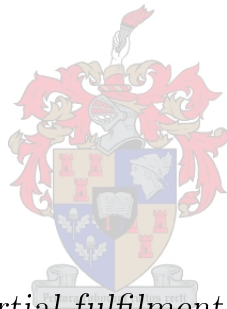


# Hardware limitations of interference suppressing beamforming

by

Shane Alexander Moyce



*Thesis presented in partial fulfilment of the requirements for  
the degree of Master of Engineering (Electrical) in the  
Faculty of Engineering at Stellenbosch University*

Supervisor: Prof. D.I.L. De Villiers

April 2019

# Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: ..... April 2019 .....

Copyright ©2019 Stellenbosch University  
All rights reserved.

# Abstract

## Hardware limitations of interference suppressing beamforming

S.A. Moyce

*Department of Electrical and Electronic Engineering,  
University of Stellenbosch,  
Private Bag X1, Matieland 7602, South Africa.*

Thesis: MEng (EE)

April 2019

This thesis proposes two blind beamforming techniques for use in global positioning system (GPS) applications as well as a dual band multi-layered micro strip antenna for use at L1 and L2 GPS frequencies. Various direction of arrival (DOA) algorithms were tested to investigate their failure points.

The antenna was built and measured. It was found that  $S_{11}$  is below -10 dB across the L1 and L2 bands from 1.22 GHz - 1.57 GHz. The gain is 1.7 dB below the simulation value in the L1 band however the L2 bands gain matches the simulation value of 7 dB. The results show that the antenna will function for GPS applications.

The algorithms were to be tested, in practice, using the transient array radio telescope (TART) system which is a 24 element radio telescope designed for testing of imaging algorithms and surveying transient events. The system makes use of the MAX2769B universal GPS receiver which produces binary samples at a sample frequency of 16.368 MHz and a centre frequency of 4.092 MHz.

Algorithm 1 uses an orthogonal subspace beamforming method to null interference and place antenna gain in the directions of the signals of interest (SOI). Algorithm 2 uses linearly constraint minimum variance (LCMV) beamforming and a beam search method in order to find available GPS satellites and null interference. Both algorithms were tested thoroughly by varying parameters such as signal to noise ratio (SNR), signal to interference ratio (SIR), integration time and the number of analogue to digital converter (ADC) bits to find their failure points and determine if they are viable techniques on the TART system.

It was found that, when using a peak search method to determine success rates, and sweeping SNR, SIR, integration time (number of samples) and the number of ADC bits, the Bartlett DOA algorithm performs the best, possibly due to the number of signals present on the simulated array.

It is shown that the success rates of algorithms 1 and 2 are similar, for the parameters that were swept. According to simulations, both methods are viable beamformers, for GPS applications, on the TART system and are not limited by its hardware. Algorithm 1 is more sensitive to variations in SIR, SNR and integration time when using 1 bit shifted data. The complex bit shifted data is created using a technique that allows one to create quadrature (Q) data by shifting the in-phase (I) data 1 bit, and is only possible due to the intermediate frequency (IF) being one quarter that of the sample frequency. Algorithm 1 also requires a priori information about the number of SOI's which is not required for algorithm 2. Algorithm 2 is more computationally expensive compared to algorithm 1 for GPS applications.

Preliminary practical results were collected using the TART system, and analysed. The measurements taken show the potential for algorithm 1 to be implemented, as a beamformer, in the TART system, however thorough practical testing was not possible due to the TART system becoming unusable.

# Uittreksel

## Hardeware beperkings van steursein onderdrukkende bundelvorming

S.A. Moyce

*Departement Elektriese en Elektroniese Ingenieurswese,  
Universiteit van Stellenbosch,  
Privaatsak X1, Matieland 7602, Suid Afrika.*

Tesis: MIng (EE)

April 2019

Hierdie tesis stel twee blinde bundelvormingstegnieke voor vir die gebruik in GPS toepassings, sowel as 'n dubbelband mikrostrook antenna vir gebruik by L1 en L2 GPS frekwensies. Verskeie invalshoek afskattings algoritmes (DOA) is getoets om hulle valingspunte te ondersoek.

Die antenna was gebou en gemeet.  $S_{11}$  is onder -10 dB oor beide die L1 en L2 frekwensiebande van 1.22 GHz - 1.57 GHz. Die aanwys is 1.7 dB onder die gesimuleerde waarde in die L1 band, maar die L2 band aanwys stem ooreen met die gesimuleerde waarde van 7 dB. Die resultate wys dat die antenna sal funksioneer vir GPS toepassings.

Die algoritmes moes prakties getoets word deur van die TART teleskoop stelsel gebruik te maak. Dit is 'n 24 element radio teleskoop ontwerp vir die toets van beeldingsalgoritmes en soektogte na kort tyd gebeurtenisse. Die stelsel maak gebruik van die MAX2769B universele GPS ontvanger wat binêre monsters teen 'n monsterfrekwensie van 16.368 MHz by 'n senterfrekwensie van 4.092 MHz lewer.

Algoritme 1 gebruik 'n ortogonale ruimte bundelvormingstegniek om 'n nul te stuur in die rigting van 'n sterk steursein en ter selfde tyd antenna aanwys te plaas in die rigtings van belangrike seine. Algoritme 2 gebruik lineêr beperkte minimum variansie (LCMV) bundelvorming en 'n bundel soektog metode om beskikbare GPS seine te vind en steurseine te kanselleer. Beide algoritmes is sorgvuldig getoets deur parameters soos sein tot reis verhouding (SNR), sein tot steursein verhouding (SIR), integrasietyd, en die aantal analoog na digitale (ADC) bisse te verstel om te valingspunte te vind en sodoende vas te stel of dit 'n bruikbare algoritme is om te gebruik op die TART stelsel.

Dit is gevind dat, deur 'n piek soektog algoritme te gebruik om sukseskoers te bepaal, en SNR, SIR, integrasietyd en aantal ADC bisse te verstel, dat die Bartlett DOA algoritme die beste vaar, moontlik as gevolg van die aantal seine teenwoordig in die gesimuleerde samestelling.

Dit is getoon dat die sukseskoers van algoritmes 1 en 2 soortgelyk is, vir die parameters wat verstel is. Volgens die simulasies is albei metodes werkbare bundelvormers, vir GPS toepassings, op die TART stelsel en word nie beperk deur die stelsel hardeware nie. Algoritme 1 is meer sensitief vir variasies in SIR, SNR en integrasie tyd wanneer 1 bis geskuifde data gebruik word. Die kompleks bis geskuifde data is geskep deur van 'n tegniek gebruik te maak wat mens toelaat om 'n kwadratuur sein (Q) op te wek deur die in fase sein (I) met 1 bis te skuif. Dit is slegs moontlik as die monsterfrekwensie presies 4 maal die tussenfrekwensie (IF) is. Algoritme 1 benodig ook vooraf kennis van die aantal SOI's, wat nie nodig is vir algoritme 2 nie. Algoritme 2 is meer bewerkingsintensief in vergelyking met algoritme 1 vir GPS toepassings.

Voorlopige praktiese resultate was met die TART stelsel bekom en geanaliseer. Die metings toon potensiaal vir algoritme 1 om geïmplementeer te word, as bundelvormer, vir die TART stelsel, maar volledige toetse kon nie voltooi word nie omdat die stelsel onklaar geraak het.

# Acknowledgements

I would like to thank:

1. Rohacell for providing the HF71 foam used in the antenna fabrication, free of charge.
2. Prof. Dirk De Villiers for his wisdom and guidance throughout the project.
3. Mr. Wessel Croukamp for fabricating the microstrip antenna.
4. Mrs. Anneke Bester for guidance during the antenna measurements.
5. Dr. Danie Ludick for his work on the TART system.

# Abbreviations

<b>ADC</b>	Analogue to digital converter
<b>DBF</b>	Digital beamforming
<b>SNR</b>	Signal to noise ratio
<b>SFDR</b>	Spurious frequency dynamic range
<b>GPS</b>	Global positioning satellite
<b>FPGA</b>	Field programmable gate array
<b>SOI</b>	Signal of interest
<b>CW</b>	Continuous wave
<b>DOA</b>	Direction of arrival
<b>RFI</b>	Radio frequency interference
<b>RAM</b>	Random access memory
<b>RHCP</b>	Right hand circularly polarized
<b>AWGN</b>	Additive White Gaussian Noise
<b>BPSK</b>	Binary phase shift key
<b>ISR</b>	Interference to signal ratio
<b>SIR</b>	Signal to interference ratio
<b>SINR</b>	Signal to interference and noise ratio
<b>LNA</b>	Low noise amplifier
<b>ULA</b>	Uniform linear array
<b>IF</b>	Intermediate frequency



# Contents

<b>Declaration</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>Uittreksel</b>	<b>iv</b>
<b>Acknowledgements</b>	<b>vi</b>
<b>Abbreviations</b>	<b>vii</b>
<b>Contents</b>	<b>viii</b>
<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Thesis Objectives . . . . .	2
1.3 Thesis Outline and Contributions . . . . .	3
<b>2 GPS and Beamforming Background Literature</b>	<b>5</b>
2.1 GPS Signals . . . . .	5
2.2 GPS Signal Properties . . . . .	5
2.3 GPS Received Power . . . . .	6
2.4 GPS Receiver Structure . . . . .	7
2.5 GPS Signal Acquisition . . . . .	7
2.6 Beamforming . . . . .	10
2.6.1 NLMS . . . . .	11
2.6.2 Null Steering . . . . .	12
2.6.3 MVDR (Capon) . . . . .	12
2.6.4 Linear Constraint Minimum Variance (LCMV) . . . . .	13
<b>3 Interference and Mitigation Background Literature</b>	<b>16</b>
3.1 Assumptions . . . . .	16

3.2	Signal Model . . . . .	16
3.3	Orthogonal Subspace Projection . . . . .	18
3.4	Interference . . . . .	20
3.5	Threshold Detection . . . . .	21
3.6	Mitigation . . . . .	21
3.7	ADC Dynamic Range and Saturation . . . . .	21
<b>4</b>	<b>Antenna Design Background Literature</b>	<b>24</b>
4.1	Introduction . . . . .	24
4.2	Microstrip Feed Techniques . . . . .	26
4.3	Return Loss . . . . .	28
4.4	Polarization . . . . .	28
4.5	Axial Ratio Bandwidth . . . . .	30
4.6	Antenna Bandwidth . . . . .	30
4.7	Antenna Gain . . . . .	30
4.8	3 dB Beam Width . . . . .	31
4.9	Dual Band Antennas . . . . .	32
4.10	Wilkinson Combiner . . . . .	32
<b>5</b>	<b>Antenna Design Process</b>	<b>34</b>
5.1	Introduction . . . . .	34
5.2	Method . . . . .	34
5.2.1	Design . . . . .	34
5.2.2	Simulation . . . . .	36
5.3	Fabricated Antenna Results . . . . .	41
5.4	Conclusions from the Design Process . . . . .	45
<b>6</b>	<b>Testing Procedure</b>	<b>47</b>
6.1	Simulation Details . . . . .	47
6.2	TART . . . . .	49
6.3	Practical Data Collection . . . . .	49
6.3.1	Phase Calibration . . . . .	51
<b>7</b>	<b>DOA Algorithms</b>	<b>57</b>
7.1	MUSIC . . . . .	57
7.2	MVDR (Capon) . . . . .	58
7.3	Bartlett . . . . .	58
7.4	Simulation Results and Discussion . . . . .	59
<b>8</b>	<b>Proposed Algorithms</b>	<b>62</b>
8.1	Adaptive Beamforming Method . . . . .	62
8.2	Proposed Algorithm 1 . . . . .	62
8.2.1	Simulation Results and Discussion . . . . .	64
8.2.1.1	No Interference . . . . .	64

<i>CONTENTS</i>	<b>x</b>
8.2.1.2 Interference . . . . .	65
8.3 Proposed Algorithm 2 . . . . .	70
8.3.1 Simulation Results and Discussion . . . . .	73
8.4 Practical Results . . . . .	81
<b>9 Conclusions and Future Work</b>	<b>85</b>
9.1 Conclusions . . . . .	85
9.2 Future Work . . . . .	86
<b>Appendices</b>	<b>87</b>
<b>A Matlab codes</b>	<b>88</b>
A.1 DOA algorithms . . . . .	88
A.2 Beamforming algorithm 1 . . . . .	92
A.3 Beamforming algorithm 2 . . . . .	96
A.4 Phase calibration . . . . .	100
<b>List of References</b>	<b>105</b>

# List of Figures

2.1	L1 and L2 band GPS signal construction using C/A (1.023 Mbits/s) or P (10.23 Mbits/s) codes, the NAV (50 bits/s) message and L1 (1.57542 GHz) or L2 (1.2276 GHz) carrier. . . . .	6
2.2	Basic GPS receiver architecture. . . . .	7
2.3	A block diagram of the circular convolution acquisition process which results in a correlation plot of the C/A code replica with the incoming data. After squaring the absolute value, the data is passed to a threshold detection algorithm. . . . .	9
2.4	I/Q channel carrier removal process whose output is the incoming data referred to in Figure 2.3. . . . .	9
2.5	Classic adaptive beamforming architecture. . . . .	10
2.6	NLMS adaptive beamforming architecture. . . . .	11
2.7	Null steering beamforming architecture. . . . .	12
2.8	MVDR(Capon) adaptive beamforming architecture. . . . .	13
2.9	LCMV beamforming architecture. . . . .	14
3.1	Diagram defining $\theta$ and $\phi$ for the uniform linear array along the $x$ axis. . . . .	17
3.2	Block diagram of the subspace projection process which results in interference free input data. . . . .	19
3.3	Illustration of the orthogonal projection concept. . . . .	19
3.4	Diagram illustrating the concept of spurious frequency dynamic range (SFDR). . . . .	22
3.5	Illustration of ADC saturation when the input voltage is larger than the ADC saturation voltage and the effect of no AGC. . . . .	23
4.1	Basic microstrip layout. . . . .	25
4.2	Various microstrip feed techniques. . . . .	27
4.3	Various circular polarization techniques for microstrip patches. . . . .	29
4.4	Illustration of antenna directivity showing the directive antenna pattern in pink and the isotropic pattern in light blue. . . . .	31
4.5	Illustration of 3 dB beamwidth showing the beam pattern in green and the extent of the 3 dB angular range in blue. . . . .	31
4.6	The layout of a multi-layered aperture fed microstrip patch antenna. . . . .	32

4.7	Circuit diagram of the Wilkinson combiner used in the antenna design in Section 5 with the quarter wave length lines at $\sqrt{2}Z_0$ . . .	33
5.1	The final 3D layout of the feed network and aperture used on the antenna. Showing the phase of each arm coupling to the aperture, which is the cross shown in red, and which results in circular polarization of the antenna. . . . .	36
5.2	Simulated axial ratio (blue) and RHCP gain (red). . . . .	36
5.3	Simulated RHCP polar beampattern of the final antenna design at 1.22 GHz. . . . .	37
5.4	Simulated Cartesian beampattern of the final antenna design at 1.22 GHz, showing RHCP in red and LHCP in blue. . . . .	37
5.5	Simulated RHCP 3D beampattern of the final antenna design at 1.22 GHz. . . . .	38
5.6	Simulated RHCP polar beampattern of the final antenna design at 1.57 GHz. . . . .	38
5.7	Simulated Cartesian beampattern of the final antenna design at 1.57 GHz, showing RHCP in red and LHCP in blue. . . . .	39
5.8	Simulated RHCP 3D beampattern of the final antenna design at 1.57 GHz. . . . .	39
5.9	Final 3D layout of the dual band multi-layered microstrip patch antenna. . . . .	40
5.10	Simulated S-parameters of the final antenna design. . . . .	41
5.11	Fabricated version of the antenna designed in Chapter 5. . . . .	42
5.12	Comparison of measured (blue) and simulated (red) $S_{11}$ results of the dual band microstrip antenna designed in Chapter 5. . . . .	42
5.13	Measured and simulated beampatterns of the antenna shown in Figure 5.11 at 1.22 GHz. . . . .	43
5.14	Measured and simulated beampatterns of the antenna shown in Figure 5.11 at 1.57 GHz. . . . .	44
5.15	Measured RHCP gain (red) of the antenna shown in Figure 5.11 vs CST simulation gain (blue) . . . . .	45
5.16	Image showing the S-parameters that were measured while a faulty port was used, illustrating the importance of constructing an accurate port. . . . .	46
6.1	Illustration of the setup for practical data collection . . . . .	50
6.2	Image of the 12 element ULA, TART system and its power supply on the roof of the Stellenbosch University microwave laboratory. . .	50
6.3	Illustration of the far field region and the $22.5^\circ$ phase error that occurs at a distance of $R = 2D^2/\lambda$ . The image shows the ideal wave front, which is planar, and the actual wave front which is spherical. . . . .	52

6.4	Illustration of the clock synchronisation phase error that can occur when the mixers of each channel in an array are not perfectly synchronized. . . . .	53
6.5	Phase reference patterns (MUSIC (blue) and MUSICSmooth (red) spectra) used to create the cost function for the optimizer. The pattern was created by running the MUSIC and MUSIC smooth algorithms with simulated data known to arrive at the array elements simultaneously. . . . .	54
6.6	Patterns from the initial spectra, of the input data received from the TART system, prior to phase calibration. . . . .	55
6.7	Image of the results produced by using the phase values given by the optimisation. . . . .	55
7.1	Surface plots showing the results the DOA algorithm simulation-Integration time vs SNR. The simulation was run using 12 bits, as explained in Section 6. The plots show the success percentage of each algorithm. . . . .	59
7.2	Surface plots showing the results the DOA algorithm simulation-Number of ADC bits vs SNR. The integration time = 16000 samples. The plots show the success percentage of each algorithm. . . .	60
7.3	Graphs showing the results of the DOA algorithm simulation-I/Q & bit shift vs SNR. The integration time = 16000 samples. The plots show the success percentage of each algorithm. . . . .	61
8.1	Beam pattern produced by algorithm 1 (blue) vs LCMV (red) with no interference using 1 bit shifted data. The SOIs are at $60^\circ$ , $30^\circ$ , $-5^\circ$ , $-40^\circ$ . . . . .	64
8.2	A graph showing the beampattern produced by algorithm 1 (red), LCMV (black), nullsteering (blue) with interference. The SOIs are at $-60^\circ$ , $-30^\circ$ , $-5^\circ$ , $30^\circ$ and interference at $60^\circ$ . The algorithm was run using 1 bit shifted data . . . . .	65
8.3	A graph showing the beampattern produced by algorithm 1 (red), LCMV (black), nullsteering (blue) with interference. The SOIs are at $-60^\circ$ , $-30^\circ$ , $-5^\circ$ , $30^\circ$ and interference at $60^\circ$ . The algorithm was run using 12 bit shifted data . . . . .	66
8.4	Surface plot showing the success percent of algorithm 1, using 1 bit shifted data, while varying SIR and Integration time . . . . .	67
8.5	Surface plot showing the success percent of algorithm 1, using 1 bit shifted data, while varying SNR and SIR . . . . .	67
8.6	Image showing the success percent of all the simulations run with algorithm 1, which were detailed in Section 6. . . . .	68
8.7	Graphs showing the success percent of algorithm 1, with various parameter sweeps, checking the difference between I/Q and bit shifted data. . . . .	69

8.8	Diagram showing the 3 dB beam width of a 12 element array. . . .	71
8.9	Diagram of search space used for algorithm 2, showing the multiple beams searching from $-45^{\circ}$ - $45^{\circ}$ . Each beam has its own null at the interference, $-20^{\circ}$ , which is represented by its MUSIC spectrum. . .	72
8.10	A graph comparing the success percent of Algorithm 2, using 1 bit shifted data, with LCMV, beam steering and single antenna with SNR = -27 dB and varying SIR. . . . .	73
8.11	A graph comparing the success percent of Algorithm 2, using 1 bit shifted data, with LCMV, beam steering and single antenna with SNR = -23 dB and varying SIR. . . . .	74
8.12	Surface plot showing the success percent of algorithm 2, using 1 bit shifted data, while varying SIR and Integration time . . . . .	75
8.13	Surface plot showing the success percent of algorithm 2, using 1 bit shifted data, while varying SNR and SIR . . . . .	75
8.14	Image showing the success percent of all the simulations run with algorithm 2 using LCMV which were detailed in Section 6, Table 6.2.	77
8.15	Graphs showing the success percent of algorithm 2, with various parameter sweeps, checking the difference between I/Q and bit shifted data. . . . .	78
8.16	Image showing a side by side comparison of the success percent of all the simulations run with algorithm 2 using LCMV and algorithm 1, which were detailed in Chapter 6, Table 6.2. . . . .	80
8.17	DOA spectra of the practical data collected from TART. The test has two SOIs, each with the same power level. . . . .	81
8.18	DOA spectra of the practical data collected from TART. The test has two SOIs and the array has been shifted $\approx 8^{\circ}$ . . . . .	82
8.19	DOA spectra of the practical data collected from TART. The test has two SOIs, each with a different power level. . . . .	82
8.20	DOA spectra of the practical data collected from TART. The test has two SOIs, each with the same power level and the SOIs have been moved closer together. . . . .	83
8.21	DOA spectra of the practical data collected from TART. each with the same power level and the SOIs have been moved closer together. The resolution can be seen to be an issue with signals so closely spaced, for a 12 element ULA. . . . .	83

# List of Tables

6.1	Table showing the variation of parameters when testing the various DOA algorithms mentioned in Chapter 7. The results of these simulations can be seen in Chapter 8.4. . . . . .	48
6.2	Table showing the variation of parameters when testing algorithms 1 and 2, the results of which can be seen in Chapter 8.4. . . . .	48



# Chapter 1

## Introduction

### 1.1 Motivation

Global Positioning Systems (GPS) were originally developed by the US military. They make use of satellites as reference points to find a 3D coordinate position on earth. The distance can be found using very sensitive clocks to measure the time it takes for the signals from each satellite to reach the GPS receiver. The time is extracted through correlation of locally generated codes and the codes sent from the satellites. 3 satellites are used to find ones position and a fourth is used to remove code timing errors to allow for less expensive and sensitive clocks to be used in commercial GPS [1].

Nullsteering and beamforming antenna arrays are becoming an important part of navigation systems due to the advancement and increased use of signal jammers and increase in radio frequency interference (RFI). A null steering antenna array adaptively steers a beam pattern null in the direction of a signal jammer(s) to prevent corruption of the GPS signals and therefore loss of signal acquisition. The signal jammers operate in the L1 frequency band and due to the difference in power of the desired signal and jammer, the gain acquired from the spread spectrum GPS signal cannot prevent GPS from being easily jammed.

The beamforming array should track the signal jammer (i.e. drone) and continuously point a null in its direction while simultaneously creating beams in the directions of various GPS satellites to increase signal to noise ratio (SNR) and attenuate multipath.

Conventional beamformers require some prior information about the signals of interest (SOI) however the fact that the direction of incoming signals, Doppler, phase delays, array manifold or a reference signal may not always be available, creates the need for interference mitigation and beamforming without this information, otherwise known as blind beamforming.

Digital beamforming (DBF) is widely used and most of these systems are complex and have high cost. Lowering the cost of a DBF system requires

implementing a lower resolution analogue to digital converter (ADC), fewer number of antennae or a less expensive digital back end. The transient array radio telescope (TART) system makes use of 1 bit in-phase (I) output data at a centre frequency of 4.092 MHz with a sample rate of 16.368 MHz. It uses 24 commercial ceramic GPS patch antennae and MAX2769B receivers in a low cost and simple system created for testing imaging algorithms and surveying transient events. Implementing blind beamforming algorithms with simple hardware, such as the TART system, is important because access to complex hardware in engineering projects is generally limited due to financial constraints.

## 1.2 Thesis Objectives

Digital beamformers require fairly complex hardware and the hardware quickly increases in complexity when more antennas are used in an array due to the large increase in the amount of hardware used and data throughput. Field-programmable gate arrays (FPGA) are becoming more widely used in digital backends for a variety of functions [2]. The FPGA and ADC are two of the most important components in a DBF system. The ADC has design features such as the SNR, spurious frequency dynamic response (SFDR), sample rate and cost that must be addressed when implementing into a digital system and FPGA's have parameters such as random access memory (RAM), processing speed, number of gates and cost that determine how effective they are for a specific application. The ADC and FPGA must be chosen so that the FPGA can handle the amount of data from the ADC output. Many FPGA's have integrated ADC's that allow them to be used as a single chip.

The TART system uses very simple/cost effective hardware which outputs 1 bit I data and the quadrature (Q) data is formed through bit shifting the I data 1 bit. This is possible because the sample frequency is 4 times that of the intermediate frequency (IF) i.e.  $4f_{IF} = f_{sample}$  therefore each bit is  $90^\circ$  of the IF data. A Matlab model of this system was created with automatic gain control (AGC) that allows for voltage levels at the ADC of  $\pm 1$  V. The ADC resolution is changed to investigate if the TART system can be used for digital beamforming.

A dual band antenna for L1 and L2 GPS frequency bands has been designed and measured to investigate its efficacy in the desired antenna array.

The thesis will also investigate the hardware limitations of a few direction of arrival (DOA) algorithms as well as two blind interference suppressing beamforming algorithms. Various parameters of the receiver, such as number of ADC bits and use of the I/Q channels, will be swept and the results compared to determine the failure point of these algorithms. Typical GPS receivers, such as the MAX2769B which is used in TART, allow for a low number of ADC bits therefore lowering the data throughput and the use of I and Q data requires

more physical connections. If the SNR and SIR are not high enough then the receiver sensitivity will have to be high, meaning a more complex costly system. These and the other variables chosen to be swept represent a significant impact on the system complexity and cost.

A GPS signal model is simulated for the SOI's as well as a continuous wave (CW) jammer at L1 frequencies.

**Goals:**

1. Develop blind beamforming algorithms
2. Develop and test a GPS antenna
3. Investigate hardware limitations of DOA and blind beamforming algorithms

## 1.3 Thesis Outline and Contributions

The thesis sections and their contributions are as follows:

- Chapter 2 gives background information on the basics of GPS including the received signal power, receiver architecture and signal properties. Various existing beamforming algorithms are investigated and information on analogue to digital conversion is given.
- Chapter 3 gives fundamental information on microstrip antennae.
- Chapter 4 presents the design of a GPS microstrip antenna.
  - Contribution : Design and testing of a multi-layered GPS microstrip antenna.
- Chapter 5 provides information on orthogonal projection and the DOA algorithms that will be tested. This chapter also gives the details of the proposed blind beamforming techniques.
  - Contribution : Proposal and modelling of 2 blind beamforming techniques.
- Chapter 6 provides details of the setup used for practical data collection as well as a few important considerations when using hardware.
- Chapter 7 is an analysis of the results of the simulations.
  - Contribution : Determining the hardware limitations of the proposed beamforming and DOA algorithms.

- Chapter 8 presents the conclusions drawn from the results of the simulations.

## Chapter 2

# GPS and Beamforming Background Literature

This chapter provides necessary background information on the fundamental principles used in this thesis. First a broad overview of GPS systems and signals will be presented, followed by beamforming literature.

### 2.1 GPS Signals

GPS signals are right hand circularly polarized (RHCP) binary phase shift keying (BPSK) signals. There are the bands L1, L2, and L5 which are at 1.57542 GHz, 1.2276 GHz and 1.17645 GHz respectively. This thesis will only be concerned with the L1 band, however the antenna designed during the course of this thesis is a dual band GPS antenna and therefore is designed to function at the L1 and L2 bands. The orthogonality of the received signal structure is what enables the use of subspace techniques for interference mitigation and beamforming.

### 2.2 GPS Signal Properties

As shown below in Figure 2.1, the GPS signal consists of three components namely the carrier, course acquisition (C/A) code and navigation message. The L1 band consists of the carrier, course acquisition or pseudorandom noise (PRN) code and the navigation (NAV) message. The period of the NAV message code is far longer than the C/A code, therefore for this thesis the NAV message is assumed to be 1. The precision (P) code is used by the military and allows for a very fine position lock while civilian GPS only makes use of the C/A code which generally gives a position lock with approximately a 1 m accuracy [3].

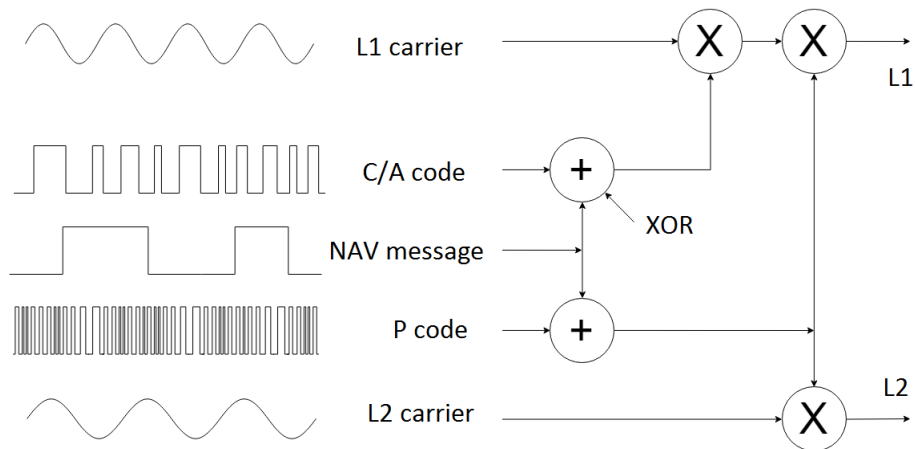


Figure 2.1: L1 and L2 band GPS signal construction using C/A (1.023 Mbit-s/s) or P (10.23 Mbits/s) codes, the NAV (50 bits/s) message and L1 (1.57542 GHz) or L2 (1.2276 GHz) carrier.

The C/A code has two main properties namely its use as a direct sequence spread spectrum (DS-SS) signal and its correlation properties to other C/A codes. The first property is used to spread the power of the GPS signal over a frequency band that is far wider than the bandwidth required for the information and therefore giving a processing gain of  $10\log\left\{\frac{\text{Chip rate}}{\text{Data rate}}\right\} = 10\log\left\{\frac{1.023 \times 10^6}{50}\right\} = 43.1$  dB. A narrowband interference signal will undergo the same spreading as the signal of interest however it will not be despread. This means that the energy of the interference signal is spread over a large bandwidth. The processing gain comes after band pass filtering using the same bandwidth of the despread desired signal. The signal gain occurs because only a small amount of the noise and interference is passed by the filter. The second property is used to allow for a unique choice between different satellites signals, as each satellite has its own C/A code. The C/A codes have a low cross correlation value between each other and also between time shifted versions of themselves.

## 2.3 GPS Received Power

Although the spreading code allows the power of the GPS signal to be spread over the C/A code bandwidth of 2.046 MHz, the amount of received power from GPS signal at the earth's surface is still very small, at least 20 dB below the thermal noise floor, for a typical GPS receiver. The signal strength at the receiver is typically -122 dBm [4] and will vary with factors such as hardware noise temperature, antenna array gain and satellite position. The signal power can be as low as -130 dBm while the thermal noise floor is between -110 dBm & -115 dBm giving a typical SNR of  $\approx -20$  dB.

## 2.4 GPS Receiver Structure

The general structure of a GPS receiver consists of an analogue front end containing LNA's, mixers and filters, followed by an ADC and digital backend. The GPS signal is received by the antenna's and amplified with very low noise (0.5 - 1 dB) by 30 - 50 dB. The signal is then mixed down to the intermediate frequency (IF) of 4.092 MHz by the mixer and the higher frequencies are filtered out by a 5th order, 2 MHz bandwidth, Butterworth band pass filter centred at the IF. After filtering, the lower frequency signal can be digitized by the ADC, and AGC is used to ensure the ADC introduces minimal noise (see Section 3.7). After digitization the signal is passed to the digital backend for processing involving beamforming weight calculations or GPS signal acquisition. A block diagram of the process can be seen below in Figure 2.2. A circuit diagram of the MAX2769 universal GPS receiver chip can be seen in [5].

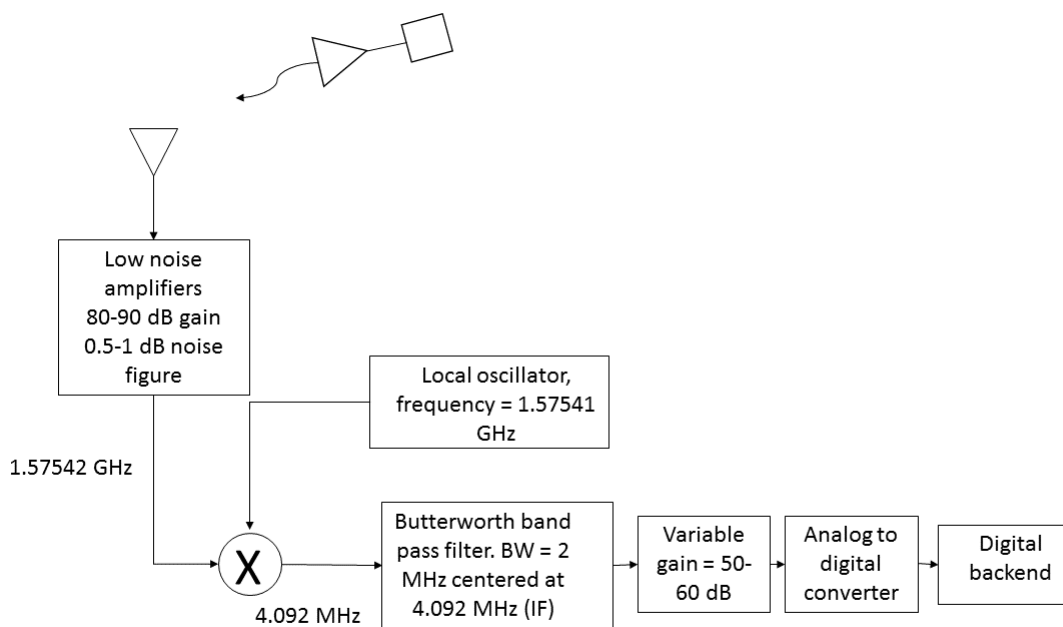


Figure 2.2: Basic GPS receiver architecture.

## 2.5 GPS Signal Acquisition

A GPS position lock initially begins with signal acquisition. Of course the signal must be detected before decoding can happen. The acquisition process determines a rough estimate of the signal code phase and carrier frequency which is related to distance from the antenna and the movement of the receiver

and transmitter. This is done by generating a local replica of the incoming code and searching in 3 dimensions. The search determines the satellite pseudo-random noise (PRN) number, C/A code phase and carrier frequency. The estimates of the frequency and phase are then used by the tracking loops of the receiver. The code phase shift is an estimate of the time it takes for the signal to travel from the satellite to the receiver. Knowing that the signal propagates at the speed of light allows one to determine the distance that the signal travelled.

The carrier frequency shift is caused by a number of different factors such as clock temperature of the receiver and the movement of the satellites and receiver. The largest effect of the frequency is caused by the movement of the satellite which can cause a Doppler shift of  $\pm 5$  kHz. A common search space for the Doppler shifts is  $\pm 10$  kHz with 500 Hz steps or bins [6] [7][8].

