

# Mechanical Antenna Beam Steering

Intruder Detection with Electromagnetic Waves

Rikke Udengaard

Bachelor of Engineering in Electronics, Spring 24

Fifth Semester Project







**Electronics and IT**  
Aalborg University  
[www.aau.dk](http://www.aau.dk)

# **AALBORG UNIVERSITY**

## STUDENT REPORT

**Title:**

Mechanical Antenna Beam Steering

**Theme:**

Digital and Analogue Systems Interacting with the environment

**Project Period:**

Spring Semester 2024

**Participant(s):**

Rikke Udengaard

**Supervisor(s):**

Rocio Rodriguez Cano  
Jan H. Mikkelsen

**Copies:** 1**Page Numbers:** 40**Date of Completion:**

May 16, 2024

**Abstract:**

In this semester project an algorithm is developed, that ...

*The content of this report is freely available, but publication (with reference) may only be pursued due to agreement with the author.*

# Contents

<b>Preface</b>	<b>vi</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Technical Analysis</b>	<b>2</b>
2.1 Fundamentals of Antennas . . . . .	2
2.1.1 Propagation In Free Space . . . . .	2
2.1.2 Polarization . . . . .	4
2.1.3 Radiation Characteristics . . . . .	4
2.1.4 Friis Transmission Equation . . . . .	6
2.1.5 S-Parameters . . . . .	6
2.1.6 Antenna Designs . . . . .	8
2.2 Beam Steering Methods . . . . .	11
2.2.1 Electrical Steering . . . . .	11
2.2.2 Manual and Mechanical Steering . . . . .	11
<b>3 Requirements and Delimitations</b>	<b>14</b>
3.1 Delimitations . . . . .	14
3.2 Functional Requirements . . . . .	14
3.3 Technical Requirements . . . . .	15
<b>4 Design of Antenna Beam Steering</b>	<b>16</b>
4.1 Antenna Design . . . . .	17
4.2 Choice of Turntable . . . . .	17
4.3 Choice of Vector Network Analyzer . . . . .	17
4.4 Design of Software Control Program . . . . .	17
4.4.1 Communication Interfaces . . . . .	18
4.4.2 Concurrency . . . . .	19
<b>5 Implementation</b>	<b>20</b>
5.1 Turntable Control . . . . .	20
5.2 VNA Control . . . . .	24

5.3	Concurrency in Main Control Flow . . . . .	25
<b>6</b>	<b>Tests</b>	<b>30</b>
6.1	Test of S-Parameters . . . . .	30
6.2	Test of Radiation Pattern . . . . .	31
6.3	Accept Test . . . . .	35
6.4	Summary of Tests . . . . .	36
	<b>Bibliography</b>	<b>37</b>
<b>A</b>	<b>Source Code Link</b>	<b>39</b>
<b>B</b>	<b>short</b>	<b>40</b>

# Preface

This project was developed by Rikke Udengaard in the Spring semester of 2024 at Department of Electronic System at Aalborg University. The theme of this 5th semester project is *Digital and Analog Systems in Cooperation with the World*.

The final product includes source code that is available on *Github*. The link to the repository can be found in appendix A with some of the source code is explained in chapter 5. The repository also contains the test result logs which are also available in condensed form in the appendix B.

The block diagrams used throughout this report have been generated with `app.diagrams.net/`. *CST Studio* and *Matlab* have been used by the author to generate the remaining figures. The sources for the figures are listed in the figure captions.

Prefixes and units are written in accordance to the SI system of units. The bibliography is found after the main report and before the appendix. The references are numbered in alphabetical order according to the author's surname.

The author would like to thank Kim Olesen for his assistance in the antenna lab at Aalborg University including measurements of the antenna. The author would also like to thank Rocio Rodriguez Cano for her active role as supervisor during the project.

Aalborg University, May 16, 2024

---

Rikke Udengaard  
<rudeng20@student.aau.dk>

# Chapter 1

## Introduction

add abstract (last thing to do!!)

add introduction (second to last step!!)

## Chapter 2

# Technical Analysis

The purpose of beam steering is to increase the antenna gain in a specific direction to achieve reduction in interference and to save power. This is enabled by having the main lobe of a radiation pattern of a directional antenna pointing towards the target of transmission and reception. Beam steering is purposeful for narrow directional beams. Beam steering can be performed using manual, mechanical or electronic with the main differences being type of implementation and increasing speed of change of directivity from manual to electronic [19] [7]. This chapter explores the properties of antennas and beam steering methods, in order to understand antenna beam steering.

### 2.1 Fundamentals of Antennas

In order to develop a beam steering device for antennas it is necessary to understand antennas and their properties. Propagation, polarization, radiation characteristics are all properties of antennas that can vary based on the type of antenna.

#### 2.1.1 Propagation In Free Space

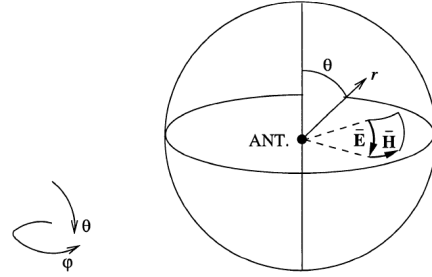
Propagation of radio waves can be described with Maxwell's equations using the spherical coordinate system  $(r, \theta, \phi, t)$  for antennas. The Maxwell equations in differential form are as follows

$$\begin{aligned}\nabla \times \mathbf{E} &= -\frac{\partial}{\partial t} \mathbf{B} \\ \nabla \times \mathbf{H} &= \mathbf{J} + \frac{\partial}{\partial t} \mathbf{D} \\ \nabla \cdot \mathbf{B} &= 0 \\ \nabla \cdot \mathbf{D} &= \rho\end{aligned}\tag{2.1}$$



with  $\mathbf{E}$  being the electric field with unit  $[\text{V m}^{-1}]$ ,  $\mathbf{B}$  being induction  $[\text{T}]$ ,  $\mathbf{H}$  is magnetic field  $[\text{A m}^{-1}]$ ,  $\mathbf{D}$  being dielectric displacement  $[\text{A s m}^{-2}]$ ,  $\mathbf{J}$  being the current density  $[\text{A m}^{-2}]$  and  $\rho$  being electric charge density  $[\text{C m}^{-3}]$ .

The electric field and the magnetic field are always connected; the electric field is created by the magnetic field and vice versa. The electric field and the magnetic field in spherical coordinates are illustrated on figure 2.1 below.



**Figure 2.1:** Electromagnetic field around a small antenna in far field range visualised in the spherical coordinate system [6, p. 58].

As visualised on figure 2.1 the electric field only depends on the  $\theta$ -component and the magnetic field only on the  $\phi$ -component when in the far field. The vector  $\mathbf{r}$  is direction of the observation point. The length of  $\mathbf{r}$  is the distance to the observation point ie. distance between transmitting and receiving antenna [6, p. 59].

The Poynting vector describes the power density and direction of the Electro-magnetic flux and is the cross product of the electric and magnetic field

$$\mathbf{S} = \mathbf{E} \times \mathbf{H} \quad [\text{W m}^{-2}] \quad (2.2)$$

which points in the same direction as the wave propagation [18, p. 3].

### Multipath propagation

Because electromagnetic waves can reflect on surrounding surfaces, be changed by condition of the transmission medium or be Doppler shifted due to movement of objects, receiver or transmitter, the signal that reaches the receiving antenna can have travelled other paths than the direct line-of-sight path from the transmitter. The received signal is a summation of all input signals regardless of phase angles, phase shifts or direction and therefore the received signal might be distorted [14, pp. 1-2].

### 2.1.2 Polarization

Different antenna designs have different radiation patterns and polarization. Table 2.1 lists a number of different antenna designs and their polarization

Type	Polarization
Isotropic antenna	
Dipole antenna	Linear
Patch antenna	Linear, circular
Horn antenna	Linear, circular
Helical antenna	Circular

**Table 2.1:** Table showing polarization of some typical antenna designs [18, p. 11].

Polarization describes the classification of the plane propagated wave, which is a electromagnetic wave that propagates with constant velocity in a specific direction. The direction of the propagation of the electromagnetic wave is always perpendicular to the direction of the electric field  $\mathbf{E}$  and both are perpendicular to the direction of the magnetic field  $\mathbf{H}$ . If the direction of the electric field is constant with time and position, the polarization of the propagated wave is classified as linear. If the direction of the electric field changes by rotating uniformly around the axis of the propagated wave, the wave has circular polarization. The polarization of the receiving and transmitting antennas affects how the signal is detected. The signal is best received when the polarization of the receiving antenna is the same as the polarization for the transmitting antenna. Mismatch in polarization will result in less received signal power or more signal noise [12, p. 82-84].

### 2.1.3 Radiation Characteristics

The electromagnetic waves are radiated to the near field and then far field. The far field is mathematically described as the distance  $r > R_2$ , with  $R_2$  defined as

$$R_2 = \frac{2D^2}{\lambda} \quad [\text{m}] \quad (2.3)$$

with  $D$  being the largest dimension of the antenna or antenna array and  $\lambda$  being the wavelength of the carrier frequency [18, p. 4].

The radiation characteristics of an antenna can be described by the directivity, which is defined as the ratio of the maximum power density  $S(\theta, \phi)_{max}$  radiated to the average power density  $S(\theta, \phi)_{avg}$  radiated by an antenna. The directivity is unitless [12, p. 63]. An isotropic antenna is a theoretical antenna which radiates homogeneously in all directions, meaning that the magnitude of the power density

vector  $\mathbf{S}$  at a distance vector  $\mathbf{r}$  is constant

$$D(\theta, \phi) = \left| \frac{S(r, \theta, \phi)}{S_{max}} \right| = 1 \quad (2.4)$$

It is this theoretical isotropic radiator that the gain of antennas are in respect to. The gain of a directive antenna in a certain direction is called the antenna gain  $G$  [18, p. 12]. The directivity doesn't depend on the distance  $r$  in the far field meaning that at the receiver antenna, the relation  $r \gg R_2$  is assumed.

