THE DEVELOPMENT OF A SMALL SCALE RADIO ASTRONOMY IMAGE SYNTHESIS ARRAY FOR RESEARCH IN RADIO FREQUENCY INTERFERENCE MITIGATION

by

Jacob Lee Campbell

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Electrical and Computer Engineering

Brigham Young University

December 2005

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Jacob Lee Campbell

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date	Karl F. Warnick, Chair
Date	Brian D. Jeffs
Date	Michael A. Jensen

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Jacob Lee Campbell in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date

Karl F. Warnick
Chair, Graduate Committee

Accepted for the Department

Michael A. Jensen
Graduate Coordinator

Accepted for the College

Alan R. Parkinson
Dean, Ira A. Fulton College of Engineering and Technology

ABSTRACT

THE DEVELOPMENT OF A SMALL SCALE RADIO ASTRONOMY IMAGE SYNTHESIS ARRAY FOR RESEARCH IN RADIO FREQUENCY INTERFERENCE MITIGATION

Jacob Lee Campbell

Department of Electrical and Computer Engineering

Master of Science

Radio astronomy synthesis imaging arrays are composed of many parabolic reflector antennas. These antennas are designed to be extremely sensitive to detect faint emissions from astronomical sources. Unfortunately, this also makes them susceptible to radio frequency interference (RFI) from man made sources such as orbiting satellites.

The radio astronomy research group at Brigham Young University (BYU) is investigating methods to mitigate the effects of RFI in radio astronomy synthesis imaging. Though real-time RFI mitigation has been demonstrated for a large single dish telescope, for synthesis imaging arrays our prior work has consisted solely of algorithm development and computer simulations. To test our algorithms on experimental data we need an image synthesis array at BYU. The primary contribution of this Master's thesis is the design and construction of a working image synthesis array on the roof of the Clyde Building at BYU.

This thesis describes the design of the antenna placement for the synthesis array. Antenna placement is the primary factor for determining image quality since the placement dictates the shape of the synthesized beam. Simulations were performed, prior to the array's construction, to predict the quality of images from the array. Another contribution of this thesis is signal processing code to generate correlations of the signals from the antennas. Code was written to calibrate measured data and generate an image from the correlations. Code was also written to steer the antennas and track astronomical phenomena.

The performance of the array is evaluated in this thesis. The culmination of this thesis is a radio image of the supernova remnant Cassiopeia A. This thesis concludes with simulations of an RFI mitigation experiment that can be performed with the new array (pending certain improvements to the array).

ACKNOWLEDGMENTS

Many people have helped me with my thesis.

I want to thank Dr. Karl Warnick for his guidance through my research, this thesis, and through my graduate studies.

I would also like to thank Dr. Brian Jeffs for recruiting me to this project, and for his advice in my research.

I express my appreciation to Andy Poulsen, Chad Hansen, and Devin Pratt who helped me with my research and homework.

I would also like to thank James Nagel and Micah Lilrose who helped me build and maintain the VSA. I leave the array in their capable hands.

Above all I would like to thank my family and especially my parents for their love, prayers, and support. Thank you Rod, Janet, Jesse, Andrew, Mark, Lisa, Cari, and Sharon.

This thesis is dedicated to Rod and Janet Campbell.

Contents

Abstract **ix**

Table of Contents **xv**

List of Tables **xxi**

List of Figures **xxiii**

1 Masters Thesis Introduction **1**

1.1 Radio Astronomy Synthesis Imaging 1

1.2 Radio Interference in Radio Astronomy 2

1.3 Thesis Contributions 3

1.4 Thesis Organization 3

2 Mathematical Foundations and Background **5**

2.1 Synthesis Imaging Formulation 5

2.1.1 General Formulation of Observed Field 5

2.1.2 Simplifying Assumption 1: Ignore Polarization 7

2.1.3 Simplifying Assumption 2: Immense Distance to Source 7

2.1.4 Simplifying Assumption 3: Celestial Sphere is Empty 7

2.1.5 Measuring Spatial Coherence 8

2.1.6 Simplifying Assumption 4: Spatial Correlation 9

2.1.7 Observed Intensity 10

2.1.8 Coordinate System for Imaging, Part I 10

2.1.9 Coordinate System for Imaging, Part II 11

2.1.10 Error Bounds for Small Field Imaging 14

2.1.11 A Note on Antenna Response 14
2.2 The Interferometer and Measured Visibility 15
2.3 Computing u and v 17
2.4 Fourier Transform Imaging 21
2.4.1 Measuring Spatial Frequency 22
2.4.2 Sampling the uv plane 23
2.4.3 Inverting the Transform 24

3 Array Design 27

3.1 Introduction 27
3.2 Early Design Decisions 27
3.3 Antenna Placement 28
3.3.1 Placement Logistics 28
3.3.2 Final Placement Concerns 31
3.3.3 U V Plane Sampling 31
3.4 Receiving and Processing 38
3.4.1 Analog Receiver 38
3.4.2 Signal Processing 39

4 Simulation and Analysis 41

4.1 Introduction 41
4.2 The Noise Free Model: Dirty Beam and Resolution 41
4.3 Model with Noise 46
4.3.1 Noise in Radio Astronomy 46
4.3.2 Simulations with Noise 49
4.4 Phase Stability Simulations 51
4.4.1 Random Phase Error 55
4.4.2 Calibration Offset Error 55
4.4.3 Linearly Drifting Phase 56

5 Performance **61**

5.1 Introduction . . . 61

5.2 The Completed VSA . . . 61

5.2.1 The Antennas . . . 61

5.2.2 Software . . . 66

5.3 First Fringe Function . . . 69

5.4 Observing a Calibrator with the VSA . . . 70

5.4.1 Flagging and Editing Data . . . 70

5.4.2 Temporal Data . . . 71

5.4.3 Phase and Amplitude Analysis . . . 72

5.4.4 Measured versus Predicted . . . 73

5.4.5 Cross Coupling . . . 74

5.4.6 DC Offset Calibration . . . 76

5.4.7 Phase Offset Calibration . . . 77

5.5 An Image of Cassiopeia A . . . 77

6 RFI Experiments **87**

6.1 Introduction . . . 87

6.2 Subspace Projection . . . 87

6.3 Cross Subspace Projection . . . 89

6.4 Simulation of CSP for the VSA . . . 90

7 Conclusions and Future Work **97**

7.1 Imaging a high SNR Source . . . 97

7.2 Imaging Other Astronomical Phenomena . . . 97

7.3 RFI Scenarios . . . 98

7.4 Improvements to the Array . . . 99

7.5 Conclusions . . . 100

A Operating Procedures and Guidelines **103**

A.1 Outdoor Equipment . . . 103

A.2 Equipment Check 105
A.3 Pointing Calibration 107
A.4 Weather Conditions 109
A.5 Using the Pentek DSP 110
A.6 Checklist 110
A.7 Antennas Tracking 112

B VSA Software **115**
B.1 Code for the DSP 115
B.1.1 C Source 115
B.1.2 Intermediate and Output Files 118
B.1.3 Helper Code 119
B.1.4 Initial Data Analysis Code 121
B.2 Observation Code 123
B.2.1 Astro and satellite 123
B.2.2 Tracking and Steering Software 124
B.2.3 Weather Software 128
B.2.4 Management Scripts for Executables 129
B.3 Analysis and Image Generation Code 129
B.3.1 Making an Image 129
B.3.2 Initial Analysis 130
B.3.3 cruncher.m 130
B.3.4 gather.m 133
B.3.5 Tools for Imaging 134

C Recommended Improvements **137**
C.1 Problems 137
C.1.1 Cross Talk 137
C.1.2 Antenna Positions 138
C.1.3 Pentek DSP Issues 138
C.1.4 Absolute Gain Calibration 141

C.2 General Improvements 141
C.2.1 Fiber Optic Cable 142
C.2.2 A New Host PC 142
C.2.3 Software Improvements 142

D DSP Technical Literature **145**
D.1 Introduction 145
D.1.1 Assumptions 145
D.1.2 Our Hardware 146
D.1.3 Finding Documentation 146
D.1.4 Appendix Structure 146
D.2 P6216: Digital Drop Receiver 147
D.3 P4291: Quad Processor DSP Board 147
D.3.1 Hardware Documentation 147
D.3.2 Software Documentation 148
D.4 TMSC320C6701 Floating Point Processor 148
D.4.1 Introduction 148
D.4.2 Hardware Documentation 148
D.4.3 Software Documentation 149
D.5 Transferring Data to the Host PC 150
D.5.1 Swiftnet Application Program Interface 150

E Bibliography Addendum **151**

Bibliography **153**

List of Tables

5.1 Coordinates of each antenna with respect to Antenna 1 (specified in meters). 66
A.1 Results of surveying. 108
A.2 Weather thresholds for the VSA. 109

List of Figures

2.1 The general setup for synthesis imaging. 6
2.2 The field on the surface of the celestial sphere. 8
2.3 A plane wave impinging on two antennas at oblique incidence. 9
2.4 Direction cosines of the vector **s**. 12
2.5 Coordinate system for small field imaging. 12
2.6 An interferometer block diagram. 15
2.7 A block diagram of an interferometer with down conversion. 17
2.8 A baseline projected onto the uv plane. 18
2.9 An east-west linear array. 20
2.10 Baselines for the linear array in Figure 2.9. 20
2.11 Linear array uv coverage for 40° declination. 21
2.12 Linear array uv coverage for 15° declination. 22
2.13 Dirty beam for the uv coverage in Figure 2.11. 25
3.1 Block diagram test platform concept. 28
3.2 Satellite photo of the Clyde Building [1]. 29
3.3 Clyde Building roof layout [2]. 30
3.4 A view of the eastern sky at 35 ft away from the fifth floor. 32
3.5 A view of the eastern sky at 50 ft away from the fifth floor. 32
3.6 Available area for antenna placement. 33
3.7 Final configuration choices. 34
3.8 uv coverage for option 1 at declination 20 degrees. 35
3.9 uv coverage for option 1 at declination 40 degrees. 35
3.10 uv coverage for option 1 at declination 60 degrees. 36
3.11 uv coverage for option 2 at declination 20 degrees. 36

3.12 uv coverage for option 2 at declination 40 degrees. 37
3.13 uv coverage for option 2 at declination 60 degrees. 37
3.14 Dirty beam for the coverage in Figure 3.8 (option 1). 38
3.15 Dirty beam for the coverage in Figure 3.11 (option 2). 39
3.16 The final placement of antennas for the VSA. 40
4.1 uv coverage for 16 hours. 43
4.2 Dirty beam for uv coverage in Figure 4.1 44
4.3 Slice through the dirty beam along the l and m axes. 45
4.4 Two sources: slice through the m axis, separation 0.16 degrees. 46
4.5 Two sources: slice through the m axis, separation 0.12 degrees. 47
4.6 Simulated dirty image using noise model. 51
4.7 Fringe function for the signal between antennas 1 and 4. 52
4.8 Fringe function generated with the stochastic model of Equation 4.9. 53
4.9 The error free image I_0. 54
4.10 Error for Gaussian and uniform phase noise at every E field sample. . 56
4.11 Image with Gaussian error in phase 57
4.12 Error for Gaussian and uniform chosen phase offset. 58
4.13 Image with Gaussian error in phase offset, $\sigma = 10$ degrees. 59
4.14 Image error for linearly drifting phase error. 60
5.1 Antenna 1 tracking Cassiopeia A 62
5.2 Antenna 2 tracking Cassiopeia A 63
5.3 Antenna 3 and antenna 4 tracking Cassiopeia A 64
5.4 A close up of antenna 1 (Eve). 65
5.5 The Walkenhorst/Hansen receiver. 66
5.6 The position controllers for the antenna positioners and the LNA power supplies. 67
5.7 The host PC and the Pentek DSP. 67
5.8 A screen shot of the controlcenter GUI. 68
5.9 The first fringe function observed with the VSA. The array was observing Cassiopeia A. 71

5.10 Time data for the real and imaginary outputs from each of the four channels. 72
5.11 Five channels of corrupted phase data. 73
5.12 Five channels of corrupted modulus data 74
5.13 Predicted and measured phase for baseline 2,3 75
5.14 Predicted real and imaginary correlator outputs for baseline 3,4. . . . 76
5.15 Measured real and imaginary correlator outputs for baseline 3,4. . . . 77
5.16 A GPS satellite spectrum from channel 3 of the Walkenhorst/Hansen receiver. 78
5.17 A GPS satellite spectrum from channel 2 of the Walkenhorst/Hansen receiver. 79
5.18 Predicted and measured phase for baseline 3,4 with DC offset removed. 80
5.19 Unedited cross correlation data for baseline 1,3. 81
5.20 Effects of phase offset calibration for baseline 1,3. 82
5.21 uv plane coverage for the dirty image in Figure 5.23. 83
5.22 Dirty beam for the uv coverage in Figure 5.21. 84
5.23 Dirty image of Cassiopeia A. 85
6.1 Interference-free image of three equal power point sources. 91
6.2 Interference corrupted image of three equal power point sources. . . . 92
6.3 The angular separation between the phase tracking center and the interferer. 93
6.4 Interference to noise ratio in the feeds of the synthesis array antennas. 94
6.5 An image filtered using CSP. 95
A.1 The LNAs in the feed of antenna 2. 104
A.2 The azimuth positioner for antenna 3. 104
A.3 The elevation positioner for antenna 3. 105
A.4 The repeater box for antenna 2. 106
A.5 The cables for antenna 2. 106
A.6 Antenna cables enter the building. 107
A.7 The coordinate system for surveying. 108

A.8 Antenna 4 (Rachel) tracking Cassiopeia A in the early morning, April 2005. 113
A.9 Antenna 3 (Rebekah), in the foreground, and antenna 1 (Eve), in the background, tracking Cassiopeia A in the early morning, April 2005. . 114
B.1 Inter-processor setup for the correlator. The inter-processor bififo between processors C and D is not shown. 116
B.2 DSP with four-way splitter on the inputs to each channel. 122
B.3 The last two buffers of sampled data for each channel using a signal generator as an input. 123
B.4 Phase between channels of the correlator using a signal generator as an input. 124
B.5 A close up of the controlcenter GUI. 125
B.6 Unedited cross correlation data for baseline 1,3. 132
B.7 Edited cross correlation data for baseline 1,3. 133
C.1 A GPS satellite spectrum from channel 1 of the Walkenhorst/Hansen receiver. 138
C.2 A GPS satellite spectrum from channel 2 of the Walkenhorst/Hansen receiver. 139
C.3 A GPS satellite spectrum from channel 3 of the Walkenhorst/Hansen receiver. 140
C.4 A GPS satellite spectrum from channel 4 of the Walkenhorst/Hansen receiver. 141

Chapter 1

Masters Thesis Introduction

1.1 Radio Astronomy Synthesis Imaging

Radio astronomy is the study of the sky at radio frequencies. Synthesis imaging arrays are one of the primary tools used in radio astronomy. They are used to produce radio images of the sky with a high angular resolution.

Synthesis imaging in radio astronomy has a rich history of mathematical, scientific, and technological advances and discoveries. Imaging arrays are composed of interferometers and developments in radio imaging have often paralleled previous work in optical interferometry. Synthesis imaging has roots in the optical interferometry work of Michelson and Pease. An early radio interferometer was built by Ryle and Vonberg in 1946 using two dipole antenna arrays. Interferometric arrays were developed in the 1950s, and the imaging arrays of today were developed in the 1960s and 1970s [3].

Modern arrays are sophisticated and complicated instruments that are composed of radio telescopes, receiving hardware, and data processing hardware. Typically, radio telescopes are high gain parabolic dish antennas. These are usually equipped with liquid helium-cooled, low noise amplifiers making radio telescopes extremely sensitive. The antennas of an array may be spread out over kilometers, tens of kilometers, or thousands of kilometers. Weather conditions and man made interference are monitored at the array location. The signals from the antennas are received and processed. The image formation for modern arrays is often performed on supercomputers.

The most well known system is the Very Large Array (VLA) in Socorro, New Mexico. The VLA is composed of 27 parabolic dish antennas. Each dish is fully steerable, 25 meters in diameter, and weighs approximately 230 tons. They are spread out over 36 kilometers [4].

Some educational institutions and amateur astronomers have attempted to duplicate interferometers and synthesis arrays on a small scale. For instance, the National Radio Astronomy Observatory (NRAO) provides a course on radio astronomy for educators in which they use a ”2-element Instructional Interferometer” [5] at Socorro, New Mexico. This interferometer has two 2.1 meter dishes that 24 meters apart. Another example is the Small Radio Telescope interferometer at Haystack Observatory [6] which uses 2.3 meter dishes and operates in L band.

This thesis describes the development of a small scale imaging array built on a small budget for educational purposes.

1.2 Radio Interference in Radio Astronomy

The small array will also be used for research in interference mitigation. As previously mentioned, synthesis arrays are composed of extremely sensitive antennas. This sensitivity makes them vulnerable to radio frequency interference (RFI). The concept of eliminating or ameliorating the effects of RFI is called RFI mitigation.

RFI in radio astronomy comes from many sources. These sources vary depending on the location of the telescope and the frequencies the telescope operates at. RFI sources include, but are certainly not limited to, FM radio, cellular phone base stations, television broadcast, radar, and orbiting satellites.

The methods used to mitigate RFI are as varied as the sources of RFI. The major classes of RFI mitigation include detection and time blanking, subtraction methods in which a copy of the interference is removed from the data, adaptive beam forming, and spatial filtering by subspace projection methods.

Our research here at Brigham Young University is directed toward developing and testing RFI mitigation methods (see [7], [8], and [9]). The array described by this thesis will allow us to test our mitigation algorithms using real data.

1.3 Thesis Contributions

The primary contribution of this thesis is development of a working small scale radio interferometric array to test RFI mitigation algorithms and to serve as an educational instrument for astronomy and engineering students. This thesis describes the design of the antenna placement, which is fundamental to image synthesis. It also documents hardware design decisions such as the choice of antennas, azimuth and elevation positioners, and LNAs. This thesis also briefly discusses the construction of the array.

Another contribution of this thesis is software development to run the synthesis array. Matlab code was written to steer the antennas. C code was written for the DSP data acquisition platform the array uses. Matlab code was written to analyze measured data and generate images.

Simulations were performed, prior to the array's construction, to characterize the array and predict its performance. In particular, this thesis presents simulations to assess the array's imaging capabilities in the presence of phase error and phase instability. This thesis also includes simulations of an RFI algorithm called cross subspace projection (CSP) [9]. CSP is an algorithm we wish to test on the array.

Lastly, the appendices of this thesis provide documentation for operating and maintaining the array.

1.4 Thesis Organization

This thesis is organized as follows. Chapter 2 covers the mathematical formulation of synthesis imaging and provides mathematical background for the chapters that follow. Chapter 3 documents the design of the array, with particular emphasis antenna placement. Chapter 4 presents a number of simulations that were performed to characterize the array. Chapter 5 describes the completed array, early observations made with the array, and displays the first synthesized image made with the array. Chapter 6 documents simulations of the CSP interference mitigation algorithm. Chapter 7 provides some conclusions and proposals of future experiments for the array.

Chapter 2

Mathematical Foundations and Background

This chapter is intended to provide mathematical background for the chapters that follow. It will review image synthesis in radio astronomy, the uv plane, interferometry, and Fourier transform imaging.

2.1 Synthesis Imaging Formulation

The treatment of synthesis imaging presented here closely follows that of Clark [10]. Extra detail will be presented for the benefit of those who will continue to work on the project.

Clark formulates synthesis imaging by deriving the form of the observed electric field, from which he derives the correlated field. Then he recasts the correlated field and introduces the uv plane to arrive at the two-dimensional Fourier transform relationship between observed visibility and astronomical sky intensity. We will follow Clark's treatment by beginning from a general setup, and as the derivation proceeds, introduce the assumptions used to reach the desired relationship between visibility and intensity, which are defined below.

2.1.1 General Formulation of Observed Field

We begin by finding the form of the observed electric field radiated by an astronomical source. Figure 2.1 shows the experimental setup of synthesis imaging. Some distant astronomical phenomena is located at position vector $\mathbf{R}$, referred to by Clark as "IT". The astronomical phenomena emits energy at radio frequencies, and

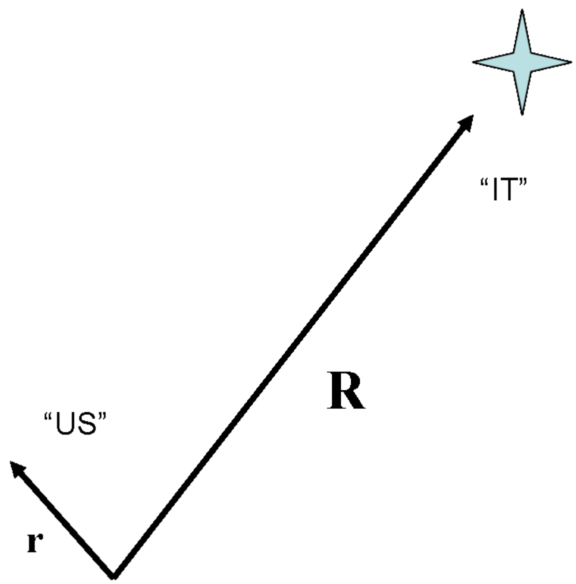

Figure 2.1: The general setup for synthesis imaging.

this electromagnetic radiation is observed at position $\mathbf{r}$ by "US". The field radiated by IT is denoted $\mathbf{E}(\mathbf{R}, t)$.

If $\mathbf{E}(\mathbf{R}, t)$ is of limited duration in time, it can be decomposed into its equivalent frequency representation as a sum of weighted complex exponentials. This form of the field is much more convenient to deal with. The analysis that follows deals with only one of the frequency components, in other words, the source radiates at a single frequency. If the source contains multiple frequency components, which is more realistic, the analysis ends by summing over all frequencies to obtain the total response.

The single frequency component is given by $\mathbf{E}_\nu(\mathbf{R})$, where ν is frequency in Hz. The electric field is observed at $\mathbf{r}$ by an antenna. Since Maxwell's laws are linear, the response at $\mathbf{r}$ is the superposition of all fields at that location, and the measured field takes the following form:

$$\mathbf{E}_\nu(\mathbf{r}) = \int\int\int \mathbf{P}(\mathbf{R}, \mathbf{r})\mathbf{E}_\nu(\mathbf{R})\, dR, \tag{2.1}$$

where the integration in Equation 2.1 takes place over all of space. $\mathbf{P}(\mathbf{R}, \mathbf{r})$ is called the propagator and it describes how the fields at $\mathbf{r}$ are influenced by the fields at $\mathbf{R}$. Eventually the formulation will show that the propagator is a Green's function, but at this point it is best thought of as a spatial impulse response in a convolution integral.

2.1.2 Simplifying Assumption 1: Ignore Polarization

The first simplifying assumption we will make is, as Clark points out, mostly pedagogical. We will ignore the fact that the electric field is a vector quantity, or in other words, we will ignore polarization. This allows us to treat $\mathbf{E}$ and $\mathbf{P}$ as scalar quantities and makes the mathematics more intuitive. Otherwise, we would have to deal with the tensor form of Equation 2.1.

2.1.3 Simplifying Assumption 2: Immense Distance to Source

The second assumption we make is that the immense distance to the source precludes all hope of resolving any depth features. Instead, we attempt to recover information about the surface distribution of the field over a celestial sphere at radius $\mathbf{R_0}$.

With these two simplifications, Equation 2.1 becomes

$$E_\nu(\mathbf{r}) = \int\int P(\mathbf{R}_0, \mathbf{r}) E_\nu^{cs}(\mathbf{R}_0) dS, \tag{2.2}$$

where the integration now takes place over the surface of the celestial sphere. The physical picture for Equation 2.2 is shown in Figure 2.2.

2.1.4 Simplifying Assumption 3: Celestial Sphere is Empty

The third simplifying assumption we will make is that the space inside the celestial sphere is empty. Strictly speaking, the space inside the celestial sphere is not entirely empty, our home planet being the most obvious exception to this assumption. However, it is mostly empty, and this assumption allows us to invoke Huygens' principle to give us the form of the propagator:

$$P(\mathbf{R}, \mathbf{r}) = \frac{e^{j2\pi\nu|\mathbf{R}-\mathbf{r}|/c}}{|\mathbf{R}-\mathbf{r}|}. \tag{2.3}$$

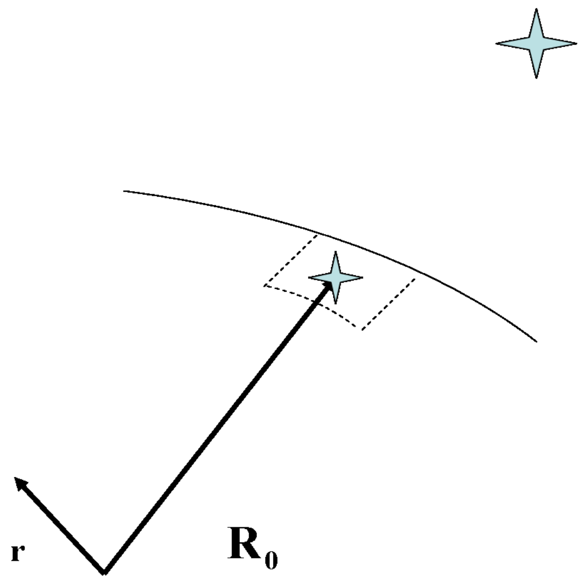

Figure 2.2: The field on the surface of the celestial sphere.

When we insert the form of the propagator into Equation 2.2, the form of the observed field at location $\mathbf{r}$ becomes

$$E(\mathbf{r}) = \int E_\nu^{cs}(\mathbf{R_0}) \frac{e^{j2\pi\nu|\mathbf{R_0}-\mathbf{r}|/c}}{|\mathbf{R_0}-\mathbf{r}|} dS, \tag{2.4}$$

where $E_\nu^{cs}(\mathbf{R_0})$ denotes the field on the celestial sphere. Equation 2.4 is the field that is measured at one location, i.e. by one antenna in a synthesis array.

2.1.5 Measuring Spatial Coherence

By itself, the observed field is not terribly informative: we only have an amplitude and a phase measurement. In synthesis imaging phases and amplitudes from many different antennas are combined to form an image. Specifically, we combine these measurements to sample the spatial coherence between the signals at different antennas.

To measure spatial coherence between two antennas their respective signals are correlated as $V_\nu(\mathbf{r_1}, \mathbf{r_2}) = \langle E(\mathbf{r}_1) E^*(\mathbf{r}_2) \rangle$. The asterisk indicates complex conjugation. The measured correlations are called visibilities.

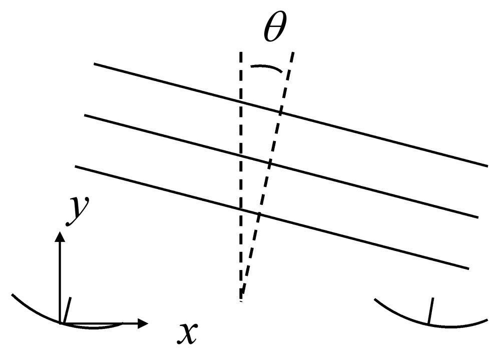

Figure 2.3: A plane wave impinging on two antennas at oblique incidence.

To provide some intuition into what spatial coherence is, consider the following example. Figure 2.3 shows a plane wave impinging on two antennas located on the x axis separated by a distance d. For simplicity, the field from the plane wave is time harmonic and the amplitude of the wave is equal to A. The field at each antenna is given by $Ae^{j\mathbf{k}\cdot\mathbf{r_i}}$. For the antenna on the left $\mathbf{r_1} = (0,0)$ and for the antenna on the right $\mathbf{r_2} = (d,0)$. The spatial coherence function for these antennas is given by $V(\mathbf{r_1},\mathbf{r_2}) = A^2 e^{jkd\sin\theta}$. Notice the modulus of $V(\mathbf{r_1},\mathbf{r_2})$ provides information on the source amplitude. Also, the phase is 0 when $\theta = 0$, and increases with θ. Thus the phase of $V(\mathbf{r_1},\mathbf{r_2})$ gives information on the source location.