The search can be done in a number of ways. The classic method is to search through each Doppler frequency bin and code phase shift individually, however this method is very computationally inefficient, therefore circular convolution is used, which accomplishes the code phase search steps in parallel. The process can be shown mathematically [7] by,

$$R[m] = x[n] \otimes CA[-n] = \|\mathcal{F}^{-1}\{\mathcal{F}\{x[n]\}\mathcal{F}\{CA[n]\}^*\}\|^2, \quad (2.1)$$

where  $R$  is the output correlation between the received and replica C/A codes.  $x$  is the incoming data and  $CA$  is the replica C/A code. The expression above shows that the incoming data, after beamforming and removing the carrier, is passed through an FFT along with the conjugate of the C/A code to be tested. These two data sets are multiplied and inverse Fourier transformed. The absolute value of this data is squared which results in a correlation plot of the incoming data and each bit shift of the replica C/A code. The peaks in this plot are checked against a threshold level to determine signal acquisition, at which point the tracking process takes over. A block diagram of the acquisition process can be seen below in Figure 2.3.



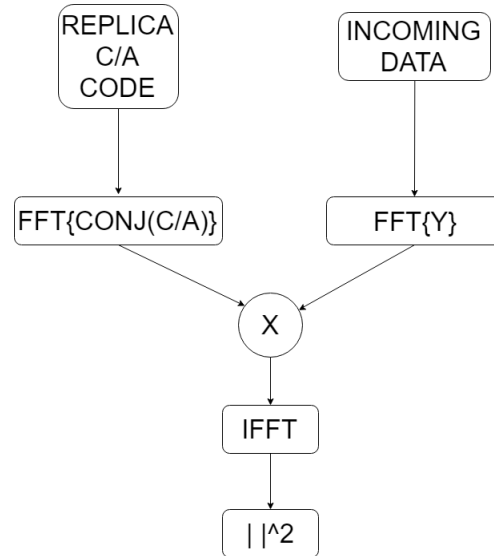


Figure 2.3: A block diagram of the circular convolution acquisition process which results in a correlation plot of the C/A code replica with the incoming data. After squaring the absolute value, the data is passed to a threshold detection algorithm.

The carrier frequency is removed by multiplying the incoming data by a carrier frequency of  $f_{IF} + f_D$  where  $f_D$  is the Doppler frequency search bin and  $f_{IF}$  is the intermediate frequency of the signal. I and Q channels are formed by multiplying the data by cosine and sine signals respectively, each with the frequency mentioned above. The Q channel is multiplied by the imaginary unit  $j$  and the two are summed to create  $I + jQ$  which is the complex incoming data to be passed to the FFT in Figure 2.3. A block diagram of the process can be seen in Figure 2.4.

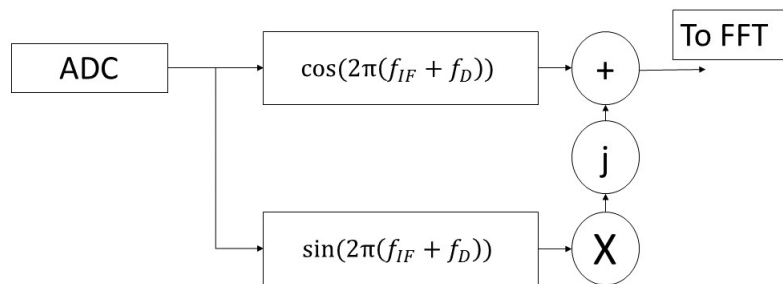


Figure 2.4: I/Q channel carrier removal process whose output is the incoming data referred to in Figure 2.3.

The process of signal acquisition is done after beamforming to ensure the highest possible SINR. The next section gives basic details on some common beamforming algorithms.

## 2.6 Beamforming

Beamforming is a technique where by the incoming signals from an antenna array are given complex weights and summed. The complex weights can be calculated in a number of ways, some of which are detailed in this section.

This process increases signal gain and decreases multipath and RFI far better than a single antenna because one makes use of spatial filtering to create nulls and beams in specific directions which cannot be done with a single antenna. Beamforming can be used in a static environment which is conventional beamforming and a dynamic environment which is termed adaptive beamforming. A block diagram detailing the process of an adaptive beamformer is shown below in Figure 2.5.

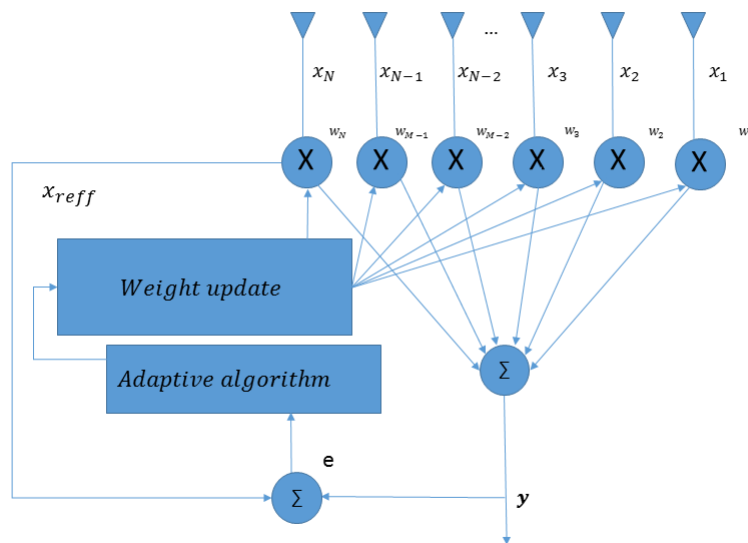


Figure 2.5: Classic adaptive beamforming architecture.

Conventional beamforming applies fixed weights to place beams in known, fixed locations. GPS beamforming requires adaptive beamforming due to the movement of the satellites in orbit and general movement of GPS receivers on earth. This method of beamforming applies varying weights to the antenna elements according to the properties of the incoming signal. The beam pattern of the array adapts for the movement of the transmitter, receiver and potential jammers in the receiver's field of view.

There have been many studies into adaptive beamforming through the control of the phase of the excitation of the antenna array elements. The methods include sequential quadratic programming (SQP) [9] and multi-objective optimisation, genetic algorithms (GA's) [10], particle swarm optimization (PSO)

[11] and hybrid methods to find the complex weights. This thesis has investigated a few simple algorithms such as normalized least mean square (NLMS), minimum variance distortionless response (MVDR), linearly constrained minimum variance (LCMV) and null steering.

### 2.6.1 NLMS

The NLMS is an extension of the least mean square (LMS) algorithm and was developed in 1960 by Widrow and Hoff. It is an iterative algorithm that updates beamforming weights, using gradient descent, to converge to the optimum which is a solution that gives a minimum value of the cost function. The cost function is the mean squared error between the reference signal and the combination of the updated weights and the input signal,  $\min \{|e(k)|^2\}$  where  $e(k) = x_{ref}(k) - \bar{\mathbf{w}}^H \bar{\mathbf{x}}$  [12]. The algorithm is executed as shown below and in Figure 2.6.

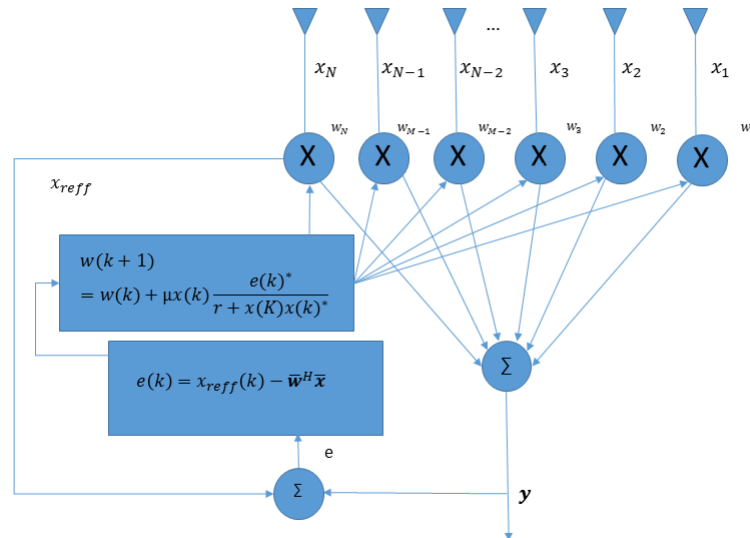


Figure 2.6: NLMS adaptive beamforming architecture.

$w$  is the weight vector,  $\mu$  is the step size,  $x_{ref}$  is the incoming data,  $e$  is the iteration error and  $r$  is a safety factor which is a small positive constant. The  $\cdot^H$  operator denotes the Hermitian transpose. The convergence time of the LMS algorithm is dependent on the step size  $\mu$  and the input data. The NLMS algorithm is a more robust version of the LMS algorithm because its convergence rate is less dependent to the step size used and SNR. The convergence rate for NLMS is faster than that of LMS [12].

## 2.6.2 Null Steering

Null steering is similar to beamforming however rather than placing a beam in the direction of an SOI, it places a null in the direction of an interferer and maintains uniformity in all other directions. Nulls can be placed conventionally (statically) or adaptively and the algorithm makes use of a reference signal and an incoming data covariance matrix. The reference signal is cross correlated with the incoming data which is combined with the covariance matrix of the incoming data to get the optimal weights. A block diagram of the process can be seen below in Figure 2.7.

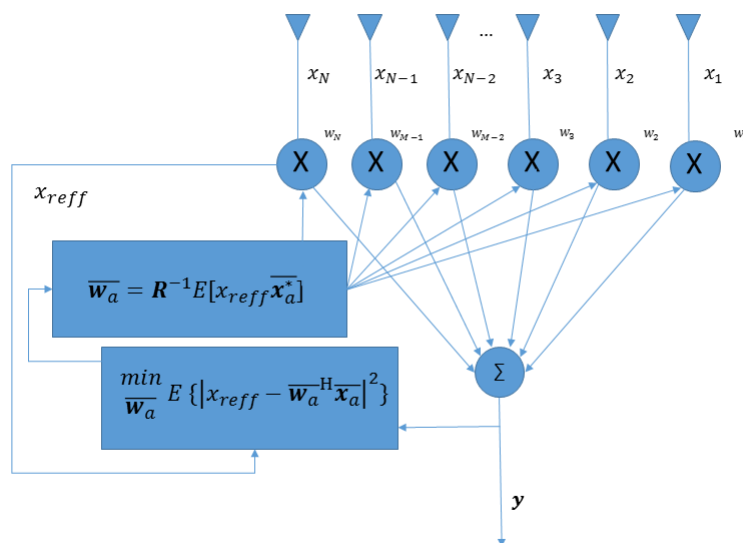


Figure 2.7: Null steering beamforming architecture.

$\mathbf{R}$  is the covariance matrix of the incoming data,  $\bar{\mathbf{x}}_a$  is the incoming data vector and  $\bar{\mathbf{w}}_a$  is the weight vector.

## 2.6.3 MVDR (Capon)

This beamforming technique aims to minimize the output power of the array while placing unity gain in the direction of an SOI. Multiple SOI's can be searched for when using multi-objective optimisation techniques [9]. The output of the antenna array can be represented by:

$$y(t) = \sum_{n=1} w_n^H x_n(t), \quad (2.2)$$

which can be written in vector form as:

$$\bar{\mathbf{Y}} = \bar{\mathbf{w}}^H \mathbf{X}, \quad (2.3)$$

with  $\bar{\mathbf{Y}} \in \mathbb{C}^{1 \times K}$ ,  $\bar{\mathbf{w}} \in \mathbb{C}^{N \times 1}$  and  $\mathbf{X} \in \mathbb{C}^{N \times K}$  with  $N$  antenna elements and  $K$  samples.

The output power of the array is given by:

$$E \{ \bar{\mathbf{Y}}^2 \} = \bar{\mathbf{w}}^H \mathbf{R} \bar{\mathbf{w}} \quad (2.4)$$

with  $\mathbf{R}$  being the covariance matrix of  $\mathbf{X}$  and  $\mathbf{R} \in \mathbb{C}^{N \times N}$ . The algorithm objective can be written mathematically as a constrained minimization problem:

$$\min \{ \bar{\mathbf{w}}^H \mathbf{R} \bar{\mathbf{w}} \} \text{ subject to } \bar{\mathbf{w}}^H \bar{\mathbf{a}} = 1, \quad (2.5)$$

where  $\bar{\mathbf{a}} \in \mathbb{C}^{1 \times N}$  is a steering vector corresponding to a desired signal's DOA. The solution to the above problem is the weight vector:

$$\bar{\mathbf{w}} = \frac{\mathbf{R}^{-1} \bar{\mathbf{a}}}{\bar{\mathbf{a}}^H \mathbf{R}^{-1} \bar{\mathbf{a}}}. \quad (2.6)$$

A block diagram of the MVDR algorithm can be seen below in Figure 2.8. The structure of the beamformer is very similar to the LCMV beamformer however as seen from (2.5), the constraint only allows for a signal from 1 direction to not be reduced.

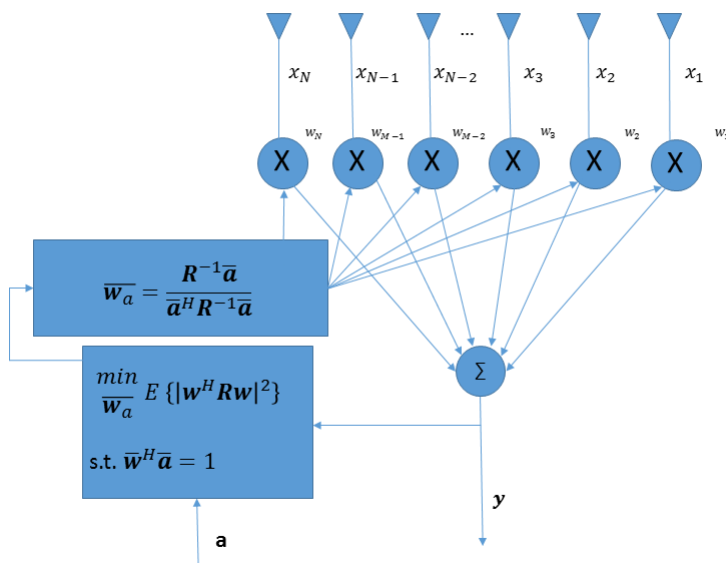


Figure 2.8: MVDR(Capon) adaptive beamforming architecture.

#### 2.6.4 Linear Constraint Minimum Variance (LCMV)

The LCMV algorithm aims to lower the total output energy of the system while subject to certain constraints. The constraints are usually unity gain

## CHAPTER 2. GPS AND BEAMFORMING BACKGROUND LITERATURE 14

the direction of SOI's and nulls in the directions of interference. The output power of the beamformer is given by (2.4) and the constraints are defined as follows:

$$\mathbf{S}^H \bar{\mathbf{w}} = \bar{\mathbf{g}}, \quad (2.7)$$

where  $\mathbf{S}$  is a matrix of the steering vectors of the signals and  $\bar{\mathbf{g}}$  is a vector of constraints.

$$e^{-j\bar{\mathbf{k}}\bar{\mathbf{r}}} \quad (2.8)$$

$$\begin{bmatrix} e^{-j\bar{\mathbf{k}}_1\bar{\mathbf{r}}} \\ e^{-j\bar{\mathbf{k}}_2\bar{\mathbf{r}}} \\ e^{-j\bar{\mathbf{k}}_3\bar{\mathbf{r}}} \end{bmatrix} \bar{\mathbf{w}} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \quad (2.9)$$

$\bar{\mathbf{k}}_1$  and  $\bar{\mathbf{k}}_2$  are from desired signals and  $\bar{\mathbf{k}}_3$  is from an interferer. The solution to the above problem is:

$$\bar{\mathbf{w}} = \mathbf{R}^{-1} \mathbf{S} (\mathbf{S}^H \mathbf{R}^{-1} \mathbf{S})^{-1} \bar{\mathbf{g}} \quad (2.10)$$

A simple block diagram of the algorithm can be seen below in Figure 2.9.

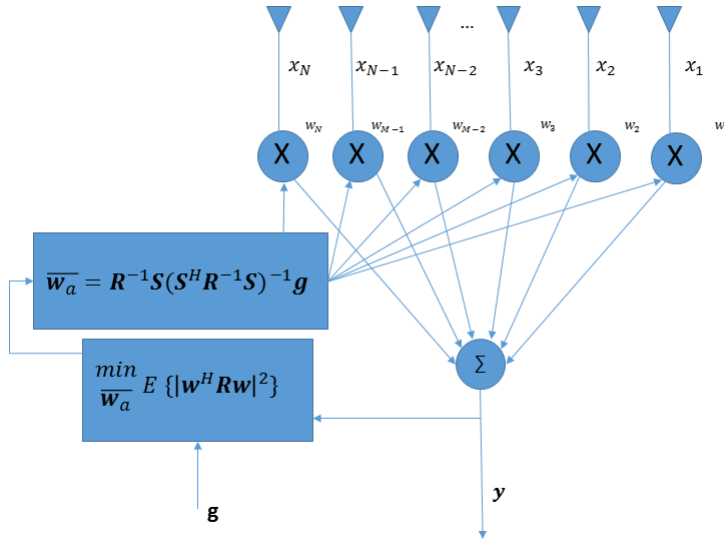


Figure 2.9: LCMV beamforming architecture.

The LCMV and MVDR algorithms do not require a reference signal however they do require the DOA of the signals to be known.

The LCMV algorithm is a generalized version of the MVDR and can process  $N$  constraints as opposed to one for MVDR. Both beamformers minimize the total output energy of the array. However the LCMV beamformers attempts to null the interference while MVDR attempts to optimally lower the total SINR therefore  $\text{SIR}_{MVDR} < \text{SIR}_{LCMV}$  and  $\text{SNR}_{MVDR} > \text{SNR}_{LCMV}$ . MVDR

*CHAPTER 2. GPS AND BEAMFORMING BACKGROUND LITERATURE* **15**

has better performance than LCMV for when using collinear signals [13]. The NLMS algorithm is an improved version of the LMS algorithm with better convergence rates and speeds and ability to process larger noise variance. The computational complexity for both these algorithms is low. All of the algorithms perform poorly at processing coherent interference signals however adaptations have been made to compensate for coherence [14]. It has been shown that the speed of the LMS algorithm is better than that of LCMV however the convergence rate is much lower [15]. As the SNR decreases the difference in bit error rate between the LMS and LCMV algorithms becomes negligible.

## Chapter 3

# Interference and Mitigation Background Literature

This chapter provides a brief background information on interference and mitigation techniques. The assumptions and signal model will precede information on the orthogonal subspace projection technique, followed by different types of interference sources. Details on threshold detection will precede information on mitigation of interference and its effect on ADCs.

### 3.1 Assumptions

The following assumptions were made for the antennae and array.

1. The GPS, interference and noise signals are all mutually uncorrelated.
2. The antennae are immersed in free space and atmospheric effects on GPS signals are ignored.
3. Transmit sources are in antenna far field therefore the incoming waves can be considered plane.
4. Antennae are identical isotropic receivers in simulation.
5. The effects of mutual coupling of the antennae are ignored.

### 3.2 Signal Model

The  $L$  incoming GPS signals,  $M$  interference signals and noise that reach the antenna array can be represented as:

$$x(t) = \sum_{i=1}^L a_{li}s_l(t) + \sum_{q=1}^M a_{mq}i_m(t) + n(t), \quad (3.1)$$



where  $a_{li}$  represents the steering vectors of the  $L$  incoming desired signals,  $s_l(t)$ ,  $a_{mq}$  represents the steering vectors of the  $M$  interference signals,  $i_m(t)$ .  $n(t)$  is assumed to be Gaussian white noise.

$$s_l(t) = Ax(t - \tau)\cos\{2\pi(f_L + f_D)t + \varphi\}, \quad (3.2)$$

$A$  is the signal amplitude,  $x(t - \tau)$  is the delayed C/A code,  $f_L$  is the  $L$  carrier frequency of 1.57542 GHz,  $f_D$  is the Doppler frequency caused by movement of the satellites and  $\varphi$  is the initial phase of the signal. The signal,  $x(t)$ , can be represented as a matrix  $\mathbf{X} \in \mathbb{C}^{N \times K}$ , where  $K$  is the number of sample points and  $N$  is the number of antenna elements, and (3.1) can be written in vector form as:

$$\mathbf{X} = \mathbf{A}_l \mathbf{S} + \mathbf{A}_m \mathbf{I} + \mathbf{N}. \quad (3.3)$$

where  $\mathbf{A}_l \in \mathbb{C}^{N \times L}$  is the steering vector matrix of the GPS signals on the array:

$$[\bar{\mathbf{a}}_{l1}(\theta_1, \phi_1) \quad \bar{\mathbf{a}}_{l2}(\theta_2, \phi_2) \quad \dots] \quad (3.4)$$

$$\bar{\mathbf{a}}_{l1}(\theta_1, \phi_1) = e^{-j\bar{\mathbf{k}}_l \cdot \bar{\mathbf{r}}}. \quad (3.5)$$

The steering vector above  $\bar{\mathbf{a}}_{l1}(\theta_1, \phi_1) \in \mathbb{C}^{1 \times N}$  is due to the  $l^{\text{th}}$  signal at the  $i^{\text{th}}$  antenna element.  $(\theta_k, \phi_k)$  are represented in Figure 3.1. The ULA is orientated along the  $x$  axis therefore  $\phi_k$  is assumed to be  $0^\circ$ .

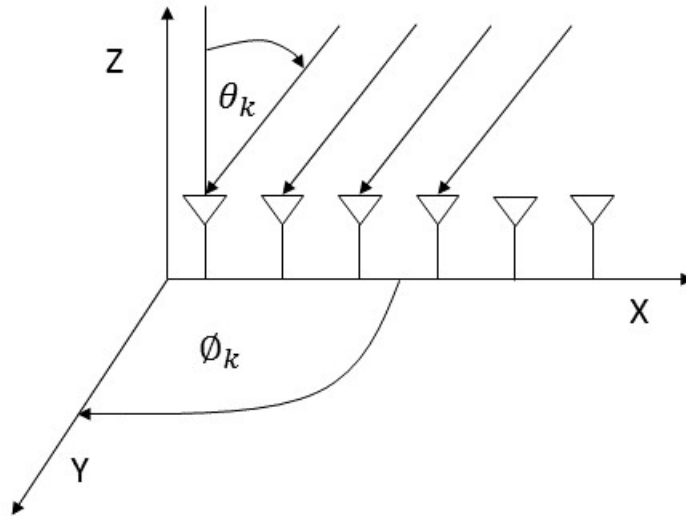


Figure 3.1: Diagram defining  $\theta$  and  $\phi$  for the uniform linear array along the  $x$  axis.

$\mathbf{S} \in \mathbb{C}^{L \times K}$  is the vector matrix of incoming GPS signals:

$$\begin{bmatrix} s_1(1) & s_2(1) & \dots \\ s_1(2) & s_2(2) & \dots \\ \vdots & \vdots & \ddots \end{bmatrix} \quad (3.6)$$

$\mathbf{A}_m \in \mathbb{C}^{N \times M}$  is the steering vector matrix of the interference signals on the array:

$$[\bar{\mathbf{a}}_{m1}(\theta_1, \phi_1) \quad \bar{\mathbf{a}}_{m2}(\theta_2, \phi_2) \quad \dots] \quad (3.7)$$

$\mathbf{I} \in \mathbb{C}^{M \times K}$  is the steering vector matrix of incoming interference signals:

$$\begin{bmatrix} i_1(1) & i_2(2) & \dots \\ i_1(1) & i_2(2) & \dots \\ \vdots & \vdots & \ddots \end{bmatrix} \quad (3.8)$$

$\mathbf{N} \in \mathbb{C}^{N \times K}$  is a matrix of noise values.

The covariance matrix,  $\mathbf{R}$ , of the signal  $\mathbf{X}$  is:

$$\mathbf{R} = E[\mathbf{X}\mathbf{X}^H] \in \mathbb{C}^{N \times N} \quad (3.9)$$

or using estimation with a limited number of samples,  $K$ :

$$\mathbf{R} = \frac{1}{K} \sum_{k=1}^K \mathbf{X}(k)\mathbf{X}(k)^H. \quad (3.10)$$

### 3.3 Orthogonal Subspace Projection

Subspace projection of incoming data to remove interference is a blind beamforming technique. The method uses the orthogonal signal properties of the interference and SOI+noise to create a subspace orthogonal to the interference and project the incoming data onto this subspace, effectively nulling the interference. A block diagram of the process can be seen below in Figure 3.2. The Eigenvalues of  $\mathbf{R}_{xx}$  are  $\lambda_k, k = 1, 2, 3 \dots N$  and if  $\lambda_1 > \lambda_2 > \lambda_3 \dots$ , then  $\bar{\mathbf{e}}_1$  is the Eigenvector of the largest signal in  $\mathbf{R}_{xx}$  and  $\mathbf{U}$  is partitioned with  $M$  interference signal Eigenvectors,  $\mathbf{U} = [\bar{\mathbf{e}}_1, \bar{\mathbf{e}}_2, \bar{\mathbf{e}}_3 \dots \bar{\mathbf{e}}_M] \in \mathbb{C}^{(M) \times N}$ . The Eigenvectors  $\bar{\mathbf{e}}_k, \bar{\mathbf{e}}_j$  are orthogonal to each other for  $k, j = 1, 2 \dots N$  and  $k \neq j$ .  $\mathbf{I}$  in Figure 3.2 is the identity matrix.  $\mathbf{V}$  is the projection matrix that is orthogonal to  $\mathbf{U}$  given by  $\mathbf{V} = \mathbf{I} - \mathbf{U}\mathbf{U}^\dagger$  where  $\cdot^\dagger$  is the Moore-Penrose inverse.  $\tilde{\mathbf{X}}$  is interference free data.

The subspace beamforming method is given in Section 8.2 and only uses two more steps to find the weights. The method uses the orthogonal subspace created to find weights that allow for beams in the directions of the SOI's as well as a null in the direction of a potential interferer.

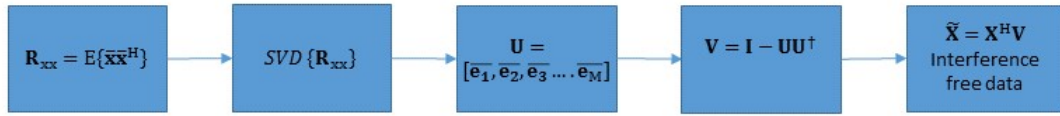


Figure 3.2: Block diagram of the subspace projection process which results in interference free input data.

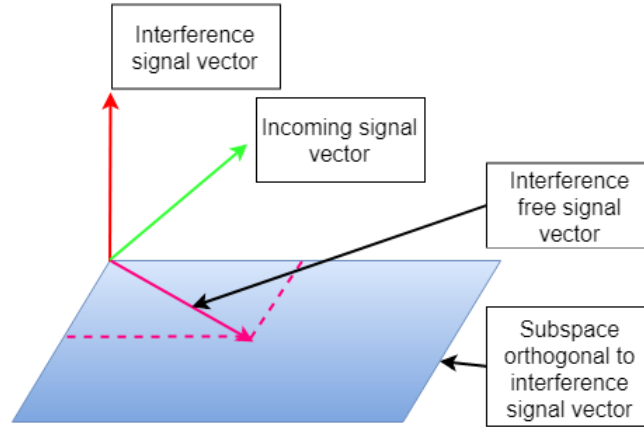


Figure 3.3: Illustration of the orthogonal projection concept.

Figure 3.3 illustrates the concept of interference mitigation through orthogonal projection of the incoming signals.

$$\mathbf{R} = E[\mathbf{X}\mathbf{X}^H] = \mathbf{R}_i + \mathbf{R}_l + \mathbf{R}_n = \sum_{i=1}^M \lambda_i \bar{\mathbf{e}}_i \bar{\mathbf{e}}_i^* + \sum_{j=M+1}^N \lambda_j \bar{\mathbf{e}}_j \bar{\mathbf{e}}_j^* + \sigma_n^2 \mathbf{I} \quad (3.11)$$

$\sum_{i=1}^M \lambda_i \bar{\mathbf{e}}_i \bar{\mathbf{e}}_i^*$  represents the interference Eigenvectors and values and  $\sum_{j=M+1}^N \lambda_j \bar{\mathbf{e}}_j \bar{\mathbf{e}}_j^*$  represents the SOI+noise Eigenvectors and values. The Eigenvectors above,  $\bar{\mathbf{e}}_i$  and  $\bar{\mathbf{e}}_j$ , are the same as seen in the  $\mathbf{U}$  matrix in Section 3.3.  $\sigma^2$  is the variance, or power, of the background noise. Due to the magnitude of the power of the interference signals compared to the noise and GPS signals, the covariance matrix is dominated by the Eigenvalues of the interference.

After singular value decomposition (SVD) of  $\mathbf{R}$ , a subspace,  $\mathbf{V} \in \mathbb{C}^{(N-M) \times N}$  can be created that is orthogonal to the interference subspace and the input data is projected onto the subspace as mentioned above. The SVD is performed because  $\mathbf{R}$  is a positive-definite matrix ( $\bar{\mathbf{z}}^H \mathbf{R} \bar{\mathbf{z}} > 0$  for all  $\bar{\mathbf{z}} \in \mathbb{C}^{1 \times N}$ ) and the

SVD and Eigen decomposition are the same.

$$\mathbf{V} = \mathbf{I} - \mathbf{U}\mathbf{U}^\dagger, \quad (3.12)$$

where  $\mathbf{U} = [\bar{\mathbf{e}}_1, \bar{\mathbf{e}}_2, \bar{\mathbf{e}}_3 \dots \bar{\mathbf{e}}_M] \in \mathbb{C}^{(M) \times N}$  is a matrix of the stacked Eigenvectors that correspond to the SOI+noise,  $\mathbf{I} \in \mathbb{Z}^{N \times N}$  is the identity matrix, and the  $\dagger$  operator is the Moore-Penrose pseudoinverse i.e.  $\mathbf{U}^\dagger = (\mathbf{U}^H \mathbf{U})^{-1} \mathbf{U}^H$ . The  $^H$  operator is the Hermitian transpose.

The new input data matrix, that is interference free is  $\tilde{\mathbf{X}}$ .

$$\tilde{\mathbf{X}} = \mathbf{X}^H \mathbf{V}, \quad (3.13)$$

or

$$\tilde{\mathbf{X}} = [\mathbf{A}_s \bar{\mathbf{S}} + \mathbf{A}_i \bar{\mathbf{I}} + \bar{\mathbf{N}}]^H \mathbf{V}, \quad (3.14)$$

and because the interference is orthogonal to the new created subspace i.e.  $\mathbf{A}_i \bar{\mathbf{I}}^H \mathbf{V} = 0$ ,

$$\tilde{\mathbf{X}} = [\mathbf{A}_s \bar{\mathbf{S}} + \bar{\mathbf{N}}]^H \mathbf{V}. \quad (3.15)$$

Of course this method requires the number of interference signals to be determined. This can be accomplished using the methods mentioned in Section 3.5

### 3.4 Interference

RFI can be generated from intentional or unintentional sources. Unintentional RFI radiates from sources like industrial machinery, cellphones and radio towers. Intentional sources include military naval, land and air vehicles. Signal jammers however don't last long in modern warfare due to the ability of opposing armies to locate the source and send a counter measure almost instantaneously.

RFI can corrupt any communications system and render it unusable. For GPS systems this means a loss of signal acquisition or a degradation of position accuracy, depending on signal to interference ratio (SIR). Interference sources can be either wideband or narrow band which is a measure of the signal bandwidth relative to the GPS bandwidth.

There are many types of interference sources for GPS signals and due to the low power of the SOI, it is susceptible to these sources. Some of the sources seen are:

1. Wideband Gaussian noise
2. Wideband spread spectrum
3. Narrow band continuous wave
4. Narrow band swept continuous wave

It takes approximately 20-30 dB of interference power to jam the GPS acquisition process [16].

## 3.5 Threshold Detection

An important part of blind beamforming is the ability to detect and distinguish interference signals, particularly when using orthogonal projection and subspace techniques. There are various ways to accomplish this such as the SVD of covariance matrix and checking power levels against a threshold[6]. Another method is to measure the L2 norm of the input signal and check if the level is above the noise floor, if so, an interference signal is most likely transmitting. The second method however does not identify the number of interference signals present. Performing an Eigenvalue decomposition of the  $R$  matrix and checking to see the number of Eigenvalues that are approximately equal is also a viable method for threshold detection in the GPS environment [17]. If, for instance, there are 12 antennae and two interference signals, there should be 10 Eigenvalues that are approximately equal and the remaining values, which will be larger than the others, are related to the interference.

## 3.6 Mitigation

Interference is an undesired part of any communication system and there are a number of methods, practices and technologies, to mitigate these signals. The practices involve using RF knowledge to identify potential inference signals as well as methods to lower their effect. Checking for and isolating nearby transmitters, grounding equipment properly and using insulated transmission lines are examples of good practice. Technologies, including computer software, are available to mitigate interference. Narrow band filters can be used to filter out of band interference. Very directive antennae or narrowband antennae can be used effectively as electronic or spatial filters. Many beamforming algorithms are designed to nullify interference such as LCMV and null steering which were mentioned previously. Subspace methods also exist that create a signal space that is orthogonal to the interference and project the input signal onto this subspace. Details of subspace orthogonal projection techniques are given later.

## 3.7 ADC Dynamic Range and Saturation

An ADC used in the front end of the receiver system will have a spurious free dynamic range (SFDR). This dynamic range is a measure of the ratio between the highest signal and the next highest spurious frequency power that the ADC can pick up which is illustrated in Figure 3.4. The image represents the SFDR of a component with non-linear harmonics present. The dynamic range of an ADC is important when designing an interference mitigating antenna array. If the interference signal is too large and the SOI is considered to be the spurious signal, then if the SFDR is not large enough, the SOI's are lost to the ADC.

This is shown in Chapter 8.4 when the interference power level increases the signals are lost even after orthogonal projection. In practical systems there will be a trade-off between SNR and SFDR of the ADC because they are inversely proportional to each other. Spurious signal components are removed by spreading them over a large bandwidth which increases the noise floor and increases SFDR, however the SNR will drop. In a properly designed system the SFDR should be larger than the SNR because non-linear spurs should be below the noise floor.

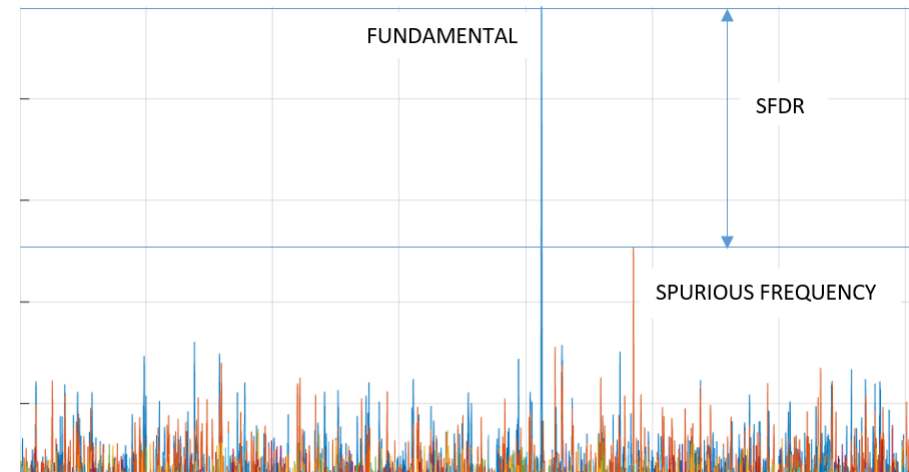


Figure 3.4: Diagram illustrating the concept of spurious frequency dynamic range (SFDR).