The total radiated power  $P_r$  is found by the surface integral of the power density. Assuming a spherical surface, the total radiated power is described as

$$P_r = \int_0^{2\pi} \int_0^\pi S(\theta, \phi) r^2 \sin \theta d\theta d\phi \quad [\text{W}] \quad (2.5)$$

And further, averaging the radiated power over every direction in the sphere gives the relation

$$P_{avg} = \frac{P_r}{4\pi r^2} \quad [\text{W m}^{-2}] \quad (2.6)$$

Replacing the average power density  $S(\theta, \phi)_{avg}$  by the average power in every direction  $P_{avg}$ , the directivity of an antenna  $D(\theta, \phi)$  can be defined as

$$D(\theta, \phi) = 4\pi r^2 \frac{S(\theta, \phi)_{max}}{P_r} \quad (2.7)$$

with  $S(\theta, \phi)_{max}$  being the maximum power density and  $P_r$  being the total radiated power of the antenna.

The power of the source  $P_{s,t}$  to a transmitting antenna might not equal the radiated power  $P_{r,t}$  due to power loss  $p_{l,t}$ . Power loss can happen because of reflection loss in the input medium (typically cable), conductor loss and inductor loss. The efficiency of the transmitting antenna  $\eta$  is described as the ratio of the radiated power to the sourced power

$$\eta = \frac{P_{r,t}}{P_{s,t}} = \frac{P_{r,t}}{P_{r,t} + P_{l,t}} \quad (2.8)$$

The gain  $G(\theta, \phi)$  of the antenna is the effective directivity, meaning how well the receiving or transmitting antenna is able to convert, respectively, electromagnetic waves or power into the other. The gain of a transmitting antenna can be calculated as

$$G_t(\theta, \phi) = \eta D_t(\theta, \phi) = 4\pi r^2 \frac{S_t(\theta, \phi)_{max}}{P_{s,t}} \quad (2.9)$$

or expressed in decibel with respect to the isotropic radiator [18, p. 10] [22, pp. 1.8-1.10].

$$g_{dBi} = 10 \log_{10}(G) \quad [\text{dB}] \quad (2.10)$$

### 2.1.4 Friis Transmission Equation

The Friis transmission equation explains how the received power at a receiver antenna is related to the power of the transmitting antenna. The receiver antenna receives energy from the transmitting antenna and the effectiveness of this is described as the effective area  $A_r(\theta, \phi)$  assuming that the antenna is placed in the origin of the spherical coordinate system. If the antenna has the property of reciprocity, the effective area and the gain of the receiver antenna is related by

$$A_r(\theta, \phi) = \frac{\lambda^2}{4\pi} G_r(\theta, \phi) \quad [\text{m}^2] \quad (2.11)$$

If the gain of the transmitting antenna  $G_t$  is in the direction of the receiver antenna  $G_r$  then the angular dependencies of the antenna properties can be suppressed. The power of the receiving antenna is equal to the power density  $S_t(\theta, \phi)$  multiplied by the effective area of the receiver antenna  $A_r$ , expressed as

$$P_r = S_t A_r \quad [\text{W}] \quad (2.12)$$

As previously mentioned the directivity of an antenna does not depend on the distance  $r$  from the antenna, and likewise so with the power density  $S_t$ , so the value of  $S_t$  is equal regardless of the distance from the antenna in the far field with respects to the angular dependencies. Substituting  $S_t$  and  $A_r$  in equation 2.12 for  $S_t$  isolated in equation 2.9 and  $A_r$  from equation 2.11 yields

$$\begin{aligned} P_r &= \frac{G_t P_t}{4\pi r^2} \frac{\lambda^2 G_r}{4\pi} \\ &= G_t G_r P_t \left( \frac{\lambda}{4\pi r} \right)^2 \end{aligned} \quad [\text{W}] \quad (2.13)$$

Also called Friis transmission equation [22, pp. 1.10-1.11].  $G_t$  is the gain of the transmitting antenna in the direction of the receiver and  $G_r$  is the gain of the receiving antenna in the direction of the transmitter. The radiation characteristics of an antenna in the far field is called the antenna radiation pattern and will look different depending on the design of the antenna. The radiation pattern is dependent on the angular properties  $\theta$  and  $\phi$  and is usually visualised in a plane parallel to the electric field and called an **E** plane pattern or elevation plane pattern, or parallel to the magnetic field and called a **H** plane pattern or Azimuth plane pattern [12, p. 79-80][18, p. 13-14].

### 2.1.5 S-Parameters

S-parameters are used to describe the input and output relationship of a system's ports at microwave frequency [20]. S-parameters are used because voltages and

currents can be difficult to measure directly in the microwave frequency spectrum. S-parameters describe a network in waves instead of voltages and currents [17]. S-parameters can be used to describe a  $n$ -port,  $n \geq 1$ , system and are dependent on frequency [20]. Some typical systems are one-port and two-port systems, visualised in figures 2.2 and 2.3.

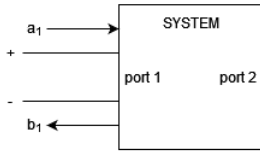


Figure 2.2: One-port system.

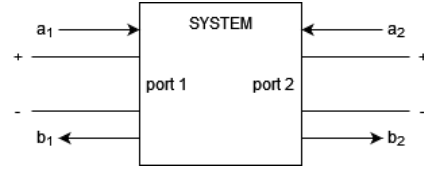


Figure 2.3: Two-port system.

The parameters  $a$  and  $b$  represent the wave flow in the system and they are a linear combination of the voltages and currents at the ports [17]. Looking at a two-port system, such as a system with a receiver antenna and transmitter antenna, the S-parameter matrix looks as follows:

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \quad (2.14)$$

The S-parameters represent different information about the system.

Parameter	Description
$S_{11}$	$S_{11} = \frac{b_1}{a_1}$ . Forward reflection coefficient. Describes the input return loss $\Gamma$ i.e. what is reflected from the port rather than radiated or absorbed.
$S_{12}$	$S_{12} = \frac{b_1}{a_2}$ . Reverse transmission coefficient i.e. how much is transmitted from port 2 to port 1.
$S_{21}$	$S_{21} = \frac{b_2}{a_1}$ . The forward transmission coefficient i.e. how much is transmitted from port 1 to port 2.
$S_{22}$	$S_{22} = \frac{b_2}{a_2}$ . Reverse reflection coefficient i.e. output matching.

Table 2.2: Explanation of S-parameters [20].

Ideally, an electrical load is designed to have the least possible absorption. This means the power transfer is maximised or the signal reflection is minimised, depending on the use case. This is done by matching the input impedance of the electrical load to the output impedance of the signal source, such as a cable. Minimal reflection is achieved when the complex output impedance is equal to the complex

input impedance and maximum power transfer is achieved when the complex output impedance is equal to the complex conjugate of the input impedance. If the input impedance of an electrical load is denoted  $Z_L$  and the characteristics output impedance of a signal source is denoted  $Z_0$  [20][17], the return loss can be calculated as

$$\Gamma = \frac{Z_L - Z_0}{Z_L + Z_0} \quad (2.15)$$

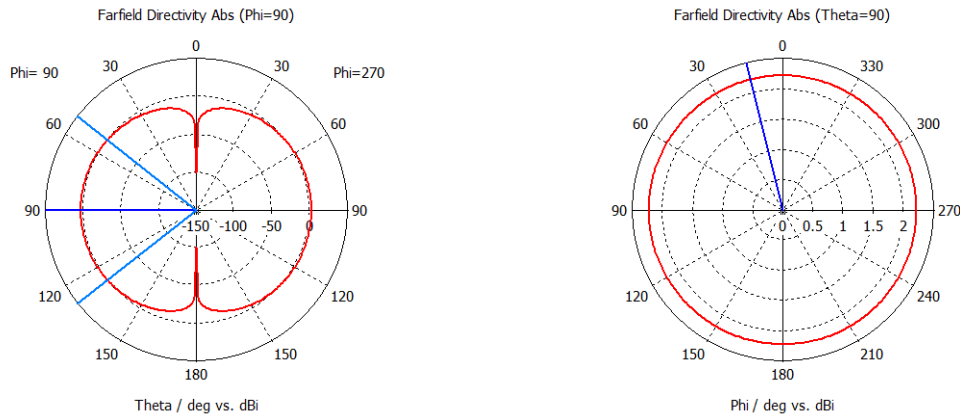
Using the reflection coefficient the output flow of the system's port one is equal to  $b_1 = \Gamma a_1$  [17].  $S_{11}$  and the reflection coefficient  $\gamma$  are related as follows:

$$S_{11} = 20 \log_{10} (|\Gamma|) \quad [\text{dB}] \quad (2.16)$$

### 2.1.6 Antenna Designs

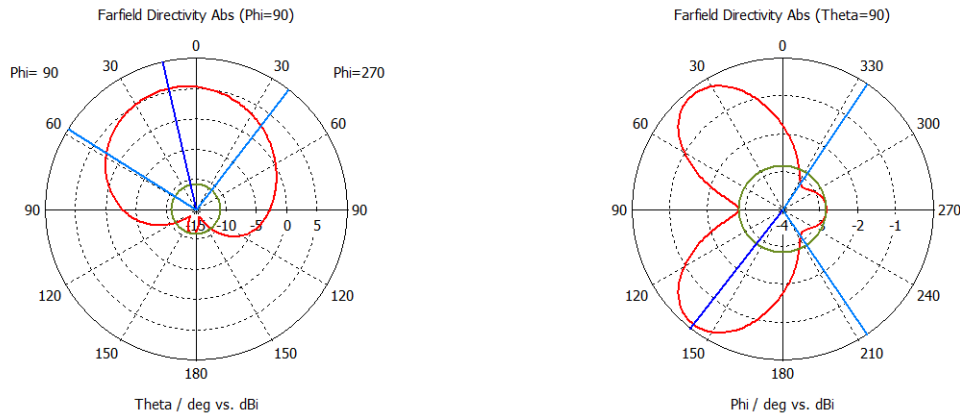
The design of the antenna affect the radiation characteristics, the bandwidth and the range of frequencies that the antenna is able to transmit or receive [12, p. 76]. Bandwidth is the range of frequencies on which the antenna is expected to operate in the wanted manner. There are several criterias that can be used to decide on the bandwidth for an antenna, for example the range where the reflection coefficient (see 2.1.5) is less than a specified value or when the polarization fits a certain shape (see 2.1.2). The antenna design affects the operating frequency range of the antenna and so an antenna can also be designed to operate with a given bandwidth. The radiation pattern of an antenna varies with the frequency but the general shape is primarily decided by the design of the antenna [3]. The antenna types described in table 2.1 are some typical design types along with the isotropic antenna which is a theoretical antenna [18, p. 11].

A center-driven dipole antenna is two wires or rods pointing at the opposite direction of each other, for example towards positive and negative  $z$  in the spherical coordinate system. In the Azimuth plane the radiation pattern is a circle centered at the centre of the dipole antenna, whereas in the elevation plane, the radiation pattern is two ears extending from the centre to either side (see figure 2.4) [18, pp. 12-14]. The figure 2.4 shows a typical radiation pattern for a one-wavelength dipole antenna. As seen, the dipole antenna has a very uniform pattern, but increasing the length of the dipole can increase the directiveness. However, the length also affects the input impedance. The dipole antenna will always have linear polarization [4].



**Figure 2.4:** Dipole antenna radiation pattern. Figures generated with *CST Studio* dipole antenna example ( $f = 2.4$  GHz).

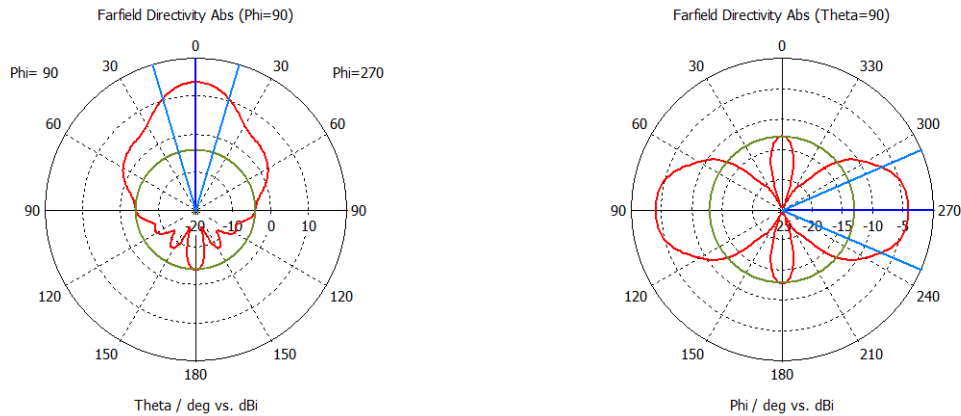
A patch antenna is in its simplest form a piece of metal on top of a grounded surface. The metal can have different shape and size to accommodate different operating frequencies, bandwidths and gains. The feeding of the patch can also affect the antenna parameters such as the impedance match. The patch antenna is usually only used for UHF applications, because the flatness of the antenna allows it to fit into narrow areas, where the patch size must also be small. The operating frequencies depend on the size of the patch. The patch antenna has lower efficiency than many other antenna design types, and a small bandwidth. The radiation pattern depends, as in all cases, on the design of the patch, and can have a large variety of shape [22, p. 7.1-7.5]. An example can be seen in 2.5.



**Figure 2.5:** Patch antenna radiation pattern. Figures generated with *CST Studio* ( $f = 2.4$  GHz).

Horn antennas come in many shapes and sizes which affect the radiation pattern and gain but common amongst the different designs is high beam directivity.

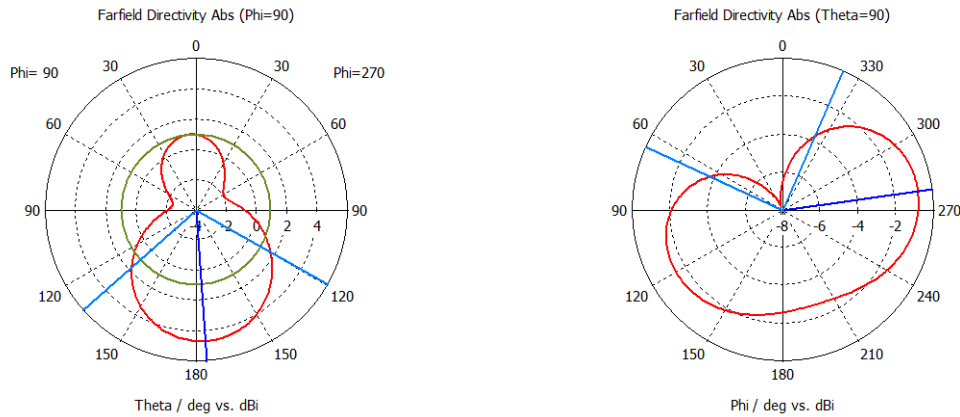
A pyramidal, rectangular horn antenna is a common horn antenna, and has a fan-shaped radiation pattern. The gain can be calculated by knowing its dimensions and the beamwidths of the fan can be changed by the varying the aperture dimensions. Horn antennas can be designed to cover both wide and narrow bandwidths, linear or circular polarization and have a certain gain. It is therefore that horn antennas are regarded as being able to fulfill many different applications [22, p. 14.1-14.3].



**Figure 2.6:** Horn antenna radiation pattern. Figures generated with *CST Studio* horn antenna example ( $f = 2.4$  GHz).

A helical antenna is one or several conductors in a helical shape connected to a ground plane and can be configured in many modes, usually *normal* or *axial* mode. Normal mode is achieved when the diameter of the helix is smaller than a wavelength and axial mode is when the circumference is close to the wavelength. A helix antenna has circular polarization in either right-hand or left-hand direction. A helix antenna has a wide spectrum for which impedance characteristics can be matched [22, p. 12.2].





**Figure 2.7:** Helical antenna radiation pattern. Figures generated with *CST Studio* ( $f = 2.4$  GHz).

## 2.2 Beam Steering Methods

Although the design of the antenna or antenna arrays decide the pattern of radiation, it is possible to modify this pattern or the direction of the pattern to achieve different directivities for other applications. The beam of the antenna can be steered in different directions for example, in order to cover a larger physical area with a single or few antennas or to focus a beam in a certain direction. Beam steering can be done manually, mechanically or electrically.

### 2.2.1 Electrical Steering

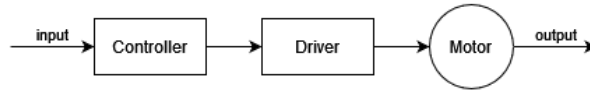
Electrical beam steering is the conventional steering method which builds upon beam forming methods, which entails forming a single beam from a phased antenna array and controlling the shape and direction of this beam. Beam forming build upon beam forming and is performed by having the array of multiple antennas and changing the phase of the signal in real-time. Electrical beam steering does therefore not require changing the hardware or the setup. Looking at a array of antennas, each element can be fed seperately to be able to change the phase and magnitude of each antenna element. Otherwise, if the antennas are fed with the same signal, the electromagnetic waves will combine and strengthen in the direction perpendicular to the antenna plane [13].

### 2.2.2 Manual and Mechanical Steering

Manual or mechanical steering differs from electrical steering in that it is not the beam that is steered but the antenna as a whole. For mechanical steering the antenna can be mounted on a rotating platform or arm that is controlled by a

motor. There are several motor types that can rotate a platform or arm holding an antenna, for example a step motor or a servo motor.

A Step motor has a rotor with two segments separated by a permanent magnet. The rotor segments have teeth which become poles of opposite polarity when the permanent magnet is axially magnetized. The rotor segments are also always skewed so that the teeth of one segment aligns with the gap between the teeth of the other rotor segment. The number of teeth determine the number of position steps around the center axis. The stator then generates a rotating magnetic field when the windings in the stator are supplied with current. The driver of the step motor controls the input current with PWM signals. The step motor usually works in an open loop configuration because of the design does not require feedback, however, overloading the motor might cause loss of synchronisation [21]. The figure 2.8 below shows a block diagram of an open loop configuration.

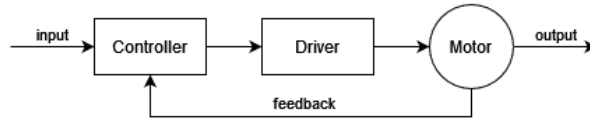


**Figure 2.8:** Open loop motor control block diagram [1].

Mathematically, the open loop configuration of the input  $R(s)$ , controller  $G(s)$ , driver  $D(s)$ , motor  $M(s)$  and the output  $Y(s)$  can be described by the open loop transfer function without feedback.

$$\frac{Y(s)}{R(s)} = G(s)D(s)M(s) \quad (2.17)$$

Servo motors use a radially magnetized rotor, which means the number of poles determines the number of position steps around the center axis. This usually mean that a servo motor has less available positions than a step motor. The servo motor works much like the step motor but an encoder is also used in the motor to give feedback to minimize error in the position. This makes the servo motor a PID loop configuration, such as seen on figure 2.9 [21].