Returning to the formulation of synthesis imaging, the spatial coherence function for the observed field of Equation 2.4 is given by

$$V_\nu(\mathbf{r_1},\mathbf{r_2}) = \left\langle \int\int E_\nu^{cs}(\mathbf{R}_1)\frac{e^{j2\pi\nu|\mathbf{R}_1-\mathbf{r}_1|/c}}{|\mathbf{R}_1-\mathbf{r}_1|}E_\nu^{cs*}(\mathbf{R}_2)\frac{e^{-j2\pi\nu|\mathbf{R}_2-\mathbf{r}_2|/c}}{|\mathbf{R}_2-\mathbf{r}_2|}dS_1 dS_2\right\rangle \tag{2.5}$$

where $\mathbf{R}_1$ and $\mathbf{R}_2$ are points on the surface of the celestial sphere and dS_1 and dS_2 are the differential elements for $\mathbf{R_1}$ and $\mathbf{R_2}$, respectively.

2.1.6 Simplifying Assumption 4: Spatial Correlation

The final simplifying assumption is that radiation from different locations on the celestial sphere is spatially uncorrelated. In other words, $\langle E(\mathbf{R}_1), E^*(\mathbf{R}_2)\rangle = 0$ when $\mathbf{R}_1 \neq \mathbf{R}_2$. Using this final simplification and interchanging the order of averaging and integration yields

$$V_\nu(\mathbf{r_1}, \mathbf{r_2}) = \int \langle |E_\nu^{cs}(\mathbf{R})|^2 \rangle |\mathbf{R}|^2 \frac{e^{j2\pi\nu|\mathbf{R}-\mathbf{r}_1|/c}}{|\mathbf{R}-\mathbf{r}_1|} \frac{e^{-j2\pi\nu|\mathbf{R}-\mathbf{r}_2|/c}}{|\mathbf{R}-\mathbf{r}_2|} dS. \tag{2.6}$$

2.1.7 Observed Intensity

We will recast Equation 2.6 in order to simplify it and introduce the names used by astronomers for quantities in Equation 2.6 (see [11] [12]). The unit vector $\mathbf{s}$ is defined as

$$\mathbf{s} = \frac{\mathbf{R}}{|\mathbf{R}|}. \tag{2.7}$$

Observed intensity is defined as

$$I_\nu = |\mathbf{R}|^2 \langle |E_\nu(\mathbf{s})|^2 \rangle. \tag{2.8}$$

Lastly, the element solid angle is given by

$$dS = |\mathbf{R}|^2 d\Omega. \tag{2.9}$$

In simplifying Equation 2.6, the following two approximations are used: the far-field approximation: $|\mathbf{R} - \mathbf{r}| \approx |\mathbf{R}| - \frac{\mathbf{r}\cdot\mathbf{R}}{|\mathbf{R}|}$ (see [13]), and $|\mathbf{R} - \mathbf{r_2}| - |\mathbf{R} - \mathbf{r_1}| \approx |\mathbf{r_1} - \mathbf{r_2}|$ when $|\mathbf{R}| >> |\mathbf{r_1}|, |\mathbf{r_2}|$. This produces

$$V_\nu(\mathbf{r_1}, \mathbf{r_2}) = \int I(\mathbf{s}) e^{-j2\pi\nu\mathbf{s}\cdot(\mathbf{r_1}-\mathbf{r_2})/c} d\Omega. \tag{2.10}$$

This resembles a Fourier transform. Notice that $V_\nu(\mathbf{r_1}, \mathbf{r_2})$ is a function of $\mathbf{r_1} - \mathbf{r_2}$, the vector difference between the locations of the two antennas. Astronomers call the vector difference between two antennas a baseline. Denoting the baseline as $\mathbf{b} = \mathbf{r_1} - \mathbf{r_2}$ and inserting this in Equation 2.10 gives

$$V_\nu(\mathbf{r_1}, \mathbf{r_2}) = \int I(\mathbf{s}) e^{-j2\pi\nu\mathbf{s}\cdot\mathbf{b}/c} d\Omega. \tag{2.11}$$

2.1.8 Coordinate System for Imaging, Part I

There are two possible coordinate systems to cast Equation 2.10 as a 2-D Fourier transform. The first requires that constraining baselines to lie in a plane, referred to in astronomical literature as the uv plane. The second system requires that the angular extent of the image to be small.

All of the baselines in a plane gives

$$\mathbf{b} = \mathbf{r_1} - \mathbf{r_2} = \lambda(u, v, w = 0), \tag{2.12}$$

$$\mathbf{s} = (l, m, n = \sqrt{1 - l^2 - m^2}), \tag{2.13}$$

$$d\Omega = \frac{dldm}{\sqrt{1 - l^2 - m^2}}. \tag{2.14}$$

In Equation 2.12, λ is the wavelength for the observing frequency ν, and u, v, and w are the vector components of $\mathbf{b}$ measured in wavelengths. The w axis points toward the celestial north pole, and the uv plane is parallel to the earth's equator [14]. In Equation 2.13, l, m, and n are the direction cosines of $\mathbf{s}$ for the angles from the u, v, and w axes respectively, as shown in Figure 2.4 (see [15]).

This leads to the following form for V_ν:

$$V_\nu(u, v) = \int\int I_\nu(l, m) e^{-j2\pi(ul+vm)} \frac{dldm}{\sqrt{1 - l^2 - m^2}}, \tag{2.15}$$

which is a modified Fourier transform. The factor of $1/\sqrt{1 - l^2 - m^2}$ comes from computing the Jacobian for the transformation from spherical to rectangular coordinates. The inversion of Equation 2.15 is given by

$$I_\nu(l, m) = \sqrt{1 - l^2 - m^2} \int\int V_\nu(u, v) e^{-j2\pi(ul+vm)} dudv. \tag{2.16}$$

This form of the transform is only valid when the baselines of a synthesis array lie in a plane. Due to Earth rotation, this condition is only met for synthesis arrays whose antennas all lie in an East-West line [14].

2.1.9 Coordinate System for Imaging, Part II

For arrays that are not strictly East-West, we use a different relative coordinate system to cast Equation 2.10 as a Fourier transform. In this it is assumed that the angular extent of the portion of the sky being imaged is small. It is assumed that the

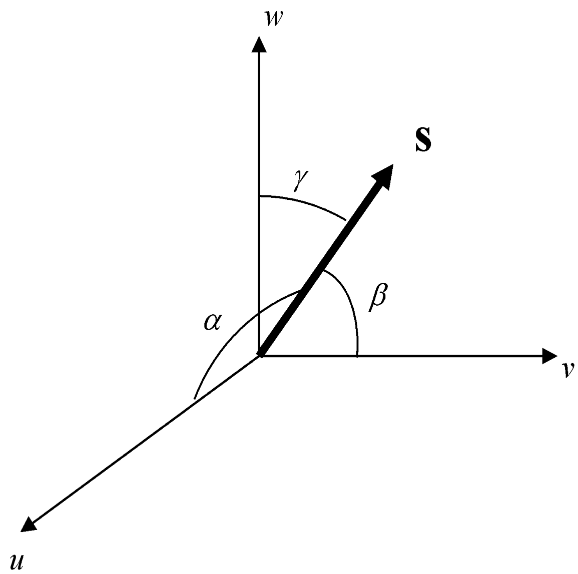

Figure 2.4: Direction cosines of the vector $\mathbf{s}$.

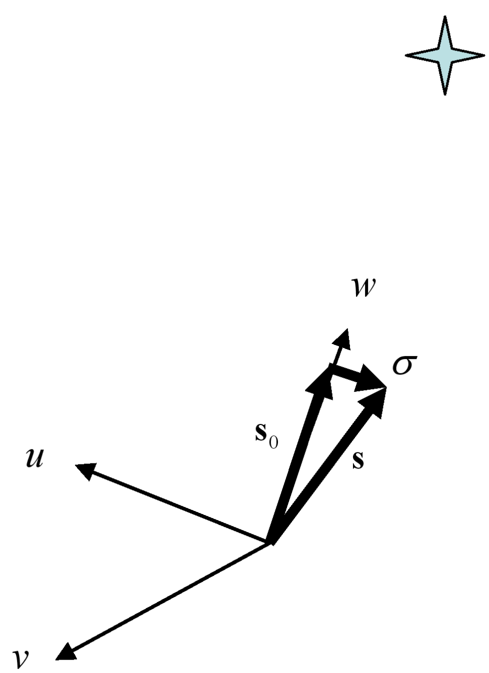

Figure 2.5: Coordinate system for small field imaging.

antennas will track the astronomical source and the w axis of this coordinate system is pointed toward the center of the source, as shown in Figure 2.5. $\mathbf{s}$ is written as

$$\mathbf{s} = \mathbf{s_0} + \sigma,$$
$$\mathbf{s_0} = (0, 0, 1),$$
$$\sigma = (l, m, 0).$$

The vector $\mathbf{s_0}$ is referred to as the phase tracking center and points at the center of the astronomical phenomena being imaged. Once again the baseline vector is defined as $\mathbf{b} = \mathbf{r_1} - \mathbf{r_2} = \lambda(u, v, w)$, and as before, Equation 2.10 becomes

$$V_\nu(\mathbf{r_1}, \mathbf{r_2}) = \int I(\mathbf{s}) e^{-j2\pi\nu \mathbf{s}\cdot\mathbf{b}/c} d\Omega. \tag{2.17}$$

Expanding the exponent gives

$$1/\lambda(\mathbf{s} \cdot \mathbf{b}) = 1/\lambda((\mathbf{s_0} + \sigma) \cdot \mathbf{b}),$$

$$1/\lambda(\mathbf{s_0} \cdot \mathbf{b}) = w,$$

$$1/\lambda(\sigma \cdot \mathbf{b}) = ul + vm.$$

In this coordinate system, the uv plane is perpendicular to $\mathbf{s_0}$. u and v are the projection of the baseline vector $\mathbf{b}$ along the u and v axis respectively. Substituting into Equation 2.10 yields a Fourier transform

$$V_\nu(u, v, w) = e^{-j2\pi w} \int\int I_\nu(l, m) e^{-j2\pi(ul+vm)} \frac{dldm}{\sqrt{1 - l^2 - m^2}}. \tag{2.18}$$

Since $|l^2 + m^2|$ is small, the factor of $1/\sqrt{1 - l^2 - m^2}$ can be neglected. In practice, the visibilities measured in Equation 2.18 are phase rotated by $e^{j2\pi w}$, so that Equation 2.18 is independent of w. Using these simplifications produces the two-dimensional transformation relationship that forms the basis for most synthesis imaging in radio astronomy:

$$V_\nu(u, v) = \int\int I_\nu(l, m) e^{-j2\pi(ul+vm)} dldm. \tag{2.19}$$

With enough samples of $V_\nu(u, v)$, we can invert the Fourier transform and obtain $I_\nu(l, m)$.

2.1.10 Error Bounds for Small Field Imaging

It is important to carefully consider the assumptions under which Equation 2.19 is valid. The assumption that both $\mathbf{s_0}$ and $\mathbf{s}$ are unit vectors can not be strictly true. This can be seen by considering for $|\sigma| = |(l, m, 0)| \neq 0$,

$$|\mathbf{s}| = |(l, m, 1)| \neq 1.$$

To use the small field approximation, we must place bounds on $|\sigma|$. In small field imaging, $\mathbf{s}$ is defined as in Equation 2.13, with l, m, and n being direction cosines from the u, v, and w axes respectively as shown in Figure 2.4. Now, if we evaluate the exponent in Equation 2.17, we find

$$(1/\lambda)\mathbf{s} \cdot \mathbf{b} = lu + mv + w(\sqrt{1 - (l^2 + m^2)}).$$

Equation 2.11 becomes

$$V_\nu(u, v, w) = \int\int I_\nu(l, m) e^{-j2\pi[ul+vm+w(\sqrt{1-(l^2+m^2)})]} \frac{dldm}{\sqrt{1 - l^2 - m^2}}. \tag{2.20}$$

The radical in the exponential of Equation 2.20 can be simplified using the approximation $\sqrt{1 - (l^2 + m^2)} \approx 1 - (l^2 + m^2)/2$, leading to

$$V_\nu(u, v, w) = e^{-j2\pi w} \int\int I_\nu(l, m) e^{-j2\pi[ul+vm-w(l^2+m^2)/2]} \frac{dldm}{\sqrt{1 - l^2 - m^2}}. \tag{2.21}$$

Note that $l^2 + m^2$ is the squared distance from the w axis to the point (l, m). Using $\sin(x) \approx x$ (a small angle approximation), $l^2 + m^2 \approx \theta^2$, where θ is the angle from the point (l, m) to the w axis. Using this approximation, Equation 2.21 becomes

$$V_\nu(u, v, w) = e^{-j2\pi w} \int\int I_\nu(l, m) e^{-j2\pi[ul+vm-w\theta^2/2]} \frac{dldm}{\sqrt{1 - l^2 - m^2}}. \tag{2.22}$$

The phase error for radiation from the point (l, m) is $\pi w(l^2 + m^2)$ [14]. At the edge of a synthesized image of width θ_f, the phase error is $\pi w \left(\frac{\theta_f}{2}\right)^2$ [12].

2.1.11 A Note on Antenna Response

The preceding derivation completely neglected the effect of the antenna reception pattern. If it is included in Equation 2.19, the form of the Fourier relationship

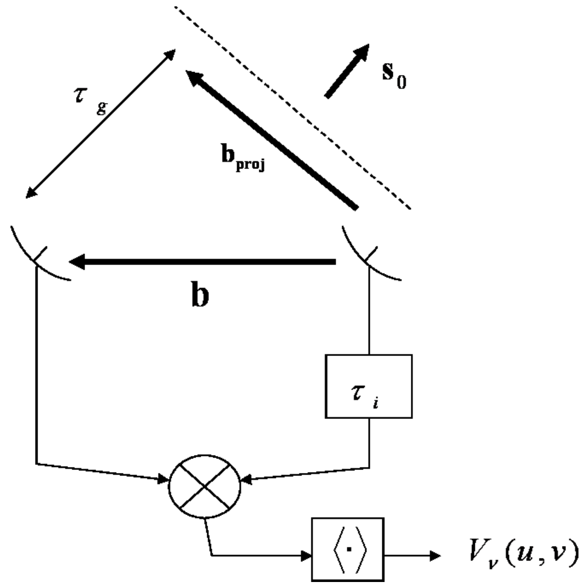

Figure 2.6: An interferometer block diagram.

becomes

$$V_\nu(u,v) = \int\int A_\nu(l,m) I_\nu(l,m) e^{-j2\pi(ul+vm)} dl dm.$$

The antenna reception pattern is what enables moving the limits of the integration to infinity, because the antennas reception pattern is close to zero outside of a small range of l and m values.

2.2 The Interferometer and Measured Visibility

This section will discuss some of the details of implementing an image synthesis array. In particular, this section will describe the basic building block of a synthesis array: the interferometer. This will be a brief discussion. More detailed treatments of an interferometer can be found in [14], [11], [12], and [16].

Each pair of antennas in an imaging array makes up an interferometer. Figure 2.6 shows the basic structure of an interferometer. An incoming plane wave is depicted as a dashed line. The plane wave reaches one of the antennas first, and the signal received by that antenna is delayed by τ_i to compensate for the geometrical delay τ_g.

The signals that reach the multiplier are thus time aligned, and are then multiplied and averaged to generate a measured visibility, V_ν.

Figure 2.6 shows some of the quantities previously described in this chapter. The delay inserted into right-side signal path accomplishes the phase rotation described in Section 2.1.9. The delay can be thought of as moving the right antenna into the uv plane. Indeed, if we measure τ_g in wavelengths, we find $\tau_g = \mathbf{s_0} \cdot \mathbf{b} = w$. Equivalently, the inserted delay moves the phase tracking center to zenith.

The projection of the baseline vector into the plane perpendicular to the phase tracking center $\mathbf{b_{proj}}$, is also shown. The values of u and v for a measured visibility are computed from $\mathbf{b_{proj}}$. Computing u and v for a pair of antennas will be covered in Section 2.3 in detail.

Usually, the signal processing shown in Figure 2.6 does not occur at the observing RF frequency. In practice, the signals are down converted to an intermediate frequency (IF) for analog to digital (A/D) conversion and signal processing. With these modifications, Figure 2.6 becomes Figure 2.7.

The down-conversion shown in Figure 2.7 is referred to as a low-side mix. In a low-side mix, the IF frequency is computed as $\omega_{IF} = \omega_{RF} - \omega_{LO}$. As labelled in the figure, the signals coming out of the antennas are $v_1 = Ae^{j\omega_{RF}(t-\tau_g)}$ and $v_2 = Ae^{j\omega_{RF}t}$. The signals at the input of the correlator multiplier preceding the averaging are

$$c_1 = Ae^{j\omega_{RF}(t-\tau_g)-j\omega_{LO}t} = Ae^{j\omega_{IF}t-j\omega_{RF}\tau_g}$$

and

$$c_2 = (Ae^{j\omega_{RF}t-j(\omega_{LO}(t+\phi))-j\omega_{IF}\tau_i})^* = Ae^{-j\omega_{IF}t+j\phi+j\omega_{IF}\tau_i}.$$

After correlating c_1 and c_2, the Visibility takes this form:

$$V_\nu(u,v) = |A|^2 e^{-j\omega_{RF}\tau_g+j\phi+j\omega_{IF}\tau_i}. \tag{2.23}$$

The inserted delay, τ_i, is set equal to τ_g making Equation 2.23 equal to

$$V_\nu(u,v) = |A|^2 e^{-j\omega_{LO}\tau_g+j\phi}. \tag{2.24}$$

ϕ is set to $\omega_{LO}\tau_g$ so that radiation from the direction of the phase tracking center has a phase of zero, the same phase as a source at zenith.

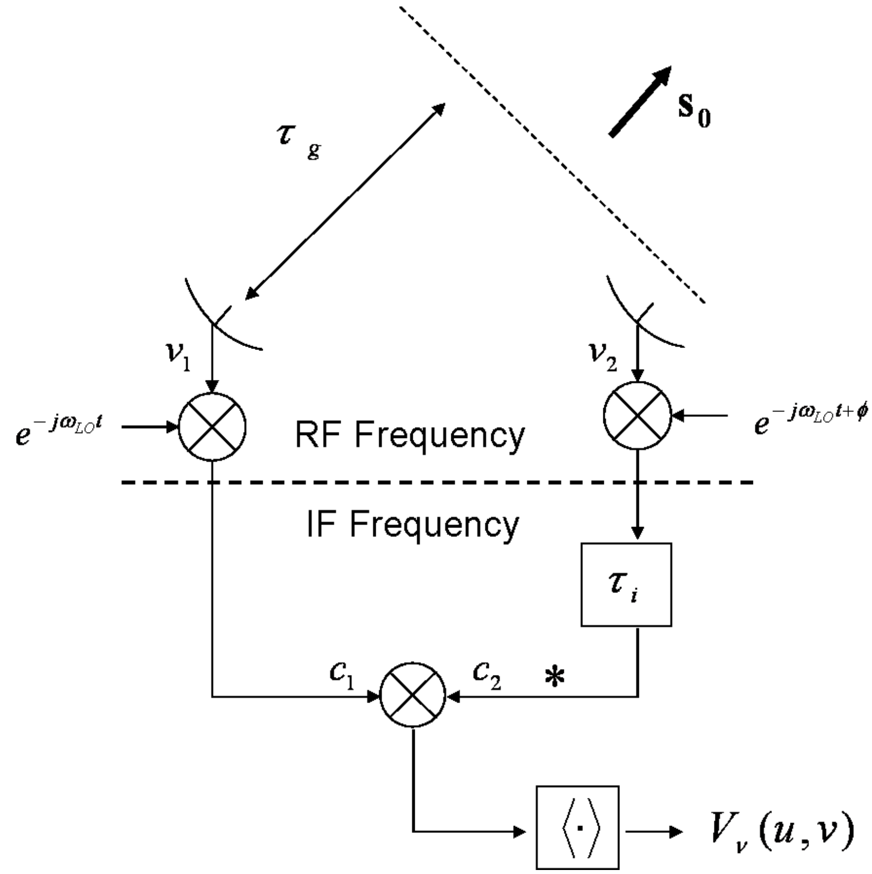

Figure 2.7: A block diagram of an interferometer with down conversion.

2.3 Computing u and v

As shown in Figure 2.6, $\mathbf{b_{proj}}$ is the projection of $\mathbf{b}$, the baseline vector, onto the uv plane, the plane perpendicular to $\mathbf{s_0}$. This is shown in greater detail in Figure 2.8. Astronomers orient the uv plane so that the v axis points toward the celestial north pole, the w axis points at the astronomical source, and the u axis points toward east [14] [16]. The components of $\mathbf{b_{proj}}$ along the u and v axis are the values of u and v for the measured visibility $V_\nu(u, v)$.

In what follows it is assumed that the reader has some familiarity with astronomical coordinate systems. A good introductory reference on positional astronomy is [17]. The convention found in astronomical literature has been used to specify locations on the celestial sphere. δ_0 is the symbol that is used for declination, which is

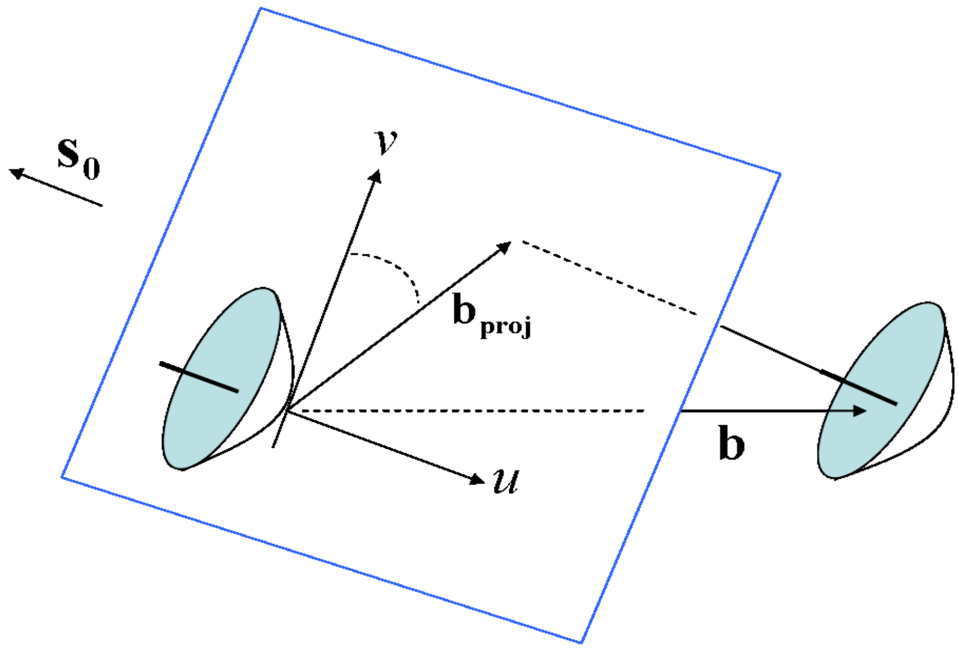

Figure 2.8: A baseline projected onto the uv plane.

specified in degrees, and H_0 is the symbol for hour angle, which is specified in hours (h).

In order to compute u and v for the measured visibility, we must know the locations of the antennas to a high degree of precession. One antenna in an image synthesis array is chosen as the zero or reference location, and the locations of the other antennas are given in meters East, North, and toward Zenith, with respect to the reference.

Although this is a convenient and intuitive system for specifying the location of the antennas, it is not the coordinate system used by astronomers in practice. The system astronomers use is aligned with earth's rotation axis, with the z axis pointed at the celestial north pole: $\delta_0 = 90°$, the x axis is pointed at $H_0 = -6h$, $\delta_0 = 0°$, and the y axis is pointed at $H_0 = 0h$, $\delta_0 = 0°$ [14] [16] [18].

To determine the locations of the antennas in this coordinate system we perform the following coordinate rotation:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 & -\sin\gamma & \cos\gamma \\ 1 & 0 & 0 \\ 0 & \cos\gamma & \sin\gamma \end{bmatrix} \begin{bmatrix} e \\ n \\ u \end{bmatrix}, \tag{2.25}$$

which is dependent on γ, the geographical latitude of the array [18]. In this new coordinate system, to find the baseline vector components for a pair of antennas we subtract their respective x, y, and z coordinates to arrive at

$$L_x = x_1 - x_2 \tag{2.26}$$

$$L_y = y_1 - y_2 \tag{2.27}$$

$$L_z = z_1 - z_2. \tag{2.28}$$

u, v, and w are found from L_x L_y and L_z using the following rotation [14]:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} \sin H_0 & \cos H_0 & 0 \\ -\sin\delta_0\cos H_0 & \sin\delta_0\sin H_0 & \cos\delta_0 \\ \cos\delta_0\cos H_0 & -\cos\delta_0\sin H_0 & \sin\delta_0 \end{bmatrix} \begin{bmatrix} L_x \\ L_y \\ L_z \end{bmatrix}. \tag{2.29}$$

It is easily seen from Equation 2.29 that u, v, and w are not constant for a given antenna configuration. As the earth rotates on its axis, the hour angle changes at a rate of 2π rad every 24 hours. This changes u and v as the antennas track the source. In general the declination, δ_0, in Equation 2.29 will be different for different sources which also leads to different values of u and v.

Given these considerations, the locations of the antennas are designed to fill out the uv plane as evenly as possible [14] [19]. To illustrate these principles, consider an array with antenna positions as shown in Figure 2.9. This array consists of four antennas spread out on an east-west line at a latitude of 40°. For simplicity, let the operating frequency be 300 MHz, i.e. λ is 1 meter. For a source that is directly overhead, that is hour angle $H_0 = 0h$ and declination $\delta_0 = 40°$, the uv plane is parallel to the plane of the antennas and is shown in Figure 2.10.

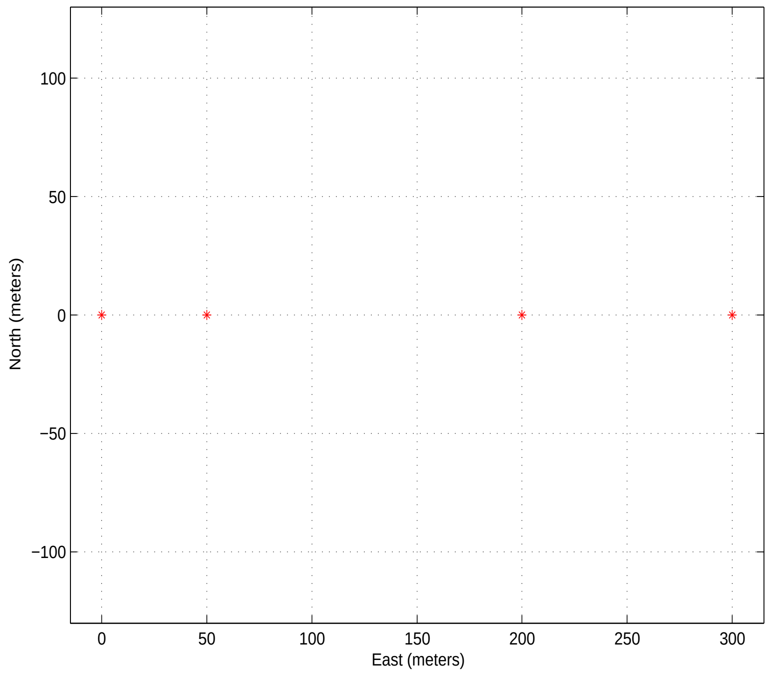

Figure 2.9: An east-west linear array.