An important consideration for an ADC is to ensure that the input signal falls between the max input voltage range of the ADC otherwise clipping and distortion of the signal will occur.

AGC prevents amplitude saturation from occurring by controlling the gain of a variable gain amplifier (VGA) and therefore reducing signal distortion and increasing system performance by utilizing the full resolution of the ADC. A method of controlling the gain is to establish a control loop that ensures a pre-determined magnitude bit density [5] at the output or by using the cumulative distribution function or root mean square (RMS) of the input [18], [19]. The system gain is a variable that has been varied in simulation, during the course of this thesis, to investigate the effect of ADC saturation. A simple diagram illustrating ADC saturation and the levels that the AGC attempts to keep the input signal between is shown below in Figure 3.5.

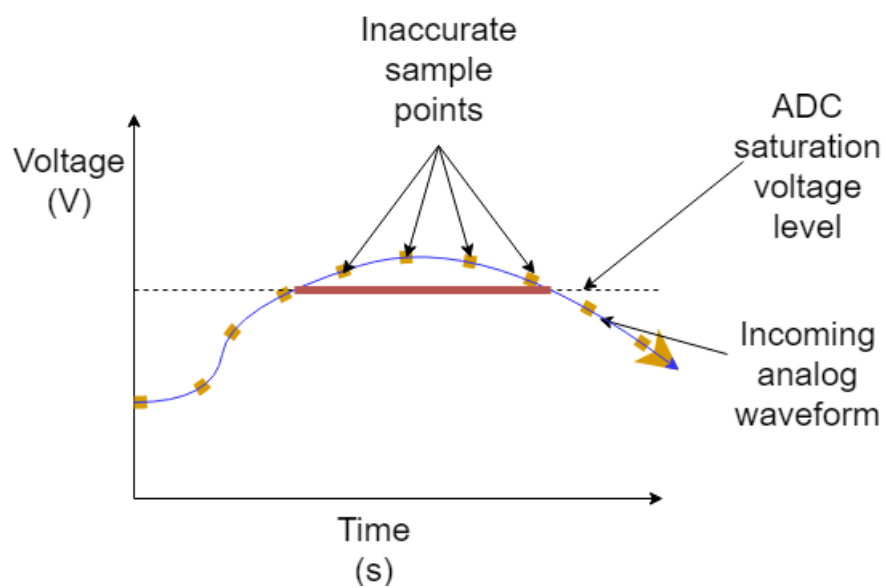


Figure 3.5: Illustration of ADC saturation when the input voltage is larger than the ADC saturation voltage and the effect of no AGC.

## Chapter 4

# Antenna Design Background Literature

This chapter gives necessary information on the antenna properties that are relevant to this study, to provide a greater understanding of the design procedure detailed in Chapter 5. A brief explanation of Wilkinson dividers, which are used in the feed network of the antenna, will be given at the end of the chapter.

### 4.1 Introduction

All antennae are structures that radiate or receive electromagnetic (EM) energy through some medium. The most common medium for EM propagation is air e.g. cars, cellphones and planes however water, vacuum and numerous variances are possible media. The radiation of EM waves through an antenna is caused by time varying charges [20]. Antennae are a combination of EM and mechanical engineering due to the complicated structures sometimes required. The collaboration is needed to ensure the antenna is physically possible to construct and has the lowest cost.

Some antennae propagate their energy through a medium in an omnidirectional manner such as cellphone and radio towers however most antennae create some signal gain in a specific direction which gives the antenna the function of a directional filter. The gain of an antenna is linked to the antenna aperture size which is linked to the wavelength of the EM wave under consideration [21]. Many modern day antenna designs are first created using computer aided design (CAD) software and their EM properties are simulated using computer software such as AWR [22], CST [23] or FEKO [24]. This approach lowers the R&D costs involved with antenna design and allows for optimization.

Antenna measurement equipment is an important part of the design process to ensure that the antenna specifications are met before it is put into use. The



equipment is sensitive and expensive however it has become a vital part of antenna design particularly for higher frequency antenna designs. There are a number of different classes and types of antenna such as wires, horns and reflectors which can be narrow or wideband [21].

Microstrip antennae are made from a radiating patch laid over a dielectric substrate and a ground plane as seen below in Figure 4.1. They are cheap and easy to produce and can be made shorter than a half wavelength depending on the substrate used. Microstrip antennae are also lightweight, durable and generally planar, as seen below, which makes them preferable to other antennae in many practical applications. The disadvantages of microstrip antennae are that they are lossy and they generally have a low bandwidth. Another potential disadvantage is that the gain of microstrip antennae is low.

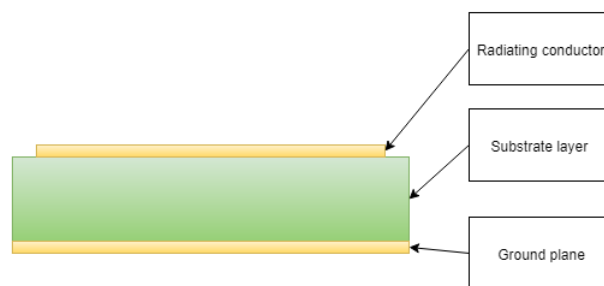


Figure 4.1: Basic microstrip layout.

There are a number of characteristics that may be designed for such as: [21] [25] [26]

1. Power gain
2. Radiation pattern
3. Polarization
4. Input Impedance
5. Bandwidth

## 4.2 Microstrip Feed Techniques

There are a number of different ways to excite a patch antenna such as probe, line and aperture feeds [21] as seen below in Figure 4.2. Probe feeding involves drilling a hole through the ground plane and substrate for a coaxial probe to be fed through and soldered on to the patch. Due to the need for drilling of holes, this technique is not used when a thick substrate is required to increase the bandwidth of the antenna due to the inductance of the large probe. The probe fed method is also not ideal for large arrays due to the number of holes to be drilled and probes to be soldered. A method to overcome the need for holes is to feed the patch with a microstrip transmission line. This technique however still gives a relatively low bandwidth and the transmission line will also radiate undesirably effecting the overall beampattern.

The cons of these feed techniques can be overcome by aperture coupled feeding of the patch which involves energy coupling from an isolated feed line, through an aperture in a ground plane and onto the patch. This method allows for greater substrate thickness and therefore increased bandwidth, also an isolated feed line which minimizes feed radiation. A method to overcome back radiation from the feed line of the aperture coupled feed is to use a ground plane below the feed line or use a lossy material over the feedline [27]. Another method is to use thinner substrate for the antenna or lower impedance lines to feed the antenna.

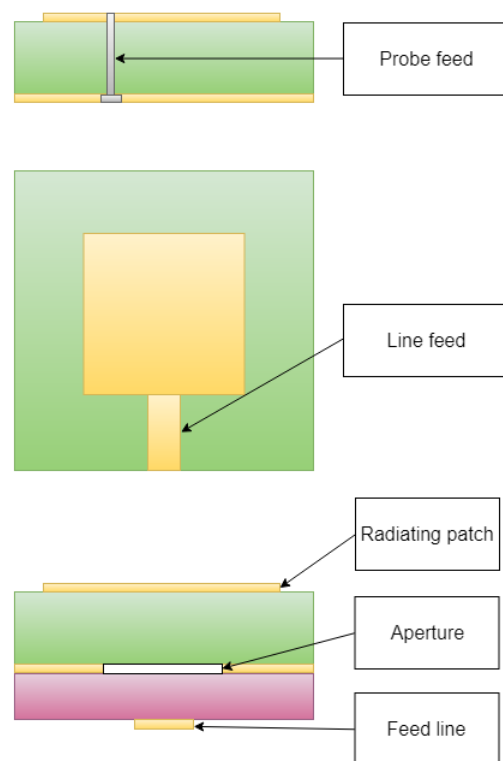


Figure 4.2: Various microstrip feed techniques.

The choice and construction of an antenna feed will have a large effect on the return loss of the antenna.

### 4.3 Return Loss

The return loss of an antenna is the amount of energy that is reflected back to the source. An impedance mismatch of components and antenna can result in a large return loss and due to the loss of energy being radiated, the antenna efficiency will diminish. Return loss can be viewed alternatively as  $S_{11}$  or voltage standing wave ratio (VSWR). Return loss is defined as:

$$RL = -10 \log_{10} \left\{ \frac{p_i}{p_r} \right\} \text{ dB}, \quad (4.1)$$

where  $p_i$  is the input power at the port and  $p_r$  is the reflected power at the port.

This thesis displays return loss as a measure of  $S_{11}$ . The scattering parameters or S-parameters indicate the relationship between the input and output ports in a network.  $S_{ii}$  is the reflection coefficient seen at port  $i$  when all other ports are terminated with a matched load [28].

$$RL = -20 \log_{10} \left\{ \frac{\text{VSWR} - 1}{\text{VSWR} + 1} \right\} = -20 \log_{10} \{S_{11}\} \text{ dB} \quad (4.2)$$

Desirable  $S_{11}$  parameters will show a low value, typically chosen at less than -10 dB for microstrip, over the desired frequency band.

### 4.4 Polarization

The polarization of an antenna can be linear, RHCP, left hand circularly polarized (LHCP) or elliptical. If the electric field only varies as a function of  $y$  or  $x$  then it is said to be  $y$  or  $x$ -polarized. If there is only an  $x$  or  $y$  component with no phase shift between them then the wave is linearly polarized. When the wave has an  $x$  and  $y$  component and there is a  $90^\circ$  phase shift between the components, then it is circularly or elliptically polarized depending on the ratio of the magnitude of the  $x$  and  $y$  components of the field.

The axial ratio of an antenna is the ratio between the  $x$  and  $y$  components of its electric field as shown in (4.3).

$$AR = 10 \log_{10} \left[ \frac{|E_{co}| + |E_{xp}|}{|E_{co}| - |E_{xp}|} \right]^2 \text{ dB} [21] \quad (4.3)$$

For GPS applications the  $E_{co}$  is the RCHP polarized E-field and  $E_{xp}$  is the LHCP polarized E-field. For a perfectly linearly polarized antenna, the axial ratio is infinity. For a perfectly circularly polarized antenna the axial ratio is 1 or 0 dB. When using circular polarization, information can be modulated onto the two orthogonal E-field components which allows for double the information transfer on the same frequency.

There are many ways of achieving the desired polarization of an antenna such as truncation of the patch corners [29] [30] or using two off centred probes or aperture slots, some of which can be seen below in Figure 4.3. Crossed aperture slots allow for a higher axial ratio bandwidth than with a single slot [27]. [21] also mentions that circular polarization can be achieved through sequential excitation of four patches that are  $90^\circ$  phase shifted. This technique excites two orthogonal linearly independent modes in the antenna that are  $90^\circ$  out of phase which gives a large axial ratio.

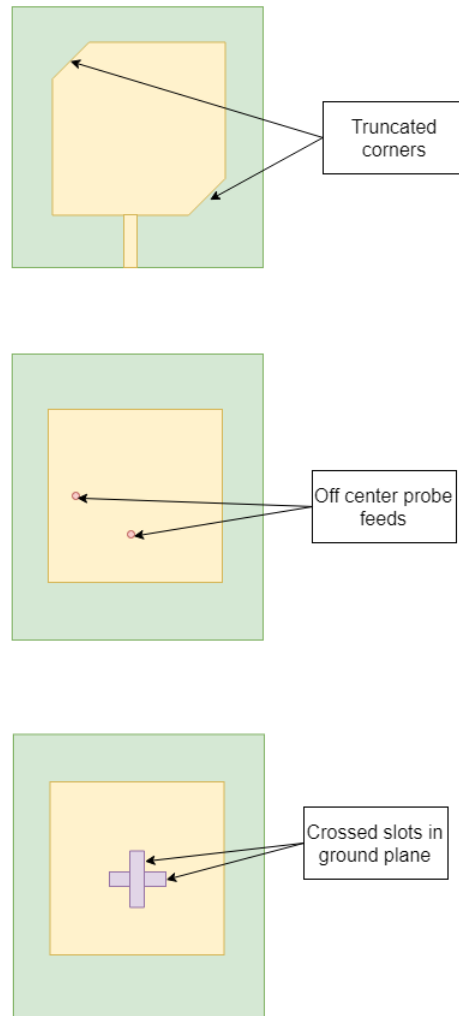


Figure 4.3: Various circular polarization techniques for microstrip patches.

## 4.5 Axial Ratio Bandwidth

Equation (4.3) shows the definition for axial ratio. It is a measure of the circular polarization purity of an antenna. The axial ratio bandwidth of an antenna is the bandwidth over which the axial ratio is below a certain threshold. Ideally for circularly polarized signals, the axial ratio should be 1 or 0 dB as mentioned in Section 4.4. This value is impossible to obtain practically therefore a typical value of 3 dB is chosen.

## 4.6 Antenna Bandwidth

The bandwidth of an antenna is the frequency range over which it operates at or above a predefined threshold. Bandwidth can be given in hertz or as a ratio of the band of operation over the centre frequency which is called fractional bandwidth (FBW) which is shown below:

$$\text{FBW} = \frac{f_{max} - f_{min}}{f_c} \quad (4.4)$$

As seen from Section 4.5 bandwidth can refer to different aspects of an antenna. Impedance bandwidth is the bandwidth over which  $S_{11}$  of the antenna is below a certain threshold. See Section 4.3.

## 4.7 Antenna Gain

The gain of an antenna is the ratio of peak radiation power between the antenna and an isotropic source in a given direction, known as directivity, multiplied by the antenna efficiency. The antenna gain is measured over an angular range which will be explained in the next section. A diagram illustrating the concept of antenna directivity can be seen below in Figure 4.4.

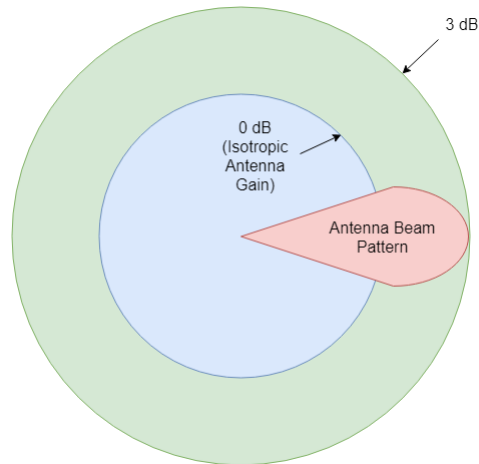


Figure 4.4: Illustration of antenna directivity showing the directive antenna pattern in pink and the isotropic pattern in light blue.

Another definition of gain is the IEEE standard which takes into account the energy reflected at the ports [31]. The gain used in this thesis is the IEEE standard.

## 4.8 3 dB Beam Width

The 3 dB beam width of an antenna is the angular range over which the normalized gain of the antenna is greater than or equal to -3 dB. The gain is normalized to the peak gain in the radiation pattern of the main lobe. This figure is also called the half power beam width. An illustration of 3 dB beam width is shown below in Figure 4.5.

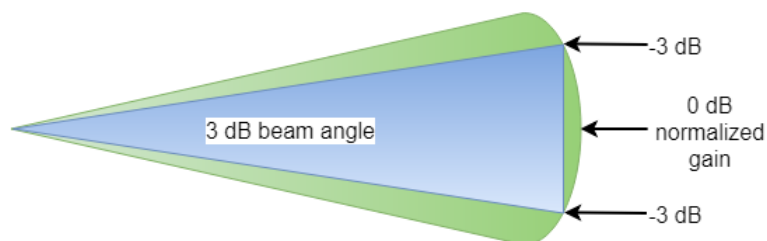


Figure 4.5: Illustration of 3 dB beamwidth showing the beampattern in green and the extent of the 3 dB angular range in blue.

## 4.9 Dual Band Antennas

A dual band antenna can receive signals from two frequency bands and has become very popular in modern wireless communications because it allows for the use of antenna elements that are singular or stacked. This technique lowers the required space of the antenna and lowers the cost and complexity of the overall system.

There are a number of methods to create dual band patches such as non-rectangular patches [32] or altering the ground plane [33].

It has been shown that aperture coupled patch antennae can be designed with 25% bandwidths which would cover the L1 and L2 frequency bands however because the actual bands of interest are only about 40 MHz, this antenna may be susceptible to out of band RFI [27]. A stacked patch microstrip antenna can give dual band performance [21], such as the one below in Figure 4.6. The thickness of this antenna also increases its bandwidth.

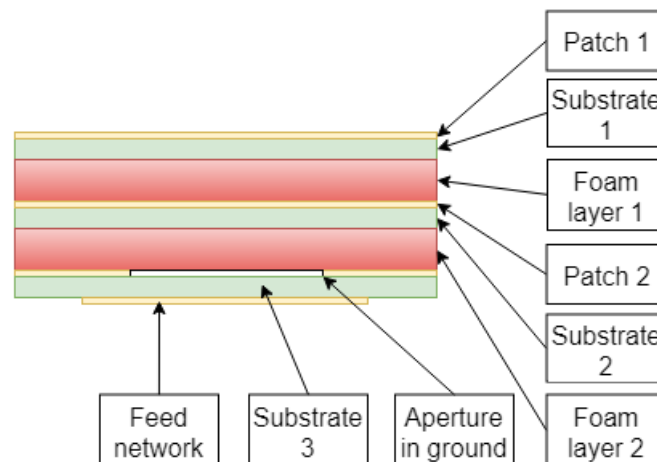


Figure 4.6: The layout of a multi-layered aperture fed microstrip patch antenna.

The requirements for a GPS antenna are that it must operate either in L1 (1.57542 GHz), L2 (1.2276 GHz) or both bands. The polarization of the antenna must be RHCP. The design in Chapter 5 is of a dual band, multi-layered, aperture coupled microstrip patch antenna.

## 4.10 Wilkinson Combiner

This section is added due to the use of the Wilkinson combiner in the feed network of the microstrip antenna mentioned in Section 5.2.

The Wilkinson combiner shown in Figure 4.7 is a 3 port passive device that



adds power from two inputs with equal phase. It consists of 2 quarter wave length lines at an impedance of  $\sqrt{2}Z_0$  that lead from ports two and three to port one, where  $Z_0$  is the characteristic impedance of the system. An isolation resistor, with  $R = 2Z_0$ , is placed between ports two and three. The isolation resistor absorbs energy that is input to ports 2 or 3 which could be unwanted reflections. The resistor only absorbs differential or odd mode energy if there are reflections from ports 2 and 3 [28].

The advantages of the Wilkinson combiner are its simplicity, isolation and cost when printed on a circuit board. Its disadvantage is its frequency response which is limited due to the use of quarter wave transformers.

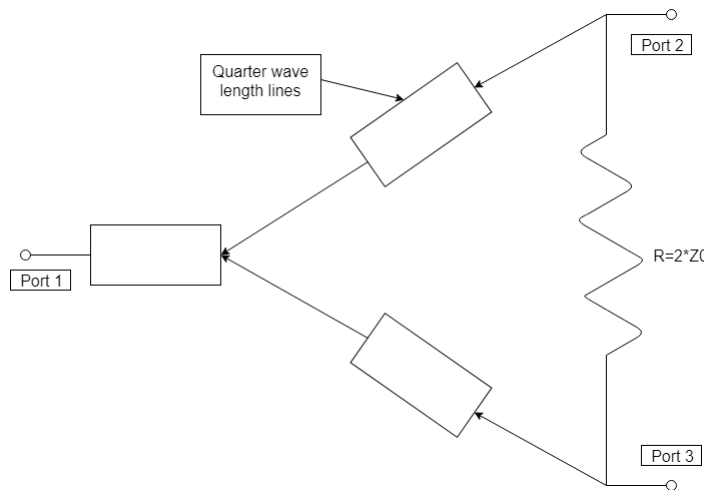


Figure 4.7: Circuit diagram of the Wilkinson combiner used in the antenna design in Section 5 with the quarter wave length lines at  $\sqrt{2}Z_0$ .

# Chapter 5

## Antenna Design Process

### 5.1 Introduction

This chapter presents the design and simulation of a dual band microstrip antenna for L1 and L2 GPS frequencies. The antenna is a dual band multi-layered microstrip patch antenna (see Section 4.9). The task was initially undertaken to implement the design in a uniform linear array (ULA). The available substrate with a fairly low dielectric constant, and foam which is predominantly air, meant that the size of the antenna would be large while having a large bandwidth and low return loss [34]. Conclusions drawn from the design, construction and measurement processes are shown in Section 5.4.

### 5.2 Method

#### 5.2.1 Design

The antenna design is based on [27] and [35].  $S_{11}$  should be below -10 dB from 1.1-1.6 GHz and the axial ratio should be below 3.5 dB across the same frequency band.

The design of this antenna, using formulae, is difficult because of the interaction of different parameters, of which there are more than 20. The initial dimensions of the feed network were designed for a characteristic impedance of  $50 \Omega$  at 1.35 GHz and the square patches have initial dimensions which were calculated using common equations, which gave results of 83 mm for the top patch and 65 mm for the bottom patch. These values were tuned together because of strong coupling between the patches. The design of the Wilkinson combiners was done in AWR for fast optimisation, then the parameters were transferred to CST for a more accurate simulation. The process was iterative because optimisation was not feasible due to the long simulation time. The results shown below are after much refining and tuning of the parameters. The parameters that have the largest effect on the antenna are the substrate thick-

nesses and dielectric constant, patch widths, the aperture width and length and the feed network layout, which has multiple parameters. Increasing the substrate thickness and decreasing the dielectric constant will increase the bandwidth however it will also decrease the amount of coupling between the aperture and patches. The patch dimensions dictate the frequencies of the L1 and L2 bands. The size of the aperture controls the coupling between the feed lines and the antennae. The size of the aperture also has an effect on the impedance match of the antenna. The feed network is important because it effects the phase differences between the feed arms and therefore effects the polarization purity of the antennae. It has been shown that dual band designs similar to the one proposed in this chapter, have large tolerances to the manufacture process [27].

A feed network consisting of 3 Wilkinson combiners, each with a  $90^\circ$  phase shifter and a further  $180^\circ$  phase shifter to give phases of  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$  and  $270^\circ$  at the feed arms of the network with equal amplitude. Each Wilkinson combiner has a  $100\ \Omega$  isolation chip and the inputs and outputs are matched to  $50\ \Omega$ . These  $90^\circ$  phase shifts ensure a good axial ratio bandwidth of the antenna by creating two orthogonal linearly polarized modes with equal amplitude and  $90^\circ$  out of phase. Above the feed network is a crossed slot aperture which must be designed for a centre frequency approximately in the middle of the two bands of interest at 1.35 GHz. The final 3D layout of the feed network can be seen in Figure 5.1. The phase shifts of the arms are labelled. The  $180^\circ$  phase shift length can be seen in the upper section of the layout. The Wilkinson combiners were created in AWR using circuit elements and the dimension values were transferred to CST.

A challenge with the design of the feed network is to ensure that the  $0^\circ$  and  $90^\circ$  arms of the Wilkinson combiners remain fixed as other parameters are adjusted during tuning.

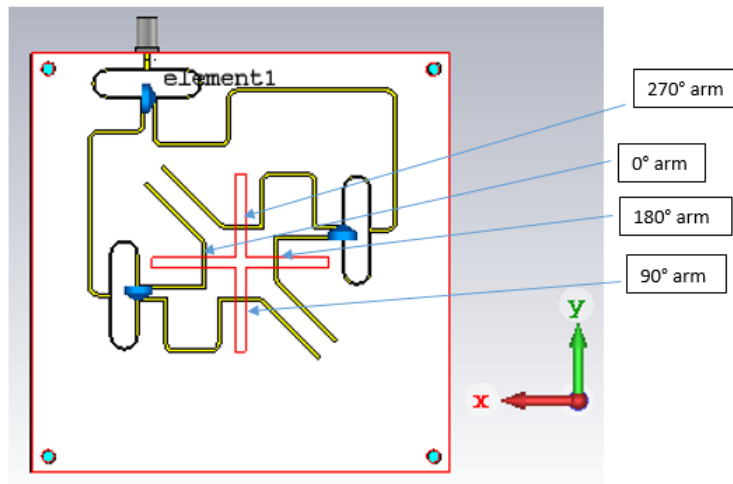


Figure 5.1: The final 3D layout of the feed network and aperture used on the antenna. Showing the phase of each arm coupling to the aperture, which is the cross shown in red, and which results in circular polarization of the antenna.

### 5.2.2 Simulation

As mentioned previously, the simulation of the antenna was done on CST. Post processing was performed to plot a graph of the axial ratio and IEEE RHCP gain vs frequency which can be seen below in Figure 5.2.

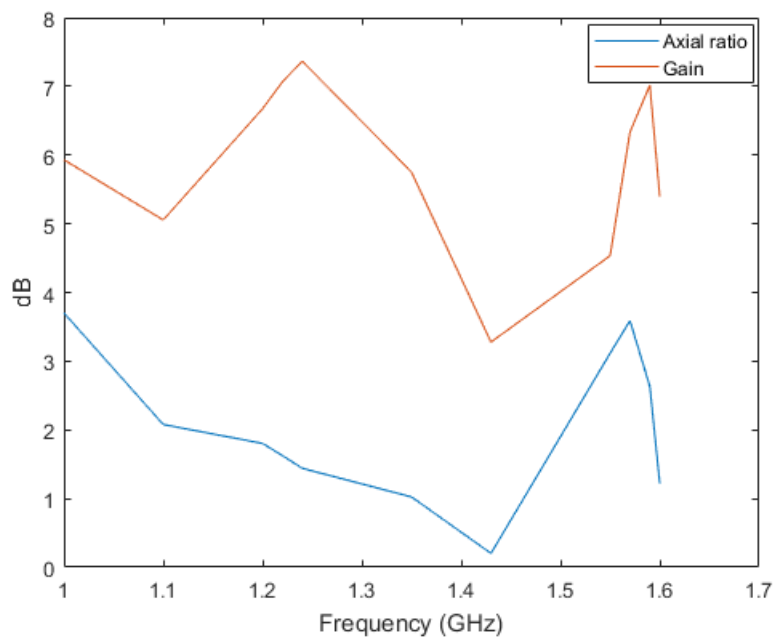


Figure 5.2: Simulated axial ratio (blue) and RHCP gain (red).

The images below, Figures 5.3, 5.4, 5.5, 5.6, 5.7 and 5.8 are the various CST beampattern plots at the L1 and L2 bands.

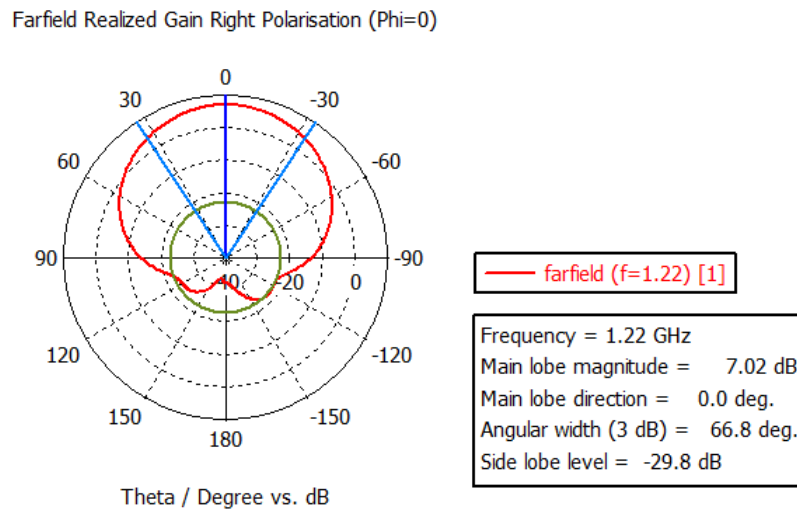


Figure 5.3: Simulated RHCP polar beampattern of the final antenna design at 1.22 GHz.

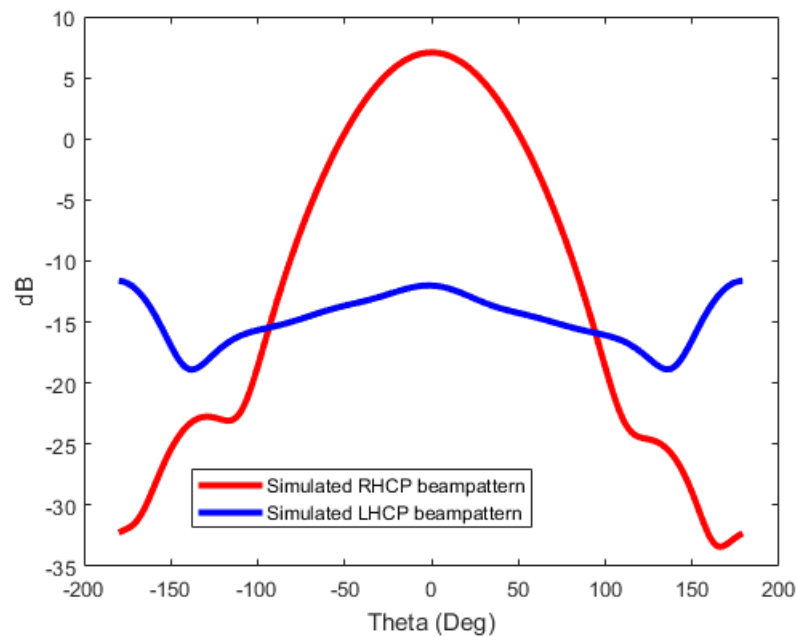


Figure 5.4: Simulated Cartesian beampattern of the final antenna design at 1.22 GHz, showing RHCP in red and LHCP in blue.

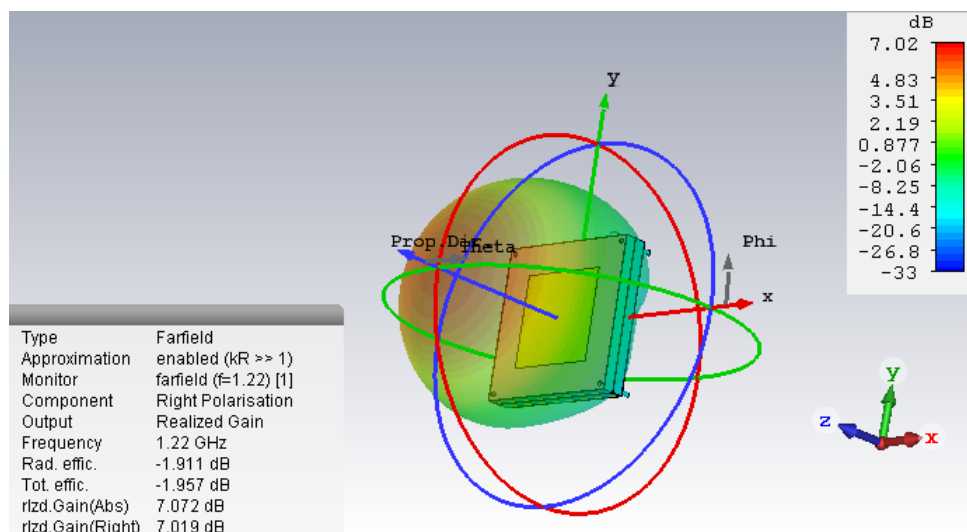


Figure 5.5: Simulated RHCP 3D beampattern of the final antenna design at 1.22 GHz.

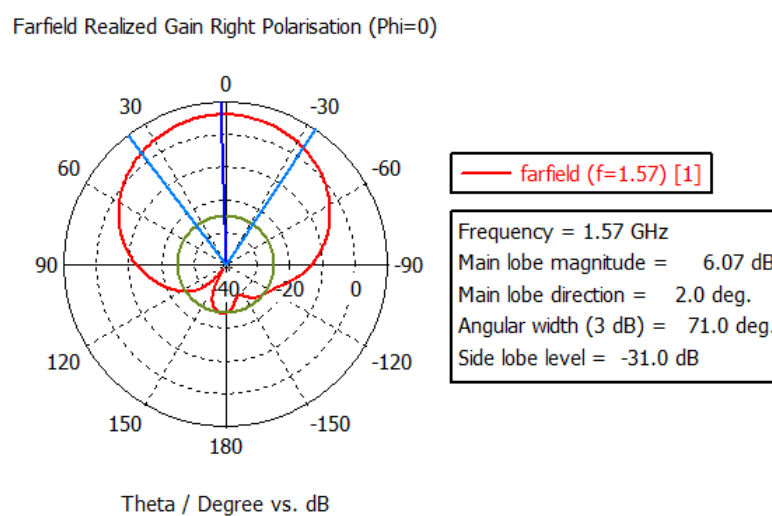


Figure 5.6: Simulated RHCP polar beampattern of the final antenna design at 1.57 GHz.

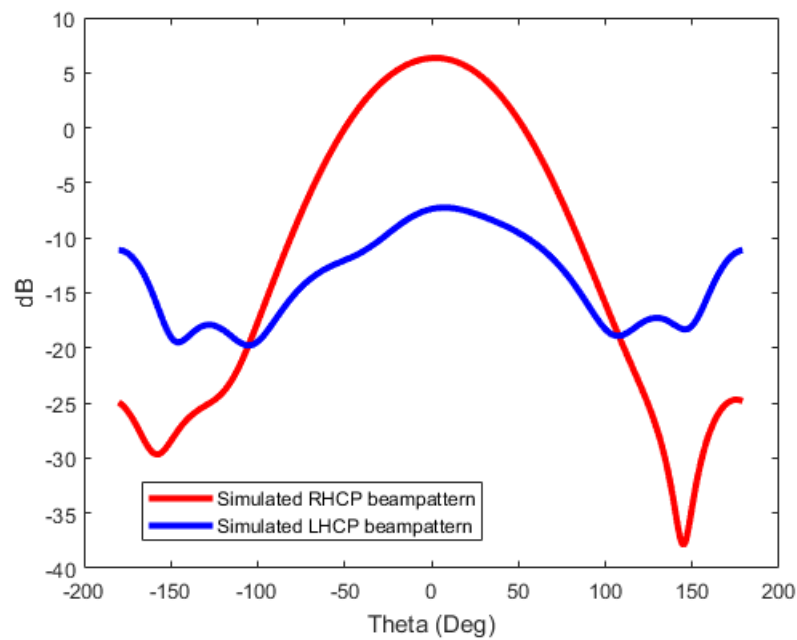


Figure 5.7: Simulated Cartesian beampattern of the final antenna design at 1.57 GHz, showing RHCP in red and LHCP in blue.

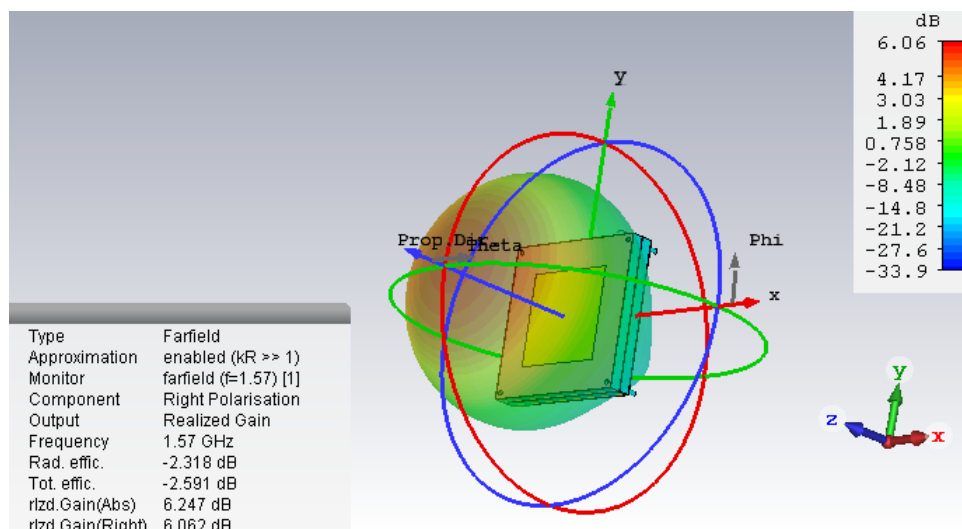


Figure 5.8: Simulated RHCP 3D beampattern of the final antenna design at 1.57 GHz.