**Figure 2.9:** Closed loop motor control block diagram [1].

Mathematically, the closed loop configuration of the input  $R(s)$ , controller  $G(s)$ , driver  $D(s)$ , motor  $M(s)$ , feedback  $H(s)$  and the output  $Y(s)$  can be described by

the closed loop transfer function

$$\frac{Y(s)}{R(s)} = \frac{G(s)D(s)M(s)}{1 + G(s)D(s)M(s)H(s)} \quad (2.18)$$

The step and servo motor differ when it comes to the relationship between the speed and the torque characteristics. The step motor has high torque at low speeds which drops significantly as the speed increases. The servo motor has a flat torque up until high speeds. Usually, the step motor is most ideal for applications where discrete position accuracy and fast responsiveness is valued whereas the servo motor is most ideal in applications with high or varying loads [21].

## Chapter 3

# Requirements and Delimitations

The purpose of this project is to develop a setup where a receiving antenna is able to detect a transmitting antenna at a arbitrary point in space and testing this setup. Detection of radio waves can for example be used for intruder detection or can be further developed to include data transmission and thereby achieving higher throughput if the receiving antenna is able to locate and focus on the transmitting device.

Concluding on these two scenarios, it is required that the receiving antenna must be able to be controlled ie. steered in any direction and the electromagnetic radiation in the area must be read.

### 3.1 Delimitations

The setup developed in this project will only be designed to work in a two-dimensional plane. This limits the need for beam steering in all directions. Moreover, the choice of antenna will affect the discrete measurement points in the plane because of how the 3dB bandwidth will limit the accuracy of measurements. This means that the project will focus on beam steering with mechanical turning, because electrical steering is above the scope and requirements of the project.

### 3.2 Functional Requirements

The following table 3.1 outlines the functional requirements. The requirements in this section describe the functionality of the setup.

ID	Requirement	Traceability
F.1	Mechanically controlled turning along Azimuth angle $\phi$	Chapter 3 introduction, section 3.1
F.2	Data reading of electromagnetic radiation in the surrounding space	Chapter 3 introduction, section 3.1
F.3	Controlled test environmental variables	Section 3.1
F.4	Automatic control of turning and measuring	Necessary to be able to reproduce test

Table 3.1: Table of functional requirements.

### 3.3 Technical Requirements

The table 3.2 below outlines the technical requirements. The requirements in this section describe the technical needs of the setup.

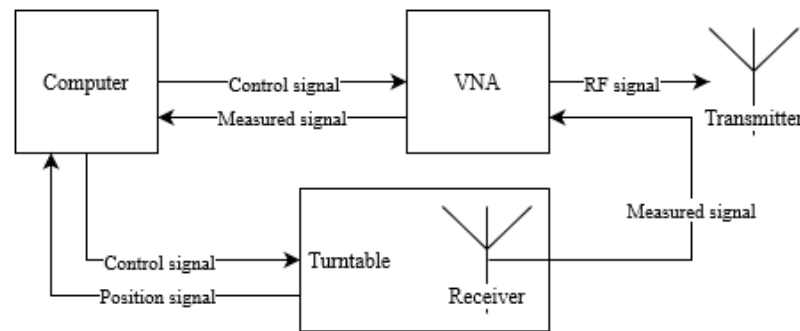
ID	Requirement	Traceability
T.1	Operational turning angle of $0^\circ$ to $360^\circ$	Chapter 3 introduction
T.2	High antenna directivity	Section 2.1.6
T.3	Matching antenna polarization	Section 2.1.2

Table 3.2: Table of technical requirements.

## Chapter 4

# Design of Antenna Beam Steering

The full system that will be designed includes a VNA, a turntable, two antennas and a computer. This chapter will describe the interfaces between these and their individual functionalities. In order to get an overview of the full system consider the figure 4.1



**Figure 4.1:** Overview of system design.

The system composes of three integrated modules and two antennas. The first antenna is a transmitter antenna and the second antenna is a receiver antenna, which is meant to be able to measure the gain in a wide area. This wide area is determined by the turning of the turntable on which the receiver antenna is mounted. Both the transmitter antenna and receiver antenna are connected to a VNA, but for the transmitter antenna, the functionality of the VNA used is the RF generator. For the receiver antenna, the received signal measured is then further sent to the computer for data analysis. The computer controls the turntable and the VNA. This includes defining settings before the analysis and taking decisions based on the returned data: the position of the turntable and the measured antenna signal from the VNA.

## 4.1 Antenna Design

The most important characteristics of the antenna chosen for the design is that it is directional. The antenna must be directional in order to detect transmission from another antenna. The receiver and transmitter antenna must have the same phase characteristics to ensure that this does not introduce loss in the system. Therefore, both antennas must be designed to match in this aspect. The receiver antenna and the transmitter antenna is chosen to be the same type of antenna with the exact same shape. This means that all antenna parameters are the same.

The chosen antenna is a homemade, pyramidal horn antenna. It has a rectangle cross-section, straight curvature of the side walls and therefore a linear polarization. The aperture of the antenna is XX by XX.

make figure of antenna+insert image

consider adding calculations of gain at different frequencies

## 4.2 Choice of Turntable

The turntable must be able to be remotely controlled and precise, so that the angular position of the antenna can be determined confidently. The turntable must be able to turn to a remotely specified angle at each turn.

The *Head Acoustics GmbH HRT I* turntable was available and fulfills the requirements. It has the ability to turn  $360^\circ$  with angle steps of  $0.1^\circ$ . It can be manually controlled and remotely controlled with software tools. It is able to turn at a maximum speed of 2 rpm by a stepper motor [8].

## 4.3 Choice of Vector Network Analyzer

The VNA must have a minimum of two ports, one for input and one for output, be able to be calibrated to normalize the transmission lines to the antennas and must be able to be remotely controlled. Moreover, it must cover the frequency range of the horn antennas. The *Rohde & Schwarz ZVB8* covers the range from 300 kHz to 8 GHz with down to 1 Hz resolution. It can have a measurement time of  $< 4.5$  ms with simultaneous data transfer, which allows for faster execution of tests. The internal noise level is  $< -110$  dBm [16].

## 4.4 Design of Software Control Program

Using one computer for the control of every aspect of the system allows for the devices to be programmed to work concurrently. Both the VNA and the turntable

are known to be able to be controlled in Python, there the control of the turntable and VNA is programmed in Python. The control of the turntable is implemented as advised by the manual (see [9]), with the supplied software *RC-HRT I*. However, in order to control the turntable concurrently with the VNA, the *RC-HRT I* software is interfaced via Python with the module *pywin32*, which provides access to the Windows APIs [11]. *Rohde & Schwarz* have published a Python module for control of the VNA which is used. The module adds an API for the remote control of the VNA, which is communicated with over TCP using SCPI-commands (*Standard Commands for Programmable Instruments*). Finally, thread parallelism is achieved with the module *threading*, which is what allows for the design of the control flow to work more efficiently.

#### 4.4.1 Communication Interfaces

The communication interfaces of the program is limited by the choice of the VNA and the turntable. The *HEAD Acoustics, HRT I* turntable that has been chosen can be controlled by serial communication using the RS485-standard [8]. However, it can also be controlled by its accompanying Windows driver and software control program, which can be interfaced to Python with the *Pywin32* module that accesses the Windows APIs. This requires the Windows registry to be updated with the UUID found in the control interface manual. With the Windows APIs accessible, the turntable can be controlled with predefined methods [10].

The *R&S ZVB8* VNA has a LAN type interface for control and data transfer [16]. LAN allows for mutual communication between devices. LAN covers both the physical layer in the form of cabling and network cards, the latter which also contains part of the link layer in the form of logic decisions, and the link layer which includes the link protocol [5, p. 153]. In order to communicate across a LAN the computers must be connected to the same network and know each other's addresses. Using the TCP/IP protocol the communication can be extended further outside the LAN to the wider internet. Even on the same LAN, two computers can communication with each other using their IP addresses [5, p. 174-175]. On top of this, TCP ensures a connection between the two applications running on the computer and the VNA. At each end of the connection a port number is used to identify the applications. Moreover, the TCP header includes an acknowledgement number which the applications use to acknowledge to each other, that the previous message was correctly received [5, p. 313]. The content of the TCP packets is SCPI commands and device-specific commands that follow the SCPI-standard [15, p. 5.4].



#### 4.4.2 Concurrency

The turntable is controlled via serial communication while the data from the VNA is read via a network cable. Because communication with the turntable and the turning process and communication with the VNA are both I/O bound tasks, there is a possibility for some waiting time instead of CPU time in a control program. Therefore, in order to achieve real-time communication and data processing, the wait time is exploited by use of the Python module *Threading*. *Threading* uses a single processor and pre-emptive multitasking to achieve concurrency [2]. This means that at any time one thread is not using CPU computation the processor can switch to another thread and continue computation here. The processor also saves the current state of each thread so that it can return to the exact same place. However, this also means that reading and writing to global variables must be protected, because the processor can switch in the middle of a statement [2]. Likewise, if a thread needs to wait for another thread to execute a specific task, it is necessary that the user defines a wait flag, which synchronises the wait between relevant threads.

## Chapter 5

# Implementation

In this chapter the implementation of the software to control the turntable and the VNA is described based on the design principles laid out in chapter 4. The background for the implementation is to fulfill the requirements set out in chapter 3. In the project a Windows computer is used to interface to the turntable and VNA, and the software implementation is influenced by this choice.

