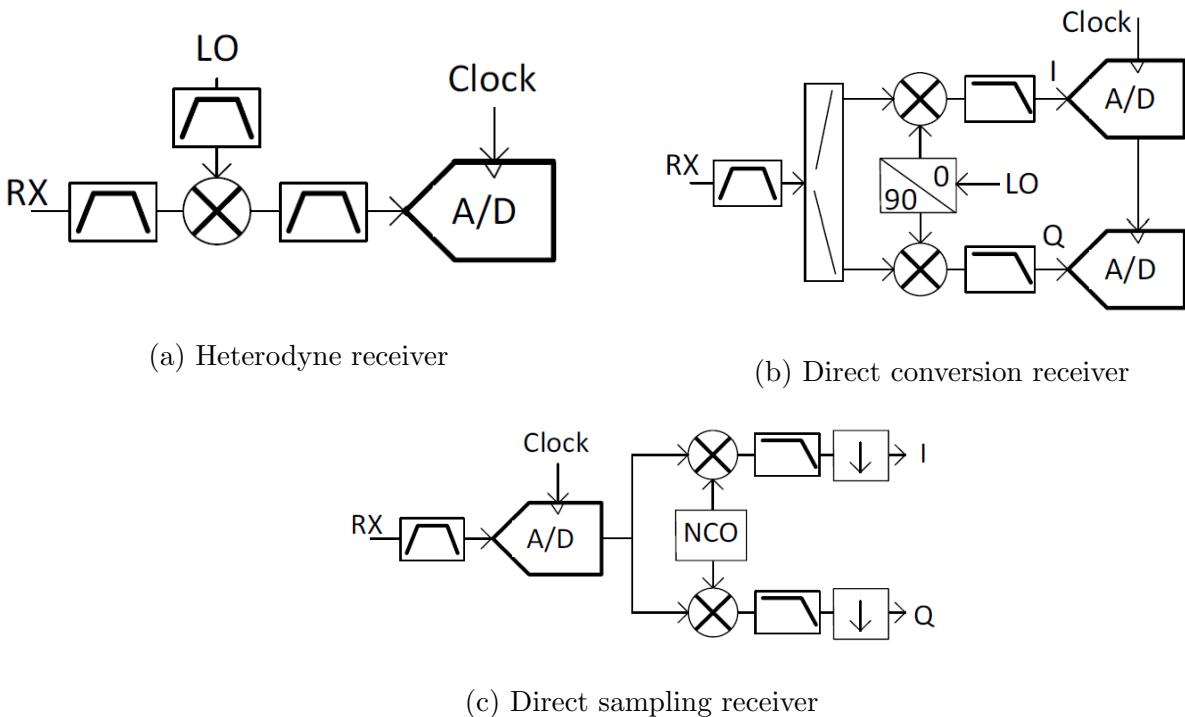


Figure 2.4: Receiver Architectures. [6]

Most of the SDRs in the market use the Direct conversion architecture, as it is compatible with many kinds of signals such as AM, FM, PSK and QAM. Additionally, most SDRs use RF front-ends packed in a single IC, providing a power amplifier, mixer, RF filter, ADC/DAC and data interface for it to be used with many microcontrollers. Many of these ICs also include several coherent TX and RX channels using a single LO. Two important manufacturers of transceiver ICs are Lime Microsystems and Analog Devices.

2.2.3 Available Software-Defined Radio Units

Many SDR units are available to buy today, the vast majority of them are meant for research and industrial applications with companies such as Ettus Research and National Instruments being two of the most popular SDR manufacturers. These devices are expected to have many features such as

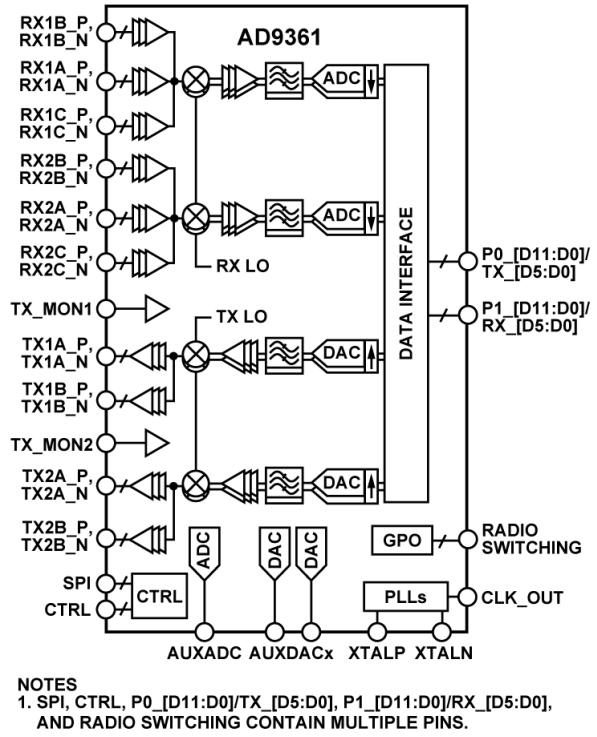


Figure 2.5: Block diagram of the AD9361 RF transceiver. [6]

SDR	Ettus B210	Hack RF	RTL-SDR	LimeSDR	PlutoSDR	BladeRF 2
Max Freq	6 GHz	6 GHz	1.7 GHz	3.8 GHz	3.8 GHz	6 GHz
RF Bandwidth	61.44 MHz	20 MHz	2.4 MHz	61.44 MHz	20 MHz	61.44 MHz
Interfacing	USB3.0	USB2.0	USB 2.0	USB3.0	USB 2.0	USB3.0
Sample Depth	12-bit	8-bit	8-bit	12-bit	12-bit	12-bit
Duplex	Full	Half	N/A	Full	Full	Full
MIMO	2x2	1x1	0x1	2x2	1x1	2x2
TX Power	10 dBm	15 dBm	N/A	10 dBm	7 dBm	8 dBm

Table 2.1: SDR comparison table [7, 8, 1]

full-duplexing, high-performance FPGAs, wide frequency range (Even supporting the mm-Wave), wide baseband bandwidth, MIMO or compatibility with existing development tools.

Other SDR units are intended to be used by students, makers or even engineers; basically, everyone who is budget is not large enough to get a device from the previously mentioned manufacturers. Platforms such as the RTL-SDR, LimeSDR, HackRF, BladeRF and PlutoSDR are examples of devices that target this budget range. These devices are examples of open-source projects (Excepting the Pluto SDR) maintained by the community, some with modest capabilities and compatibility. Meanwhile, companies like Analog Devices back the design and production of their Pluto SDR, which is intended for students to learn digital communication systems.

2.3 Beamforming in Software-Defined Radio

Since the essential requirements of Beamforming mostly rely on the placing of the antenna array and not so much on signal processing, it is suitable to be implemented in SDR systems. Of-the-shelf SDR devices already have the required hardware to drive multiple antennas, making of Digital Beamforming the most convenient of the three alternatives. Implementation as in [9, 10] used specialised hardware in order to set up the multiple SDR units quickly, and LabVIEW was used as the interface to control and monitor the antenna array; showing the capabilities of high-end SDR equipment. Low-cost SDRs are not as flexible as the previously mentioned as additional effort, knowledge and sometimes hardware modifications are required to do a similar implementation as [9, 10] in these devices.

Many non-ideal conditions can result in beamformers showing erratic behaviour or can even break them. Impairments such as correlation error, mixer frequency shifts or non-synchronised oscillators among others must be considered when designing a Digital Beamformer [11]. Fortunately, SDR manufacturers have already solved most of the previously mentioned impairments; however, the most critical impairment to consider when using SDR based beamformers is related to synchronisation.

2.3.1 Coherence and Synchronisation

According to [11, 12], Digital beamforming requires both transmitter and receiver to be coherent or pseudo-coherent; this implies that the phase between channels must be consistent, zero for pure coherent systems or constant for pseudo-coherent. Also, the sampling clock of the ADC/DAC must be perfectly aligned between channels. Distributing a LO signal to every SDR or RF front-end is the easiest way to synchronise the relative phase of each properly, but noise in the distributed signal can significantly affect the performance of the beamformer. An alternative to this is to keep each SDR LO and distribute a low-frequency trigger signal that keeps every LO aligned without having a noiseless signal source.

2.3.2 Interfacing

Transfer speed is an important parameter to consider, as receiving data from multiple antennas does increase the required interfacing bandwidth from the SDR to the processor in a factor of the number of antennas; for example, having a four-element antenna array each one sampled at 10 MSPS transferring the IQ samples as a 32-bit integer type results in the following required throughput.

$$R_{req} = (4 \text{ antennas}) (32 \frac{\text{bits}}{\text{sample}}) (10E6 \frac{\text{samples}}{\text{second} \times \text{antenna}}) = 1.28 \text{ Gbps}$$

The past example requires a total of 1.28 Gbps, which is much higher than the maximum theoretical speed of USB2.0 (480 Mbps), meaning that a faster protocol is required such as USB3.0 (5 Gbps - 20 Gbps) or Ethernet (1 Mbps - 400 Gbps) [13, 14, 15]. On top of that, the real throughput of these protocols dramatically depends on the frame size. However, designers can choose the sample rate, and SDRs have different bit-depths, so the required throughput differs from one device to another.

2.3.3 Transceiver Architecture

As previously stated, SDRs implement external transceiver ICs that use the Direct-Conversion architecture as shown in figure 2.4, meaning that signal can only be processed in baseband. This design decision constrains the beamforming type to Phase-shift Beamforming, in which the time-delays shown in equations 2.3 and 3.7 are treated as phase shifts in the IQ signals. However, Phase-Shift Beamforming introduces the maximum steering angle, as shown in equation 2.9 and non-ideal radiation patterns. This

limitation is caused by the phase coherence in the TX or RX channel, as the real beamforming equations consider a time-delay between the carrier signals and the resulting signal is shown in equation 2.11. [16]

$$y_m(t) = I(t - \tau_m) \cos(\omega t) - Q(t - \tau_m) \sin(\omega t) \quad (2.11)$$

$$y_m(t) = I(t - \tau_m) \cos(\omega(t - \tau_m)) - Q(t - \tau_m) \sin(\omega(t - \tau_m)) \quad (2.12)$$

More flexible architectures as the Direct Sampling in figure allows True Time-Delay Beamforming, which allows the signal to be modified even in broadband resulting in a transmitted or received signal similar to equation 2.12, removing the steering angle constrain of 45° before getting grafting lobes, assuming an optimal beamformer. As this beamforming type seems to be more effective than Phase-Shift Beamforming, it requires a shallow sampling rate in the order of picoseconds, which is unobtainable for low-cost ADC/DAC.

Chapter 3

Methodology

3.1 Introduction

This chapter describes the work done to accomplish the objectives in section 1.3. The work done for this project is divided into several disciplines, being these listed as follows.

- Simulation
- SDR selection
- MATLAB interface for SDR
- Antenna mounting
- HIL Demo

The previous bullets are not in chronological order as several stages were being developed simultaneously, but the result of this section is a demonstration to show the test bench capabilities.

3.2 Simulation

A simulation model of an end-to-end communication system was made to test the beamforming algorithm in both transmitter and receiver sides. As the objective of this section is to deliver a usable beamforming model supporting both TX and RX arrays, the HIL approach is appealing, as the subsystems of the simulation can be reused in the HIL model with a few modifications. This simulation was done using Simulink since it supports many algorithms and models commonly used in digital communications and also supports HIL workflows with many SDRs.

Many iterations of the end-to-end simulation were created, including more features and with real-life impairments and characteristics. The simulation and its subsystems were developed simultaneously and then tested in different models, as described in chapter 4. The final iteration of the simulation model includes the following subsystems. The resulting model is shown in figure 3.1.

- UE movement pattern
- Modulator
- Receiver
- TX beamformer
- RX beamformer
- Wireless channel

Similar to the subsystems, a setup MATLAB script was created to manage the block's parameters in the system and to automate changes to many parameters. The script will be described as the equations related to each subsystem.

Simulation results are available in chapter 4 as the model described in this section is generic and many parameters depend on the selected SDR.

3.2.1 Problem Statement and Initial Assumptions

The simulation model assumes two devices are located in space, the base station (BS) and the User Equipment. The BS is located at the origin of a coordinate system and is unable to move in any direction. The UE is located at a distance d_{UE} and is allowed to rotate around the BS. The UE is oriented in a constant direction Az_{UE} , which represents the azimuth angle of the UE (absolute orientation, relative to the coordinate system). Finally, the UE and BS are placed at the same elevation.

The BS consists of two M-element antenna arrays for both TX and RX stages and a processing unit. The UE has a single M-element RX antenna array, single antenna TX stage and a processing unit. All the antenna arrays in the model are assumed to be optimal ULAs composed of isotropic antennas.

The goal of this system is for the BS to track and lock the UE using both of its antenna arrays in order to maintain a constant Bit Error Rate in the end-to-end communications system. Besides, the UE can use beamforming at the receiver end to enhance the Signal-to-Noise (SNR).

The challenge for both devices is to locate each other's direction using their transmitted signals. This model assumes that all the antenna arrays are placed along the y axis; antenna elements overlap in the case of the BS, as it has two arrays. The UE's location relative to the BS will be called the Direction of Departure (DOD), as it is the angle at which the BS must steer the TX beam to reach the UE. Similarly, the BS's location relative to the UE's orientation will be called the Direction of Arrival (DOA), as it is the orientation at which the UE must steer its RX beam to capture the BS's signal optimally. Both angles are measured from the positive x axis to the positive y axis.

The UE's position relative to the BS is an arbitrary function of time. The sinusoid gives the rotation function in equation 3.1 where DOD_{max} is the maximum angle of departure, and f_{UE} is the frequency of the sinusoid; both of these are input parameters. As this simulation is a discrete system, it must account a sampling time T_s and a sample number n . The DOA can be easily derived by adding the

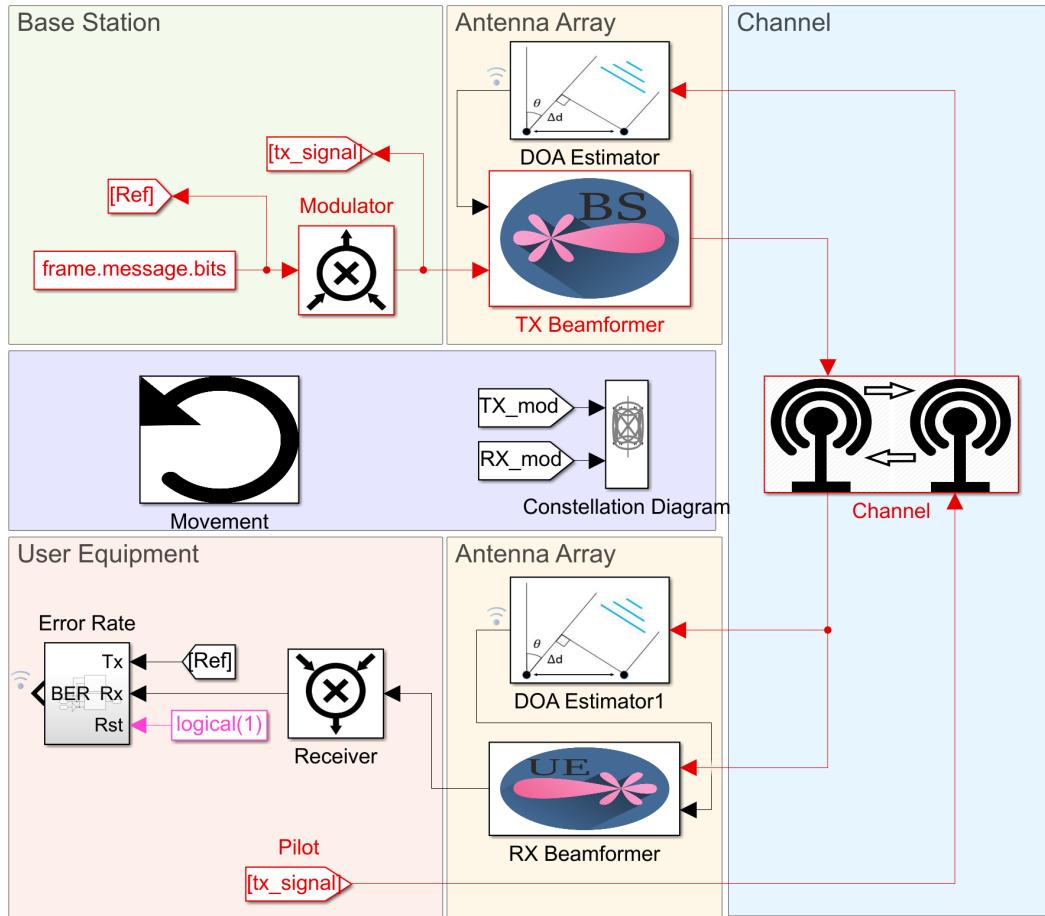


Figure 3.1: Simulink model for the UE tracking simulation.

UE's azimuth to the DOA. The constraint in equation 3.1 restricts the DOA to a range of $\pm 90^\circ$; as this is the input range of the blocks in the following sections.

$$DOD(n) = DOD_{max} \sin(2\pi n T_s f_{UE}), |DOD_{max} + Az_{UE}| < 90 \quad (3.1)$$

$$DOA(n) = DOD(n) + Az_{UE} \quad (3.2)$$

Figure 3.3, shows the Simulink model for the system kinematics. The Sine Wave and subtraction blocks represent equations 3.1 and 3.2, and there also is a manual switch for keeping a constant DOD. Also, a visual representation of the simulated environment is shown in figure 3.2.