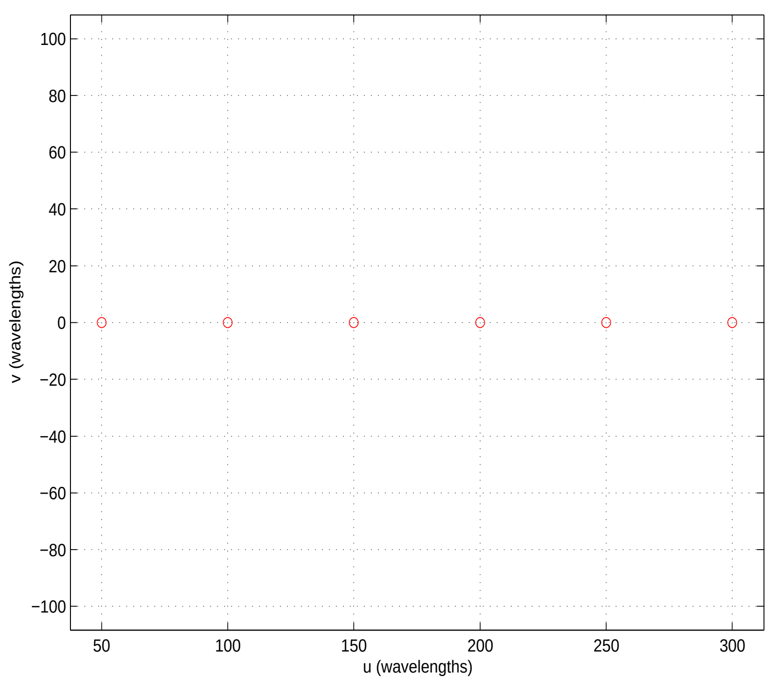

Figure 2.10: Baselines for the linear array in Figure 2.9.

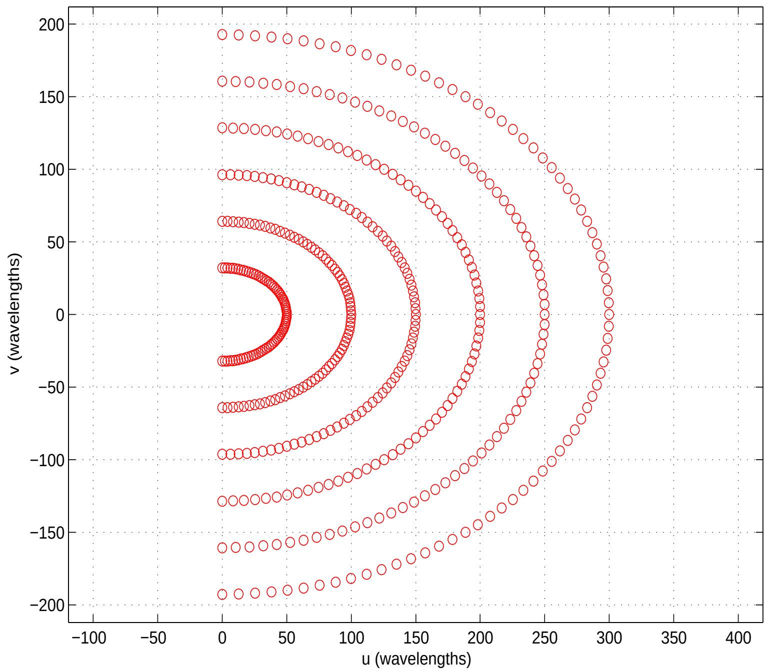

Figure 2.11: Linear array uv coverage for 40° declination.

Consider the same source, with declination $\delta_0 = 40°$, as it moves across the sky from $H_0 = -6h$ to $H_0 = 6h$. As the source moves so does the orientation of the uv plane and consequently the vector projection of each baseline onto the uv plane changes to give the uv coverage shown in Figure 2.11. Figure 2.12 also shows uv plane coverage for a 12 hour period, this time the source declination is 15° instead of 40°. One can see from the figures that decreasing the declination has the effect of compressing the uv coverage in the v direction. We shall see in the next section that the changes in uv coverage have ramifications for the image that is generated from the corresponding visibilities.

2.4 Fourier Transform Imaging

This section examines the process of generating an image from measured visibility data. To begin with, we will examine Equation 2.19 in some detail. It is

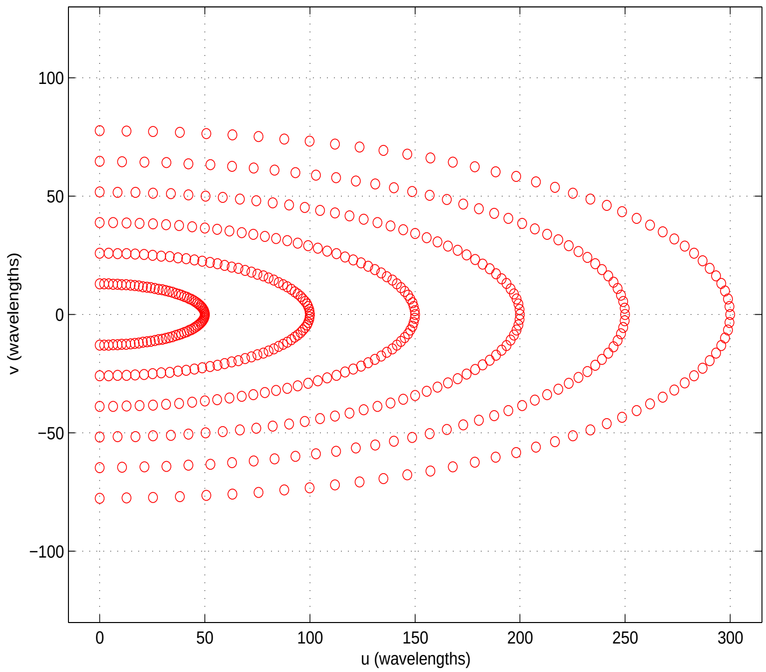

Figure 2.12: Linear array uv coverage for 15° declination.

repeated below, for convenience, as Equation 2.30:

$$V_\nu(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I_\nu(l, m) e^{-j2\pi(ul+vm)} dldm. \tag{2.30}$$

The Fourier inversion of Equation 2.30 is also repeated for convenience as Equation 2.31:

$$I_\nu(l, m) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} V_\nu(u, v) e^{j2\pi(ul+vm)} dudv. \tag{2.31}$$

2.4.1 Measuring Spatial Frequency

Equation 2.30 states that visibility is the Fourier transform of intensity. Therefore, when we measure visibility, we are measuring the spatial frequency content of the intensity distribution of the sky.

To illustrate the concept of spatial frequency consider an (unrealistic) intensity distribution:

$$I_\nu(l, m) = 2 \cos(2\pi(100l + 50m)).$$

This corresponds to waves in the sky with a period in the l direction of $1/100 = 0.01$ rad and a period in the m direction of $1/50 = 0.02$ rad. The visibility function for this intensity distribution is given by

$$V_\nu(u,v) = \delta(u-100, v-50) + \delta(u+100, v+50).$$

In other words, there is only frequency content for this intensity distribution at two points in the uv plane: $(u = 100, v = 50)$ and $(u = -100, v = -50)$. It is important to note that since intensity is a real function, visibility has complex conjugate symmetry. That is $V_\nu(u,v) = V_\nu^*(-u,-v)$.

2.4.2 Sampling the uv plane

The inversion in Equation 2.31 requires knowledge of u and v on a continuum extending from $-\infty$ to ∞. In practice, this is not possible for a number of reasons. First, measured visibilities must be estimated. The time required to estimate a visibility is on the order of 1 to 10 seconds during which u and v are changing. To assign a (u,v) point to a visibility, u and v are averaged over the time it took to estimate that visibility.

Secondly, to obtain a continuum of u and v would require moving the antennas of the array to fill out the spatial frequency plane. Antennas used in synthesis imaging weigh thousands of pounds, so moving the antennas is usually not feasible. Therefore, $V_\nu(u,v)$ is only know at a discrete number of points (as shown in Figures 2.11 and 2.12).

This discrete sampling of the uv plane can be understood in terms of Fourier theory. The treatment below follows that of [20] (see also [11]). The sampled visibilities $V_\nu^S(u,v)$ are equal to the continuous value of $V_\nu(u,v)$ multiplied by a sampling function composed of a spatial distribution of deltas:

$$V_\nu^S(u,v) = S(u,v)V_\nu(u,v), \tag{2.32}$$

$$S(u,v) = \sum_k \delta(u-u_k, v-v_k). \tag{2.33}$$

Fourier theory states that the multiplication of functions in the frequency domain corresponds to the convolution of those functions in the spatial domain. With this in mind, consider the case of a intensity distribution given by $I_\nu(l,m) = \delta(l-0, m-0)$, i.e. a single point source. Then $V_\nu(u,v) = 1$, and the image becomes

$$S(u,v) \times 1 \rightleftharpoons B(l,m) * \delta(l-0, m-0). \tag{2.34}$$

In astronomical literature, $B(l,m)$ is called the synthesized or dirty beam. It is the point-spread function of the synthesis array, and, as seen in Equation 2.34, $B(l,m)$ is the Fourier transform of the sampling function. If we think of the intensity distribution as a sum of point sources, i.e. $I(l,m) = \sum_a \delta(l-l_a, m-m_a)$, then the image is (roughly) a dirty beam at the location of each point source. Astronomers refer to the Fourier transform of the sampling function as the dirty image, or dirty map. In terms of Fourier transforms:

$$S(u,v) \times V_\nu(u,v) \rightleftharpoons B(l,m) * I_\nu(l,m) = I^D(l,m). \tag{2.35}$$

The dirty beam for the uv coverage of Figure 2.11 is shown in Figure 2.13.

At this point, it should be clear that the placement of the array elements relative to one another determines the shape of the synthesized beam and consequently the quality of overall image. The long baselines sample the uv plane at higher frequencies and dictate the resolution of the image. The short baselines measure the low frequency content of the image. Ideally, we would like to sample the uv plane uniformly at regularly spaced intervals. From the figures in this chapter, we can see that the uv plane is not uniformly sampled.

2.4.3 Inverting the Transform

To generate an image a Fourier transform similar to Equation 2.31 must be performed for the sampled data. There are two ways to accomplish this. First, the dirty image can be generated pixel by pixel. For each value of l and m we perform the following sum over all M visibilities [20]:

$$I_\nu^D(l,m) = \frac{1}{M}\sum_k V_\nu(u_k, v_k) e^{j2\pi(u_k l + v_k m)}. \tag{2.36}$$

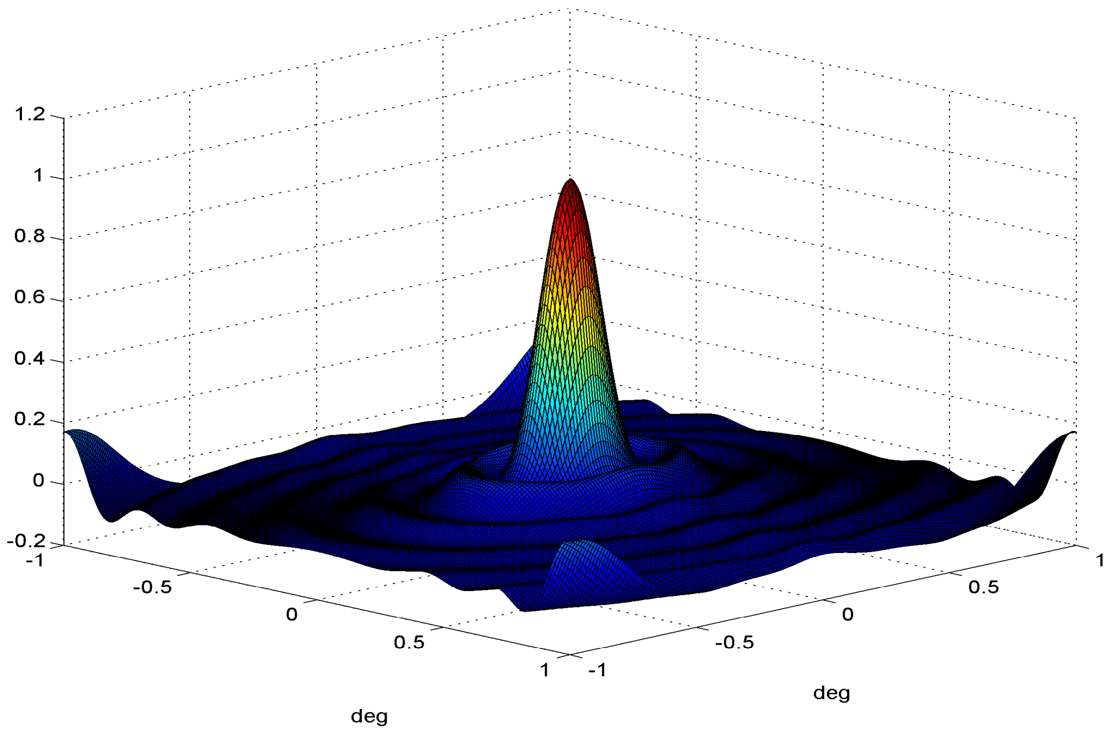

Figure 2.13: Dirty beam for the uv coverage in Figure 2.11.

This method is referred to as the direct Fourier transform (and should not be confused with the discrete Fourier transform). In practice, this method is rarely used because for large arrays and long observation times M is on the order of millions making the direct method a "computational pig".

The second method to perform the transform is by means of a Fast Fourier Transform (FFT). While the FFT is much faster than the direct method, FFT requires that the data lie on a rectangular grid. As shown in the figures of uv coverage, visibilities rarely lie on a rectangular grid, and some sort of interpolation or averaging is used to estimate the value of the visibilities on a rectangular grid. This process is referred to as gridding and the reader is referred to [20] for more detail.

The simplest form of gridding is to generate a rectangular grid of cells, and average the visibilities that lie within the cell. The average visibility over the cell is assigned the u, v coordinates at the center of the cell. This form of gridding is called cell averaging. Mathematically it is represented as

$$V_\nu(u_c, v_c) = \frac{1}{P} \sum_{k \subset C} V_\nu(u_k, v_k), \tag{2.37}$$

where C is the set of points that lie within the cell, and P is the number of points in C.

Chapter 3

Array Design

3.1 Introduction

After consulting with our collaborators at NRAO and studying image synthesis literature, we arrived at an overall system design for our image synthesis test platform. Figure 3.1 shows the building blocks of the image synthesis array: four satellite dishes, low noise amplifiers, an analog receiver, a combination digital downconverter (also referred to as a digital drop receiver) and digital signal processor, and a desktop personal computer. The remainder of the chapter is devoted to describing the design of the array we have called the BYU Very Small Array (VSA). Section 3.2 documents some of the early design decisions concerning low noise amplifiers and antennas. Section 3.3 describes the process of deciding where to place each antenna. Section 3.4 describes the analog receiver block and the digital signal processor block of the VSA.

3.2 Early Design Decisions

When the VSA was in the early planning stages, we decided that the antennas of the array would be steerable, three meter diameter satellite dishes. These dishes are low cost and our research group has had prior experience building them. Previous to the VSA, our group built a small antenna array consisting of three of the three meter dishes. This small array has been used to test RFI mitigation algorithms. For more detail the reader is referred to the Master's theses of Brett Walkenhorst [21], Andrew Poulsen [8], and Chad Hansen [22].

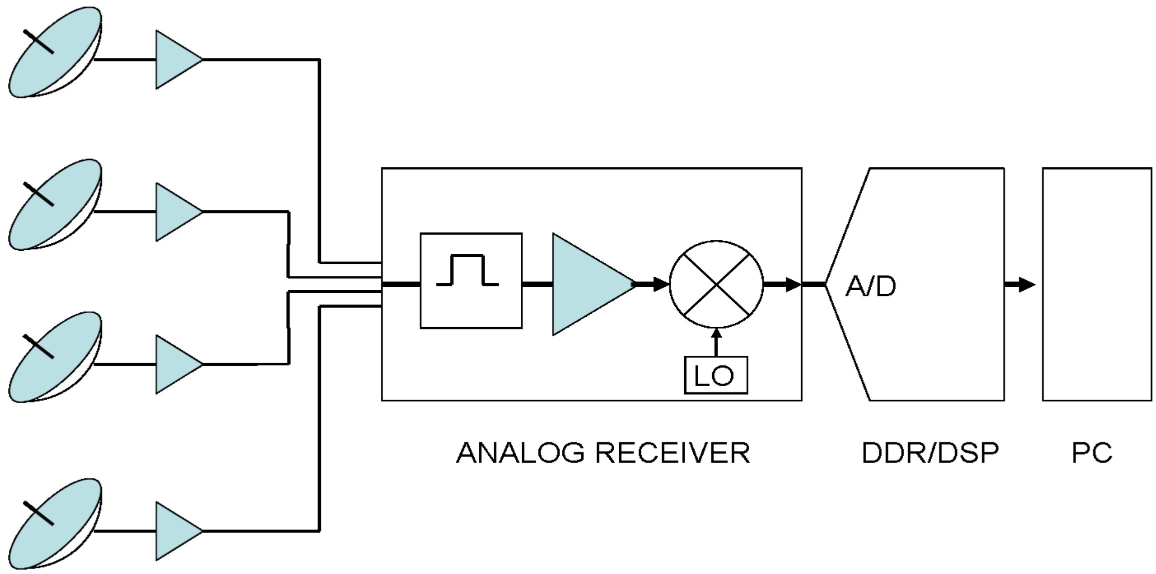

Figure 3.1: Block diagram test platform concept.

The low noise amplifiers for the array were chosen based on the advice of our collaborators at NRAO. The LNAs for the VSA are made by Mini-Circuits, part number ZEL-1217LN. These are tuned for operation in L-band, have approximately 25 dB of gain, and have a noise figure of 1.5 dB, according to the Mini-Circuits specifications.

3.3 Antenna Placement

3.3.1 Placement Logistics

A decision was made to place the VSA antennas on the roof of the Clyde Building (CB) and the neighboring buildings at Brigham Young University (BYU). The CB houses the department of Electrical and Computer Engineering and the research labs for our group. Logistically, having the antennas on the rooftops made things simpler in terms of building the antennas, running signal cables, and having more of the sky visible to the antennas.

There were drawbacks associated with this decision. The size of the roof limited the size of the baselines. Also, in terms of RFI, the CB roof is a “noisy”

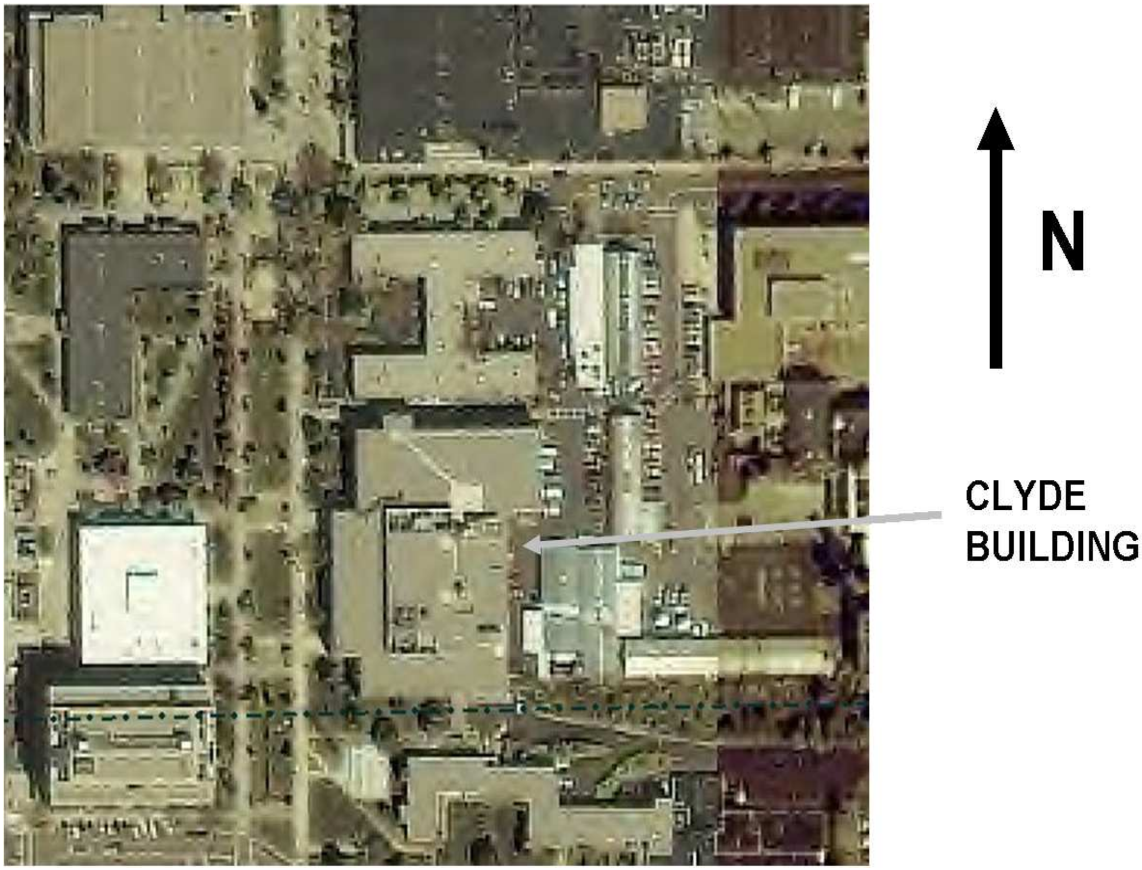

Figure 3.2: Satellite photo of the Clyde Building [1].

location. However, for RFI studies and educational purposes, these issues were not serious problems.

Figure 3.2 shows a satellite image of the CB and the surrounding buildings on the campus of Brigham Young University. The CB is sandwiched by the Fletcher Building (an 'H' shaped building) to the North and the Knight Mangum Building to the South. Initially, we planned to place an antenna on the roof of the Fletcher Building to create longer baselines and increase the resolution of the array. University red-tape ended that plan, and in the end, we settled for placing all of the antennas on the roof of the CB.

The CB is a five floor building with four floors above ground. The diagram in Figure 3.3 shows the building footprint of the CB, i.e. the area that was available for

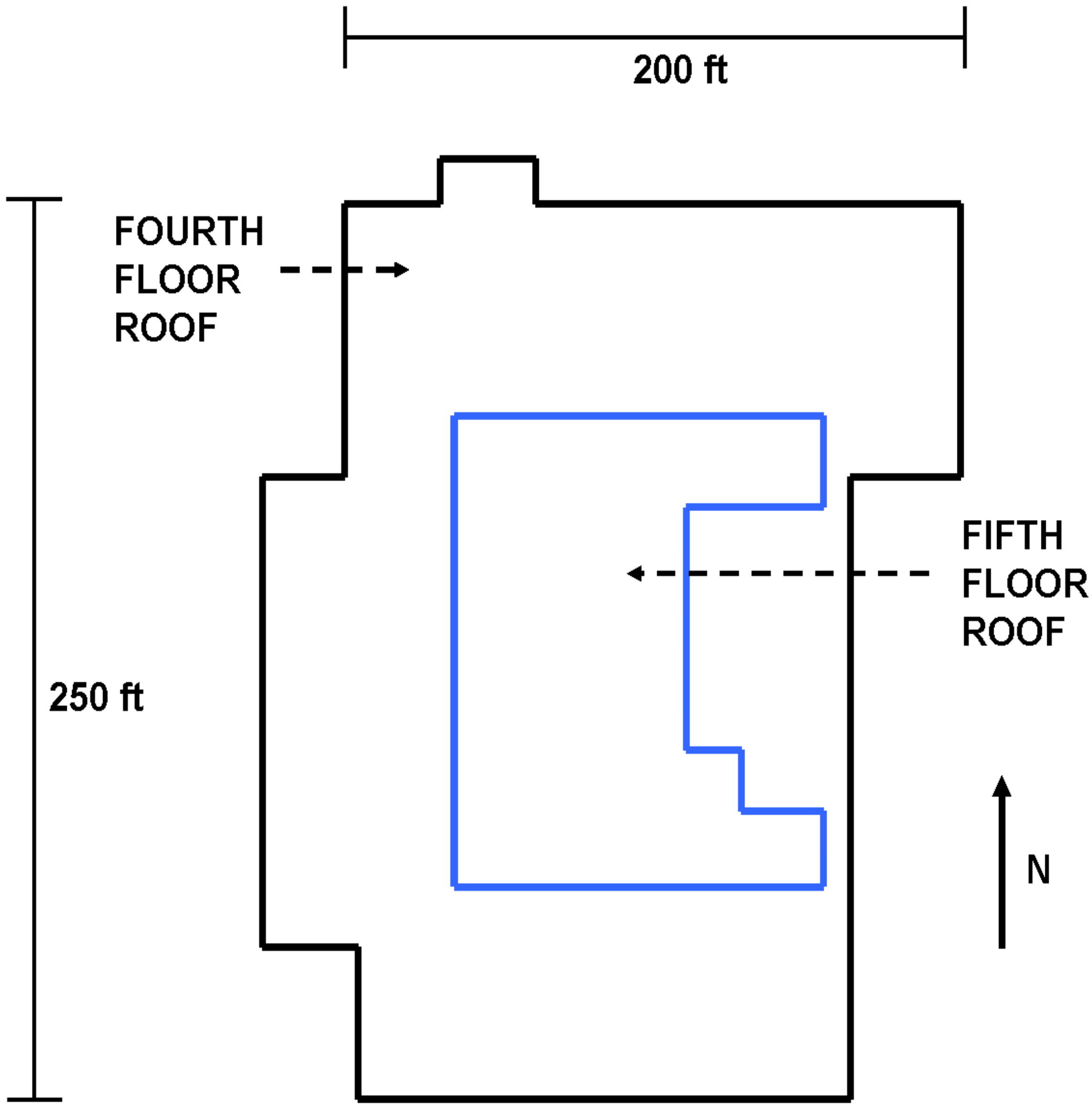

Figure 3.3: Clyde Building roof layout [2].

antenna placement. [1] The figure also shows that the roof of the CB has two levels, with part of the roof covering the fourth floor and part covering the fifth floor.

After some consideration, we decided to place all four antennas on the fourth floor roof to keep them coplanar. The guiding principle for this decision was simplicity. Having all of the antennas in a plane made modelling the array simpler. Once we

[1] The aerial view diagrams of the CB were created from [2].

had decided to locate the antennas on the fourth floor roof, we had to determine the final location of each antenna.

3.3.2 Final Placement Concerns

There were a number of concerns that guided the final placement of the antennas. Our primary concern was safety. Placing the antennas at the edge of the building would give the longest possible baselines. However, the edge of the fourth floor roof presents a fifty foot precipice to the unwary and absent-minded graduate student. For safety's sake, we decided that the base of each antenna needed to be at least six feet away from the edge. We also stipulated that anyone working on the antennas was required to wear a safety harness.

Another concern that guided the final placement of the antennas was how much sky would be visible to each antenna. Figure 3.4 is a photograph from the fourth floor roof facing east. It was taken standing approximately 35 ft away from the fifth floor. The photograph shows the the Wasatch mountains peeking out over the top of the fifth floor roof. Figure 3.5 shows a similar view except now the vantage point is approximately 50 ft away from the fifth floor. From the photographs, one can surmise that the closer an antenna is placed to the fifth floor, the less sky is visible to that antenna.

With these concerns in mind, the area available for final antenna placement was chosen and is shown in Figure 3.6. Notice that the part of the fourth floor roof east of the fifth floor, inside the "horse shoe", was not included in the available area. This is because the space is occupied by one of the building's air cooling systems.

3.3.3 U V Plane Sampling

With the available antenna placement area chosen, the final positions of the antennas were determined based on how well any particular arrangement sampled the uv plane (see Section 2.3, [14], and [19]). As with any interferometry array, the antennas were placed so that they would sample the uv plane as uniformly and completely as possible.

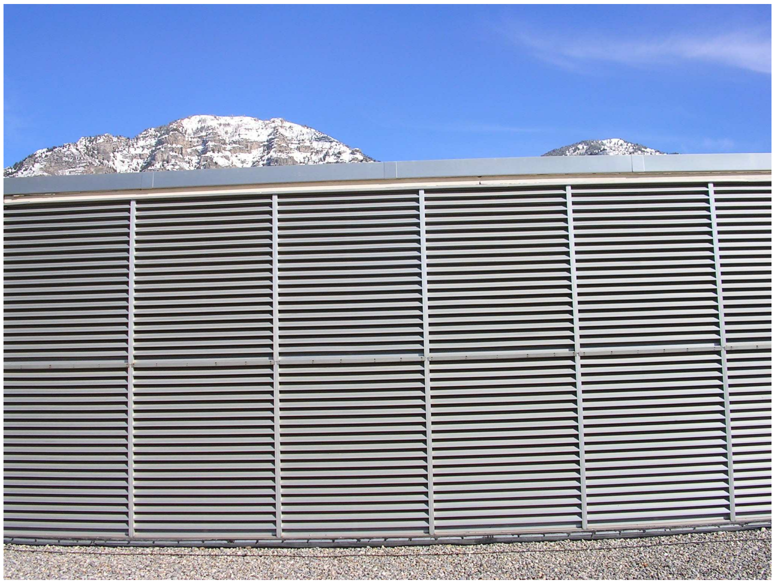

Figure 3.4: A view of the eastern sky at 35 ft away from the fifth floor.

