UPPSALA
UNIVERSITET

UPTEC F 23057
Examensarbete 30 hp
September 2023

Software Defined VNA

Measure Coupling Between Elements in an Antenna Array

Alexander Söderlund

Civilingenjörsprogrammet i teknisk fysik

UPPSALA
UNIVERSITET

Software Defined VNA

Alexander Söderlund

Abstract

With improvements in hardware of antenna receivers, fully digital arrays have been made possible to use in real time systems. These systems have a greater complexity with the advantage of more dynamic system design. This is due to the ability to tune the system using only software and no mechanical movement.

The dynamic can be used to construct a system capable of bidirectional communication, on a single frequency and at the same time. This is possible with a fully digital array if the coupling between the antenna elements are known. A way of measuring this coupling with the current hardware setup would enable such bidirectional communication. The necessary components to determine the coupling are a signal generator and a coherent multichannel receiver, which is used for sending and receiving data. Using these components a VNA can be created, entirely by software. The VNA can later be used to measures the coupling coefficients of the antenna elements live when the system is in use.

In this thesis the working principle of the the software defined VNA is proven, a testing environment for an antenna array is described and adjustments for real hardware are used to create a result which can be strongly correlated to measurements of a lab grade VNA used as reference.

Teknisk-naturvetenskapliga fakulteten

Uppsala universitet, Utgivningsort Uppsala/Visby

Handledare: Johan Malmström Ämnesgranskare: Mikael Sternard

Examinator: Tomas Nyberg

Sammanfattning

Utvecklingen inom antenna teknologi sker snabbt och stora krav ställs på effektivitet, data mängd och kostnad. För att följa med i denna utveckling krävs system som har maximal funktionallitet utifrån den hårdvara som används. Med förbättringar inom digital databehandling ökar användningsområdet av antennuppställningar med flera individuella kanaler. Detta möjliggör nya funktioner eftersom mer information om signalerna blir kända.

En sådan ny funktion är att sända med delar av antennuppställningen och ta emot data med den resterande. För att inte den mottagna signalen ska bli överröstad av den skickade måste dess påverkan tas bort. För att detta ska vara möjligt måste kopplingen mellan alla antennelement vara känd. Det fina med det hela är att i en fullt digital antennmottagare finns alla de komponenter som krävs för att mäta denna elektriska kopplingen inom antennuppställningen.

Denna uppsats fokuserar på att bevisa hur det är möjligt att mäta kopplingskoefficienterna mellan antennelement i en antennuppställning med enbart en signalgenerator och en koherent flerkanalig mottagare, vilket finns i en antenn mottagare. En möjlig mätmetod förklaras och mjukvara som mäter koefficienterna diskuteras och optimeras. Detta avslutas i att resultatet jämförs och visas ha god likhet med resultat från labbutrustning speciellt designat för att mäta denna kopplingskoefficient.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

Antennas is one main cornerstone in the modern way of communication, it allows for wireless transmission of data over long distances. The technology has been used for a long time with the first antenna experiment conducted in the 1880s by the German Heinrich Hertz, where he proved that two conductor near each other could transfer energy between each other without needing to be connected [2]. This concept was improved upon for many years as radio transmission got popular around the world in the 20th century.

Designing the antenna elements were for a long time a balancing between direction sensitivity and good gain, one single antenna could be built to be sensitive in one particular direction and have low reception in others. To change the reception, these antenna elements had to physically be rotated or moved, either manually as with a TV antenna on the roof or systematically as with a radar station that continuously move. But even with a fast moving radar are limited to some max speed due to inertia and the strengths of the components. Thus, a solution with a stationary antenna but changeable reception angles had to be implemented in order to have greater control of the system and to communicate in different directions in small time spans.

Using multiple antenna elements in a group are the solution still in use today in most two-way communication systems. The idea is to have an array of small antennas with a set distance from each other (within half a wavelength) and sum them for receiving or send the same signal to each element, this creates a beamform with good reception in front of the antenna with regions with hi and lo gain. To further improve this system the ability to add different weights for each element are implemented electronically, allowing for fast switching between different coefficients. These coefficients are in general complex thus both changing both the amplitude and phase of the signal at each element. Depending on what set of coefficients are used, the direction of good reception can now be moved without the need of physically moving anything.

Not to settle with this, the engineers saw further improvements in the system. One such is the ability to set multiple different weights at one time for receiving antennas, which is possible when the signal of each element is know prior to the summation. To access this information, all the individual elements are sampled, resulting in an array of measurements which specific weight arrays can be multiplied with to get information from the corresponding angles at one time. The amount of calculations and complexity increase with the size of the system and number of interesting directions at one time, but with enough computational power this system should be able to output the same result

as the traditional summation technique. This type of system is referred to as fully digital antenna arrays when the sampling is done at passband frequency. To lower the amount of samples needed a alternate solution exist called zero IF systems, where the signal at each antenna is mixed down to a lower frequency before sampling. This have much less computational requirements because less samples are needed, but only a small subset of frequencies can be detected at the time. The subset are determined by the center frequency for mixing of the RF signal with the local oscillator of the receiver. This zero IF system can also be referred to as multichannel coherent receiver.

With these digital arrays, a new opportunity of two way communication at the same time and using the same frequency is possible if the coupling between the different antenna elements is small enough. Two way communication in this case refers to using a subset of the antenna elements for transmission and using the other for receiving. A working two way communication system using the same frequencies will require multiple systems to work together and this project will try measuring the coupling between the antenna elements that can be used as one part of the bigger communication system.

1.2 Objective

The main objective of this project is to provide software for measuring of the coupling coefficients between antenna elements in a digital antenna array. These coefficients, called S-parameters are traditionally measured in Vector Network Analysers (VNA), which are a standalone lab equipment. However, for live measurements of S-parameters in the field, it is not feasible to have an external device in each transmission systems since this would add to much cost and weight to the systems. This is why a software implemented VNA is needed that can perform the same measurements as the standalone counterpart but with the parts already included in the transmission chain.

1.3 Scope

The software should include all the components for the measurement system. That is to send a test signal using some signal generator, extract the ADC signals from the individual antenna elements and process this data to output the S-parameters. The entire measurement process should ideally be automated and simply be run from one single command.

1.4 Limitations

This project do not have the goal of building a system that is capable of two way communication. The goal is to measure the system rather then doing the beamforming calculations. Some assumptions must still be made to focus on the important factors of the system, below some of the most important limitations and assumptions are listed.

- All the coupled signal is assumed to be from the antenna elements, thus the EMC inside the circuit board is neglected.
- All channels are assumed to be perfectly coherent, in reality they are calibrated but potentially small differences can still be present.
- Computational time will not be an important for the result. In other words, the calculations do not have to be in real time. Recorded data can be processed at a later time. This will ensure the best result but in reality this abundance of time is not valid.
- Only one antenna will transmit because of the physical layout of the experiment. If a transceiver would have been used instead of a pure receiver, this could have been done on all channels and a full s-parameter mapping would be possible. The measurement system should be able to scale without much modification, this can not be proven however.

2 Theory

2.1 Radio receivers

Wireless communication is a complex system with many applications and different subsystems, all must work for information to be transported. One of these subsystems are the receiver, which is often referred to as the antenna when viewing the entire communication system. Upon closer examination the receiver has many components one of which are the antenna. The receiver system can be pure analog and transport for example sound, but it can also output a digital signal using an Analog to Digital Converter (ADC). In most modern systems, digitisation are used for all signal types.

For a basic understanding of the key parts of a receiver, consider the block diagram in Figure 1. This receiver consists of one single antenna element in the multichannel receiver system, the signal from the antenna is filtered and amplified while still at Radio Frequency (RF). Next, the signal is mixed with a local oscillator to lower the frequency of the signal. The mixer can remove the carrier frequency by moving the signal to an Intermediate Frequency (IF) or move the signal to be completely centred at zero, which is referred to as Zero IF. The local oscillator can often be tuned to some frequency of interest.

At this point in the system the analog and digital receivers differs from each other. Analog receivers uses circuits to filter, amplify and demodulate the signal to output sound or other data. While digital receivers sample the signal at IF and do any further processing digitally with a Digital Signal Processor. A digital receiver will have multiple mixing stages in the signal chain, however often these are all in one integrated circuit called RF tuner.

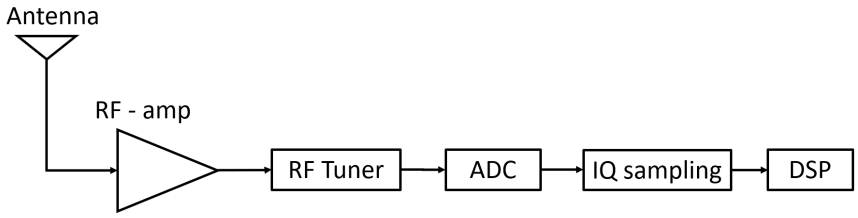


Figure 1 Radio receiver block diagram.

2.1.1 Antenna arrays

As discussed in the introduction, antennas are usually used in a group to have greater control over the transmission and reception. Below the concept of beamforming and key differences between digital and analog arrays are discussed.

With more antenna elements in an array, the ability to change direction of best reception increases. This technique is called beamforming and are achieved through phase offsets on each element. Changing the phase between the antenna elements will result in different locations for constructive and destructive interference, locations with destructive will have low gain and vice versa. In Figure 2, the beams (high gain) are illustrated to show how more elements will change the shape of the beams, by also changing the phase of the individual elements these beams can be directed to other angles.

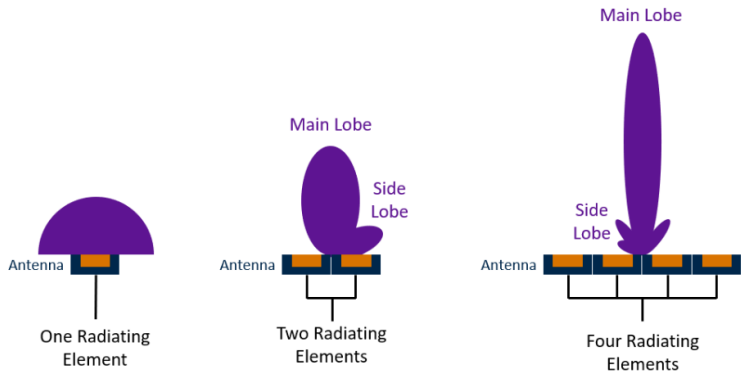


Figure 2 Illustration of beamforming for different numbers of antenna elements. [18]

Electromagnetic waves generated from an antenna will at any point in space be a superposition of all waves present at that location. These waves are generally generated from different antenna elements in an array or even from multiple different antennas. This superposition is what makes beamforming such a useful tool since the signal will add constructively at some locations and destructively at others.

The concept of superposition are easier to understand from the perspective of a transmitting antenna but the same applies to receiving arrays. For the later case the superposition happens as the signals are added together, see Figure 3. If a perfect world without noise were to be considered, a signal would be the same in all elements if the signals origin were in front of the array. If the signal arrives from another direction, instead a time delayed version of the signal would be present at each antenna element. For a narrow band signal, this time delay can be considered as a phase shifts. Using this a system can

multiply with some complex number before all elements are added together. This complex number makes sure that the addition will constructively add signals in the direction of interest and signals from other directions will thus interfere destructively increasing the Signal to Noise Ratio (SNR) of the received signal, a block diagram of this system is shown in Figure 3. If noise is added to this system and are assumed to be white, Gaussian and to be independent for each element, this summation will result in a better SNR compared to the use of only one antenna element, because the noise will cancel itself out as the number of elements are increased.

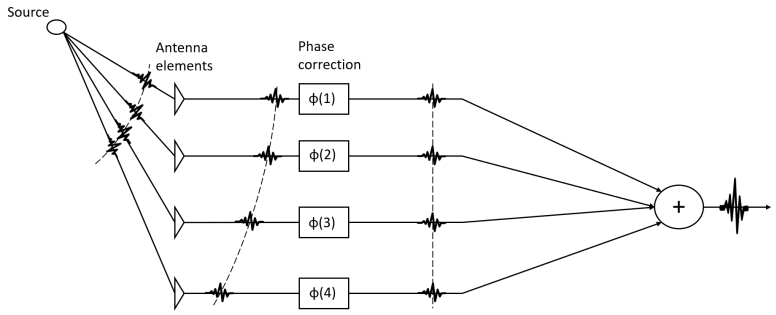


Figure 3 Block diagram of beamforming at receiver level.

2.1.2 Digital arrays

When a signal reaches the antenna array, each antenna element will intercept a unique interpretation of the signal. The different antenna element will have its own signal channel in the receiver system, these channels carry spatial information that can be used to beamform. In Figure 3, an analog array is shown, this refers to the fact that the signal is summed after the phase correction. At the point of summation, the spatial information is lost and the output is indistinguishable from using a single antenna with a specific beamforming geometry. An alternate solution is to use the information channels as long as possible in the information chain. This allows for more flexibility of the antenna system since more information are available.

To use as much of the received signal as possible it should be sampled directly after the antennas as in Figure 4, this method has the name direct sampling. Using the Nyqvist criteria: *The sampling frequency must be at least twice the highest frequency contained in the signal, or information about the signal will be lost* [10], obeying this criteria with will prevent aliasing to occur. For example, an 800 MHz signal need to be sampled with at least 1.6 GHz, this will require advanced ADC circuits and much processing power since the data rate will be proportional to the sampling rate. Furthermore, the data rate will grow with the amounts of antenna elements. With better preforming CPUs and

integrated circuits, this problem can be solved but it will require expensive hardware both for calculations and ADC circuits with high enough sampling speed.

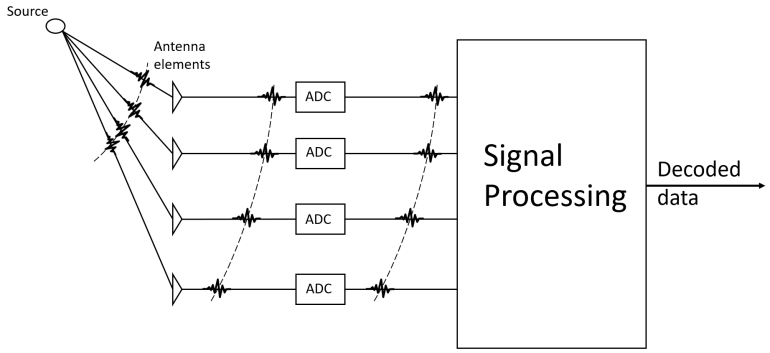


Figure 4 Idea behind direct sampling.

In the direct sampling system, any phase and amplitude corrections are performed in software. If enough processing power is available, the receiver system can be tuned to different directions and frequencies at the same time. This is achieved by taking the incoming data and perform multiple calculations in parallel.

Putting individual radio receivers at each antenna elements and using them with a single clock source, allows for the use of the extra information from the different antenna elements at the same time as sampling frequency of the ADC can be kept low. This solution will have a narrow bandwidth compared to direct sampling method but can use cheaper ADC circuits. The fact that all channels are run with one clock source will make all the channels coherent, which mean that the samples are taken at the same point in time. For beamforming and other methods which involves reception in different directions, the coherence fact is vital to obtain good result.

2.1.3 KrakenSDR

The receiver module used in this thesis project will be an off the shelf product called KrakenSDR. This is a radio receiver with five separate coherent antenna reception channels. KrakenSDR uses a tuner circuit to convert the incoming signals to zero IF, this keeps the complexity of the system and computational requirements relatively low.

The bandwidth of the receiver is 2.4 MHz, with a sampling frequency of the same magnitude. This is possible because the output is given in IQ samples (see section 2.2.2), thus allowing for the use of double the normal bandwidth. The available frequency range of the KrakenSDR are 24 MHz to 1.766 GHz (from a R820T2 chip [9]) which

is standard for software Defined Radio (SDR) applications. The tuner circuit feed the ADC which are a RTL2832U chip with 8-bit resolution.

In Figure 5, a block diagram of one channel in the KrakenSDR are shown. This diagram have the same main components as presented in the basic block diagram in Figure 1 but with more details for this specific receiver. Note the clock are located away from the other components, this is to illustrate that the clock is shared among all channels.

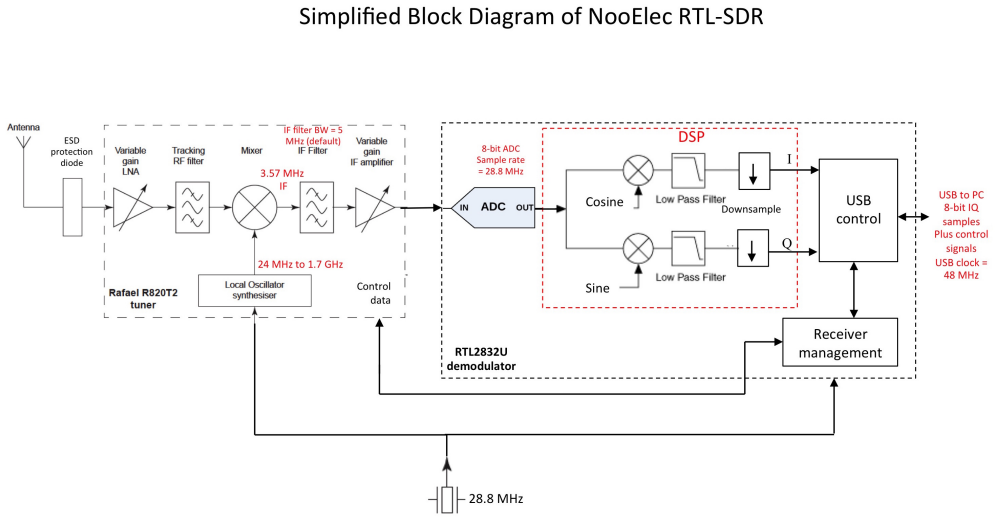


Figure 5 One channel RTL-SDR [14]. KrakenSDR have five parallel all run from the same clock source.