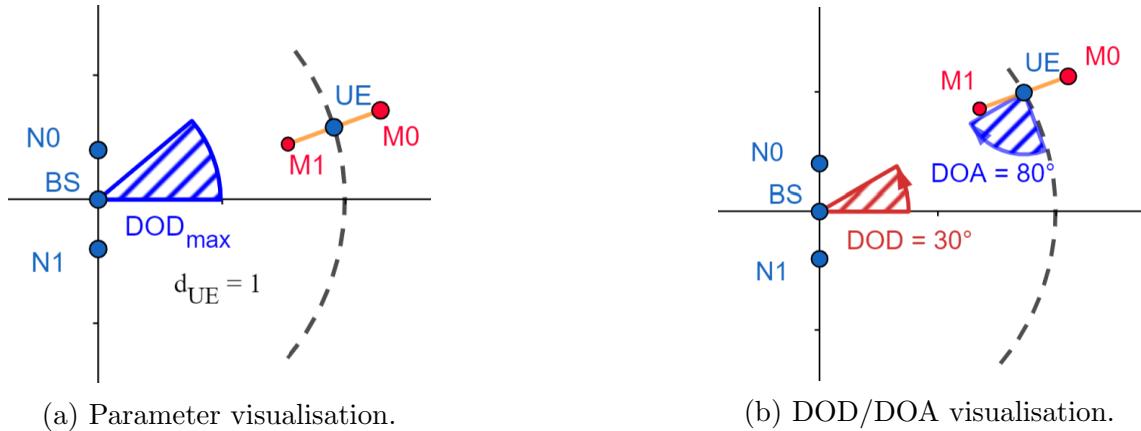


Figure 3.2: Example of the coordinate system assuming $d_{UE} = 1$, $Az_{UE} = 20^\circ$ and $M = 2$. Points M0 and M1 are the antenna array.

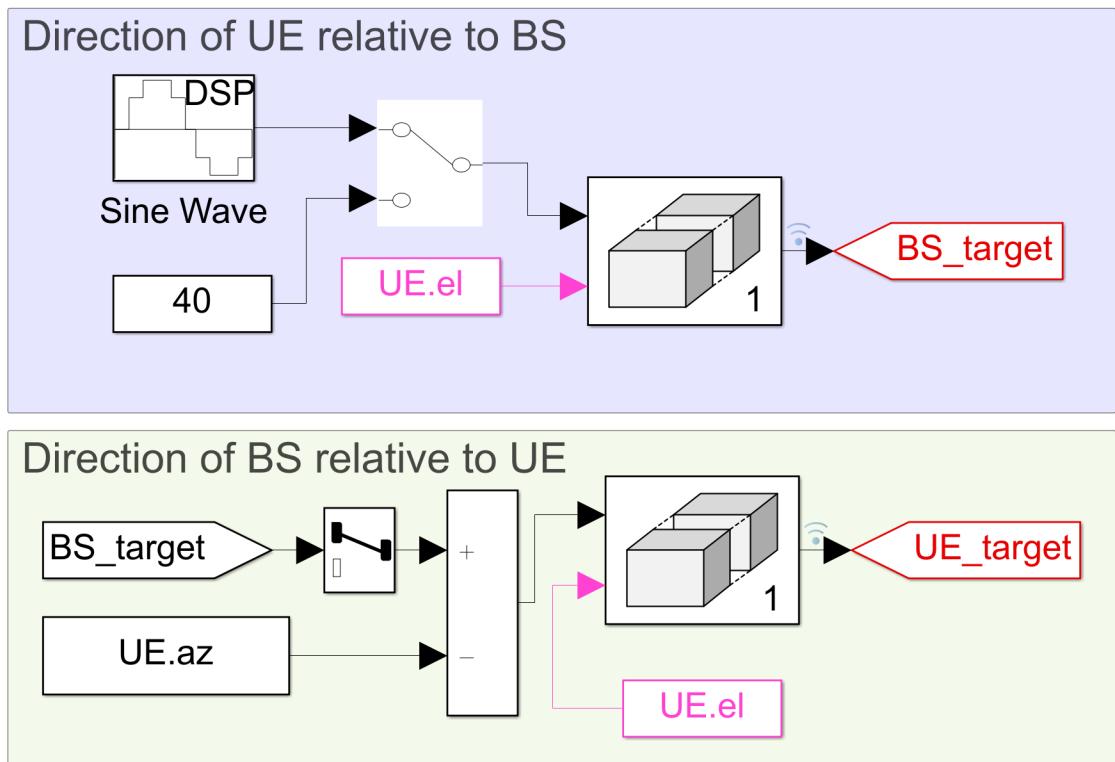


Figure 3.3: Simulink subsystem for the UE's movement.

3.2.2 Modulator and Receiver

The purpose of this section is to define the parameters and techniques used in the wireless communication system that were used around the beamforming algorithm. As this simulation will later involve actual hardware in it, the subsystems must consider that SDRs are discrete systems that operate using frames with many symbols per iteration. Similar to the derivation of equation 3.1, the simulation model requires a fixed time step T_s in which the modulator and receiver algorithms are executed.

Modulator

The simulation is supposed to transmit a frame of N_M message bits; this sequence remains constant through the simulation time. This sequence is meant to be processed and modulated so the receiver algorithm can output the same message sequence. This task is done in the Modulator subsystem, which performs the following actions: Scramble, add preamble, modulate and interpolate. These actions are done in the same order as listed.

The scrambler is used to mix the message bit in case these are many consecutive appearances of a bit pattern, particularly helping the Automatic Gain Controller stage at the receiver. The preamble is a constant bit sequence that helps the receiver know where a frame starts; the length of the sequence N_P is essential as the larger it is, the easier it is to detect it, at the expense of occupying more space in the sent frame. After both scrambled message and preamble sequences are added together, the resulting sequence is then modulated using QPSK, outputting a new sequence of N_S complex symbols.

$$N_S = \frac{N_M + N_P}{2} \quad (3.3)$$

The last step in the modulator block is interpolating the symbol sequence. This step is vital as the SDR introduces a baseband sampling rate (f_{bb}), which in many units has a relatively high minimum value. The up-scaling is done using a Raised Cosine (RC) filter, which also reduces the Inter-Symbol-Interference. The RC filter can interpolate the symbol sequence to satisfy any sample rate while reducing ISI by slightly increasing the signal's bandwidth using the RC's roll-off factor (β). The interpolation factor (N_{int}) is given by equation 3.5, which uses the simulation's symbols per time step ($N_{s/s}$); also, the RC filter is an IIR filter, its span must be limited.

The modulator algorithm is coded into Simulink, as shown in figure 3.4.

$$N_{s/s} = T_s f_{bb} \quad (3.4)$$

$$N_{int} = \frac{N_{s/s}}{N_S} \quad (3.5)$$

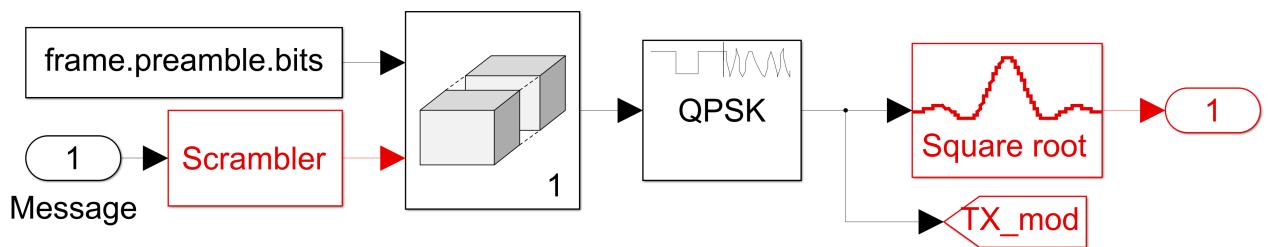


Figure 3.4: Simulink subsystem for the Modulator

Receiver

The receiver part includes the counter-operation of every step of the modulator and some other stages to consider the channel and timing impairments properly. This block's task is to output the message bits that were the input of the Modulator block. The stages are executed as follows: Decimation, normalisation, carrier synchronisation, symbol synchronisation, frame synchronisation, phase correction, preamble removal, de-modulation and de-scrambling.

Similar to the modulator, the Raised Cosine filter improves ISI while decimating the received signal; the only difference in this stage is that the decimation factor N_{dec} must be smaller than the interpolation factor ($N_{dec} < N_{int}$) as the following stages require an input signal with many samples per symbol. The Automatic Gain Controller (AGC) normalises the signal's amplitude, usually for its IQ component's maximum value to be unitary; AGC uses an open-loop approach that averages a sampling window, and it may not act in every simulation step, as the signal is not expected to change its amplitude very abruptly. The AGC helps the Preamble Detector to output a repeatable result as it is greatly affected by the signal's power. The carrier synchronisation step corrects the frequency and phase impairment introduced by the LOs at the TX and RX ends. This stage may introduce some errors as it has a transient state in which it tries to compensate for the signal's phase error. The output of this block should roughly match the output of the Modulator block, but a phase error could still be present; the magnitude of the phase error depends on the modulation scheme; QPSK phase error is described in equation 3.6.

$$\alpha_{error} = k \frac{\pi}{4}, k = 0, 1, 2, 3 \quad (3.6)$$

The symbol synchroniser stage uses the up-sampled signal to get a down-scaled symbol sequence that contains the sent frame. Then, the frame synchroniser stage outputs the frame by using a preamble detector. The detector uses the magnitude of the cross-correlation function to get the position of the frame after the symbol synchronisation; then, it is fed to the frame synchroniser. Using the cross-correlation approach also discards the remaining phase error from the carrier synchronisation stage. The frame synchroniser outputs the frame and whether it is valid or not. [17]

Finally, if a valid frame is detected, the phase correction stage removes the phase error from the carrier synchronisation by averaging the phase difference of the preamble of the received frame and the intended preamble, then the average is rounded to match the expected phase error for QPSK. Finally, a phase shift is applied to the entire received frame, so then the preamble can be removed, demodulated and de-scrambled. [17]

The resulting Simulink model for the receiver is shown in figure 3.5 and 3.6, which represents a subsystem within the receiver block.

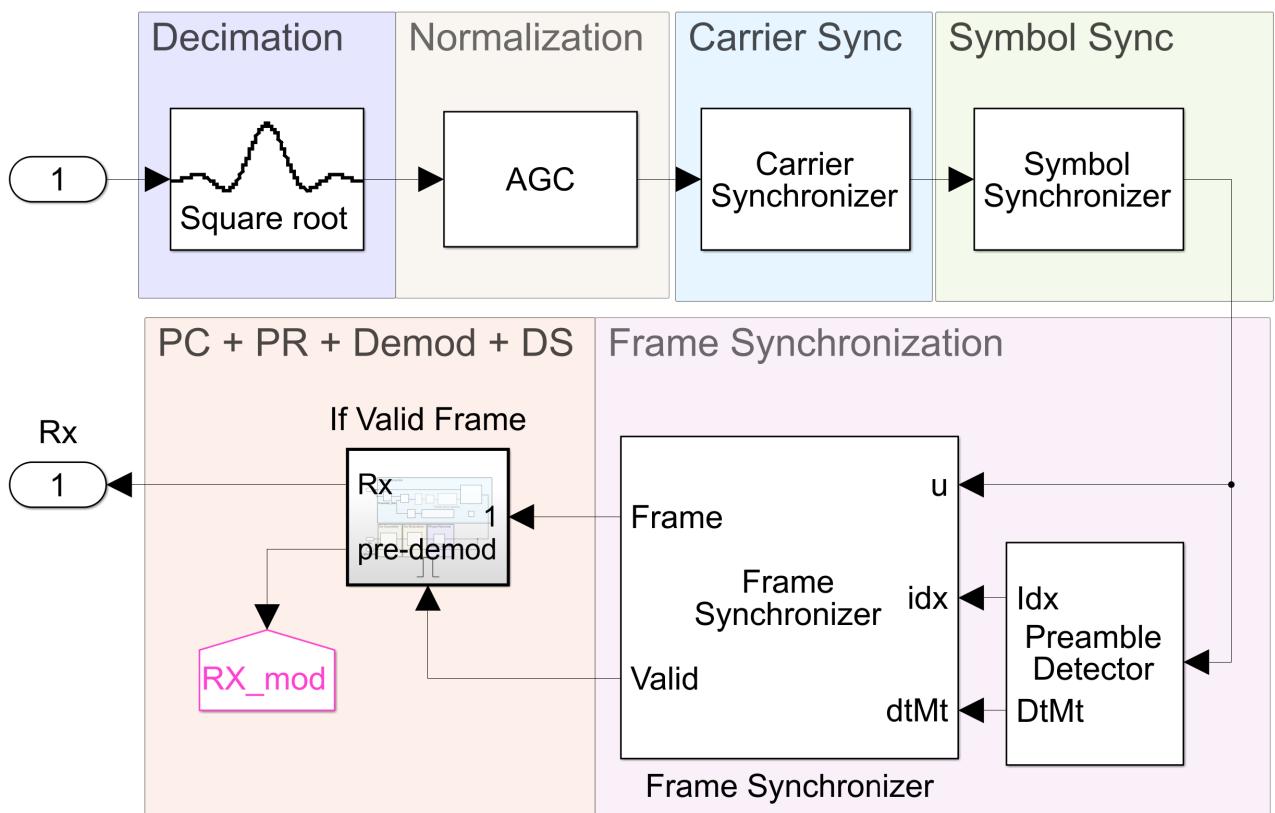


Figure 3.5: Simulink subsystem for the Receiver.

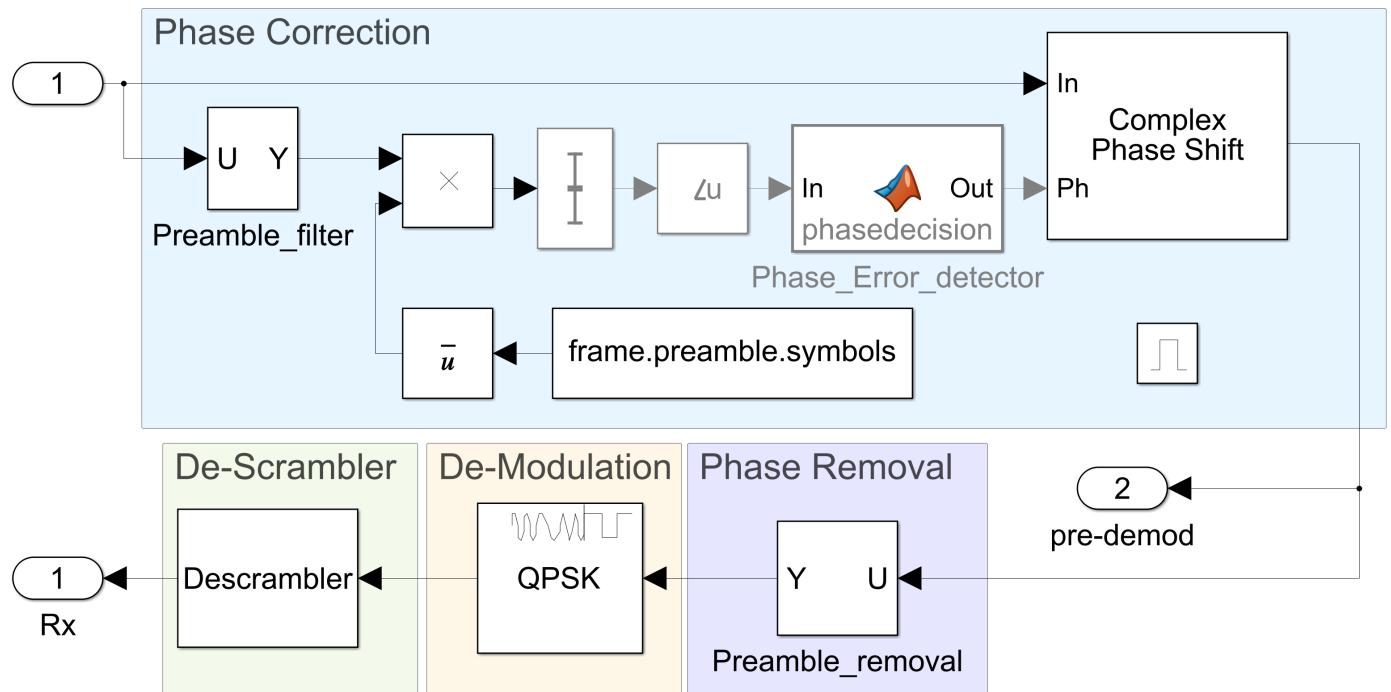


Figure 3.6: If-valid-Frame Simulink subsystem shown in figure 3.5.

3.2.3 Wireless Channel

An important part of every communications system simulation is the channel model. As this simulation model intends to prove the efficiency of the beamforming algorithm, it only includes Additive White Gaussian Noise (AWGN) components. Some experiments in later sections may include the Simulink subsystem in figure 3.7 and other impairments.

The channel subsystem also features the antenna array emulators for the UE and BS, as they affect the signal in different ways. The TX antenna array receives the signal coming from the TX beamformer, which is a $[N_{s/s} \times M]$ matrix. This array modifies the signal's gain depending on the actual location of the UE and the beamformer's steering angle; when the TX steering angle roughly matches the UE's angle, the signal will be amplified; otherwise, it will be attenuated.

The RX antenna arrays split the incoming signal into M different signals, one for each antenna element. The AWGN impairment must be added to the split signal, for them to have different added noise signals. The different noise in each element impacts the DOA estimation. The RX antenna array also receives the actual DOA relative to the source device, which will attenuate or amplify the signal depending on the UE's location and the beamformer's steering angle. The receiver antenna array emulator also adds the corresponding phase-shift to the signal sensed by each antenna element according to the beamforming equations in section 2.1.2.