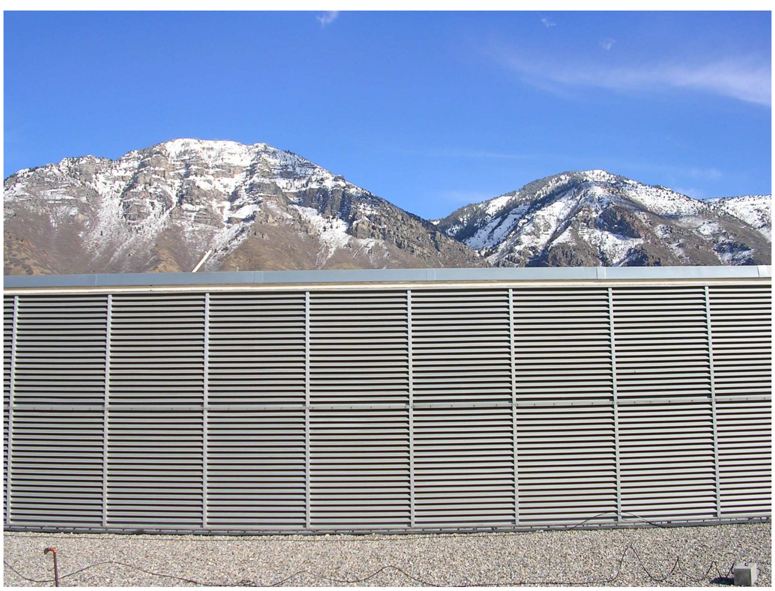

Figure 3.5: A view of the eastern sky at 50 ft away from the fifth floor.

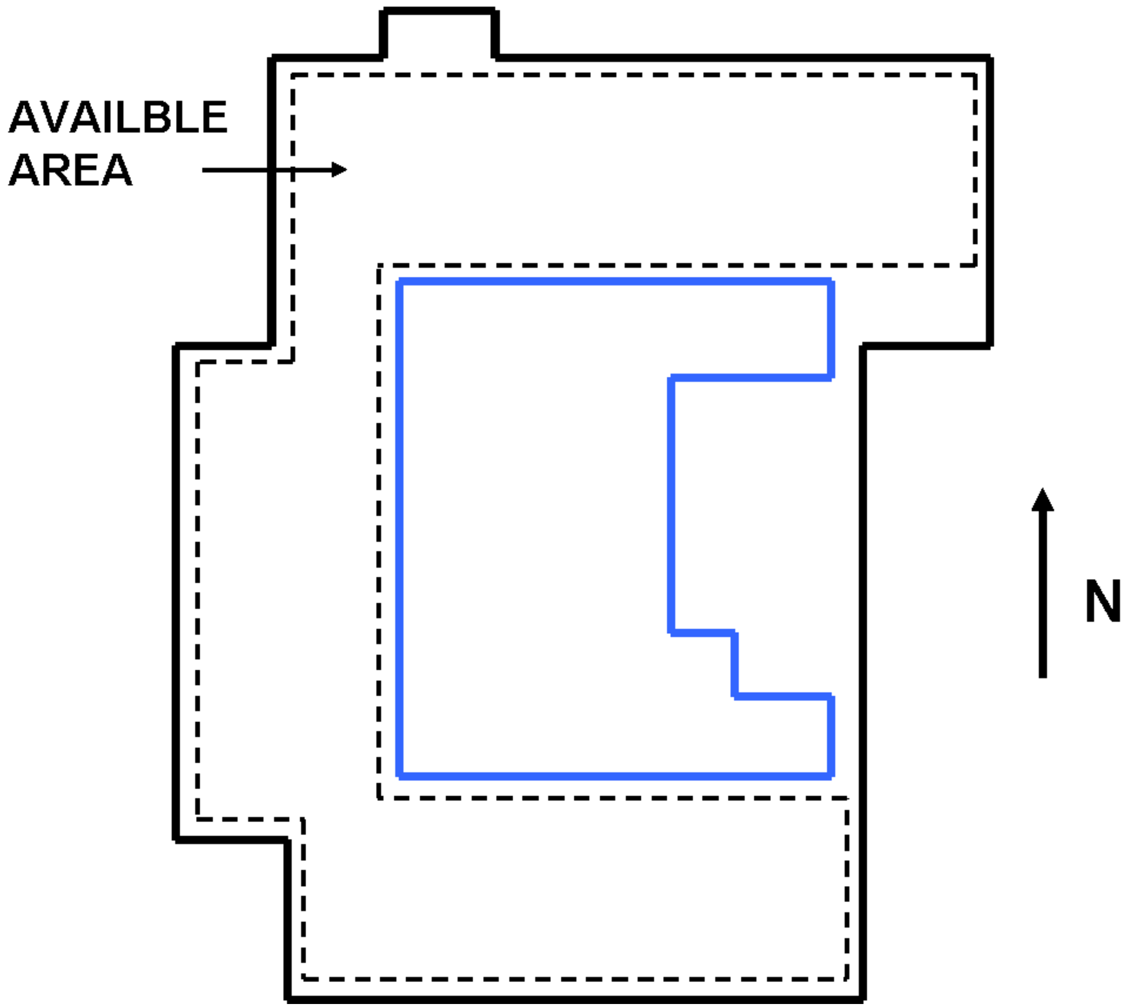

Figure 3.6: Available area for antenna placement.

To obtain maximum resolution from the array, two antennas were located as far from each other as possible: one antenna at the north east corner and another at the south west corner. Another antenna was placed at the north west corner of the building. The placement of the last antenna was a decision between the two general configurations shown in Figure 3.7 (called option 1 and option 2). Once that choice was made, the location of the final antenna was fine tuned based on uv coverage.

To chose between option 1 and option 2, we looked at the uv coverage for each. Using Equation 2.25 and Equation 2.29, uv coverage was computed for the two options at different declinations of astronomical sources.

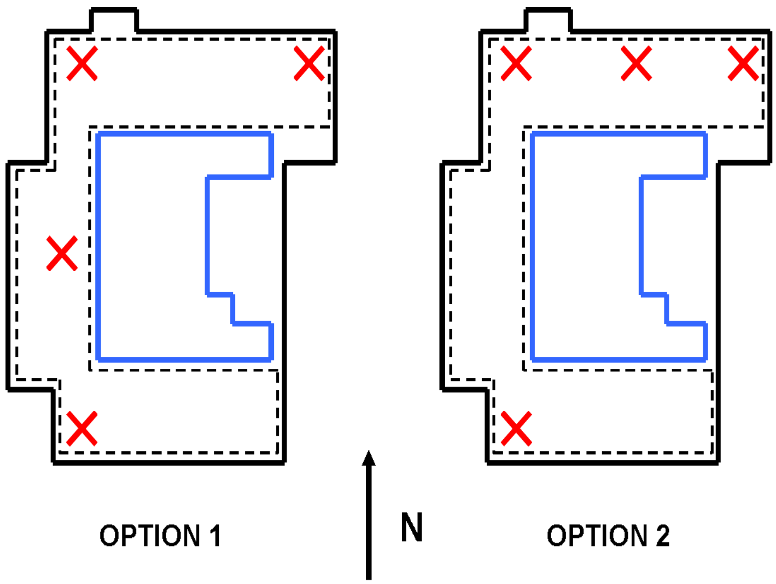

Figure 3.7: Final configuration choices.

Figures 3.8, 3.9, and 3.10 show uv coverage for option 1. Figures 3.11, 3.12, and 3.13 show uv coverage for option 2. The center frequency for each coverage was set to 1.6 GHz. Each coverage shows 12 hours worth of tracking.

The coverage for option 1 and option 2 is similar at high declinations, i.e. the coverage at declination 60 degrees is similar. The coverage differs for low declinations. At declination 20 degrees, option one has even coverage of the uv plane, whereas option 2 has large gaps in the uv coverage in the v direction (see Figure 3.11).

These gaps in coverage are undesirable because they lead to large side lobes extremely close to the main beam of the dirty beam. This effect can be seen by comparing the dirty beams as shown in Figures 3.14 and 3.15. Note the large side lobes in the dirty beam of option 2 are not present in the dirty beam of option 1.

For these reasons, option 1 was chosen as the final configuration. Some fine tuning of the last antenna position led to the final placement for the antennas in the VSA, which is shown in Figure 3.16 (see Table 5.1 for the measured position of each antenna).

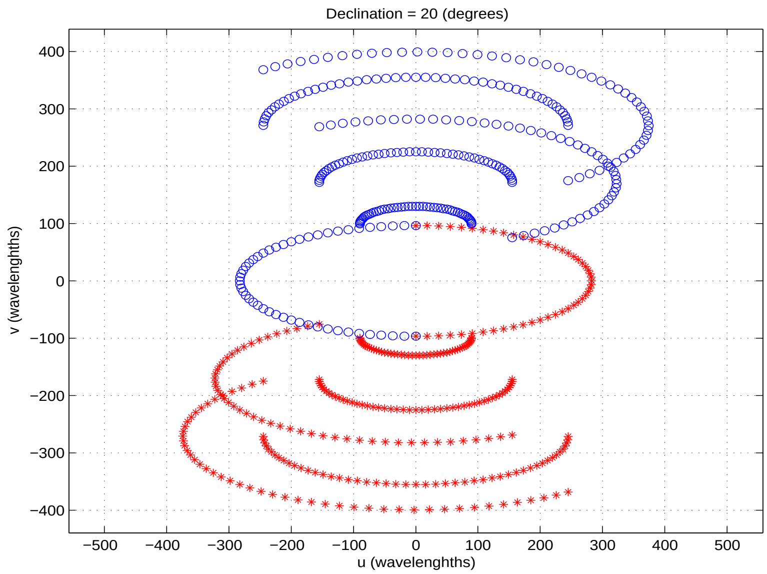

Figure 3.8: uv coverage for option 1 at declination 20 degrees.

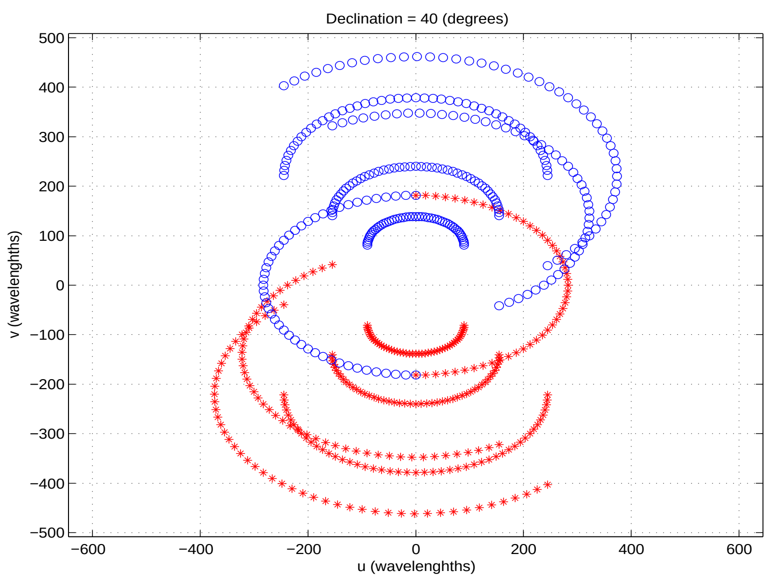

Figure 3.9: uv coverage for option 1 at declination 40 degrees.

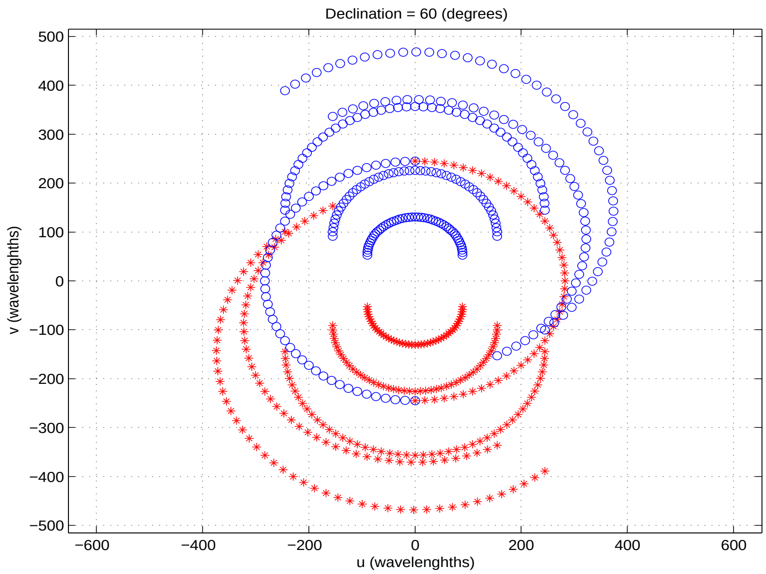

Figure 3.10: uv coverage for option 1 at declination 60 degrees.

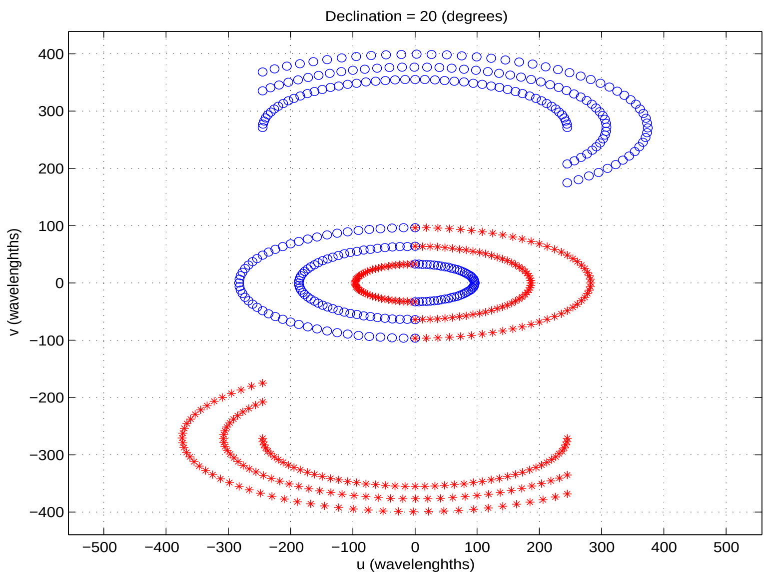

Figure 3.11: uv coverage for option 2 at declination 20 degrees.

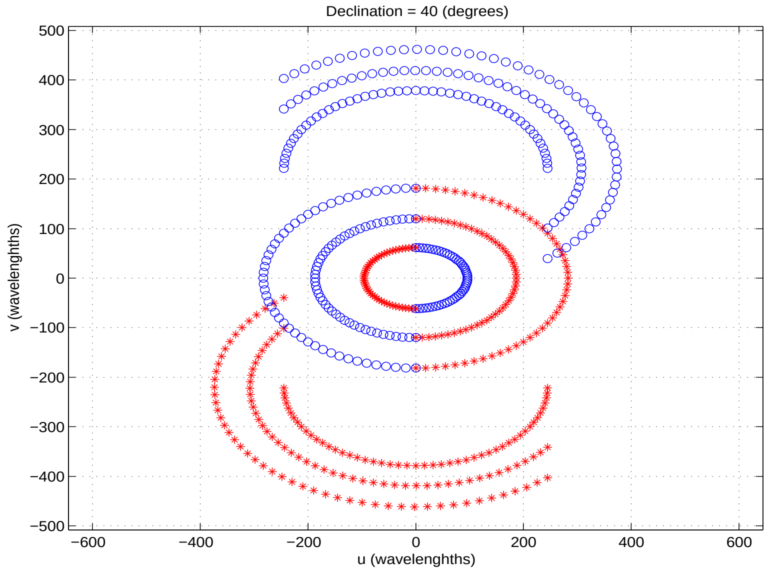

Figure 3.12: uv coverage for option 2 at declination 40 degrees.

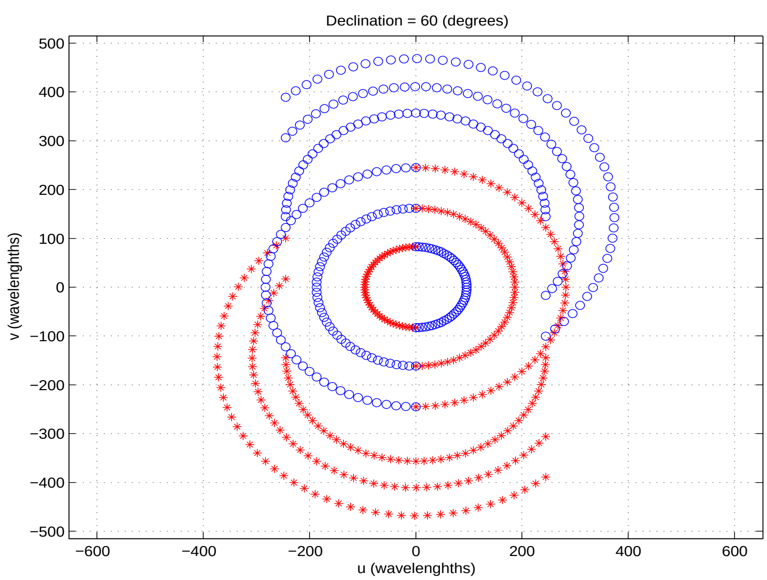

Figure 3.13: uv coverage for option 2 at declination 60 degrees.

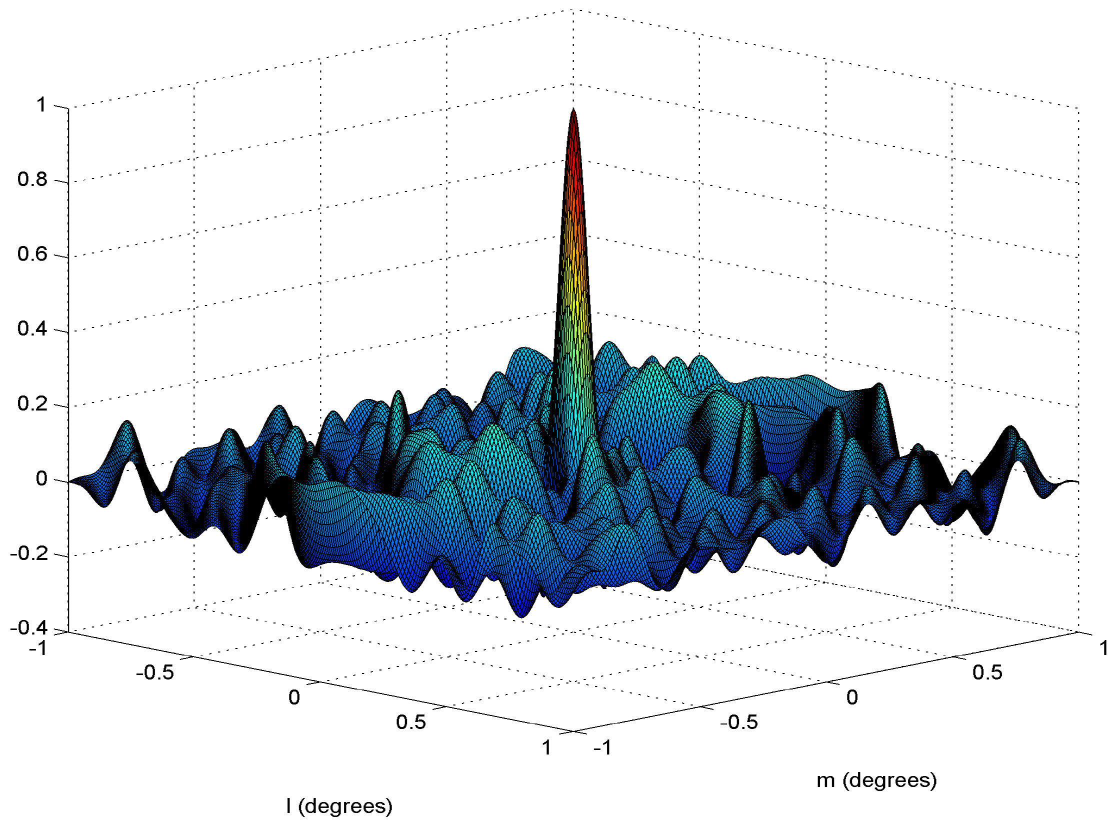

Figure 3.14: Dirty beam for the coverage in Figure 3.8 (option 1).

3.4 Receiving and Processing

3.4.1 Analog Receiver

The analog receiver for the VSA image synthesis array was built during prior research conducted by our group. The receiver's design, construction, and specifications are described in detail in [21] and [22]. It was designed with a number of future uses in mind, including a four element synthesis array.

The receiver has four channels and it can be tuned to any frequency between 1.2 GHz and 1.8 GHz. It has a final IF located between 8 and 24 MHz. The IF output is sampled and processed by the signal processing subsystem described below.

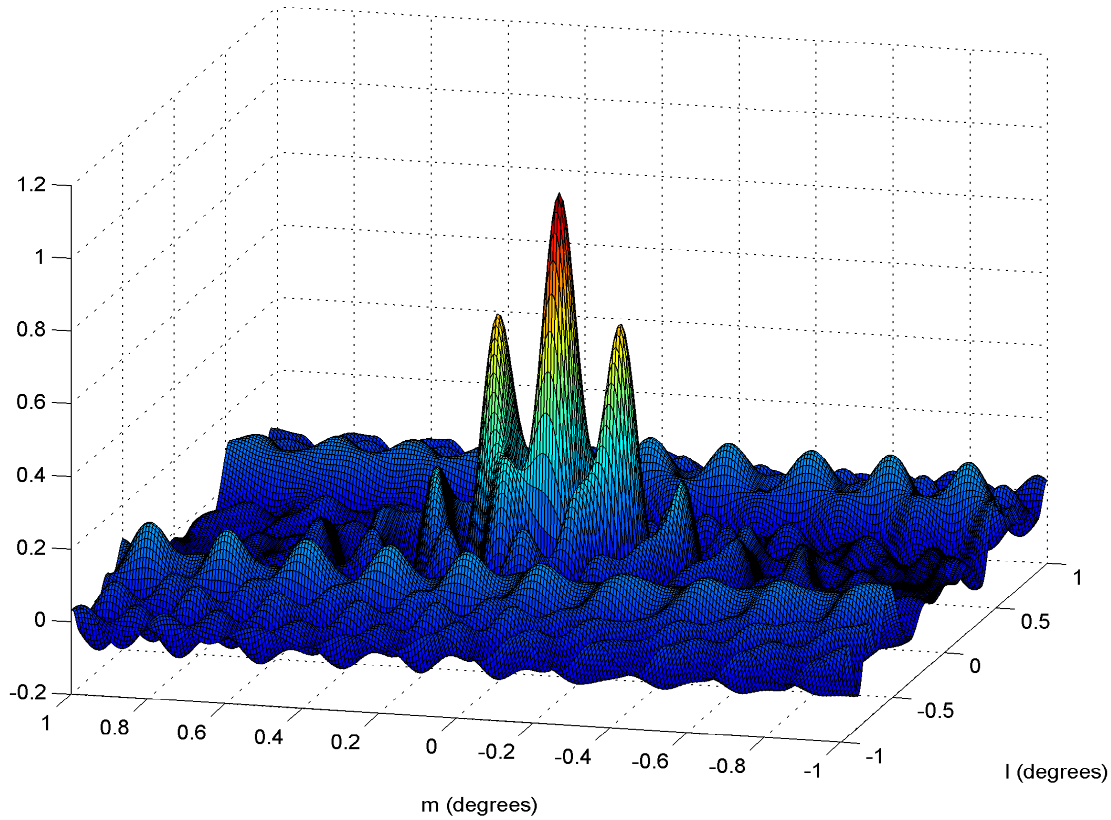

Figure 3.15: Dirty beam for the coverage in Figure 3.11 (option 2).

3.4.2 Signal Processing

The signal processing resources of the VSA consist of two of Pentek's P6216 DDRs and one P4291 quad-microprocessor DSP platform. The P6216s and the P4291 are referred to in this thesis as the DSP. The DSP is described in detail in [8].

The DSP platform is based on Texas Instruments (TI) TMSC3206701 32-bit floating point microprocessors. The software written by members of the research project has allowed us to program the DSP as an LMS adaptive filter, a spectrum analyzer, and a beam former (see [8]).

The software created for the VSA imaging array configures the DSP as a four channel correlator, and is described in detail in Appendix B. The correlator generates 4x4 correlation matrices for bandwidths of 1 or 2 MHz. The correlation integration

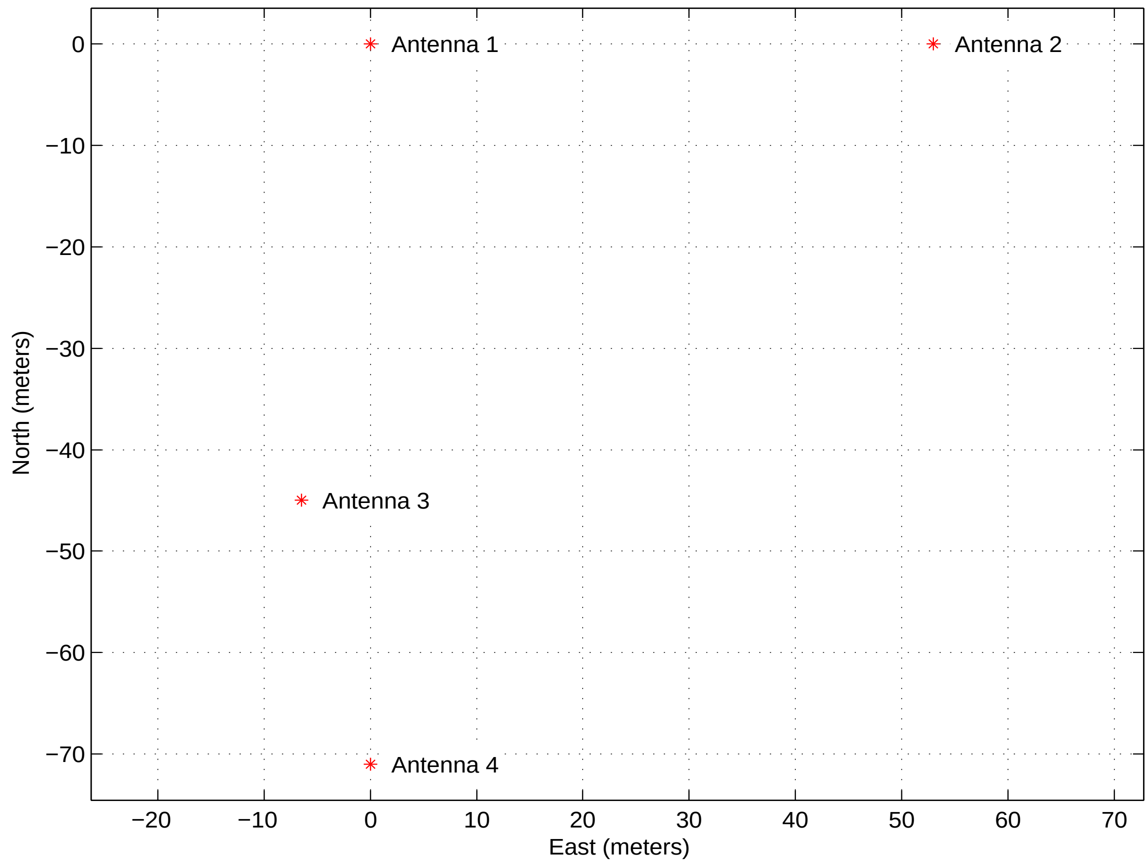

Figure 3.16: The final placement of antennas for the VSA.

time is programmable and can be as small as 2 milliseconds and as large as 60 seconds. The data from the DDR is complex base-banded and so the correlation matrices are also complex.

The correlation matrices are transferred to a desktop personal computer (referred to as the host PC) for storage and processing. The host PC is responsible for computing u and v for the measured visibilities and for performing the Fourier inversion to generate the image.