2.2 Complex signals

2.2.1 Fourier Transform

Signals are bounded to be transmitted as time signals, meaning that something is changing over time like voltage in a wire. But what carry information is often how the signals changes over time rather then the specific value, thus an important tool for signal processing is the ability to extract frequencies from a time signal. This is possible with the Fourier transform, which are sets of algorithms to convert signals from time domain to frequency domain. The transform algorithm to be used is determined from whether the signal is continuous or discrete in time and if it is periodic or aperiodic. In this project the signals will be the outputs from ADCs and for the duration of sampling the generated RF signal will be held constant. Thus the discrete and periodic algorithm should be used, that is referred to as **D**iscrete **F**ourier **T**ransform.

This method will take a sample set and what sampling frequency that was used and outputs the relative strengths of different frequency bins. As the length of the time series are increased the number of frequency bins are increased resulting in better resolution of the DFT. The longer time series have a negative effect on computational time as the DFT has a computational complexity of order N^2, meaning an exponential growth in computational time.

The actual solver uses **F**ast **F**ourier **T**ransform (FFT) which recursively solves the problem. This process will be fastest when the number of samples are a power of two, due to the recursive algorithm. Using this method reduces the computational complexity to $Nlog_2N$. Which saves both time and memory, allowing for longer time series to be used. [15]

2.2.2 IQ representation of signals

A normal way to describe signals in RF engineering are IQ-samples, where I denotes in-phase and Q stands for quadrature. When a signal is represented as IQ-samples it will have complex values.

The complex numbers enable data displacement on the complex number plane instead of only on the real axis. This mean that the amount of information per unit energy will be squared. The signals are mapped to the complex plane naturally as it is mixed at the receiver. The phase of the mixer signal will effect what part of the received signal is collected, due to orthogonality the best way of mixing a signal is to mix the same signal with two sine wave phase shifted by 90° (i.e. a sine and a cosine). The output of the two mixed signals will be independent from each other, thus separate information can be extracted from the two outputs. The component mixed with the sine wave will be mapped onto the imaginary axis and the cosine values will be mapped to the real.

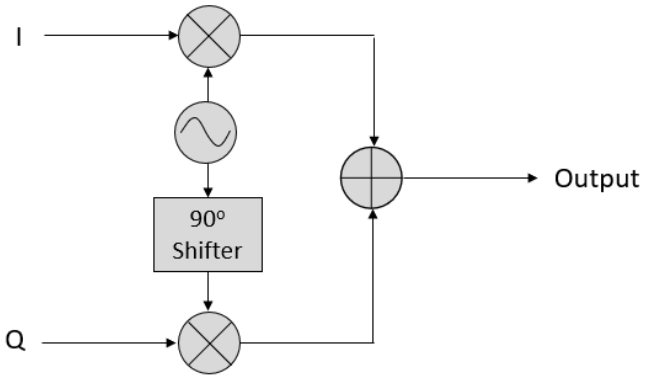


Figure 6 Example of a transmitter which takes IQ samples as input and outputs the RF signal). Source: [8].

More in depth, a complex signal (S) needs to be transmitted over the air, note that this signal will have an asymmetric spectrum due to being complex. In order to transmit signals wireless, the frequency generally needs to be much higher compared to the data rate. This would be on the order of 10 kHz to 10 GHz or more depending on the type of transmitter. To transform the data, signal S to the signal transmitted T it needs to be mixed with the carrier frequency $f_c = 2\pi\omega_c$ used for this system. This is mathematically done by multiplying signal S with the complex exponential of the carrier frequency, see equation 3. In an actual transmitter the real and imaginary values are converted to a analog signal separately before they are multiplied (mixed) and then added together as a real value signal, see Figure 6 for example of a transmitter.

$$S(t) = I(t) + iQ(t) \tag{1}$$

$$e^{i\omega t} = \cos(\omega t) + i\sin(\omega t) \tag{2}$$

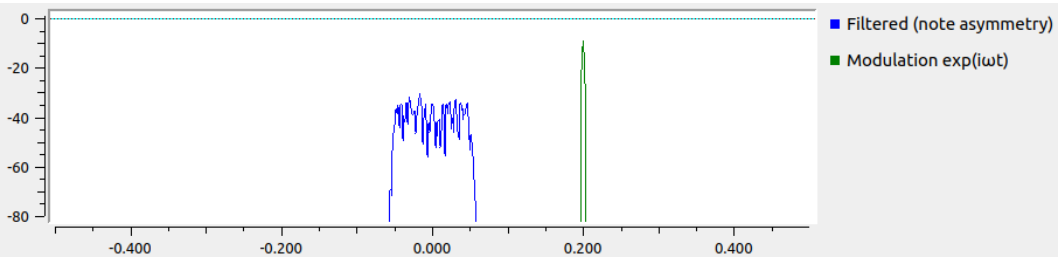


Figure 7 Spectrum example of a data signal (blue) and the carry frequency [$f_c = 0.2MHz$] (green). Y-axis contain the power of the signal in dB and the x-axis show the frequency in MHz. Code to generate figure [4].

$$S(t)e^{i\omega_c t} = (I(t) + iQ(t))(\cos(\omega_c t) + i\sin(\omega_c t)) = \\ [I(t)\cos(\omega_c t) - Q(t)\sin(\omega_c t)] + i[I(t)\sin(\omega_c t) + Q(t)\cos(\omega_c t)] \tag{3}$$

$$Re(S(t)e^{i\omega_c t}) = I(t)\cos(\omega_c t) - Q(t)\sin(\omega_c t) = T(t) \tag{4}$$

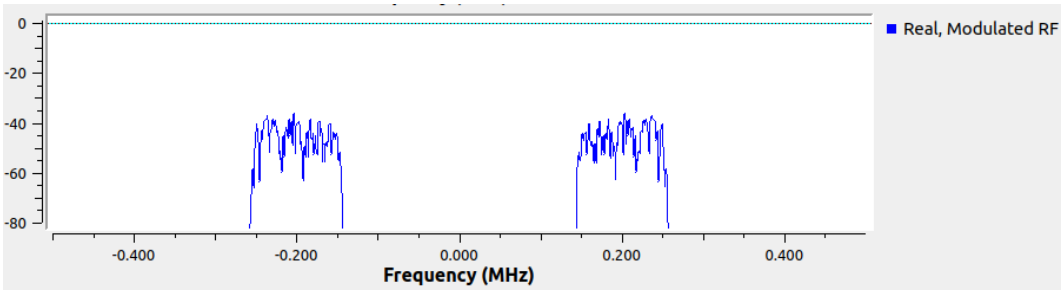


Figure 8 Spectrum example of signal sent from transmitter, note symmetry around zero hertz. Code to generate figure [4].

In Equation (4), the transmission signals components are described. Per definition this signal is completely real and thus will have a symmetric spectrum in the frequency domain centred around zero. Thus a mirror image of the signal will be present at $-f_c$, see Figure 8. At the receiver this signal can not be processed at the RF frequency because of limitations in ADC sampling rates. The solution is to again mix the signal with a cosine of the center frequency f_c. The output will have multiple frequency components due to trigonometric identities, one will be at baseband (zero hertz) and the other will be at twice the center frequency ($2f_c$). These two components will be identical but located at different locations of the spectrum, thus they contain the same information and the high frequency component can be filtered out using a low pass filter. The result of this will be a spectrum symmetric around zero which will contain half of the information in the original source. The other half will be generated by mixing the received signal with $90°$ phase offset (i.e. a sine) of same frequency. The mixing with a sine wave will follow the exact same steps to obtain a symmetric spectrum around zero containing the other half of the information, see Figure 9. Since both the sine and cosine spectrums are symmetrical they can be combined into one plot by simply discarding the half of the spectrum for each of the signals and assign the negative frequencies to the sine spectrum and the positive to the cosine, this is shown in Figure 10.

The mathematical expression of what happens at the receiver with the signal T is shown below in Equation (5) to (9). In the first two equations, the hard brackets, [] indicate what happens in the transmitter and the rest is carried out at the receiver.

For the cosine term:

$$\begin{aligned} T(t)\cos(\omega_c t) &= [I(t)\cos(\omega_c t) - Q(t)\sin(\omega_c t)]\cos(\omega_c t) \\ &= I(t)\cos^2(\omega_c t) + Q(t)\sin(\omega_c t)\cos(\omega_c t) \end{aligned} \tag{5}$$

For the sine terms:

$$\begin{aligned} T(t)\sin(\omega_c t) &= [I(t)\cos(\omega_c t) - Q(t)\sin(\omega_c t)]\sin(\omega_c t) \\ &= I(t)\cos(\omega_c t)\sin(\omega_c t) + Q(t)\sin^2(\omega_c t) \end{aligned} \tag{6}$$

Some trigonometric identities used to compute the received signal. [11]

$$\begin{aligned} \cos(x)^2 &= \frac{1}{2} + \frac{1}{2}\cos(2x) \\ \sin(x)^2 &= \frac{1}{2} - \frac{1}{2}\cos(2x) \\ \cos(x)\sin(x) &= \frac{1}{2}\sin(2x) \end{aligned} \tag{7}$$

Using the identities in Equation (7), the received signal can be described as in Equation (8).

$$\begin{aligned} T(t)\cos(\omega_c t) &= I(t)(\frac{1}{2} + \frac{1}{2}\cos(2x)) + Q(t)\frac{1}{2}\sin(2\omega_c t) \\ T(t)\sin(\omega_c t) &= I(t)\frac{1}{2}\sin(2\omega_c t) + Q(t)(\frac{1}{2} - \frac{1}{2}\cos(2x)) \end{aligned} \tag{8}$$

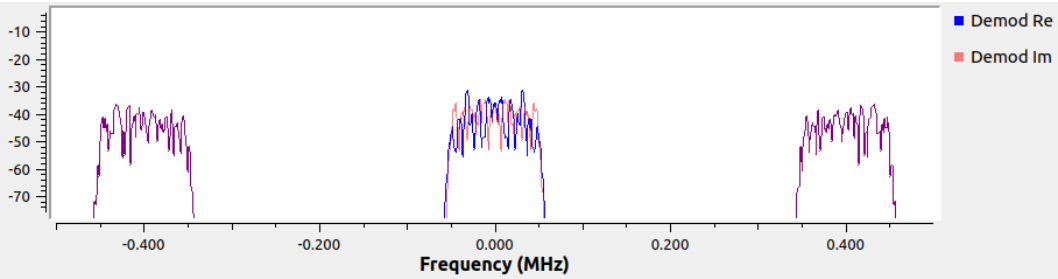


Figure 9 Spectrum example of signal demodulated at receiver, cosine term generates blue spectrum and sine generates the red. Note both spectrums are symmetric around zero hertz i.e. real value signal. Code to generate figure [4].

The received signal can be low passed filtered to remove all terms with frequency components above ω_c, this yields:

$$\begin{aligned} T(t)\cos(\omega_c t) &= I(t)\frac{1}{2} \\ T(t)\sin(\omega_c t) &= Q(t)(\frac{1}{2}) \end{aligned} \tag{9}$$

The original complex signal can now be regained digitally by assigning I to be the real part and Q to be the imaginary part of the signal, this signal will be a copy of S but with a different amplitude due to losses and amplification along the system.

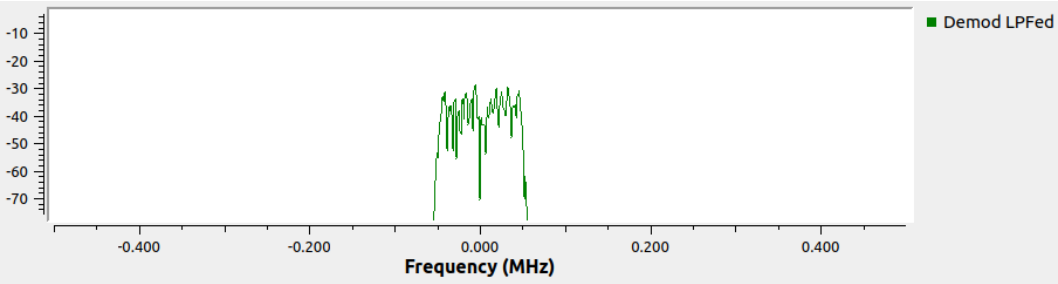


Figure 10 Spectrum example of the signal as viewed from the receiver. The spectrum generated from a complex signal and not symmetric around zero. Code to generate figure [4].

2.2.3 Example of IQ system

To prove that the experimental setup used for this thesis project works as intended, a prototype program was written in GNU radio. This program generates an RF signal with a I and Q term, this is achieved by adding a sine with amplitude I and a cosine with amplitude Q.

This signal is then split and multiplied with a sine and cosine of the same frequency as used before. This output is feed to a lowpass filter to remove the 2ω component of the signals. What the output will read will thus be a DC voltage of the same level as the original I and Q values, this is possible with only frequency knowledge on the receiver side (after the signals were added). The program is written in a way that the I and Q component can be altered live to see that the DC voltage follows the changes perfectly. In figure 11 a screenshot of the program is provided.

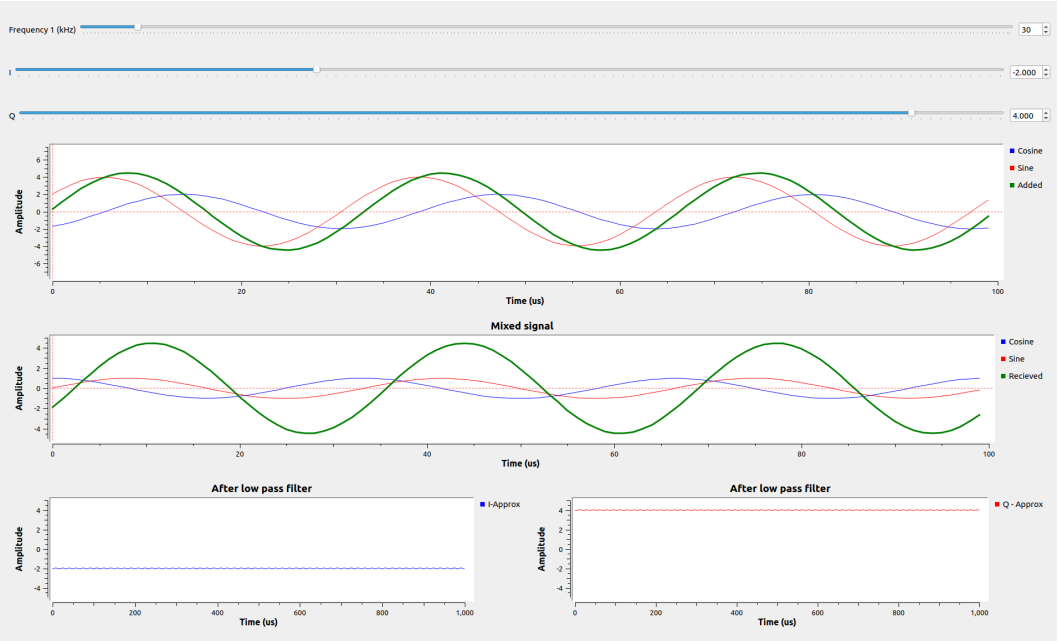


Figure 11 Example of the orthogonality of sine and cosine. Code to generate figure [13]. SDRSineCosineIQ.grc

2.3 Characterise RF networks

Radio frequency components have different characteristics depending on the frequency they operate at. For this reason it will be necessary to find coefficients that describe how a signal propagates in the component. These coefficients must also be determined separately for the different frequencies the component will be used with.

When creating an electrical system, many different components will be used together. Thus, it would be ineffective to determine the characteristics of each component separately. An easier approach is to group components together to form a subsystem. This subsystem can be considered as one component when determining the characteristics, which makes the main system easier to understand. In RF engineering the word network is used for characteristics, a network can consist of a single component or a subgroup of a greater system. In this section, the main principles of characterising an RF network are explained.

2.3.1 Reflections

When working with RF system, the formulas used in DC electronic computations will not be valid, or need to be changed to handle the AC components. The effects of capacitance and inductance becomes much more important and it will have big impact on the characteristics of the system. To get to grips with what is special with AC calculations or rather RF (radio Frequency) calculations below some of the most important concepts presented.

All electronic components have different characterisations depending on how they are built, different materials and geometry result in different characteristics. When a transmission line is constructed for RF signals it will be built with a characteristic impedance (z_0) of somewhere between $50\ \Omega$ and $70\ \Omega$, this characteristic is a ratio between the voltage and current in the transmission line.