The VNA and the turntable are controlled in Python. The code is set up to have a module for each device and a main control program. In the main control program the concurrency is also implemented. The software is programmed with an object-oriented approach.

### 5.1 Turntable Control

As previously mentioned in section 4.4.1 the turntable is controlled in Python with the *Pywin32* module that gives simple access to the Windows APIs. The turntable comes with a interface documentation document that further elaborates on the functions that the turntable will accept. These are implemented in the control program.

First, a `TurnTableController` class is defined. The initialisation of the class will call the `__init__` method that creates the instance variables and tries to connect to the turntable. The connection is checked with a `if/elif/else` statement. If the connection is not possible, the error handler is set to exit the program. Else, if the connection is established the settings for the turntable will be set. The final `else` statement is used to catch any other state, which is treated as an error and the program is exited.

The turntable has multiple settings that need to be specified in order to ensure correct operation. These settings are set in the method `turntable_set` defined in the turntable class in the turntable module. The speed of the rotation, the rpm, and the acceleration function are predefined from the manufacturer to be between

0 and 2 rpm and with four different levels of steepness in acceleration to full speed. Likewise the turntable can either turn in a bipolar or unipolar direction. The desired unipolar direction is set, however, if this setting can not be made, the angle\_max is updated from 360, the unipolar maximum, to 180, the bipolar maximum value. Lastly, the current position cur\_pos is read and compared to the start position start\_pos in a while loop, and the turntable is instructed to turn to the start position in the opposite direction of its normal turning direction.

```

1 def __init__(self, instance: str, ttc, clockwise: bool = False,
  start_pos: float = 0.0, angle_min: float = ANGLE_MIN, angle_max:
  float = ANGLE_MAX) -> None:
2     """ Initialize instance variables, connect and set settings. """
3     self.instance = instance
4     self.ttc = ttc
5     self.clockwise = clockwise
6     self.start_pos = start_pos
7     self.angle_min = angle_min
8     self.angle_max = angle_max
9     if self.ttc.Count == 0:
10        logger.error("No turntable is connected.")
11        exit()
12    elif self.connect() == EConnectionState.ecsConnectedOn.value:
13        self.turntable_set(rpm=2, func=EAccelerationFunction.afImpulse
  , start_pos=self.start_pos)
14    else:
15        logger.error("Unknown error in startup sequence.")
16        exit()
17
18 def turntable_set(self, rpm: int, func: EAccelerationFunction, start_pos:
  float) -> None:
19    """ Establish basic settings: rpm, acceleration function and start
  position. """
20    cur_pos: int = round(self.position)
21    try:
22        self.tt.Velocity = rpm
23        self.tt.AccelerationFunction = func.value
24    except Exception as e:
25        logger.error(f"Unable to set settings for turntable {self.
  instance}, exiting with error code {e}.")
26        exit()
27    if self.tt.DisplayPolarity == EPolarity.epolBipolar.value:
28        try:
29            self.tt.DisplayPolarity = EPolarity.epolUnipolar.value
30        except Exception:
31            self.angle_max = 180.0
32            logger.error(f"Unable to set polarity to unipolar for
  turntable {self.instance}.")
33    while cur_pos != round(start_pos):
34        logger.info(f"Current position is {cur_pos} not start position
  {round(start_pos)}. Moving {self.instance} to start position.")

```

```

35         if self.clockwise:
36             self.go_to_CCW(start_pos)
37         else:
38             self.go_to_CW(start_pos)
39         cur_pos = round(self.position)
40         logger.info(f"Settings are velocity: {round(self.tt.Velocity)},
41                     function: {EAccelerationFunction(self.tt.AccelerationFunction)}.")
42         logger.info(f"Current position for {self.instance} is {cur_pos}.")

```

**Listing 5.1:** Method to establish settings for turntable and reach start position.

The run method of the turntable takes the value of the increment `inc` as input argument and accesses the current position `cur_pos` and the value of the turning direction `clockwise` to determine and turn either in a clockwise or counter-clockwise direction. Finally the position is checked against the maximum and minimum angle values which are defined in the beginning of the module to be `ANGLE_MIN = 0.0` and `ANGLE_MAX = 360.0`, effectively making this statement redundant in this specific setup, as the turntable position cannot have any value outside this range.

The `step_CW` method is equal to the `step_CCW` method except for the turning direction. First, the size of the step is set before the step is made, according to the definition in the documentation. For the `go_to_CW` method, the interface method to be called is `GoToCW` which that a floating point number of degrees as input. Both methods include the option to force the program to execute no other tasks while the turntable is turning which is done by setting the `wait` flag which will call the `wait_while_driving` method. This method calls `time.sleep()` which effectively utilises the single processor fully. The `wait_while_driving` method also includes a wait time check that will break the wait while loop and allow the code to continue executing.

```

1 def run(self, inc: float) -> None:
2     """ Run the turntable. Changes the current position with [inc]
3     degrees. """
4     cur_pos: float = self.position
5     if self.clockwise:
6         self.step_CW(inc)
7     else:
8         self.step_CCW(inc)
9     if (self.angle_min > cur_pos > self.angle_max):
10         logger.error(f"Current position is illegal. Resetting: {
11             self.instance}")
12         self.reset(self.instance, self.clockwise)
13
14 def step_CW(self, degrees, wait: bool = False) -> None:
15     """ Step [degrees] in clockwise direction. """
16     self.tt.StepSize = float(degrees)
17     self.tt.StepCW()
18     if wait:

```

```

17         self.wait_while_driving()
18
19     def go_to_CW(self, degrees: float, wait: bool = False) -> None:
20         """ Go to [degrees] while moving in clockwise direction. """
21         self.tt.GotoCW(float(degrees))
22         if wait:
23             self.wait_while_driving()
24
25     def wait_while_driving(self) -> None:
26         """ Ensures that the program waits for the turntable to reach
27             position before execution further code. """
28         seconds_waited = 0.0
29         while self.tt.IsMoving:
30             time.sleep(0.5)
31             seconds_waited += 0.5
32             if seconds_waited > 120:
33                 logger.warning('Timeout while waiting for stop.')
34                 break

```

**Listing 5.2:** Methods for turning the turntable to the wanted position.

The reset method used in the run method's position check will first call the stop method. This method will force stop the turntable by calling MoveAbort(). The reset method then rotates based on the setting of the clockwise boolean variable to the start position start\_pos.

```

1     def stop(self) -> None:
2         """ Stop turning. """
3         if self.tt.IsMoving:
4             self.tt.MoveAbort()
5             logger.warning(f"Turntable was moving while connection was
6                 stopped for {self.instance}.")
7             logger.info(f"Turntable {self.instance} is stopped.")
8
9     def reset(self) -> None:
10        """ Reset Turntable to start position. """
11        self.stop()
12        if self.clockwise:
13            self.go_to_CCW(self.start_pos)
14        else:
15            self.go_to_CW(self.start_pos)

```

**Listing 5.3:** Method for resetting the turntable to start position.

The turntable class includes a single attribute, which is the position of the turntable. The position is established as an attribute by using the @property decorator on the method returning the position variable. By setting the position as an attribute without a setter decorator the position variable becomes read-only.

## 5.2 VNA Control

The VNA is interfaced via a TCP connection. This means the implementation of the VNA is independent on the underlying system.

The automation of the measurements with the VNA are made with the *Rohde & Schwarz* instrument library, that provides an easy interface instead of implementing the SCPI commands. Firstly, the connection with the VNA must be established. The IP address of the VNA has been manually set in the settings on the VNA and the assigned port has been extracted from the computer system.

The VNA control is implemented as a class with a number of method. In the `__init__` method of the class the connection is established to the VNA over TCP. Further, relevant settings are provided as class variables. This is for example the frequency on which the measurements must be made. This is provided as a floating point number when the VNA class is instantiated. The number of measurements, also called channel points, is also set in the `__init__` method to 1, since only one measurement needs to be made at one frequency. If the measurements should've been made over a frequency range, the `start_frequency_Hz` and `stop_frequency_Hz` could be modified accordingly and the number of measurements to be made in the range must be set with the `points` variable.

The `connect` method implements an exception handler in case the `open_tcp` method returns an error and exits the program, because it is not possible to execute further without this connection. If the connection is successfully established a single channel with the identifier `ch=1` is created as well.

The specific S-parameter to be measured is also set when the class is instantiated. The value is used in the `vna_set` method that establishes the trace settings, such as which S-parameter to measure and the format of the measurements, which is magnitude in dB.

```

1 def __init__(self, trace_id: str, s_param: str, freq: float,
2   ip_address = '172.0.0.1', port: int = 5025, channel = 1) -> None:
3     """ Initialize instance variables and connect.
4     Instrument Type: ZVB8 with 2 Ports
5     Part Number: 1145.1010k08
6     Serial Number: 100113
7     Device ID: 1145.1010K08-100113-DD
8     IEC Bus Address: 20
9     IP Addresses: IP Address 172.0.0.1 (Localhost) Subnet Mask:
10    255.0.0.0 """
11    self.port: int = port
12    out: tuple = self.connect(ip_address, port, channel)
13    self.vna: Vna = out[0]
14    self.ch: Channel = out[1]
15    self.trace_id = trace_id
16    self.s_param = s_param
17    self.ch.start_frequency_Hz = freq, 'GHz'

```

```

16     self.ch.stop_frequency_Hz = freq, 'GHz'
17     self.ch.points = 1
18
19 def connect(self, ip_address: str, port: int, ch: int) -> tuple:
20     """ Try to connect to VNA with TCP. """
21     try:
22         sock = Vna()
23         sock.open_tcp(ip_address, port)
24     except Exception as e:
25         logger.error(f'Cannot connect to VNA because {e}.')
26         exit()
27     else:
28         logger.info(f'Connection established to VNA. Creating channel {ch}.')
29         channel = sock.channel(ch)
30         return sock, channel
31
32 def vna_set(self) -> None:
33     """ Establish trace for VNA. """
34     self.vna.create_trace(self.trace_id, 1, self.s_param)
35     self.vna.trace(self.trace_id).format = TraceFormat.magnitude_dB

```

Listing 5.4: Method for initialisation of VNA settings including creating VNA trace.