The antenna was fabricated using Rogers 0.508 mm thick RO4003C substrate with a conductor thickness of  $35 \mu\text{m}$ . Nylon screws were used to fasten the antenna layers together and a ground plate has been attached 11.3 mm below the feed network. The 2 foam layers are both 10 mm thick and are

Rohacell HF71 foam with a dielectric constant of 1.075 which helps to increase the bandwidth of the antenna due to the increase in the thickness of the antenna without a large effect on the coupling between patches. The increase in bandwidth is due to the decrease in stored reactive energy in the antenna when the thickness is increased. The loss in reactive energy means an decrease in the  $Q$  factor of the antenna which is defined as,  $Q = \frac{f_c}{3 \text{ dB bandwidth}}$ , therefore if the centre frequency is fixed, the bandwidth of the antenna increases. All of these aspects of the antenna can be seen in Figure 5.9. The simulated SMA port can also be seen. Figure 5.10 shows the simulated  $S_{11}$  vs frequency. The results are acceptable being below -10 dB across the entire band from 1.1-1.6 GHz.

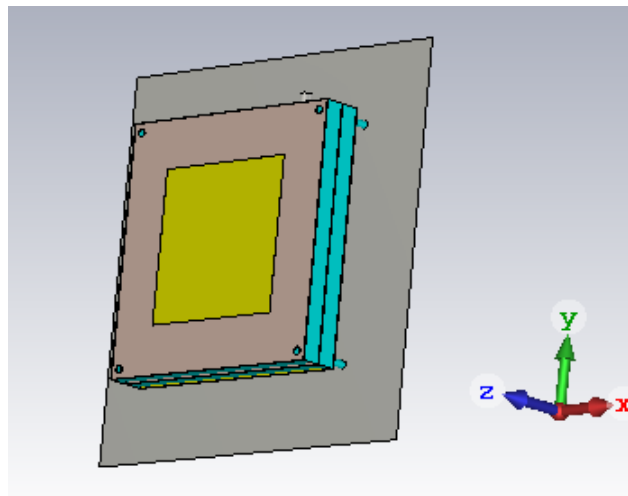


Figure 5.9: Final 3D layout of the dual band multi-layered microstrip patch antenna.



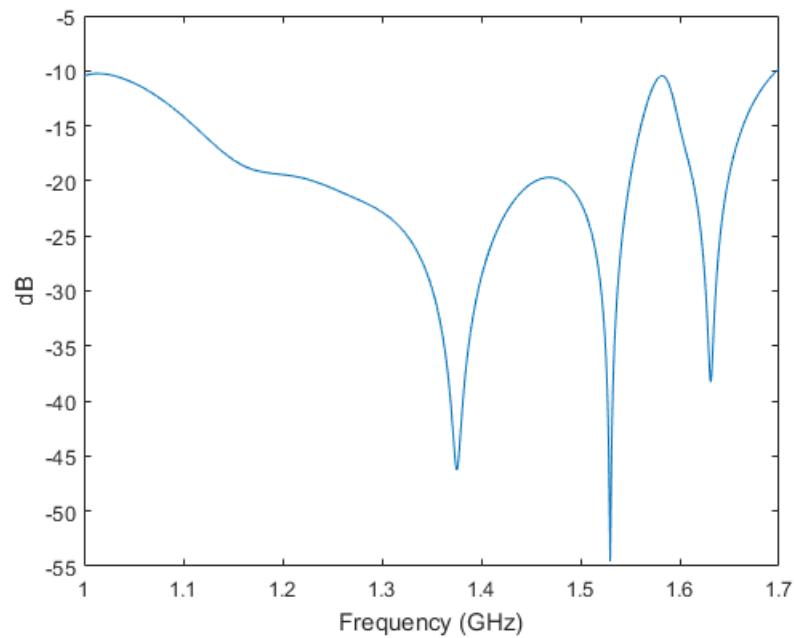


Figure 5.10: Simulated S-parameters of the final antenna design.

The results of the antenna measurements can be seen in Chapter 8.4.

### 5.3 Fabricated Antenna Results

The antenna was fabricated in the machine workshop at Stellenbosch University, and measured using a Rhode & Schwarz vector network analyser (VNA) and an-echoic chamber. The measurement probe was a linearly polarized antenna so two measurements were made, with 2 orthogonal polarizations, vertical and horizontal. The results show comparisons between the measurements and simulations. An image of the final fabricated antenna can be seen in Figure 5.11.

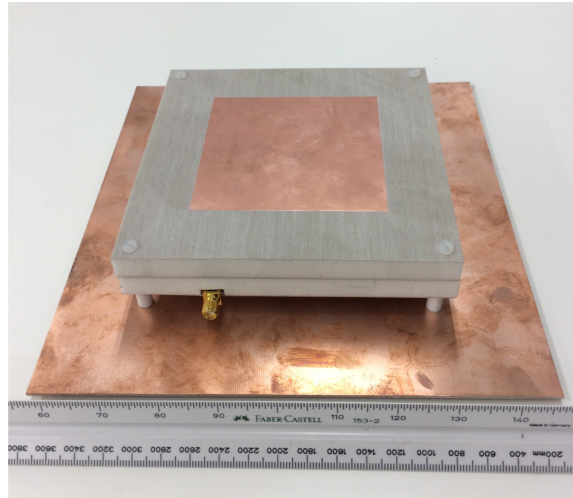


Figure 5.11: Fabricated version of the antenna designed in Chapter 5.

A comparison between the measured and simulated  $S_{11}$  results can be seen below in Figure 5.12. There is a clear correlation between the graphs however the resonance that was designed for at 1.35 GHz cannot be seen. The reflection is acceptable, being below -10 dB across the L1 and L2 bands. There is a slight shift in frequency response seen in the S-parameters which could be a result of slightly inaccurate patch dimensions.

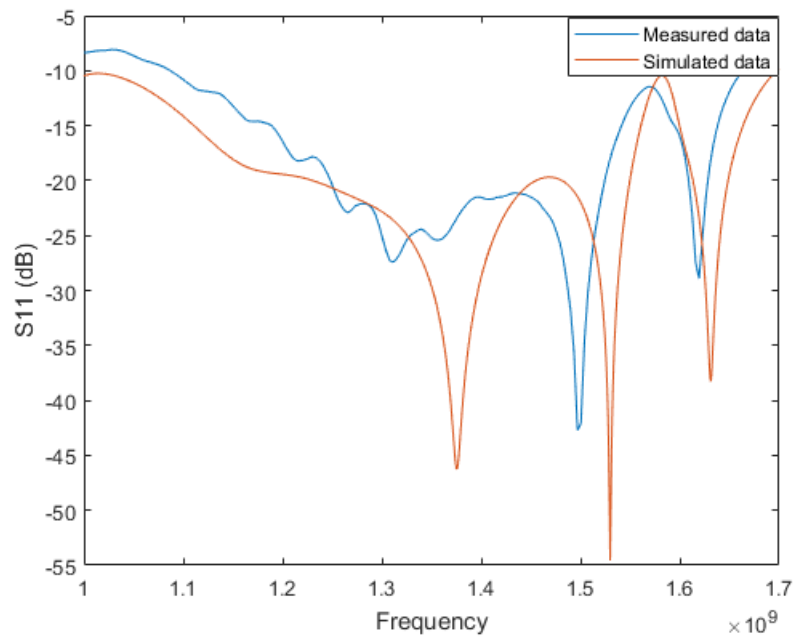


Figure 5.12: Comparison of measured (blue) and simulated (red)  $S_{11}$  results of the dual band microstrip antenna designed in Chapter 5.

The use of low dielectric substrate was responsible for the size of the antennae. The sizes of the patch antennae are  $\pm 8 \text{ cm}^2$  and the profile of the feed network means that the total size of the antenna is approximately  $13 \text{ cm}^2$ . These dimensions mean that the antenna will be impossible to implement in a half wave length array at L1 frequencies ( $\frac{\lambda}{2}=95.2 \text{ mm}$ ). The dimensions of the antenna can be altered, by changing the substrate permittivity, to allow for use in an array however the antennae will still be in close proximity and therefore the mutual coupling of the antennae will have to be thoroughly investigated.

The axial ratio at 1.22 GHz and 1.575 GHz is 3.2 dB and 1.6 dB respectively at  $0^\circ$  azimuth. These results are somewhat different from the simulated values of 1.6 dB at 1.22 GHz and 3.5 dB at 1.575 GHz and the result at 1.575 GHz is slightly higher than the requirements however with further tuning it should be possible to lower this value. As mentioned before, the parameters were not optimized, just tuned, due to long simulation times.

The graphs below, Figures 5.13 and 5.14 show the measured and simulated RHCP and LCHP beampatterns at the bands of interest. The measurements show that the L2 band has a -3 dB beamwidth of  $71^\circ$  and the main lobe is centred at  $4^\circ$ . The simulation shows that the 3 dB beamwidth for the L2 band is  $67^\circ$  and the beam is centred at  $0^\circ$ .

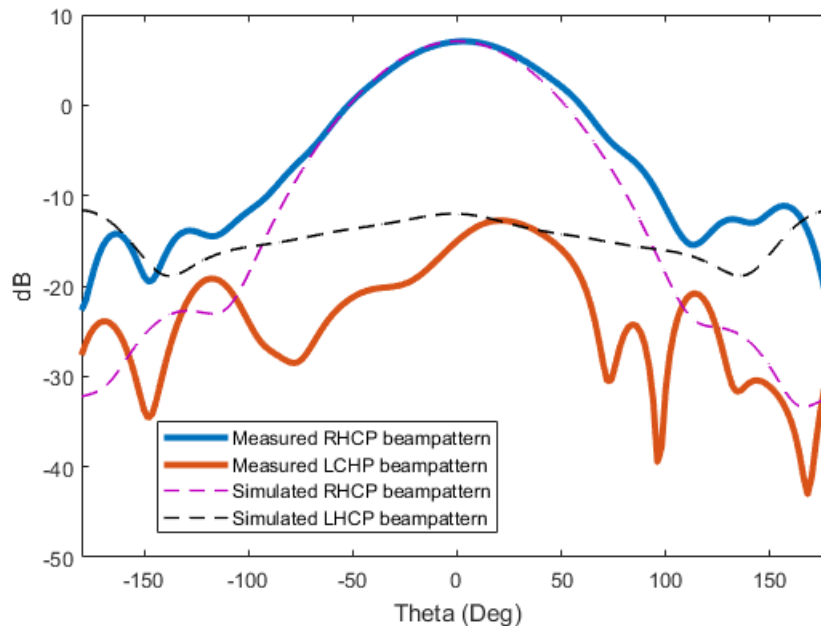


Figure 5.13: Measured and simulated beampatterns of the antenna shown in Figure 5.11 at 1.22 GHz.

Figure 5.14 below shows the measured results for the L1 GPS band. The graph shows that the antenna has a -3 dB beamwidth of  $70^\circ$  and the main

lobe is centred at  $-4^\circ$ . The simulations show the 3 dB beamwidth as  $71^\circ$  and the main lobe is centred at  $2^\circ$ . The difference in beamwidth is small, however the off centred main lobes could cause problems in an array.

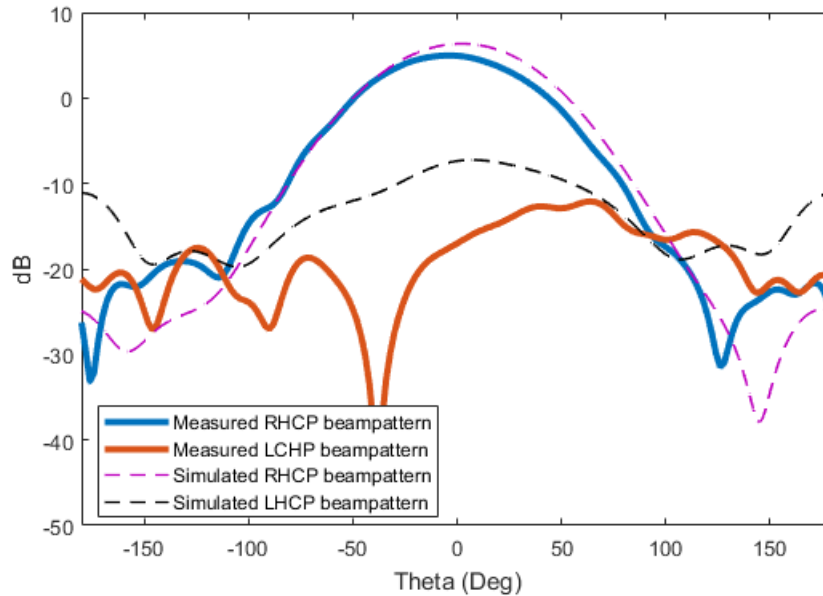


Figure 5.14: Measured and simulated beampatterns of the antenna shown in Figure 5.11 at 1.57 GHz.

The graph below, Figure 5.15 shows the simulated and measured RHCP gains. There is a drop in gain at the L1 band of approximately 1.7 dB.

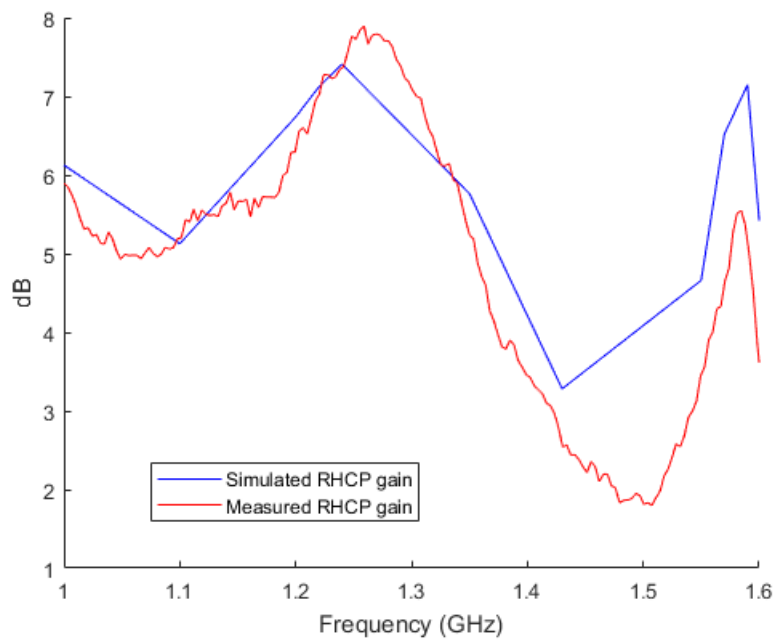


Figure 5.15: Measured RHCP gain (red) of the antenna shown in Figure 5.11 vs CST simulation gain (blue)

There is a gain of approximately 7 dB in the L2 band and 4.8 dB in the L1 band. The gain in the L2 band matches that of simulation results, the L1 bands gain is lower than the simulated value of 6.5 dB.

The conclusion drawn from the measured results is that the antenna will function as a dual band GPS antenna however due to the size of the antenna, it cannot easily be implemented into an array with  $\frac{\lambda}{2}$  element spacing.

## 5.4 Conclusions from the Design Process

During the design and construction of the antenna, a few things were noted. The feed of the antenna is very important to simulate as accurately as possible. This stage is critical because if the antenna is not matched correctly then very little energy will enter the antenna and the results will be inaccurate. It is also important to ensure the connector is adequately soldered on the input feedline for the same reason mentioned above. The measurement below in Figure 5.16 were made on the antenna when the input track of the antenna was microscopically cracked after the SMA was indelicately connected to the vector network analyser (VNA). It shows the sensitivity of the antenna at the input stage because of the large differences between these results and the correct  $S_{11}$  measurements seen in Chapter 8.4. The axial ratio of the antenna is also largely altered.

Time domain simulations were done to find the point at which the reflections occurred. Shorting the feed network at various points gave an indication that the reflection was directly at the input of the feed network.

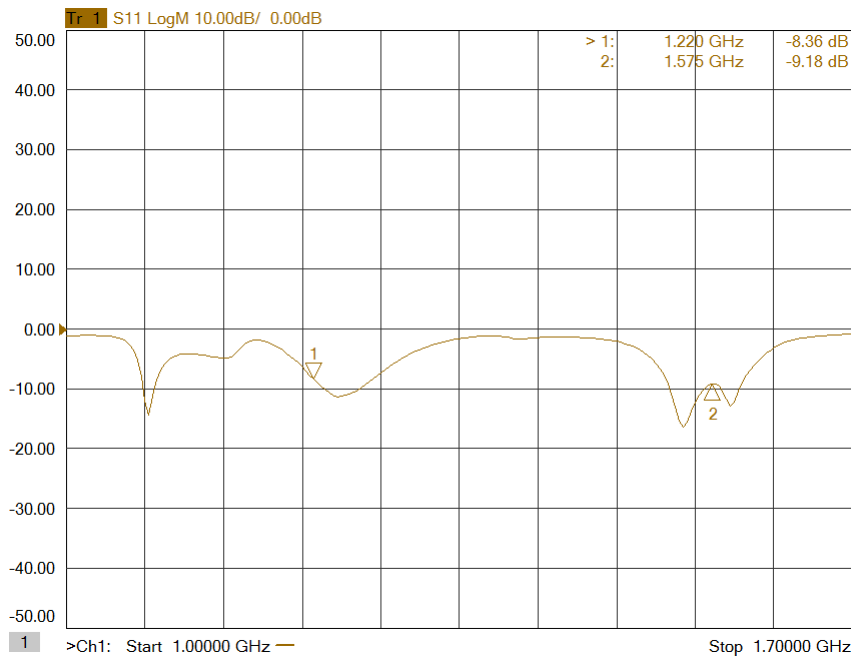


Figure 5.16: Image showing the S-parameters that were measured while a faulty port was used, illustrating the importance of constructing an accurate port.

Checking the mesh of the simulated antenna to ensure it is accurate is very important. Mesh reconstruction and simulation in time and frequency domains, then comparing the results are simple ways to ensure accurate simulations.

# Chapter 6

## Testing Procedure

This chapter details the simulations that were run on the DOA algorithms as well as algorithms 1 and 2. The DOA algorithms search for the direction of the incoming signals, algorithms 1 and 2 are blind beamforming algorithms that null interference and create array gain in the directions of SOI's. Information is also given about the steps involved with practical data collection.

### 6.1 Simulation Details

The DOA algorithms are simulated while varying the SNR and the number of bits of the ADC, integration time and using I/Q or bit shifted data, which is created through the shifting of the I data 1 bit and is possible because the sample frequency is 4 times that of the intermediate frequency. Oscillator and mixers errors can result in the I/Q and bit shifted data being unequal. Using real non-complex data introduces an uncertainty to the beamforming results. The results show a beam in the direction of a signal as desired however if the signal was at  $25^\circ$ , for instance, then the beamformer would give beams at  $25^\circ$  and  $-25^\circ$  because the phase of the signal is not calculated. Many times, throughout the thesis this 'dual band' was seen, and it was determined that it was a result of the phase of the signal being improperly calculated.

The purpose of the parameter sweeps are to determine the hardware limits of the algorithms. As mentioned in Section 1.2, the changes in parameters represent significant changes in hardware.

The parameters that are not swept in a particular simulation are fixed to, SIR = -30 dB, integration time = 16000 samples, SNR = -20 dB and number of bits = 12. The signals are coming from  $60^\circ$ ,  $30^\circ$ ,  $-5^\circ$  and  $-40^\circ$  with power levels of -129 dBm and a noise floor of -110 dBm. The DOA algorithms were deemed successful if the 4 largest peaks of the spectra were within  $5^\circ$  of the true directions of arrival.

The antenna array is a 12 element,  $\frac{\lambda}{2}$  ULA where  $\lambda = 0.19$  m. The antenna elements are off the shelf L1 GPS patch antennae with a ceramic substrate.

1	SNR/Integration time
2	SNR/Number of ADC bits
3	SNR/IQ vs Bit shift

Table 6.1: Table showing the variation of parameters when testing the various DOA algorithms mentioned in Chapter 7. The results of these simulations can be seen in Chapter 8.4.

A simulation was run to study the effect of AGC at the VGA by fixing the system gain and varying the SNR. The simulations performance can be seen to degrade when the SNR is increased with no AGC due to increased noise caused by the ADC clipping the input signal.

Algorithms 1 and 2 were tested with the same parameter sweeps but with the addition of SIR variation. Algorithm 1 was tested with and without an interferer. The number of bits was varied between the tests in order to find the failure point of algorithms 1 and 2. Due to their nature a direct comparison of the algorithms is not viable.

1	SNR/Integration time
2	SNR/Number of ADC bits
3	SNR/SIR
4	Number of ADC bits/Integration time
5	SIR/Number of ADC bits
6	SIR/Integration time
7	IQ-Bits shift vs SNR
8	IQ-Bits shift vs SIR
9	IQ-Bits shift vs Number of ADC bits
10	IQ-Bits shift vs Integration time
11	1 bits shift - SIR vs Integration time
12	1 bits shift - SIR vs SNR

Table 6.2: Table showing the variation of parameters when testing algorithms 1 and 2, the results of which can be seen in Chapter 8.4.

Algorithm 1 was tested using 4 GPS signals, as the minimum requirement for a GPS position lock, and the algorithm has been tested with and without interference. The beam patterns are compared to those of LCMV which is an indication of the pattern produced by algorithm 2.

Algorithm 1 was also tested very similarly to the DOA algorithms. The test was deemed successful if the beam pattern peaks and null were within  $5^\circ$  of the SOI and interference respectively.  $5^\circ$  was chosen to keep this simulation



similar to that performed for algorithm 2, which can have a beam pattern error of  $\approx 5^\circ$  through its search space. This point is mentioned in Section 8.3. The parameters that are not varied are fixed to SIR = -30 dB, integration time = 16000 samples, SNR = -20 dB and number of bits = 12.

Algorithm 2 was tested thoroughly by comparing a single antenna, simple beam steering without a null, and LCMV. Each circumstance was tested by varying the number of ADC bits, SIR, SNR and integration time. The null produced by the LCMV algorithm was unrealistically low and to account for the deep null, the SIR was decreased to -69 dB as opposed to -30 dB for algorithm 1. The algorithm was deemed successful if the correct frequency, phase and direction of a satellite was determined. Each variable step was run 100 times and the success percentage of the algorithms was recorded. 36 simulations were run while varying different parameters to investigate the failure point of each circumstance (LCMV, beam steering or a single antenna).

The variation of the SNR in the system will affect the required performance of the LNA's. If the signal has more power at the antenna then amplification is less vital. Therefore if the required SNR is lower the hardware will be less complex and cheaper. The need for I/Q data means that the system will have more channels to account for before the digital back end. The sole collection of the I data means only one cable will be used between the GPS receiver and digital back end, which again lowers the cost and complexity of the system.

## 6.2 TART

TART [36] or transient array radio telescope is a simple 24-channel radio telescope developed by Maximilian Scheel, Timothy C. A. Molteno and Charles Shaw and operates at the L1 GPS frequency. The system samples at a rate of 16.368 MHz and mixes the input signal down to 4.092 MHz, meaning that each bit of the output represents  $90^\circ$  of the intermediate frequency data. This allows one to only collect the in-phase (I) data, then bit shift this data in order to create the quadrature (Q) data. This is what is referred to when 1 bit shifted data is mentioned. The algorithms mentioned previously are tested in simulation to see if the TART system is viable hardware for adaptive beamforming.

## 6.3 Practical Data Collection

This section details the procedure for data collection using the TART system. Two log periodic antennae are used, one as an SOI and the other as an interference signal, as well as a 12 element ULA. An image of the setup can be

seen in Figure 6.1. An image of the fabricated ULA, the TART system and its power supply can be seen in Figure 6.2.

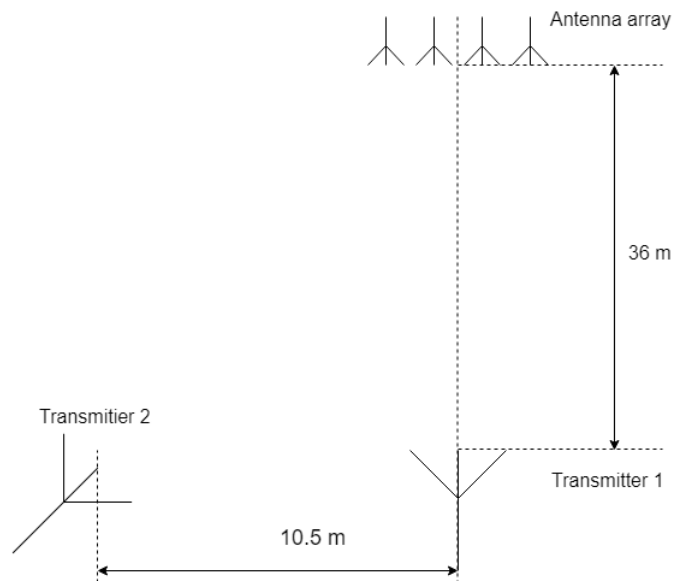


Figure 6.1: Illustration of the setup for practical data collection

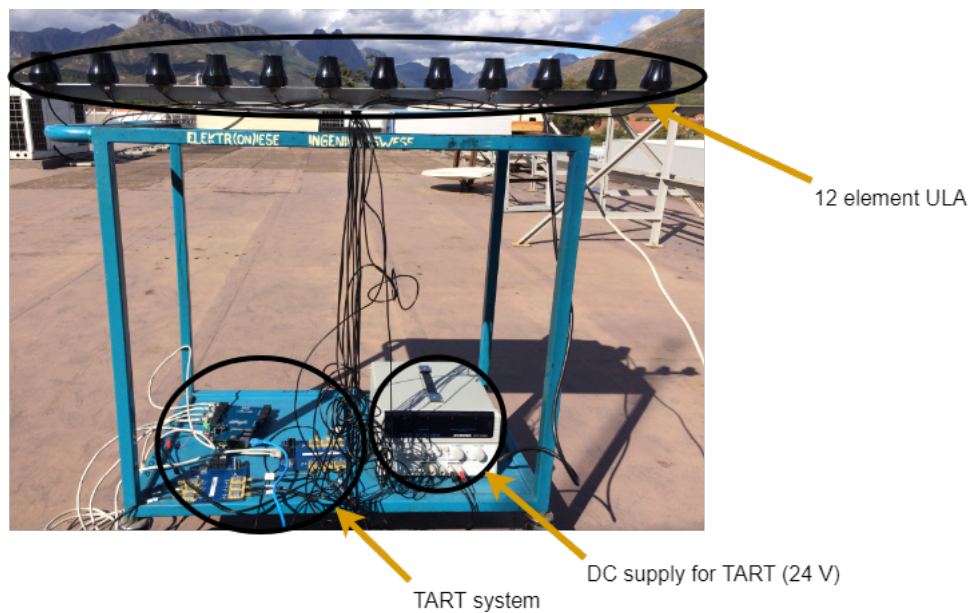


Figure 6.2: Image of the 12 element ULA, TART system and its power supply on the roof of the Stellenbosch University microwave laboratory.

The Friis equation is used to determine the signal power level at a receive antenna, given the effective apertures of both transmit and receive antennae, the power of the transmit antenna, their distance apart and the frequency of operation. The equation was used to determine the power levels of the transmitters used in the practical set up. The equation is defined as:

$$P_r = G_t G_r P_t \left( \frac{\lambda}{4\pi r} \right)^2 \text{ W}, \quad (6.1)$$

or

$$P_r = G_t + G_r + P_t + 20 \log_{10} \left( \frac{\lambda}{4\pi r} \right) \text{ dB}, \quad (6.2)$$

where  $G_t$  and  $G_r$  are the gains of the transmitter and receiver respectively.  $P_t$  and  $P_r$  are the powers of the transmit and receive antennae respectively. The power is in Watts.  $r$  is the distance between the 2 antenna centres. The log periodic antennae used have a gain of  $\approx 6$  dB and the ULA has a gain of  $10 \log_{10}(N)$  where  $N$  is the number of antennae. Therefore using the above equation, in order to get -129 dBm at the receiver, the transmitter must have a power level of at least -77.6 dBm in an ideal situation. The power should however be increased due to the losses in the cables, which at 1.5 GHz is 0.6 dB/m. Therefore for a 5 meter cable, and a desired receive power of -129 dBm the transmit power should be closer to -76.1 dBm due to the 1.5 dB loss in the cables. The TART system outputs its data into .pkl files and the time data must be extracted. In order to accomplish this a python program was created to extract the time data from each channel and write it into a text file which is then read by Matlab and passed to the various algorithms. After collection of the time data, phase calibration must be performed prior to signal processing.

### 6.3.1 Phase Calibration

Beam forming hardware introduces phase and amplitude errors to each channel and can lead to undesirable beam pattern characteristics such as large side lobes and beams in undesired directions. If the errors introduced by the system are accounted for, practical behaviour of an array should match its model very well. Errors can come from temperature variations of components and time varying outputs of components such as phase shifters and amplifiers and mixers, which are very difficult to calibrate out. They can also come from unequal cable lengths for each channel, bending flexible RF cables and using far field assumptions.

The image below, Figure 6.3, illustrates the errors introduced by far field approximation. The far field region is the area between the array and receive antenna where the radius is greater than  $2D^2/\lambda$  and  $D$  is the largest diameter of the antenna or array, which in this context is  $12d$  where  $d$  is the antenna element spacing or  $\frac{\lambda}{2}$ . For the array shown in Figure 6.2, the radius required is

at least 13.5 m at L1 frequencies. The ideal phase front is a plane wave where the signal arrives at all antennae simultaneously however the phase error is still  $22.5^\circ$  at this distance.

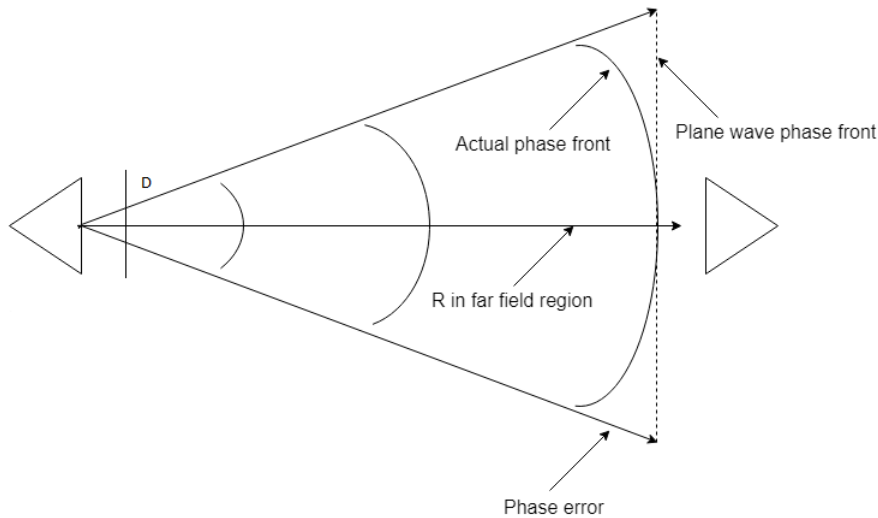


Figure 6.3: Illustration of the far field region and the  $22.5^\circ$  phase error that occurs at a distance of  $R = 2D^2/\lambda$ . The image shows the ideal wave front, which is planar, and the actual wave front which is spherical.

Errors are caused by each channel having a unique phase and amplitude characteristic. Calibration of the system corrects the errors between channels in order to ensure that each channel has equal phase and amplitude. Phase errors can easily be simulated in order to see the effects of changes in cable length. If the clocks of each mixer are not synchronized this will also cause a large amount of relative phase error between channels as illustrated in Figure 6.4. The jitter cleaners on each TART module are what ensures clock synchronisation. The jitter cleaner may, however, lock out of sync every time the system is turned on. This is the main cause of errors as well as the cable lengths and far field approximation. Rotating the array during practical testing can cause the calibration step to become invalid due to the far field approximation. The effect should be small in the case of the setup shown in Figure 6.1 and the results still usable.

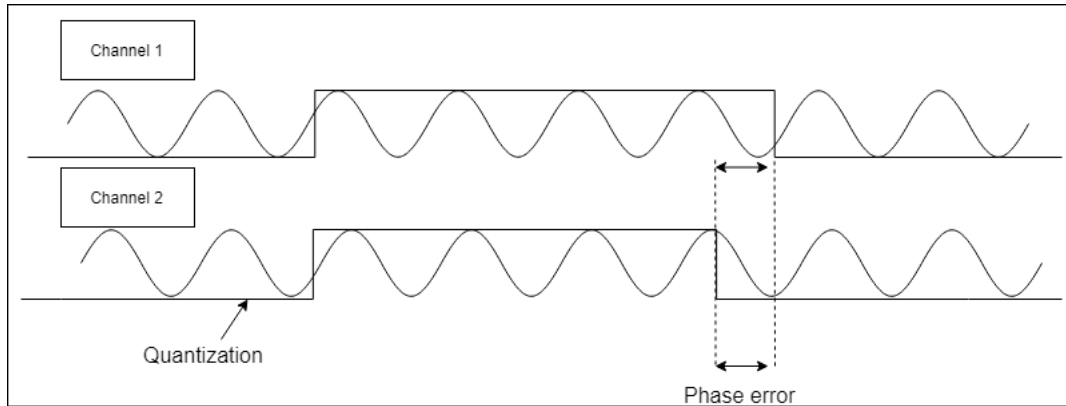


Figure 6.4: Illustration of the clock synchronisation phase error that can occur when the mixers of each channel in an array are not perfectly synchronized.

In order to account for the different phase errors an optimization program was run to find the optimum calibration weights for each channel. The calibration weights for each channel are the optimisation variables. The cost function of the minimisation problem was the L2 norm of the error between the MUSIC and MUSIC smooth functions and their references. The reference patterns are created with simulated data with no phase errors introduced and a local optimisation was run, using `fmincon` in Matlab, to match this pattern as closely as possible. The practical calibration was run with an antenna at the broadside to ensure the arriving wave approximates a plane wave and reaches each antenna almost simultaneously.

The images below, Figures 6.5, 6.6 and 6.7 show the various results of the optimization process. Figure 6.6 shows the MUSIC spectra of the initial data which can be seen to be heavily corrupted.

Occasionally the local optimizer does not present the output MUSIC spectra at  $0^\circ$ . When this is the case, a linear phase is added to move the spectra to  $0^\circ$ . The additional phase is defined as:

$$e^{[0:M-1]*j*\pi*\sin(\theta)} \quad (6.3)$$

where  $\theta$  is the angular error output from the local optimization.