- R - Resistance per length
- L - Inductance per length
- G - conductance per length
- C - capacitance per length

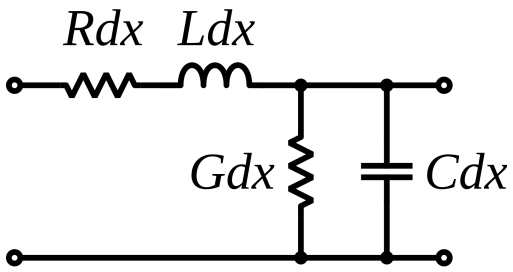


Figure 12 Parameters that determines the characteristic impedance of a transmission line. Source: [5]

Using Figure 12 and telegrapher's equation, the characteristic impedance can be solved for.

$$\frac{V_+}{I_+} = z_0 = \sqrt{\frac{R + j\omega L}{G + j\omega C}} \tag{10}$$

If the transmission line can be idealised as lossless, the real parts will be zero and the equation can be written as.

$$\frac{V_+}{I_+} = z_0 = \sqrt{\frac{L}{C}} \tag{11}$$

When doing calculations on transmission lines and for RF overall it is important to note in which direction waves are travelling. This is why the notation for voltage and current in Equation (11) have a plus subscript, they travel from left to right. The same could have been calculated in the other direction but then the notation should be V_- and I_-. The reason why waves can travel in both direction is because of reflections which occurs as the impedance changes along the signal path. This reflection Γ will depend on the difference in impedance between the load (z_L) and characteristic impedance (z_0), see Equation (12). The value of Γ will be directly related to how much power is lost due to reflection, Equation (13). [3]

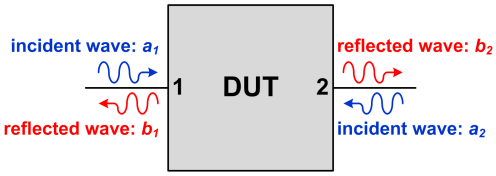
$$\Gamma = \frac{z_L - z_0}{z_L + z_0} \tag{12}$$

$$RL[dB] = -20log_{10}|\Gamma| \tag{13}$$

Since the impedance are complex, the reflection coefficient Γ will be complex. Converting this complex number into polar coordinates results in magnitude and phase which will correspond to how the reflected wave behaves compared to the incident. As an example, if Γ would be -1 (completely real) the reflected wave would be an exact copy of the incident wave but with a $180°$ offset, -1 in polar coordinates are $1\angle 180°$.

2.3.2 S-parameters

As stated previously, RF components will have different behaviour depending on the stimulus frequency. Often the most important factors needed for characterisation are amplitude difference and phase shift of the signal. These two factors can be read directly from scattering parameters (S-parameters) that can be calculated or measured from a system. As the name implies S-parameters are ratios of the signal as it scatters or splits in the system. Any devices can be characterised with S-parameters like cables, antennas, junctions and many more. The type of device have no effect of how the S-parameters are measured, thus components or networks are often considered as Device Under Test (DUT) as in Figure 13.



(a) Illustration. Source: [16]

$$S = \begin{bmatrix} S_{11} & S_{21} \\ S_{12} & S_{22} \end{bmatrix}$$

(b) Matrix representation

Figure 13 Example of a two port network. The coefficients a_i and b_i are used to compute the S-parameters. DUT = Device Under Test

S-parameters are represented using a square matrix with as many columns as ports in the network. The columns in this matrix represent output power while the rows represents input power. Element S_{ij} will be the power at port i given that j is sending. Like the reflection coefficient, the S-parameters will be complex and contains both information about magnitude and phase at the output.

On the diagonal in a S-matrix, the parameters represent reflection coefficient for the specified port, i.e. $S_{11} = \Gamma$ at Port 1. While the other S-parameters contain information of transmitted signals. Most passive components have a characteristic that is called reciprocity which is that the direction of the signal do not matter. An example of reciprocity would be, a signal sent from Port 2 and measured at Port 1, would have an S-parameter identical to if the signal was sent from Port 1 and measured at Port 2. In more mathematical sense, the matrix will be symmetrical around the diagonal or $S_{ij} = S_{ji}$. [12]

The S-matrix will characterise important aspects of a system, that should be clear. However as previously stated the characteristics of the system will change depending of the frequency input. Since the s-parameter should have info about the phase difference between output and input and if a cable are considered. The phase of one end related to the other will change drastically depending on the wavelength. To be able to use the approximated s-parameter for calibration the frequency used for generation of the s-parameter should be close to the frequency used later. For full characterisation of a system this mean that a range of different frequencies must be approximated to latter be able to interpolate the phase and magnitude at the specific frequency used in the system. Otherwise a accurate phase measurement can not be accomplished, see how s-parameters can change in figure 14.

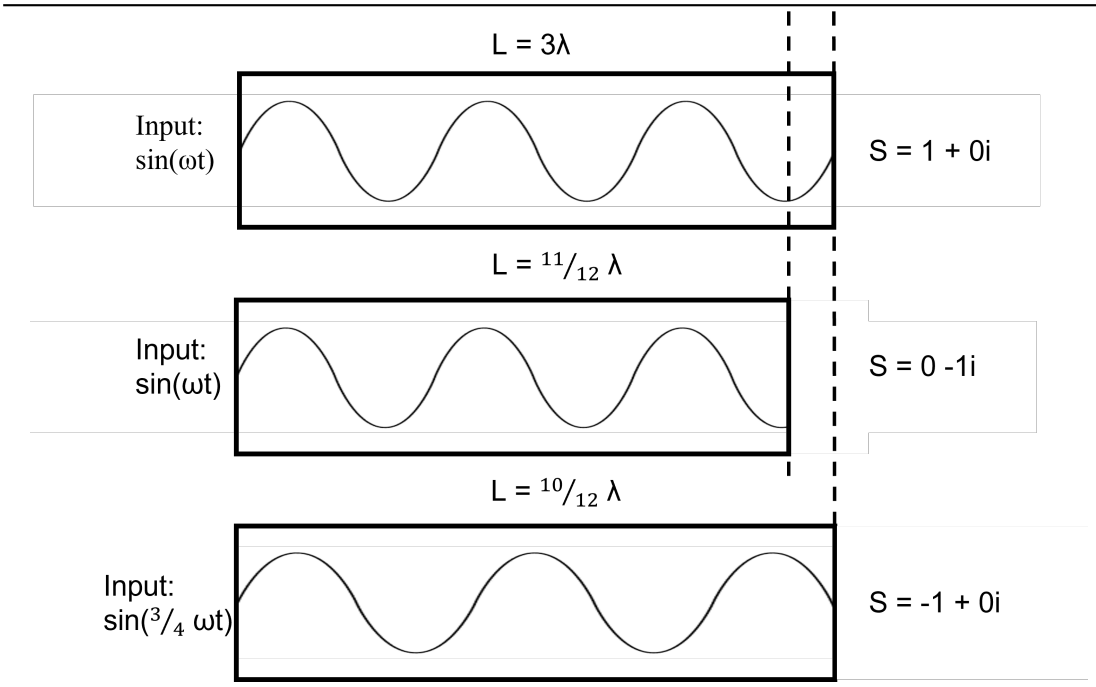


Figure 14 Example of how s-parameters change depending on signals and hardware.

2.3.3 Vector Network Analyser

With a clear definition of S-parameters, the question of how to measure them remains. This is done with a component called network analyser, they exist in two versions, Scalar (SNA) and Vector Network Analyser (VNA). A SNA can only measure amplitudes while the VNA can fully characterise a system thus returning both phase and amplitude data. Recreation of the latter will be of main focus in this report.

The S-parameters can not directly be measured in a system but instead calculated using other quantities. The parameters that will need for this thesis project are a_i and b_i in Figure 13a, which are transmitted and output waves to and from the DUT. In order for a VNA to measure these quantities it must contain some receiver to interpreter the signals as they are in RF domain and a signal generator to create the test or stimulus signal. A schematic of a two port VNA can be seen in Figure 15. Where a the signal generator are located at the top. The generator feeds a signal through a switch to send the power towards port one or two, never both at the same time. After the switch a splitter divides divides the signal into two parts. The division can be even $(50/50)$ or directional (example -20dB) to send more power to the DUT, this will not effect the results as long as any attenuation of the signal are properly calibrated for, done in software. One part of the split signal goes directly to a sampler to be used as reference. The other passes

through a directional coupler and into the DUT. A directional coupler is a component that are sensitive to the direction of the waves sent through them.

At the DUT the signal will again be split, some of the signal will be reflected and returned to the coupler and the other will propagate in the device and exits at other ports (in Figure 15 the input will be P1 and output is P2). All signals exiting the system (reflected and transmitted) will encounter a directional coupler, which is setup to pick signals travelling away from the DUT. Both directional couplers will be feed to sampling units to be measured. With these three information sets the processor can determine the ratio of reflected and transmitted power to the input power and from this generate the S-parameters. Then the main switch change the source to feed the other side of the DUT and preform the same measurement, the result will be the same if and only if the DUT are reciprocity.

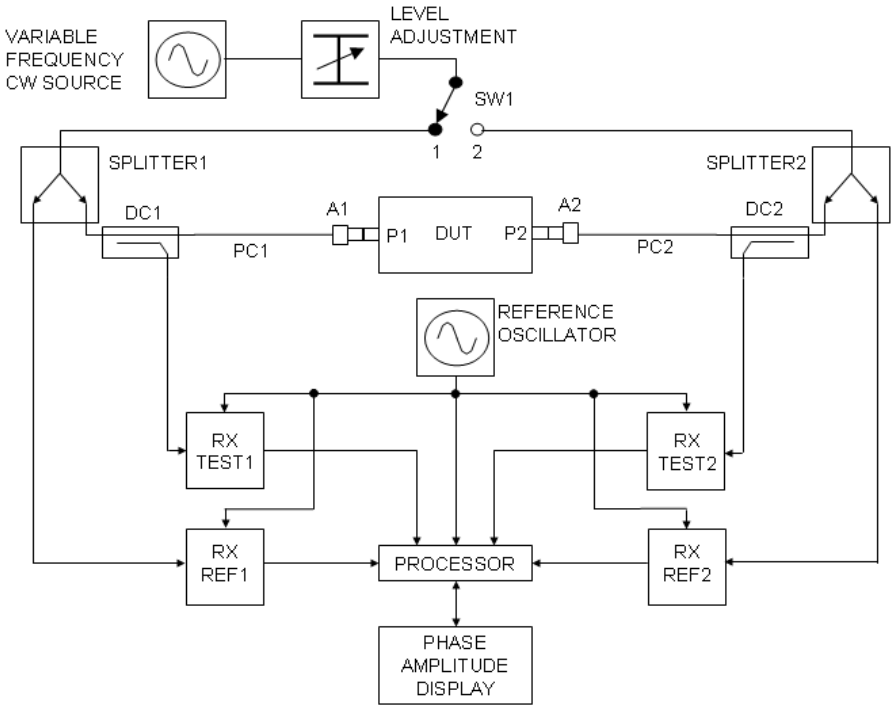


Figure 15 2 port VNA. Source: [17].

2.4 Free Space Path Loss

A signal sent between two antennas can take multiple different paths to reach the receiving antenna. One such path is Line Of Sight (LOS) where the signal directly can travel

through the air and be picked up at the receiver. Besides the LOS component it is also possible for the signal to reflect on objects and reach the receiver by an indirect path, but this will often result in lower signal strength compare to the direct path. For signals interacting between elements in an antenna array the distance between the antenna elements can be considered small enough to discard any contributions from components other than LOS. If this assumption is used a approximation of received signal strength can be calculated theoretically to compare to later measured values.

For a signal with purely a LOS path, the formula for receiver gain is called Free-space path loss, see (14). In this equation G denotes gain of the receiving and sending antenna respectively, when using unidirectional antennas the gain will be one. [7]

$$\frac{P_r}{P_t} = G_T G_R \left(\frac{\lambda}{4\pi d} \right)^2 \tag{14}$$

3 Method

3.1 Measurement idea

Using an antenna receiver with multiple coherent (time synced) channels and a signal generator, it should be possible to preform measurements of how a signal changes between the channels in the receiver. Changes with respect of magnitude and phase shift are of interest as well as these being dependant on the frequency of the signal. Connecting antennas to each receiver channel and also connecting one of the antennas to the signal generator allows for the receiver to have information of both sent and received signal at each antenna element. Thus evaluation of phase (or time) shift and attenuation between the antenna elements is possible.

More explicitly the signal generator would output some waveform, like a pure sine wave of a given frequency. This signal would be sent to the antenna and one of the receiver channels, from the antenna the signal would be transmitted to a different antenna element nearby and that signal would be collected using another receiver channel. Preforming some signal processing method to these signals should allow the signals to be shifted to frequency domain, where by the sine-in sine-out principle, information about the signal can be extracted from just the bin corresponding of the generators frequency setting.

This component should be complex in order to contain information both about magnitude and phase of the signal. Extracting the specific frequency component for both channels and comparing them with the transmitted signal, will show how much less energy was collected at the receiving antennas and the relative phase.

3.1.1 The data

When running KrakenSDR with the lab setup packets will be collected in python using ethernet sockets. This does not change the way Kraken is run on the computer, a memory buffer will still contain the data from each individual channel as it sent over the USB connection from Kraken to the computer. If these buffers are not emptied fast enough samples will start to be dropped and thus synchronisation is lost. This should be detected by the Kraken receiver software and synchronisation procedure started.

To empty the buffers and send the data from memory to python ethernet is used. The data are sorted into packets which contain a 5×2^{20} matrix where the rows are the five receiver channels and the columns are respective sample. This packet will also

contain header information from the python file ”iq_header.py”, this header will have information such as center frequency, gain settings, synchronisation flag (true or false) and also timestamps. The time for each sample can not be determined due to the fact that the data stream has no timestamps, only samples are being transferred from Kraken. Thus the timestamps of the packets are from the moment the packet was created not from when the samples were taken.

In python the data can be collected using the provided class “ReceiverRTLSDR”, it has functions for reading packets and change center frequency of Kraken. By running the function to collect a packet, a numpy array of the bits are received. From the header information it can be determined if the packet was good or if Kraken was in calibration mode when the packet was sent and thus the entire package can be discarded. To confirm that everything is working as expected the spectrum of the package can be viewed. By first tuning Kraken to a frequency where data should be present, like TV broadcast, FM radio or something else of known frequency and collecting one packet of data. This packet will include about one million samples and to convert the signal to frequency domain a good method is to use the Fast Fourier Transform.

If the signal looks something like in figure 16 the package was collected with success.

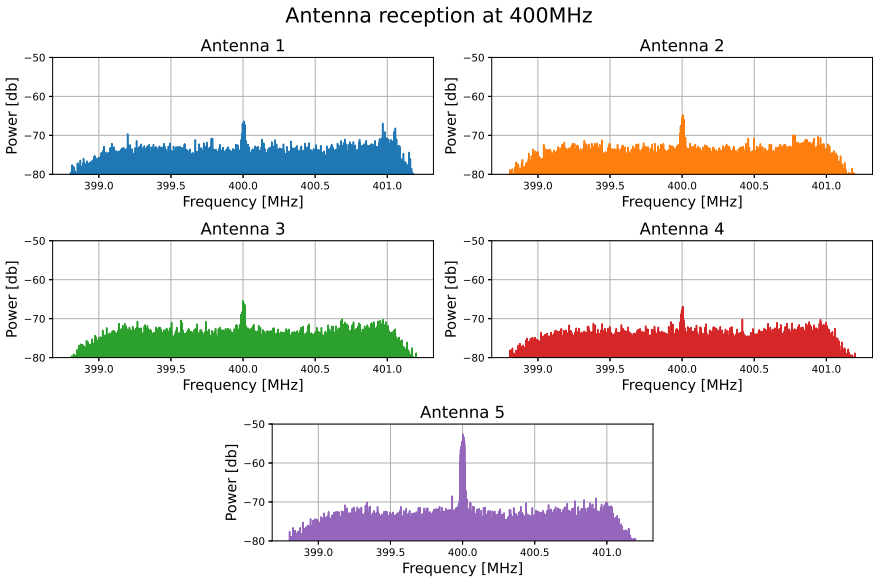


Figure 16 Example of a random packet received from Kraken. Verifies that packets are collected and that some data are present.

3.2 Signal generator

Because KrakenSDR only works as a receiver, a separate unit must be used to output the test signals. This will be solved by using a signal generator from AnaPico and the exact unit is APSYN140. This generator has a frequency range from 10 MHz to 40 GHz which is plenty enough since Kraken has a range between 24 MHz to 1.8 GHz.

The APSYN140 generator as a RF output port of type SMA female and this will be the only port used on the unit, except for an ethernet connection to the computer which will run all code. Due to this ethernet connection the generator can be controlled using python sockets and simple text commands. The main commands used can be seen in Table 1 and for full descriptions read the programmer's Manual — Signal Generators & Frequency Synthesizers at [1].

Table 1: Useful commands for APSYN140 Signal Generator.

String	Description
:SOUR:FREQ:CW 100 \n	Set frequency to 100MHz
:SOUR:POW:LEV:IMM:AMPL -10 \n	Set Power level to -10 dBm
:OUTP:STATE ON \n	Turn on Generator
:OUTP:STATE OFF \n	Turn off generator

3.2.1 Connect APSYN140

To first get the generator to work with a computer can be tricky. Connecting it with ethernet is the best easiest option but this will disable the use of wi-fi during lab testing. After connecting the computer and signal generator with a cable and turn on the generator it will try to communicate with the computer using a predefined IP address of "169.254.8.211". If the network settings of the computer is not set to Link-Local Only this will not be possible and the generator can not be found. If no changes to the computer was done previously the setting is most probably on Automatic (DHPC) which mean that the computer will try to assign the generator a IP-address.