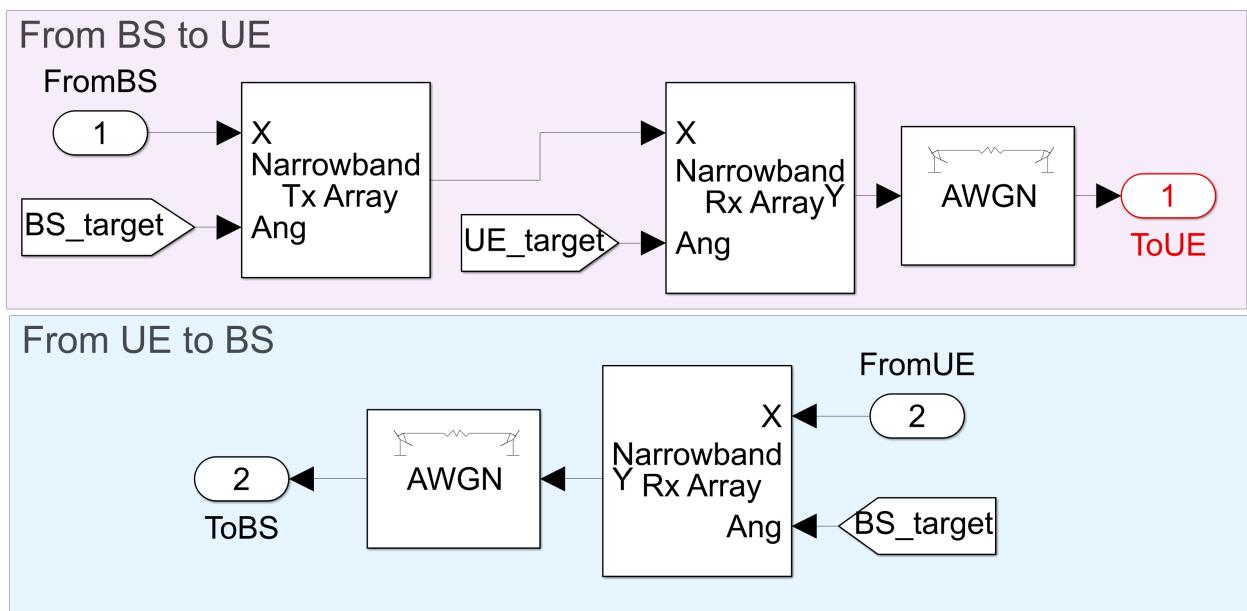


Figure 3.7: Simulink subsystem representing the wireless channel.

3.2.4 Beamformers and Direction of Arrival Estimation

This section describes the design and functionality of the antenna array algorithms as all three of them (two in the BS and one in the UE) perform different tasks. As stated in section 3.2.1, the objective is for the BS to track the UE's movement in order for it to steer its TX beam towards the UE and guarantee a stable link. This task is done by the BS estimating the DOA using the UE's Tx signal and then feeding that signal to the TX beamforming algorithm. Also, the UE supports RX beamforming and is expected to estimate DOA from the BS's TX signal and steer its RX beam towards the BS.

The steering vector of both beamformers is left as an input to make the blocks generic and suitable for other uses as non-adaptive beamforming.

Receiving Beamformer at the User Equipment

The beamformer at the UE shown in figure 3.8 uses the matrix form of equation 2.2 now presented as equation 3.7, as it only needs an $[N_{s/s} \times M]$ matrix representing the signal coming from each element of the antenna array (Obtained from the channel subsystem in section 3.2.3) and $[M \times 1]$ vector representing w_m . This equation results in a $[N_{s/s} \times 1]$ vector representing the beamformed signal.

$$\mathbf{y}(n) = \mathbf{x}(n)\mathbf{w}^* \quad (3.7)$$

Simulink provides a Phase-Shift beamformer that works for RX beamformers. This block receives the $\mathbf{x}(n)$ matrix and a vector containing the DOA and elevation angles; the elevation is assumed to be zero, as stated in section 3.1. The beamformer block is used to output the steering weight vector, but only the exponential part of the complex number as in equation 2.7. The block is used this way to enhance its flexibility, as its weights can be combined with a custom A_m vector. As the simulated antenna array obeys equation 2.5, side lobes are not expected, so the magnitude of the complex vector \mathbf{w} is defined as an M-element unitary vector. This system is shown in figure 3.8.

Transmitting Beamformer at the Base Station

This beamformer in figure 3.9 is very similar to the RX beamformer, the input and output signal dimensions are inverted, as this algorithm expects a $[N_{s/s} \times 1]$ matrix as input and delivers a $[N_{s/s} \times M]$ matrix. The input vector is the signal coming from the Modulator block, and the output matrix columns are the signal to be fed to each antenna element in the array.

Again, the beamformer block is used to generate the steering vector and recreate the beamforming equations manually. This block requires a $[N_{s/s} \times M]$ matrix only to output the corresponding weights, which is generated by multiplying the input $[N_{s/s} \times 1]$ vector times a M-element row vector. The expected behaviour of this subsystem is to receive an estimated DOA from the local RX beamformer that matches the real DOD for the signal being transmitted from the BS is sent towards the UE. This subsystem is shown in figure 3.9. The resulting equation is given by equation 3.8 where \cdot^H represents the Hermitian operator.

$$\mathbf{y}(n) = \mathbf{x}(n)\mathbf{w}^H \quad (3.8)$$

Direction of Arrival Estimation

The estimated DOA is obtained using the RX antenna array in the BS and UE. This array is not linked to any beamforming algorithm, but the former is exclusively used to estimate the position of the UE. The Simulink subsystem for the DOA estimation is shown in figure 3.10.

The DOA estimator is based on the MUSIC algorithm as this one is more accurate and has a higher resolution than the ESPRIT algorithm [18]. This subsystem requires the ULA's parameters to estimate the MUSIC spectrum properly. As stated before, the peaks of the MUSIC spectrum are considered as the DOA. This system is shown in figure 3.10. The output of this block may be used as the steering angle for both beamformers.

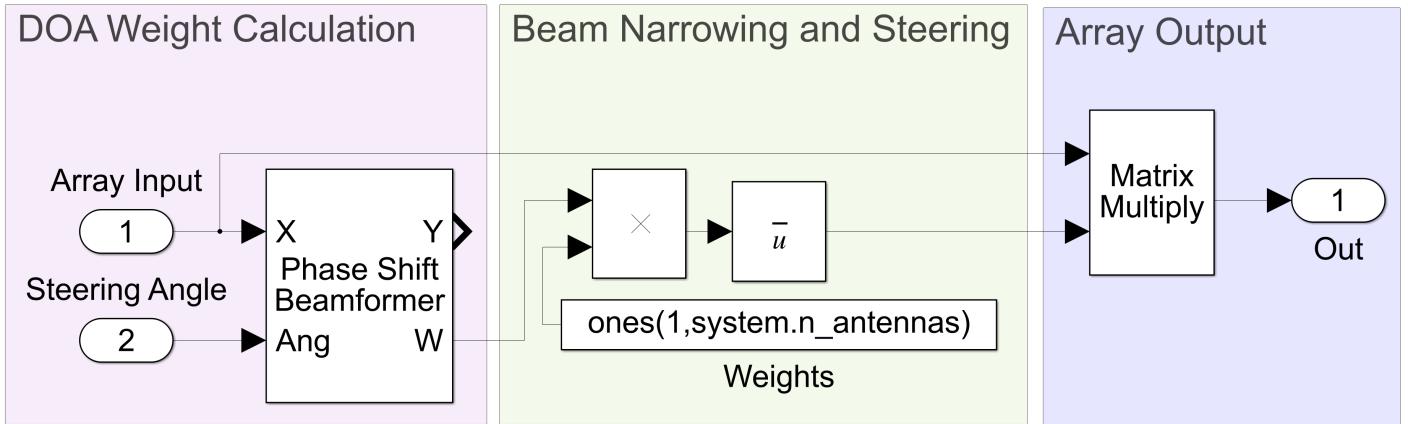


Figure 3.8: Simulink subsystem for the RX beamformer at the UE

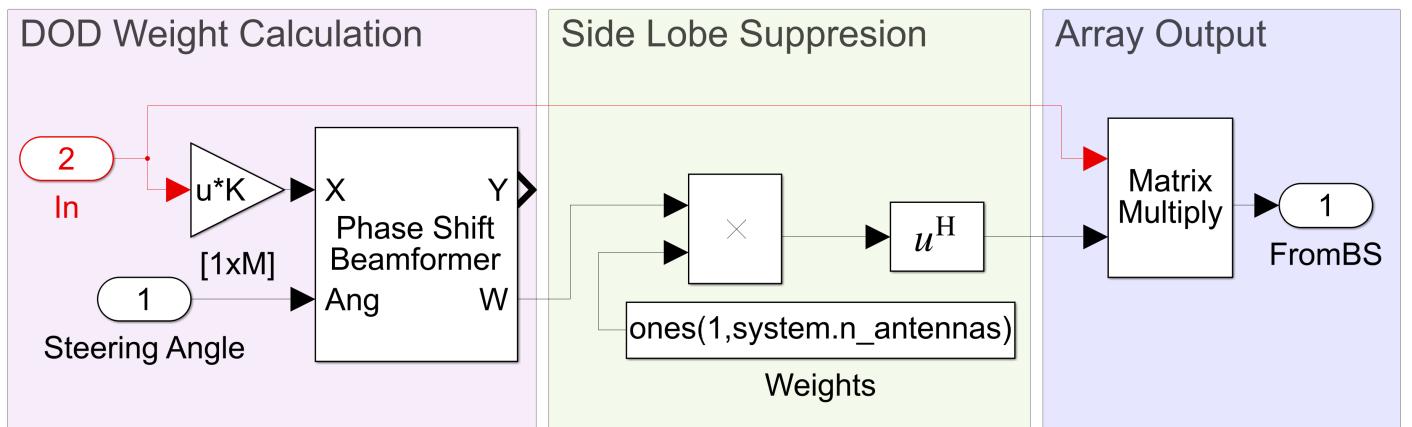


Figure 3.9: Simulink subsystem for the TX beamformer at the BS

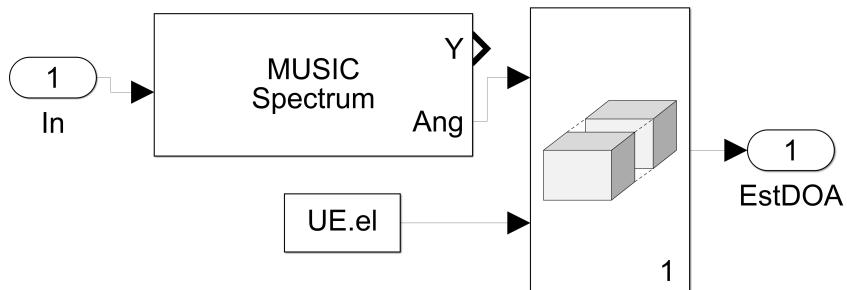


Figure 3.10: Simulink subsystem for the DOA estimation.

3.3 SDR Selection

3.3.1 Introduction

This section describes the requirements of the beamforming algorithm for it to be executed in a SDR and the characteristics and features of the selected unit. The filtering and analysis of each unit was done based in the content from sections 2.1.2, 2.2.3 and 2.3. The best SDR candidates are listed in table 2.1.

3.3.2 Requirements

As stated in section 2.2.3, many SDR units are varying in cost and performance. Some of the presented options are low-cost and intended to be used for learning or just as a cost-effective alternative to higher-end products as the Ettus Research devices.

MIMO

Beamforming is a spatial diversity technique, meaning that the most obvious requirement for the SDR to have is MIMO capabilities (More than one antenna at the transmitter or receiver end). This requirement is the most important, as the beamforming equations require many coherent signal sources.

SDRs such as the Ettus B210, LimeSDR and BladeRF would be obvious choices considering the previous constraint; as all of these support 2x2 MIMO or even 4x4 as is the case of the PCI version of the LimeSDR. However, hardware modifications and can be done to the HackRF [19], RTL-SDR [20, 21, 22] and PlutoSDR [23] to support coherent streams. These modifications vary in difficulty and resulting performance and may not be suitable to modify a device only to serve this purpose permanently.

Transfer Speed

As stated in section 2.3.2, spatial diversity applications require high-speed data transfers to guarantee a deterministic performance. Many parts of the end-to-end simulation, especially the receiver, would be broken if the SDR's RX buffer overflows. A reliable data transfer is expected from the SDR even using its MIMO scheme; this requires a reliable communication protocol such as USB3.0, PCI and Ethernet.

Again, the mid-high tier SDRs satisfy this requirement, as the Ettus B210, LimeSDR and BladeRF all support USB3.0.

Software Support

This requirement is essential if the development cycle is based on MBD or HIL. The community maintains many of these SDRs, so their software support is not official and is not expected to work flawlessly or to exploit all the capabilities of the device. This requirement is satisfied by the most available and the units backed by companies such as the Ettus B210, RTL-SDR, PlutoSDR and BladeRF.

3.3.3 Selected Hardware

Only two of the listed SDRs satisfied all three requirements: The Ettus Research's B210 and Nuand's BladeRF 2.0 A4. The latter SDR is the one selected as the B210 is considered a baseline to what low-cost SDR unis should offer.

The BladeRF is a mid-tier SDR containing an Altera Cyclone IV FPGA, an Analog Devices' AD9361 RF transceiver (as the one shown in figure 2.5) and a Cypress FX3 USB3.0 SuperSpeed controller; as shown in figure 3.11. It supports 2x2 MIMO and also can be linked to another BladeRF, maintaining

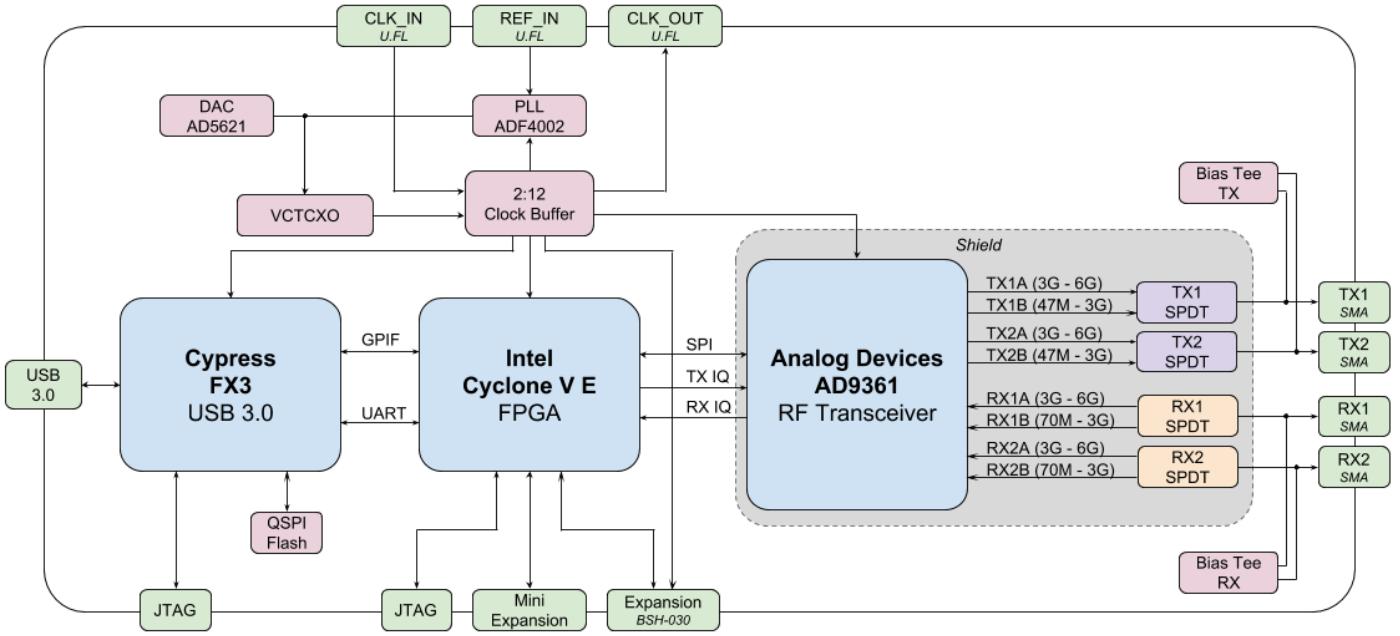


Figure 3.11: BladeRF 2.0 A4 block diagram. [1]

Parameter	Minimum	Nominal	Maximum
MIMO (M)	-	2	-
Sampling Rate (f_{bb})	521 kSPS	-	61.44 MSPS
TX RF Tuning Frequency (f_c)	70 MHz	-	6 GHz
RX RF Tuning Frequency (f_c)	47 MHz	-	6 GHz
TX Power	-	-	8 dBm

Table 3.1: BladeRF's RF specifications. [1]

pseudo-coherence to achieve 4x4 MIMO. Finally, it has support for C/C++, Python, GNU-Radio and MATLAB/Simulink (1x1 SISO).

The simulation scenarios in chapter 4 consider this SDR's characteristics and features in the equations in section 3.2, these are shown in table 3.1.

3.4 Hardware Interface for MATLAB/Simulink

The BladeRF supports MATLAB and Simulink bindings of the C/C++, allowing using the SDR within these software packages. As the first tests using the SDR, a communication system based on the Modulator and Receiver block was developed. By using the previous blocks with the Simulink system-object provided by Nuand, it was certain that these SDR blocks do not support 2x2 MIMO, which was a requirement for the following tests. The problem that this section tackle is the development of a MIMO interface for the BladeRF so that it can be used in the demonstration model.

The resulting MATLAB and Simulink scripts are based on the provided source code and the documentation for the C/C++ library. Although this script is built based on what is provided, the community interested in the BladeRF would greatly benefit from this effort.

3.4.1 libbladeRF

Nuand's libbladeRF is the official interface library for the BladeRF in C/C++. Their Doxygen site [2] describes the use of each module of the API, including some examples of 1x1 setup and boilerplate code. The template code describes a setup sequence to follow and how to use the synchronous interface for transmitting and receiving signals; this sequence is described in the flowchart in figure 3.12.