Chapter 4

Simulation and Analysis

4.1 Introduction

Before building the VSA, a number of simulations were carried out to analyze the array's performance. We wanted to know the shape of our synthesized beam, what kind of images we could expect, and the resolution of the array, among other things. We were also concerned that we would not be able to accurately measure the phase component of the spatial coherence between our antennas. A series of simulations were performed in order to analyze how much error could be tolerated in the measurement of the phase between antenna pairs. This chapter describes these simulations and the results.

This chapter is structured as follows. Section 4.2 and 4.3 document the initial image simulations. Section 4.4 describes the simulations used to quantify the amount of tolerable error in measuring the phase of the visibilities.

4.2 The Noise Free Model: Dirty Beam and Resolution

This section describes the noise free simulations performed to analyze the response of the Very Small Array (VSA). Our collaborators at NRAO suggested that after the VSA was built, we test it by imaging Cassiopeia A. Cassiopeia A is the brightest radio source in the sky, excluding the sun. Cassiopeia A is a supernova remnant located at a right ascension of 23 hours and a declination of 58 degrees. In the simulations described below the simulated astronomical source (or sources) was placed at these coordinates.

The mathematical model is as follows. The electric field at a given antenna, E_i, is the sum over the sources in the sky:

$$E_i = \sum_{b=1}^{q} \alpha_b e^{j\mathbf{k}_b \cdot \mathbf{r}_i}, \tag{4.1}$$

where α_b is the amplitude of source b, $\mathbf{k}_b$ is the k vector of the source b, and $\mathbf{r}_i$ is the vector location of antenna i.

This model is especially attractive because of its simplicity. It is important to note the simplifications in Equation 4.1. The sky being modelled is empty except for the q sources. Note also that Equation 4.1 does not model any directional antenna response. The antennas are isotropic, point receivers, and no attempt has been made to model polarization effects. The total antenna response is a scalar quantity.

Assuming there are p antennas in the array, the response from each antenna may be conveniently stacked into a vector:

$$\mathbf{x} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_p \end{bmatrix}. \tag{4.2}$$

If the model included noise, the correlation matrix, R, would be estimated as $R = \left\langle \mathbf{x}\mathbf{x}^H \right\rangle$. Since this model is noise-free R is given as $R = \mathbf{x}\mathbf{x}^H$. This model is useful for examining dirty beam shapes and quantifying the resolution of the array.

Figure 4.1 shows the uv coverage for a source at the right ascension and declination for Cassiopeia A. The coverage spans 16 hours of source tracking, with correlations generated at 100 equal intervals during that time. The observing frequency is 1.6 GHz. Figure 4.2 shows the dirty beam for the uv coverage in Figure 4.1 as obtained by the direct method Fourier transform of Equation 2.36 with all of the visibilities equal to one, i.e. it is the Fourier transform of the sampling function. The dirty beam in the following figures has been normalized to have unit amplitude at the peak.

Figure 4.3 shows the beam profile along the l and m axes. From these figures, it is easy to see why the synthesized beam is called the dirty beam: sidelobes are high

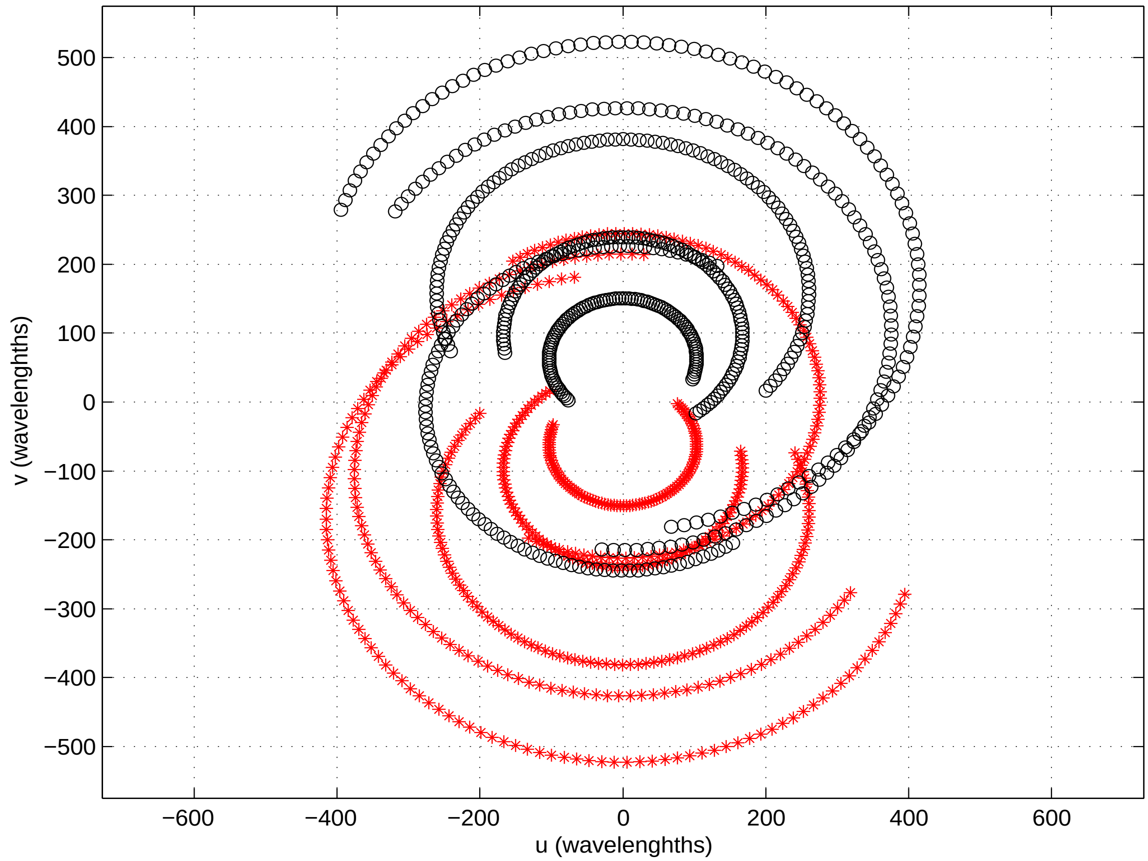

Figure 4.1: uv coverage for 16 hours.

compared to the height of the main beam. One might also hope that the synthesized beam has radial symmetry. Such is not the case, the main beam is narrower in the m direction than in the l direction.

In practice, radio astronomers will "weight" the measured visibilities to shape the synthesized beam and lower its sidelobes [20]. They will also process a synthesized image using a process called deconvolution in a further effort to overcome the undesirable effects of the dirty beam [23].

Examining the dirty beam, in particular the main lobe of the dirty beam, also provides an estimate of the resolution of the array. Here, resolution means how close two point sources can be before they appear as a single source. Equivalently, resolution can be thought of as the width of a point source in the final image. Anything

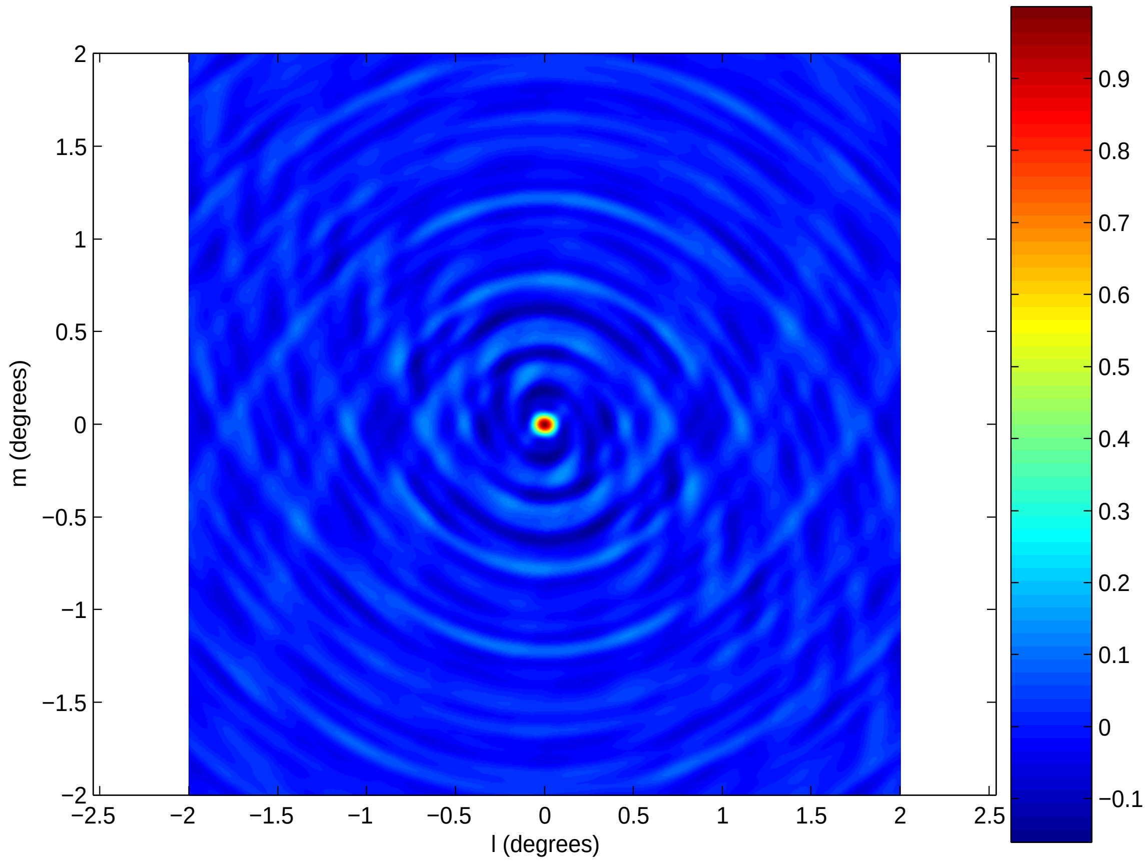

Figure 4.2: Dirty beam for uv coverage in Figure 4.1

in the sky with angular extent greater than a point source will be resolved by the array.

The main lobe of the dirty beam is approximately 0.1°, or 6 arcmin, wide at the half amplitude point. Doubling this figure gives an estimate of the array's resolution: 12 arcmin.

Another way to estimate the resolution of the array, is to simulate two sources of the same amplitude. Figure 4.4 shows the m axis for two sources separated by 0.16° (9.6 arcmin) and Figure 4.5 shows the m axis for two sources separated by 0.12° (7.2 arcmin). The array still resolves the sources separated by 9.6 arcmin. The figures show that 12 arcmin is a reasonable, albeit conservative, estimate of the resolution for this source declination and observing frequency.

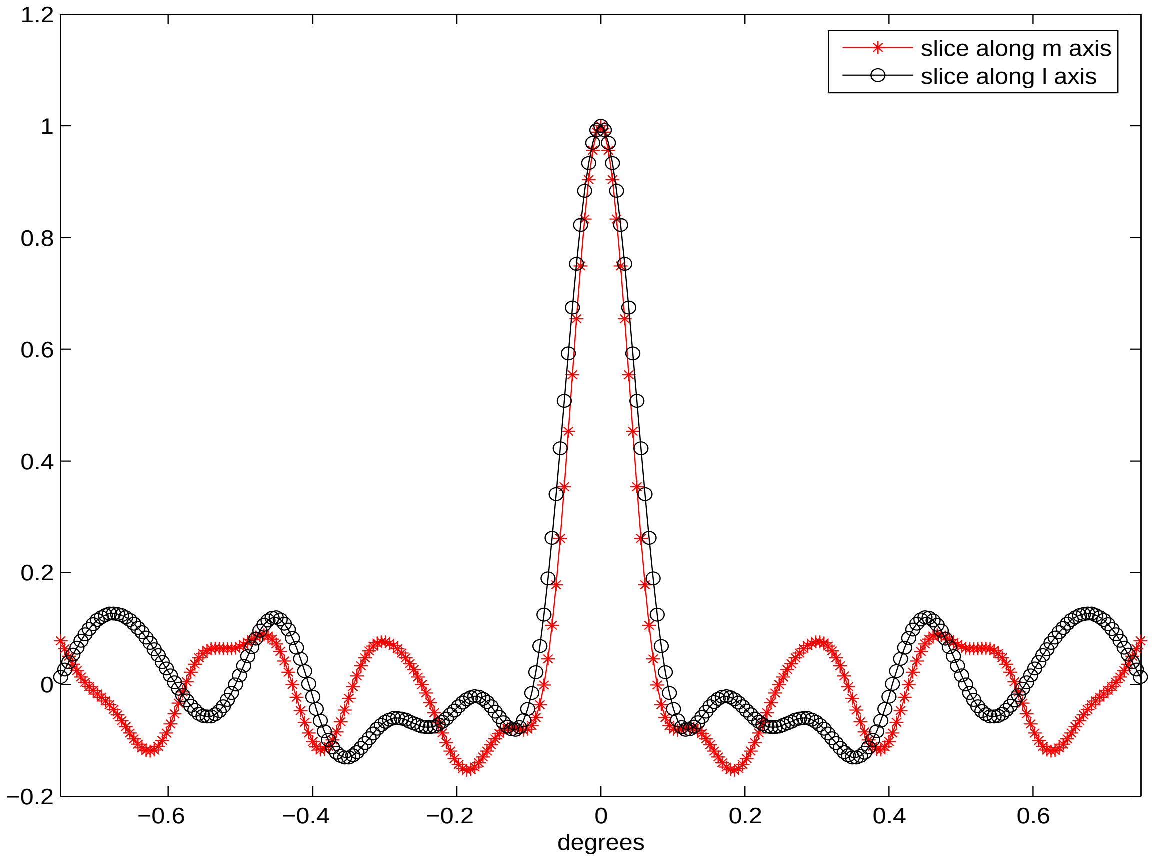

Figure 4.3: Slice through the dirty beam along the l and m axes.

As a final comment, the resolution of the array is set by the longest baseline. If b_{max} is the longest antenna separation (in meters) and λ is the wavelength for the observing frequency, then $\Delta\theta_p$ (the width of a point source in rads) can be approximated as:

$$\Delta\theta_p = \frac{\lambda}{b_{max}}.$$

To see where this comes from consider the two peaks in Figure 4.4 to be adjacent peaks of a sinusoid. The smallest angular separation between the peaks corresponds to the smallest period that is supported by the uv coverage. This period is given by λ/b_{max}. For the VSA, b_{max} is 98.6 meters. At an observing frequency of 1.6 GHz, this gives $\Delta\theta_p$ equal to 0.1090° or 6.5 arcmin.

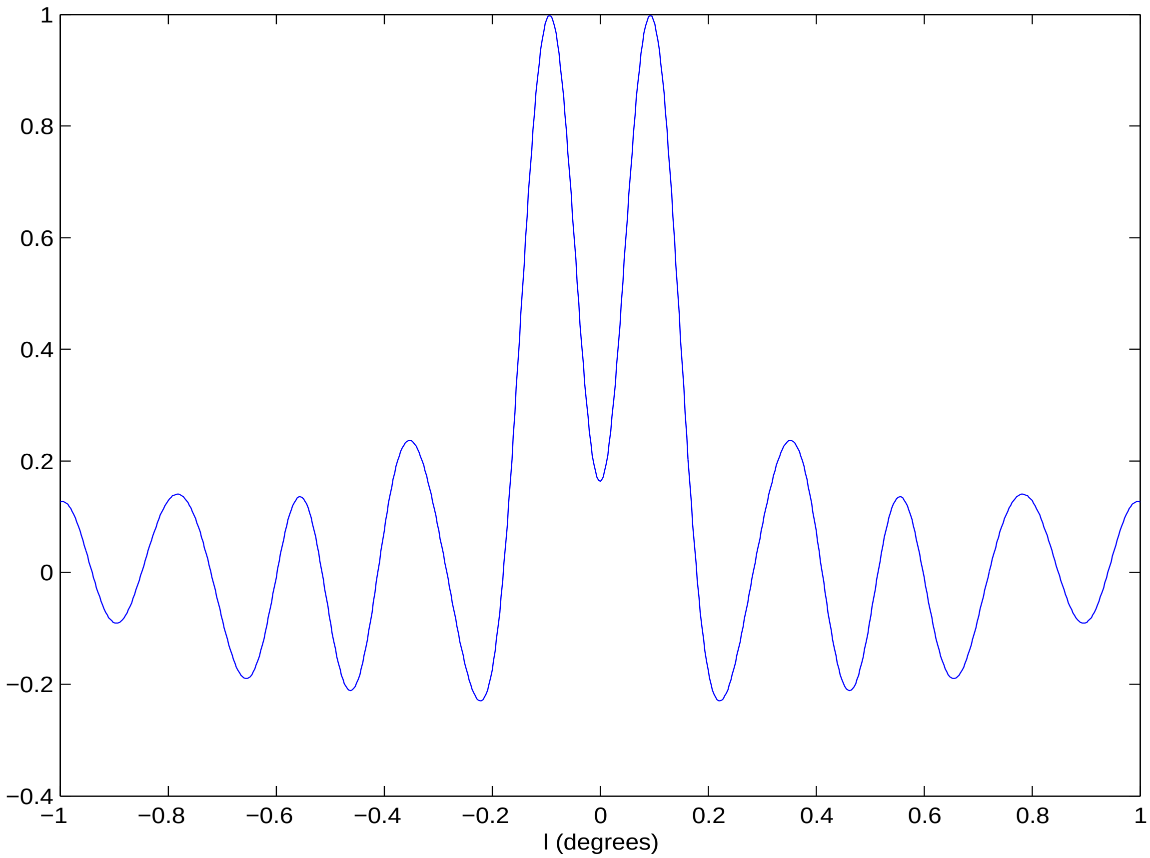

Figure 4.4: Two sources: slice through the m axis, separation 0.16 degrees.

4.3 Model with Noise

The images simulated without noise are really expected values of images. To produce more realistic simulations, noise is added to the model given by Equation 4.1. Before presenting the mathematical model for this section, a brief review is presented of noise in radio astronomy observations. Noise and its effects in radio astronomy synthesis imaging are covered by Walker and Wrobel in [24].

4.3.1 Noise in Radio Astronomy

Astronomers describe system noise power, and power from astronomical sources, in terms of an equivalent noise temperature using the Rayleigh-Jeans approximation (for more detail see Pozar [25]). The (noise) power for a source with an equivalent

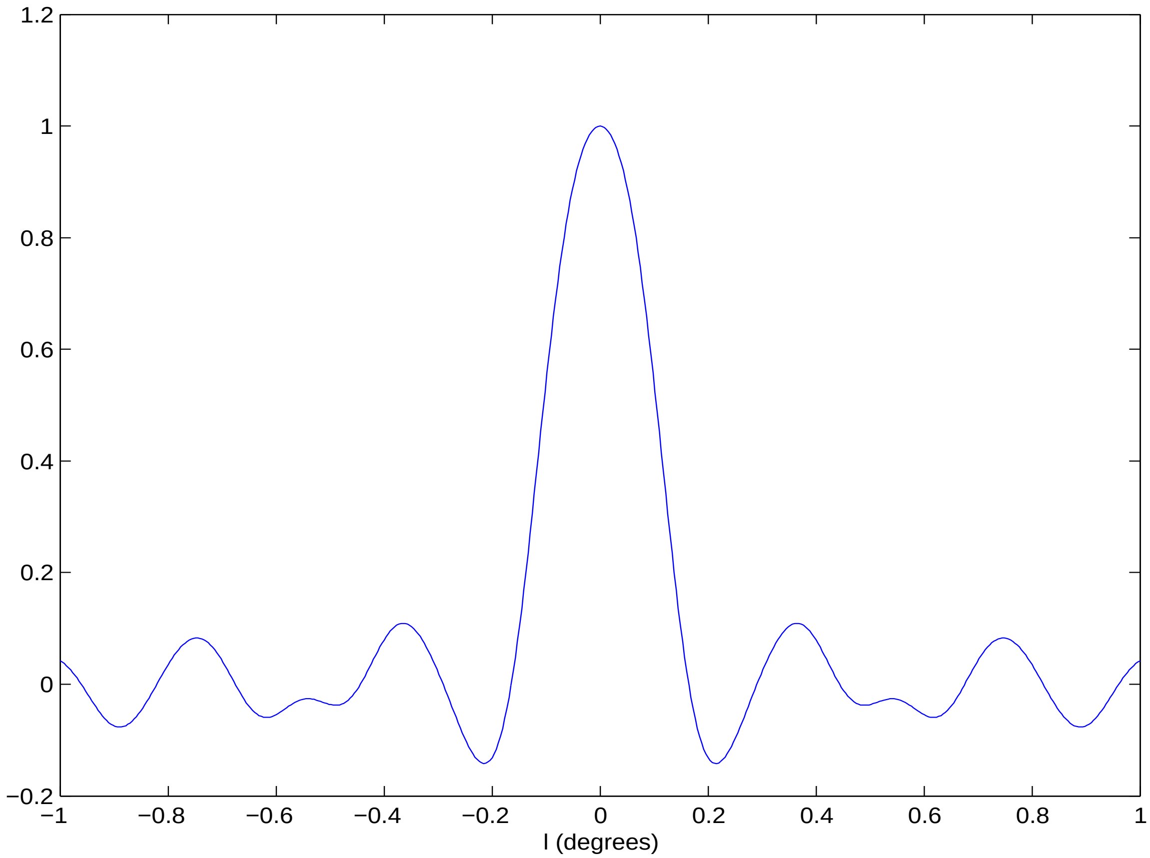

Figure 4.5: Two sources: slice through the m axis, separation 0.12 degrees.

temperature T in Kelvin is given by

$$P = k_B T B, \tag{4.3}$$

where k_B is Boltzmann's constant and B is the observing bandwidth in Hz [24]. Given the noise temperature of an antenna, one can find the noise power from Equation 4.3.

The power of an astronomical source is usually given in units of flux density. The power flux per unit bandwidth from an astronomical source is measured in Janskys, 1 Jy = 10^{-26}W/m^2/Hz. For example, Cassiopeia A has a flux density of 1500 Jy at 1.4 GHz. The power from an astronomical source at the feed of an antenna is given by

$$P_a = \frac{1}{2} S \eta_A A B, \tag{4.4}$$

where S is the flux density in Janskys, η_A is the antenna efficiency, A is the collecting area of the antenna in meters squared, and, once again, B is the observing bandwidth in Hz. The factor of 1/2 is due to the fact that one antenna only receives half the power from an unpolarized source. The power of the source can be expressed in terms of an equivalent temperature, T_a, by solving Equation 4.3 for T and inserting Equation 4.4 to get [24]

$$T_a = \frac{S\eta_A A}{2k_B}. \tag{4.5}$$

As an example, consider the noise characteristics of the VSA. The dominant noise source is the thermal noise of the LNAs. (This may not be true for arrays with liquid helium cooled LNAs like those used in most synthesis arrays.) The VSA has a noise figure, set by the LNAs, of 1.5 dB. Noise figure, F is related to equivalent system noise temperature by $T_{sys} = T_0(F - 1)$, where T_0 is room temperature: 290 Kelvin [25]. The system noise temperature for the VSA is therefore: $290(10^{0.15} - 1) = 120$ Kelvin. The antennas for the VSA are 3-meter (diameter) dishes with an antenna efficiency of 50%. This is a typical efficiency figure for these types of dishes (see [26]). Using Equation 4.5, Cassiopeia A will produce 1.92 Kelvin of temperature in the antennas of the VSA. This corresponds to about -18 dB SNR.

Radio astronomers deal with noise by integrating (averaging) over time. Each time sample can be treated as an independent and identically distributed random variable. For a sampled bandwidth B, the Nyquist sampling theorem [27] requires that the sample rate be twice the sampled bandwidth. If the integration occurs for a time t_{int}, then the number of independent samples that are integrated is $N_s = 2Bt_{int}$. The standard deviation of the noise (and the rms power for Gaussian noise) is decreased by a factor of $1/\sqrt{N_s}$ [24].

In terms of the equivalent noise temperature, if T_Δ is the integrated noise temperature, then:

$$T_\Delta = \frac{T_{sys}}{\sqrt{2Bt_{int}}} = \frac{T_{sys}}{\sqrt{N_s}}. \tag{4.6}$$

For a source with flux S, the integrated SNR, SNR_Δ, is found by computing the ratio of T_a (from Equation 4.5) to T_Δ (from Equation 4.6) to give (in dB)

$$\text{SNR}_\Delta = 10\log_{10}\frac{S\eta_A A\sqrt{Ns}}{2k_B T_{sys}} = \text{SNR} + 5\log_{10} N_s. \tag{4.7}$$

Equation 4.7 states that a magnitude increase in the number of samples integrated corresponds to a 5 dB increase in SNR. As stated in [24], the above analysis applies for both the real and complex channels of the receiver. For the VSA to observe Cassiopeia A with 20 dB signal to noise ratio and a 1 MHz bandwidth requires an integration time of approximately 20 seconds.

The preceding equation applies to the temporal signal from each correlator (described in the next subsection). A similar analysis applies to the SNR of the final image. Following Wrobel's treatment in [24], the SNR in each pixel can be found from examining the Fourier inversion equation (see Equation 2.36). Each pixel in the image is a linear combination of all the visibilities. The $l = 0, m = 0$ pixel is given by

$$I_\nu^D(0,0) = \frac{1}{M}\sum_k V_\nu(u_k, v_k),$$

where M is the number of visibilities. The standard deviation of the noise in each pixel of the image is reduced by $1/\sqrt{M}$ (for more detail see [24]).

4.3.2 Simulations with Noise

With the background given above, we can now modify the model of Equation 4.1 to include residual noise after correlation. The field at each antenna is still the sum over the sources in the sky, but now we include a noise term in the sum:

$$E_i = \eta_i + \sum_{b=1}^{q}\alpha_b e^{j\mathbf{k}_b\cdot\mathbf{r}_i}, \tag{4.8}$$

or more succinctly: $E_i = \eta_i + s_i$. Since the VSA noise is dominated by the thermal noise of the LNAs, the noise has a Gaussian distribution with zero mean. The standard deviation will be determined below. The vector $\mathbf{x}$ in Equation 4.2 is now the sum of a signal vector and a noise vector: $\mathbf{x} = \mathbf{s} + \eta$.

The true correlation matrix is the expectation of the outer product of $\mathbf{x}$ with itself: $R = E\{\mathbf{x}\mathbf{x}^H\}$. It is expanded as

$$\begin{aligned} R &= \left\langle \mathbf{x}\mathbf{x}^H \right\rangle \\ &= \left\langle \mathbf{s}\mathbf{s}^H \right\rangle + \left\langle \mathbf{s}\eta^H \right\rangle + \left\langle \eta\mathbf{s}^H \right\rangle + \left\langle \eta\eta^H \right\rangle \\ &= R_s + \left\langle \mathbf{s}\eta^H \right\rangle + \left\langle \eta\mathbf{s}^H \right\rangle + R_\eta. \end{aligned} \tag{4.9}$$

The first term in the sum is the quantity of interest, the signal correlation matrix R_s. The source amplitudes, α_b, in Equation 4.8 are chosen so that the diagonal elements of R_s are equal to the sum of the source powers, that is, $\alpha_b = \sqrt{\frac{1}{2}S_b\eta_A AB}$.

The noise is modelled as white, Gaussian noise that is uncorrelated with the signal. The last quantity, R_η, is the noise correlation matrix. The expectation of R_η is therefore $\sigma_\eta^2 I$, where $\sigma_\eta^2 = k_B B T_{sys}$. As stated above, the power in the noise decreases as $1/\sqrt{N_s}$. The residual noise in R_η is modelled as gaussian with standard deviation (i.e. rms power) equal to $\frac{k_B B T_{sys}}{\sqrt{2N_s}}$. The factor of $\sqrt(2)$ is due to the SNR increase of complex correlation (see [14]) [1].

The residual noise due to the cross correlation terms is not included in this model. The expected value of the cross correlation terms is zero since the signal and the noise are uncorrelated. Even if the cross correlations were included, the they are small compared to the noise, since $T_a \ll T_{sys}$, so that their residual noise contributions are negligible.

With our residual noise model in place, we can simulate more realistic images for the VSA. We can also simulate fringe functions.