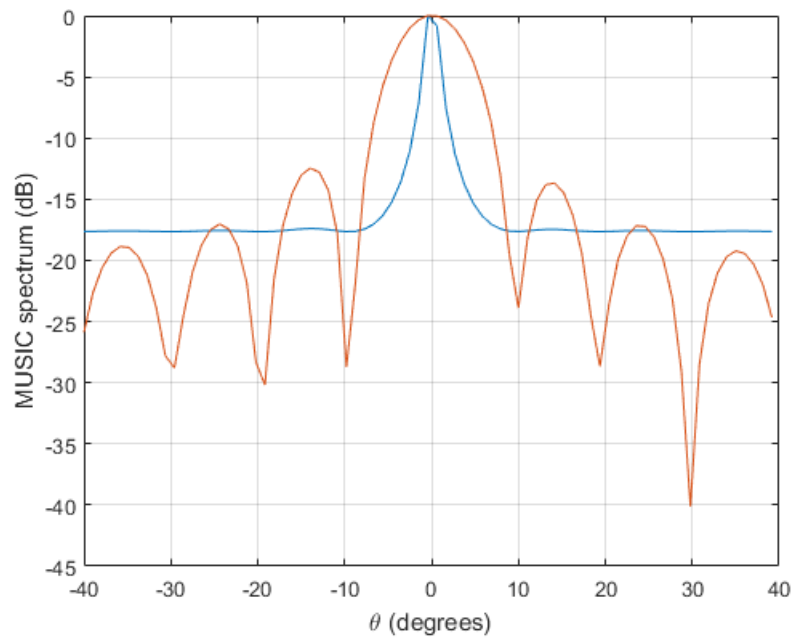


Figure 6.5: Phase reference patterns (MUSIC (blue) and MUSICSmooth (red) spectra) used to create the cost function for the optimizer. The pattern was created by running the MUSIC and MUSIC smooth algorithms with simulated data known to arrive at the array elements simultaneously.

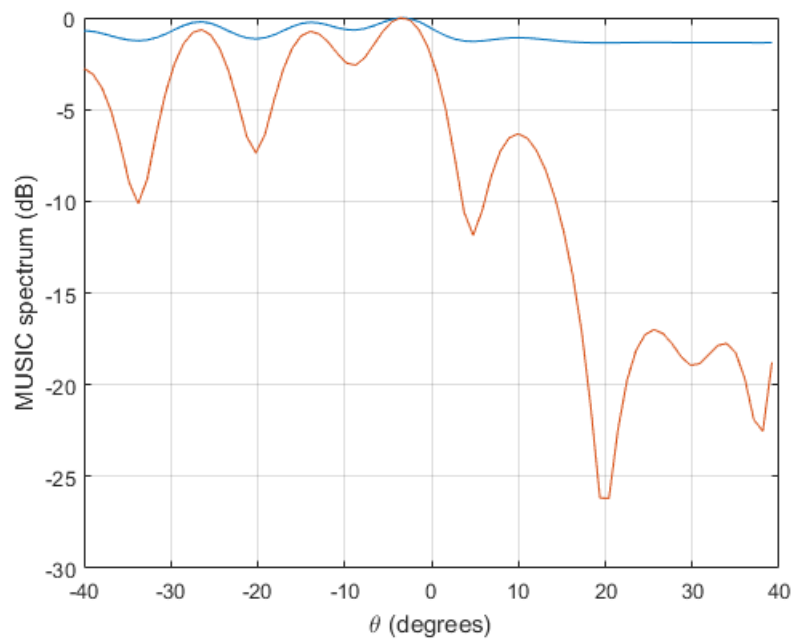


Figure 6.6: Patterns from the initial spectra, of the input data received from the TART system, prior to phase calibration.

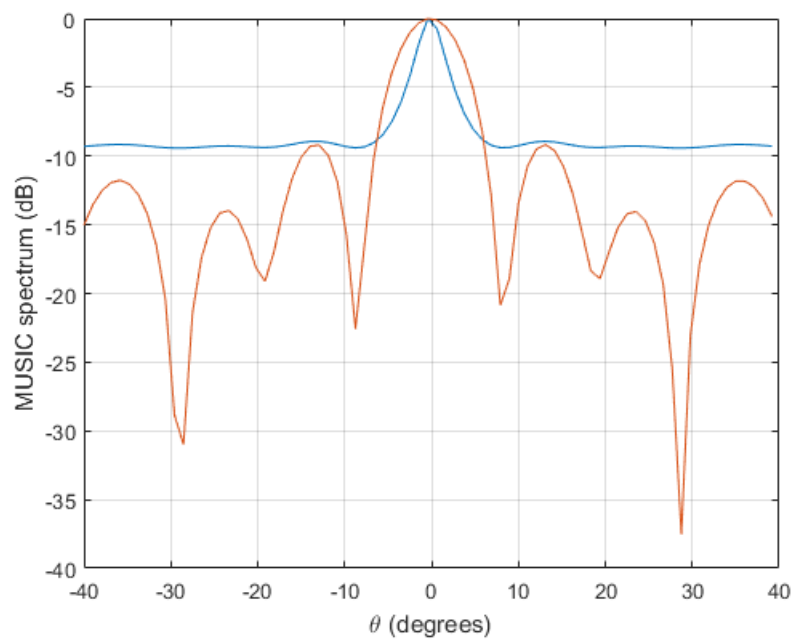


Figure 6.7: Image of the results produced by using the phase values given by the optimisation.

Figure 6.6 show the importance of phase calibration in order to ensure accurate practical results. Phase calibration may have to be performed each time practical results are collected due to cables not being phase stable. Rotating the array will shift the phase slightly. Another effect of moving the array is that the far field approximation can cause issues with the phase of the received signal at each element. The effect of movement of the array during tests should however be small and the results of the calibration should remain valid.



# Chapter 7

## DOA Algorithms

### 7.1 MUSIC

The multiple signal classification (MUSIC) [37] algorithm uses Eigen decomposition of an incoming signal covariance matrix in order to find the DOA of the signals. If  $[\lambda_1, \lambda_2, \lambda_3 \dots \lambda_N]$  are the Eigenvalues of the matrix  $\mathbf{R}$  and  $[\bar{\mathbf{e}}_1, \bar{\mathbf{e}}_2, \bar{\mathbf{e}}_3 \dots \bar{\mathbf{e}}_N]$  are the Eigenvector corresponding to those values, in a GPS application, the largest of these vectors will correspond to the interference signals while the rest of the signals correspond to the SOI+noise.

If  $[\lambda_1 \geq \lambda_2 \geq \lambda_3 \dots \lambda_N]$  then  $span[\bar{\mathbf{e}}_1, \bar{\mathbf{e}}_2, \bar{\mathbf{e}}_3 \dots \bar{\mathbf{e}}_M] \perp span[\bar{\mathbf{e}}_{M+1}, \bar{\mathbf{e}}_{M+2}, \bar{\mathbf{e}}_{M+3} \dots \bar{\mathbf{e}}_N]$  where  $N$  is the number of antenna elements.

$[\bar{\mathbf{e}}_{M+1}, \bar{\mathbf{e}}_{M+2}, \bar{\mathbf{e}}_{M+3} \dots \bar{\mathbf{e}}_N]$  are the SOI+noise Eigenvectors and can be stacked into a matrix,

$$\mathbf{P} = [\bar{\mathbf{e}}_{M+1}, \bar{\mathbf{e}}_{M+2} \dots \bar{\mathbf{e}}_N]. \quad (7.1)$$

The noise subspace Eigenvectors in the matrix  $\mathbf{P}$  are orthogonal to the steering vectors of the incoming interference signals and using the formula below the DOA's of the signals can be found because  $P_{MUSIC}(\theta)$  will peak when the steering vector of the signal is orthogonal to the Eigenvectors of the SOI+noise.

$$P_{MUSIC}(\theta) = \frac{1}{|\bar{\mathbf{a}}(\theta)^H \mathbf{P} \mathbf{P}^H \bar{\mathbf{a}}(\theta)|} \quad (7.2)$$

$\bar{\mathbf{a}}(\theta)$  is the steering vector of a signal with an angle of arrival of  $\theta$ .  $\theta$  is defined in Figure 3.1.

The music smooth algorithm [38] is another subspace based method that is similar to MUSIC. The algorithm produces smoother peaks and the wave form is easier to interpolate. The MUSIC (smooth) DOA spectrum is:

$$P_{Smooth}(\theta) = 1 - \frac{\bar{\mathbf{a}}(\theta)^H \mathbf{P} \mathbf{P}^H \bar{\mathbf{a}}(\theta)}{\bar{\mathbf{a}}(\theta)^H \bar{\mathbf{a}}(\theta)} \quad (7.3)$$

with  $\mathbf{P} = [\bar{\mathbf{e}}_{M+1}, \bar{\mathbf{e}}_{M+2} \dots]$ .

## 7.2 MVDR (Capon)

Minimum variance distortionless response (MVDR)[39] can be used as a DOA algorithm. The MVDR power spectrum is,

$$P_{MVDR}(\theta) = \frac{1}{|\bar{\mathbf{a}}(\theta)^H \mathbf{R}^{-1} \bar{\mathbf{a}}(\theta)|}, \quad (7.4)$$

where  $\bar{\mathbf{a}}(\theta)$  is the search steering vector and  $\mathbf{R}$  is the signal covariance matrix. The peaks in the MVDR spectrum will occur when the search steering vector is orthogonal to the subspace of the remaining signals.

This algorithm, like the MUSIC algorithm, does a search through each angle of theta specified and is therefore not computationally efficient. There are methods such as root music and ESPRIT [37] that are far more computationally efficient because they do not require an extensive power spectrum search.

## 7.3 Bartlett

The Bartlett [40] DOA algorithm is one of the earliest and simplest. The algorithm places a beam in the direction of a signal that creates the highest received power. From (2.4) and (3.9), the output power of the array is:

$$E \{ \mathbf{Y}^2 \} = \bar{\mathbf{w}}^H \mathbf{R} \bar{\mathbf{w}},$$

and the covariance matrix is,

$$\mathbf{R} = E[\mathbf{X}\mathbf{X}^H].$$

A solution to the weight vector is,

$$\bar{\mathbf{w}} = \frac{\bar{\mathbf{a}}(\theta)}{\sqrt{\bar{\mathbf{a}}(\theta)^H \bar{\mathbf{a}}(\theta)}}. \quad (7.5)$$

If the weight vector is inserted into the output power formula above then the power equation becomes:

$$P_{Bartlett}(\theta) = \bar{\mathbf{a}}(\theta)^H \mathbf{R} \bar{\mathbf{a}}(\theta) \quad (7.6)$$

The angles of the signals can be found by searching the peaks of the power spectrum above.

For further reference all of the DOA algorithms mentioned in Chapter 7 can be seen in [38], [41], [37], [6], [42], [43] and [44].

## 7.4 Simulation Results and Discussion

The DOA algorithms were tested using a variety of parameter sweeps. Each DOA spectrum was put through a peak search in order to find the relevant peaks. The DOA algorithms were put through 100 Monte Carlo trials to determine the success rate of each algorithm. The algorithms under investigation are Bartlett, MUSIC, MUSIC smooth and Capon. The algorithm was deemed successful if all 4 of the search angles are within  $5^\circ$  of the desired angle, as described in Chapter 6. The results below Figures 7.1 and 7.2 show the results of the tests mentioned in Chapter 6 Table 6.1 and show that Bartlett has the highest success rate.

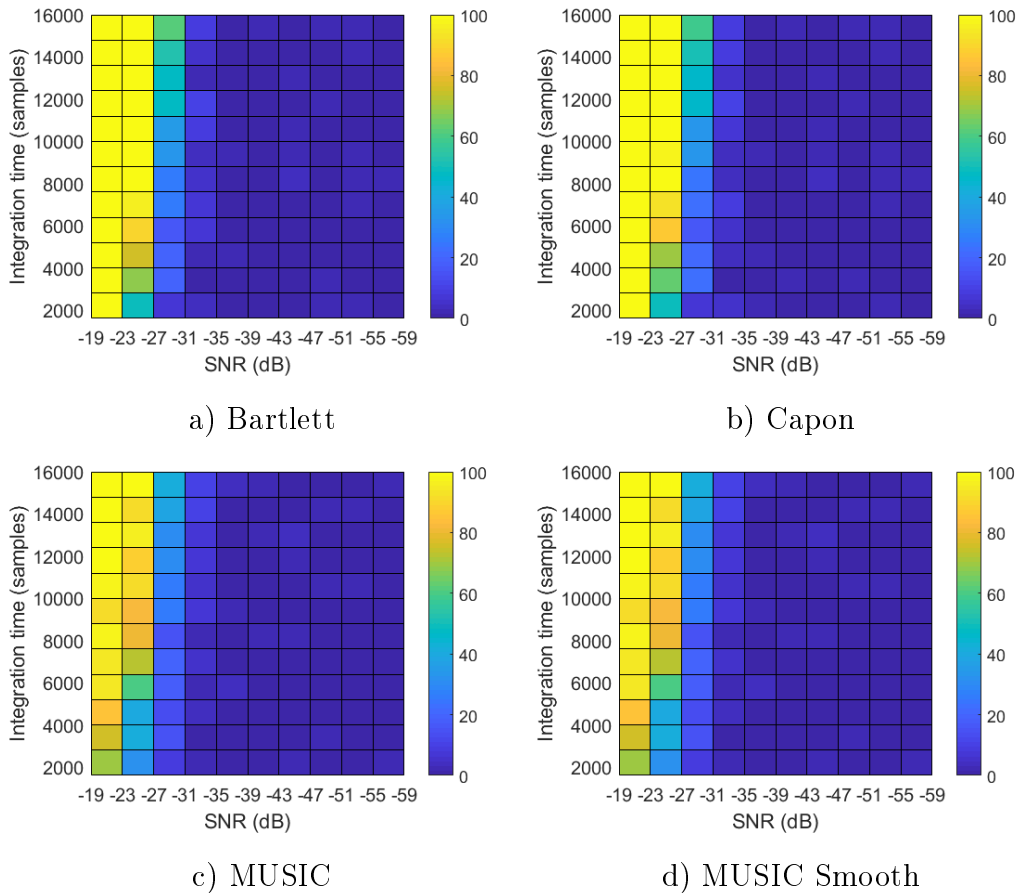


Figure 7.1: Surface plots showing the results the DOA algorithm simulation-Integration time vs SNR. The simulation was run using 12 bits, as explained in Section 6. The plots show the success percentage of each algorithm.

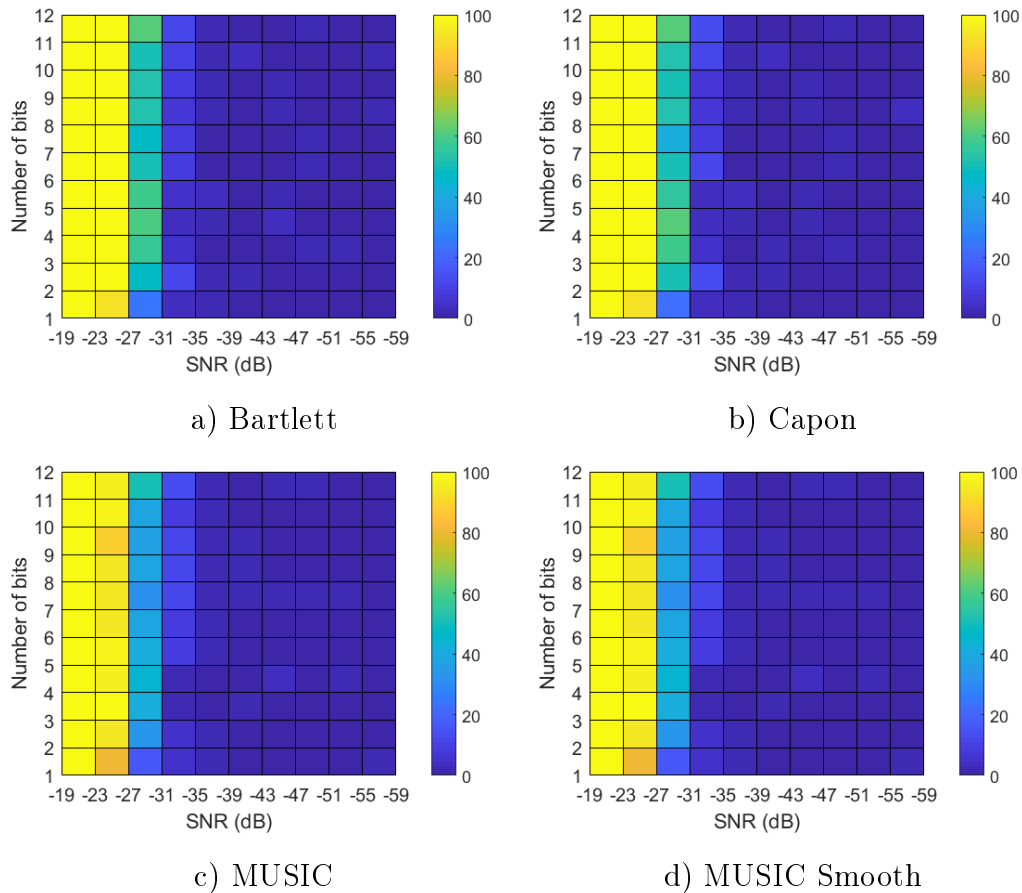


Figure 7.2: Surface plots showing the results the DOA algorithm simulation-Number of ADC bits vs SNR. The integration time = 16000 samples. The plots show the success percentage of each algorithm.

Figure 7.3 shows the effect of I/Q vs bit shifted data. The simulations were performed with 1 bit and an integration time of 16000 samples. The results show that the effect of bit shifted data is small and that bit shifted data is a viable alternative to I/Q data when the samples frequency is 4 times the IF.

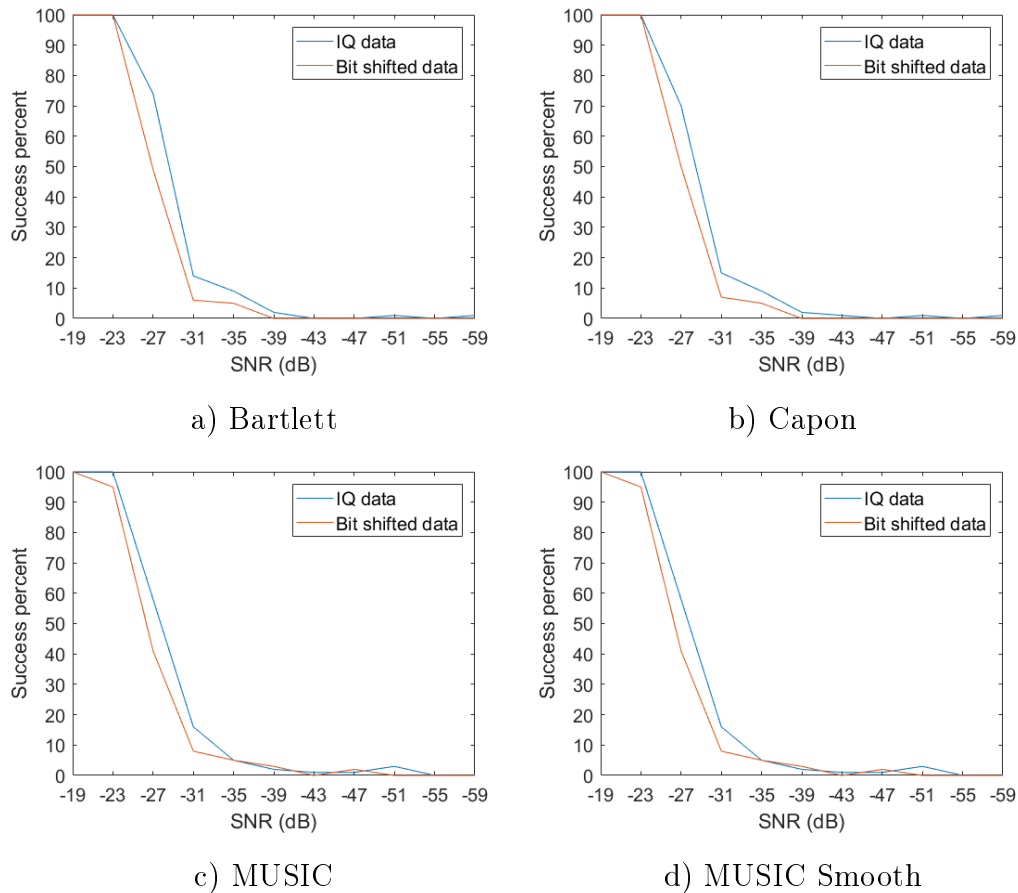


Figure 7.3: Graphs showing the results of the DOA algorithm simulation-I/Q & bit shift vs SNR. The integration time = 16000 samples. The plots show the success percentage of each algorithm.

The results show that the Bartlett algorithm performs the best, in these circumstances. This results may be due to the number of sources present. It has been shown that MUSIC delivers poor results when there are more than 2 signals present [45] and Capon has slightly better performance, whose results are similar to Bartlett in these simulations. The peak search method may also have been more beneficial with a smoother spectrum, such as Bartlett, rather than with a spectrum with sharp discontinuities, which is common with MUSIC. MUSIC has been reported to be more accurate than Bartlett and Capon however with the  $5^\circ$  method, Bartlett is superior.

# Chapter 8

## Proposed Algorithms

### 8.1 Adaptive Beamforming Method

After a null has been established in the direction of the interference signal, either using a beamforming or subspace technique as explained in Sections 2.6.3, 2.6.1, 2.6.2, 2.6.4 and 3.3, antenna gain should be established in the directions of the various GPS satellites in the receiver's field of view. This lowers the effect of multipath and increases SNR. Many of the methods used include the use of a reference signal. The need for a reference signal, to find the optimum weights for the antenna elements, lowers the degrees of freedom the array will have and is therefore undesirable.

The method proposed, algorithm 1, is a three-step algorithm. The first step is to create an input signal vector that is interference free using orthogonal projection. The second step is to create a subspace that is orthogonal to the remaining noise signals. The algorithm then compares the beampatterns of the different columns in the subspace and selects the pattern with the highest peaks at the SOI's and the lowest value at the interference, these are the weights used in beamforming.

[46], [47], [48] and [16] all use Eigenmode decomposition of the covariance matrix and make use of the orthogonality of the incoming signals. Orthogonal subspace creation is a common theme in these papers however the method in Section 8.2 finds the weight vector in a different manner.

### 8.2 Proposed Algorithm 1

The section provides details on the algorithm 1, the subspace beam forming technique.

From (3.11) the covariance matrix of  $\mathbf{X}$  can be represented by:

$$\mathbf{R} = E[\mathbf{X}\mathbf{X}^H] = \mathbf{R}_i + \mathbf{R}_l + \mathbf{R}_n = \sum_{i=1}^M \lambda_i \bar{\mathbf{e}}_i \bar{\mathbf{e}}_i^* + \sum_{j=M+1}^N \lambda_j \bar{\mathbf{e}}_j \bar{\mathbf{e}}_j^* + \sigma_n^2 \mathbf{I},$$

where  $\sum_{i=1}^M \lambda_i \bar{\mathbf{e}}_i \bar{\mathbf{e}}_i^*$  represents the interference Eigenvectors and values and  $\sum_{j=M+1}^N \lambda_j \bar{\mathbf{e}}_j \bar{\mathbf{e}}_j^*$  represents the SOI+noise Eigenvectors and values. Due to the magnitude of the power of the interference signals compared to the noise and GPS signals, the covariance matrix is dominated by the Eigenvalues of the interference. After a SVD of  $\mathbf{R}$ , a subspace,  $\mathbf{V}$  can be created that is orthogonal to the interference subspace and the input data can be projected onto the subspace as mentioned in Section 3.3. The orthogonal subspace is given by (3.12)

$$\mathbf{V} = \mathbf{I} - \mathbf{U}\mathbf{U}^\dagger,$$

where  $\mathbf{U} = [\bar{\mathbf{e}}_1, \bar{\mathbf{e}}_2, \bar{\mathbf{e}}_3 \dots \bar{\mathbf{e}}_M]$  then the new input data matrix, that is interference free, is  $\tilde{\mathbf{X}}$  and is given by (3.13).

$$\tilde{\mathbf{X}} = \mathbf{X}^H \mathbf{V},$$

and the new covariance matrix;

$$\tilde{\mathbf{R}}_{xx} = E[\tilde{\mathbf{X}}\tilde{\mathbf{X}}^H]. \quad (8.1)$$

Now using the assumption that the signal and noise are orthogonal, the antenna weights can be found.

$$\bar{\mathbf{W}} \perp \bar{\mathbf{a}}_k, \quad (8.2)$$

where  $\bar{\mathbf{a}}_k$  spans the noise subspace. This is accomplished by creating a subspace orthogonal to the noise,  $\tilde{\mathbf{V}}$ , as done previously with  $\mathbf{R}$ , and using an arbitrary column of the subspace. A possible drawback of this algorithm is that one is required to know the number of desired signals.

The algorithm works as follows:

1. Determine the covariance matrix of sampled data and perform singular value decomposition (SVD of  $\mathbf{R}_{xx}$ ).
2. Estimate the number of interference signals (Section 3.5).
3. Sort Eigen vectors, the highest Eigen vectors correspond to the interference subspace 1-M and partition ( $\mathbf{U} = [\bar{\mathbf{e}}_1, \bar{\mathbf{e}}_2, \bar{\mathbf{e}}_3 \dots \bar{\mathbf{e}}_M]$ ).
4. Create a subspace orthogonal to this subspace,  $\mathbf{V}$  using (3.12), see Section 3.3.
5. Project input data,  $\mathbf{X}$  onto this subspace and create new covariance matrix  $\tilde{\mathbf{R}}_{xx}$ .
6. Find the noise subspace of the interference free data,  $\mathbf{G} = [\bar{\mathbf{e}}_L, \bar{\mathbf{e}}_{L+1}, \bar{\mathbf{e}}_{L+2} \dots \bar{\mathbf{e}}_N]$ .
7. Create a subspace orthogonal to this subspace,  $\tilde{\mathbf{V}}$  using (3.12).

8. Antenna weights are selected as a column of the orthogonal subspace.

A typical assumption for the signal powers is -122 dBm for the GPS signals, -110 dBm for the noise and varying interference signal powers. The average signal power of the input data block should not be above the noise floor, which will be the threshold for interference detection. If the input power is above this threshold it should indicate an interference signal. This is a very simple way to detect the presence of interference signals, which should have more power than the thermal noise floor. To the best of my knowledge this is a new beam forming technique.

## 8.2.1 Simulation Results and Discussion

### 8.2.1.1 No Interference

Algorithm 1 was tested without interference and the power levels of the GPS signals are all -129 dBm and the noise floor of -110 dBm. Figure 8.1 shows the beam pattern for the given power levels with 16000 samples and 1 bit shifted data. Algorithm 1 is labelled subspace beam forming and it has been compared to LCMV. All of the test are performed on a simulated ULA along the  $x$ -axis therefore  $\phi = 0^\circ$ .

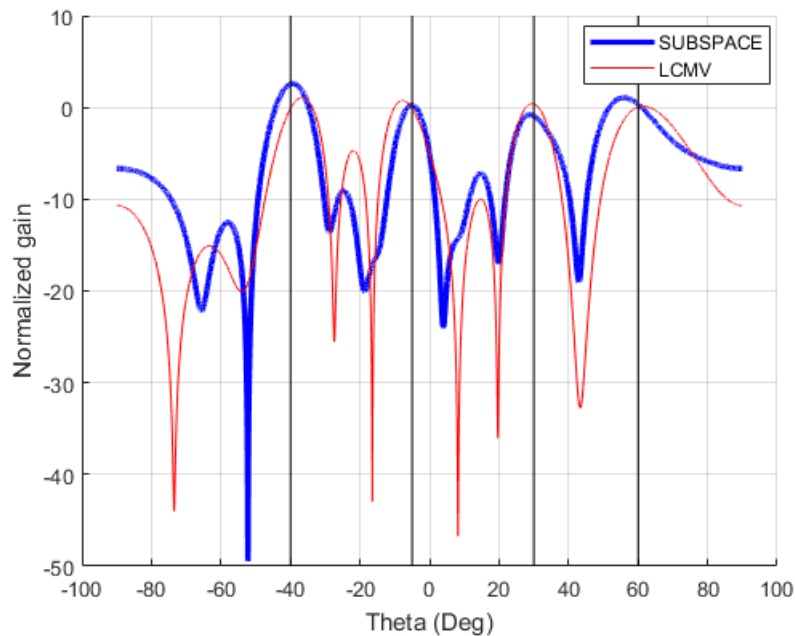


Figure 8.1: Beam pattern produced by algorithm 1 (blue) vs LCMV (red) with no interference using 1 bit shifted data. The SOIs are at  $60^\circ$ ,  $30^\circ$ ,  $-5^\circ$ ,  $-40^\circ$ .



Algorithm 1 does not need to suppress interference therefore the initial orthogonal subspace created is the identity matrix. Figure 8.1 clearly shows the desired angles with vertical black lines. The peaks of the beamformers fall within the success margin of  $5^\circ$ . The results show that algorithm 1 is a viable beamformer for GPS applications when no interference is present.

### 8.2.1.2 Interference

The interference included during the testing of algorithm one was 2 MHz filtered noise with a power level of -110 dBm. SOI power levels at -122, -123, -123, -121 dBm. The noise floor was set to -105 dBm and Figure 8.2, shows the results for 1 bit shifted data. A comparison of subspace, nullsteering and LCMV are shown. The power of the interference cannot be much higher than -110 dBm for 1 bit data.

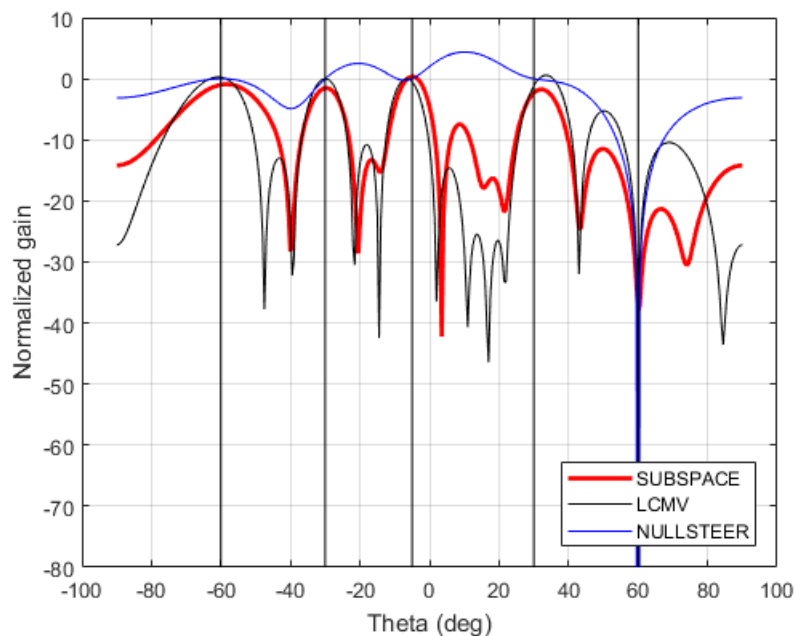


Figure 8.2: A graph showing the beampattern produced by algorithm 1 (red), LCMV (black), nullsteering (blue) with interference. The SOIs are at  $-60^\circ$ ,  $-30^\circ$ ,  $-5^\circ$ ,  $30^\circ$  and interference at  $60^\circ$ . The algorithm was run using 1 bit shifted data

The image shows that the antenna gain and null have been established in the directions of the DOA's and interference respectively. The largest SNR of the signals was a typical -18 dB and the largest SIR was -13 dB. The null in the direction of the interference is  $\approx -35$  dB. The SIR is rather large for this test however this is set at this level because the ADC is utilising 1 bit.

From the beampattern in Figure 8.2, algorithm 1 is a viable blind beamforming technique and can be tested on the TART system.

After testing with 1 bit shifted data, the number of bits was increased to 12 and the algorithm was seen to handle far more interference power as explained in Section 3.7. Figure 8.3 shows the beampattern using the same SOI power levels as used in Figure 8.2 however the interference level is increased to -90 dBm.

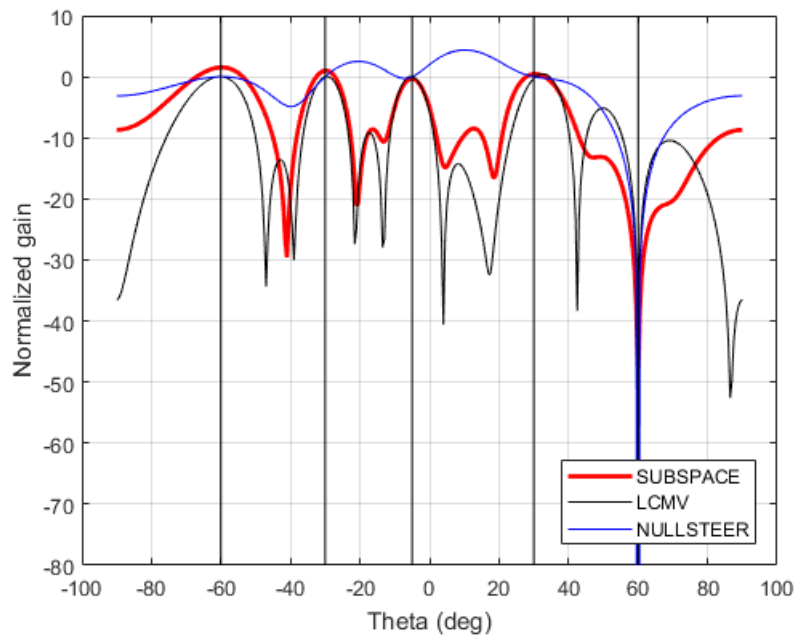


Figure 8.3: A graph showing the beampattern produced by algorithm 1 (red), LCMV (black), nullsteering (blue) with interference. The SOIs are at  $-60^\circ$ ,  $-30^\circ$ ,  $-5^\circ$ ,  $30^\circ$  and interference at  $60^\circ$ . The algorithm was run using 12 bit shifted data

Figures 8.4 and 8.5 show the results of simulations run on algorithm 1 while using 1 bit shifted data and varying SIR, integration time and SNR. The purpose of the test is to determine the possibility of using algorithm 1 on the TART system.

The results show that, with the success definition of algorithm 1, it is more sensitive than algorithm 2 when using 1 bit shifted data. The success rates of both tests are clearly lower than those in Figures 8.12 and 8.13. The results also show that algorithm 1 will function as a blind beamformer on the TART system if the SNR and SIR are above -25 dB and the number of samples is  $\approx 16000$ .