When the correct IP address has been established the web browser interface apuasyn20-0117.local could be accessed from which a nice user interface can be used to control the generator. At this stage a simple test to check that everything works as expected was carried out by pressing *RF ON* and a LED on the generator should illuminate. If the generator are connected to an antenna at this stage, a signal should be picked up by the receiver, which if transformed to frequency spectra will look something like in Figure 17. Before using KrakenSDR at the same time as the generator it is important

to change output power of the generator power using the commands from Table 1, the power should be set to -10dBm (minimum power level to protect KrakenSDR). Both KrakenSDR and the generator can now be tuned to the same frequencies without overloading the ADC chips in the receiver.

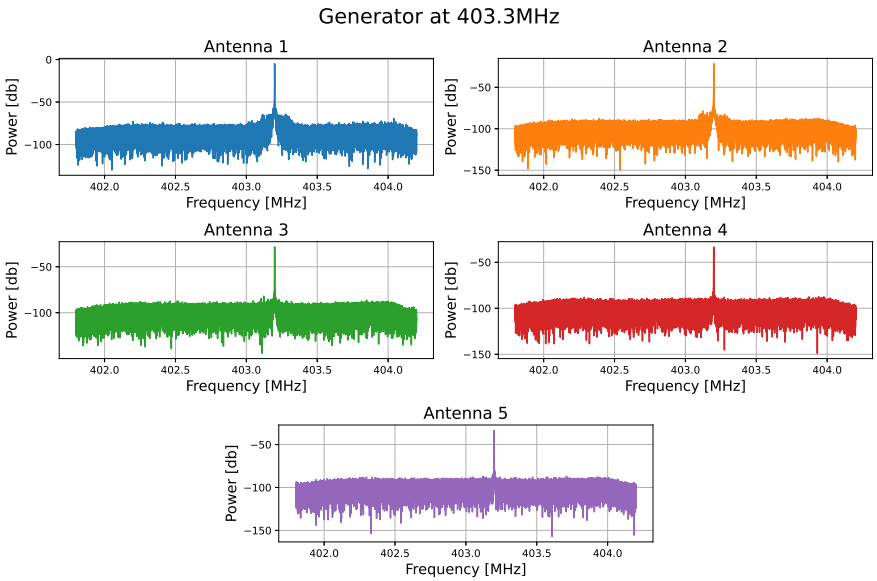


Figure 17 Example of signals from the generator picked up by the receiver, this verifies that packets are collected by the receiver and that the generator works.

3.2.2 Offset

The receiver and generator will not be running from the same clock source since they are two different systems. This will be a problem when exact frequency components needs to be extracted since the difference in referance time will mean that if one system outputs a frequency another might interpret that oscillation as slower or faster depending on the tolerance and error in one or both of the clock sources. This is clearly visible in Figure 22 (most noticeable on the right graphs) where the zero frequency of the receiver do not overlap with the output from the generator.

This will be a problem since the measurement idea is to take the FFT value at the peak value of each channel to know the signal strength at each antenna. But if the generator and receiver are not synced it can not be predetermined which frequency bin to use. The fact that this offset is present are in itself not a problem because the only system doing the measurements are the receiver, thus can the frequency of this component be seen as absolute or correct. For the purposes of the project the deviations in frequency

from the clocks will have negligible effect on the result because the s-parameters can be approximated as constant for the little shift present. If this would not be true a characterisation of the frequency offset of the receiver has to be determined using some calibration coefficients and a known reference clock with good enough tolerances.

However the important factor is that all channels are coherent. And since they all run from the same clock source this is true. The fact that the generator will output something with an offset will not really matter to the calculations since the output is also measured by the receiver at one of the channels. Using this channel it can be determined the relative frequency of the incoming signal by finding the index corresponding the the maximum FFT value. Comparing the bin with this index for all channels will give the best **S**ignal to **N**oise **R**atio. Practically the best way to find the interesting index is to use the *numpy.argmax()* function in python.

In figure 22 two different sizes were considered. Comparing them results in a understanding that the generator frequency do not fully align with the internal frequency of the receiver. This will be a problem since. For any sizes of FFT this can be solved using the *numpy.argmax()* function in python to find the frequency bin with strongest signal. Considering the bigger FFT this will be crucial to achieve good results.

3.3 Lab setup

A good understanding of the setup of the measurements are important to find potential errors and to get reproducible results. A schematic of the main lab setup used in this report are presented in 18. A laptop is used to run Python as the main controller for both KrakenSDR and the generator. The antennas are placed with with a set distance of 8 cm from each other to keep the setup constant during the entire thesis project. In Figure 19 the physical setup is also shown.

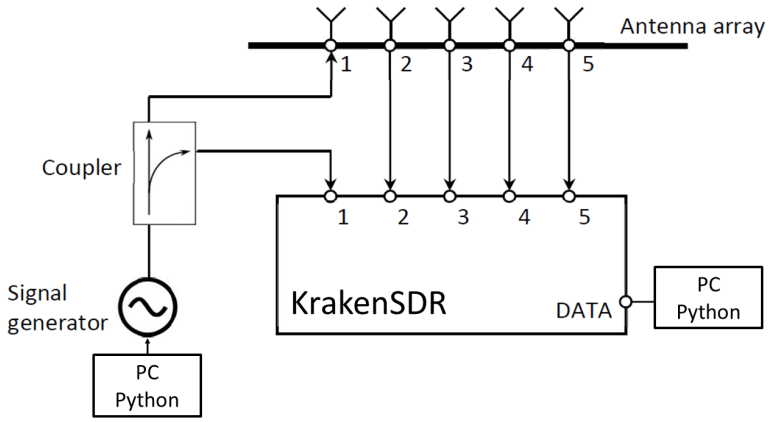


Figure 18 Schematic of the lab setup.

Figure 19 Picture of the antenna elements.

3.3.1 Calibrating the system

When the concept of coherent antennas are used it is important that the system are calibrated because small differences in cable length will result wrong measurements, mainly with respect to phase but amplitude can also be effected. The important factor to remember is that KrakenSDR only sees the world as it is at the five antenna ports, the signal that is sampled is a snapshot of the voltage at this point at each timestamp.

This signal is really not interesting for most user cases since the signal of interest is the reading at the antenna. In radio frequency measurements the idea of ideal transmission lines fall apart because everything responds differently at different frequencies thus to compensate for something like a transmission line it must be characterised at different frequencies to get a table of transfer functions or S-parameters. This is a one time measurement that should be done with a calibrated VNA to later compensate the signal.

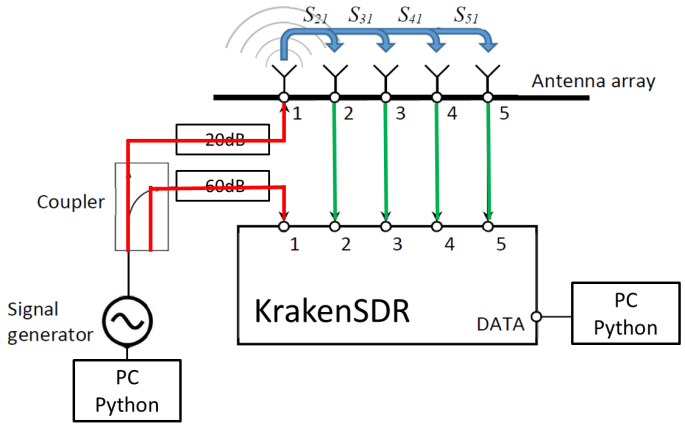


Figure 20 Lab setup with signal paths. Green represents provided KrakenSDR cables and red represents the paths that has to be calibrated.

If all channels between the antennas and KrakenSDR are identical, the measurements will be synced automatically but this is not always the case. When using cables of different length, the phase of the incoming signal will be different depending on the frequency of use. Thus a calibration coefficient for the phase must be introduced. This coefficient will be a complex number or a transfer function and will describe the relationship between the antennas and Kraken. This transfer function are provided for the cables that came with Kraken but for the reference signal the problem become more complex since the signal originates at the signal generator, measured at Kraken and the signal at the antenna is the wanted signal. To compute this transfer function it is not possible to measure the S-parameter between Kraken and the Antenna because the coupler isolates the signals thus this would result in high attenuation which is not present in the real system. The solution will instead be to calculate the transfer function between Kraken and the generator. Then combining this with the transfer function between the generator and the antenna. For this to be possible a characterisation of all components used is necessary.

A cable will have four S-parameters as in (15), when preforming the measurement on a VNA all four will be characterised. Due to reciprocity of a cable, only two components will be unique ($S_{21} = S_{12}$ and $S_{11} = S_{22}$).

$$S = \begin{bmatrix} S_{11} & S_{21} \\ S_{12} & S_{22} \end{bmatrix} \tag{15}$$

The coupler will have the following characterisation matrix:

$$S = \begin{bmatrix} S_{11} & S_{21} & S_{31} \\ S_{12} & S_{22} & S_{32} \\ S_{13} & S_{23} & S_{33} \end{bmatrix} \tag{16}$$

There are two interesting parameters to keep track of, the coupling between the signal generator (Port 1) and antenna (Port 2), this will be S_{21}. Also the coupling between KrakenSDR (Port 3) and the generator (Port 1), i.e. S_{31}. Since Port 2 and 3 can be considered isolated, because of the characteristics of the design, S_{21} and S_{31} can be used independently from the entire S-matrix. Thus, resulting in a way of knowing the signal at Port 2 from measurements at Port 3. The signal at Port 3 (V_k voltage at KrakenSDR) can be written as a function of the voltage at Port 1 (V_G voltage at Generator):

$$V_K = S_{C,31} S_{Blue,21} V_G \tag{17}$$

By rewriting this expression, the generator signal (V_G) is obtained:

$$V_G = S_{C,31}^{-1} S_{Blue,21}^{-1} V_K \tag{18}$$

The antenna voltage (V_A) can be written as a function from the generator voltage.

$$V_A = S_{C,21} S_{Black,21} V_G \tag{19}$$

Writing these as a function of each other and expression becomes:

$$V_A = S_{C,21} S_{Black,21} S_{C,31}^{-1} S_{Blue,21}^{-1} V_K \tag{20}$$

Or written as a transfer function $H(f)$, note the addition of a frequency dependence. This was always the case but when working with S-parameters the frequency dependence is often not expressed. However for transfer functions it is usually displayed.

$$H(f) = \frac{V_A}{V_K} = S_{C,21}(f) \cdot S_{Black,21}(f) \cdot S_{C,31}^{-1}(f) \cdot S_{Blue,21}^{-1}(f) \tag{21}$$

3.3.2 Apply calibration coefficients

Installation of the KrakenSDR software provides data for the cables included with the KrakenSDR unit. This will be 5 separate files for each receiver channel. However the file can be changed to and replaced with other values if another cables are used. For more information on how to change the settings see Appendix B.1.

Every time Kraken changes center frequency, the software will look at the frequencies in the touchstone files and apply the S-parameters at the closest possible frequency. It is important that the frequency list must be same for all channels. The original calibration data have 101 different frequencies, ranging from 10 MHz to 2 GHz with 20 MHz steps. When measurements were done on the new cables and coupler the output data was generated with 201 points between 10 MHz and 3 GHz. These measurements are shown in Appendix A Figure 39 to 42. Since the frequencies had to match the the number of points had to be reduced, for this interpolation can be used. However, a way of interpolating two complex number in not fully mathematically defined thus creates a problem. The angle *or* amplitude can be interpolated but choosing one will compromise the other. Of most importance in this case will be the angle of the output since this will reflect the phase shift thus that component is often used for the interpolation. In figure 42 the difference between interpolation results of angle and magnitude is clearly shown.

Using (21) and the S-parameters from the VNA, the transfer function can be comprised to a 1d array of complex numbers. This is displayed in Figure 21.

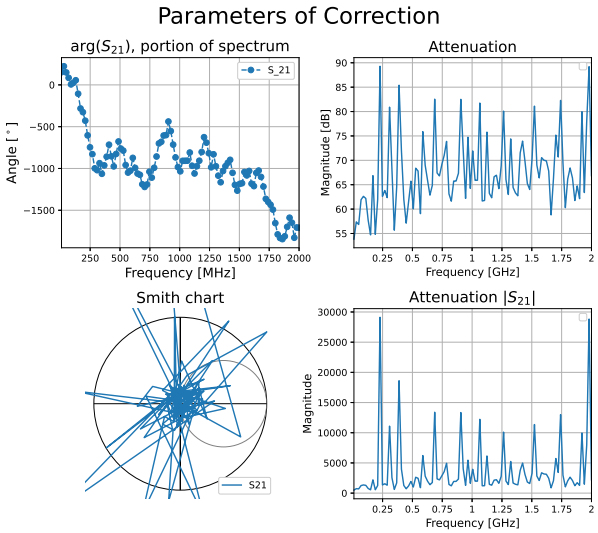


Figure 21 Diagrams over the final compensation factors used in the system.

3.4 Measurements

When preforming the experiments ADC samples will be collected and labelled with the generator frequency used for the respective packet. It is also important to know what center frequency Kraken was set to since that information is lost in the mixing to zero IF conversion. The way data is stored are Numpy arrays stored as .npy files. This is a way to store numpy files at disc memory instead of ram memory, thus preventing the ram memory to overflow during data accusation. This method is a good solution to have zero data loss to be able to fully examine the data afterwards.

If good settings for FFT conversion are obtained it is a good idea to run a FFT live and store the output data from this method instead. Thus reducing the amount of data given that the length of the FFT is less than the package size. The output from FFT will still be numpy arrays and can be stored in the same way as for IQ samples.

The output from FFT will be a complex array representing the frequency components of the recorded signal. Finding the frequency that the generator operated at should be simple in a low noise environment simply by finding the maximum amplitude around the frequency bin of the known generator. Since the generator and Kraken uses separate clocks it would be good practice to assume that the frequencies do not match one to one. Having a code that find a maximum value on port 0 and uses that frequency bin for comparison with the other channels, this will hopefully generate stable result.

3.4.1 FFT size

An important part of having a working RF system is to be able to handle all incoming data. The bottle neck in a system is often the initial computations since this is where the amount of data is reduced drastically. For the applications of this project the main computational tool will be the use of FFT algorithm to extract frequency components of the signal. The frequency resolution of the output will have a one to one ratio to the input values. If the input is 1024 samples the output frequencies will have 1024 unique bins. The idea is that if more of the signal is included the more details about it can be extracted. This has to be considered along with the fact that the algorithm will be more complex as size increases, thus resulting in longer computational times. To find good balance between precision and speed different FFT sizes was considered and compared to each other, note that the size of a FFT should always be a power of two for optimal speed.

An initial conclusion of comparing different sizes for the FFT was that the maximum values of each bin was correlated to the sample size. This was due to how the FFT

algorithm works, if the output is not normalised it will scale with the number of input samples. By dividing each bin with the sample size this effect is removed and all values will be scaled between zero and one. In figure 22 two different sizes were considered. Comparing them results in a understanding that the size of the FFT do not really matter since the relative size of each channel are about the same. For the phase however the bigger FFT looks to have more stable phase, this could also be an artefact from the smaller plot window for the bigger FFT signal.

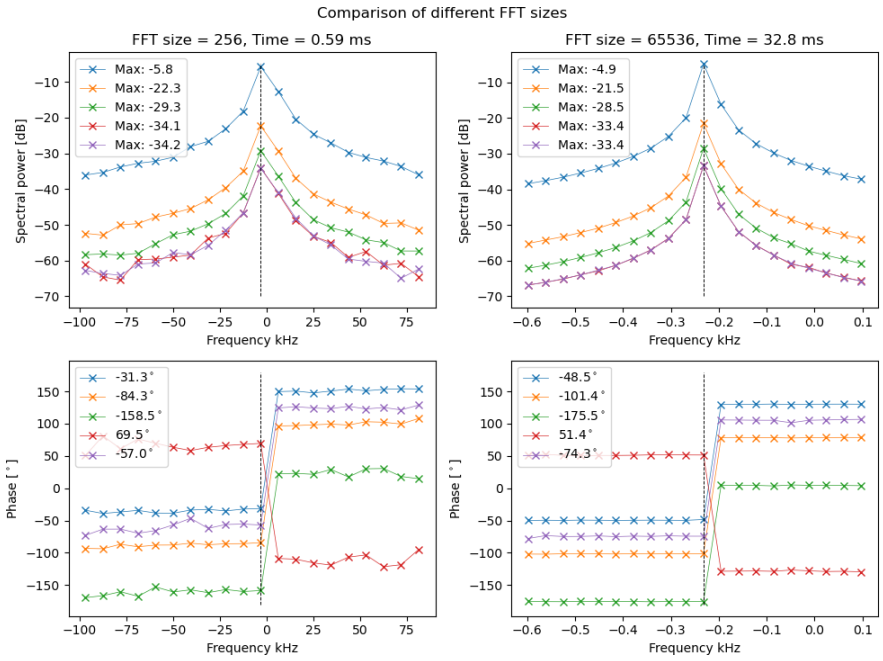


Figure 22 Differences in obtained frequency components of two different sizes for FFT. Zero on frequency axis is the current generator frequency (if the local oscillators were matched perfectly), note that the x-axis are different on left and right graphs. Only 20 samples are shown for any trace.