The run method returns the data measured formatted as a tuple with frequency at index 0 and power at index 1. The run method calls the trace method from the Vna class from the *Rohde & Schwarz* instrument library to retrieve the trace ID, to call the `measure_formatted_data` method on the trace.

```

1 def run(self) -> tuple:
2     """Measure S-Parameters for [trace_id]."""
3     x, y = self.vna.trace(self.trace_id).measure_formatted_data()
4     return x, y

```

Listing 5.5: Method for getting measurements from VNA.

The VNA can be completely reset by calling the `preset` method from the *Rohde & Schwarz* instrument library, however, this will also remove the calibration of the VNA that is performed to normalize the transmission line. The calibration data can be saved as a file and reopened for continuous use of the same calibration data. The calibration must therefore be performed manually before the execution of the setup, otherwise the data will be faulty.

## 5.3 Concurrency in Main Control Flow

The start and end positions and the angle increase are initialised as constants. The current position `cur_pos` must be read both from the VNA and the turntable, and must be set by the turntable, therefore this variable is defined globally together with a lock `lock_cur_pos`, so that only one thread will access the variable at a

time. Likewise with the variable for maximum position `max_pos` which also has its own event handler `max_pos_event_handler`, so that the turntable thread will wait for the VNA thread to find the maximum position before reading the variable and turning to that position.

```

1 START_POS = 10.0
2 END_POS = 150.0
3 INCREASE = 20.0
4
5 lock_cur_pos = Lock()
6 cur_pos: int = None
7
8 max_pos_event_handler = Event()
9 lock_max_pos = Lock()
10 max_pos: int = None

```

**Listing 5.6:** Global constants and variables.

The turntable is interfaced with the windows API made available with the *Pywin32* module. The code in line 2 to 3 is the initialisation of this interface communication. Then, the global variables are accessed via *Python's* `global` keyword. The current position is read from the `turntable.position` property defined in the *turntable* module. The while loop turns the turntable from the `START_POS` to the `END_POS` in increments of `INCREASE`. In the while loop, the turntable sets its own event handler `turntable_event_handler` to `true` to indicate to the VNA thread that it can now run, and then the turntable thread waits for the VNA thread. Hereafter the turntable turns and updates the current position `cur_pos`. Once the end position `END_POS` is reached, the while loop is exited and this is indicated to the VNA thread with the end event handler. Lastly, the turntable thread turns to the maximum position `max_pos` when the VNA thread has indicated, that this variable has been set.

```

1 def run_tt_in_thread(ttc: CDispatch, turntable_event_handler: Event,
2   vna_event_handler: Event, end: Event, ttc_id) -> None:
3     CoInitialize()
4     ttc = Dispatch(CoGetInterfaceAndReleaseStream(ttc_id, IID_IDispatch))
5     turntable = TurnTableController(instance="hrt i (64980128)", ttc=ttc,
6       clockwise=True, start_pos=START_POS)
7
8     global cur_pos
9     global max_pos
10    count: int = 1
11
12    with lock_cur_pos:
13      cur_pos = round(turntable.position)
14
15    while START_POS <= cur_pos < END_POS:
16      turntable_event_handler.set()
17      turntable_event_handler.clear()

```



```

16     vna_event_handler.wait()
17     turntable.run(INCREASE)
18     with lock_cur_pos:
19         cur_pos = round(turntable.position)
20         logging.info(f"Current position for {turntable.instance} is
{cur_pos}.")
21         count += 1
22     turntable_event_handler.set()
23     turntable_event_handler.clear()
24     end.set()
25
26     max_pos_event_handler.wait()
27     with lock_max_pos:
28         if turntable.clockwise:
29             turntable.go_to_CW(max_pos)
30         else:
31             turntable.go_to_CCW(max_pos)
32
33     logging.info(f'Turntable thread is closed. {count} positions
measured.')
```

Listing 5.7: Thread function for running VNA.

Similarly the VNA thread accesses the global variables `cur_pos` and `max_pos` but then two lists for measurement position `data_pos` and power `data_pow` are defined. The while loop for the VNA is runs as long as the turntable thread has not set the end event handler to true, indicating that turning has finished. The while loop begins with waiting for the turntable to indicate it has finished turning with the `turntable_event_handler`. Then the power measurement is made with the execution of the code in line 10. The two lists containing the measurement data are updated by first calling the current position lock `lock_cur_pos` before appending to the lists. When the turntable thread has indicated that turning is finished, the VNA thread immediately calculates the maximum position and updates the global `max_pos` variable.

```

1 def run_vna_in_thread(vna: NetworkAnalyzer, turntable_event_handler:
Event, vna_event_handler: Event, end: Event) -> None:
2     global cur_pos
3     global max_pos
4     data_pos: list = []
5     data_pow: list = []
6     count: int = 0
7
8     while not end.is_set(): # run only when end == false
9         turntable_event_handler.wait()
10        _, pow = vna.run()
11        vna_event_handler.set()
12        vna_event_handler.clear()
13        logging.info(f'Power measurement is {pow}.')
14        with lock_cur_pos:
```

```

15         data_pos.append(cur_pos)
16         data_pow.append(pow)
17         count += 1
18
19     max_gain: float = max(data_pow)
20     with lock_max_pos:
21         max_pos = data_pos[data_pow.index(max_gain)]
22     logging.info(f'Max gain measured is {max_gain} at position {max_pos}
23     .')
24     max_pos_event_handler.set()
25     logging.info(f'VNA thread is closed. {count} measurements made.')

```

Listing 5.8: Thread function for running VNA.

The main function has to create the setup for controlling the turntable with the Windows API in a *Python* thread, the VNA instance and trace settings, the two threads, event handlers, start the threads and ensure that both threads close correctly when finished by using the `.join()` function on each thread.

```

1 def main():
2     logging.basicConfig(filename=f'./tests/test-{time.strftime("%Y%m%d-%H%M")}-log.txt',
3     filemode='a', format="%(asctime)s: %(name)s: %(message)s", level=logging.INFO,
4     datefmt="%Y-%m-%d %H:%M:%S")
5
6     CoInitialize()
7     ttc = Dispatch("TurnTableControlLib.TurnTableControl")
8     ttc_id = CoMarshalInterThreadInterfaceInStream(IID_IDispatch, ttc)
9
10    vna = NetworkAnalyzer(trace_id='trc1', s_param='s21', freq=5.65)
11    vna.vna_set()
12    logging.info(f'VNA with trace id {vna.trace_id} is created.
13    Measuring {vna.s_param}.')
14    logging.info(f'Settings are: {vna.get_settings()}')
15
16    turntable_event_handler = Event()
17    vna_event_handler_handler = Event()
18    end = Event()
19
20    turntable_thread = Thread(target=run_tt_in_thread, kwargs={'ttc_id':
21    ttc_id, 'ttc': ttc, 'turntable_event_handler':
22    turntable_event_handler, 'vna_event_handler':
23    vna_event_handler_handler, 'end': end})
24    turntable_event_handler.clear()
25
26    vna_thread = Thread(target=run_vna_in_thread, kwargs={'vna': vna,
27    'turntable_event_handler': turntable_event_handler,
28    'vna_event_handler': vna_event_handler_handler, 'end': end})
29
30    turntable_thread.start()
31    vna_thread.start()

```

```
24  
25     turntable_thread.join()  
26     vna_thread.join()
```

**Listing 5.9:** Main function.

Calling the `main` function ensures that the entire setup meaning the turning of the turntable and the measuring of magnitude of the received signal by the VNA runs automatically before saving the data to a text file for further analysis.

# Chapter 6

## Tests

This chapter includes test descriptions and results to validate the design against the requirements. First, the two directional antennas are tested to find their gain and ensure, that the transmitter and receiver have their maximum gain at the same frequency. Finally, the full test of the beam steering and transmitter locator is tested with the transmitter fixed at a known location in the test area.

### 6.1 Test of S-Parameters

The aim of this test is to test the reflection coefficient of the receiving antenna. The measurement will be used to choose the resonance frequency for the acceptance test.

#### Equipment

To perform the test the following equipment is needed:

- Rohde & Schwarz ZNA Vector Network Analyzer (10 MHz-43.5 GHz)
- Rohde & Schwarz ZN-Z54 Calibration Unit (9 kHz-40 GHz)
- Antenna cable

#### Procedure

The test is performed once. The following explains the procedure for the test:

1. Add power to VNA, connect antenna cable to VNA and Calibration unit.
2. Make calibration test by choosing *Cal* → *Quick Start Calibration* → *Apply* on the VNA.

3. Disconnect antenna cable and reconnect to DUT.
4. Set start and stop frequencies.
5. Ensure that VNA measures  $S_{11}$ -parameter by choosing  $Meas \rightarrow S_{11}$ .

The receiver antenna is a horn antenna which is designed to work in the spectrum from 4 GHz to 7 GHz. The measurement is performed in the spectrum 3 GHz to 8 GHz in order to encompass the entire antenna frequency spectrum, with steps of 25 MHz.