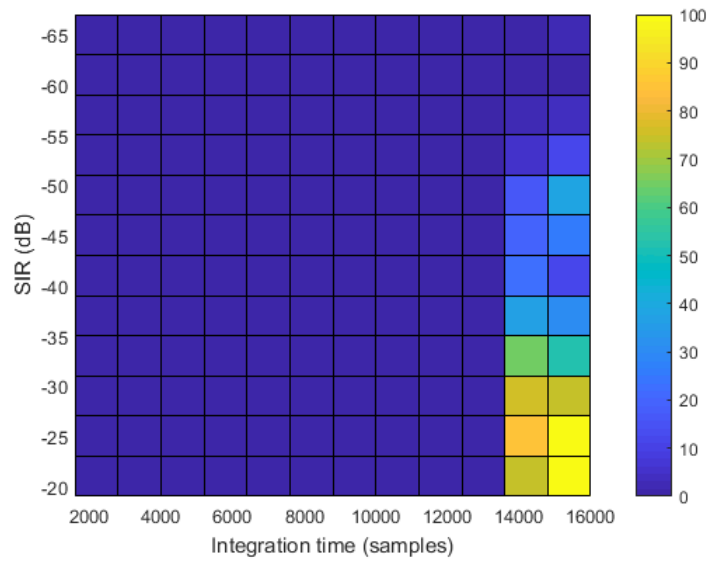


Figure 8.4: Surface plot showing the success percent of algorithm 1, using 1 bit shifted data, while varying SIR and Integration time

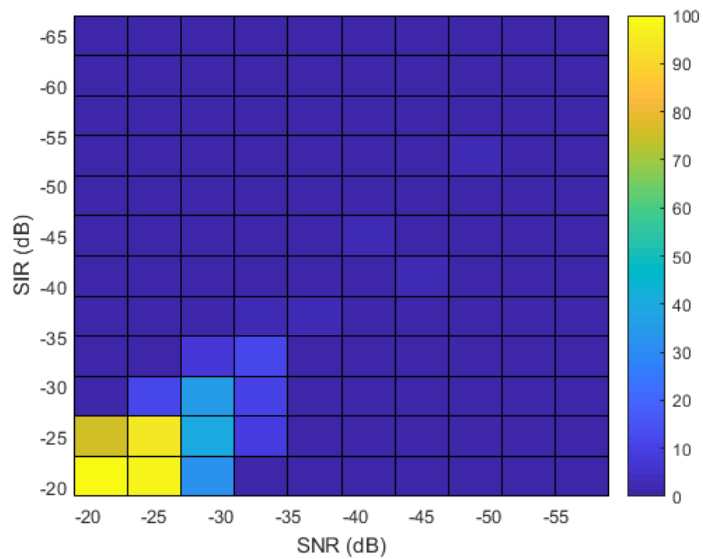


Figure 8.5: Surface plot showing the success percent of algorithm 1, using 1 bit shifted data, while varying SNR and SIR

Algorithm 1 was tested using the parameter variations mentioned in Chapter 6 Table 6.2. The success percent of each simulation can be seen in Figure 8.6 and a comparison of algorithms 1 and 2 can be seen in Figure 8.16.

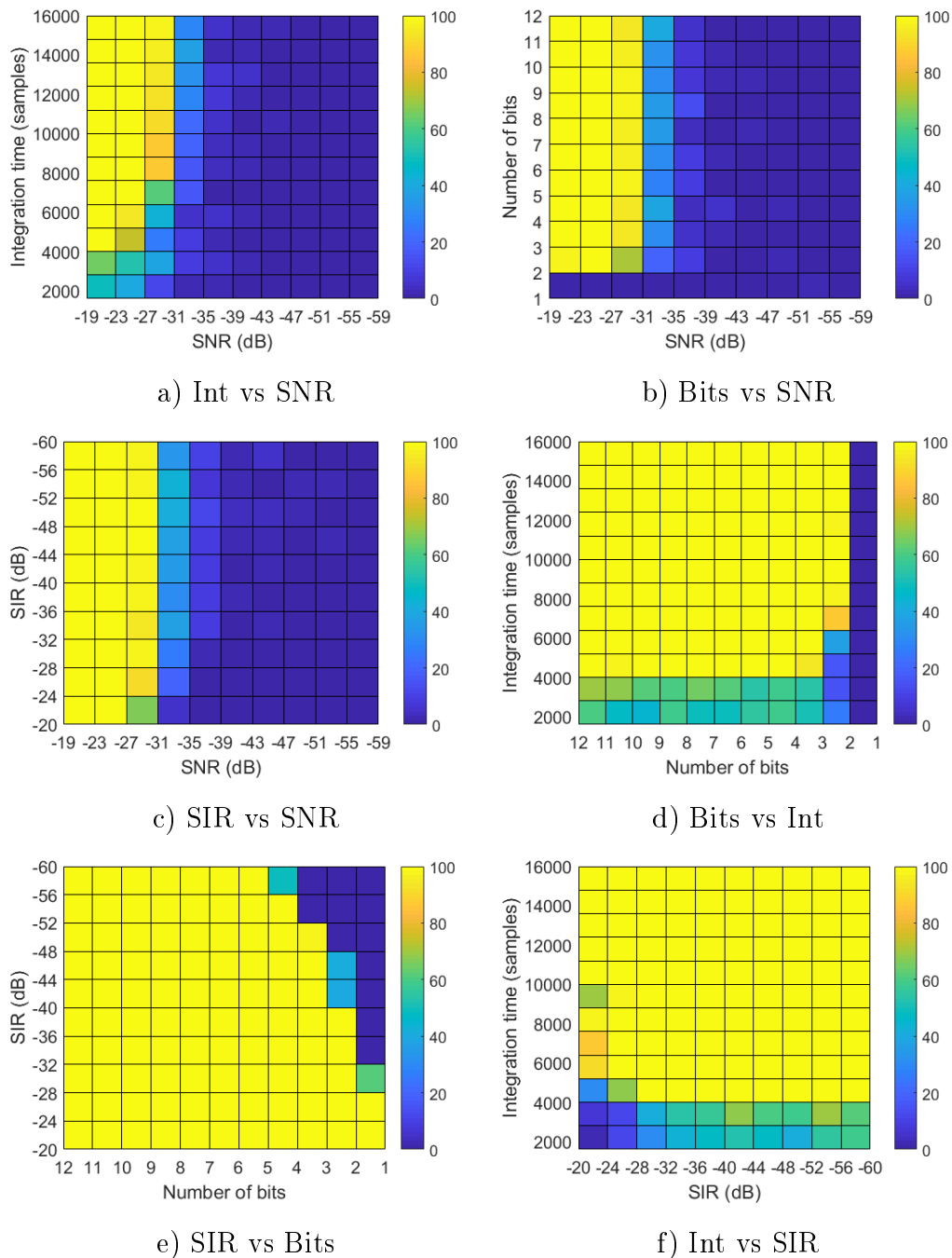


Figure 8.6: Image showing the success percent of all the simulations run with algorithm 1, which were detailed in Section 6.

Figure 8.6d shows that algorithm 1 will not work with an SIR of less than -30 dB unless the number of ADC bits is increased to at least 3. The figure also shows that the algorithm requires more than 2000 samples with a SNR of -15 dB.

Figure 8.6e shows that the algorithm functions with 1 bit as long as the SIR is greater than -28 dB and if the ADC has more than 5 bits the algorithm can suppress more than 60 dB of interference. Figure 8.6a shows that the SNR cannot be above -30 dB regardless of the number of samples taken. The level of the SIR is the reason the algorithm has no success with 1 bit data however 8.6e shows that if the SIR is risen to -28 dB the algorithm has 100% success. Figure 8.7, shows the results of the test to investigate the sensitivity of algorithm 1 to I/Q and bit shifted data.

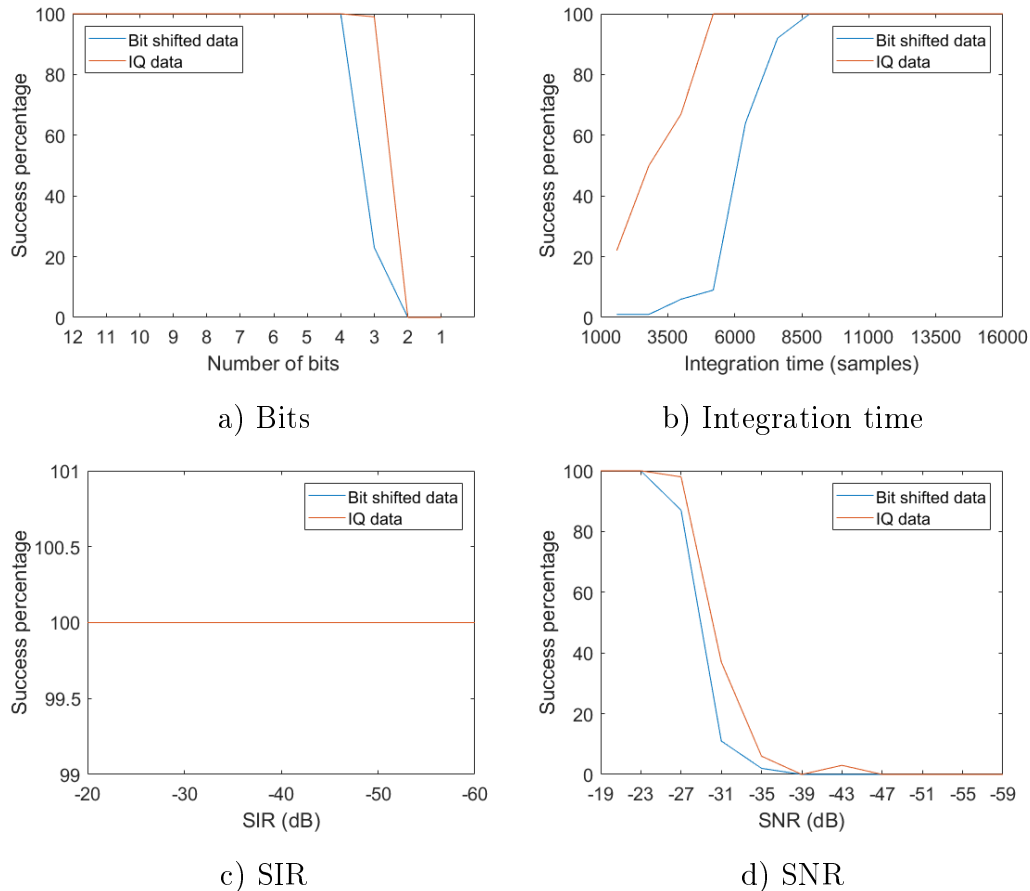


Figure 8.7: Graphs showing the success percent of algorithm 1, with various parameter sweeps, checking the difference between I/Q and bit shifted data.

The results in Figure 8.7 show that algorithm 1 is more sensitive to changes between I/Q and bit shifted data when compared to algorithm 2. Algorithm 1 is most sensitive to integration time. If the number of samples collected is a limiting and fixed number then it is beneficial to use I/Q data when using algorithm 1. However this means that more connections will need to be implemented between the receiver and digital backend to account for the Q channel of each antenna element, which increases the system complexity.

### 8.3 Proposed Algorithm 2

This DBF GPS algorithm makes the assumption that there is an interference signal present, which can be verified using one of the threshold techniques mentioned in Section 3.5. The algorithm makes use of MUSIC (Section 7.1) to find the interference signal and creates beams, using LCMV (Section 2.6.4), that search the remaining angular space of the array. The algorithm finds the highest correlation peak after searching through all satellite C/A codes via circular convolution (see Section 2.5) and the highest peak gives the DOA of the signals because correlating a C/A code with different C/A code produces low correlation. After the directions of the interference and SOI are found a beam and null are formed for the highest SINR. Both of the proposed algorithms are blind techniques. Algorithm 2 is a 4D search through the signal code phases, Doppler frequency shifts, C/A code numbers and beam angles. Circular convolution acquisition is effectively performed at each beam angle for each C/A code. This technique will work for coherent interference and SOI signals if the threshold detection techniques mentioned in Section 3.5 are used. If the sources are approximately collinear then nulling of the interference signal will of course degrade the SOI's.

The resolution of the search is determined by the number of antennas used in the array, more antennas will provide a sharper beampattern and therefore the DOA of the SOI can be found more accurately. A greater number of antennae will increase array complexity and algorithm run time. For simplicity the antennae used in simulation have omni-directional beampatterns therefore the 3 dB beam width of the array is determined by the array factor (AF) alone. The 3 dB beam width can be seen below in Figure 8.8. It shows that the array produces a beam of approximately 8° broadside. A formula for the beamwidth of a ULA is given by,

$$\Delta = \frac{\lambda}{D},$$

where  $D$  aperture of the array and  $\lambda$  is the wavelength. This formula gives a 3 dB beamwidth of 9.54°.

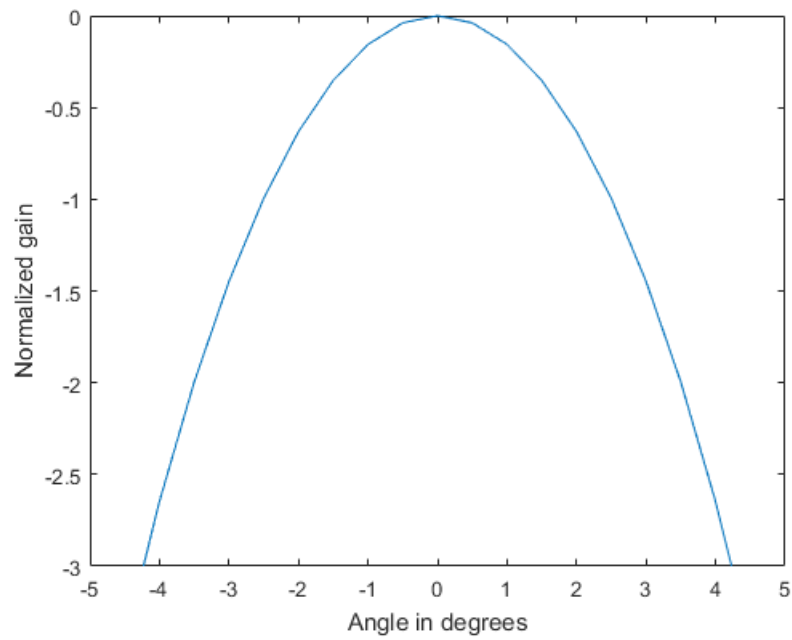


Figure 8.8: Diagram showing the 3 dB beam width of a 12 element array.

If the beam angle is chosen to be  $9^\circ$  then to cover the  $180^\circ$  search space, 20 beams should be created, however one of those beams will be a null in the direction of the interferer. An example of the search space can be seen below in Figure 8.9. The MUSIC spectrum and null can be seen as well as the search beams in various colors.

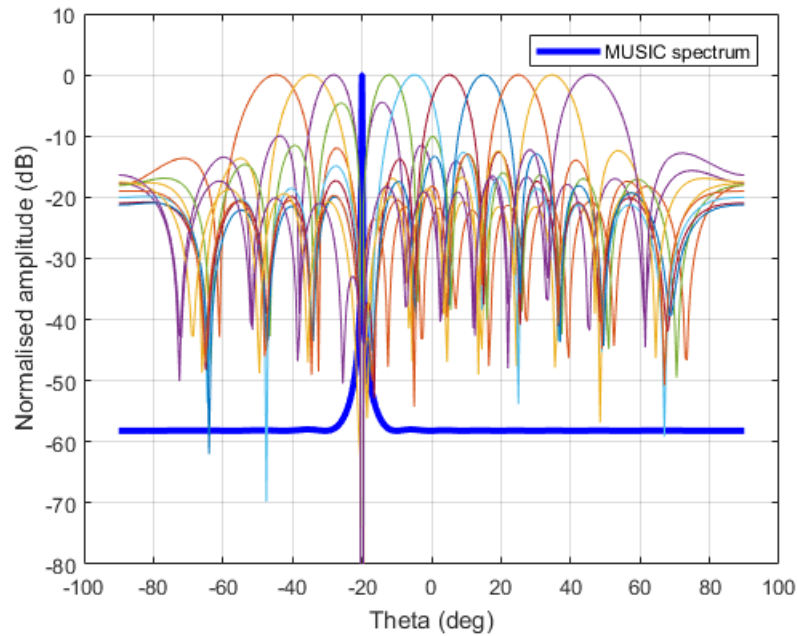


Figure 8.9: Diagram of search space used for algorithm 2, showing the multiple beams searching from  $-45^{\circ}$ - $45^{\circ}$ . Each beam has its own null at the interference,  $-20^{\circ}$ , which is represented by its MUSIC spectrum.

The algorithm works as follows:

1. Determine the covariance matrix of sampled data and perform singular value decomposition (SVD of  $\mathbf{R}_{xx}$ ).
2. Estimate the number of interference signals (Section 3.5).
3. Sort Eigen vectors, the highest Eigen vectors correspond to the interference subspace 1-M and partition ( $\mathbf{U} = [\bar{\mathbf{e}}_1, \bar{\mathbf{e}}_2, \bar{\mathbf{e}}_3 \dots \bar{\mathbf{e}}_M]$ ).
4. Use the MUSIC algorithm (7.2) to find the DOA of the interference and store this direction.
5. Define a search resolution and create beams to search the array space not occupied by the interference. The interference in each beam should be minimized.
6. Perform circular convolution on the incoming data from each beam and record the correlation peak height.
7. Check that the highest correlation peak meets a desired threshold to ensure the existence of valid data.

This technique is simpler conceptually however is more computationally expensive.



### 8.3.1 Simulation Results and Discussion

Algorithm 2 makes use of MUSIC to find the interference signal and places a null in its direction. It also establishes multiple beams that search the remaining angular space of the array, each with the aforementioned null. The algorithm finds the highest correlation peak after searching through all the satellite PRN codes via circular convolution, and the highest peak gives the DOA of the signals.

As mentioned in Chapter 6, multiple simulations were run to test this algorithm. This section shows the most telling results from the simulations. Figures 8.10 and 8.11 show the clear improvement in the ability to handle interference when moving from a single antenna, to simple beam steering to LCMV beamforming even when using 1 bit shifted data.

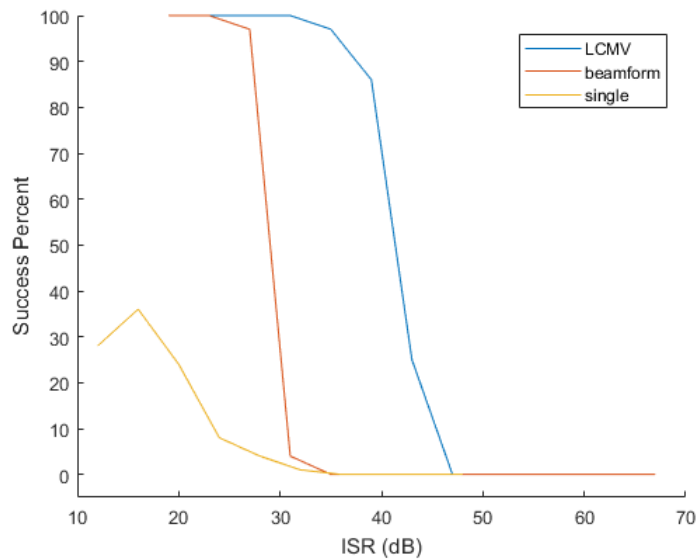


Figure 8.10: A graph comparing the success percent of Algorithm 2, using 1 bit shifted data, with LCMV, beam steering and single antenna with SNR = -27 dB and varying SIR.

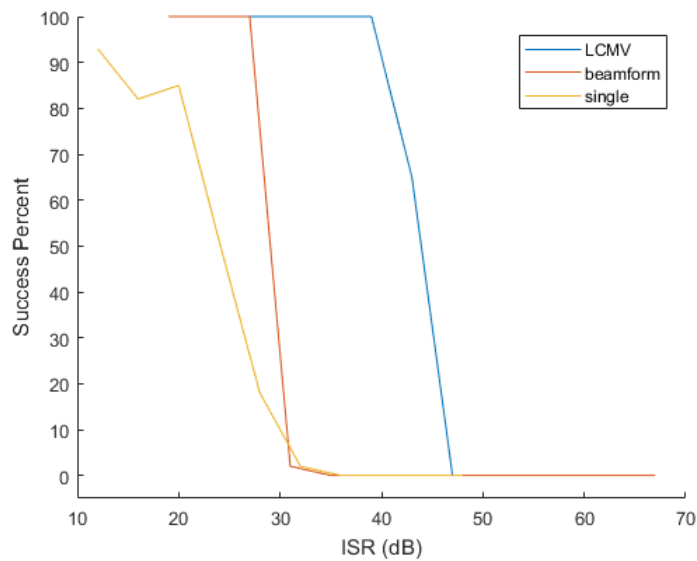


Figure 8.11: A graph comparing the success percent of Algorithm 2, using 1 bit shifted data, with LCMV, beam steering and single antenna with SNR = -23 dB and varying SIR.

The following results are simulations of algorithm 2 with LCMV beamforming with 1 bit shifted data. The parameters that were changed were integration time, SNR and SIR. Figure 8.12 shows the success percentage of the first simulation.

The results show that 9000-10000 samples are required for 100% success rate with a SNR and SIR of -19 dB and -39 dB respectively. Figure 8.12 also shows that with one period of C/A code, which will be  $\approx 16000$  samples at 16 MHz, algorithm 2 can mitigate  $\approx 50$  dB of interference with an SNR -19 dB using 1 bit shifted data.

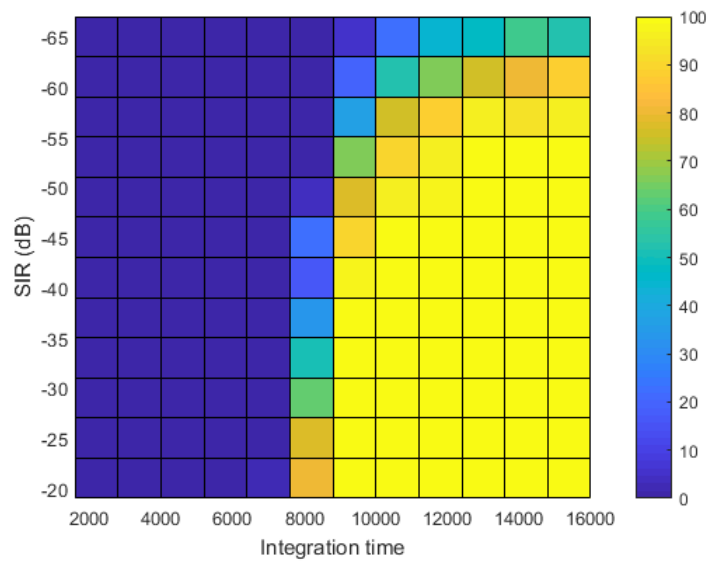


Figure 8.12: Surface plot showing the success percent of algorithm 2, using 1 bit shifted data, while varying SIR and Integration time

Figure 8.13 below shows the results from the SIR vs SNR test which show that algorithm 2 is not successful with more than 39 dB of SIR and -27 dB of SNR with 1 bit shifted data and 16000 samples.

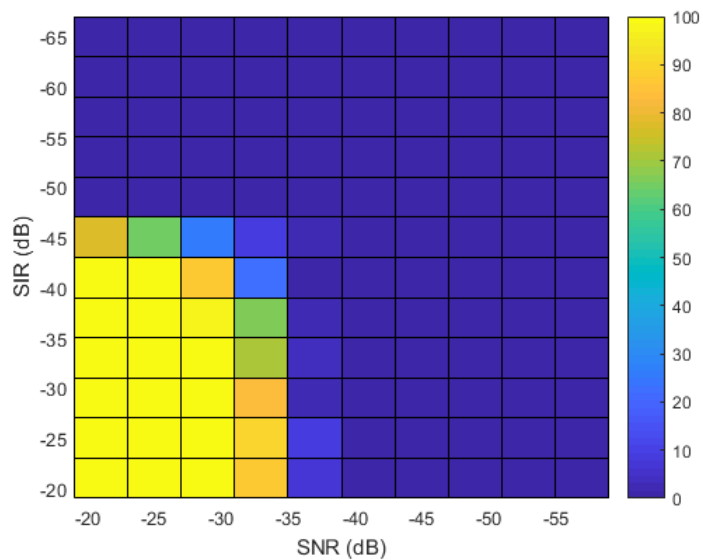


Figure 8.13: Surface plot showing the success percent of algorithm 2, using 1 bit shifted data, while varying SNR and SIR

These results show that with typical power levels of -129 dBm and -110 dBm for the GPS signal and noise floor respectively, algorithm 2 has 100% success rate with a SIR of -23 dB with 1 bit shifted data. These results show that the TART system will work as a beamforming system.

The next simulations show the performance of algorithm 2 with varying parameter values. The variables that are fixed are set to their ideal states i.e. SNR = -19 dB, No. bits = 12, however the SIR is kept low at -69 dB, showing the effect of SFDR.

Figure 8.14a shows the results of varying integration time and SNR with 12 bits and SIR of -69 dB. The figure shows that algorithm 2 can handle a small SIR of -69 dB while still removing 27 dB of noise which shows the benefit of the LCMV null. Figure 8.14c shows that algorithm 2 has 100% success, with the SIR and SNR mentioned previously, with 10000 samples. Figure 8.14b reiterates the effect of SFDR. Algorithm 2 cannot use less than 8 bits with an SIR of -69 dB which is close to the 8 dB/bit of SFDR mentioned in [49]. Figure 8.14e shows that if the SIR is larger, only 1 bit is required for 100% success. Figure 8.14c shows that algorithm 2 can suppress a large amount of interference, due to the null and high SFDR with 12 bits, however as the noise is increased the algorithm fails because the LCMV null has no effect on the noise of the system. The figure shows that algorithm 2 can suppress 60 dB of interference and a -27 dB SNR while still performing at 100% success. Figure 8.14 shows the results of all 6 simulations with algorithm 2. Figure 8.14e shows that algorithm 2 has 100% success rate, with low SIR until the number of ADC bits falls below 4. The image also shows that 1 bit shifted data can mitigate  $\approx 40$  dB of interference. Figure 8.14c shows the effect of the LCMV null. As the interference increases, the success rate remains steady however when the SNR is too high the algorithm fails. Figure 8.14a shows that with an SNR of -20 dB, algorithm 2 requires more than 3000 samples.

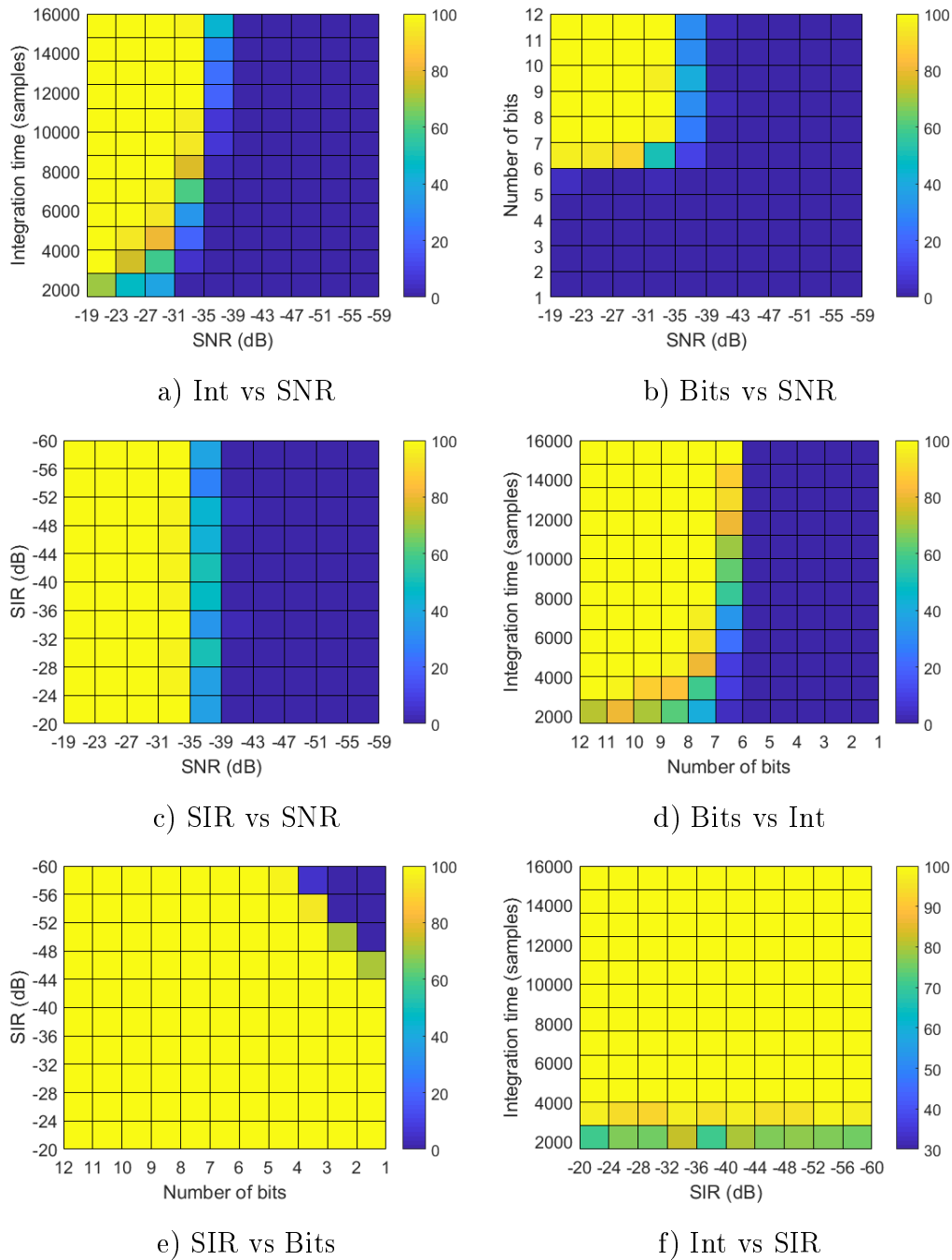


Figure 8.14: Image showing the success percent of all the simulations run with algorithm 2 using LCMV which were detailed in Section 6, Table 6.2.

The following tests investigate the effect of 1 bit shifted data while varying certain parameters. The first test investigates the success percent of the algorithm while varying the number of ADC bits. The SIR is set to -69 dB to investigate the effect of SFDR. Figure 8.15, shows the small effect of bit

shifted data when compared to I/Q data which confirms the results seen in Figure 7.3.

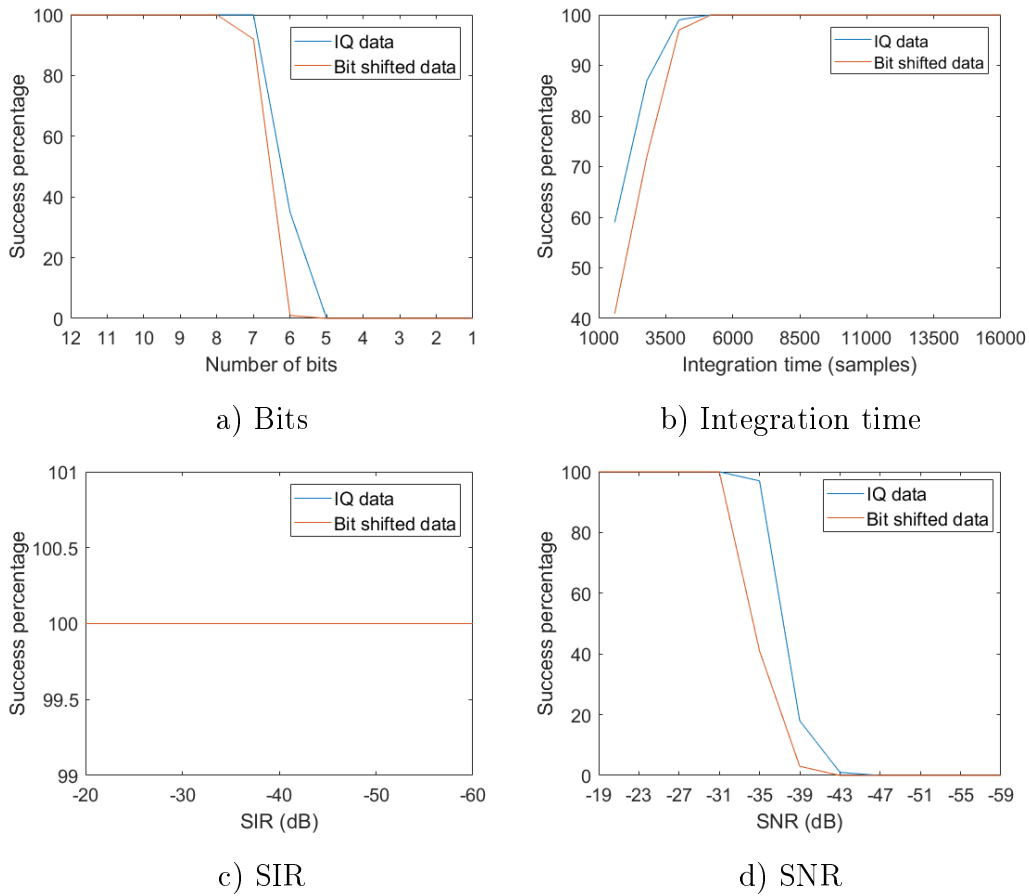
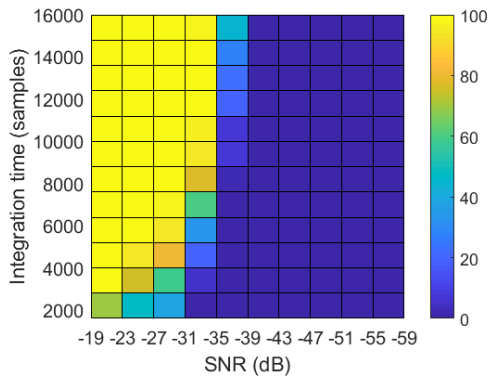


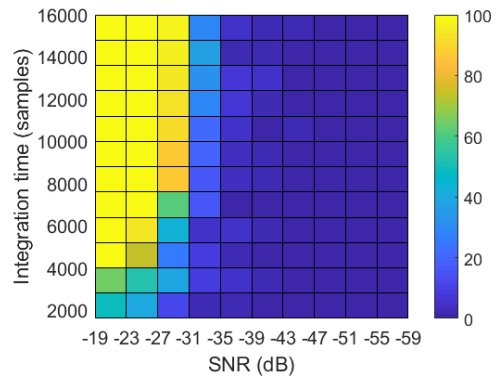
Figure 8.15: Graphs showing the success percent of algorithm 2, with various parameter sweeps, checking the difference between I/Q and bit shifted data.

I/Q data clearly has better performance however the success rate of the bit shifted data is also high. If there is a large SNR then the bit shifted data fails much sooner than I/Q data, as seen in Figure 8.15d due to the extra noise added by the bit shifting process. The results above prove the viability of the TART system as a GPS beamformer for algorithm 2.

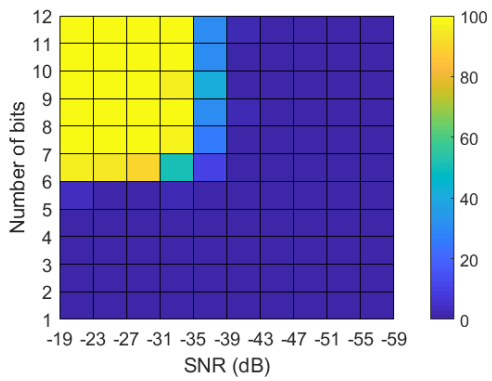
Figure 8.16 shows a side by side comparison of the results, from algorithms 1 and 2, for easy viewing and analysis. It shows that the performance of algorithms 1 and 2 are similar when undergoing the same parameters changes.