Figure 4.6 shows a simulated image with the phase tracking center set to the coordinates of Cassiopeia. There are three simulated sources with the same source flux (in this case 1500 Jy). Mixing of the side lobes from each source causes severe undulation in the dirty image.

Figure 4.7 shows a fringe function that composes part of the final dirty image shown in 4.6. This function is the product of the signals from Antenna 1 and Antenna 4 (see Figure 3.16), and corresponds to the temporal evolution of the correlation

[1]The author is indebted to Bas Van der Tol for introducing this model to him.

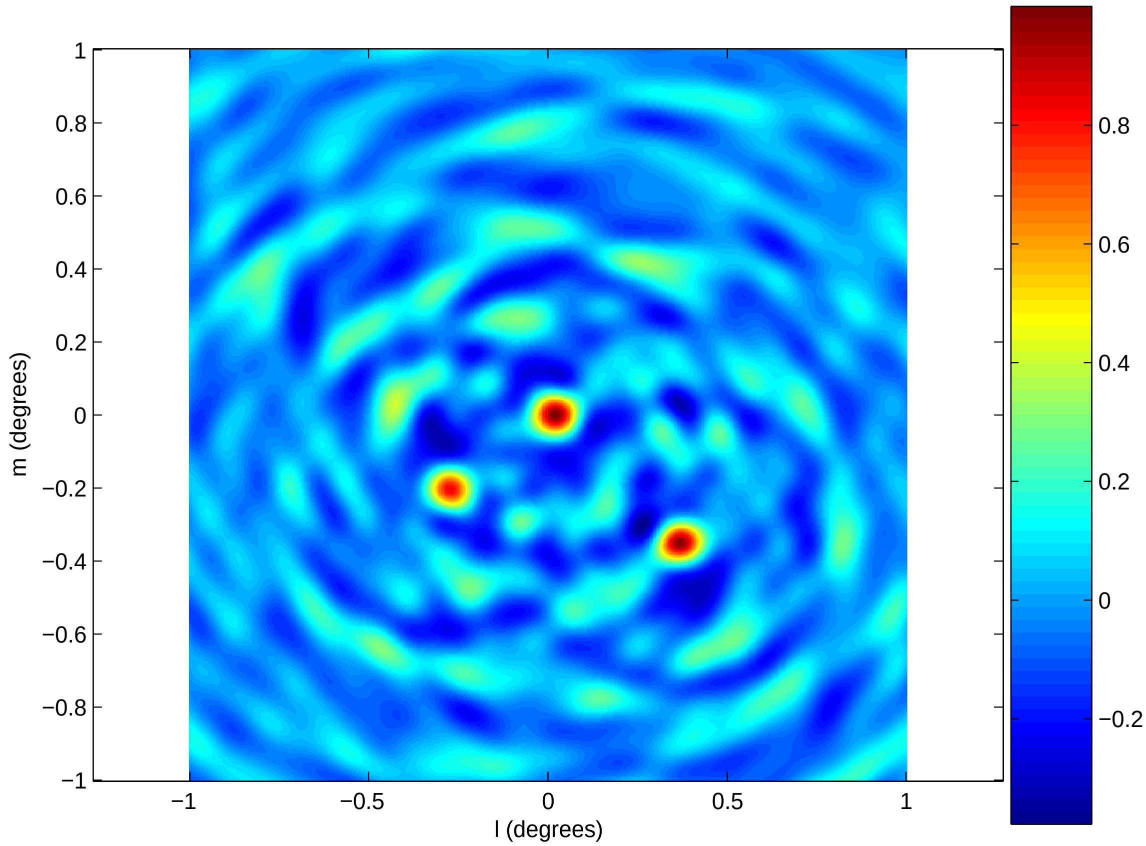

Figure 4.6: Simulated dirty image using noise model.

element $R_{1,4}$. Figure 4.8 provides a zoomed in view of the fringe function in Figure 4.7. Figure 4.8 shows the residual noise as small amplitude variations of the fringe function.

4.4 Phase Stability Simulations

As mentioned in the introduction to this chapter, one of our concerns for the VSA was that we would not be able to accurately measure the phase component of the spatial coherence between our antennas. We were concerned that the phase would be noisy or even free-running due to temperature induced drift and instrument instabilities. As discussed in Section 2.1, the measured spatial coherence, or visibility, consists of an amplitude and a phase measurement. In radio astronomy literature,

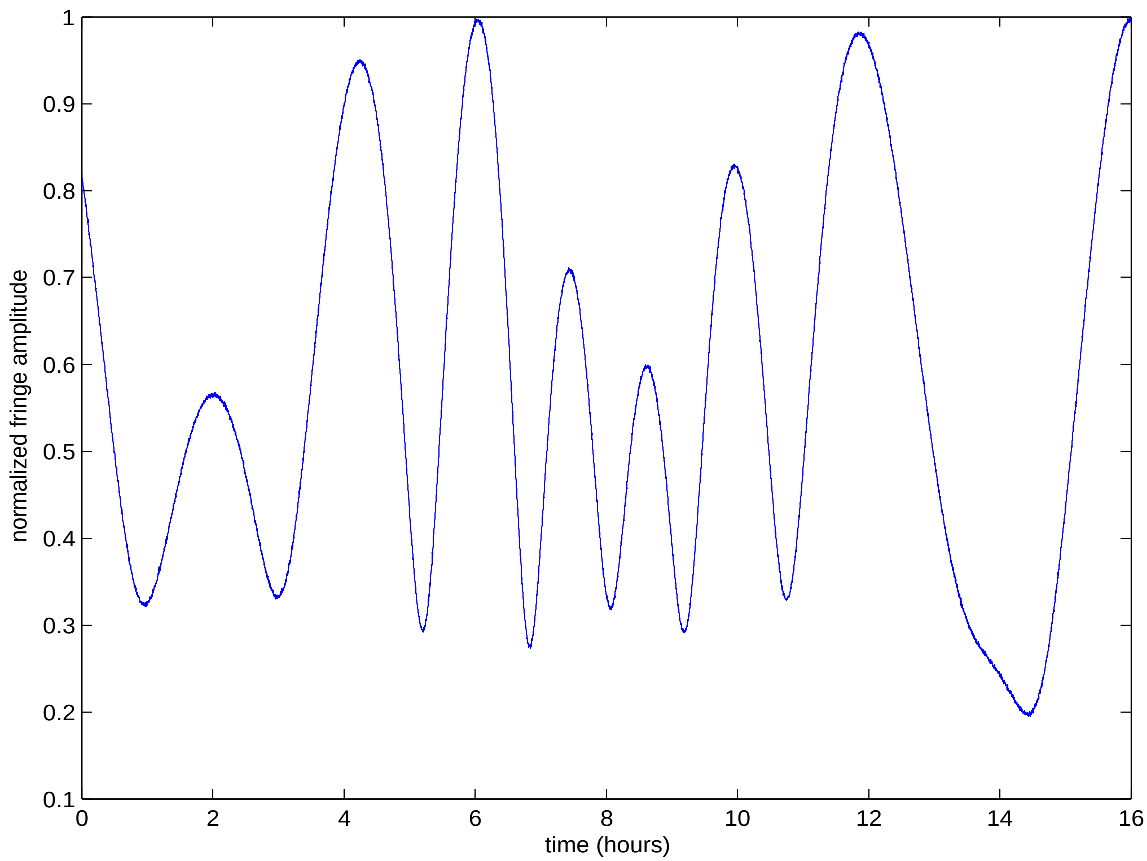

Figure 4.7: Fringe function for the signal between antennas 1 and 4.

measuring the relative phase between antenna pairs is considered the most difficult measurement, and the most crucial. For example, the measured phase between elements of the VLA is calibrated to have "phase stability of better than one degree of phase per gigahertz of observing frequency" [28].

The remainder of this section is devoted to describing simulations performed to examine the effects of phase error and quantify the amount of tolerable phase error. The source model for these simulations is the residual noise model of the previous section, as defined by Equation 4.8, with a phase error term, θ_e added:

$$E_i(n) = \sum_{b=1}^{q} \alpha_b e^{j\mathbf{k}_b \cdot \mathbf{r}_i + \theta_e(n)}. \tag{4.10}$$

Equation 4.10 gives the field at each antenna in the VSA, except the phase reference

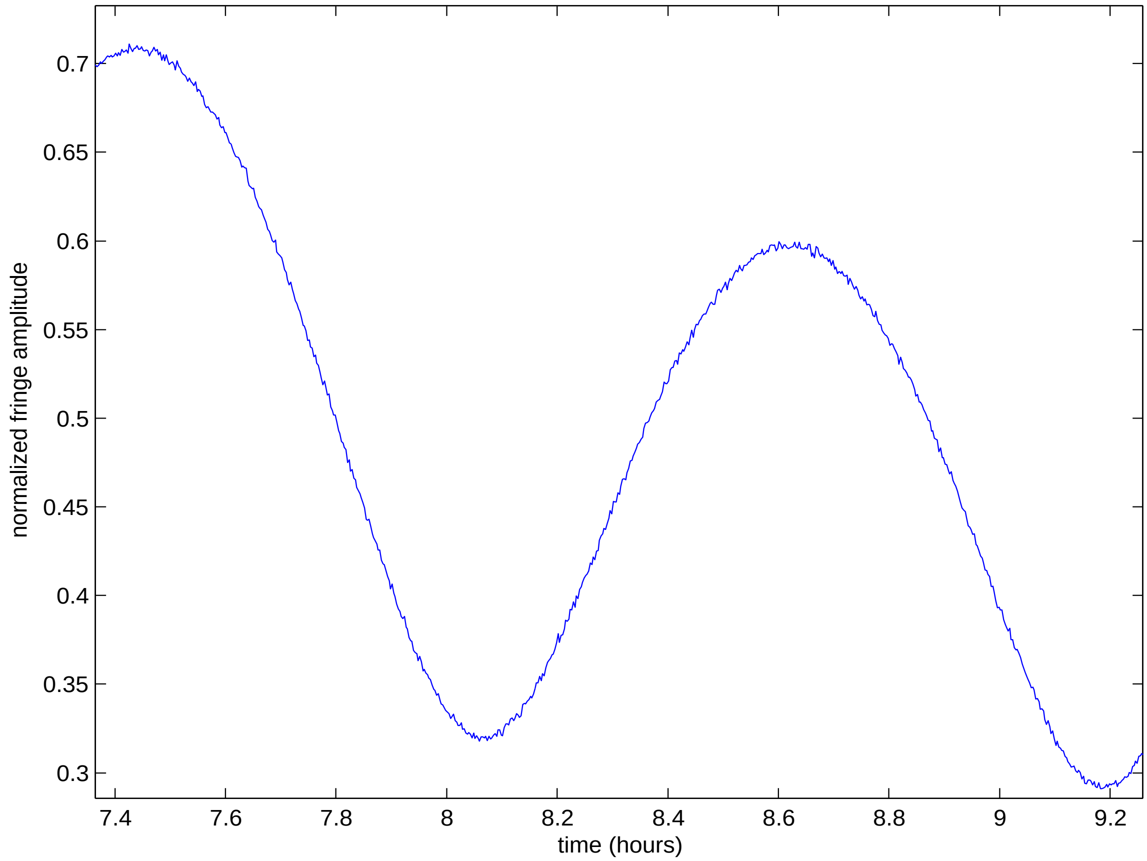

Figure 4.8: Fringe function generated with the stochastic model of Equation 4.9.

antenna. Since it is only relative phase that is measured by a synthesis array, one antenna in the array is modelled as error free. Equation 4.10 shows explicitly the time dependence of θ_e. Implementing Equation 4.10 in simulation requires specifying an erroneous phase at each time the E field is sampled.

The simulations that are described below were used to determine the effects of phase error on the final image. The thresholds set on phase error are based on how much error is introduced into the final image. To quantify the amount of error in the final image, an L_2 matrix norm was used. The two matrices in Equation 4.11 are an error-free image, I_0, and an error corrupted image, I_e:

$$e_I = \| I_0 - I_e \|_2 . \tag{4.11}$$

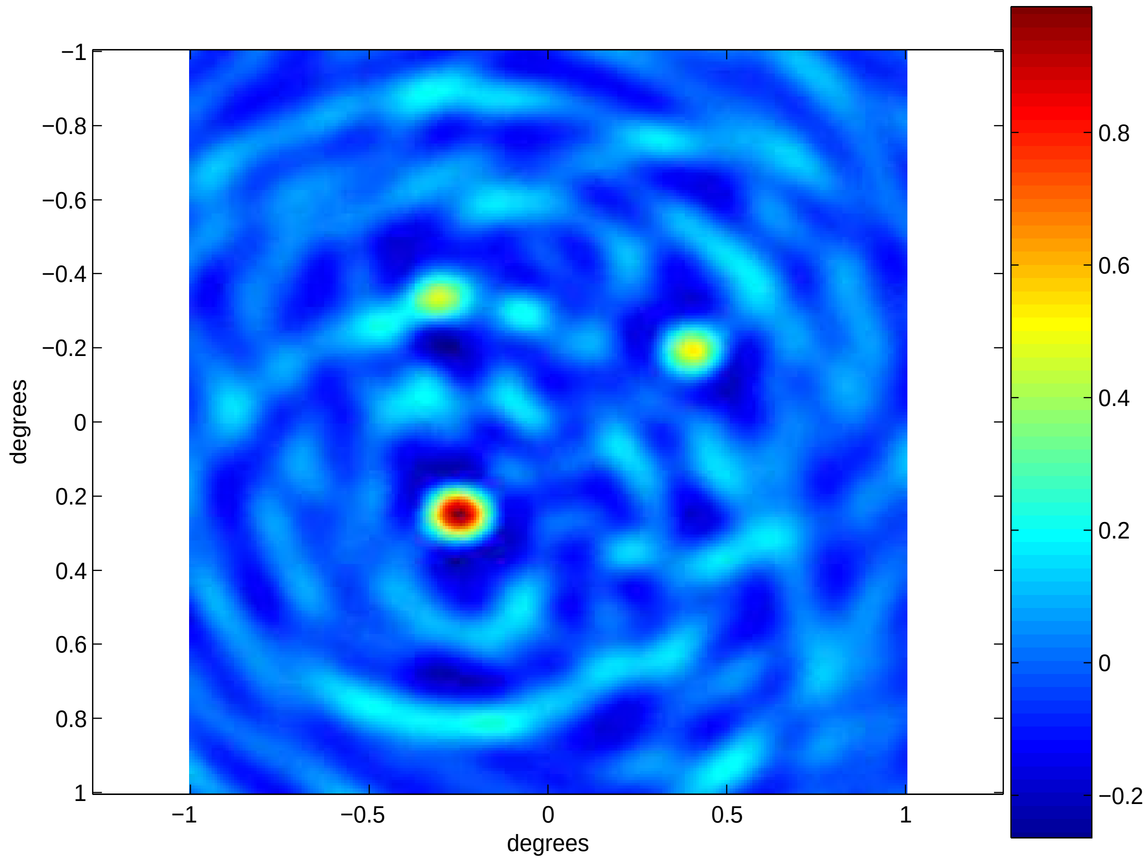

Figure 4.9: The error free image I_0.

The quantity e_I is the image error in the final image due to the error in the phase. The error free image I_0 is shown in Figure 4.9. As shown in the figure, I_0 consists of three point sources each with different source intensity.

The model for the phase error term θ_e, is based (loosely) on the types of phase error expected from physical components that make up the VSA. The next subsections describe a specific error model and display the effects that type of error has on the final image. As in the previous section, residual noise is added as a perturbation of the true signal correlation matrix (see Equation 4.9).

4.4.1 Random Phase Error

The first type of error modelled was random phase error at each time sample. Two different distributions for random phase error were used: Gaussian and uniform. For the normal case,

$$\theta_e(n) \sim N(0, \sigma^2). \tag{4.12}$$

For the uniform case,

$$\theta_e(n) \sim U(-a, a). \tag{4.13}$$

This type of model for error serves to simulate thermal effects.

Figure 4.10 shows the image error, e_I, as a function of standard deviation for the model in equations 4.12 and 4.13. For the uniform distributed phase noise, the value for a in Equation 4.13 is chosen based on the standard deviation: $\sigma = a/\sqrt{3}$.

We examined simulated images with different values of σ at regular intervals along the curves shown in Figure 4.10. We decided to place the threshold for this type of error at the elbow in the log domain, at $\sigma = 15$ degrees (for both the Gaussian and uniformly distributed error). This threshold is arbitrary and is based on the fact that images below this threshold are visually indistinguishable from I_0. Figure 4.11 shows an image with Gaussian phase noise at every sample with $\sigma = 15$ degrees.

4.4.2 Calibration Offset Error

Some of the simulations were directed at setting a threshold for calibration error. To simulate calibration error, an initial phase offset is chosen for each antenna. The phase offset remains constant for the duration of the experiment (it does not vary with each n). The values for the phase offsets were chosen randomly from two distributions: Gaussian and uniform. For the Gaussian scenario,

$$\theta_e \sim N(0, \sigma^2). \tag{4.14}$$

For the uniform scenario,

$$\theta_e \sim U(-a, a). \tag{4.15}$$

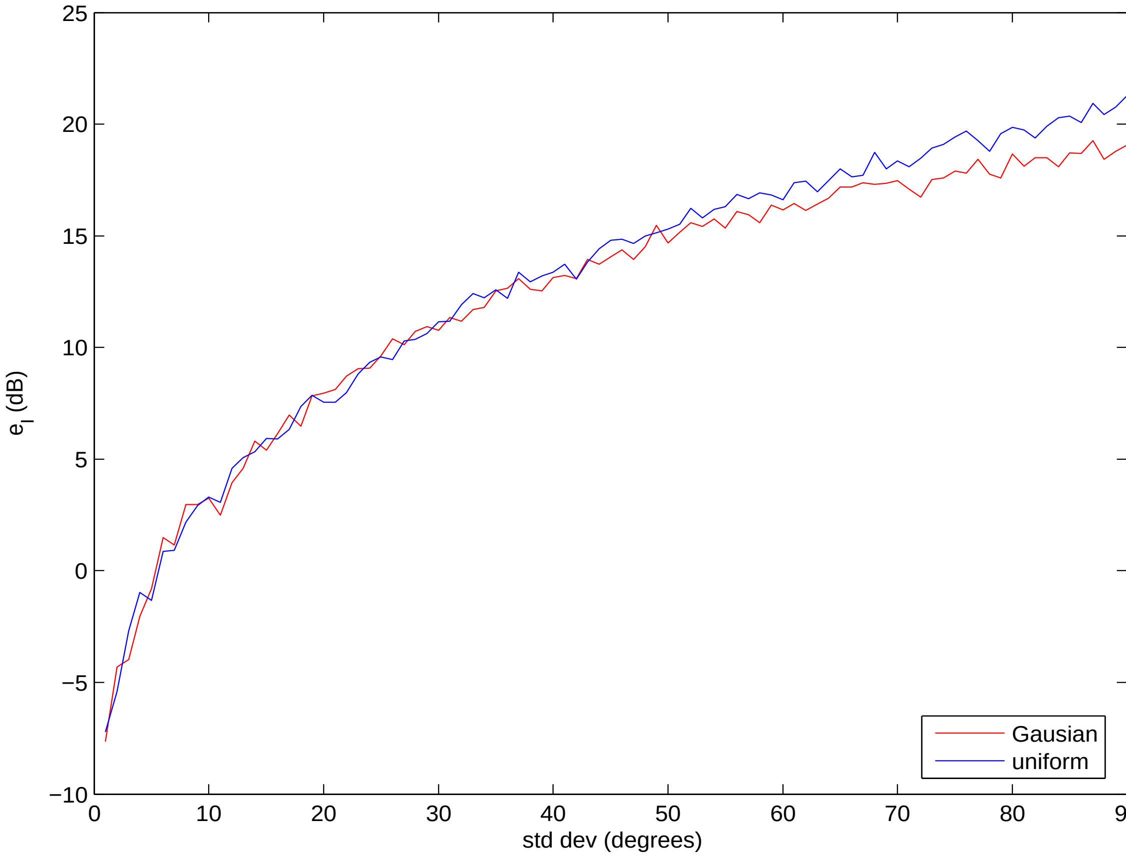

Figure 4.10: Error for Gaussian and uniform phase noise at every E field sample.

The results of the calibration offset simulations are shown in Figure 4.12. (This figure is similar to Figure 4.10 in structure.) Just as in the preceding subsection, we looked at images with different values of σ at regular points along the two curves shown in the figure. We set the threshold at $\sigma = 10$ degrees for both the uniform and Gaussian distributed offset error. Figure 4.13 shows an image with offset error distributed $\theta_e \sim N(0, \sigma = 10°)$. Once again, the image is visually close to I_0.

4.4.3 Linearly Drifting Phase

A final series of simulations was performed modelling the phase error as linearly drifting. This is modelled as

$$\theta_e = \theta_d n, \tag{4.16}$$

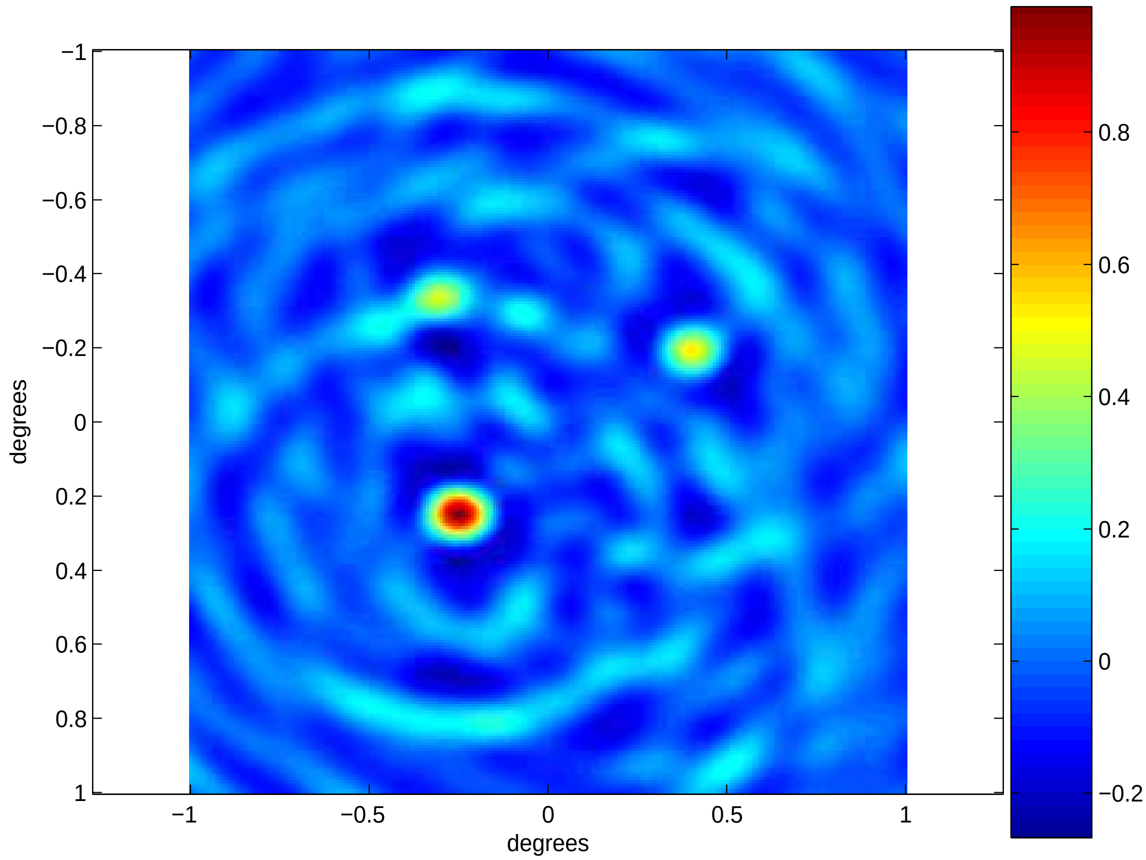

Figure 4.11: Image with Gaussian error in phase at every E field sample, $\sigma = 15$ degrees.

where θ_d is the drift rate of the phase in degrees per integration time. It is a random parameter chosen from a normal distribution:

$$\theta_d \sim N(0, \sigma^2). \tag{4.17}$$

θ_d is set at the beginning of a given simulation and remains constant for the duration. The results of the simulations are shown in Figure 4.14.

This type of error is the most severe that was modelled. After examining images with different values for σ, the threshold was set at $\sigma = 3°/\text{hr}$. The units on θ_d have been normalized to hours to allow for different integration times.

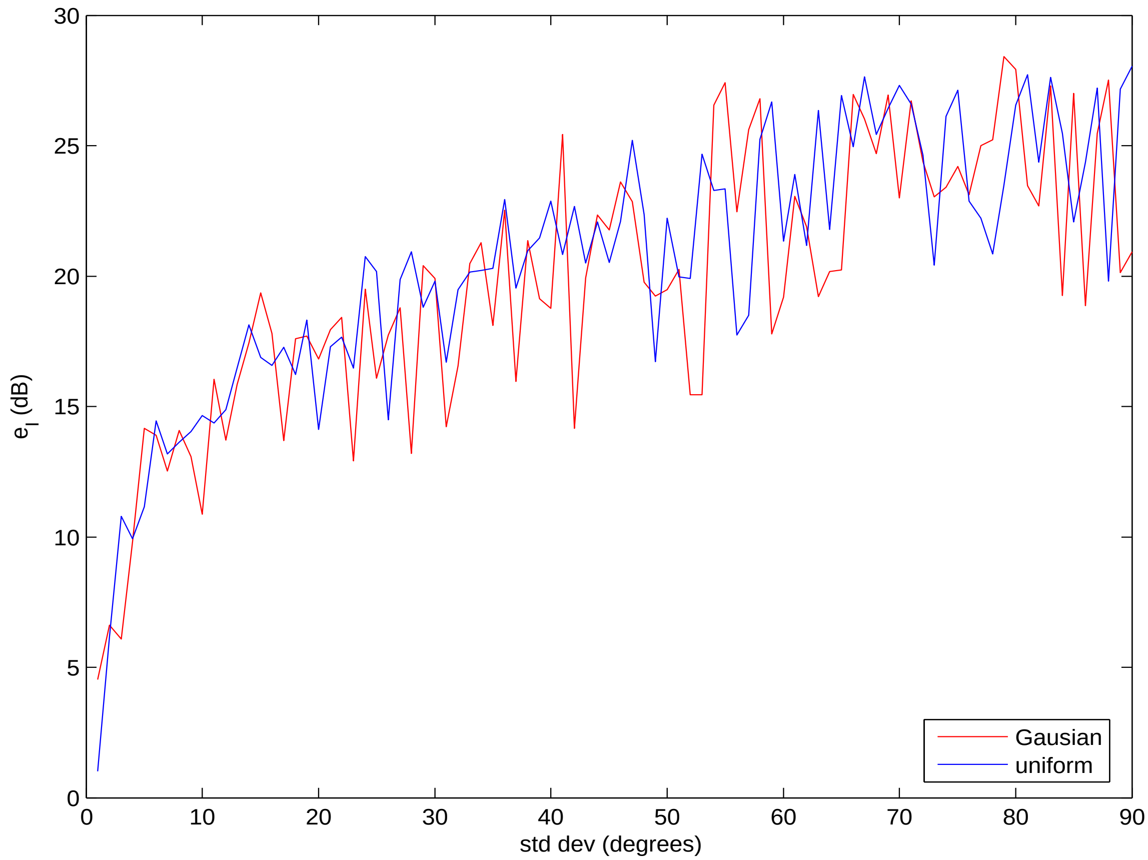

Figure 4.12: Error for Gaussian and uniform chosen phase offset.