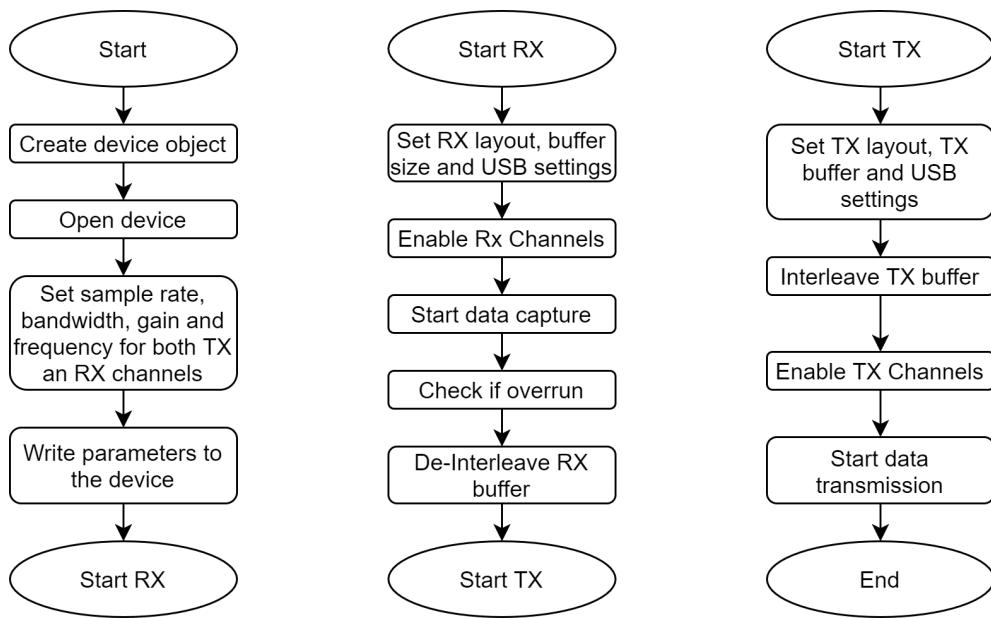


Figure 3.12: Device setup and utilisation. [2]

Initial Configuration

The first step of the template is to create a device object using the `bladerf_devinfo` and `bladerf` structures and the `bladerf_init_devinfo` function. Then, the sample rate, bandwidth, gain and tuning frequency are written to each channel using the `bladerf` structure; this process must be repeated for every intended channel, being four times for 2x2 MIMO. After this step, the individual RX and TX front-ends can be configured and used. [2]

Front-End Configuration and Data handling

The next step is to configure both TX and RX channels by using the `bladerf_sync_config` function and the device structure. The required parameters for this functions are the layout, the USB transfer parameters and the sample format; the first is a macro definition which defines the layout the front-end will have (Using RX0, RX1 or both; same logic with TX) and the second are four parameters that describe the USB transfer buffer size, number of buffers and timeout. Lastly, the format parameter denotes the use of metadata for synchronous transfers. After configuring the RX front-end, the user must enable the channels configured in the last step of section 3.4.1 using the `bladerf_enable_module`.

The receiving process is triggered using the `bladerf_sync_rx`, which fills the RX buffer with the received samples. The number of received samples parameter in the previous function is the total number of samples received for all the channels, and the RX buffer must be twice the size to hold the samples if both channels are used. The received samples are given in the Signed-Complex 16-bit Q11 (SC16Q11) format and must be de-interleaved as described in section 3.4.1 before processing them. The overrun flag may be set if the size of the received data does not equal the expected size. [2]

The transmission is done similar to the reception, and the TX buffer must be interleaved to match the SC16Q11 format and then transmitted using the `blade_sync_tx` function.

Signed-Complex 16-bit Q11 Format

This format is widely used in industrial instrumentation and is a standard way to organise data from several ADCs or going so several DACs. BladeRF's libbladeRF handles all the data going through the ADCs and DACs using this format, meaning that the user must make sure that the TX buffer is interleaved before calling `bladerf_sync_tx` and de-interleave the received samples after calling `bladerf_sync_rx`.

The SC16Q11 format uses 16-bit signed integers to represent the I and Q components of the samples and packs both numbers into a single 32-bit integer. This packed sample is little-endian, and the first 16 bits correspond to the I sample and the last 16 bits to the Q sample. The order of samples is also important for the format, as the first 32-bit integer in the buffer represents sample 0 of channel 0, the second integer represents sample 0 of channel 1 and so on. A summary of the SC16Q11 format is given in table 3.2. [2]

Byte offset	Bit 31 - 16	Bit 15 - 0	Description
0x00	$Q_0(0)$	$I_0(0)$	Ch 0, sample 0
0x04	$Q_1(0)$	$I_1(0)$	Ch 1, sample 0
0x08	$Q_0(1)$	$I_0(1)$	Ch 0, sample 1
0x0C	$Q_1(1)$	$I_1(1)$	Ch 1, sample 1
0x10	$Q_0(2)$	$I_0(2)$	Ch 0, sample 2
0x14	$Q_1(2)$	$I_1(2)$	Ch 1, sample 2
:	:	:	:
$2(N_{s/s} - 1) \times 0x04$	$Q_0(N_{s/s} - 1)$	$I_0(N_{s/s} - 1)$	Ch 0, sample $N_{s/s}$
$2(N_{s/s} - 1) \times 0x04 + 1$	$Q_1(N_{s/s} - 1)$	$I_1(N_{s/s} - 1)$	Ch 1, sample $N_{s/s}$

Table 3.2: SC16Q11 layout example for 2 channels. [2]

3.4.2 MATLAB System Object

The MATLAB script provided by Nuand performs a process similar to the one described in the previous section. The main difference is that only a single channel in either the TX or RX front-is active at

any time, defying the definition of the MIMO scheme. Since the MATLAB script is a binding script to the libbladeRF API, the same approach described in section 3.4.1 can be programmed while conserving most of the existing code. The challenge of this step was to read through the existing MATLAB script, find where each section of the algorithm takes place and do the necessary changes.

The changes made to the original version of the script were copied into a separate file with a different system object called `bladeRF_MIMO`. This new object performs the initial steps, but for both channels, always. The entire front-ends can be enabled or disabled independently, but if enabled, both channels are active, and all the configuration is applied to both channels; although the gain can be set independently if the onboard AGC is not enabled. The channel layout macro was changed at the front-end configuration to use both channels simultaneously. Also, the interleaving and de-interleaving stages had to be handled manually in MATLAB.

The new script now can receive and output a $[N_{s/s} \times 2]$ matrix, where each column is the data being transmitted or received by each channel and the rows represent the sample number.

Simulink requires an additional wrapper object for the object to be compatible with the other blocks, as Simulink often compiles the blocks in the model for them to be executed efficiently, opposed to MATLAB's interpreted execution.

3.5 Antenna Mounting

Beamforming requires that the antenna elements in the array are placed at a constant distance given by equation 2.5 to avoid spatial aliasing. Placing the antennas at a known distance is also a requirement for the DOA estimation, which requires the array distancing and operative frequency to estimate the DOA properly. These reasons were enough to build a structure that holds the SDR's antennas in place. This structure was designed using Fusion 360 (A CAD/CAM/CAE software) to make the manufacture and reproduction of this work easier.

The CAD model; similar to the mathematical model in the simulation (section 3.2) is a parametric model that changes its geometry according to the input parameters. These parameters are the carrier frequency f_c and number of antenna elements M . The carrier frequency determines the distance between elements as described in equation 2.5, as it assumes the placement to be optimal. Other parameters as the mast height and the lip height are considered input parameters.

$$d = \frac{\lambda}{2} = \frac{c}{2f_c}$$

The structure has three main components, a circular base, mast and the beam. The base holds the structure and its diameter must maintain the antenna array in place without falling. The mast links the base and the beam; while routing the cables to each antenna element inside of it. The beam holds the antennas using little indentations in the lip above the beam. The antennas are supposed to be placed using zip ties. Most of the dimensions of depend on the f_c and M as shown in figures 3.13.

Parameter	Formula	$f_c = 890$ MHz	$f_c = 430$ MHz	$f_c = 2.4$ GHz, $M = 4$
Antenna distancing	$d = \lambda/2$	168.42	348.59	62.45
Beam length	$L_{beam} = 1.15 d (M - 1)$	197.72	409.25	73.32
Beam width	$W_{beam} = 0.35 (M - 1)^{-0.7}$	69.20	143.23	25.66
Mast diameter	$d_{mast} = 0.625 W_{beam}$	43.25	89.52	16.04
Base diameter	$d_{base} = 0.8 L_{beam}$	158.18	327.40	58.69

Table 3.3: Formulas used to determine each length in the structure and examples for different frequencies. Units are in millimetres and $M = 2$ unless specified.



(a) $f_c = 890$ MHz, $M = 2$ (b) $f_c = 430$ MHz, $M = 2$ (c) $f_c = 2.4$ GHz, $M = 4$

Figure 3.13: Examples of different antenna structures for various carrier frequencies and antenna elements. The mast height remains constant across experiments to appreciate the differences.

Two structures were manufactured using cardboard boxes, as this material was available at the time. The used parameters were $f_c = 890$ MHz, $M = 2$, mast height of 150 mm and lip length of 30 mm. The carrier frequency was chosen using an unused section in the UK's ISM band, which results in a relatively small structure.

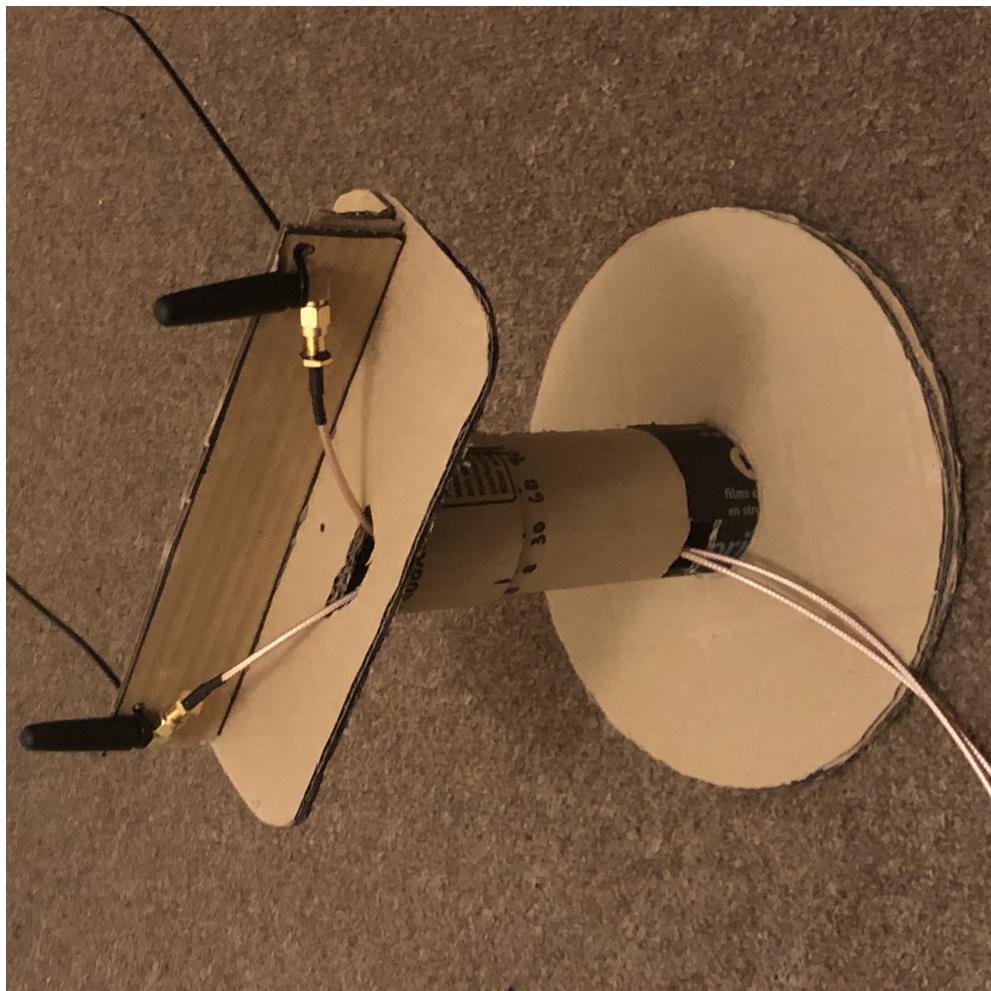
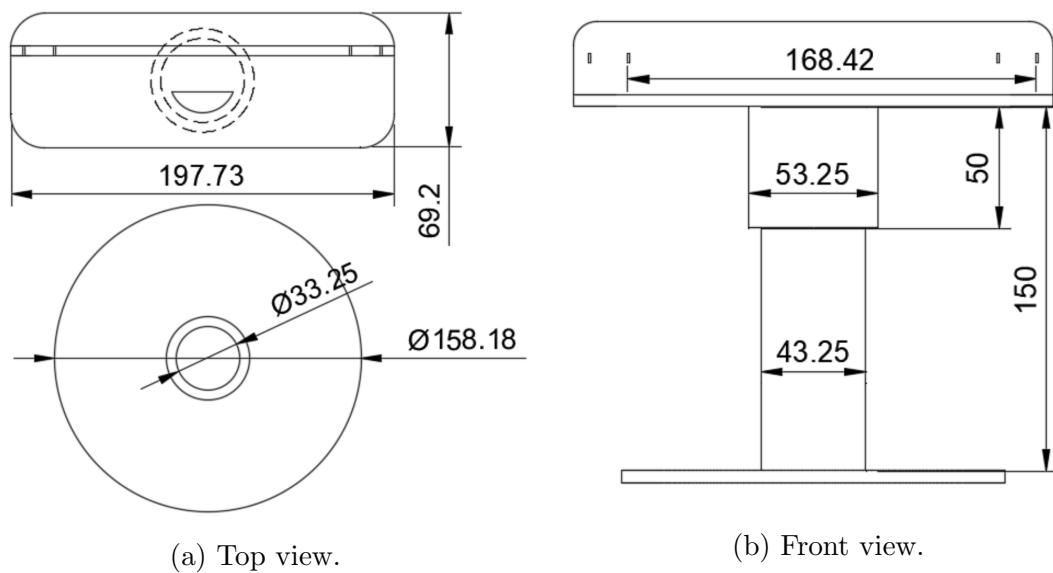


Figure 3.14: Handmade antenna structure with antennas and SMA cables.



(a) Top view.

(b) Front view.

Figure 3.15: Technical drawing of the antenna structure considering $f_c = 890$ MHz and $M = 2$.

3.6 Hardware-in-the-Loop Demonstration

The HIL demo is the combination of all the previous sections to have a template model to test individual components and further develop the UE tracker, similar to the simulation. The demo consists of transmitting and receiving a fixed frame using the BladeRF. The same subsystems used to develop the simulation in Section 3.2 were used in this section for modulating, receiving and to process the beamforming algorithm.

The user can input the DOD at the transmitting end and the DOA at the receiver or use a function similar to the one for the UE movement in section 3.2.1; and even change it from one to the other in run time. The DOA at the receiver side is estimated using the same algorithm as in the simulation, and the user can select whether to input the steering angle manually or to use the measured DOA; again, that can be alternated at run time. Some performance metrics, such as the BER and Error Vector Magnitude (EVM) and the RMS power of the beamformed signal, are logged to check the system's performance.

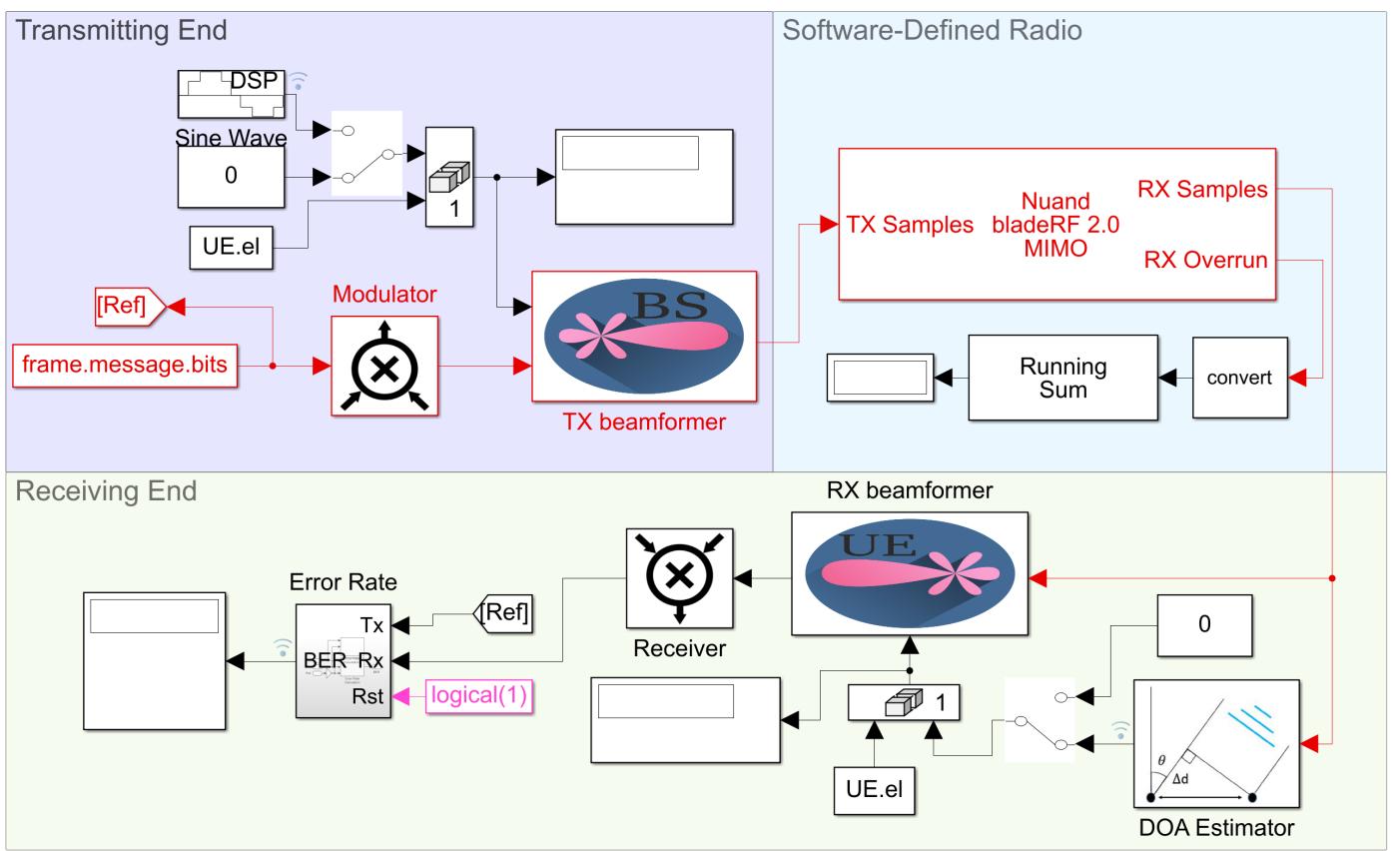


Figure 3.16: Simulink model for the HIL demonstration.