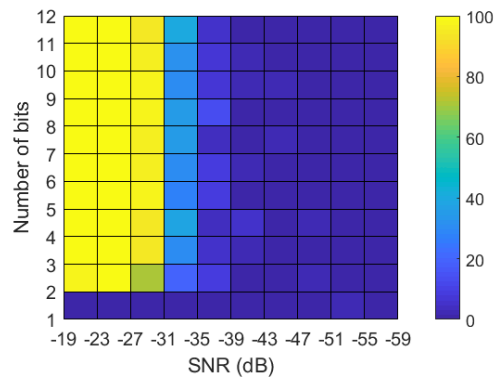
a) Int vs SNR (alg:2)



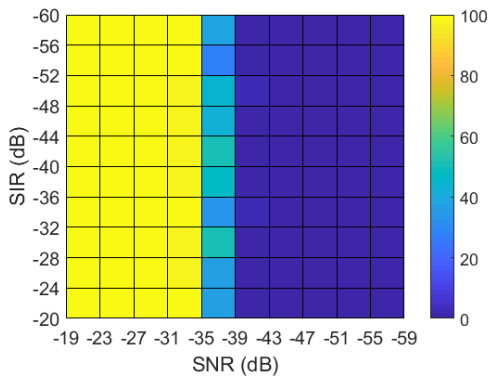
b) Int vs SNR (alg:1)



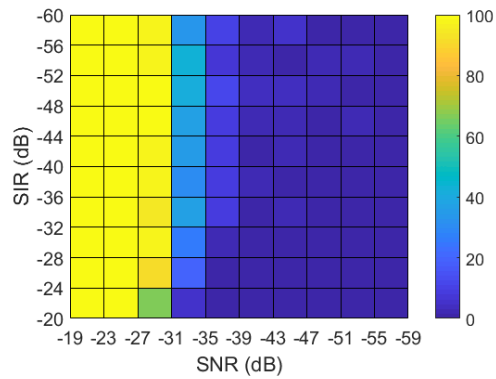
c) Bits vs SNR (alg:2)



d) Bits vs SNR (alg:1)



e) SIR vs SNR (alg:2)



f) SIR vs SNR (alg:1)

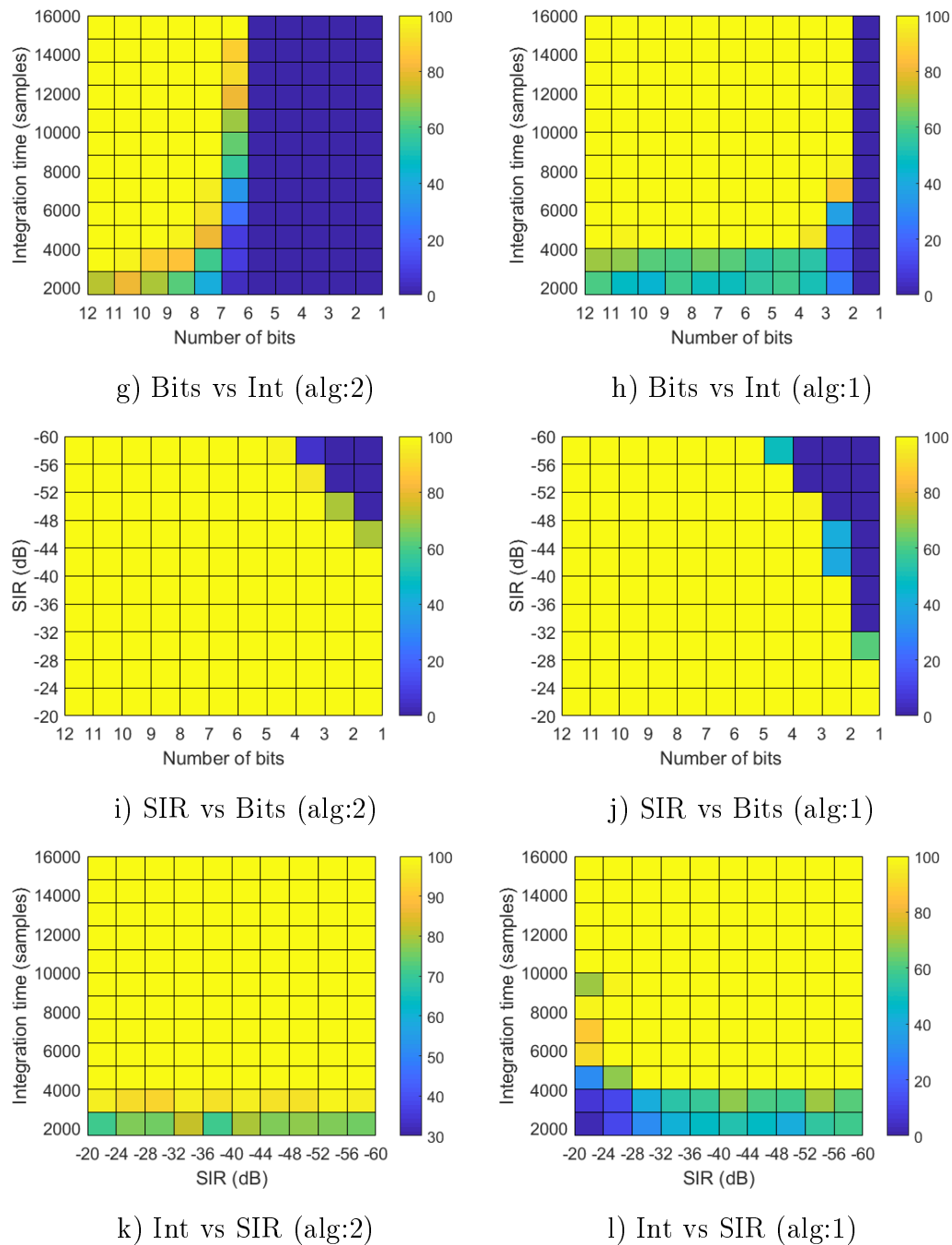


Figure 8.16: Image showing a side by side comparison of the success percent of all the simulations run with algorithm 2 using LCMV and algorithm 1, which were detailed in Chapter 6, Table 6.2.

The deep null produced by the LCMV algorithm was the reason for the differences of the graphs between algorithms 1 and 2. The null was produced in simulation because effects like unequal cables lengths, unstable oscillators



and a non-plane phase front where not accounted for in simulation, giving a near perfect null. The simulations above were run to investigate the failure points of the algorithms and in order to account for the deep null, the SIR was decreased to -69 dB as opposed to -30 dB for algorithm 1, as mentioned in Chapter 6. The effect of the decrease in SIR can be seen in Figures 8.16c, 8.16d, 8.16g and 8.16h. The increase in interference level demands a larger number of bits for a successful simulation.

## 8.4 Practical Results

The practical data was collected using the setup shown in Figure 6.1. The TART system was used to collect preliminary results, thereafter the system became unusable. The graphs shown below, Figures 8.17, 8.18, 8.19, 8.20 and 8.21 show the results of various tests using two SOIs. The DOA spectra and the results of algorithm 1 are shown and normalized to their peak amplitudes. 1 signal is at  $0^\circ$  and the initial angle of separation between the 2 SOIs is  $-\arctan\left\{\frac{10.5}{36}\right\} = -16.26^\circ$ . 16000 samples were used for each test.

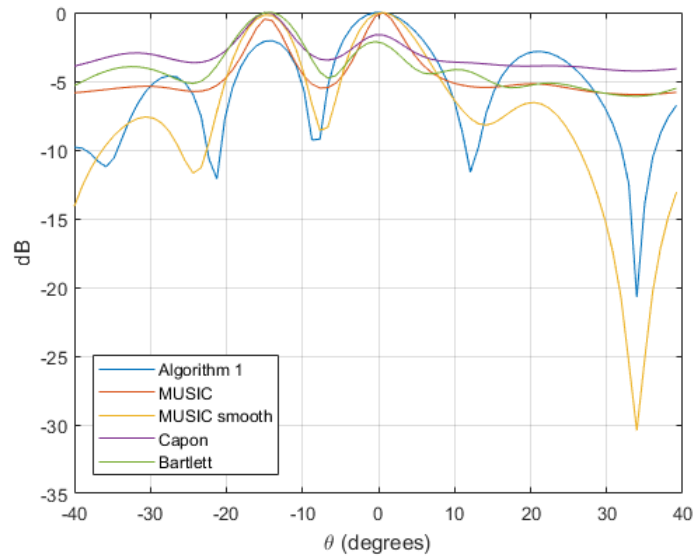


Figure 8.17: DOA spectra of the practical data collected from TART. The test has two SOIs, each with the same power level.

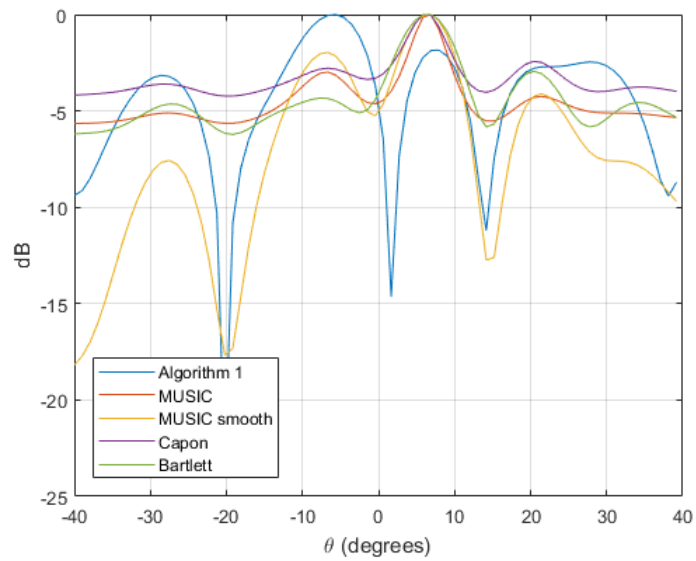


Figure 8.18: DOA spectra of the practical data collected from TART. The test has two SOIs and the array has been shifted  $\approx 8^\circ$

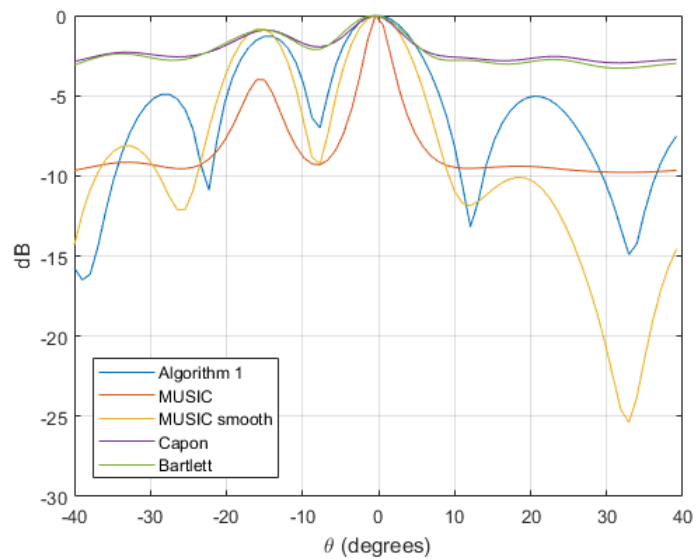


Figure 8.19: DOA spectra of the practical data collected from TART. The test has two SOIs, each with a different power level.

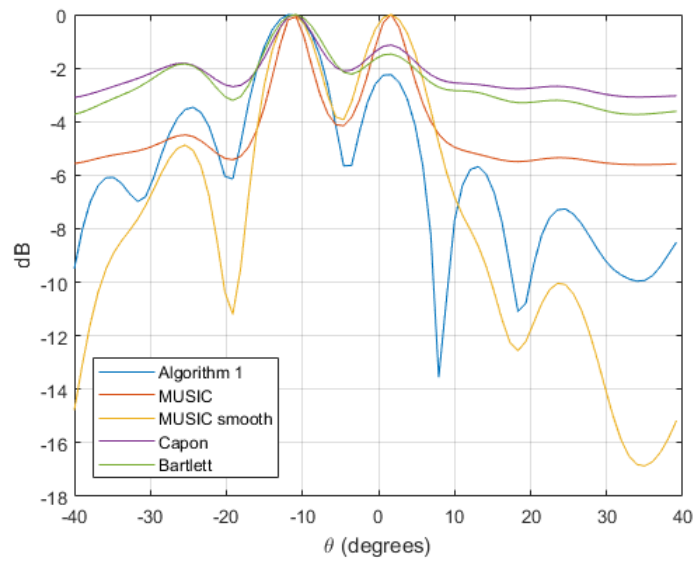


Figure 8.20: DOA spectra of the practical data collected from TART. The test has two SOIs, each with the same power level and the SOIs have been moved closer together.

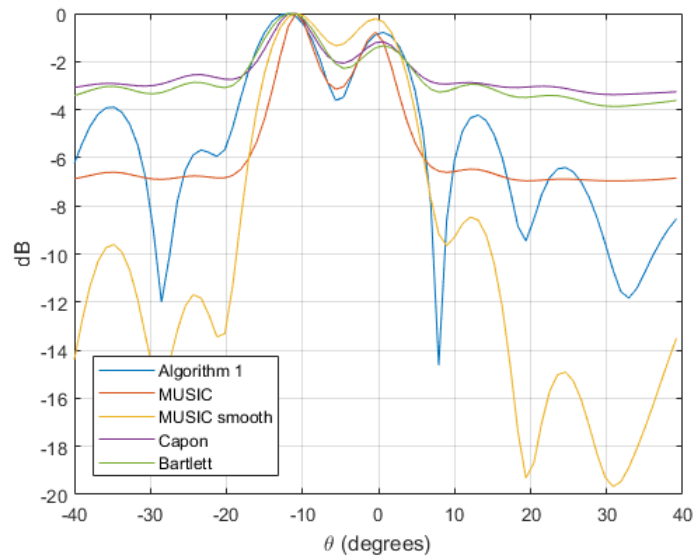


Figure 8.21: DOA spectra of the practical data collected from TART. each with the same power level and the SOIs have been moved closer together. The resolution can be seen to be an issue with signals so closely spaced, for a 12 element ULA.

The results show that algorithm 1 can function as a beamformer on the TART system, and is not limited by its hardware, however the 2 beams are not of equal amplitude in Figures 8.17, 8.18, 8.20 and 8.21. The signals in the tests, represented by these images, are of equal power therefore the beams should be of approximately equal amplitude. The sidelobes of the beams are also high when compared to the beampattern seen in Figure 8.1. The SNR of the signals was not taken into account during these preliminary tests which may account for the lack in performance in algorithm 1. The TART system became unusable before practical data could be collected to test algorithms 1 and 2 with interference.

# Chapter 9

## Conclusions and Future Work

This chapter details the conclusions drawn from the results in Chapter 8.4 and ways to expand on this thesis.

### 9.1 Conclusions

The results from Section 7.4 show that, when using a peak search, Capon and Bartlett slightly outperform MUSIC and MUSIC Smooth, possibly due to the number of signals present.

A simple computational efficiency comparison of algorithms 1 and 2 was done using a single signal for simplicity. Algorithm 1 requires 24 circular convolutions to find a single GPS satellite, in the worst case scenario, because there is no need to search through multiple beams, only the satellite numbers. Algorithm 2 however requires  $24 \times 20 = 480$  circular convolutions, in the worst case scenario, to find a single beam because each satellite must be searched for in each beam. Algorithm 2 is therefore more computationally expensive. Due to the results shown in Figure 8.16, algorithm 1 is superior to algorithm 2 at finding the minimum number of satellites needed for a position lock. However, when using algorithm 1, a priori information is required in order to know the number of GPS satellites in the array's field of view, which is not information generally available. This information is not a requirement for algorithm 2.

Both algorithms are viable blind beam forming techniques for use in GPS applications on the TART system and, in simulation, are not limited by its hardware. Algorithm 1 cannot suppress more than 28 dB of interference with 1 bit data. Algorithm 2 is a more naive beam forming technique however it will be more effective than algorithm 1 when searching for the DOA's of a large number of signals, without having to increase the number of antennae. Algorithm 1 is more sensitive, than algorithm 2, to parameter changes when using 1 bit shifted data. This means that the use of I/Q data would be more beneficial to algorithm 1, which requires an increases in channel cables and

therefore system complexity.

The performance of the dual band microstrip antenna was similar to that of the simulation. The antenna can function as a GPS antenna however due to its size, it cannot be easily used in a  $\frac{\lambda}{2}$  element spaced array without accounting for mutual coupling of the elements.

The practical measurements taken show the potential for algorithm 1 to be implemented, as a beamformer, in the TART system. The SNR of the system appears to be an issue and would have been specified if secondary measurements were taken. One or the other of the beams produced by algorithm 1 was feeble. Communication issues between the FPGA and raspberry pi meant that thorough practical testing was not possible.

## 9.2 Future Work

Implementing algorithms 1 and 2 on the TART system and comparing the results to simulation is work for the future. Compressed sensing is a way to collect the input data and reduce the number of samples required [50]. It collects samples, fewer than Nyquist [51], and uses sparse transforms to recreate the data without loss of information. The efficacy of this technique when used in a GPS application while using algorithms 1 or 2 can be explored. Both beam forming techniques can be expanded to be implemented on planar or perhaps sparse arrays.

Investigating other applications for algorithms 1 and 2 and determining how efficient the algorithms are for these applications is something to consider for further study. Practical data could be collected using a more accurate setup such as laser bore sighting and poles that can rotate more controllably. The amount of multipath present during the practical tests could be lowered by placing the antennae in a more open space. The ability of these algorithms to mitigate highly dynamic interference sources can be investigated.

# Appendices

# Appendix A

## Matlab codes

### A.1 DOA algorithms

```

clear all
%% SNR and integration time variation
counter1=1;
counter2=1;
SNR=-140:4:-100;
int_t=16000:-1200:1000;
angles=-45:10:45;
avs=1;
while (avs<=102)
    %% Describe incoming signals and initialize variables
    samples=16000;
    N=samples;
    y1=zeros(N, length(freqz), length(angles));
    M=12;
    format long

    % Incoming signal directions, frequencies, power, and phase
    th_in_deg = [60,30, -5, -40]; % In deg
    ph_in_deg = [0,0 ,0, 0]; % In deg
    f_in = [1.575e9,1.575e9, 1.575e9 ,1.575e9];% In Hz
    A_in_dBm = [-129,-129, -129, -129]; % In dBm
    P_in_deg = [0,40, 80, -90]; % In deg
    fL=1.575e9;% In deg
    fd=[3.5e3, -2.5e3, 4e3, -3.5e3];

    % Generate sampled baseband time signals
    % Create the steering vectors for the signals
    f_IF = 4e6; % IF frequency in Hz
    f_LO = fL-f_IF; % Low side inject ideal LO in Hz
    f_samp = 16e6; % Sample rate in Hz
    Nt = N; % Number of time samples
    % Sample time information
    delT = 1/f_samp;
    t0 = 0;
    t = t0:delT:(t0+delT*(Nt-1));
    Nant = M; % Number of elements
    d = 0.5*(physconst('lightspeed')/1.575e9);
    lambda=(physconst('lightspeed')/1.575e9);% Element spacing in m
    r = [(-(Nant-1)/2:(Nant-1)/2).'*d, zeros(Nant,2)];
    az = ph_in_deg(:);
    el = 90 - th_in_deg(:);

```



```

k = 2*pi.*f_in(:)./physconst('lightspeed').*[cosd(az).*cosd(el), sind(az).*cosd(el)
), sind(el)];% Uniform linear array along x-axis ([x,y,z] positions in m)
A = exp(-li*r*k. '); % Array response matrix

Nsearch = 361;
th_search_deg = linspace(-90,90,Nsearch).';
ph_search_deg = zeros(size(th_search_deg)); % Only 1D search here
f_search = ones(size(th_search_deg))*(1.575e9);
az_search = ph_search_deg;
el_search = 90 - th_search_deg;
% Corresponding points on array manifold
k_search = 2*pi.*f_search./physconst('lightspeed').*[cosd(az_search).*cosd(
el_search), sind(az_search).*cosd(el_search), sind(el_search)];
A_search = exp(-li*r*k_search. ');
%% uncorrelated noise for each channel
vc=SNR(counter1);
nsigs=(wgn(12, N, vc, 50)+li*wgn(12, N, vc, 50))';
si =(10.^(((A_in_dBm)-10)/20)).*(exp(li*(2*pi*(f_in+fd).*t(:)+deg2rad(P_in_deg)))));
;
sip=si;
sigi=A*sip. ';

% total incoming signal

si2=nsigs'+ sigi;

% Gain control

val=1/abs(max(max((si2))));
si4=val*si2;
%% Mixer
F1=bsxfun(@times, si4, (exp(-li*2*pi*(f_L0).*t)));
%% A2D
xi=real(F1);
xq = imag(F1);
Nbits =bits(counter2);
A2D_maxV = 1;
A2D_minV = -1;
A2D_del = A2D_maxV - A2D_minV;
Nthresholds = 2^Nbits-1;
delThNorm = 1/(Nthresholds+1);
A2D_thresholds = (delThNorm:delThNorm:Nthresholds*delThNorm)*A2D_del + A2D_minV;
A2D_outLevels = linspace(A2D_minV,A2D_maxV,Nthresholds+1); % A2D levels signal
(assume ideal I/Q and LPF)
xi(xi < A2D_thresholds(1)) = A2D_outLevels(1);
xq(xq < A2D_thresholds(1)) = A2D_outLevels(1);
for tt = 2:Nthresholds
xi(xi >= A2D_thresholds(tt-1) & xi <= A2D_thresholds(tt)) = A2D_outLevels(tt);
xq(xq >= A2D_thresholds(tt-1) & xq <= A2D_thresholds(tt)) = A2D_outLevels(tt);
end
xi(xi > A2D_thresholds(Nthresholds)) = A2D_outLevels(Nthresholds+1);
xq(xq > A2D_thresholds(Nthresholds)) = A2D_outLevels(Nthresholds+1);
% Rebuild the complex signal from the I-Q data
x_ReIm = xi + li.*xq;
x_Hilbert = hilbert(xi)';
x_shift = xi - li*circshift(xi,-1,2); %Bit shifted data

%% I/Q or bit shifted data and covariance matrix

%X = x_ReIm;
%X = x_Hilbert;
X = x_shift;
Rxx=X*X'./N;

```

```

theta=-90:0.5:90;
[E, D, EE]=svd(Rxx); %Rxx=E*D*EE'
L=4;
NN11=E(:, L+1:M);%interference subspace spans the eigen vectors that correspond to
the highest eigen values
%everything else should be orthogonal to these.

%% DOA algorithm spectra
for i=1:length(theta)
    SS1=zeros(1,length(M));
    SS1=A_search(:, i);
    PP1=(SS1'*(NN11*NN11')*SS1);
    PS=(SS1'/Rxx)*SS1;
    PS2=((SS1'*Rxx)*SS1);
    Pmusic1(i)=abs(1/ PP1);
    Pcapon(i)=abs(1/PS);
    PBartlett(i)=abs(PS2);
    Z_smoothMUSIC(i) = 1 - (SS1'*(NN11*NN11')*SS1)/(SS1'*SS1);
end
Pmusic1=10*log10(Pmusic1./max(Pmusic1));
Pcapon=10*log10(Pcapon./max(Pcapon));
PBartlett=10*log10(PBartlett./max(PBartlett));
Pmusicsmooth=10*log10(abs(Z_smoothMUSIC)./max(abs(Z_smoothMUSIC)));
%Find the peaks and input these to condition statements
[pk1, th1 ,wi1, p1]=findpeaks(Pmusic1); %where are the signal peaks with high
promenence.
[pk2, th2 ,wi2, p2]=findpeaks(Pcapon);
[pk3, th3 ,wi3, p3]=findpeaks(PBartlett);
[pk4, th4 ,wi4, p4]=findpeaks(Pmusicsmooth);
[~, I1]=sort(pk1, 'descend');
%% Conditions
if length(I1)>=4
    angles1=theta(th1(I1(1:4)));
else
    angles1=theta(th1(I1(1:length(I1))));
    angles1(length(I1)+1:4)=0;
end
angles_music=sort(angles1, 'descend');
[~, I2]=sort(pk2, 'descend');
if length(I2)>=4
    angles2=theta(th2(I2(1:4)));
else
    angles2=theta(th2(I2(1:length(I2))));
    angles2(length(I1)+1:4)=0;
end
angles_capon=sort(angles2, 'descend');
[~, I3]=sort(pk3, 'descend');
if length(I3)>=4
    angles3=theta(th3(I3(1:4)));
else
    angles3=theta(th3(I3(1:length(I3))));
    angles3(length(I3)+1:4)=0;
end
angles_bart=sort(angles3, 'descend');
[~, I4]=sort(pk4, 'descend');
if length(I4)>=4
    angles4=theta(th4(I4(1:4)));
else
    angles4=theta(th4(I4(1:length(I4))));
    angles4(length(I4)+1:4)=0;
end
angles_smooth=sort(angles4, 'descend');

```

```

phi=linspace(-90, 90, 361*50)*pi/180;% angular domain -pi/2:pi/2
if (th_in_deg(1)-5<=angles_music(1)&&angles_music(1)<=th_in_deg(1)+5) && (
    th_in_deg(2)-5<=angles_music(2)&&angles_music(2)<=th_in_deg(2)+5) && (
    th_in_deg(3)-5<=angles_music(3)&&angles_music(3)<=th_in_deg(3)+5)
    error_music(:,counter1, counter2, avs)=abs(angles_music-(th_in_deg));
    msuccess(counter1, counter2, avs)=1;
else
    error_music(:,counter1, counter2, avs)=NaN;
    msuccess(counter1, counter2, avs)=0;

end
if (th_in_deg(1)-5<=angles_capon(1)&&angles_capon(1)<=th_in_deg(1)+5) && (
    th_in_deg(2)-5<=angles_capon(2)&&angles_capon(2)<=th_in_deg(2)+5) && (
    th_in_deg(3)-5<=angles_capon(3)&&angles_capon(3)<=th_in_deg(3)+5)
    error_capon(:,counter1, counter2, avs)=abs(angles_capon-(th_in_deg));
    csuccess(counter1, counter2, avs)=1;
else
    error_capon(:,counter1, counter2, avs)=NaN;
    csuccess(counter1, counter2, avs)=0;

end
if (th_in_deg(1)-5<=angles_bart(1)&&angles_bart(1)<=th_in_deg(1)+5) && (th_in_deg
    (2)-5<=angles_bart(2)&&angles_bart(2)<=th_in_deg(2)+5) && (th_in_deg(3)-5<=
    angles_bart(3)&&angles_bart(3)<=th_in_deg(3)+5)
    error_bart(:,counter1, counter2, avs)=abs(angles_bart-(th_in_deg));
    bsuccess(counter1, counter2, avs)=1;
else
    error_bart(:,counter1, counter2, avs)=NaN;
    bsuccess(counter1, counter2, avs)=0;

end
if (th_in_deg(1)-5<=angles_smooth(1)&&angles_smooth(1)<=th_in_deg(1)+5) && (
    th_in_deg(2)-5<=angles_smooth(2)&&angles_smooth(2)<=th_in_deg(2)+5) && (
    th_in_deg(3)-5<=angles_smooth(3)&&angles_smooth(3)<=th_in_deg(3)+5)
    error_smooth(:,counter1, counter2,avs)=abs(angles_smooth-(th_in_deg));
    ssuccess(counter1, counter2, avs)=1;
else
    error_smooth(:,counter1, counter2,avs)=NaN;
    ssuccess(counter1, counter2, avs)=0;

end
%% Check the counters
avs=avs+1;
if avs==101
    counter2=counter2+1;
    if counter2>length(SNR)
        counter1=counter1+1;
        counter2=1;
        if counter1>length(int_t)
            break;
        end
    end
    avs=1;
end
end
%% Find an average error
for i=1:4
    for k=1:length(SNR)
        for j=1:length(int_t)
            errm(i,j, k)=sum(error_music(i, j, k, :))/avs';
            errc(i,j, k)=sum(error_capon(i, j, k, :))/avs';
            errb(i,j, k)=sum(error_bart(i, j, k, :))/avs';
            errs(i,j, k)=sum(error_smooth(i, j, k, :))/avs';
        end
    end
end
end

```

```
%% Success percentages of each algorithm
```

```
msuccess=sum(msuccess, 3);
csuccess=sum(csuccess, 3);
bsuccess=sum(bsuccess, 3);
ssuccess=sum(ssuccess, 3);
```

## A.2 Beamforming algorithm 1

```
clear all
```

```
counter1=1;
counter2=1;
SNR=-140:4:-100;
int_t=16000:-1200:1000;
phi=linspace(-90, 90, 361)*pi/180;% angular domain -pi/2:pi/2
Ylim=[-80, 10];
freqz=-3e3:500:3e3;
avs=1;
while (avs<=102)
    samples=int_t(counter1);
    N=samples;
    M=12;
    format long

    %% Describe incoming signals
    % Incoming signal directions, frequencies, power, and phase
    th_in_deg = [-60,-30, -5, 30, 60]; % In deg
    ph_in_deg = [0,0, 0 ,0 ,0]; % In deg
    f_in = [1.575e9,1.575e9, 1.575e9, 1.575e9 ,1.575e9 ];% In Hz
    A_in_dBm = [-122, -123, -123, -121, -90]; % In dBm
    P_in_deg = [0,10,25 60, 90]; % In deg
    fL=1.57542e9;% In deg
    fd=[3.5e3, -2.5e3, -4e3, 4e3, 0];
    % Generate sampled baseband time signals
    f_IF = 4e6; % IF frequency in Hz
    f_L0 = fL-f_IF; % Low side inject ideal L0 in Hz
    f_samp = 16e6; % Sample rate in Hz
    Nt = N; % Number of time samples
    PGC_dB = 30-max([A_in_dBm]); % System gain in dB
    Ns = length(A_in_dBm); % Number of signals
    % Sample time information
    delT = 1/f_samp;
    t0 = 0;
    t = t0:delT:(t0+delT*(Nt-1));
    Nant = M; % Number of elements
    d = 0.5*(physconst('lightspeed')/1.575e9);
    lambda=(physconst('lightspeed')/1.575e9);% Element spacing in m
    r = [(-(Nant-1)/2:(Nant-1)/2).'*d,zeros(Nant,2)];
    az = ph_in_deg(:);
    el = 90 - th_in_deg(:);
    k = 2*pi.*f_in(:)./physconst('lightspeed').*[cosd(az).*cosd(el), sind(az).*cosd(el)
    ], sind(el)];% Uniform linear array along x-axis ([x,y,z] positions in m)
    A = exp(-1i*r*k. '); % Array response matrix

    Nsearch = 361;
    th_search_deg = linspace(-90,90,Nsearch).';
    ph_search_deg = zeros(size(th_search_deg)); % Only 1D search here
    f_search = ones(size(th_search_deg))*(1.575e9);
    az_search = ph_search_deg;
    el_search = 90 - th_search_deg;
    % Corresponding points on array manifold
```

```

% k_search = 2*pi.*f_search./physconst('lightspeed').*[cosd(ph_search_deg(:)).*
    sind(th_search_deg(:)),sind(ph_search_deg(:)).*sind(th_search_deg(:)),cosd(
    th_search_deg(:))];
k_search = 2*pi.*f_search./physconst('lightspeed').*[cosd(az_search).*cosd(
    el_search), sind(az_search).*cosd(el_search), sind(el_search)];
A_search = exp(-1i*r*k_search. ');

vc=SNR(counter2);
nsigs=(wgn(12, N, vc, 50)+1i*wgn(12, N, vc, 50))';
int=((10.^((A_in_dBm(2))-10)/20)).*randn(1, N)+(1i*(10.^((A_in_dBm(2))-10)/20))
    .*randn(1, N));
Fs = f_samp;

% create the band-limited Gaussian noise

si =(10.^(((A_in_dBm)-10)/20)).*(exp(1i*(2*pi*(f_in+fd).*t(:)+deg2rad(P_in_deg)))
    );

sip=si;%[f1 ,pkl];
%si=horzcat( si3.', int. ');
sigi=A*sip. ';

si2=nsigs'+ sigi;

val=1/abs(max(max((si2))));
si4=val*si2;

F1=bsxfun(@times, si4, (exp(-1i*2*pi*(f_L0).*t)));
xi=real(F1);
xq = imag(F1);
% A2D
Nbits =12;
A2D_maxV = 1;
A2D_minV = -1;
A2D_del = A2D_maxV - A2D_minV;
Nthresholds = 2^Nbits-1;
delThNorm = 1/(Nthresholds+1);
A2D_thresholds = (delThNorm:delThNorm:Nthresholds*delThNorm)*A2D_del + A2D_minV;
A2D_outLevels = linspace(A2D_minV,A2D_maxV,Nthresholds+1); % A2D levels signal
    (assume ideal I/Q and LPF)
xi(xi < A2D_thresholds(1)) = A2D_outLevels(1);
xq(xq < A2D_thresholds(1)) = A2D_outLevels(1);
for tt = 2:Nthresholds
    xi(xi >= A2D_thresholds(tt-1) & xi <= A2D_thresholds(tt)) = A2D_outLevels(tt);
    xq(xq >= A2D_thresholds(tt-1) & xq <= A2D_thresholds(tt)) = A2D_outLevels(tt);
end
xi(xi > A2D_thresholds(Nthresholds)) = A2D_outLevels(Nthresholds+1);
xq(xq > A2D_thresholds(Nthresholds)) = A2D_outLevels(Nthresholds+1);
% Rebuild the complex signal from the I-Q data
x_ReIm = xi + 1i.*xq;
x_Hilbert = hilbert(xi. ');
x_shift = xi - 1i*circshift(xi,-1,2);

%X = x_ReIm;
%X = x_Hilbert;
X = x_shift;
Rxx=X*X'./N;

theta=-90:0.5:90;
[E, D, EE]=svd(Rxx); %Rxx=E*D*EE'
L=1;
Noise_sig_sub=E(:, L+1:M);%take the largest eigen vectors and stack into the

```

```

    int_subspace nad others into Noise_sig_sub
Int_sub=E(:, 1:L);

F=((Int_sub)'*Int_sub)\(Int_sub)';
V=eye(M, M)-Int_sub*F; %subspace orthogonal to interference
%V=eye(M,M)-Int_sub*Int_sub';
Y=V*X;%signal projection (signal X should now be interference free)
Y1=Y*Y'./N; %new covariance matrix

[E, D, EE]=svd(Y1); %Find the remaining signals ( The GPS signals should be
    orthogonal to the noise subspace
%of the remaining matrix

NN11=E(:, 5:M);

F=(NN11'*NN11)\NN11';
V=eye(M, M)-NN11*F;
for i=1:12
    w1=V(:, i);
    [pks,locs] =findpeaks(20*log10(abs(w1'*A_search)));
    B=[pks ;locs];
    [ii,ii] = sort(-pks);
    minval=20*log10(abs(w1'*A_search));
    minval2=minval(301);
    y=pks(ii(1:2));
    K(i)=sum(y)+abs(minval2);
end
[g, h]=max(K);

w=V(:,h);

%{
figure(1)
hold on
yend = [10 10 10];
yst = [-50 -50 -50];
plot(theta,20*log10(abs(w1'*A_search)),'-','LineWidth',1.0, 'color', 'b')
%}

a1=55;
a2=65;
cv=-20*log10(abs(w1'*A_search));
cv=cv(((a1+90)/0.5)+1:((a2+90)/0.5)+1);
[pk1, th1 ,wi1, p1]=findpeaks(20*log10(abs(w1'*A_search)));
[pk2, th2 ,wi2, p2]=findpeaks(cv);%where are the signal peaks with high prominence

[~, I1]=sort(pk1, 'descend');
[~, I2]=sort(pk2, 'descend');
if length(I1)>=4
    angles1=theta(th1(I1(1:4)));
else
    angles1=theta(th1(I1(1:length(I1))));
    angles1(length(I1)+1:4)=0;
end
if length(th2)==1
    angles2=55+0.5*(th2-1);
    if cv(th2(1))>=30
        chr=1;
    else
        chr=0;
    end
elseif isempty(th2)
    angles2=0;