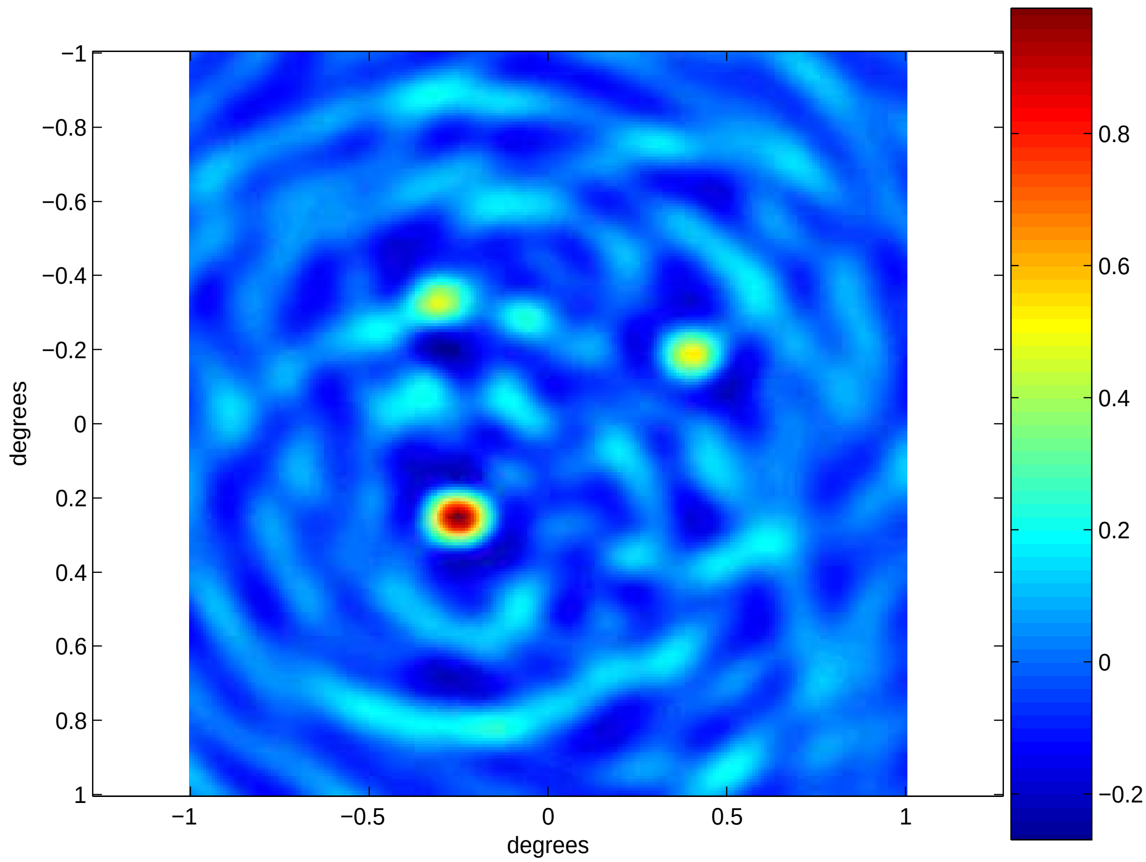

Figure 4.13: Image with Gaussian error in phase offset, $\sigma = 10$ degrees.

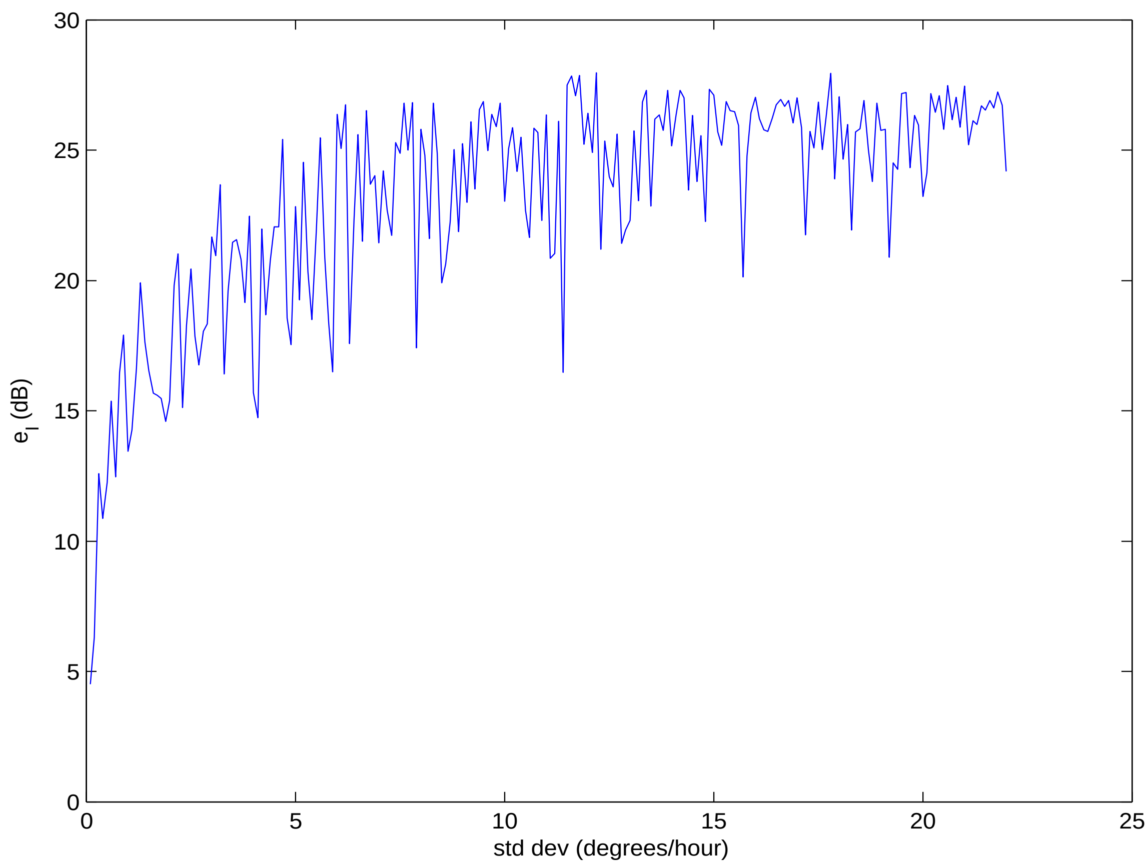

Figure 4.14: Image error for linearly drifting phase error.

Chapter 5

Performance

5.1 Introduction

This chapter describes the completed Very Small Array, the initial calibration work, and provides results from data taken with the array. The chapter is structured as follows. Section 5.2 describes the Very Small Array. It provides some photographs of equipment and details the key software for the VSA. Section 5.3 documents the "first light" observation for the VSA. Section 5.4 details the initial observations made with the VSA and the data analysis that was performed to characterize the array. Lastly, Section 5.5 documents the first image generated with the VSA.

5.2 The Completed VSA

5.2.1 The Antennas

The satellite dish antennas were built and placed according to the design work described in Chapter 3. The VSA is located at $111°38'53.61$" West longitude, $40°14'48.6''$ North latitude, at an altitude of 1429.4 meters above sea level. (See Chapter 3 for a top side view of the antenna's positions.)

Figure 5.1 shows a photograph of the phase reference antenna for the VSA: antenna 1 (named Eve). Antenna 1 is located on the Northwest corner of the Clyde Building. This picture is taken facing Northwest. Some of the buildings on BYU campus can be seen in the background. Antenna 1 is tracking Cassiopeia A.

Figure 5.2 shows a photograph of Antenna 2 of the VSA (Sarah). This picture is taken in the early evening facing East. Antenna 2 is located on the Northeast corner

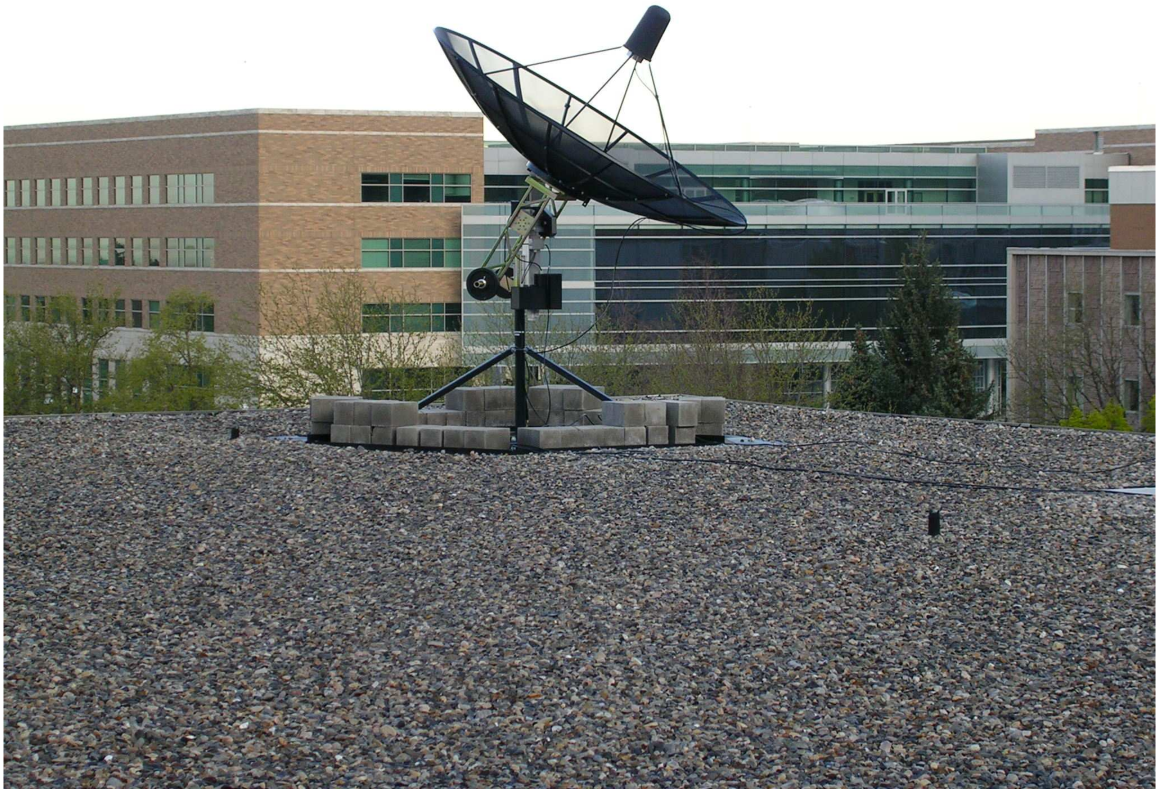

Figure 5.1: Antenna 1 (Eve) tracking Cassiopeia A in the early morning, April 2005.

of the Clyde Building. Antenna 2 is also tracking Cassiopeia A. Cascade Mountain (also known as "Y" Mountain) can be seen in the background.

Figure 5.3 shows a photograph similar to those in Figures 5.1 and 5.2. Antenna 3 (Rebekah) and Antenna 4 (Rachel) are shown tracking Cassiopeia A in the early morning. This picture is taken facing South. West Mountain can be seen in the background of the photograph. Antennas 3 and 4 are located on the West side of the Clyde Building.

The positions of the antennas relative to one another were measured with a total station (a device used in surveying to measure range and bearing), and are shown in Table 5.1. The full measurements made during surveying are found in Appendix

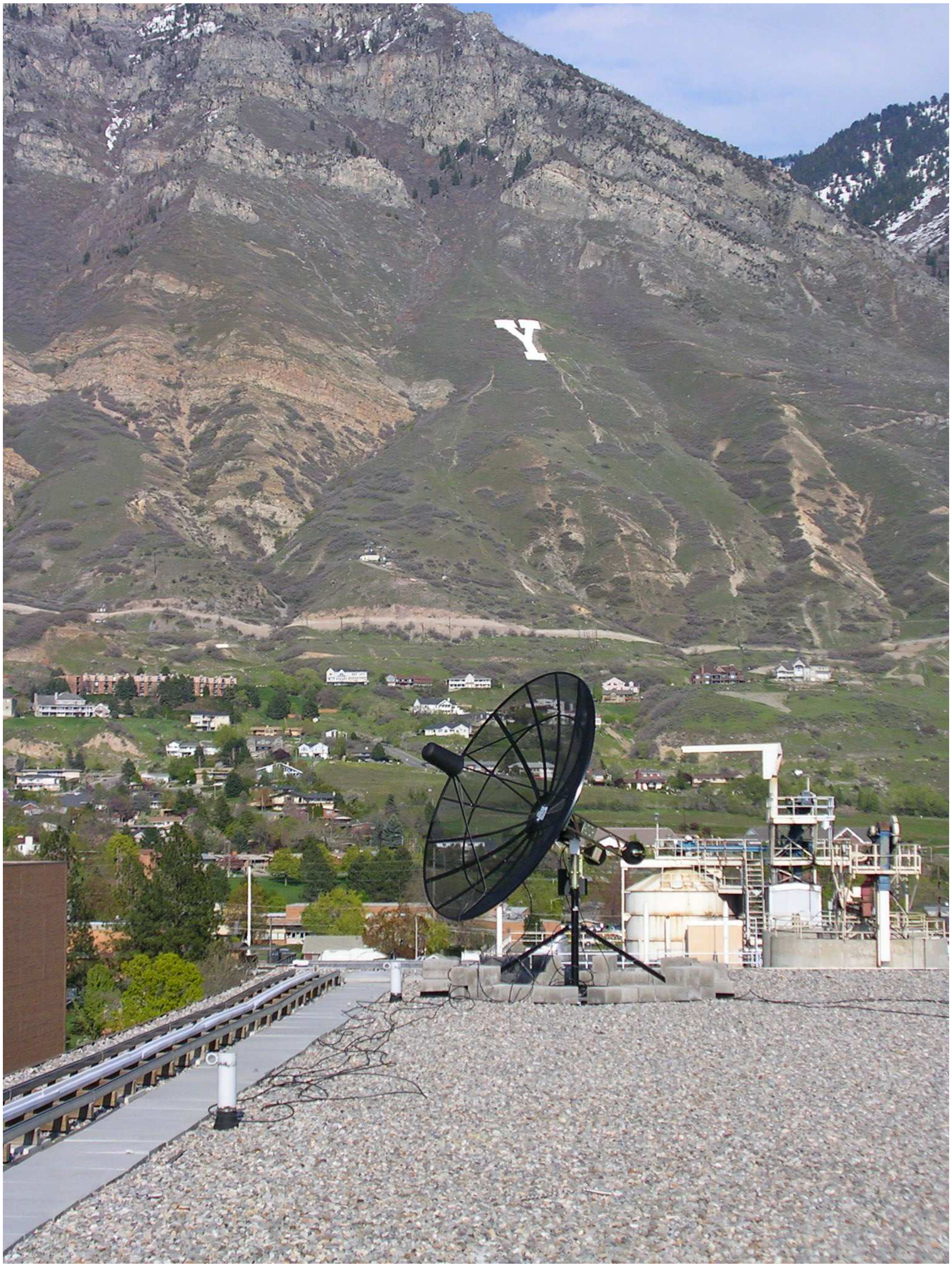

Figure 5.2: Antenna 2 (Sarah) tracking Cassiopeia A in the early evening, April 2005.

Figure 5.3: Antenna 3 (Rebekah), in the foreground, and antenna 4 (Rachel), in the background, tracking Cassiopeia A in the early morning, April 2005.

A. The coordinates are given in meters East, North, and Up (toward Zenith) with respect to antenna 1.

Figure 5.4 is a closer view of antenna 1. The photograph shows the custom positioners just below the dish of the antenna. Figure 5.5 is a photograph of the Walkenhorst/Hansen receiver [21] [22] being used to down-convert the signals from the antennas. Signal cables from the antennas can be seen on the right side of the receiver. The coaxial cables on the left side are connected to the DSP.

Figure 5.6 shows a photograph of the antenna controller boxes. Above the controller boxes are the LNA power supplies. Figure 5.7 shows the desktop computer that controls DSP and interfaces with the controller boxes. The Pentek DSP can be

Figure 5.4: A close up of antenna 1 (Eve).

Table 5.1: Coordinates of each antenna with respect to Antenna 1 (specified in meters).

Antenna	East	North	Up
1	0.0	0.0	0.0
2	54.2346	0.4068	0.00290
3	-5.7985	-45.9484	-0.0303
4	0.4402	-72.5197	0.0285

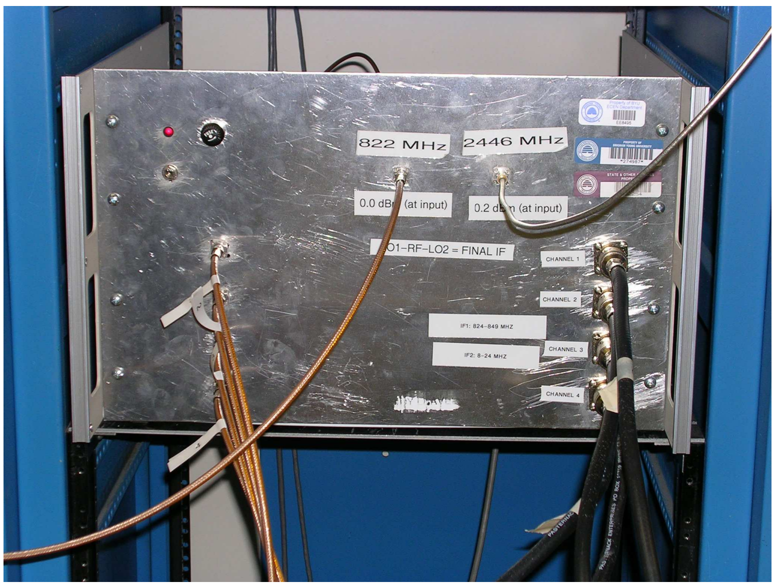

Figure 5.5: The Walkenhorst/Hansen receiver.

seen to the right of the computer monitor. The photograph also shows the coaxial cables connecting the DSP to the receiver.

5.2.2 Software

The author wrote a Matlab script called "controlcenter" to control the antennas from a desktop PC. Controlcenter provides a graphical interface for serial port control of the antenna position controllers, and includes algorithms to compute azimuth and elevation for astronomical phenomena based on the computer's clock and

Figure 5.6: The position controllers for the antenna positioners and the LNA power supplies.

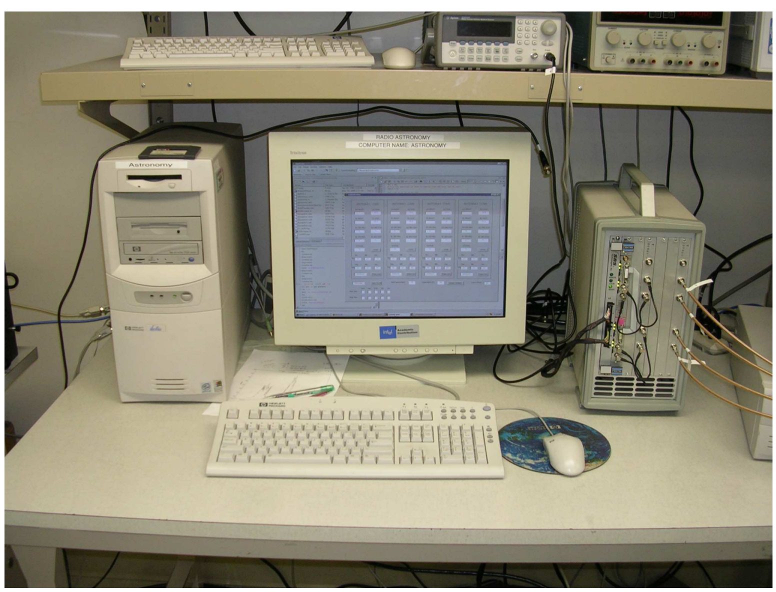

Figure 5.7: The host PC and the Pentek DSP.

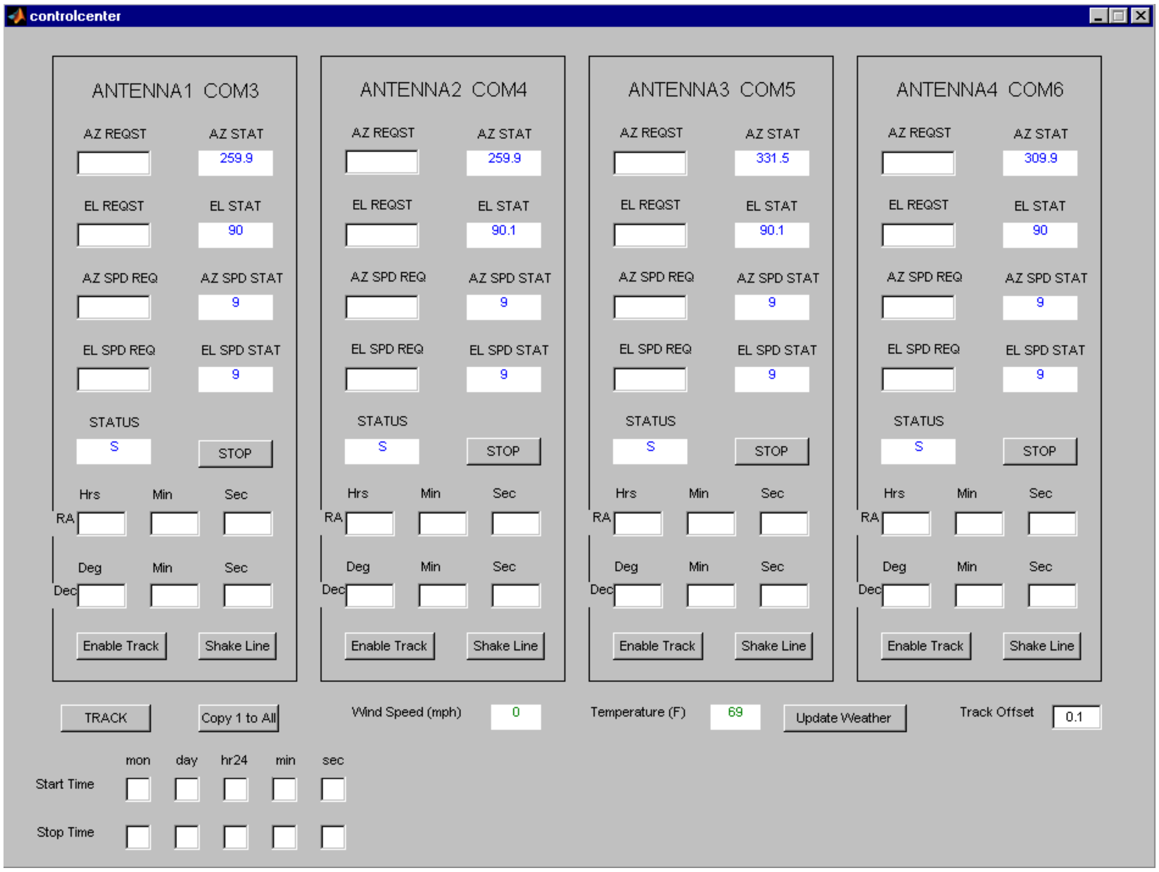

Figure 5.8: A screen shot of the controlcenter GUI.

the latitude and longitude of the array [1]. A screen shot of controlcenter's GUI is shown in Figure 5.8.

In the process of editing the data from observations it is helpful to have a record of the weather conditions that were present during observation (see Section 5.4). It is also essential to monitor weather conditions while the antennas are tracking and stow the antennas if the weather becomes severe (see Appendix A for guidelines on running the array). The author wrote Matlab software to monitor and record the weather conditions at the VSA. There are a number of weather stations on BYU Campus. A weather station on the top of the Eyring Science Center (a building just

[1]The author is once again indebted to Bas Van der Tol. The algorithms that compute azimuth and elevation for astronomical coordinates were written by Bas.

East of the Clyde Building) continuously updates a web page. The Matlab weather software queries this web page periodically and records the data in a file.

Software was also written to analyze the measured data from the VSA. The process of forming an image from measured visibilities is described in detail in Chapter 2. The analysis tools are the software implementation of the mathematics covered in that chapter. These tools include programs to edit and flag the data, calibrate the measurements, compute u and v for the measured visibilities, and generate an image from the data. Some of the analysis software is described in Section 5.4. All of the software for the VSA and processing the data from the VSA is described in Appendix B.

5.3 First Fringe Function

One of the first tests performed with the VSA was to measure a fringe function. As described in Section 2.2, the fundamental element of an image synthesis array is the interferometer. The output of an interferometer is a fringe function. The phase and amplitude of the fringe function constitute the visibilities that are Fourier transformed to create an image.

For a point source at the phase tracking center, the fringe function is given by Equation 2.24 which is repeated below as Equation 5.1 for convenience:

$$V_\nu(u,v) = |A|^2 e^{-j\omega_{RF}\tau_g}. \tag{5.1}$$

As discussed in Section 2.3, τ_g varies with time as the antennas track the source. τ_g can be predicted, given the locations of the antennas and the position of the source. From Equation 5.1, the fringe function for a point source at the phase tracking center has constant amplitude with a time-varying phase.

To measure a fringe function, the VSA tracked Cassiopeia A. Cassiopeia A has an angular width of approximately 5 arcmin, and is, therefore, a point source for the VSA (see Section 4.2). The first fringe function generated with the VSA is shown in Figure 5.9. (Note that this observation is uncalibrated.) The hour angle at the start of this observation is $H_0 = 6 : 52'$. The observing bandwidth is 1 MHz and the center frequency is 1420 MHz. The integration time is 1 second and the observation spans 16 minutes.

This fringe clearly does not display a constant amplitude. It turns out that the fringe function has DC offset which will be discussed in detail in the next section. The analytic expression for this fringe function is not given by Equation 5.1. Rather, it takes the following form:

$$V_\nu(u,v) = A_{DC} + |A|^2 e^{-j\omega_{RF}\tau_g}. \tag{5.2}$$

The modulus of V_ν in Equation 5.2 oscillates with the variation of τ_g as seen in Figure 5.9. The task of calibrating the array is also discussed in the following section. The author has left the fringe uncalibrated for nostalgia's sake.

5.4 Observing a Calibrator with the VSA

Cassiopeia A was observed with the VSA for many hours, at different observing frequencies, and under differing conditions in an effort to work the bugs out of the array and ultimately make an image from observed data. The data described in this section is Cassiopeia A data with an integration time of 10 seconds with a 1 MHz bandwidth unless otherwise specified. The VSA has not been calibrated absolutely in amplitude. The amplitude units in the figures that follow are in arbitrary units instead of Janskys or Watts.

The DSP described in Section 3.4 samples data on four channels identified by a letter: A, B, C, and D, which were connected to antennas 1, 2, 3, and 4 respectively. Some of the figures in this section use the channel letter designations instead of the antenna numbers.

5.4.1 Flagging and Editing Data

The radio astronomy literature refers to identifying and discarding erroneous or corrupted data as editing or flagging the data. Identifying anomalous data from the VSA is accomplished by examining the phase and amplitude from the correlations generated by the DSP. Both the cross correlations and the self correlation terms

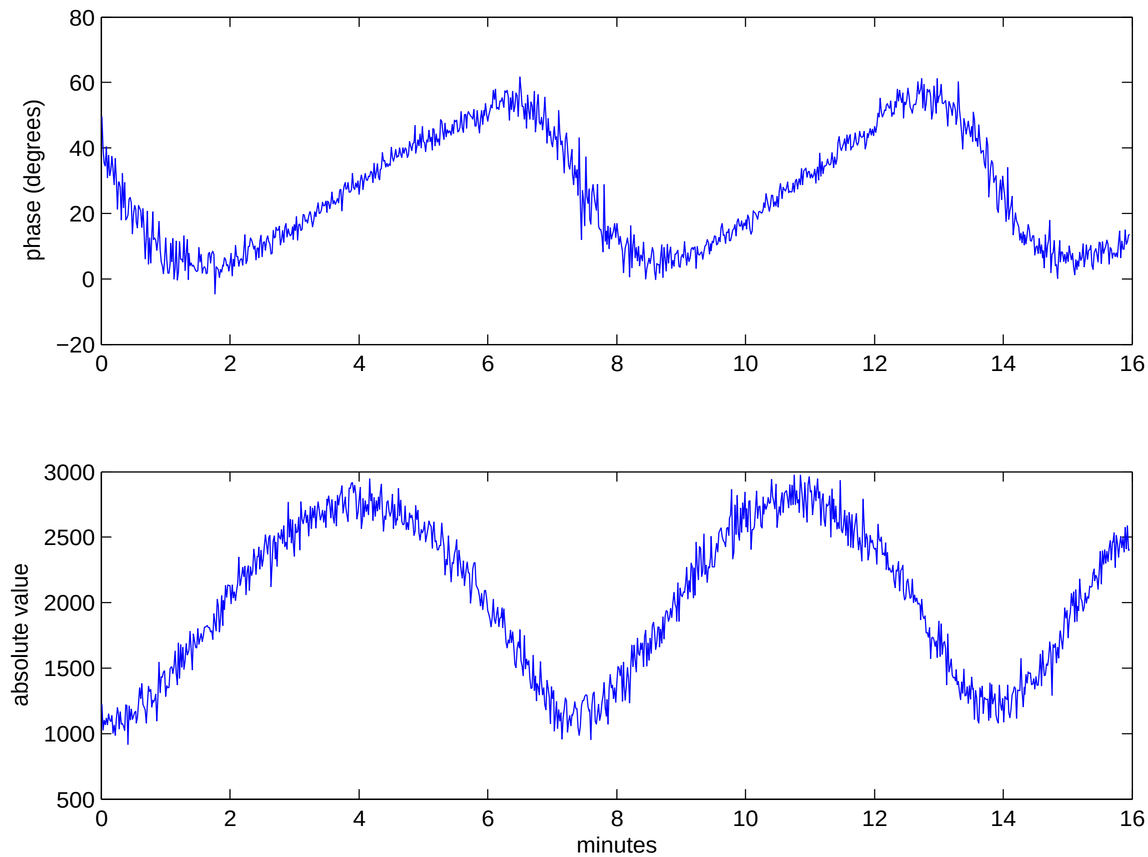

Figure 5.9: The first fringe function observed with the VSA. The array was observing Cassiopeia A.

should be examined when editing data. The earmarks for erroneous data are spikes in the phase and/or the amplitude, and noise-like phase.