Beside looking at the output of the FFT the s-parameters should also be considered. What is interesting is how the s-parameters change depending on the size of the FFT. In equation (24) the division to compute the S-parameters are shown. Using this equation on the same data set as before, the difference in s-parameter values can be visualised in the complex plane as in Figure 23. This show a minimal effect on the precision of the output as the computational complexity increases.

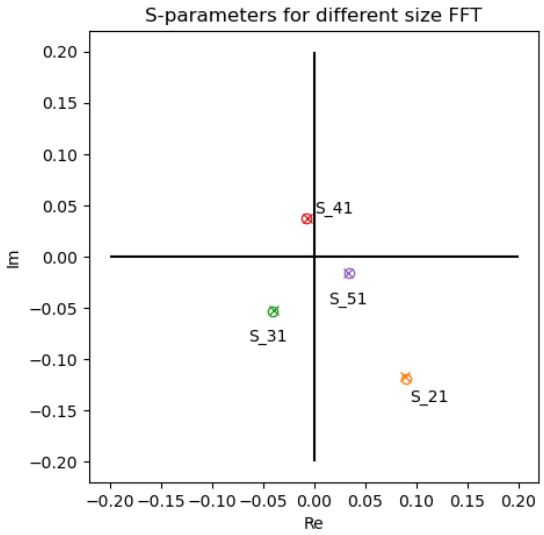


Figure 23 Differences in obtained S-parameters of two different sizes for FFT.

Table 2: Differences in s-parameters for two different size FFT

FFT size	$2^8 = 256$	$2^{16} = 65536$	Difference
Re(S_{21})	-0.683	-0.683	0.0
Im(S_{21})	0.739	0.740	0.001
Re(S_{31})	0.543	0.544	0.001
Im(S_{31})	0.647	0.651	0.004
Re(S_{41})	0.045	0.039	0.006
Im(S_{41})	-0.466	-0.462	0.004
Re(S_{51})	-0.148	-0.151	0.003
Im(S_{51})	0.585	0.588	0.003

To draw some conclusions from this a size of 256 samples are big enough to run a live calculations. To increase the precision of the calculation the size of the FFT can be at least 65 thousand samples can be used. This will require much memory thus would the bigger FFT size be a bad choice for live scripts since the data acquisition uses much ram memory.

3.4.2 Excluding algorithm

Here there should be an explanation on how the excluding algorithm works. In a nut-shell:

Packet valid if:

- The header from KrakenSDR software have the correct center frequency
- The maximum value of each FFT bin is above a threshold
- The bin with most energy have the same index for all channels.

3.4.3 S-parameter matrix

It is important to have good understanding of what should be measured in this experiment. The coupling between the antenna elements sounds clear but this is not a physical quantity which can be measured directly. Instead mathematics and measurements of other quantities must be use. The end goal is to measure some of the S-parameters for the system, where the antennas can be isolated as a DUT with 5 ports, see figure 24.

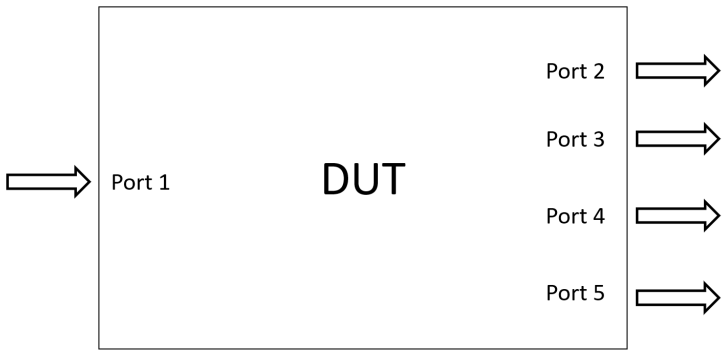


Figure 24 The system described as a DUT network

From the theory section this mean that a matrix of 5×5 elements are needed for full characterisation of the system. This is shown below:

$$\begin{array}{c|ccccc} & Tx & & & & \\ \hline & S_{11} & S_{12} & S_{13} & S_{14} & S_{15} \\ & S_{21} & S_{22} & S_{23} & S_{24} & S_{25} \\ Rx & S_{31} & S_{32} & S_{33} & S_{34} & S_{35} \\ & S_{41} & S_{42} & S_{43} & S_{44} & S_{45} \\ & S_{51} & S_{52} & S_{53} & S_{54} & S_{55} \end{array} \tag{22}$$

For the purposes of this master thesis project, a full characterisation is not necessary. Due to limitations in hardware and lab setup the only port able to transmit signals are Port 1, this will remove all columns in (22) except for the first where Port 1 is used as input. Since the reflections at Port 1 can not be measured with the current setup, the element S_{11} will also be neglected.

When removing the parameters that can not be measured from the matrix in (22) the only parameters left are:

$$\begin{array}{c|c} & Tx \\ \hline Rx & \begin{array}{c} S_{21} \\ S_{31} \\ S_{41} \\ S_{51} \end{array} \end{array} \tag{23}$$

Where each value can be expressed as a fraction of the FFT output at the given frequency ($A_i \in \mathbb{C}$):

$$S_{i1}(f) = \frac{A_i(f)}{A_1(f)} \tag{24}$$

This fraction should be evaluated at each frequency of interest, thus creating a final output matrix of size $(4 \times f)$. Where f is the number of measured frequencies and thus, each column represents a measured frequency. The rows will represent the respective S-parameters (four in total). A separate matrix will store the frequency value which each column corresponds to. These values are best stored as a touchstone file. This can be created using scikit-rf (skrf) which has a function called *.write_touchstone()*. Obtaining this file and comparing the values with a high end VNA are the main objective of this report.

3.5 Experiments

This section explains the different experiments which were used to get data from the lab setup.

3.5.1 Single packets

The first experiment which was tested was to collect single packets at Kraken. This experiment was mostly used for debugging purposes and the idea was to start the signal generator at some arbitrary frequency, with a Continuous Wave (CW). Once the generator was turned on, one packet was extracted from Kraken. Note that this packet

must be collected after the initial calibration procedure which Kraken software starts automatically. For a data packet to be recognised as valid it had to meet the following conditions:

- Type was data (Not Cal or Dummy which is sent during calibration)
- Frequency of the packet header matched with expected header (Packet can build up in a queue)
- In frequency domain the bin with maximum power was the same for all channels (else noise was picked up)

If a packet was collected with theses requirements the data acquisition was terminated and the generator was turned of. The data collected from this experiment was:

- Full ADC matrix
- FFT data of a predetermined size often 2^{12}
- Spectrum plot
- S-parameters in a scatter plot

3.5.2 Frequency sweep

For recreation of a VNA, single packets were not enough. Measurements had to be done across the frequency span of the receiver, for a well defined system many frequencies should be measured. This project never really got to the part of fully determining the entire frequency spectra. Thus, different frequencies were measured over the entire spectra to get many measurement points which later could be compared to the real VNA. This experiment was named frequency sweep and the idea was to take X uniformly spaced frequencies and run both Kraken and the generator at these frequencies one at a time. A offset between Kraken and the generator was used to avoid the risk that the energy from the mixer leaked into the measurements.

At each frequency packets were collected with the same requirements as for the single packet experiment, with the difference that more packets were collected. The signal processing software did not change frequency before ten packets were collected at each frequency. Since more packets were collected in this experiment it was not possible to save as much data as when single packets were collected. What was of most importance was the FFT output, this was used instead of ADC values because of the smaller

size of the FFT output compared to the full signal which was 5 millions samples per packet. Besides the FFT values a list of all S-parameters were also collected. These S-parameters were also shown in a scatter plot at the end of the frequency sweep. The goal from the start was to have a live window plot the S-parameters as they were collected. This was not implemented due to time constrains, the idea was tested but with a solution which run to slow and could thus not keep up with the measurements.

3.5.3 Kraken Sweep

During the frequency sweep data analysis it was discovered that Kraken had a lot of phase noise effecting the measurements. The reason for this noise was a mystery and a experiment had to be created which could tell more about what was at fault for this noise. The experiment used for this was called Kraken sweep and the idea was to only change the frequency of Kraken not the signal generator. This meant that it was a limited spectra to be used, only 2.4 MHz since that is the bandwidth of KrakenSDR. By setting the generator to some frequency and setting Kraken to center frequencies within the region where this signal can be detected allowed for many calibration procedures to be done without changing any other parameters. The S-parameter output should be the same regardless of the center frequency of KrakenSDR since the S-parameter is determined in the frequency domain which takes the center frequency in consideration. This experiment differed from the frequency sweep in that Kraken did stay at each frequency for 10 seconds no matter how many packets were collected. The reason for this was that if the generator and Kraken were to far offset from each other it might not be possible to pick up a signal and thus the measurement would never be completed. Now with a set time of each frequency step, if no packets were collected that step was simply discarded and the measurement moved on.

4 Results and Discussion

Experiments and tests were done along the entire time of the project. The earlier with the purpose of learning to use KrakenSDR and to make plans for the final test to extract S-parameters.

4.1 Single packets

Each received data packet from KrakenSDR will have five components, one from each receiver channel. By sending a sine wave from the signal generator, a signal of the same frequency should be detectable at all receiver channels. In Figure 25 ADC levels are plotted, this illustrates that a signal is present at both the reference point and at the antennas. The power of the signals can be observed to be very different in magnitude. This is due to the losses between the antennas as the signal propagates in the air. Channel 1 represents the sending antenna and thus, should have the highest signal power. A phase offset can also be seen, this is an effect of the propagation distance between sending and receiving elements.

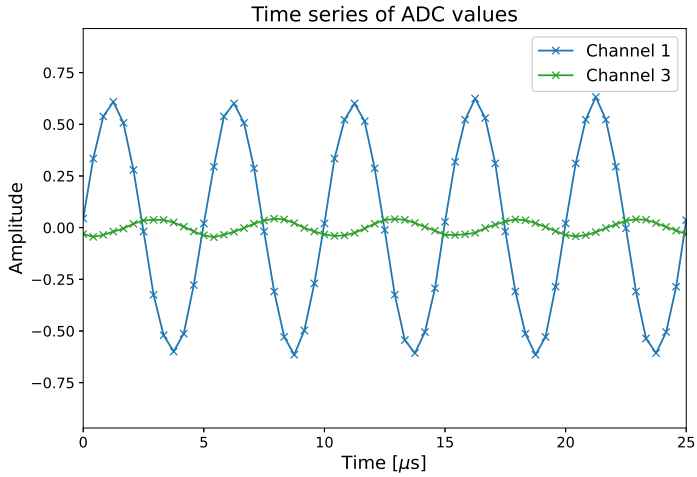


Figure 25 Time series of ADC levels for two of the five channels. First 25 μs of the half second signal.

To extract frequency components of the signals, the FFT algorithm is applied, resulting in the spectrum from Figure 26 instead of a time series. The signal generator clearly produces a signal high above the noise floor.

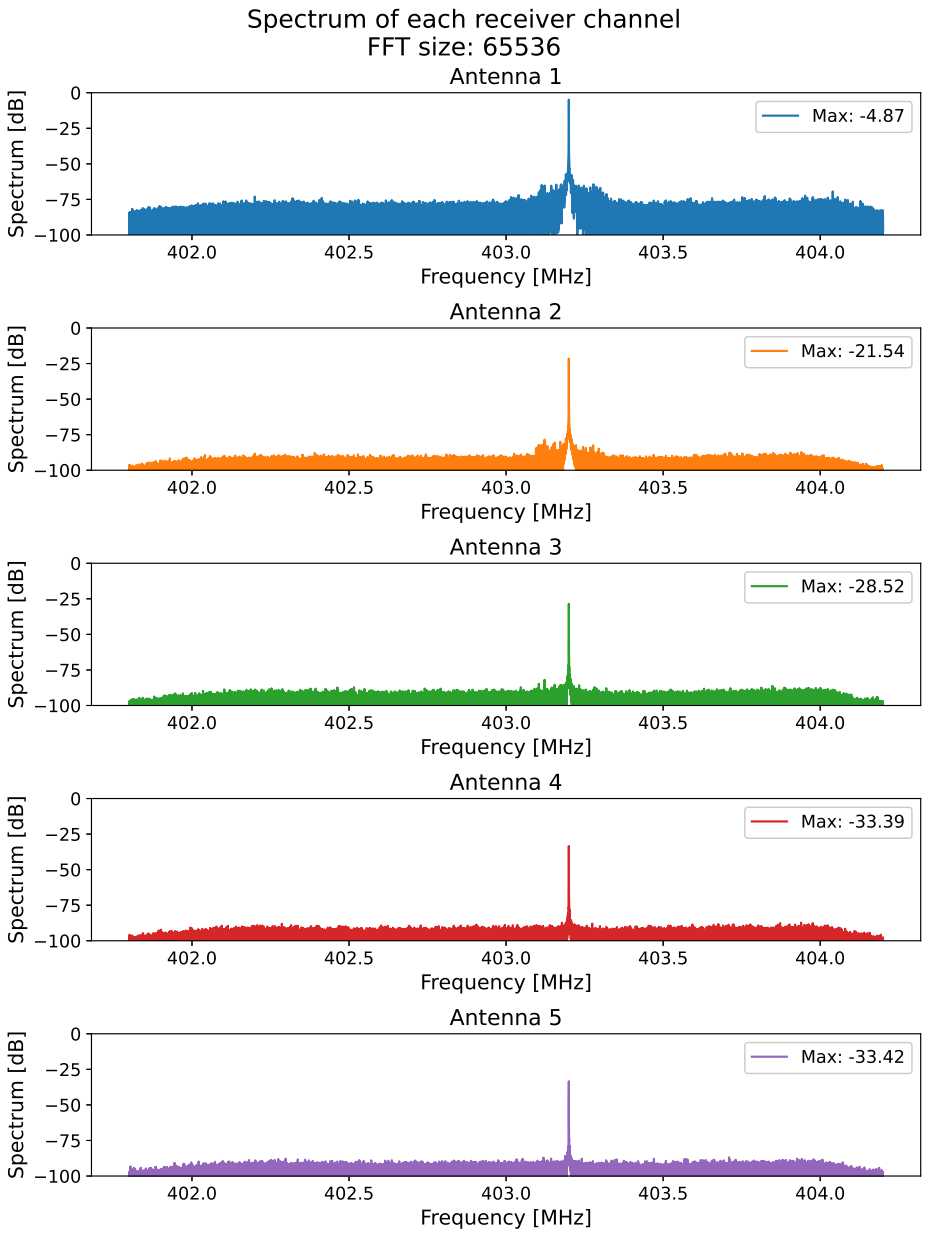


Figure 26 Spectra of all channels as signal generator is set to 403.2 MHz.

To preform the necessary calculations for estimating S-parameters, some information must be extracted from the data set used to generate the spectrum in Figure 26. These values are shown in Table 3.

Table 3: Frequency domain of the data in Figure 26.

Antenna	Index of maxima	Amplitude $\lvert A \rvert$	Phase [°]	Complex Value
1	38223	0.571	-48.45	0.378-0.427j
2	38223	0.084	-101.41	-0.017-0.082j
3	38223	0.038	-175.515	-0.037-0.003j
4	38223	0.021	51.441	0.013+0.017j
5	38223	0.021	-74.323	0.006-0.021j

Using the data from Table 3 the S-parameters can be calculated using (24). The result of this computation is shown in both Table 4 and in Figure 27. How well these values match the real S-parameters are hard to quantify without doing a reference measurement with a calibrated VNA.

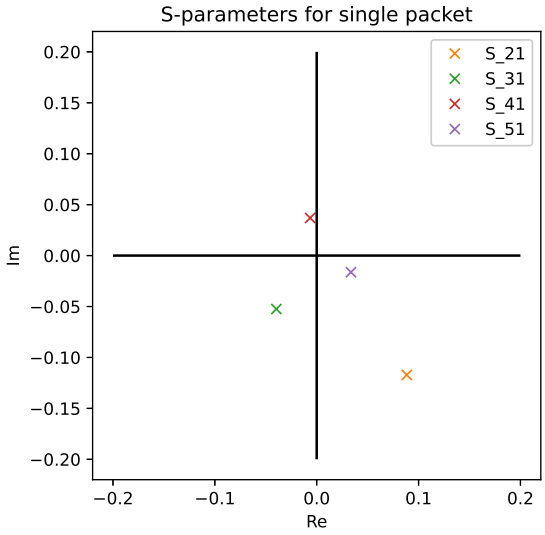


Figure 27 The calculated S-parameters for frequency 403.2 MHz.

Table 4: Values used to generate Figure 27. S-parameters for frequency 403.2 MHz.

Parameter	Complex value	Amplitude $\lvert A \rvert$	Phase [°]
S_{21}	0.088-0.117j	0.147	-52.959
S_{31}	-0.04-0.052j	0.066	-127.065
S_{41}	-0.006+0.037j	0.037	99.891
S_{51}	0.034-0.016j	0.037	-25.872

4.2 Frequency Sweep

With a working procedure to collect one packet of data, the next experimental phase of sweeping multiple frequencies could begin. In this experiment a total of 28 frequencies were included in the sweep and the time spent at each frequency was just enough to collect one valid packet. Valid in this experiment was is the packet header had the same frequency as the signal of interest. This filtered out any calibration packets and packets which had stacked up in the queue from the previous step. The range of frequencies for this experiment was 200 MHz to 800 MHz, this was chosen as a good compromise between getting an overview of the measurements across a larger frequency span, at the same time as the measurement points would be close enough to each other that they should be correlated to some degree.