## Result

The following figure shows  $S_{11}$  at the measured frequencies

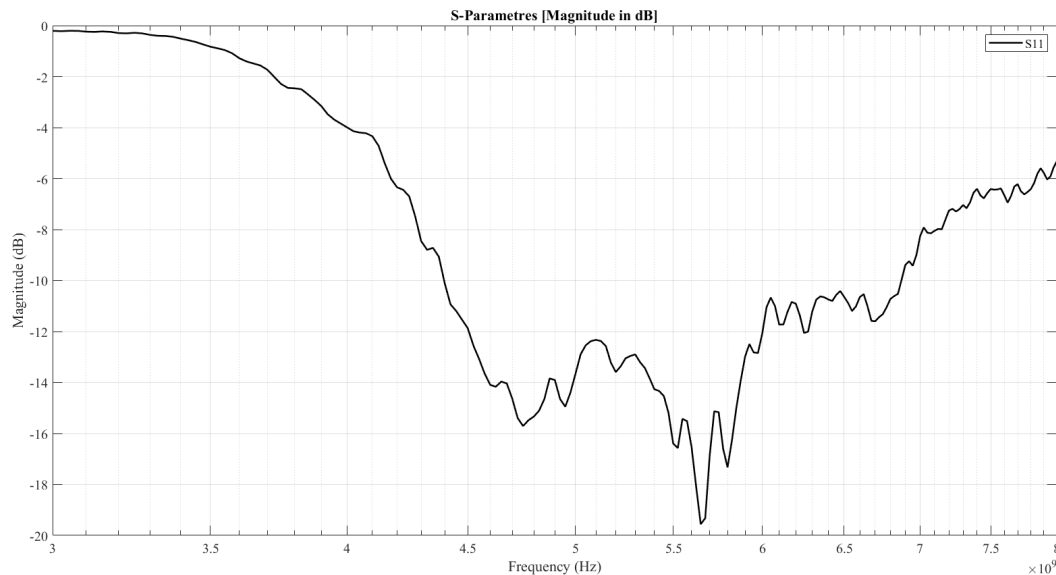


Figure 6.1: Measured  $S_{11}$ -parameter from 3 GHz to 8 GHz.

It can be seen that the antenna has a maximum  $S_{11}$  magnitude of  $-19.56$  dB at 5.65 GHz. This gives a reflection coefficient of  $\Gamma = 10^{-19.56/20} = 0.11$ . Almost 1 GHz away at 4.75 GHz the magnitude of  $S_{11}$  is  $-15.71$  dB which equals a reflection coefficient of  $\Gamma = 10^{-15.71/20} = 0.16$ .

## 6.2 Test of Radiation Pattern

The aim of this test is to know the directiveness of the receiving antenna. The measurement is used in the evaluation of the angle step of the turntable.

### Equipment

The test is performed in the anechoic chamber at Aalborg University with the provided setup equipment at the site. This includes

- Computer with relevant *MVG software*
- *MVG StarMIMO* in anechoic chamber

The *MVG StarMIMO* has a measurement bandwidth of 400 MHz to 6 GHz. As seen in the results in the test of S-parameters in section 6.1 the maximum reflection coefficient magnitude is at 5.65 GHz, therefore the step size of frequency spectrum for the radiation characteristics measurement is chosen to be 0.05 GHz.

### Procedure

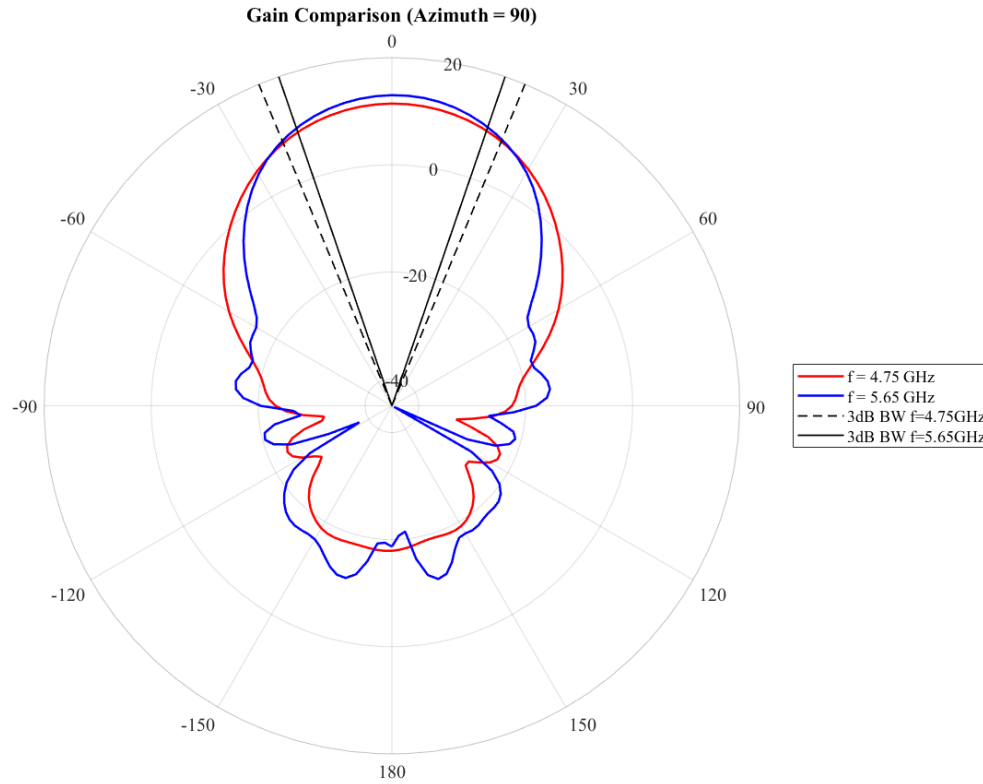
The following explains the procedure for the test:

1. Measure known antenna (*MVG SH800*) with known radiation characteristics. Use for gain reference.
2. Secure DUT to test platform.
3. Perform automated test by activating the measurement equipment outside the chamber.

Since the antenna is not designed for use below 4 GHz, this is set as the start frequency. The measurement equipment cannot measure above 6 GHz, therefore this is the end frequency. The *StarMIMO* measures at angles of  $15^\circ$  in both planes.

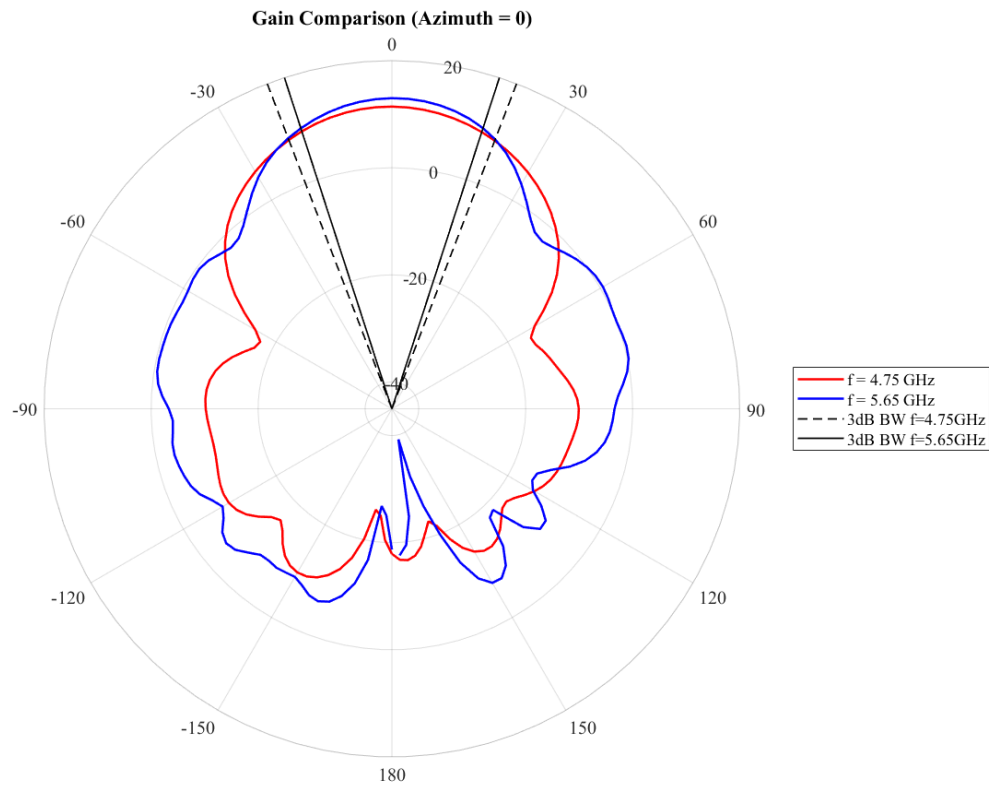
### Result

The data collected is imported into *Matlab* for visualization. The figure 6.2 shows the elevation plane of the measured horn antenna in the anechoic chamber. The red line shows the radiation pattern for the horn antenna at  $f = 4.75$  GHz. The blue line shows the radiation pattern for the horn antenna at  $f = 5.65$  GHz. The antenna has a higher gain at  $f = 5.65$  GHz of 12.99 dB, whereas at  $f = 4.75$  GHz the highest gain is 11.41 dB. The 3 dB-bandwidths are also plotted at  $19^\circ$  for  $f = 5.65$  GHz and at  $22.5^\circ$  for  $f = 4.75$  GHz.