Chapter 4

Tests and Validation

4.1 Introduction

This section describes the tests and validation methods used to evaluate the products of chapter 3. Each subsystem of the system had to be tested to guarantee the system's performance, and the testing process for each is unique in order to show its strengths and weaknesses.

The tested parts are the products of sections 3.2.2, 3.2.4, 3.4.2, the end-to-end simulation described in section 3.2 and several variations if the HIL demo in section 3.6. As expected, the beamforming related tests are prioritised, as the remaining subsystems are evaluated to diagnose any discrepancy at the end-to-end simulation and real test.

Most of the variables used on for the tests are constrained by the SDR parameters and the antenna structure, resulting in the testing parameters shown in table 4.1.

Parameter	Value
Simulation step time (T_s)	0.01 s
Carrier frequency (f_c)	890 MHz
Number of antennas (M)	2
Baseband sample rate (f_{bb})	1 MSPS
Samples per simulation step ($N_{s/s}$)	10000
Number of message bits (N_m)	160
Number of preamble bits (N_p)	40

Table 4.1: System parameters for the test scenarios.

4.2 MIMO Interface for the BladeRF

The effort presented in section 3.4 refers to a MATLAB interface for using the 2x2 MIMO scheme of the BladeRF SDR. This interface is used in the following sections as the real-world performance test; hence, its performance must be confirmed to be as expected.

The test environment for the SDR is simple. The SDR is meant to capture the signal coming from an FM radio station and re-transmit it into an empty portion of the ISM band. The goal of the re-transmission is to validate both the TX and RX stages of the MATLAB interface. The validation metric for this test is to reproduce the signal coming from the BladeRF using another SDR; the test is considered valid if the audio signal is reproduced without artefacts consequence of dropout or discontinuities. Discontinuities may be caused by incorrectly interleaving the TX buffer or de-interleaving the RX buffer, having a high sample rate that affects determinism, having a large RX gain or an incorrect AGC setting. Also, the frequency spectrum of both signals can show if any discrepancy happened during the transmission. The Simulink model used for this test is shown in figure 4.1, and consists of the `bladeRF_MIMO` Simulink block in a feedback loop. The signal sensed by the second SDR is processed separately using the SDR Console console software. The RX and TX frequencies for the SDR are 92.32 MHz and 890 MHz, respectively.

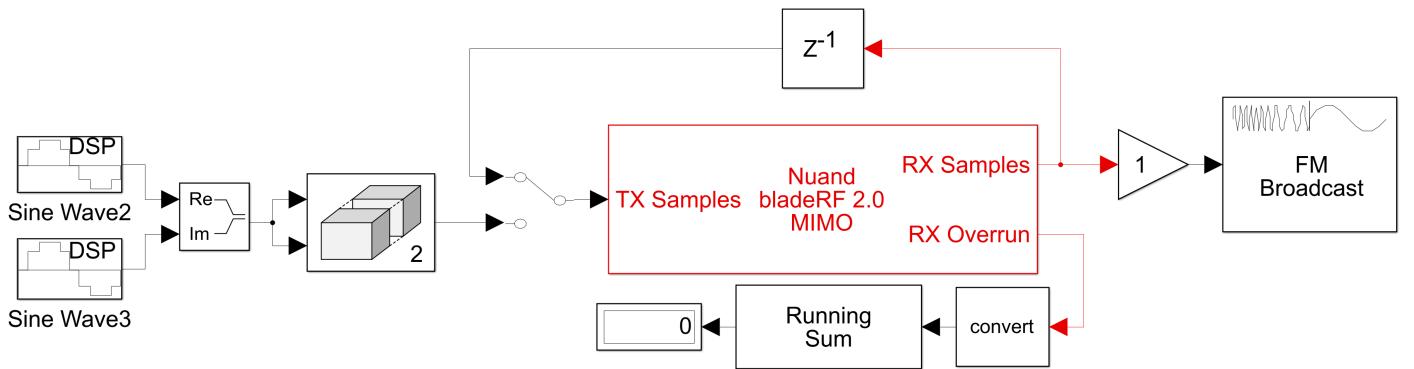


Figure 4.1: Simulink model for testing the MIMO interface of the BladeRF.

The first thing to consider is that the `bladeRF_MIMO` block can transmit and receive coherent MIMO signals without dropping any sample. This feature was evaluated by adding the running sum block to the RX Overrun output, as this block evaluates whether some samples were dropped in the current time step as described in section 3.4.1. If the number in the display keeps increasing during run time, the system is not considered deterministic as the time step of the model is not constant.

Figure 4.2 shows the frequency spectrum of both signals. The signal received from the radio station is considered the baseline signal, and its meant to be compared to the signal received from the SDR; ideally, both spectrums should be identical. Often, artefacts in the time domain such as discontinuities are sensed as big spikes that sound similar to a short burst of white noise. These bursts are shown in the frequency domain as spikes in the mid-high end of the spectrum. However, the spectrum from the re-transmitted signal closely matches the reference as the main difference is that the re-transmission increases the noise-floor and presents a slightly more powerful response in the low-powered sections of the spectrum. Finally, no figure is presented for every channel layout as the results are very similar across channels.

The second test scenario for the MIMO interface for the BladeRF is to receive a self-generated signal described in equation 4.1. The goal of this test if to visually inspect the received IQ signal from both receiver channels of the SDR. The expected result is two IQ signals with similar shape (given by equation 4.1) but phase-shifted, as the signal sensed by an antenna array delayed between antenna elements and

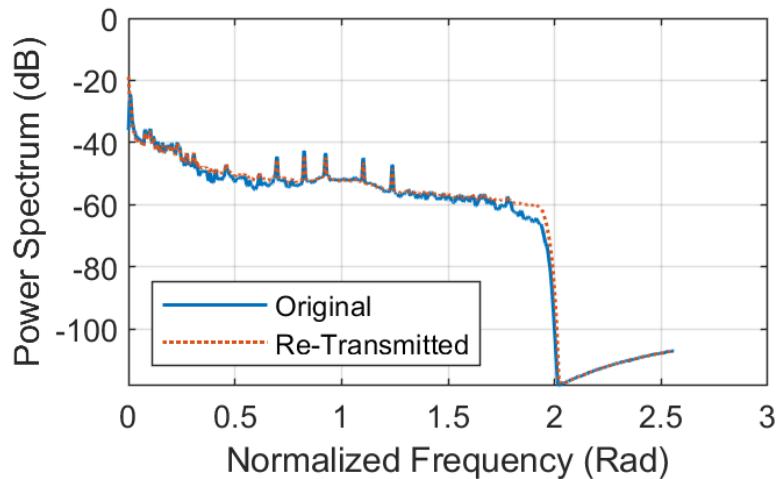


Figure 4.2: Frequency spectrum of the signals coming from the radio station and the SDR.

this delay is represented by a phase shift. The actual phase shift is not essential for this test, as this is evaluated in section 4.4.3.

Figure 4.3 shows a constellation diagram containing the IQ signals from both RX channels in the SDR. The fading colour in both trails represents the signal's path in time. Although the signal's shape resembles equation 4.1, some artefacts can be found in some segments of the trail which are caused by the SDR's AGC trying to compensate for the IQ imbalance in the IQ signal. Finally, signals from both channels RX0 and RX1 are close to orthogonal, confirming the expected output of two similar but phase-shifted signals.

$$y(t) = I(t) + jQ(t) = \sin(2\pi(100)t) + j0.6 \sin(2\pi(100)t + \frac{\pi}{2}) \quad (4.1)$$

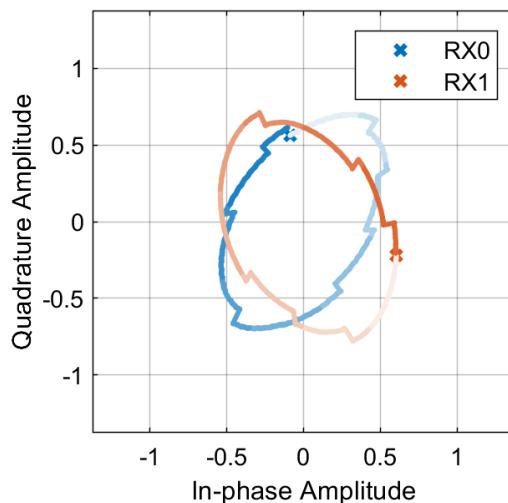


Figure 4.3: IQ signal from both RX channels in the SDR.

4.3 Modulator and Receiver

This section evaluates both modulator and receiver blocks as if they were part of a non-beamformed communications system. The overall simulation parameters are as the ones given in table 4.1 and the goal is to simulate a reliable end-to-end communication system using an SDR. The simulation will include some channel impairments such as Gaussian noise and a complex phase shift; the latter is essential as the TX and RX LOs in the SDR will cause a phase shift in the received symbols. Other parameters and details are not reviewed in this simulation as these blocks are not the main topic of this report.

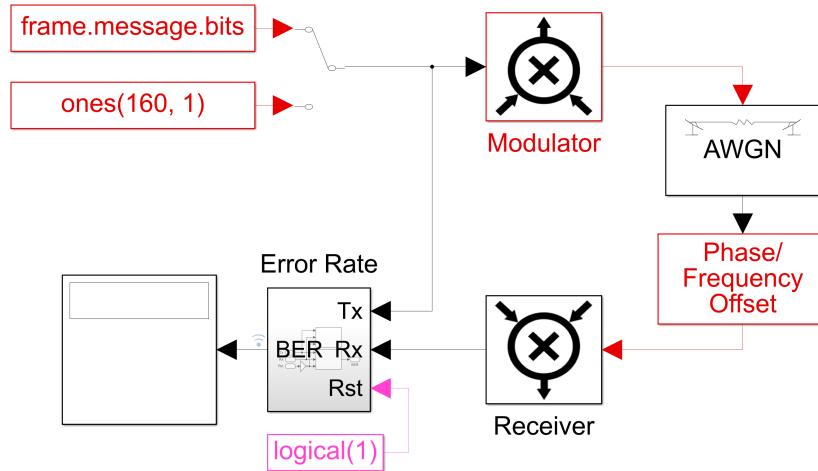


Figure 4.4: Simulink model for testing the Modulator and Receiver blocks.

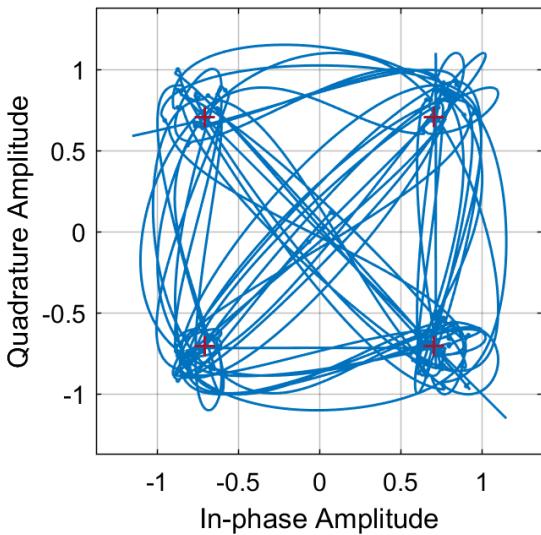
4.3.1 Modulator

Since the modulator block is considerably less complicated than the receiver block, its analysis is reduced to a constellation diagram showing the output of a constant sequence of ones as bit input, as its best metric is an end-to-end simulation which is reviewed in the next section. This sequence is used as input because the modulator must be able to spread the bits into several different symbols before interpolation. The expected output is a complex IQ signal with a few deviations from the reference constellation, as the roll-off coefficient of the Raised Cosine filter is less than 1.

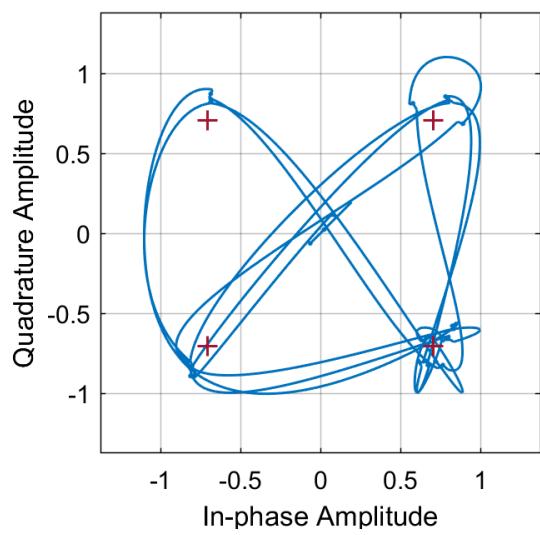
Figure 4.5b shows the output of the modulator block when given a constant one sequence. The signal is 10,000 samples long, and it shows curved paths, a consequence of the roll-off coefficient. The signal features every possible symbol from the reference constellation disregarding the constant nature of the input sequence.

Parameter	Value
Scrambler: Polynomial	$1 + z^{-1} + z^{-2} + z^{-4}$
Scrambler: Initial Condition	0 0 0 0
Raised Cosine: Roll-off	0.3
Raised Cosine: Filter Span	6
Raised Cosine: Interpolation Factor (N_{int})	100
Raised Cosine: Linear Amplitude Gain	$\sqrt{100}$

Table 4.2: Testing parameters for the modulator block.



(a) IQ signal for the default message bits.



(b) IQ signal for the constant bit sequence.

Figure 4.5: Constellation diagrams showing different modulated signals.

4.3.2 Receiver

The receiver is one of the most complex algorithms that were developed for this work. As described in section 3.2.2, it samples, synchronises, corrects and demodulates the input signal in order to get the same frame bits as the ones transmitted. The used validation metrics for this block are the elapsed time for the carrier synchroniser to reach the steady-state and the BER. The tests were done assuming a critically damped synchronizer $\zeta_{cs} = 1$.

Parameter	Value
Raised Cosine: roll-off (β)	0.3
Raised Cosine: filter span	6
Raised Cosine: Decimation Factor (N_{dec})	10
AGC: Time step	0.01 s
AGC: Output power	1 W
AGC: Average Length	100
Carrier Synchronisation: Samples per symbol	10
Carrier Synchronisation: Damping factor (ζ_{cs})	1
Symbol Synchronisation: Samples per symbol	10
Symbol Synchronisation: Damping factor (ζ_{ss})	1
Preamble Detector: Threshold	14

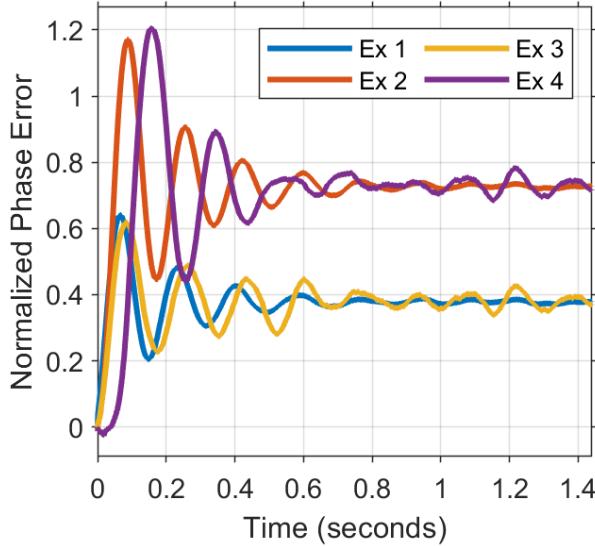
Table 4.3: Testing parameters for the receiver block.

Four simulation runs were logged using the parameters in 4.3 and different phase and noise values in order to test the overall performance of the receiver; the simulation time is 30 seconds. These parameters and the steady-state BER are shown in table 4.4. Also, an ideal baseline run is included.

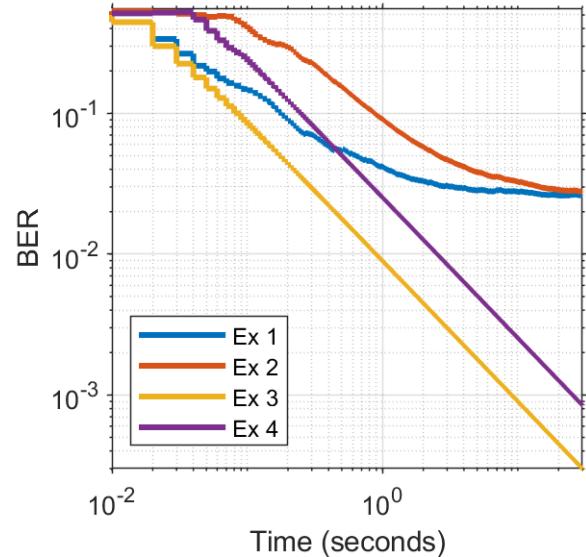
The simulations show that a more massive phase shift in the transmitted symbols results in a more extensive steady-state transient time for the carrier synchronisation, as shown in figure 4.6a. The transient time constraints the system's performance as the phase corrector is heavily impacted by the phase offset after the carrier synchronisation, resulting in erratic samples at the first few simulation

Experiment Number	Eb/No (dB)	ϕ_{err}	BER
Baseline	Noiseless	0	0.000166
1	5	20°	0.026124
2	5	130°	0.027701
3	20	20°	0.000299
4	20	130°	0.000845

Table 4.4: Results of the receiver test scenario.