The result of the frequency sweep for the S-parameter S_{21} are shown in Figure 28, for all other measured S-parameters see Appendix C. Considering S_{21} some interesting aspects are present. Both the phase and the power looks to be quite noisy, if this is a result of having the values to far apart or if the measurements simply are bad, that is hard to tell from this data. Despite the large variance of the in the magnitude plots, it still is centred around -22 dB and in the linear plot, which show a bigger picture of the possible values, the magnitude variations are less noticeable. Regarding the phase, this is plotted using the Numpy function unwrap, this allows for angles not to be bounded by -180° to 180° but instead the function makes a guess of the current phase depending on previous values, this removes the constant phase oscillations as a full rotation in the complex plain will continue to add up beyond 360°.

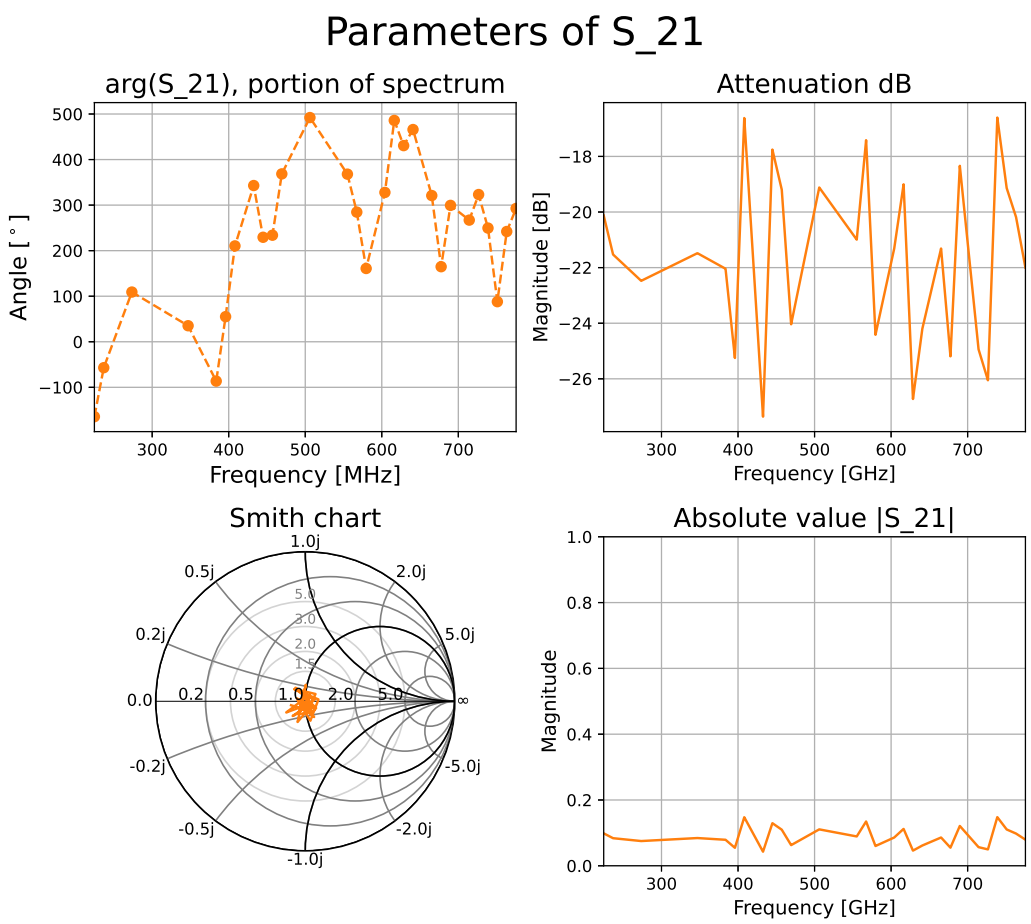


Figure 28 An overview of the measured S_{21} parameter for the frequency sweep experiment.

By comparing these measurements with the theoretical values, more understanding of the plausibility of the measurements are generated. See Figure 29 where the received power are compared to the theoretical values when using (14) to calculate received power.

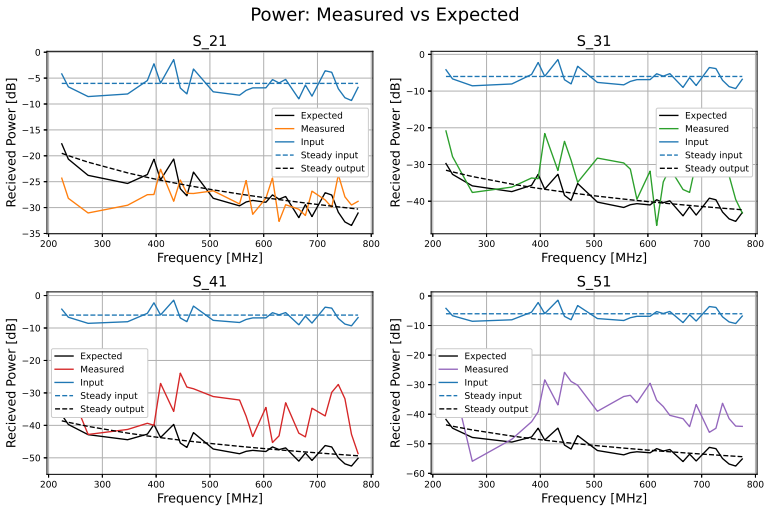


Figure 29 Comparisons between theoretical magnitude and measured.

This show that the S-parameter approximations made by the Kraken lab setup is able to compute about the correct magnitude of the real system. For the antennas further away from the source antenna the results looks to drift towards higher values. The reason for this is probably because the bigger distance allows for more reflections of the signal before it reaches the receiving antenna, the main reflection would thus be from the table where all the antennas are placed.

4.3 Validation

The frequency sweep done by Kraken also enables a comparison with a traditional VNA measurements to be made. Using a tested VNA directly on the antennas should return a better result compared to the theoretical values since his measurement will take into consideration reflections and imperfections of the antennas as well as it is a way of mapping the environment of which the signal is passed through. If only theoretical values were to be expected, the reason for using a VNA in the first place would be unnecessary. The goal of measuring the antenna array is to know exactly how the system looks at this point in time.

The reference measurements were done with a high quality VNA, which was first calibrated to ensure the best result. The VNA used could use four ports, one less then Kraken, thus the last antenna was disconnected from the setup. It was left in the array to change as few parameters as possible. The measurements outputted a $(4000 \times 4 \times 4)$ matrix since all ports had the ability to both send and receive signals, 4000 was the number

of frequency components of the output. The measurements were between 100 MHz and 1 GHz. The spectrum of the estimated S-parameters are shown in Figure 30. Note that the reflection parameter S_{11} are included in this result, this is included even though the lab setup used in this thesis project did not have the ability to compute S_{11}. However this parameter still can convey information about the system and was thus included.

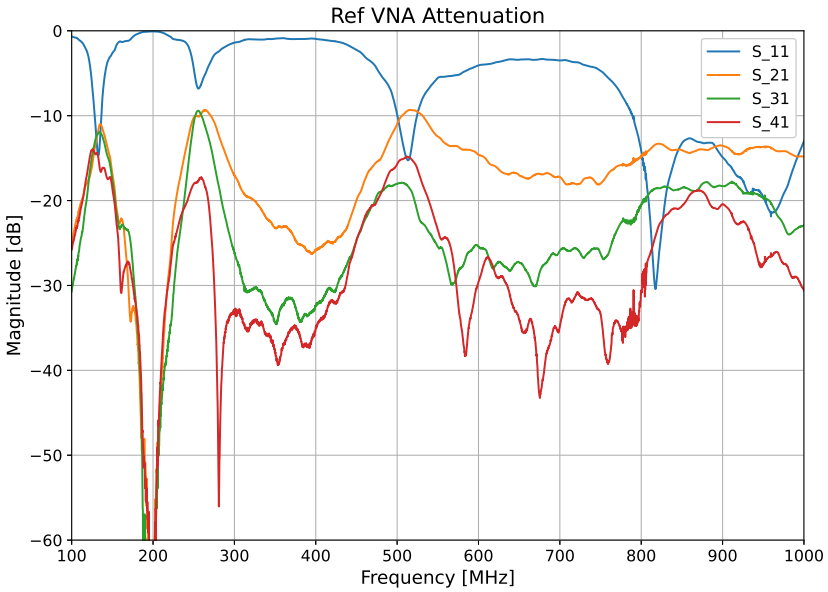


Figure 30 Example from reference VNA measurement. Areas of high and low antenna mutual coupling can be found.

In Figure 30, an interesting dynamic behaviour can be observed which was not present in the theoretical assumptions. For the parameter S_{11} regions of both high and low transmitted power are visible. Where S_{11} is 0 dB, it means that all power is reflected back from the antenna. This is because resonance occurs at this frequency and almost no power will leave the antenna. At these frequencies the other S-parameters will be low since the other antennas are not able to pick up any signal. This explains why the frequency sweep often stuck at 200 MHz. At this frequency no valid packet would never be detected. Thus, when a filter algorithm was tried, where a packet had to have a minimum power above a threshold this would never pass. As the output power was really low at this specific frequency the minimum threshold was never reached. With this knowledge future frequency sweeps should avoid 200 MHz as this point will introduce lots of problems in the calculations. The resonance peak at 200 MHz was not unique, another at about 375 MHz and 650 MHz was also observed. But these were not at

significant as the main peak and thus some signal could pass through which was enough for the signal processing to estimate a S-parameter.

The next step was to compare the measurements from KrakenSDR with the reference. To make the comparison easier, the different S-parameters were separated and plotted individually. Both for magnitude in Figure 31 and phase in Figure 32.

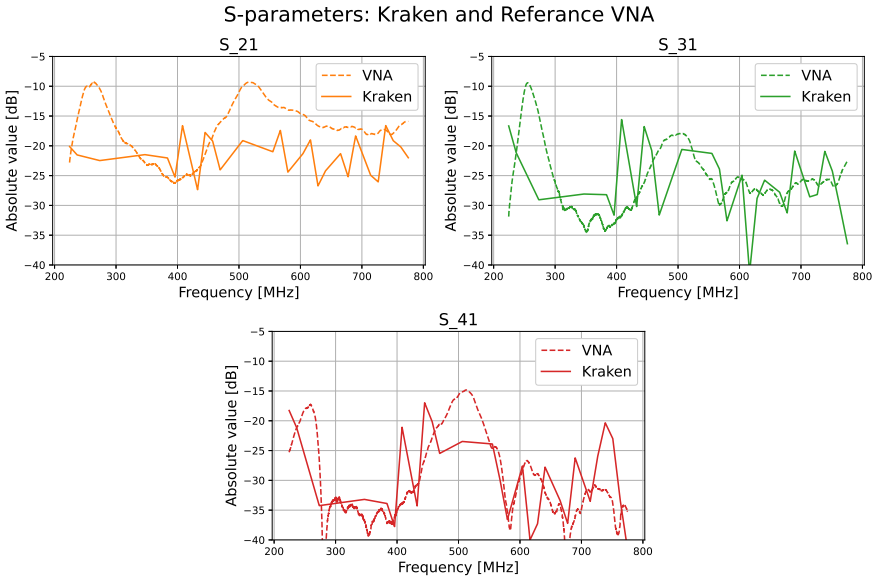


Figure 31 Comparison of Kraken and VNA magnitude.

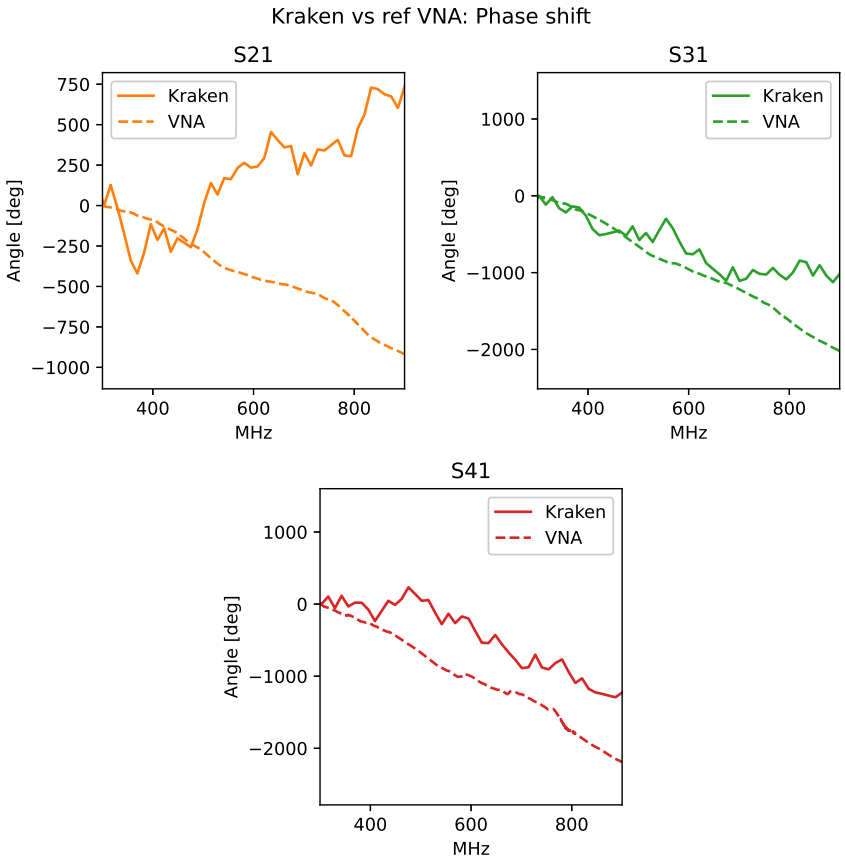


Figure 32 Example from reference VNA measurement. Phase plotted with unwrap function.

For the phase the measurements of S_{31} and S_{41} preformed by Kraken, the results correlated well with the standard VNA. However, for S_{21} a strange phenomena of the phase being added instead of subtracted occurs. To understand why, the unwrap function is removed, see Figure 33, where the extremely noisy phase measurements can be observed. Looking at this data, little reliability should be placed on the phase of the Kraken measurements at this phase. It was later in this project discovered that the using the first valid data packet introduces phase noise to these measurements. This experiment should have been retried with better filter algorithms. Because of time constraints, this was not possible.

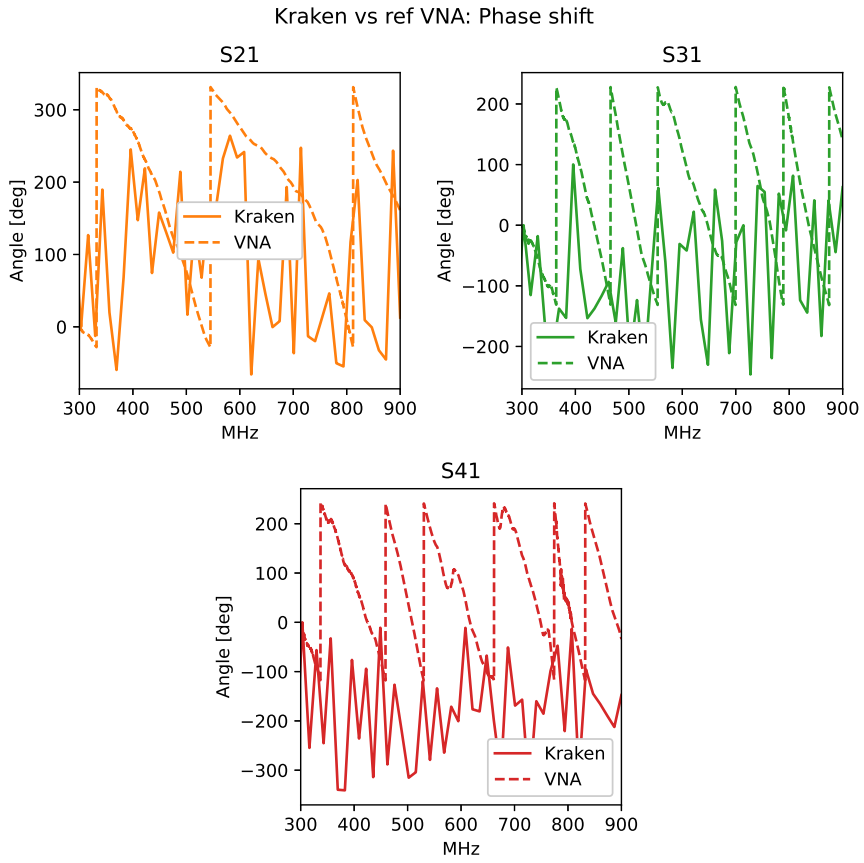


Figure 33 Example from reference VNA measurement. Phase plotted without unwrap function.