**Figure 6.2:** Measured gain in the farfield at 4.75 GHz and 5.65 GHz (elevation plane).

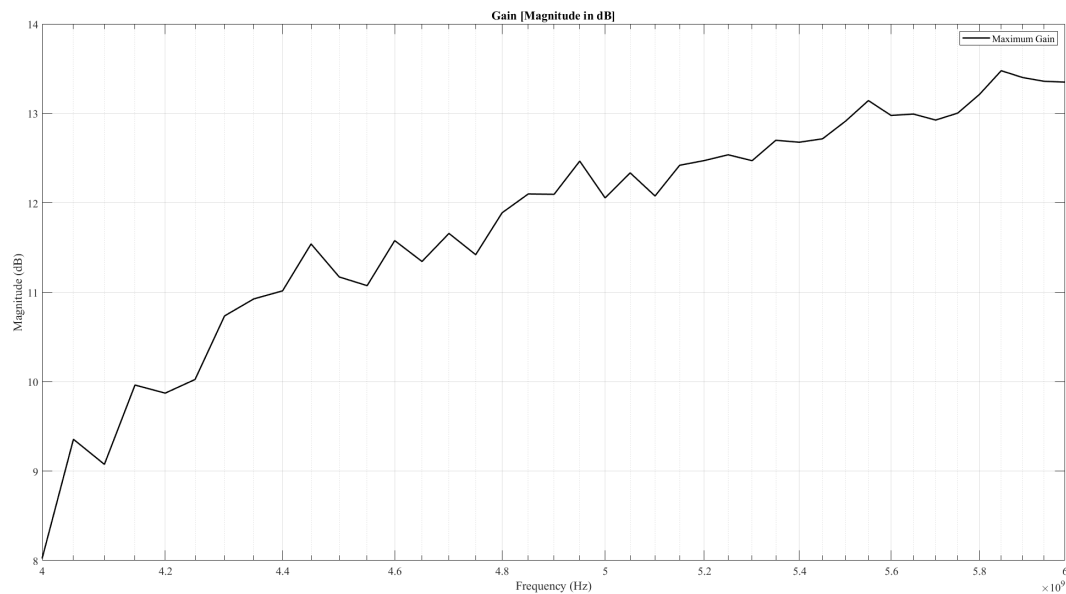
The figure 6.3 below shows the azimuth plane of the horn antenna as measured in the anechoic chamber. The red line shows the gain for the antenna at  $f = 4.75$  GHz and the blue line shows the gain for the antenna at  $f = 5.65$  GHz. The maximum gain is 12.99 dB at  $0^\circ$ .



**Figure 6.3:** Measured gain in the farfield at 4.75 GHz and 5.65 GHz (azimuth plane).

Simultaneously the maximum gain of the antenna over the frequency spectrum at any measured angle was measured as seen on the figure 6.4. The measurement shows that the gain, generally, increases as the frequency increases.





**Figure 6.4:** Measured maximum gain at any measured angle in the frequency spectrum from 4 GHz to 6 GHz.

## 6.3 Accept Test

The aim of this test is to test the full function of the developed product. The test must show that the receiver antenna on the turntable is able to scan the test area and measure the received power at fixed angles, before selection the location with the maximum received power and focusing its beam on that location.

elaborate introduction

### Equipment

To perform the test, the following equipment is needed:

- HEAD Acoustics Remote-operated Turntable, model HRT I 6498, with 24 V DC 60 W power supply
- D-sub 9-pin to USB-A cable
- PC with one USB-A port and one LAN port
- Rohde & Schwarz ZVB 8 Vector Network Analyzer
- Network cable (8-pin RJ-45 connector)
- Two 50  $\Omega$  directional antennas with XX frequency

- power supplies to the antennas ????

find exact values and names

Moreover, the test must be performed in a controlled environment, where the temperature is not below 0 °C or above 50 °C with a relative humidity in the range 20 % - 80 % [8].

add VNA environmental conditions

Test Execution

The following steps outline how to perform the test:

1. Do this. Refer to figure 6.5 to visualise the setup.

elaborate on the how

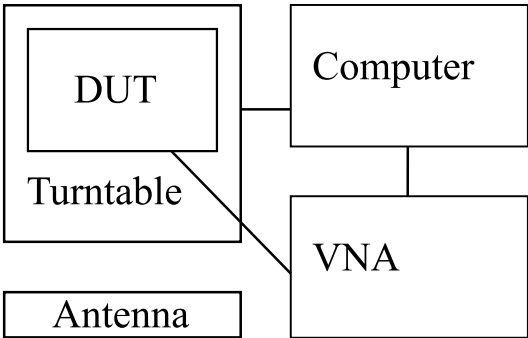


Figure 6.5: Setup for test of functionality of beam steering device.

improve this figure because it is ugly. Needs more names

Test Results

add table for test results and describe

Test errors

While the turntable is turning the cable connected to the horn antenna will move with it. This will affect the phase of the signal. However, since the purpose of the test is to measure magnitude, this error is will not effect the test result.

6.4 Summary of Tests

# Bibliography

- [1] GCM A/S. *Open vs. Closed Loop*. [gearcentralen.dk/en/open-vs-closed-loop/](http://gearcentralen.dk/en/open-vs-closed-loop/). Online; accessed 16-May-2024. u.a.
- [2] Jim Anderson. *Speed Up Your Python Program With Concurrency*. [realpython.com/python-concurrency](http://realpython.com/python-concurrency). Online; accessed 05-April-2024. u.a.
- [3] Peter Bevelacqua. *Bandwidth*. [antenna-theory.com/basics/bandwidth.php](http://antenna-theory.com/basics/bandwidth.php). Online; accessed 04-May-2024. 2015.
- [4] Peter Bevelacqua. *The Dipole Antenna*. [antenna-theory.com/antennas/dipole.php](http://antenna-theory.com/antennas/dipole.php). Online; accessed 04-May-2024. 2015.
- [5] Libor Dostálek and Alena Kabelová. *Understanding TCP/IP*. First edition. Packt Publishing Limited, 2006.
- [6] Hans Ebert and Povl Raskmark. *Grundlæggende Maxwell'ske Feltteori*. Fourth edition. Aalborg Universitet, 1998.
- [7] Sujanth Narayan. K. G, Velavikneshwaran. V, and James A. Baskaradas. "Design of Electronically Steerable Direction Shifting Microstrip Antenna Array using Beam Steering Technique". In: *2021 IEEE Indian Conference on Antennas and Propagation (InCAP)*. 2021, pp. 60–63.
- [8] HEAD Acoustics GmbH. *Data Sheet HRT I*. [cdn.head-acoustics.com/fileadmin/data/global/Datasheets/Data\\_Acquisition/HRT-I-Turntable-6498-Data-Sheet.pdf](http://cdn.head-acoustics.com/fileadmin/data/global/Datasheets/Data_Acquisition/HRT-I-Turntable-6498-Data-Sheet.pdf). Online; accessed 09-March-2024. 2023.
- [9] HEAD Acoustics GmbH. *HRT I Manual*. 2019.
- [10] HEAD Acoustics GmbH. *HRT I Turntable Control Interface Documentation*. 2018.
- [11] Mark Hammond. *Pywin32*. [pypi.org/project/pywin32](http://pypi.org/project/pywin32). Online; accessed 15-April-2024. u.a.
- [12] Andrea M. Mitofsky. *Direct Energy Conversion*. First edition. Trine University, 2018.
- [13] Electronics Notes. *Beamforming and Beamsteering Antennas*. [electronics-notes.com/articles/antennas-propagation/phased-array-antennas/beamforming-beamsteering-antenna-basics.php](http://electronics-notes.com/articles/antennas-propagation/phased-array-antennas/beamforming-beamsteering-antenna-basics.php). Online; accessed 16-May-2024. u.a.

- [14] ITU Radiocommunication Sector. *Multipath Propagation And Parameterization Of Its Characteristics*. Tech. rep. International Telecommunication Union, 2021.
- [15] Rohde and Schwarz GmbH. *Operating Manual Vector Network Analyzers*. u.a.
- [16] Rohde and Schwarz GmbH. *ZVB Vector Network Analyzers*. [testequipment.hq.rohde-schwarz.com/datasheets/Rohde-Schwarz-ZVB8-Datasheet.pdf](https://testequipment.hq.rohde-schwarz.com/datasheets/Rohde-Schwarz-ZVB8-Datasheet.pdf). Online; accessed 05-April-2024. 2007.
- [17] Min Shen. "Wireless Phy/Mac Fundamentals". Lecture notes; accessed 3-May-2024. 2014.
- [18] Chen Shun-Ping and Heinz Schmiedel. *RF Antenna Beam Forming*. First edition. Springer Cham, 2023.
- [19] Ankit Singh et al. "Beam steering in antenna". In: *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*. 2017, pp. 1–4.
- [20] Cadence PCB Solutions. *S-Parameters and the Reflection Coefficient*. [resources.pcb.cadence.com/blog/2023-s-parameters-and-the-reflection-coefficient](https://resources.pcb.cadence.com/blog/2023-s-parameters-and-the-reflection-coefficient). Online; accessed 3-May-2024. u.a.
- [21] Johann Tang. *The Choice Between Servo Motors and Stepper Motors*. [blog.orientalmotor.com/the-choice-between-servo-motors-and-stepper-motors](https://blog.orientalmotor.com/the-choice-between-servo-motors-and-stepper-motors). Online; accessed 16-May-2024. 2021.
- [22] John L. Volakis. *Antenna Engineering Handbook*. Fourth edition. McGraw-Hill Professional, 2007.

## Appendix A

### Source Code Link

The repository containing the source code, test results and the report can be found at [github.com/rikkeskov/p5-beam-steering](https://github.com/rikkeskov/p5-beam-steering).

## **Appendix B**

# **Accept Test Result Logs**