(a) Normalised phase error from the carrier synchronisation for the test simulations.



(b) BER across the simulation time for all simulations.

Figure 4.6: Results for experiments in table 4.4.

steps. These outlier samples are always introduced by the synchronisation, affecting the steady-state BER.

Although the carrier synchronisation needs some time steps to stabilise, for experiments three and four, the initial steady-state error was the only source of error on the communication system. As shown in figure 4.6b, the BER in experiments 1 and 2 is mostly affected by the low EbNo as it converges from the initial stabilisation-error to the expected BER shown in figure 4.7. However, the only errors in experiments 3 and 4 are the stabilisation errors, and no other artefacts were found in the simulation time.

The behaviour of the receiver in across Eb/No is expected, as the symbol synchronisation stage does affect the BER performance depending on the timing error, which is expected to be the performance bottleneck [17]. This simulation was also compared to the theoretical performance of QPSK in AWGN, and the result is shown in figure 4.7. The BER curve of the system with the receiver block reaches slightly worse error rates at the same Eb/No, compared to the system without the receiver block.

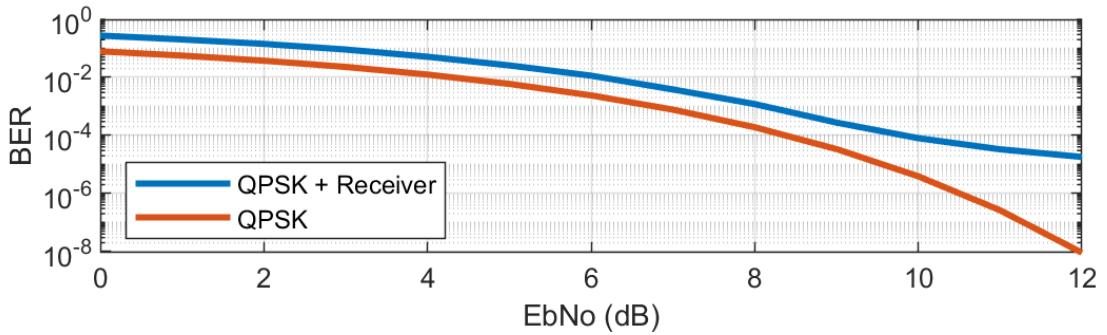


Figure 4.7: BER across different Eb/No from the receiver subsystem simulation test. $\phi_{err} = 20^\circ$.

4.4 Beamformers

Both phase-shift beamformers described in section 2.1.2 and coded in section 3.2.4 are subjected to different test to show whether they perform as expected. These tests are validated using the beamforming equations in section 2.1.2.

4.4.1 TX Beamformer

The testing in this section is meant to validate whether the TX beamformer emits a radiation pattern similar to the theoretical given in equation 2.8. The test scenario consists on the TX antenna array being placed on an open area while a second SDR measures the received power from a signal transmitted by the beamformed array at different observation points (O_m) around the array. The distance d_O from the array to O_m must be constant, to maintain path loss consistent between experiments.

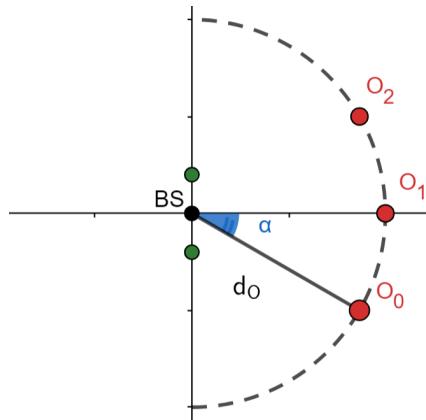


Figure 4.8: Location layout example for the TX beamformer test. The green dots are the antenna elements in the TX array, while O_0 , O_1 and O_2 are different measurement locations around the array.

The Simulink model used for this test is the HIL demo in figure 3.16, described in section 3.6. Since only the TX beamformer was being evaluated, the components in the receiving end were excluded from the model. The logged parameters of this experiments are the angle (α) between O_m and the x axis of the antenna array, and the RMS power of the signals received at O_m . The validation metric for this experiment is a comparison of the theoretical radiation pattern and the logged RMS value. Both theoretical and received samples must be normalised in order to be comparable, as the theoretical plot represents the array gain, and the real samples do include the path loss attenuation. The normalisation is done using MATLAB's `normalise` function and the z-score method that sets the output sequence's mean to zero and the standard deviation to one.

The coherency of the TX channels in the SDR can be evaluated using the previous methodology, but the beamforming algorithm is tested by steering the main beam and get the peak RMS value at a location different than 0° . The steering angle (ϕ) present in equation 2.8 must be different than 0 degrees.

The actual test was carried out in a small garden, using a rope to establish a constant distance of $d_O = 2.2$ meters and a steering angle $\phi = 30^\circ$. The small size of the experiment area constrained access to the whole semi-circle around the array, limiting the number of data points. Finally, the SDR used to measure the RMS value must be configured to have a constant RX gain, instead of using the AGC.

α	RMS($IQ(t)$)	Normalised	Theoretical
-50°	0.0017	-0.3702	-0.9080
-22°	0.0007	-1.5273	-1.681
0°	0.0022	-0.2083	0.1994
22°	0.0029	1.0182	1.2060
50°	0.0026	0.6711	0.9603

Table 4.5: Results for the measurements in O_m at $d_O = 2.2$ meters and $\phi = 30^\circ$.

Figure 4.9 contains the normalised theoretical radiation pattern, the normalised measurements from the real implementation and a second order interpolation of these measurements for improved visualisation. This figure shows that the interpolated plot greatly resembles the theoretical pattern. Overall, the measured pattern matches the theoretical better at the high-power section of the pattern (around 30°) than it does on the low-power section (around -30°). This mismatch may be caused by the lack of observation points, the channel's noise or the phase-shift beamformer nature, as non-true time-delay beamformers tend to have un-sharp responses.

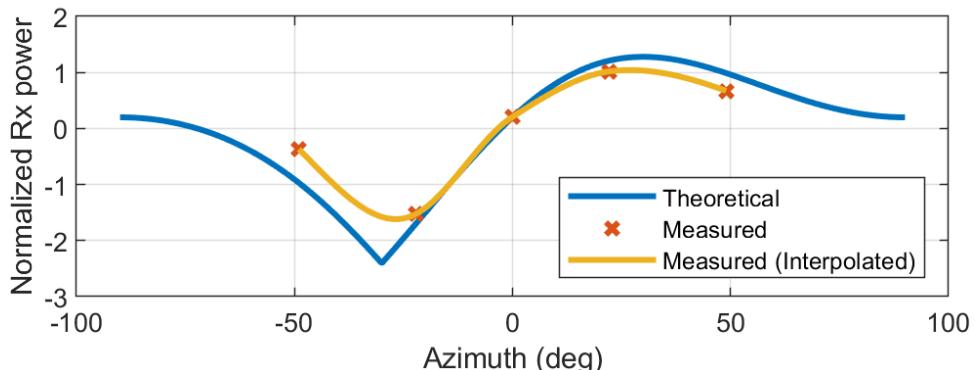


Figure 4.9: Normalised radiation pattern for the theoretical and real beamformers. The measurements of the real implementation were interpolated to enhance its visualisation.

4.4.2 RX Beamformer

The RX beamformer can also be considered interference rejecting as it can steer its main beam towards the desired signal's DOA and at the same time reject signal from sources coming from directions different to the desired. As stated in section 2.1.2, a beamformer can be modelled as a low-pass FIR filter, and the test environment for this block is meant to show the similarities of beamformers and digital filters.

Noise Rejection and Directivity

The scenario consists of three elements, the RX antenna array, a wanted signal source (S_{wan}) and an interference signal source (S_{int}). The RX antenna array is placed on an open area, and both signal sources are placed at different locations making the DOA estimated by the antenna array significantly different. Ideally, this condition can be met by placing both devices at opposite ends of the array's coordinate system, but impairments as multipath results in electromagnetic waves refracting and bouncing into walls and other surfaces; hence, reaching the antenna array in unexpected directions. The goal for this test is to prove that the beamformer can attenuate a signal coming from un-wanted direction (θ_{int}) while keeping the wanted signal by setting the beamformer's steering angle (ϕ) to the wanted signal's DOA (θ_{wan}); also the steering angle is set to a direction between θ_{wan} and θ_{int} to measure the beam's performance and sharpness, this direction is called θ_{null} .

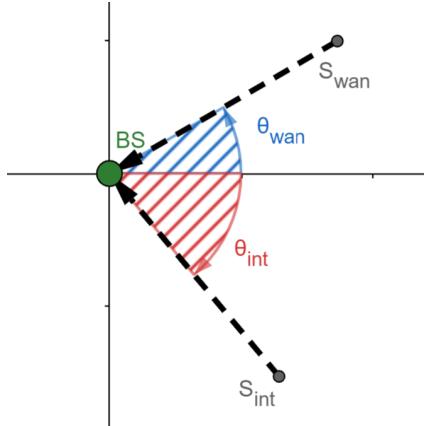


Figure 4.10: DOA layout example for the RX beamformer test. The wanted S_{wan} and interference S_{int} signal sources are arriving to the BS at different angles relative to the BS.

The metrics used for validation are the BER and EVM. The EVM outputs the resemblance of a signal to a reference constellation, considering phase, amplitude and noise variance. The EVM is based on the squared error between the measured and reference symbols. Hence, the lower the EVM, the greater the resemblance and performance [24]. The BER is used as an end-to-end performance metric and is expected to be lower when the beam is steered towards the wanted signal source.

Two experiment sets are due, with and without the interference signal. The baseline run, without the interference, is meant to show the performance of the beamformer in terms of directivity. The actual test does include both signal sources.

Both a simulation and real implementations were done for this scenario. Both consider the $\theta_{wan} = 25^\circ$, $\theta_{int} = -40^\circ$ and θ_{null} , include the modulator and receiver block and the parameters given in tables 4.2 and 4.3. The simulation also features channel impairments such as 10 dB SNR from an AWGN channel and 20° phase offset.

The simulation model in figure 4.11 consists of two signal sources, the wanted source comes from the modulator block, and the interference signal comes from a white noise source. The signal sensed by the RX antenna array is modelled as the superposition of both signals, described by equation 4.2. The EVM calculation is done after the carrier synchronisation to discard the phase-shift introduced by the channel and beamformer, being the SNR the only impairment affecting the measurement.

The expected behaviour for this scenario is that BER and EVM improve as steering angle approaches the wanted signal's DOA. This trend is expected across the baseline and test runs, although the baseline runs may not present such an improvement, as the RX antenna array only contains two elements. As mentioned in 2.1.2, the number of elements in the array translates to the filter order of a low-pass FIR filter.

$$y(t) = \sum_{m=0}^{M-1} (x_{wan}(t) + x_{int}(t) + n(t)) w_m^* \quad (4.2)$$

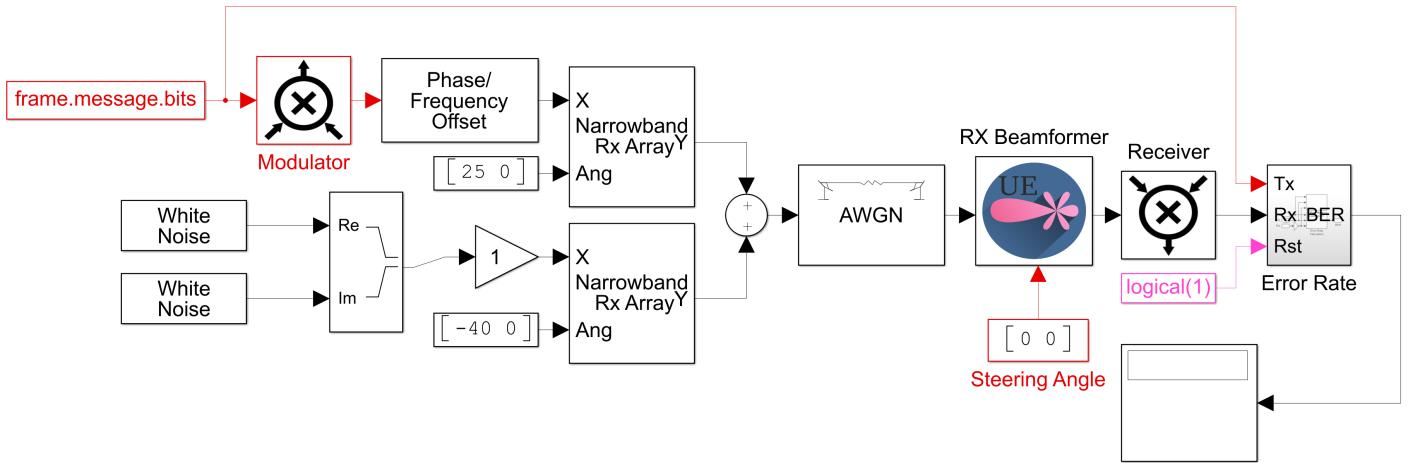


Figure 4.11: Simulink model for representing the RX beamformer test scenario.

The results for both simulation and real tests are shown in tables 4.8, 4.9, 4.6 and 4.7. The baseline results in table 4.6 show that, for a situation with a single signal source, the two-element beamformer is not completely filtering signals coming from $\phi \pm 25^\circ$; as the radiation pattern is wide enough for the AGC and carrier synchronizer to successfully recover the signals coming from $|\phi - \theta| < 25^\circ$, resulting in similar BER and EVM across the experiments with the steering angle set to θ_{wan} and θ_{null} . The real scenario baseline run in table 4.7 shows less-ideal results but with the same trend as the simulations.

ϕ	BER	EVM (%)		
		RMS	Max	95th
θ_{int}	0.002891	47.62	87.19	79.06
θ_{null}	0.000191	42.01	84.19	75.87
θ_{wan}	0.000199	41.96	84.19	75.7

Table 4.6: Baseline results from the simulated scenario.

ϕ	BER	EVM (%)		
		RMS	Max	95th
θ_{int}	0.292900	51.68	86.85	88.69
θ_{null}	0.002909	38.42	71.91	68.47
θ_{wan}	0.001330	36.73	81.39	68.03

Table 4.7: Baseline results from the real scenario.

The test result scenario shows consistent results, as the BER performance is improved when ϕ is closer to θ_{wan} . The test simulation shows much worse performance than the baseline at $\phi = \theta_{int}$; as S_{int} is being amplified while S_{wan} is attenuated. The same happens at $\phi = \theta_{wan}$, but it is worth noting that the amplification factor at the peak of the radiation pattern is not as large as the attenuation factor at the lower end, as shown in figure 4.12. Again, the simulation and real results show the same trend, validating the results.

Magnitude and Phase

The amplitude difference at different steering angles is given by the beamformer's radiation pattern, which is similar to a digital filter. The magnitude response of the beamformer is described in figure 4.9, and the amplitude difference of S_{wan} after the beamforming block at $\phi = \theta_{wan}$ and $\phi = \theta_{int}$ is shown in figure 4.13. The theoretical gain at $\phi = \theta_{wan}$ and $\phi = \theta_{int}$ is 1.41 and 0.147, respectively; hence, the gain

ϕ	BER	EVM (%)		
		RMS	Max	95th
θ_{int}	0.367300	60.60	101.3	92.34
θ_{null}	0.000166	42.22	84.24	76.25
θ_{wan}	0.000174	41.97	84.19	75.6

Table 4.8: Test results from the simulated scenario.

ϕ	BER	EVM (%)		
		RMS	Max	95th
θ_{int}	0.07144	48.47	87.47	80.23
θ_{null}	0.03146	44.13	85.66	74.94
θ_{wan}	0.02278	42.55	81.12	73.13

Table 4.9: Test results from the real scenario.

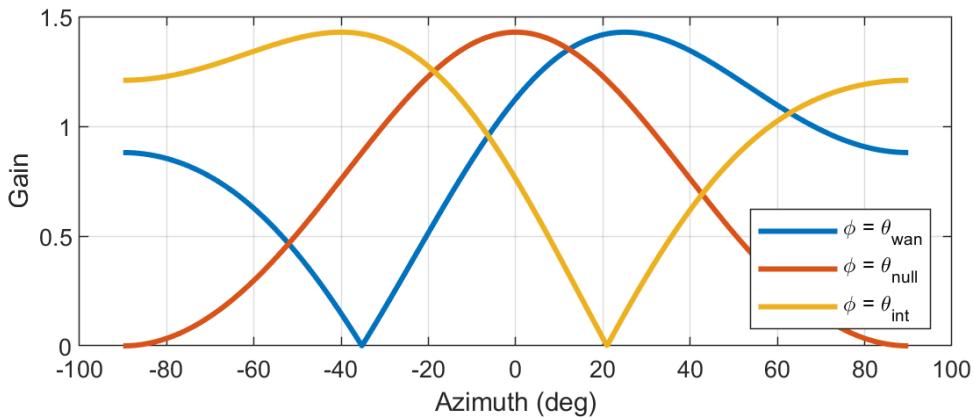


Figure 4.12: Theoretical radiation pattern for the beamformer at different steering angles.

ratio can be estimated and should be similar to the gain ratio of the real signals as shown in equation 4.3.

$$\begin{aligned} \frac{|S_{wan}|_{\phi=\theta_{wan}}}{|S_{wan}|_{\phi=\theta_{int}}} &\approx \frac{1.41}{0.147} = 9.59 \\ \frac{|S_{wan}|_{\phi=\theta_{wan}}}{|S_{wan}|_{\phi=\theta_{int}}} &= RMS\left(\frac{|S_{wan}(t)|_{\phi=\theta_{wan}}}{|S_{wan}(t)|_{\phi=\theta_{int}}}\right) = 10.12 \end{aligned} \quad (4.3)$$

Similarly, the phase-shift of the beamformed signal can be validated by using the phase angle of the radiation pattern, instead of the magnitude as is often assumed. the "phase pattern" of the beamformer is shown in figure 4.14, and a validation for the beamformed signal's phase difference can be estimated by comparing the theoretical phase-difference using the phase pattern in figure 4.14 and the actual phase difference of the real signals in figure 4.13 as shown in equation 4.4 where mod represents the modulo operator. This equation outputs a phase angle comparable with the phase pattern in figure 4.14.

$$\begin{aligned} \angle S_{wan}(t)_{\phi=\theta_{int}} - \angle S_{wan}(t)_{\phi=\theta_{wan}} &\approx 43^\circ - 0^\circ \\ \angle S_{wan}(t)_{\phi=\theta_{int}} - \angle S_{wan}(t)_{\phi=\theta_{wan}} &= \text{Mean}(\angle S_{wan}(t)_{\phi=\theta_{int}} - \angle S_{wan}(t)_{\phi=\theta_{wan}}) - 45^\circ \bmod 45^\circ \quad (4.4) \\ \angle S_{wan}(t)_{\phi=\theta_{int}} - \angle S_{wan}(t)_{\phi=\theta_{wan}} &= \text{Mod}(133.5^\circ - 45^\circ, 45^\circ) = 43.5^\circ \end{aligned}$$

The comparisons made with equations 4.3 and 4.4 show that the theoretical gain and phase of the beamformer is close to what s_{wan} presents. This validation confirms that the beamformer behaves as the theory implies.