```

```

    chr=0;
else
    angles2=55+0.5*(th2(1)-1);
    if cv(th2(1))>=30
        chr=1;
    else
        chr=0;
    end
end
angles=sort(angles1, 'ascend');
angles=[angles, angles2];
phi=linspace(-90, 90, 361*50)*pi/180;% angular domain -pi/2:pi/2
if (th_in_deg(1)-5<=angles(1)&&angles(1)<=th_in_deg(1)+5) && (th_in_deg(2)-5<=
    angles(2)&&angles(2)<=th_in_deg(2)+5) && (th_in_deg(3)-5<=angles(3)&&angles(3)
    <=th_in_deg(3)+5) && (th_in_deg(4)-5<=angles(4)&&angles(4)<=th_in_deg(4)+5)&&
    (th_in_deg(5)-5<=angles(5)&&angles(5)<=th_in_deg(5)+5)&&chr==1
    error(:,counter1, counter2, avs)=abs(angles-(th_in_deg));
    success(counter1, counter2, avs)=1;
else
    error(:,counter1, counter2, avs)=NaN;
    success(counter1, counter2, avs)=0;

end
avs=avs+1;
if avs==101
    counter2=counter2+1;
    if counter2>length(SNR)
        counter1=counter1+1;
        counter2=1;
        if counter1>length(int_t)
            break;
        end
    end
    avs=1;
end
end
%{
figure(1)
xlabel('SNR');
Markers = {'+', 'o', '*', 'x', 'v', 'd', '^', 's', '>', '<'};
ylabel(['ERROR in deg at angle ' num2str(th_in_deg(1))]);
hold on
for i=1:length(xp)
    plot(-(snr_vals)+160, squeeze(errm(1, i,:)), strcat('-',Markers{i}));
    legendInfo{i} = ['Music with No. samples= ' num2str(xp(i))];
end
legend(legendInfo);

figure(2)
xlabel('SNR');
ylabel(['ERROR in deg at angle ' num2str(th_in_deg(2))]);
hold on
for i=1:length(xp)
    plot(-(snr_vals)+160, squeeze(errm(2, i,:)), strcat('-',Markers{i}));
    legendInfo{i} = ['Music with No. samples= ' num2str(xp(i))];
end
legend(legendInfo);

figure(3)
xlabel('SNR');
ylabel(['ERROR in deg at angle ' num2str(th_in_deg(3))]);
hold on
for i=1:length(xp)

```

```

    plot(-(snr_vals)+160, squeeze(erm(3, i,:)), strcat('-',Markers{i}));
    legendInfo{i} = ['Music with No. samples= ' num2str(xp(i))];
end
legend(legendInfo);
%}
success=sum(success, 3);
save('alg1_test1.mat', 'success', 'error')

```

## A.3 Beamforming algorithm 2

```

clear all

counter1=1;
counter2=1;
SNR=-140:4:-100;
int_t=16000:-1200:1000;
angles=-45:10:45;
phi=linspace(-90, 90, 361)*pi/180;% angular domain -pi/2:pi/2
Ylim=[-80, 10];
freqz=-3e3:500:3e3;
avs=1;
while (avs<=102)
samples=int_t(counter1);
N=samples;
y1=zeros(N, length(freqz), length(angles));
SB=generateCA2(1, 16/1.023);
SB=SB(1:N);
SB(SB==0)=-1;
M=12;
format long

%% Describe incoming signals

% Incoming signal directions, frequencies, power, and phase
th_in_deg = [40, -20]; % In deg
ph_in_deg = 0;
fL=1.575e9;% In deg
fD=[2.5e3, -2.5e3];
f_in = [fL, fD];% In Hz
A_in_dBm = [-129, -60]; % In dBm
P_in_deg = 0; % In deg

% Generate sampled baseband time signals
f_IF = 4e6; % IF frequency in Hz
f_LO = fL-f_IF; % Low side inject ideal LO in Hz
f_samp = 16e6; % Sample rate in Hz
Nt = N; % Number of time samples
PGC_dB = 30-max([A_in_dBm]); % System gain in dB
Ns = length(A_in_dBm); % Number of signals
% Sample time information
delT = 1/f_samp;
t0 = 0;
t = t0:delT:(t0+delT*(Nt-1));
Nant = M; % Number of elements
d = 0.5*(physconst('lightspeed')/1.575e9);
lambda=(physconst('lightspeed')/1.575e9);% Element spacing in m
r = [(-(Nant-1)/2:(Nant-1)/2) .* d, zeros(Nant,2)];
az = ph_in_deg(:);
el = 90 - th_in_deg(:);
k = 2*pi.*f_in(:)./physconst('lightspeed').*[cosd(az).*cosd(el), sind(az).*cosd(el),
sind(el)];% Uniform linear array along x-axis ([x,y,z] positions in m)
A = exp(-1i*r*k.); % Array response matrix

```



```

Nsearch = 361;
th_search_deg = linspace(-90,90,Nsearch).';
ph_search_deg = zeros(size(th_search_deg)); % Only 1D search here
f_search = ones(size(th_search_deg))*(1.575e9);
az_search = ph_search_deg;
el_search = 90 - th_search_deg;
% Corresponding points on array manifold
% k_search = 2*pi.*f_search./physconst('lightspeed').*[cosd(ph_search_deg(:)).*sind(
    th_search_deg(:)),sind(ph_search_deg(:)).*sind(th_search_deg(:)),cosd(
    th_search_deg(:))];
k_search = 2*pi.*f_search./physconst('lightspeed').*[cosd(az_search).*cosd(el_search),
    sind(az_search).*cosd(el_search), sind(el_search)];
A_search = exp(-1i*r*k_search. ');

vc=SNR(counter2);
nsigs=(wgn(12, N, vc, 50)+1i*wgn(12, N, vc, 50))';
int=((10.^((A_in_dBm(2))-10)/20)).*randn(1, N)+(1i*(10.^((A_in_dBm(2))-10)/20)).*
    randn(1, N));
Fs = f_samp;
de = fdesign.lowpass('Fp,Fst,Ap,Ast',2/16,3/16,0.5,40);
B = design(de,'equiripple','StopbandShape','linear','StopbandDecay',20);

B1=designfilt('lowpassiir','FilterOrder',8, ...
    'PassbandFrequency',2e6,'PassbandRipple',0.2, ...
    'SampleRate',Fs);

% create the band-limited Gaussian noise
y = filter(B1,int).';

si =(10.^((A_in_dBm)-10)/20)).*(exp(1i*(2*pi*(f_in+fd).*t(:)+deg2rad(P_in_deg))));
carr=(exp(1i*(2*pi*(fL+fd(2)).*t(:)+deg2rad(90))));
k1=randi(N);
SD1=circshift(SB, k1);

f1=(si(: ,1)).*SD1';
pkl=y.*carr;
sip=[f1 ,pkl];
%si=horzcat( si3.', int. ');
sigi=A*sip. ';

si2=nsigs'+ sigi;

val=1/abs(max(max((si2))));
si4=val*si2;

F1=bsxfun(@times, si4, (exp(-1i*2*pi*(f_L0).*t)));
xi=real(F1);
xq = imag(F1);
% A2D
Nbits =12;
A2D_maxV = 1;
A2D_minV = -1;
A2D_del = A2D_maxV - A2D_minV;
Nthresholds = 2^Nbits-1;
delThNorm = 1/(Nthresholds+1);
A2D_thresholds = (delThNorm:delThNorm:Nthresholds*delThNorm)*A2D_del + A2D_minV;
A2D_outLevels = linspace(A2D_minV,A2D_maxV,Nthresholds+1); % A2D levels signal (
    assume ideal I/Q and LPF)
xi(xi < A2D_thresholds(1)) = A2D_outLevels(1);
xq(xq < A2D_thresholds(1)) = A2D_outLevels(1);
for tt = 2:Nthresholds
    xi(xi >= A2D_thresholds(tt-1) & xi <= A2D_thresholds(tt)) = A2D_outLevels(tt);

```

```

    xq(xq >= A2D_thresholds(tt-1) & xq <= A2D_thresholds(tt)) = A2D_outLevels(tt);
end
xi(xi > A2D_thresholds(Nthresholds)) = A2D_outLevels(Nthresholds+1);
xq(xq > A2D_thresholds(Nthresholds)) = A2D_outLevels(Nthresholds+1);
% Rebuild the complex signal from the I-Q data
x_ReIm = xi + 1i.*xq;
x_Hilbert = hilbert(xi)';
x_shift = xi - 1i*circshift(xi,-1,2);

%X = x_ReIm;
%X = x_Hilbert;
X = x_shift;
Rxx=X*X'./N;

theta=-90:0.5:90;
[E, D, EE]=svd(Rxx); %Rxx=E*D*EE'
L=1;
NN11=E(:, L+1:M);%interference subspace spans the eigen vectors that correspond to the
highest eigen values
%everything else should be orthogonal to these.

for i=1:length(theta)
    SS1=zeros(1,length(M));
    SS1=A_search(:, i);
    PP1=(SS1'*(NN11*NN11')*SS1);
    PS=(SS1'/Rxx)*SS1;
    PS2=((SS1'*Rxx)*SS1);
    Pmusic1(i)=abs(1/ PP1);
    Pcapon(i)=abs(1/PS);
    PBartlett(i)=abs(PS2);
    Z_smoothMUSIC(i) = 1 - (SS1'*(NN11*NN11')*SS1)/(SS1'*SS1);
end
Pmusic1=10*log10(Pmusic1./max(Pmusic1));
Pcapon=10*log10(Pcapon./max(Pcapon));
Pmusicsmooth=10*log10(abs(Z_smoothMUSIC)./max(abs(Z_smoothMUSIC)));
%{
figure(1)
plot(theta,Pmusic1,'-','LineWidth',1.0, 'color', 'b')
title('MUSIC spectrum of interference');
xlabel('Angle in degrees');
ylabel('Normalised amplitude (dB)');
grid on
%}
[pk, th ,wi, p]=findpeaks(Pmusic1); %where are the signal peaks with high prominence.
[v, in]=max(p);
Int_peak=theta(th(in));

G=[1, 0]';
%Create the steering vectors for the signals

for ang=angles
    eL_srch = 90 - [ang, Int_peak]';
    ph_srch_deg = zeros(1,length(th_in_deg));
    f_srch = ones(1,length(th_in_deg))*(1.575e9);
    az_srch = ph_srch_deg;
    k_srch = 2*pi.*f_srch./physconst('lightspeed').*[cosd(az_srch).*cosd(eL_srch),
        sind(az_srch).*cosd(eL_srch), sind(eL_srch)];
    A_srch = exp(-1i*r*k_srch.);
    S=A_srch;
    A1=(Rxx\S);
    A2=(S'*(Rxx\S));
    W1=A1*(A2\G);

```

```

Wn=(A_srch')\G;
Wx=exp([0:M-1]*1i*pi*sind(angs)');

check=W1'*X;
G1=check;
%{
figure(cnt)
hold on
plot(phi*180/pi,20*log10(abs((W1'*A_search)./max(W1'*A_search))));
grid on
xlabel('Theta (deg)');
ylim(Ylim)
hold on
title('Independent beam patterns');
%}
%remove carrier
ind=1;
for fr=freqz
    F11i=imag(bsxfun(@times, G1, exp(-1i*2*pi*(4e6+fr).*t)));
    F11q=real(bsxfun(@times, G1, exp(-1i*2*pi*(4e6+fr).*t)));

    y1(:,ind,count)=abs(fft(conj(fft(SB)).*(fft(bsxfun(@plus,F11i,1i*F11q))))).^2;

    ind=ind+1;
end
count=count+1;
end
[q1, q2]=max(max(y1, [], 2));%max phase peak for each beam with signal 1
[q3, q4]=max(max(y1));
[a1,d1]=max(q1);%beam with the max peak for signal 1-phase
[a3,d3]=max(q3);%beam with the max peak for signal 1-freq
direc=angles(d3);
frequency=freqz(q4(d3));
phase=q2(d1)-1;
%{
figure(2)
hold on
for i=1:length(angles)
    plot(linspace(0, N-1, N), y1(:, q4(d3) ,i));
end
title(['Doppler shift ' num2str(frequency) ' Hz' '—Phase shift ' num2str(phase)]);
legend(['S0I angle at ' num2str(direc)]);
xlabel('Phase shift (in bits) ');
ylabel('Correlation peak level');
grid on
%}
if (frequency==fd(1)) && (phase==k1 || phase==k1+1 || phase==k1-1) && (direc<=
    th_in_deg(1)+5 && direc>=th_in_deg(1)-5)
    temp(counter1, counter2, avs)=1;
else
    temp(counter1, counter2, avs)=0;
end
avs=avs+1;
if avs==101
    counter2=counter2+1;
    if counter2>length(SNR)
        counter1=counter1+1;
        counter2=1;
        if counter1>length(int_t)
            break;
        end
    end
end
end

```

```

    avs=1;
end
end
success_percent=sum(temp, 3);

```

## A.4 Phase calibration

```

function final_phase=phase_calib_D0A()

clear all
N=16000;
M=12;
format long


```

```

t0 = 0;
t = t0:delT:(t0+delT*(Nt-1));
Nant = M; % Number of elements
d = 0.5*(physconst('lightspeed')/1.575e9);
lambda=(physconst('lightspeed')/1.575e9); % Element spacing in m
r = [(-(Nant-1)/2:(Nant-1)/2).*d,zeros(Nant,2)];
az = ph_in_deg(:);
el = 90 - th_in_deg(:);
k = 2*pi.*f_in(:)/physconst('lightspeed').*[cosd(az).*cosd(el), sind(az).*cosd(el),
sind(el)]; % Uniform linear array along x-axis ([x,y,z] positions in m)
A = exp(-1i*r*k.); % Array response matrix

vc=-140;
nsigs=(wgn(Nant, N, vc, 50)+1i*wgn(Nant, N, vc, 50))';

% Sample time information
si = (10.^((A_in_dBm)-10)/20)).*(exp(1i*(2*pi*(f_in+fd).*t(:) + deg2rad(P_in_deg))));
% -30 dB for dBW input

% Received signals
xi_s = A*si.';

% Add the noise
xi_sn=bsxfun(@plus, nsigs', xi_s);
snr=20*log10(norm(si(:,1))/norm(nsigs));

val=1/(max(max(abs(xi_sn))));
si4=val*xi_sn;

F1 = bsxfun(@times,si4,exp(-1i*2*pi*f_L0.*t));

xi=real(F1);
xq = imag(F1);

% A2D
Nbits = 1;
A2D_maxV = 1;
A2D_minV = -1;
A2D_del = A2D_maxV - A2D_minV;
Nthresholds = 2^Nbits-1;
delThNorm = 1/(Nthresholds+1);
A2D_thresholds = (delThNorm:delThNorm:Nthresholds*delThNorm)*A2D_del + A2D_minV;
A2D_outLevels = linspace(A2D_minV,A2D_maxV,Nthresholds+1);
xi(xi < A2D_thresholds(1)) = A2D_outLevels(1);
xq(xq < A2D_thresholds(1)) = A2D_outLevels(1);
for tt = 2:Nthresholds
    xi(xi >= A2D_thresholds(tt-1) & xi <= A2D_thresholds(tt)) = A2D_outLevels(tt);
    xq(xq >= A2D_thresholds(tt-1) & xq <= A2D_thresholds(tt)) = A2D_outLevels(tt);
end
xi(xi > A2D_thresholds(Nthresholds)) = A2D_outLevels(Nthresholds+1);
xq(xq > A2D_thresholds(Nthresholds)) = A2D_outLevels(Nthresholds+1);
% Rebuild the complex signal from the I-Q data
x_ReIm = xi + 1i.*xq;
x_Hilbert = hilbert(xi).';
%input from TART
for i=1:Nant
    start = 'results channel';
    if i-1<10
%         files1(i,:) = [start ' ' num2str(i-1) '.txt']; % Character Array
        files1 = [start ' ' num2str(i+11-Nant) '.txt']; % Character Array
        fileID = fopen(files1,'r');
        results(i, :) = fscanf(fileID, '%f');
    else

```

```

%         files2(i,:) = [start ' ' num2str(i-1) '.txt'];           % Character Array
files2 = [start ' ' num2str(i+11-Nant) '.txt'];           % Character Array
fileID = fopen(files2,'r');
results(i, :) = fscanf(fileID, '%f');
end

end

x_shift1 = results - 1i*circshift(results,-1,2);
%x_shift1=vtemp(N:(2*N)-1, :).';
save('input_data.mat', 'x_shift1')
x_shift = xi - 1i*circshift(xi,-1,2);
X = x_shift;
theta=linspace(-90, 90, N);
Rxx=X*X'./N;
L=0;
% Search Vectors (full search - very inefficient)
Nsearch = N;
th_search_deg = linspace(-90,90,Nsearch).';
ph_search_deg = zeros(size(th_search_deg)); % Only 1D search here
f_search = ones(size(th_search_deg))*(1.575e9);
az_search = ph_search_deg;
el_search = 90 - th_search_deg;

Nant = M; % Number of elements
d = 0.5*(physconst('lightspeed')/1.575e9); % Element spacing in m
r = [(-(Nant-1)/2:(Nant-1)/2).*d,zeros(Nant,2)];
% Corresponding points on array manifold
% k_search = 2*pi.*f_search./physconst('lightspeed').*[cosd(ph_search_deg(:)).*sind(
    th_search_deg(:)),sind(ph_search_deg(:)).*sind(th_search_deg(:)),cosd(
    th_search_deg(:))];
k_search = 2*pi.*f_search./physconst('lightspeed').*[cosd(az_search).*cosd(el_search),
    sind(az_search).*cosd(el_search), sind(el_search)];
A_search = exp(-1i*r*k_search. '); % Array response matrix

[E, D, EE]=svd(Rxx); %Rxx=E*D*EE'

Noise_sig_sub=E(:, L+1:M);%take the largest eigen vectors and stack into the
    int_subspace nad others into Noise_sig_sub
Int_sub=E(:, 1:L);

F=(Int_sub'*Int_sub)\Int_sub';
V=eye(M, M)-Int_sub*F; %subspace orthogonal to interference
Y=V*X;%signal projection (signal X should now be interference free)

Y1=Y*Y'./N;
%Y1=cov(Y'); %new covariance matrix

[E, D, EE]=svd(Y1); %Find the remaining signals ( The GPS signals should be orthogonal
    to the noise subspace
NN11=E(:, 2:M);%noise subspace is the smallest eigen values of the new matrix
    ??????????????

F=(NN11'*NN11)\NN11';
V=eye(M, M)-NN11*F;
%{
Xr=X(:, 1:M-1)';%two subarrays
XX1=Xr*Xr'./N;
X2=X(:, 2:M)';
XX2=X2*X2'./N;
[E1, D1, EE1]=svd(XX1);
[E2, D2, EE2]=svd(XX2);

E1=E1(:, 1:3)';

```

```

E2=E2(:, 1:3)';

mat=E1/E2;
p1=eig(mat);
angles=1/sin(angle(p1)/d*(2*pi/(physconst('lightspeed')/1.575e9)));
angles2=asin(angle(p1)/d*(2*pi/(physconst('lightspeed')/1.575e9)));
angles=angles*180/pi;

EF=[E1'; E2']*[E1, E2];
DRE1=EF(1:3, 4:6);
DRE2=EF(4:6, 4:6);

matrix=(-DRE1)/DRE2;
P =eig(matrix);
angles=1/sin(angle(P)/d*(2*pi/(physconst('lightspeed')/1.575e9)));
angles2=asin(angle(P)/d*(2*pi/(physconst('lightspeed')/1.575e9)));
angles=angles*180/pi;
%}
for i=1:length(theta)
    SS1=zeros(1,length(M));
    SS1=A_search(:, i);
    PP1=(SS1'*(NN11*NN11')*SS1);
    PS=(SS1'/Rxx)*SS1;
    PS2=((SS1'*Rxx)*SS1);
    Pmusic1(i)=abs(1/ PP1);
    Pcapon(i)=abs(1/PS);
    PBartlett(i)=abs(PS2);
    Z_smoothMUSIC(i) = 1 - (SS1'*(NN11*NN11')*SS1)/(SS1'*SS1);
end
Pmusic1=10*log10(Pmusic1./max(Pmusic1));
Pcapon=10*log10(Pcapon./max(Pcapon));
PBartlett=10*log10(PBartlett./max(PBartlett));
Pmusicsmooth=10*log10(abs(Z_smoothMUSIC)./max(abs(Z_smoothMUSIC)));
%{
figure(6)
hold on
yend = 10;
yst = -50;
plot(theta,Pmusic1,'-','LineWidth',1.0, 'color', 'b')
plot(theta,Pcapon,'--','LineWidth',1.0, 'color', 'r')
plot(theta,PBartlett,'--','LineWidth',1.0, 'color', 'c')
plot(theta,Pmusicsmooth,'--','LineWidth',1.0, 'color', 'k'), grid on
for i = 1 : numel(yst)
    plot([th_in_deg(i) th_in_deg(i)], [yst(i) yend(i)], 'color', 'k');
end
legend('MUSIC', 'Capon/MVDR', 'Bartlett', 'MUSICSmooth', 'Location', 'northeast');
xlabel('Theta (deg)');
ylabel('Normalized power');
title('DOA power spectra');
hold off
%}
thRange = deg2rad([-40,40]);
%[angles, Z] = MUSICth(x,r,thRange,ph_in_deg,f_in,1,true);

[angles, Zref] = MUSICth(x_shift,r,thRange,ph_in_deg,f_in,1,true);
[anglesf, Zreff] = MUSICth(x_shift1,r,thRange,ph_in_deg,f_in,1,true);
%Zref.Zn=Pmusic1;
%Zref.ZnSmooth=Pmusicsmooth;

% Calibration
options = optimset('Display','iter');
problem=createOptimProblem('fmincon', 'objective', @(calPhase) calError(calPhase,

```

```

    x_shift1,r,thRange,ph_in_deg,f_in,1,Zref),'nonlcon', [], 'x0', ones(Nant-1,1)
    .*0.2, 'lb', ones(Nant-1,1)*(-pi), 'ub', ones(Nant-1,1)*(pi), 'options', options);
%gs=GlobalSearch;
calPhase1 = fmincon(problem);

calFact1(1) = 1;
calFact1(2:Nant) = exp(1i.*calPhase1);
xCal1 = bsxfun(@times,x_shift1,calFact1(:));
[angles, Z] = MUSICth(xCal1,r,thRange,ph_in_deg,f_in,1,true);

DF=-abs(dB10(Z.Zn));
[pk1, th1 ,w1, p1]=findpeaks(DF);
[~, Q]=max(p1);
rd=linspace(rad2deg(thRange(1)), rad2deg(thRange(2)),length(Z.Zn));
dif=rd(th1(Q))
Wx=exp([0:M-1]*1i*pi*sind(-dif));
Xs=bsxfun(@times,Wx',xCal1.').';
[angles, Z] = MUSICth(Xs,r,thRange,ph_in_deg,f_in,1,true);

final_phase=(Wx'.*calFact1).';
save('calibration_vec.mat', 'final_phase')
[angles, Z] = MUSICth(bsxfun(@times,x_shift1, final_phase),r,thRange,ph_in_deg,f_in,1,
    true);

end

```



## List of References

- [1] R.C. Daniels and R.H. Huxford: Using Global Positioning Systems (GPS): How it Works, Limitations, and Some Guidelines for Operation. 2001. [Online]. Available at: <https://fortress.wa.gov/ecy/publications/documents/0006015.pdf>
- [2] A. Camps, J. Marchan-Hernandez, I.R.-P.X.B.-L. and Prehn, R.: New Radiometer Concepts for Ocean Remote Sensing: Description of the Passive Advanced Unit (PAU) for Ocean Monitoring. In: *2006 IEEE International Symposium on Geoscience and Remote Sensing*, pp. 3988–3991. Denver, Colorado, USA, July 2006.
- [3] National Instruments: GPS Receiver Testing. 2000. [Online]. Available at: [ftp://ftp.ni.com/pub/branches/india/gps\\_receiver\\_simulation\\_testing.pdf](ftp://ftp.ni.com/pub/branches/india/gps_receiver_simulation_testing.pdf).
- [4] G. V. Rosenbaum: Determination of GPS RF signal strengths. In: *2008 IEEE/ION Position, Location and Navigation Symposium*, pp. 449–458. 2008 Hyatt Regency Hotel Monterey, CA, May 2008.
- [5] Maxim integrated: Universal GPS Receiver. 2000. [Online]. Available at: <https://datasheets.maximintegrated.com/en/ds/MAX2769B.pdf>
- [6] Y. Zheng: Adaptive Antenna Array Processing for GPS Receivers. Master's thesis, The University of Adelaide, Adelaide, Australia, July 2008. [Online]. Available at: <https://digital.library.adelaide.edu.au/dspace/bitstream/2440/49670/8/02whole.pdf>
- [7] D. Djebouri and M. Djebouri: GPS Sattelite Signal Synchronization. [Online]. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.625.9222&rep=rep1&type=pdf>
- [8] H. Lei, K. Yanhong and Z. Qishan *et al.*: Study on hardware realization of GPS signal fast acquisition. School of Electronics and Information Engineering, Beihang University, China, 2005.
- [9] M. Mouhamadou, P. Vaudon and M. Rammal: Smart antenna array patterns synthesis: Null steering and multi-user beamforming by phase control. *Progress In Electromagnetics Research*, vol. 60, pp. 95–106, 2006.

- [10] L. Yilong and Y. Beng-Kiong: Adaptive wide null steering for digital beam-forming array with the complex coded genetic algorithm. In: *Proceedings 2000 IEEE International Conference on Phased Array Systems and Technology (Cat. No.00TH8510)*, pp. 557–560. Dana Point, CA, USA, 2000.
- [11] M. K. Khawaldeh and D. I. Abu-Al-Nadi: Design of wideband beamforming using particle swarm optimization. In: *International Multi-Conference on Systems, Signals Devices*, pp. 1–5. 2012.
- [12] M.A. Nemade, Asst. Prof. D. G. Ganage and Dr. M. B. Mali: Analysis of LMS and NLMS Adaptive Beamforming Algorithms. *International Journal of Engineering Research & Technology (IJERT)*, vol. 3, no. 7, 2014.
- [13] Souden, M., Benesty, J. and Affes, S.: A study of the lcmv and mvdr noise reduction filters. *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 4925–4935, Sep 2010. ISSN 1053-587X.
- [14] Liang, Y.-C. and Chin, F.P.S.: Coherent lms algorithms. *IEEE Communications Letters*, vol. 4, no. 3, pp. 92–94, March 2000. ISSN 1089-7798.
- [15] Nan, H. and Yulin, H.: Research and comparison of anti-jamming performances of lms and lcmv algorithm based on matlab simulation. In: *IEEE 2011 10th International Conference on Electronic Measurement Instruments*, vol. 1, pp. 98–101. Aug 2011.
- [16] D. Lu, R. Wu, Z. Su and W. Huang: A two stage gps anti-jamming processor for interference suppression and multipath mitigation. In: *2007 IEEE Radar Conference*, pp. 746–749. Civil Aviation University of China, Tianjin,300300, P.R.China, April 2007.
- [17] E. Vijay, K. Madisetti and Douglas B. Williams: Detection: Determining the Number of Sources. *Digital Signal Processing Handbook*, 1999.
- [18] A.J. van Heusden: Automatic Gain Control ADC based on signal statistics for a cognitive radio cross-correlation spectrum analyzer. Master's thesis, University of Twente, Enschede, Netherlands, 2011.
- [19] Y. Vanderperren, G. Leus and W. Dehaene: An Approach for Specifying the ADC and AGC Requirements for UWB Digital Receivers. In: *2006 IET Seminar on Ultra Wideband Systems, Technologies and Applications*, pp. 196–200. London, UK, April 2006.
- [20] C.A. Balanis: *Antenna Theory: Analysis and Design*. 4th edn. Wiley, 28 December 2015.
- [21] P.S. Kildal: *Foundations of Antenna Engineering: A Unified Approach for Line-of-Sight and Multipath*. Kildal Antenna AB, 2015.
- [22] National Instruments: *AWR Microwave Office*. [Online]. Available at: <https://www.awrcorp.com/products/ni-awr-design-environment/microwave-office-software>

- [23] Dassault Systems: *CST Studio suite*. [Online].  
Available at: <https://www.cst.com/products/csts2>
- [24] Altair: *Altair FEKO*. [Online].  
Available at: <https://altairhyperworks.com/product/FEKO>
- [25] P.S. Nakar: Design of a Compact Microstrip Patch Antenna for Use in Wireless/Cellular Devices. 2004. [Online].  
Available at: <http://diginole.lib.fsu.edu/islandora/object/fsu:181039/datastream/PDF/view>
- [26] D. Ramakrishna: Design, testing and assembly of rectangular patch antenna. 2007. [Online].  
Available at: [http://ethesis.nitrkl.ac.in/4389/1/Design\\_2C\\_testing\\_and\\_assembly\\_of\\_rectangular\\_patch\\_antenna.pdf](http://ethesis.nitrkl.ac.in/4389/1/Design_2C_testing_and_assembly_of_rectangular_patch_antenna.pdf)
- [27] D.M. Pozar and S.M. Duffy: A dual-band circularly polarized aperture-coupled stacked microstrip antenna for global positioning satellite. *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 11, pp. 1618–1625, Nov 1997.
- [28] D. M. Pozar: *Microwave Engineering*. JohnWileySons Inc., 2012.
- [29] N. Hasan and Dr. S.C. Gupta: Corner truncated microstrip patch antenna. *International Journal of Advanced Technology in Engineering and Science*, vol. 2, no. 08, 2014.
- [30] K.S. Aung, S.S. Mon: Comparison of rectangular and truncated rectangular patch antenna for ku-band. *International Journal of Electronics and Computer Science Engineering*, vol. 3, no. 02.
- [31] W.V. Tilston, H.V. Cottony *et al.*: *IEEE Standard Definitions of Terms for Antennas*. The Institute of Electrical and Electronics Engineers, Inc.
- [32] A. Kavya, V. Poornima, Z.C. Alex and K. Shambavi: Design of a miniaturized dual band patch antenna for wlan applications. In: *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*, pp. 516–519. Piscataway, New Jersey, Feb 2015.
- [33] I. Ali and R.Y. Chang: Design of dual-band microstrip patch antenna with defected ground plane for modern wireless applications. In: *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*, pp. 1–5. Boston, USA, Sept 2015.
- [34] L. Zhang, Q. Zhang and C. Hu: The influence of dielectric constant on bandwidth of U-notch microstrip patch antenna. In: *2010 IEEE International Conference on Ultra-Wideband*, vol. 1, pp. 1–4. Mandarin Garden Hotel Nanjing, China, Sept 2010. ISSN 2162-6588.

- [35] L. Wen, W. Min, J. Y. Jin and C. Zhao: Design of a wideband circularly polarized microstrip antenna for compass and GPS. In: *2013 International Conference on Computational Problem-Solving (ICCP)*, pp. 195–198. Jiuzhai Valley, China, Oct 2013.
- [36] C. Shaw, M. Scheel, and T. Molteno: Transient Array Radio Telescope: Design. In: *21st Electronics New Zealand Conference, 2014*, pp. 24–33. Electronics New Zealand Inc. (ENZCon Inc, University of Waikato, Hamilton, New Zealand. [Online]. Available at: [http://www-ist.massey.ac.nz/enzcon/Past\\_Proceedings/Proceedings\\_2014.pdf](http://www-ist.massey.ac.nz/enzcon/Past_Proceedings/Proceedings_2014.pdf)
- [37] T. B. Lavate and V. K. Kokate and A. M. Sapkal: Performance Analysis of MUSIC and ESPRIT DOA Estimation Algorithms for Adaptive Array Smart Antenna in Mobile Communication. In: *2010 Second International Conference on Computer and Network Technology*, pp. 308–311. Bangkok, Thailand, April 2010.
- [38] J. Volakis: *ANTENNA ENGINEERING HANDBOOK*. McGraw Hill, 2007.
- [39] F. Akbari and S.S. Moghaddam and V.T. Vakili: MUSIC and MVDR DOA estimation algorithms with higher resolution and accuracy. In: *2010 5th International Symposium on Telecommunications*, pp. 76–81. Tehran, Iran, Dec 2010.
- [40] I.A H. Adam and M.D.R. Islam: Performance Study of Direction of Arrival (DOA) Estimation Algorithms for Linear Array Antenna. In: *2009 International Conference on Signal Processing Systems*, pp. 268–271. Singapore, May 2009.
- [41] T. S. Dhope, Prof. D. Simunic and Asst.Prof. Z. Radovan: Comparison of DoA Estimation Algorithms in SDMA System. *Automatika*, vol. 54, no. 2, pp. 199–209, 2013. [Online]. Available at: <https://doi.org/10.7305/automatika.54-2.131>
- [42] H.K. Hwang, Z. Aliyazicioglu, M.G. and Anatoly, Y.: Direction of arrival estimation using a root-MUSIC algorithm. In: *Proceedings of the international MultiConference of Engineers and Computer Scientists*, vol. 2. Citeseer, Hong Kong, 2008.
- [43] R. J. Weber and Y. Huang: Analysis for Capon and MUSIC DOA estimation algorithms. In: *2009 IEEE Antennas and Propagation Society International Symposium*, pp. 1–4. Charleston, SC, USA, June 2009. ISSN 1522-3965.
- [44] O.A. Oumar, M.F. Siyau and T.P. Sattar: Comparison between MUSIC and ESPRIT direction of arrival estimation algorithms for wireless communication systems. In: *The First International Conference on Future Generation Communication Technologies*, pp. 99–103. London, UK, Dec 2012. ISSN 2377-262X.

- [45] M.U. Shahid, M. Nauman, D. Haider and Y. Imran: Comparative analysis between direction of arrival algorithms. In: *2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS)*, pp. 451–454. Amity University Dubai, Dubai, Dec 2017.
- [46] Y. Jung-Lang and C.C. Yeh: Generalized eigenspace-based beamformers. *IEEE Transactions on Signal Processing*, vol. 43, no. 11, pp. 2453–2461, Nov 1995.
- [47] Z. Hongwei, L. Baowang and F. Juan: Adaptive Beamforming Algorithm for Interference Suppression in GNSS Receivers. *IJCSIT*, vol. 3, no. 5, Oct 2011.
- [48] R. Lang, H. Xiao, Z. Li and L. Yu: A anti-jamming method for satellite navigation system based on multi-objective optimization technique. *PloS one*, vol. 12, no. 7, p. e0180893, 2017.
- [49] M.S. Oude Alink, A.B.J. Kokkeler, E.A.M. Klumperink, K.C. Rovers, G.J.M. Smit and B. Nauta: Spurious-Free Dynamic Range of a Uniform Quantizer. *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 56, no. 6, pp. 434–438, June 2009. ISSN 1549-7747.
- [50] L. Chen, J. Zheng, M. Su and S. Huang: A Novel Beamforming Technique: Introducing a Convex Constrained Optimization and Compressed-Sensing Model. *IEEE Antennas and Propagation Magazine*, vol. 58, no. 4, pp. 48–59, Aug 2016. ISSN 1045-9243.
- [51] H. Sun, A.N. and Jiang, J.: *Sub-Nyquist Sampling and Compressed Sensing in Cognitive Radio Networks*. Springer, Berlin, Heidelberg, 2014.