4.4 Kraken Sweep

For a better understanding of the accuracy of the estimated S-parameters a series of data packets was collected at one single frequency. The purpose of this was to gain more understanding of the noisy measurements, most noticeable in Figure 33. Noise in this context refers to the varying phase values which were present both across the frequency spectra but also as one frequency was measured multiple times. The experiment which generated data for this part of the thesis were Kraken Sweep. This generated a series of S-parameters and FFT data. The S-parameters are shown in Figure 34, which shows estimations for all valid packets. Again lots of noise are present in the measurements, however a majority of the S-parameters looks to be centred around a value while some are really off. The reference VNA values are also shown and theses are the target values. However at this point it was quite clear that the setup needed a better calibration and the

main focus going forward was to reduce the noise, this allows for a calibration to be more useful in the future. Lastly S_{51} is not included in Figure 34 because that reference value was missing from the VNA.

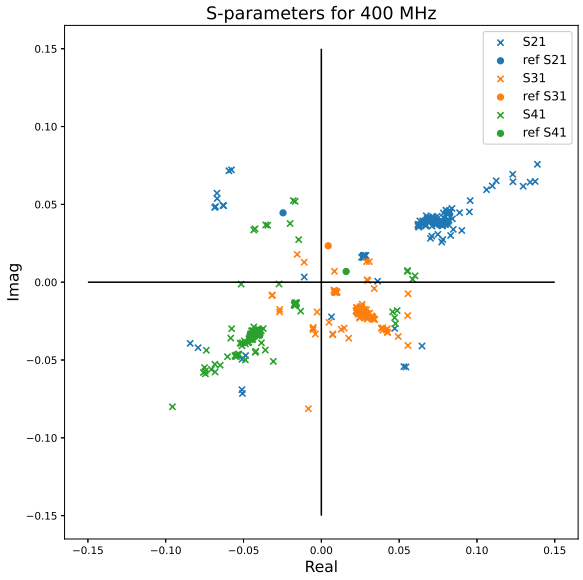


Figure 34 The S-parameters from sweeping Kraken at 20 different frequencies around 400 MHz. The number of samples included in the FFT conversion was 255.

The data producing the plot in Figure 34 was a total of 171 packets collected at 10 different frequencies. Thus, 10 center frequencies failed to collect any packets at all. To be expected as the lowest frequency tested was 398 MHz where Kraken has a bandwidth of 2.4 MHz and as the frequency moves away from the center frequency the gain of the signal decreases. For the frequency where a signal was discovered about 17 or 18 packets were collected during the 10 seconds of measurements.

Going forward with the goal of finding the reason for the bad accuracy of the measurements, a deep dive into the different packets had to be conducted. This was best solved by producing a series of plots as in Figure 35 where both the content of the FFT data and the resulting S-parameters can be visualised. These plots were generated for all collected packets and later looked through to find what the outlaying data had in common.

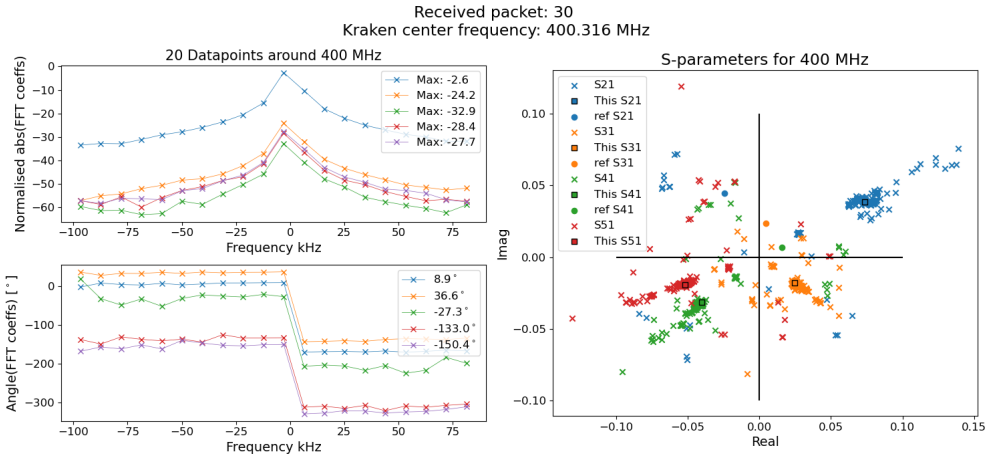


Figure 35 Collected packet id 30. A detailed visualisation of the FFT components on the left and all S-parameters on the right. The S-parameter generated from this specific packet is highlighted with a black box.

No big problems were discovered with aspect of both power spectra and phase. But as the Kraken center frequency changed, the phase of the measurements went to some random value and the accuracy decreased. This was present for some data packets before measurements settled down at the nominal values again. This phenomena was observed every time Kraken changed frequency. One of those were when 400.316 MHz which are shown in Figure 36. The first six packets are shown and for the last two the values have returned to normal again.

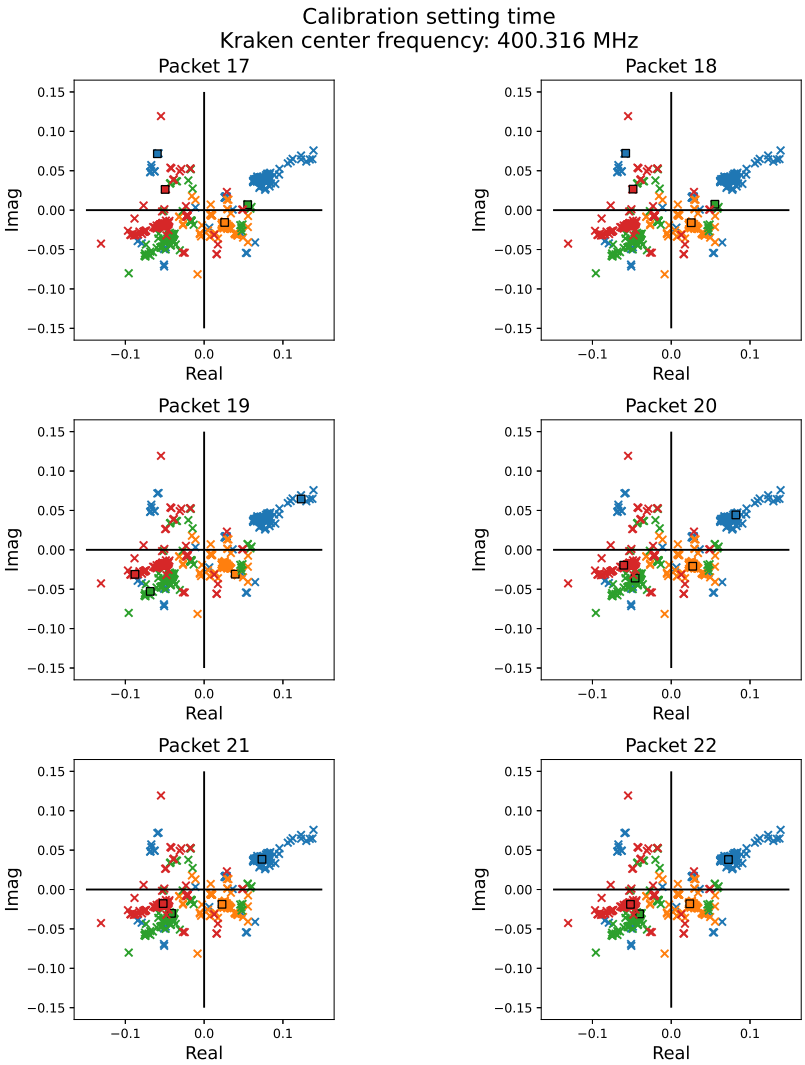


Figure 36 First to sixth received packet after Kraken calibration. Note the setting time before values stabilises.

Using this knowledge, the filter for data could be modified and improved upon. This was done by discarding the first first six packets after calibration. Applying this idea to the same data set as seen in Figure 34 reduced the data set but the ratio of good vs bad values increased. This result is shown in Figure 37, where a significant improvement in accuracy can be observed.

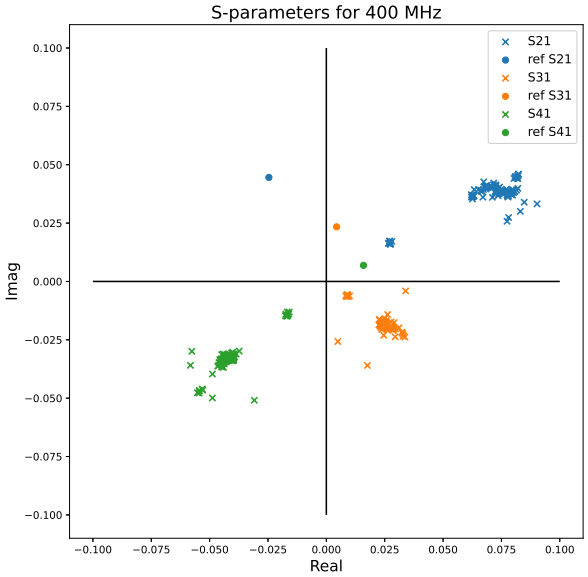


Figure 37 Scatter plot of data from Kraken Sweep experiment, first six packets after calibration is dropped.

With this improved accuracy, the results were much more reliable, but some noise or disturbances still present. Using the method of visualising S-parameters in the scatter plot with the black box one by one, the remaining bad samples could be identified. These bad values originated from two Kraken center frequencies, 398.842 MHz and 399.053 MHz see Appendix (INPUT PLOT) for scatter plots. These two frequencies turned out to be the furthest away from the generator and thus, the inaccurate values could be attributed to lower signal quality at these frequencies. To combat this, a further filter to only us frequencies which were within 0.8 MHz of the signal generator would be used in S-parameter computations. Using all of the above mentioned filters for the data the final result was pleasing as shown in Figure 38.

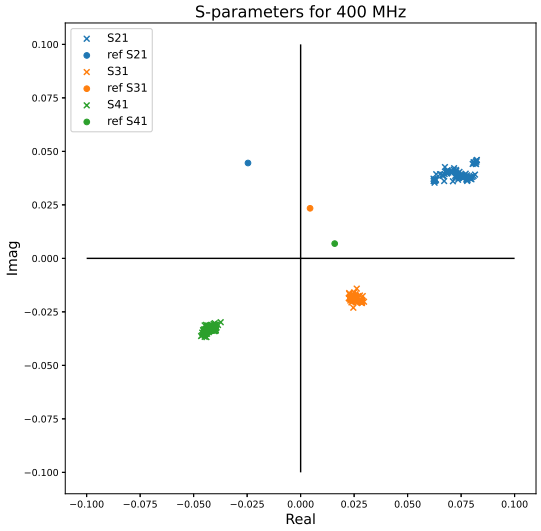


Figure 38 The final result after all filtering algorithms.

With this knowledge of the important of correct filtering of data packets, the bad results from previous experiments can be explained. In Figure 31 to 33, no packets were filtered out and thus much of the information about phase and amplitude will be effected in a negative way. Using the findings in this section, the result could have been improved if there was enough time to implement as a new experiemnt.

4.5 Improvements

As the precision of the measurements got better, the need of calibration became obvious. Before using the lab setup to estimate S-parameters, cable calibrations were done to match power and phase of the channels. The power or magnitude calibration returned good results but from the reference measurements a clear phase offset was observed. Comparing the calibration coefficients for the black cable, as one set of coefficients were from the Kraken software and the other were measured in the lab. This showed that the phase of the coefficients not matched well, for the specific frequency used in the Kraken Sweep (400 MHz) the difference were $110°$. Removing this offset was not enough to rotate the measured values to the correct positions. This indicates that there are more errors present.

A good solution would be to preform a full five port VNA calibration where as many error terms as possible are determinate. However since only one port is used as a trans-

mitter, the only necessary calibration procedure needed to be done are a through measurement. This is when the ports are connected directly to each other and measured. Connecting the receiver and transmitter directly into each other, should return one. This is because no phase or magnitude should occur when the ports are short circuited to each other. Any offset from one can be considered as the calibration value which should be used to compensate all future measurements.

5 Conclusions

This thesis have successfully been able to measure S-parameter magnitude between elements in an antenna array. An algorithm for receiving reliable data using a standard grade commercial software defined multichannel radio receiver were determined. This method achieves phase measurements with high precision but centred around wrong values. The reason for bad phase measurement are believed to be a bad cable calibration.

The overall goal of the thesis was to do a proof of concept of the software defined VNA using hardware which can be found in a typical antenna communication system. Given the result of high precision in the result with little calibration, it can be concluded that a VNA can be used in a multichannel digital antenna array communication system. This would be without any extra hardware requirements. Including the VNA system can be a great compliment to a systems ability and diversity but it will halt other communication and power will be consumed during measurements. For this reason the use of the software defined VNA is only applicable where other algorithms require known S-parameters to work as intended. One such example, as stated before would be the beamforming necessary to perform bidirectional communication.

6 Future work

To create a better VNA measurement an interesting procedure would be to use more than a single sine wave. When only one wave is used at a time it increases the total measuring time significantly, since only one data point is generated. A better solution would be to use multiple signal waves of different frequencies. As long as all frequencies are within the band limit of the receiver (for this project approximated to 0.8 MHz), this allows the FFT to compute the frequency bins for all these sine waves at the same time. If this is done with a large enough FFT size, it allows for the signals to be completely independent and thus the precision of the measurements would not be affected.

Another idea for a more efficient system would be to not perform the complete FFT on the signal but instead only calculate the specific frequency bin where the signal is expected. This removes many of the calculations which are discarded in the frequency domain. What should be calculated in this case would be the complex exponential of the specific frequency bin where the stimulus signal is located.

$$FFT[k] = e^{i\omega_k t} \cdot X[t] \tag{25}$$

In the equation above, the frequency bin with index k is calculated using the complex exponential for this frequency and the received samples $X[t]$. This method requires the system to know the frequency of the stimulus signal with high accuracy, this is harder when the receiver and signal generator runs on different clocks. However, in a full communication system, the receiving and sending units would be run on the same clock and thus have little variation in frequency.

Calculating single frequency bins can still be used if multiple stimulus signals are present. The complexity would increase linearly as more stimulus signals are added. Using this method must still be tested to show how computational time and measurements accuracy are effected. The theoretical assumption would be as more of the data being processed would be used, the ratio between accuracy and computational time should increase.

Acknowledgements

During my time at SAAB doing this project I have been in good hands of my supervisor Johan Malmström, he was the one to first propose the project and have come with good help as the inevitable problems appeared. Mats Järgenstedt was the manager of the department at SAAB where to project took place. Both Mats and Johan took great interest in the project and gave the support needed for finishing the project.

At Uppsala University, Mikael Sternard was the subject reader and contributed with help of making sure the project was heading in the right direction and that the goals of the project matched expected results. Thank you all for the help you have contributed with!

Also to be noted, this project were excellent in finding the intermediate ground between industry and the academics. Many different problems could be solved due to knowledge from previous courses at Uppsala University. This has shown that the staff at Uppsala has done a good job of finding course content and teaching this in a meaningfully way.

References

[1] *Anapico Downloads Maunals*. Accessed: 2023-04-14. URL: https://www.anapico.com/downloads/manuals/.

[2] *Antenna Electronics*. Accessed: 2023-02-15. URL: https://www.britannica.com/technology/antenna-electronics.

[3] Constantine A. Balanis. *Antenna Theory Analysis and Design*. John WileySons Inc, 2016.

[4] *BasicSDR*. Accessed: 2023-04-03. URL: https://github.com/gallicchio/basicSDR.

[5] *File:Transmission line element.svg*. Accessed: 2023-04-04. URL: https://commons.wikimedia.org/wiki/File:Transmission_line_element.svg.

[6] *github krakenrf*. Accessed: 2023-01-31. URL: https://github.com/krakenrf.

[7] Andrea Goldsmith. *Wireless Communication*. Cambridge University Press, 2005.

[8] *IQ Sampling*. Accessed: 2023-04-03. URL: https://pysdr.org/content/sampling.html.

[9] *KrakenSDR*. Accessed: 2023-01-31. URL: https://www.crowdsupply.com/krakenrf/krakensdr.

[10] *Nyqvist Criteria*. Accessed: 2023-01-31. URL: https://www.analog.com/media/en/training-seminars/tutorials/MT-002.pdf.

[11] Carl Nordin Jonny Österman. *Physics Handbook for Science and Engineering*. Studentliteratur, 2006.

[12] David M. Pozar. *Microwave Engineering*. John Wiley and Sons Inc, 2011.

[13] *Provided Scrips*. URL: https://github.com/Soder99/InterestingScripts.

[14] *RTL-SDR: Block Diagram and Information*. Accessed: 2023-02-06. URL: https://aaronscher.com/wireless_com_SDR/rtl_sdr_info.html.

[15] Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, 1999.

[16] *So, what are S-parameters anyway?* Accessed: 2023-04-04. URL: https://e2e.ti.com/blogs_/b/analogwire/posts/what-are-s-parameters.

[17] *VNA Sketch*. Accessed: 2023-04-11. URL: https://en.wikipedia.org/wiki/Network_analyzer_(electrical)#/media/File:Vna3.png.

[18] *What is 5G beamforming, beam steering and beam switching with massive MIMO*. Accessed: 2023-02-17. URL: https://www.metaswitch.com/knowledge-center/reference/what-is-beamforming-beam-steering-and-beam-switching-with-massive-mimo.