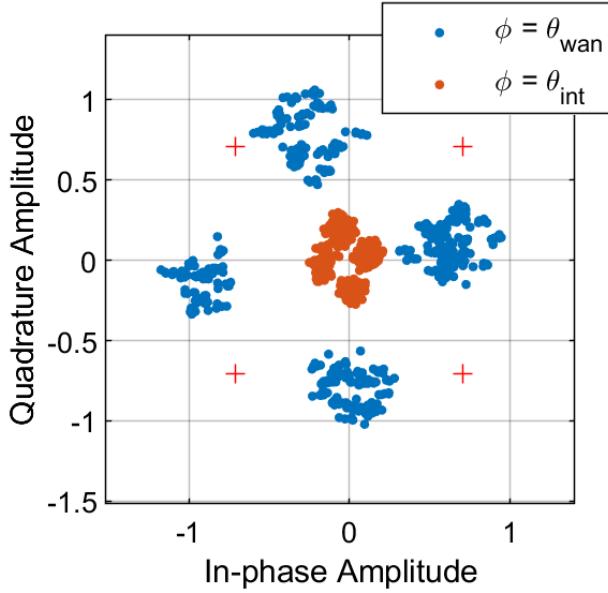


Figure 4.13: Constellation diagram of S_{wan} after the beamforming block for the real implementation using a steering angle of $\phi = \theta_{wan}$ and $\phi = \theta_{int}$.

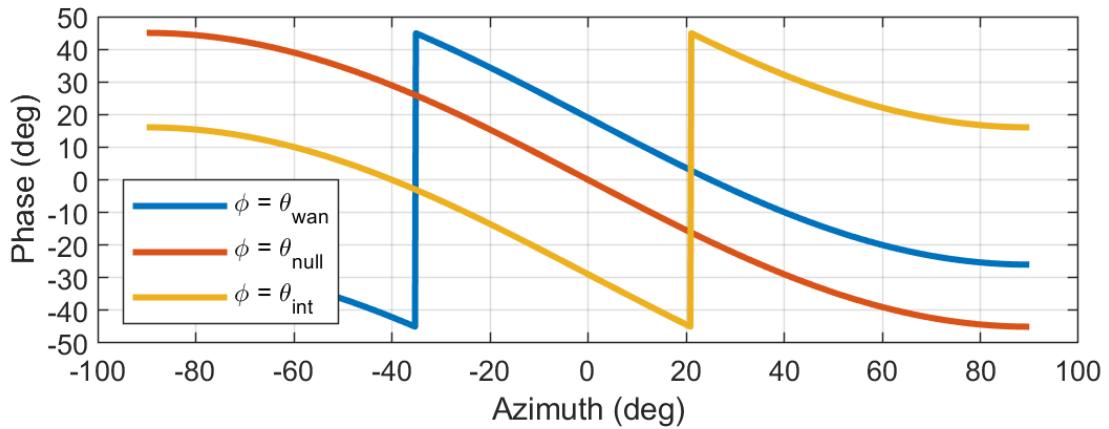


Figure 4.14: Theoretical phase pattern for the beamformer at different steering angles.

4.4.3 Direction of Arrival

The validation test for the Direction of arrival calculation is done by using the beamforming equation in section 2.1.2. As previously stated, the incidence angle of a signal arriving at the RX antenna array causes a constant phase shift between the antenna elements. The complex weight w_m contains the steering angle as its phase component; this component is described by equation 2.7. The phase offset in the baseband signal (α_{bb}) is given by the phase component of the complex weight at element $m = 1$; a reduced expression for the baseband phase angle is given by equation 4.5, the output phase is given in both radians and degrees and assumes an optimal beamformer. In addition, the phase-shift of the received signals ($\hat{\alpha}_{bb}$) can be estimated by computing the mean phase difference between the received signals by each antenna element (S_m) as shown in equation 4.6.

$$\alpha_{bb} = \angle w_1^* = -\pi \sin(\theta_{DOA}) = -180 \sin(\theta_{DOA}) \quad (4.5)$$

$$\hat{\alpha}_{bb} = \text{Mean}(\angle S_0 - \angle S_1) \quad (4.6)$$

The DOA estimation is validated using S_{wan} from the previous section when sensed by both antenna elements in the SDR as S_m for equation 4.5 and its computed DOA $\theta_{DOA} = \theta_{wan} = 25^\circ$ for equation 4.5. The results of the previous computation output $\alpha_{bb} = -76.07^\circ$ and $\hat{\alpha}_{bb} = -78.66^\circ$; the similarity of both theoretical and estimated phase-differences validate the accuracy of the DOA estimator.

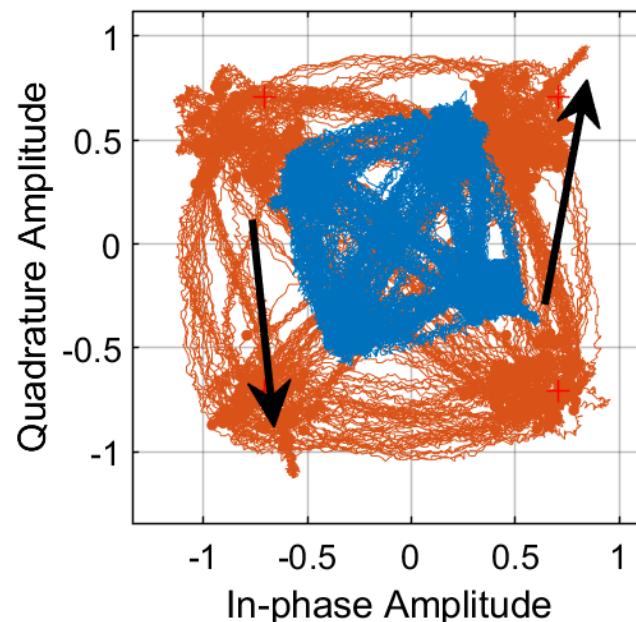


Figure 4.15: IQ signals for S_{wan} after being sensed by each antenna element in the array from the real implementation.

4.5 User Equipment Tracking

This last scenario consists of the BS tracking the UE as it moves in a large open area, as described in section 3.2. The goal of this scenario is to test all the developed parts in a real-life situation. The metrics for this test are the BER and estimated DOA by the BS. These will be validated if the BS can estimate the UE location while maintaining a relatively low BER performance.

The simulation and real runs differ. The simulation assumes a BS with two antenna arrays and a UE supporting beamforming at the receiving end, as stated in section 3.2. However, the real implementation assumes both BS and UE have a single antenna array; transmitting and receiving, respectively. This decision was made because there is no way to run a Simulink model with two SDRs transmitting and receiving simultaneously using the available hardware; as many of the received samples are dropped due to timing issues. Although these assumptions may result in different results, the BER for both runs should have a consistent trend.

4.5.1 Simulation

The simulation uses the model presented in figure 3.1, which contains all the blocks described in section 3.2. The input parameters are the ones used in previous experiments such as tables 4.1, 4.2 and 4.3, and the remaining parameters are shown in table 4.10. The test consists of four runs, each one with a different combination of components (TX/RX beamformers and receiver); these are compared to the theoretical BER for QPSK. The DOA is only shown for the run that includes all the components.

Parameter	Value
AWGN: Input Reference Power	1 W
UE: Movement period ($1/f_{UE}$)	10 s
UE: Maximum displacement angle (DOD_{max})	45°
UE: Azimuth (Az_{UE})	15°

Table 4.10: Input parameters for the UE tracking simulation test.

Experiment Number	Modulator/Receiver	TX Beamformer	RX Beamformer
1	✓		
2	✓	✓	
3	✓		✓
4	✓	✓	✓

Table 4.11: Component combination for each experiment in the UE tracking simulation test.

The result in figure 4.16 show that experiment 1 have a worse BER than the theoretical, as stated in section 4.3.2 due to the timing error introduced by the symbol synchronisation. Experiments two and three have a similar impact in BER, as they both introduce the same gain improvement at their respective end, being this a factor of $\sqrt{2}$, as shown in figure 4.9 which also applies for RX beamformers. Introducing either a TX or RX beamformer to the system significantly improves the BER even while limited by the receiver. Finally, experiment four shows the best BER performance of them all, as the effect of both beamformers, is combined.

The BER improvement introduced by both beamformers is also limited by the number of antennas in the array, as adding more antennas improves directivity, which increases the SNR at both ends. A few BER equivalents of experiment four are presented in figure 4.17 using $M = 2, 4, 8$.

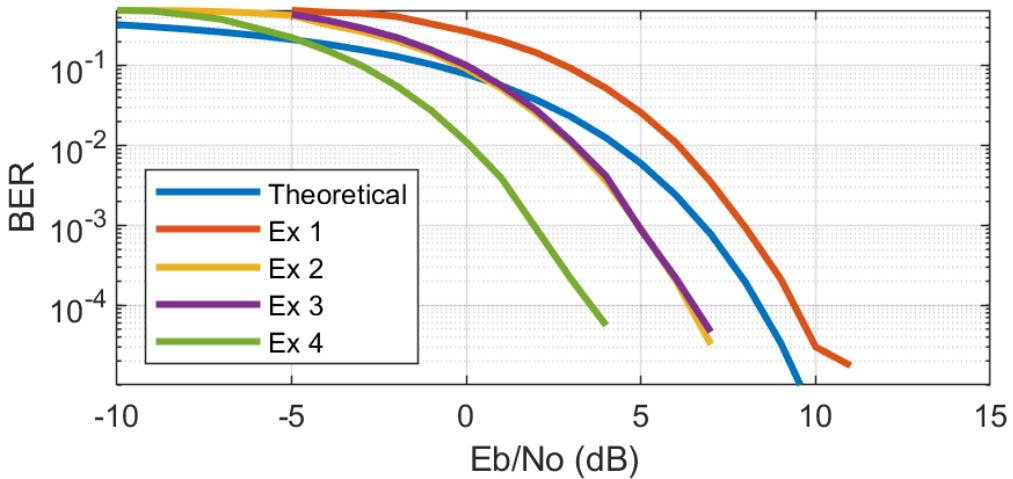


Figure 4.16: BER performance across experiments for the UE tracking simulation test

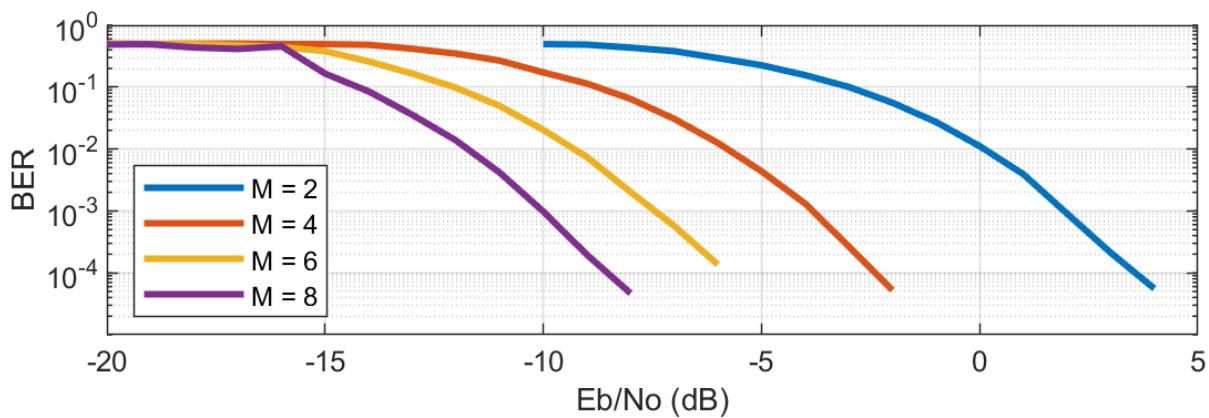


Figure 4.17: BER performance for systems with different antenna array elements for the UE tracking simulation test.

The second metric, the estimated DOA by the BS, is greatly affected by the channel's noise; hence, the steering angle may not be set to the actual location of the UE. The beam of a two-element array is not very narrow, so the estimation error does not affect the BER as much as in an array with many antenna elements. Figure 4.18 shows the measured DOD across simulation time, which proves that lower SNRs add noise to the estimation. Finally, even with low SNR conditions and poor angle estimation, the BER performance shows that a communication link can be established.

4.5.2 Real Implementation

As previously mentioned, the real implementation of the UE tracking scenario uses two antenna arrays, one representing the transmitting end of the BS while the other represents the receiving end of the UE. Since the two antenna arrays are joined together to the SDR with one metre SMA cables, the moving array cannot move freely around the other, making the test scenario more challenging. In the end, the test scenario consisted of the BS sweeping the DOD by using the same sinusoidal movement pattern as the one described in section 3.3, maximum DOD (DOD_{max}) and movement frequency (f_{UE}). The receiving end of the implementation must estimate the DOA of the received signal and steer the RX beamformer's main beam towards the incoming signal.

The Simulink model used for this test is shown in figure 3.16, only replacing the TX steering angle

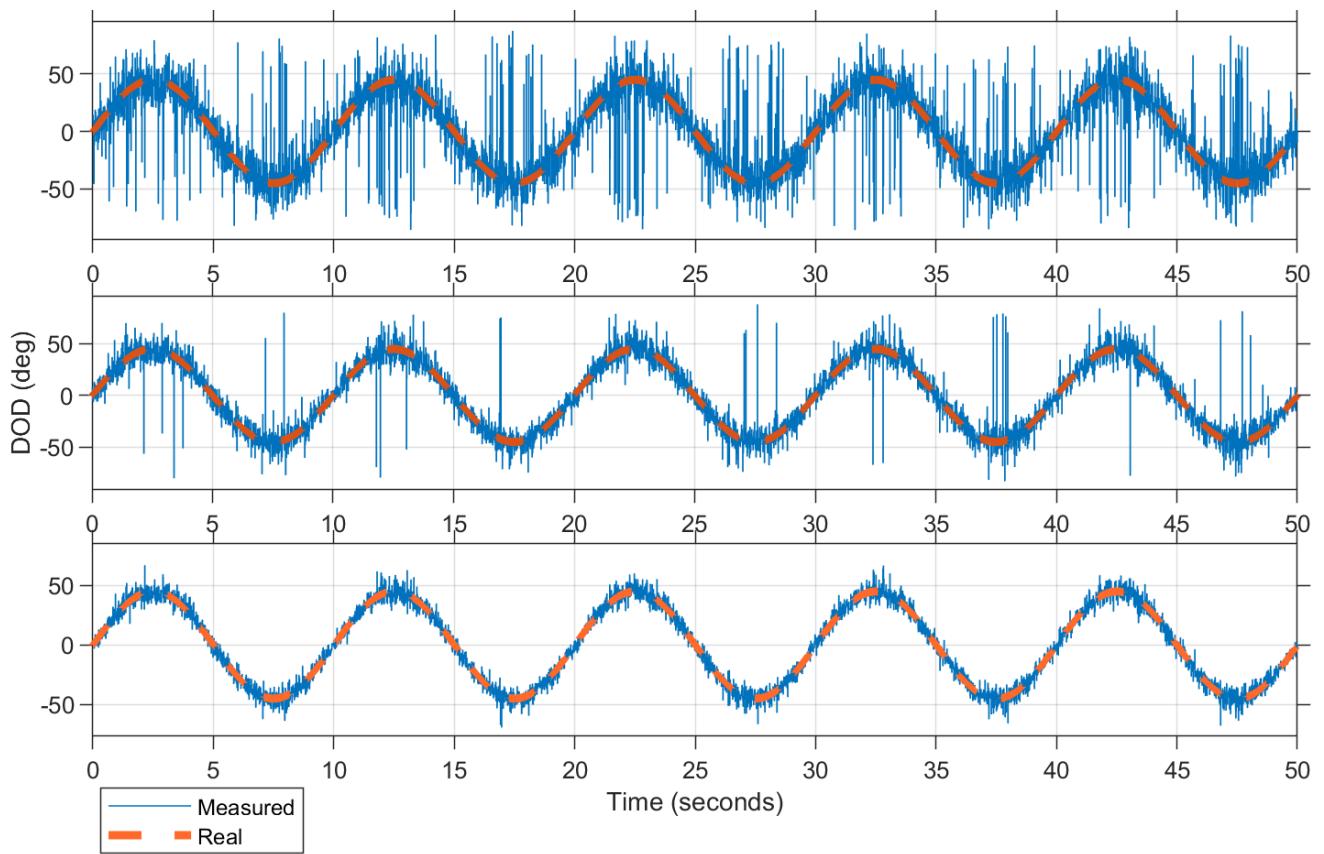


Figure 4.18: Estimated DOD using $Eb/No = -2 \text{ dB}$, 0 dB and 2 dB for the UE tracking simulation test.

by the DOD function in equation 3.1 represented in figure 3.3. The test took place in four different environments: side-by-side, facing each other, perpendicular and indoors. These scenarios vary in different ways, as the location and orientation of the antenna arrays and the resulting multipath propagation effect of the received signal affect the estimated DOA.

The system parameters are consistent for all four environments and, similar to the simulation section, uses the system parameters of the previous tests in addition to $DOD_{max} = 30^\circ$ and $1/f_{UE} = 10 \text{ seconds}$. The test runs are 30 seconds long to capture several periods of the steering pattern to validate its repeatability.

Each environment in figure 4.19 has different surfaces in which the transmitted signal can bounce and propagate. The side-by-side environment shown in figure 4.19a consists of the two antenna elements placed within a parallel axis near a wall; the x axis of both arrays point towards a glass door in which the signals can bounce quickly. The environment in figure 4.19b show both arrays facing each other at a close distance; this distance might not be enough for the signal to propagate properly by the beamformer. The perpendicular environment in figure 4.19c show both arrays placed, making both x axes perpendicular; some low-height walls surround the arrays. Finally, both arrays are placed indoor as shown in figure 4.19d; this scenario is expected to be significantly influenced by multipath propagation, as the walls and ceiling make the signal bounce greatly.

The results for these tests are shown in table 4.12 and figures 4.20 and 4.21. Table 4.12 proves that the communication link for every test environment was stable as the BER is relatively low. Figure 4.20 show the number of error of the communications system across time, and it can be noticed that the errors mostly come from the first second, as the carrier synchronisation is still reaching its steady-state. After reaching the steady-state, only few errors are introduced, and the error number remains fairly



(a) Side-by-side



(b) Facing each other



(c) Perpendicular



(d) Indoors

Figure 4.19: Real test environments for the UE tracking test

consistent across time. Most environments showed this behaviour, excepting the one in which both arrays are facing each other, as it has the most error before the steady-state and there is a considerable increase long after the fact. The close distance between arrays might cause this behaviour. However, the indoor environment showed the best BER performance, as the steady-state errors are considerably lower than the parallel and side-by-side environments. This performance may be the result of the multipath propagation of the room.

Environment	BER
Side-by-side	0.0025
Facing each other	0.0079
Perpendicular	0.0028
Indoor	0.0016

Table 4.12: Resulting BER for the UE tracking test

Figure 4.21 show the estimated DOA given a time-variant DOD for all four testing environments. Since the received signal by the RX antenna array is greatly affected by multipath and other impairments, the estimated DOA is not supposed to match the DOD; only a sinusoidal shaped signal is expected as this is the shape of the DOD. The objective of these tests is to prove that the estimated DOA by the

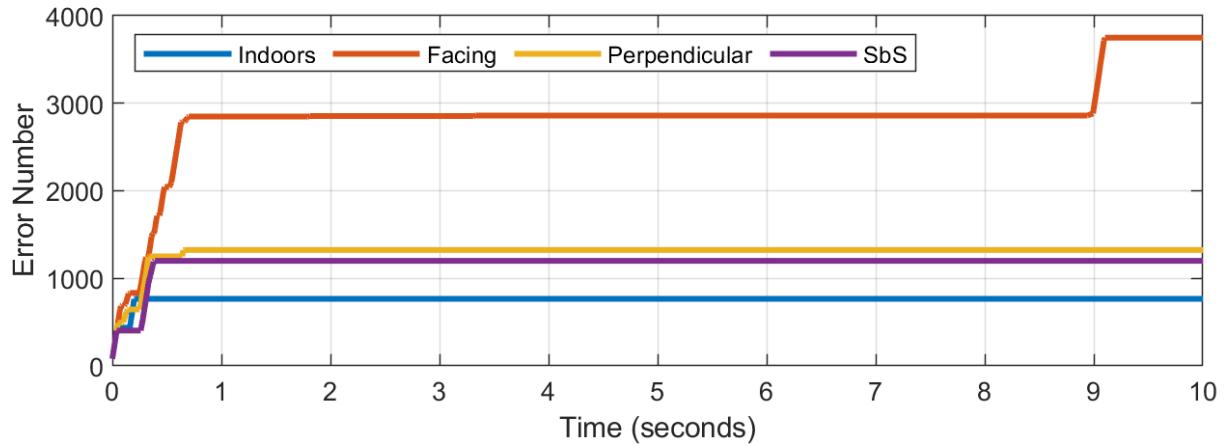


Figure 4.20: Errors across time for each environment in the UE tracking test.

receiver end is affected by the input DOD. The side-by-side environment profile in figure 4.21a shows a DOA profile close to being purely sinusoidal, which is obtained across all three periods in the test run time. The environment with both arrays facing each other in figure 4.21b shows some discontinuities, which may be caused by the ± 90 range constraint of the estimation algorithm. Considering the lower peaks are a product of this constraint, then the DOA profile would look like a rectified sinusoid, with little to no impact from the negative side of the DOD sinusoid. Figure 4.21c shows a sinusoidal pattern with reduced amplitude and a large offset, caused by the already high incidence angle between antenna arrays. Lastly, figure 4.21d shows a saturated sinusoid in the indoor environment which may be caused by the walls and the objects inside the room.

As expected, the estimated DOA seems to be affected by the input DOD, but in the case of figures 4.21b and 4.21d, these show some non-linear properties in the form of rectification and saturation, respectively.

Overall, the results were as expected, as the communication link was maintained while varying the DOD, and it was proven that the DOD affected the estimated DOA without moving the antenna arrays.

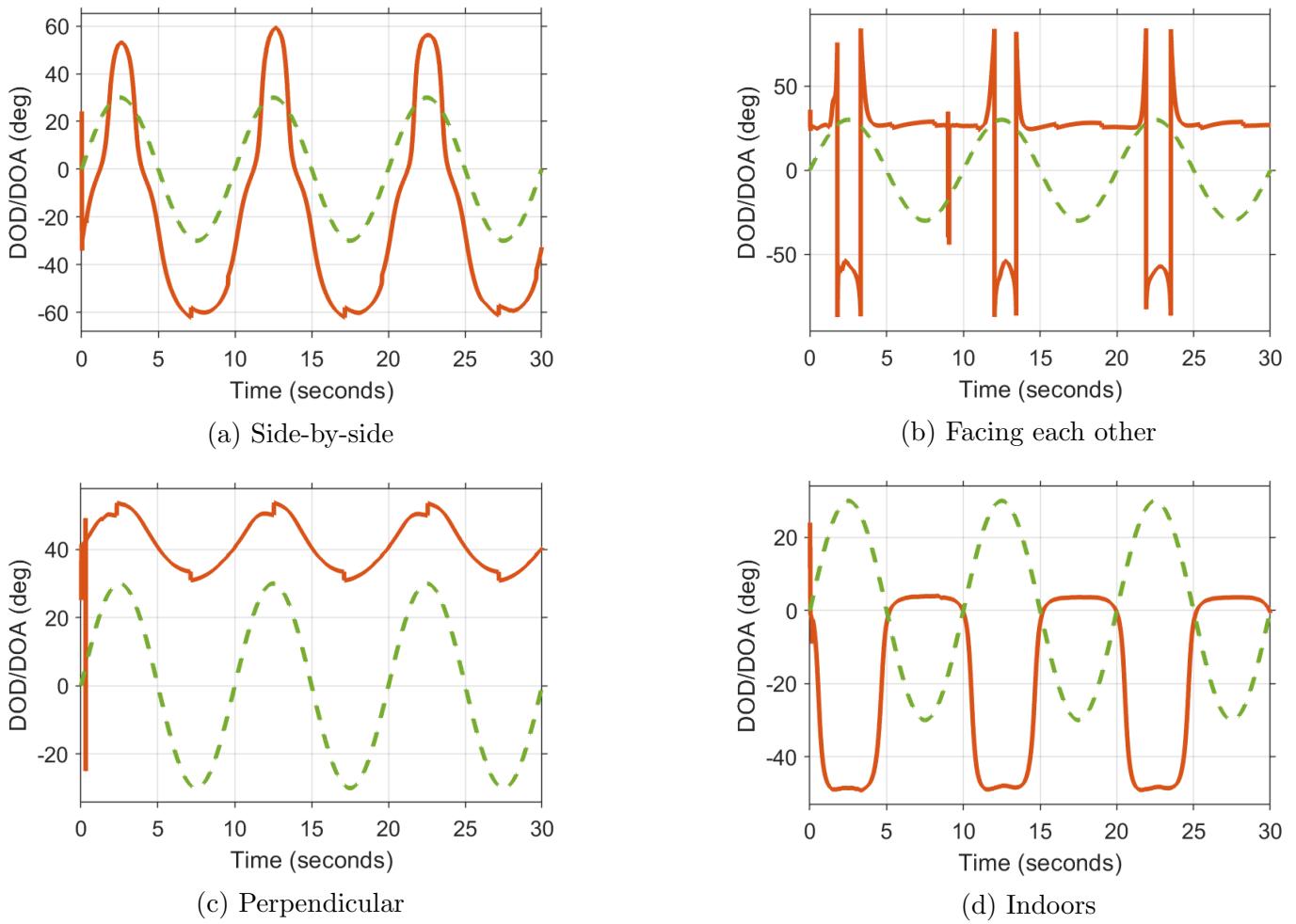


Figure 4.21: Estimated DOA (orange) from the received signals and the input DOD (green, dotted) for the real UE tracking test.

Chapter 5

Conclusions

The work done in the previous chapters show that both TX and RX digital beamformers can be implemented in low-cost SDR units just by knowing the hardware requirements and applying the algorithm in software. However, these requirements limit the hardware that can be used to a smaller sub-set, which is still more expensive than the commonly available RTL-SDR, but still a lot cheaper than the high-end brands as Ettus Research.

Also, a contribution to the SDR community was made, as the MATLAB interface for the BladeRF developed in section 3.4 is meant to be continued to fix some existing issues and then submitted into Nuand's repository for it to be available in the official repository if accepted by the manufacturer.

Chapter 4 proved that the work done in this project resulted in a beamforming system, but also required an enormous amount of effort and research to use SDRs and establish a working communications system. The proposed receiver scheme presented in section 3.2.2 could be the topic for another project, as there are many subsystems to enhance data throughput or error performance. The used receiver scheme, although it has its share of complexity, does not feature other techniques as MIMO or error detection, which can significantly improve its performance.

The final test scenario in section 4.5.2 was constrained due to limited hardware, but its results show that a proof of concept was achieved, which is the primary goal of this project. However, the similarity of the simulations to the real results (in terms of error rate) may indicate that the same trend found in the results of section 4.5.1 could be found if the real implementation was done with the same elements as the simulation.

The developed system's performance is satisfactory, as it roughly matched the theoretical responses and the simulation results. As of the objectives of section 1.3, all the primary objectives were satisfied, as the prototype uses the HIL design paradigm, the software is flexible and capable of handling different TX and RX beamformers based on the contents of chapter 2. The system is capable of tracking signals using the DOA estimator while maintaining an acceptable error performance. Also, every secondary goal was achieved, as the prototype is scalable by using multiple SDR units and the system simulation is accurate and contains the same blocks as the prototype. However, many improvements can still be introduced. Overall, the project is considered a success and a great topic for a follow-up project.

5.1 Future Work

Many things can be done to follow the work presented for this project. The nearest thing to do would be to increase the number of antenna elements by combining two or more BladeRF units; the software interface for the BladeRF would have to be updated to feature multi-IC synchronisation, and the Simulink system-object must consider the data coming from the multiple devices and align it. Other than that, the simulation and remaining subsystems already support a variable number of antennas, as shown in

figure 4.17. A different antenna array geometry can be implemented in order to support both azimuth and elevation steering.

Once many SDRs can be implemented simultaneously, the antenna array may be split to obtain multiple independent beams. This feature is useful if a Multiple Access scheme using beamforming is intended.

This same algorithm may be implemented using the GNU Radio Companion software, this to benefit from its open-source code and to combine it with different applications using its Python framework.

As a more complex flow-up, hybrid beamforming may be implemented in order to get sub-beams inside the analogue beam. This upgrade may require to modify or bypass the RF section of the SDR in order to attach its output to the analogue beamformer.

Appendix A

Repository

The Simulink model, MATLAB scripts and CAD files are available in the following git repository. The directory structure and files are described in the README section.

https://github.com/JoseAmador95/UoS_Beamforming.git

Bibliography

- [1] Nuand, “bladeRF 2.0 micro,” 11 2018. [Online]. Available: <https://www.nuand.com/bladerf-2-0-micro/>
- [2] ——, “libbladeRF 2.2.1,” 8 2019. [Online]. Available: <https://www.nuand.com/bladeRF-doc/libbladeRF/v2.2.1/>
- [3] B. Allen and M. Ghavami, *Adaptive array systems : fundamentals and applications*, 1st ed. Chichester: John Wiley& Sons, 2005.
- [4] W. Liu and S. Weiss, “Introduction,” in *Wideband Beamforming*, 1st ed. Chichester: John Wiley & Sons, Ltd, 3 2010, ch. 1, pp. 1–18.
- [5] R. W. E. e. Stewart, K. W. Barlee, D. S. W. Atkinson, and L. H. Crockett, *Software defined radio using MATLAB® & Simulink® and the RTL-SDR*, 1st ed. Glassgow: Strathclyde Academic Media, 2 2017.
- [6] Analog Devices, “AD7AD7476A/AD7477A/AD7478A Datasheet and Product Info,” Analog Devices, Norwood, Tech. Rep., 11 2011. [Online]. Available: <https://www.analog.com/en/products/ad9361.html><https://www.analog.com/en/products/ad620.html>
- [7] Lime Microsystems, “LimeSDR — Crowd Supply,” 2016. [Online]. Available: <https://www.crowdsupply.com/lime-micro/limesdr>
- [8] RTL-SDR.com, “RTL-SDR Blog V3 Datasheet,” Tech. Rep., 2020. [Online]. Available: <https://www rtl-sdr.com/wp-content/uploads/2018/02/RTL-SDR-Blog-V3-Datasheet.pdf>
- [9] D. Gaydos, P. Nayeri, and R. Haupt, “Experimental Demonstration of a Software-Defined-Radio Adaptive Beamformer,” in *2018 48th European Microwave Conference (EuMC)*, 2018, pp. 1581–1584.
- [10] P. Nayeri and R. L. Haupt, “A testbed for adaptive beamforming with software defined radio arrays,” in *2016 IEEE/ACES International Conference on Wireless Information Technology and Systems (ICWITS) and Applied Computational Electromagnetics (ACES)*, 2016, pp. 1–2.
- [11] P. Delos, “Receiver Design Considerations in Digital Beamforming Phased Arrays,” 2017. [Online]. Available: <https://www.mwrf.com/technologies/components/article/21845907/receiver-design-considerations-in-digital-beamforming-phased-arrays>
- [12] ——, “Digital Beamforming Techniques for Phased Array Systems,” 1 2017. [Online]. Available: <https://www.analog.com/en/education/education-library/webcasts/digital-beamforming-techniques-for-phased-array-systems.html>

-
- [13] C. H.-p. Intel, L. Microsoft, and N. E. C. Philips, “Universal Serial Bus Specification,” USB IF, Tech. Rep., 4 2000. [Online]. Available: <http://www.usb.org>
 - [14] USB Implementations Forum, “USB 3.1 Specification — Language Usage Guidelines from USB-IF,” USB IF, Tech. Rep., 7 2020. [Online]. Available: <http://www.usb.org/developers>
 - [15] IEEE Computer Society, *IEEE Standard for Ethernet 802.3-2018*, 6 2018, vol. 2018.
 - [16] S. Jang, R. Lu, J. Jeong, and M. P. Flynn, “A true time delay 16-element 4-beam digital beamformer,” in *Digest of Papers - IEEE Radio Frequency Integrated Circuits Symposium*, vol. 2018-June. Institute of Electrical and Electronics Engineers Inc., 8 2018, pp. 12–15.
 - [17] Mathworks, “Symbol and Frame Synchronization,” 2020. [Online]. Available: <https://uk.mathworks.com/help/comm/ug/symbol-and-frame-synchronization.html>
 - [18] O. A. Oumar, M. F. Siyau, and T. P. Sattar, “Comparison between MUSIC and ESPRIT direction of arrival estimation algorithms for wireless communication systems,” in *1st International Conference on Future Generation Communication Technologies, FGCT 2012*, 2012, pp. 99–103.
 - [19] J. Edgcombe, “Multiple device hardware level synchronization,” 2 2019. [Online]. Available: <https://github.com/mossmann/hackrf/wiki/Multiple-device-hardware-level-synchronization>
 - [20] M. Laakso, R. Rajamäki, R. Wichman, and V. Koivunen, “Phase-coherent multichannel SDR-Sparse array beamforming,” Tech. Rep., 2018.
 - [21] W. L. Myrick, “Distributed GPS beamforming with low-cost software defined radios,” in *Proceedings of the 2018 International Technical Meeting of The Institute of Navigation, ITM 2018*, vol. 2018-January. Institute of Navigation, 2018, pp. 495–501.
 - [22] Othernet, “KerberosSDR,” 2019. [Online]. Available: <https://othernet.is/products/kerberossdr-4x-coherent-rtl-sdr>
 - [23] Coherent Receiver, “Coherent Transceivers based on ADALM-PLUTO SDR,” 2020. [Online]. Available: <https://coherent-receiver.com/pluto-sdr>
 - [24] Mathworks, “Measure error vector magnitude,” 2020. [Online]. Available: <https://uk.mathworks.com/help/comm/ref/comm.evm-system-object.html>