A Calibration Data

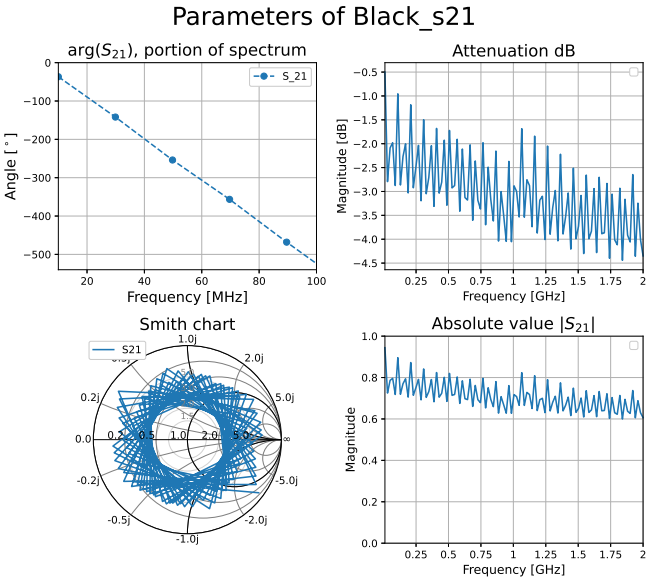


Figure 39 Data from black cable.

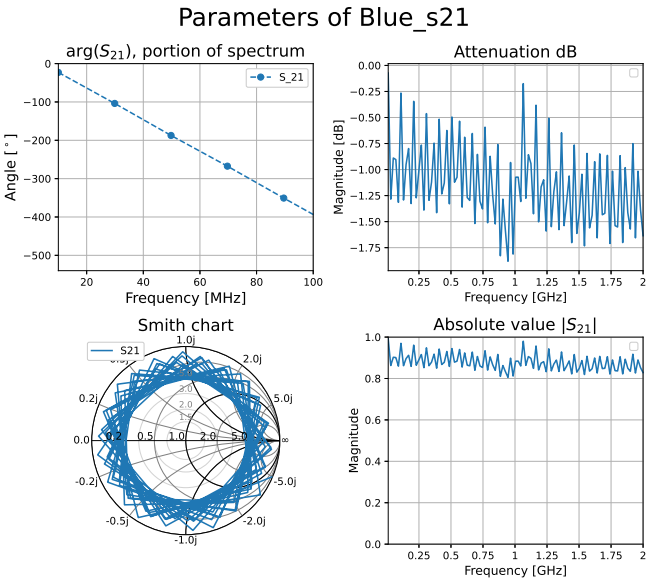


Figure 40 Data from blue cable, notice lower attenuation (better cable).

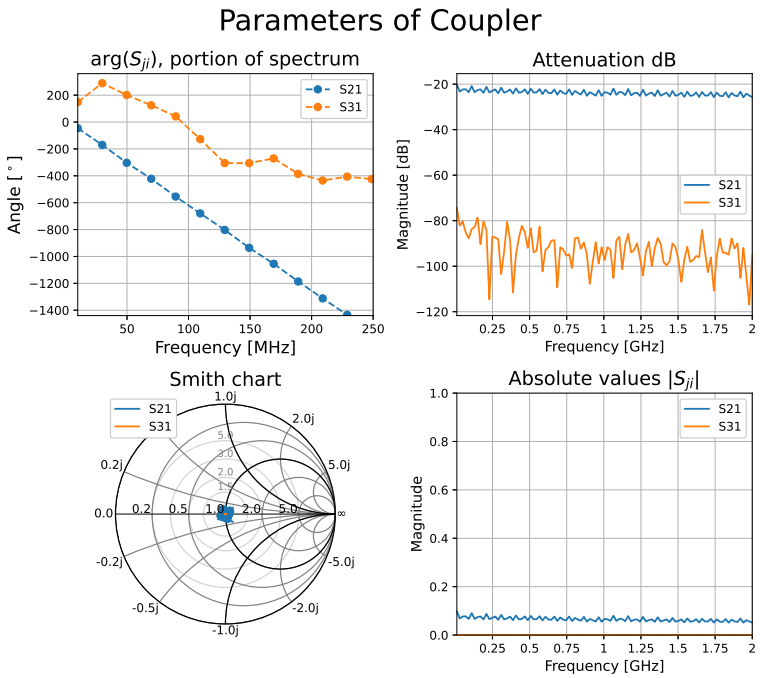


Figure 41 Data from coupler, only S_{21} and S_{31} are shown, the other will not effect the result.

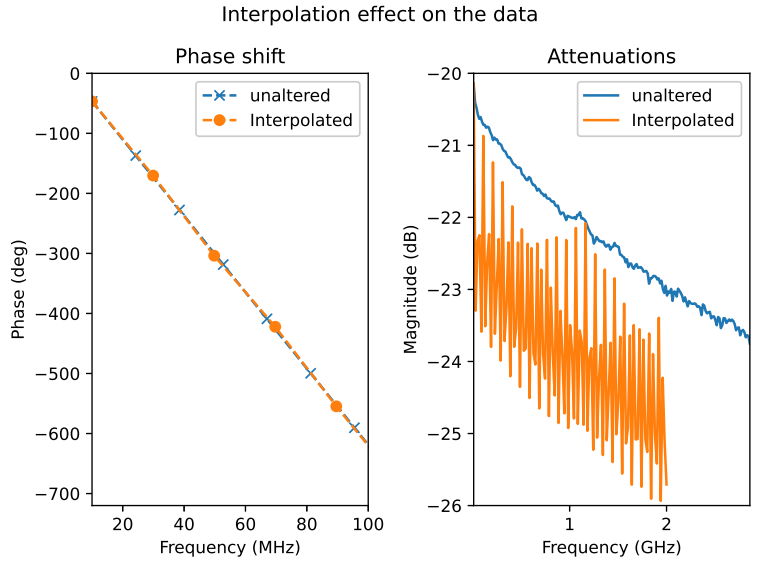


Figure 42 Check to see how interpolation effects the S-parameters. Difference is only noticeable for attenuation.

B Get started with KrakenSDR

Below some instructions to using KrakenSDR will be presented, both installation processes and general usage. All of the software from the developers of Kraken are provided in their github [6].

To install the provided software for data collection and DOA calculation enter the repository *krakensdr_docs* and download as zip. Once downloaded and unzipped go into the folder and further into *install_scripts* and type *chmod +x krakensdr_x86_install_doa.sh* (change permission to run the file) followed by *./krakensdr_x86_install_doa.sh* (run the file). This will download the software and the necessary packages, the installation takes some time and when done the folder *krakensdr_doa* should be in the home folder.

Once the installation is complete, it can be found in the *krakensdr_doa* folder, where the file *./kraken_doa_start.sh* can be run to start both the data acquisition and DOA software. The hardware can be controlled by open a browser and go to 0.0.0.0:8080 this user interface will allow to change parameters such as layout and frequency. Measurements from the antennas can also be shown in power spectra or DOA estimation, this can be useful to make sure that the hardware is working and that data is collected.

In many cases, the script for DOA estimation and web interface is not useful and will slow down other computations. Thus, the unnecessary parts of the code should be removed. The krakenrf github has supplied code for only starting the heimdall part of the software, this is data acquisition and coherence calibration. The scripts for running only heimdall is found at: https://github.com/krakenrf/gr-krakensdr and are named: *heimdall_only_start.sh* and *heimdall_only_stop.sh*. They have to be placed in the same directory as the kraken installation. Before running the start and stop script the permissions must be changed to be executable. This is done with *chmod +x heimdall_only_start.sh* and *chmod +x heimdall_only_stop.sh*, running these scripts will start a server that fetches the ADC data but this must be used along side some client that collects the data.

The best way to communicate with kraken is to write python code to communicate over ethernet with the c++ software that control the hardware of Kraken. To enable the ethernet communication, the DAQ setup must be changed from shared memory (shmem) to ethernet (eth) in the file daq_chain_config_ini near the bottom of the file.

B.1 Change Touchstone configuration

To apply the new settings to the KrakenSDR software one must change settings for the heimdall software. More precise in the code *delay_sync*, where on lines 190 to 218 the iq adjustments are set. This code firstly reads from the file *daq_chain_config.ini* and *[calibration]* → *iq_adjust_source*. This can be *explicit-time-delay* (will use values from *daq_chain_config.ini*) or *touchstone* (will use s-parameters from the s1p files in folder *_calibration*).

C Frequency Sweep Result

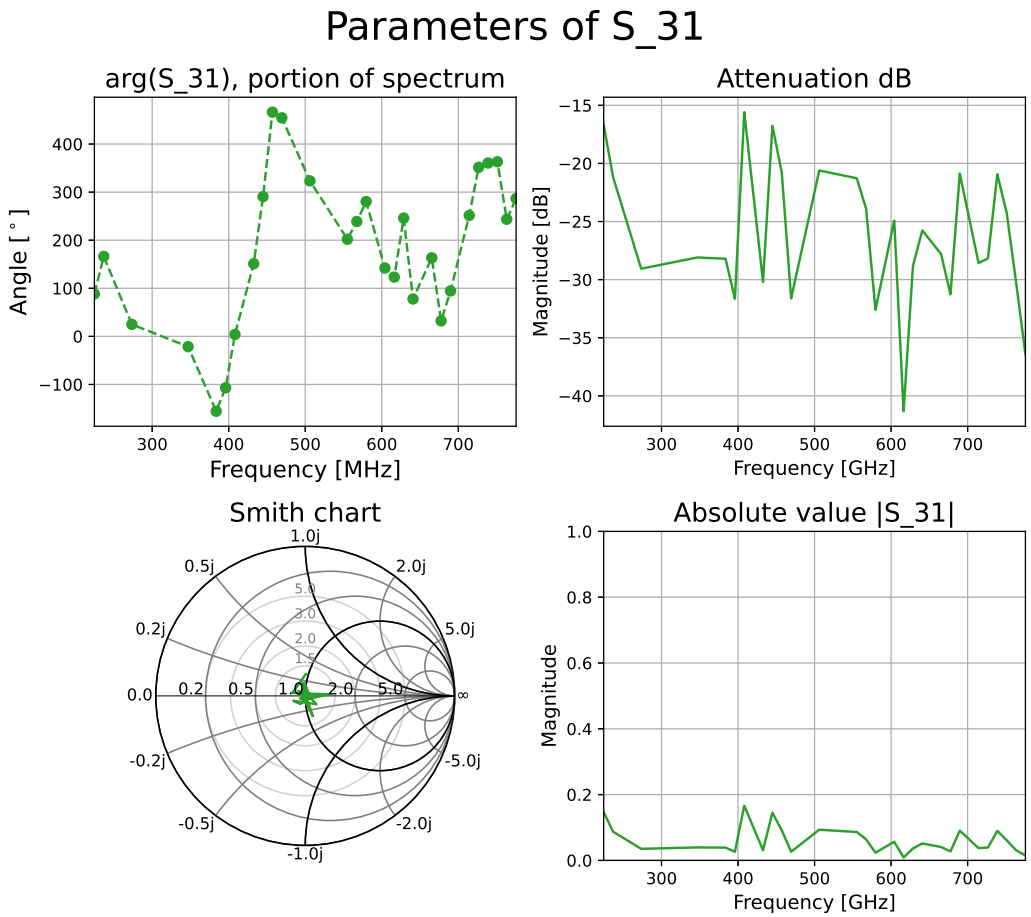


Figure 43 An overview of the measured S_{31} parameter for the frequency sweep experiment.

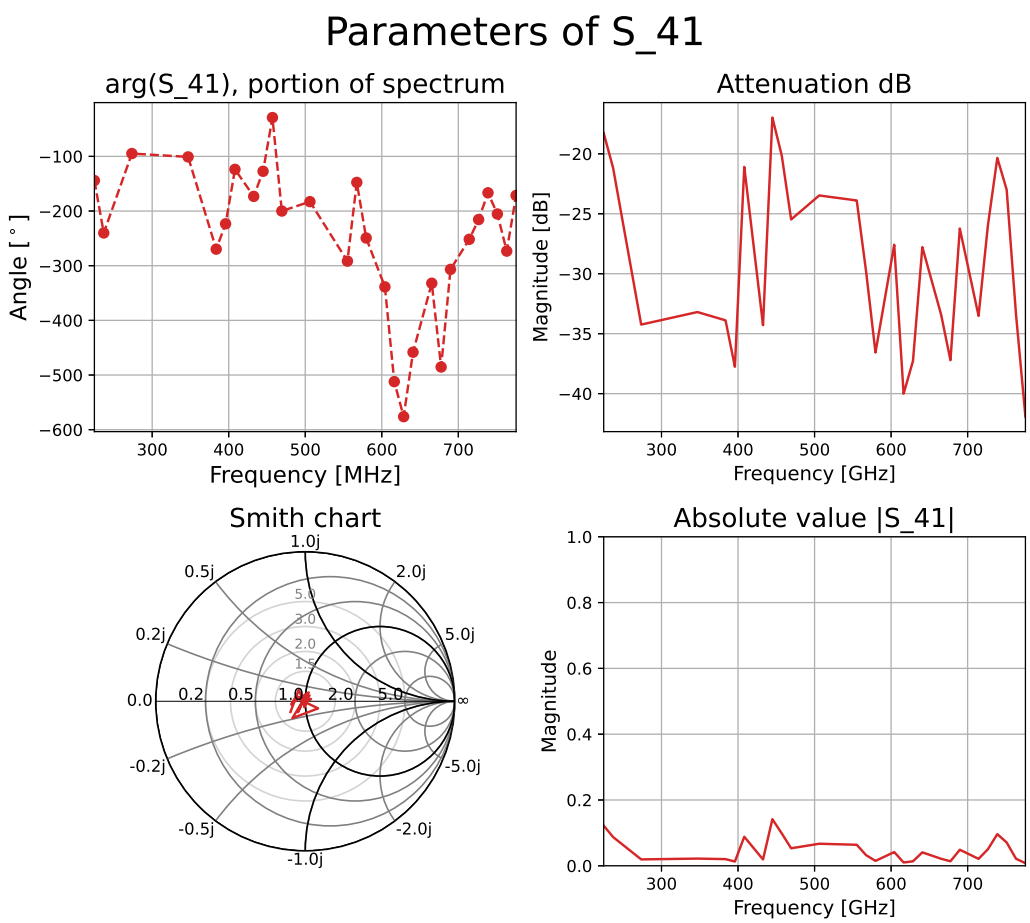


Figure 44 An overview of the measured S_{41} parameter for the frequency sweep experiment.

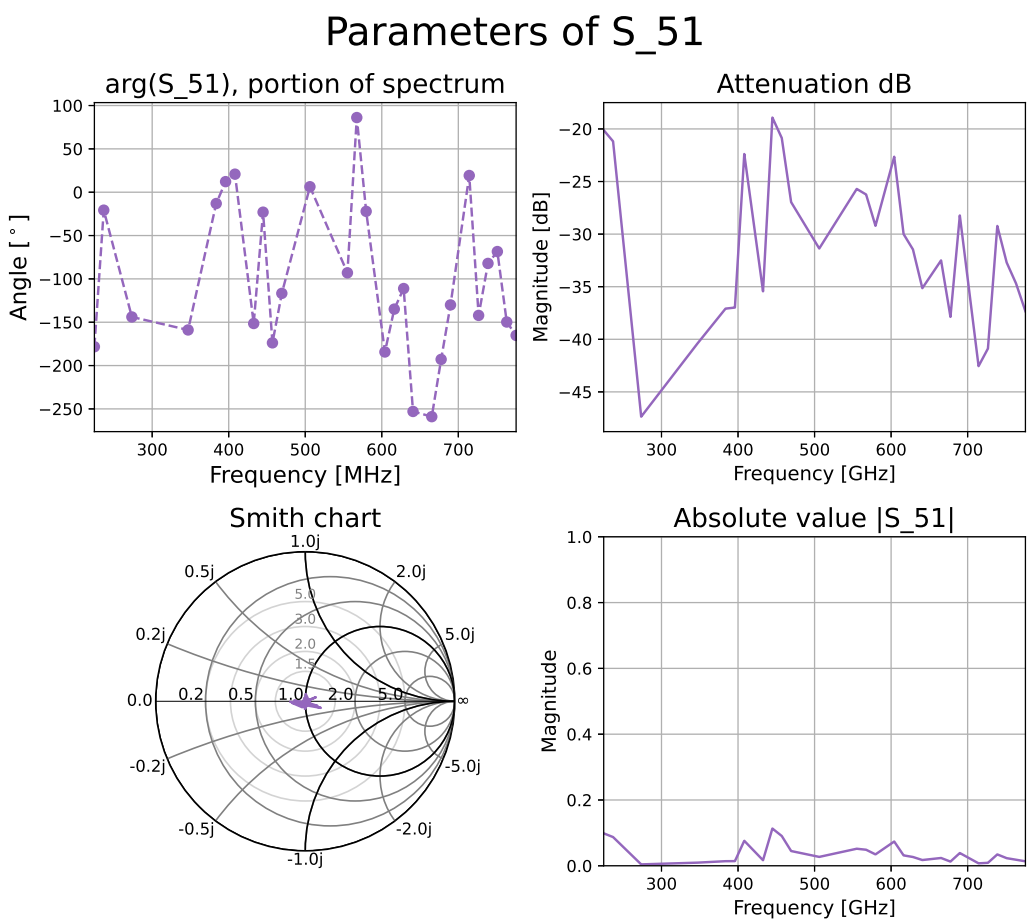


Figure 45 An overview of the measured S_{51} parameter for the frequency sweep experiment.