5.4.2 Temporal Data

The DSP correlator code outputs two buffers of temporal data for each channel. This is the sampled data from the A/D (at a decimated sample rate) prior to being correlated. For illustration, some temporal data taken during an observation is shown in Figure 5.10. The decimated sample rate for this data is 1 Ms/sec. The plots in this figure show approximately 2 msec of sampled data. Plots like these are useful for verifying the synchronization of the DSP channels and when observing strong sources

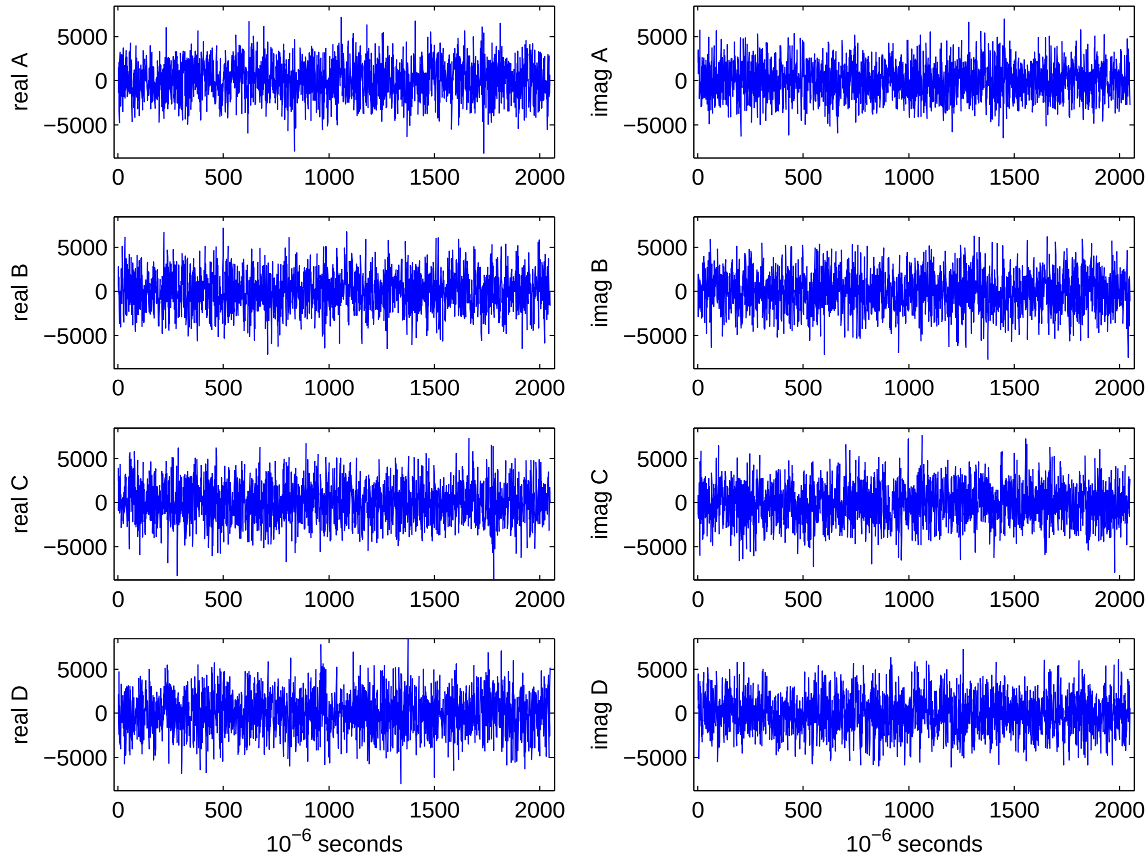

Figure 5.10: Time data for the real and imaginary outputs from each of the four channels.

with instantaneous SNR $>$ 0 dB. However, for sources with an SNR $<$ 0 dB, temporal data simply appears to be white noise (as it should).

5.4.3 Phase and Amplitude Analysis

As previously stated, the best method of identifying corrupt data is to examine the (correlated) phase and amplitude as they progress over time. Unlike the temporal data, legitimate phase and amplitude is not noise-like. To see what corrupted data looks like consider Figure 5.11. The figure shows the phase (in degrees) of each of the six cross correlation terms. It can be seen that only the phase of R_{12}, channels A and B, is not noise-like. The other five terms appear to be noise. The corresponding

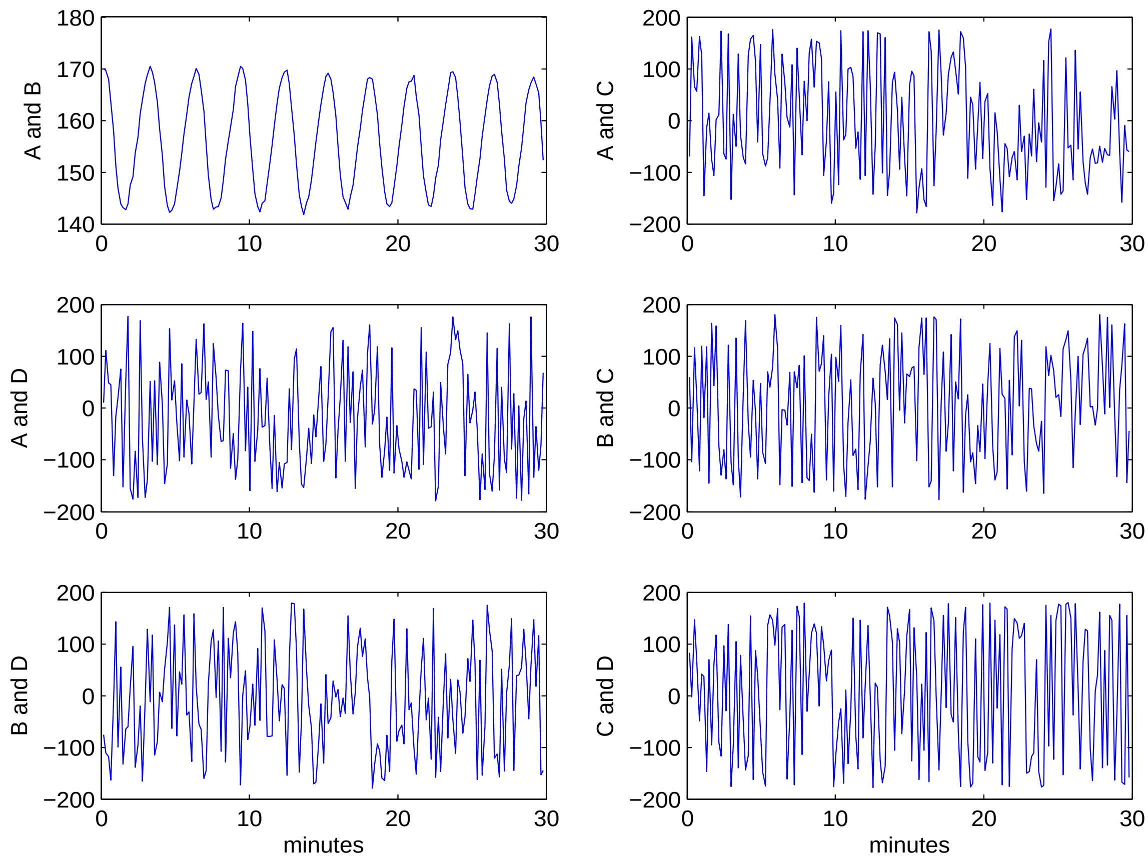

Figure 5.11: Five channels of corrupted phase data.

amplitude data is shown in Figure 5.12. To be safe, this entire data set is not included in the final image.

5.4.4 Measured versus Predicted

As already noted, Cassiopeia A is a point source for the VSA. As such, it can be used as a calibrator. For a point source, measured phase and amplitude data can be compared against their predicted values. Figure 5.13 shows the predicted and the measured phase (versus time) for the baseline between antennas 3 and 2. Measured phase does not match the predicted phase. The predicted phase wraps many times during the hour of observation time, while the measured phase simply oscillates around a mean value of approximately -50 degrees.

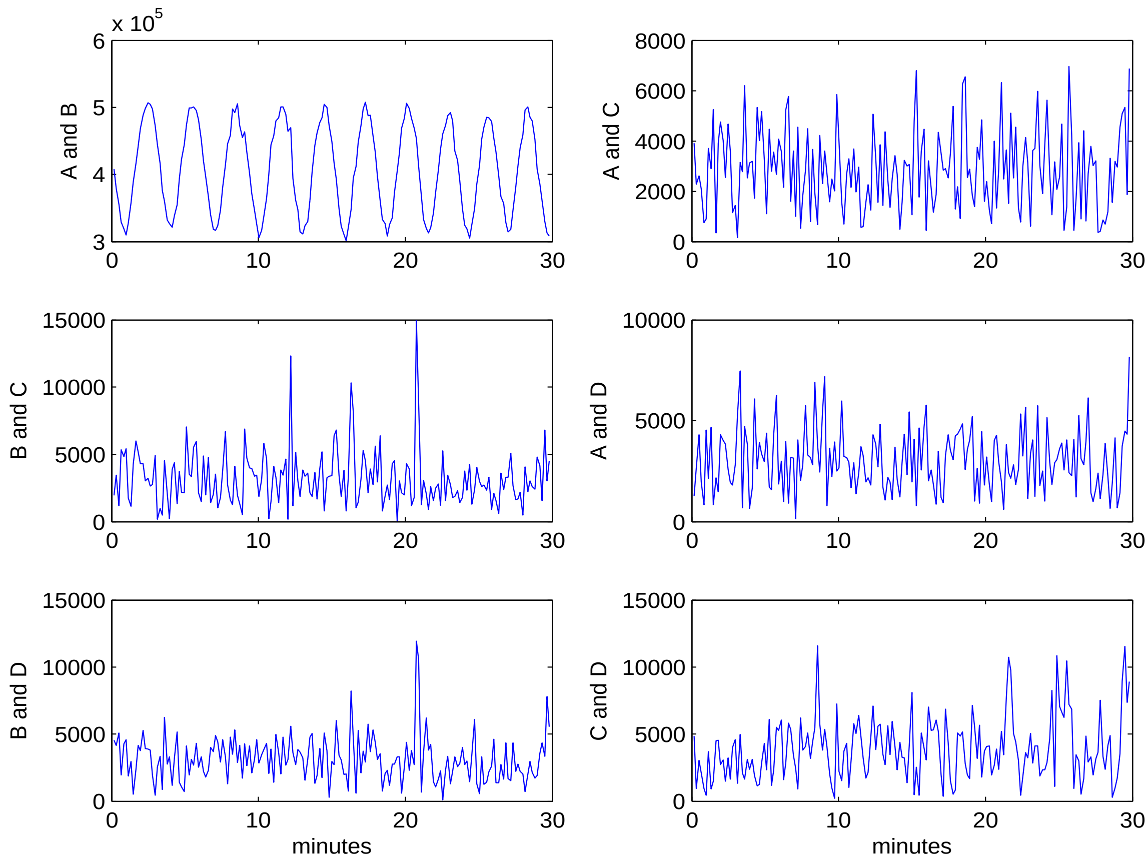

Figure 5.12: Five channels of corrupted modulus data (uncalibrated amplitude units).

Additional insight can be gained by plotting the real and imaginary parts of the correlator output versus their predicted values. Figure 5.14 shows the predicted outputs of the two correlator channels. Figure 5.15 shows the measured values.

5.4.5 Cross Coupling

It is apparent by comparing the predicted and measured data that there is a DC offset in the measurements. The predicted real and imaginary data have a DC value of zero. The measured data do not. The offset in the measured data is different for the real and imaginary channels. In addition, the offsets on each channel drift with time. Part of the reason for the DC offset is cross talk between the correlator

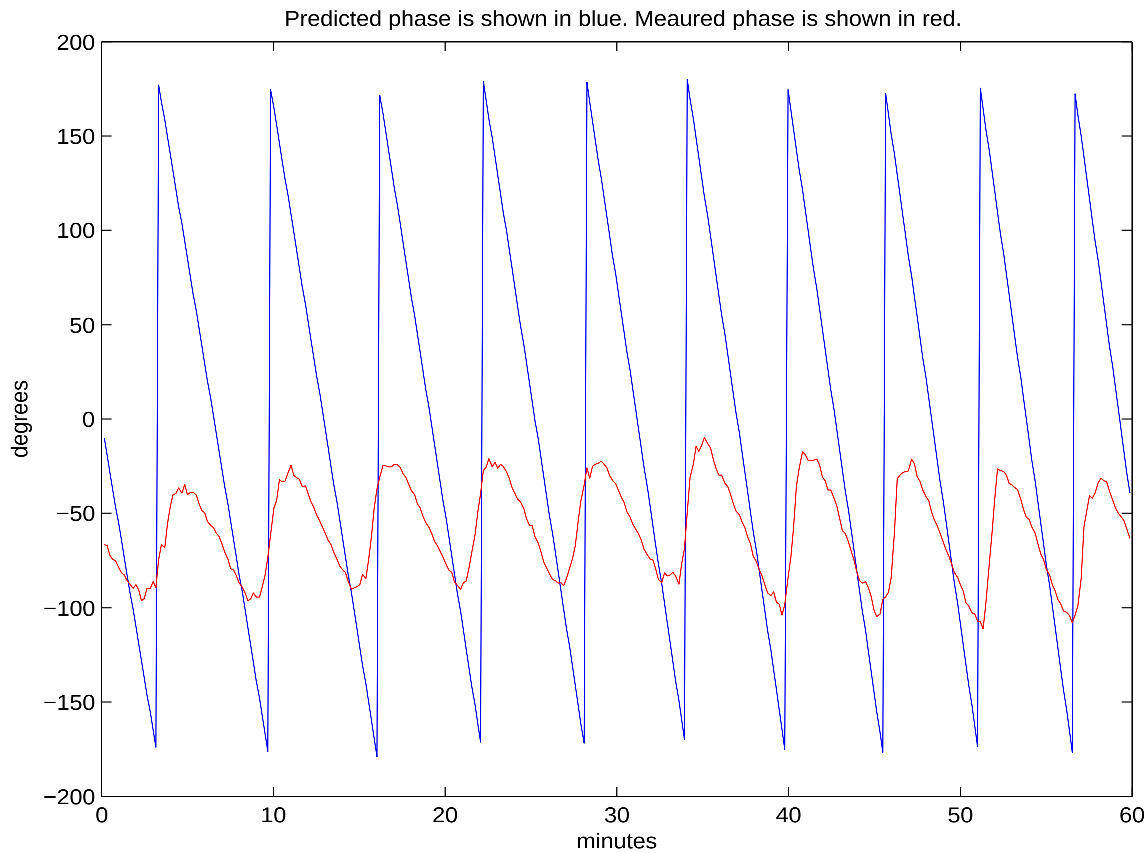

Figure 5.13: Predicted and measured phase for baseline 2,3 over a period of an hour.

channels. Fomalont et. al. state that measured visibilities will not have a considerable DC offset unless there is significant cross talk between the correlator channels [28].

An experiment was performed to quantify the level of cross talk in the VSA. A GPS satellite signal (a high SNR source) was observed with channel 3. The other three channels were terminated with 50 Ohm terminators. The received signal was measured with a spectrum analyzer at an IF of 16 MHz. The signal from channel 3 is shown in Figure 5.16. The figure shows that the peak power is approximately 0 dBm. Figure 5.17 shows the spectrum from channel 2. The figure shows the same spectrum reduced in power by approximately 32 dBm. The problem with cross talk in the VSA is discussed in more detail in Appendix C.

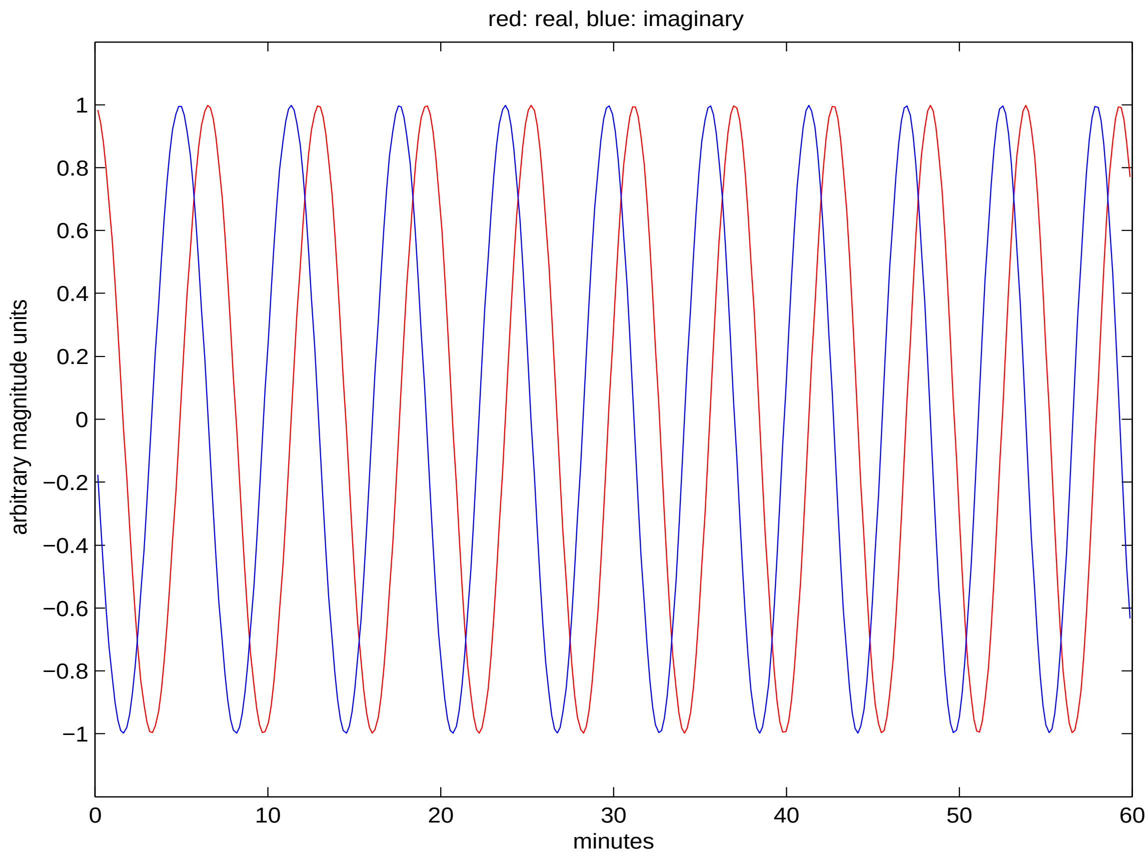

Figure 5.14: Predicted real and imaginary correlator outputs for baseline 3,4.

5.4.6 DC Offset Calibration

As part of the calibration of the data from the VSA, the offset in both the real and imaginary channels is estimated and removed. With the offset removed the measured phase data shows a closer match to prediction as seen in Figure 5.18. The magnitude data (not shown) looks like a constant value corrupted to some extent by noise, which also matches the predicted value.

As a final example of editing, consider the measured data shown in Figure 5.19. The figure shows that the first five minutes of data contains some corruption. The first five minutes should be edited out before using this data in an image.

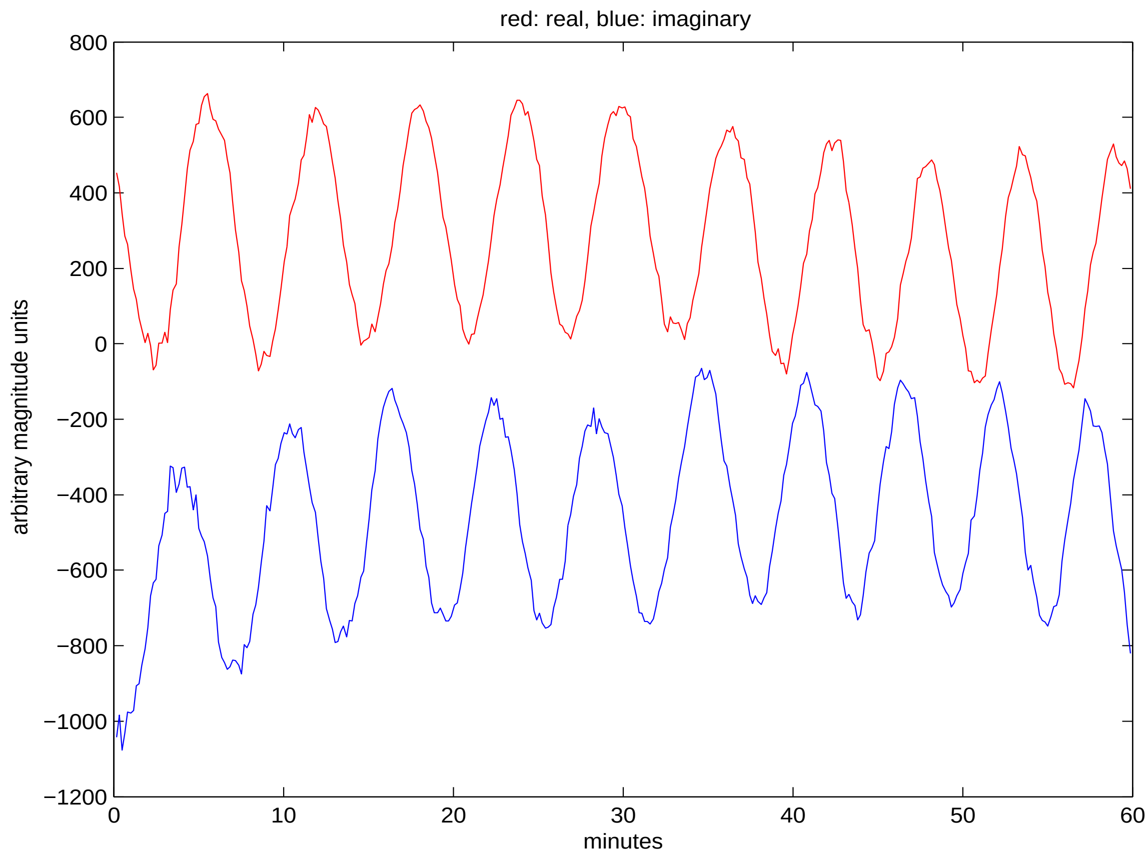

Figure 5.15: Measured real and imaginary correlator outputs for baseline 3,4.

5.4.7 Phase Offset Calibration

From Figure 5.19, one can observe there is a phase offset between the measured and predicted phase. For this data set, the phase offset is approximately -120 degrees. As a final step of calibration, this offset must be estimated (by observing a calibrator) and removed. Figure 5.20 shows the results after calibrating for phase offset.

5.5 An Image of Cassiopeia A

With the data edited and calibrated, an image can be generated using the methods described in Section 2.4. Data sets from two observations were used to generate a dirty image. Observation 1 was taken on April 27, 2005 at a center frequency of 1418 MHz. The observing bandwidth was 1 MHz and the integration time

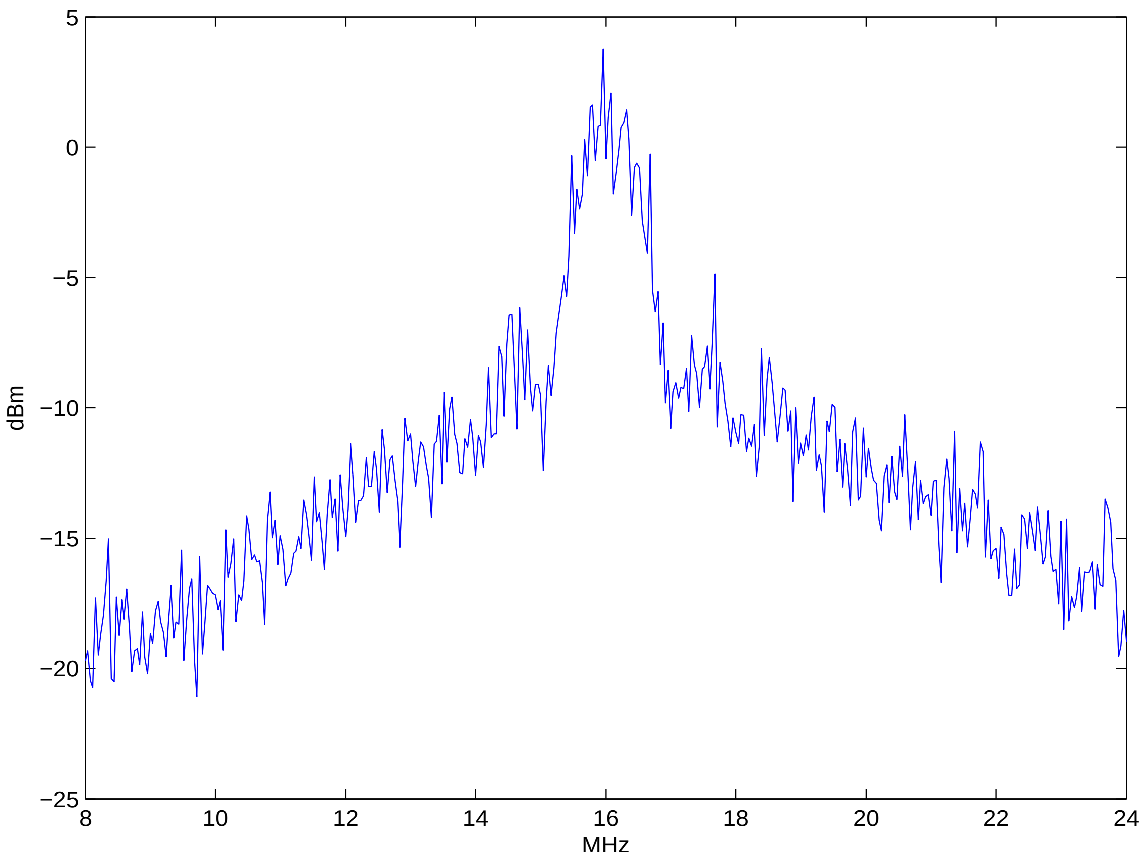

Figure 5.16: A GPS satellite spectrum from channel 3 of the Walkenhorst/Hansen receiver.

was 10 seconds. The hour angle for observation 1 spans (approximately) $H_0 = -7h$ to $H_0 = 1h$.

Observation 2 was taken on May 18, 2005 at a center frequency of 1308 MHz. Like observation 1, this observation was performed at an observing bandwidth of 1 MHz and an integration time of 10 seconds. The hour angle for observation 2 spans (approximately) $H_0 = 2h$ to $H_0 = 7h$.

The first dirty image generated with the VSA is shown in Figure 5.23. The uv coverage and the synthesized beam for the dirty image are shown in Figures 5.21 and 5.22, respectively. In Figure 5.21, the measured visibility locations are shown in red, while their complex conjugate locations are shown in blue.

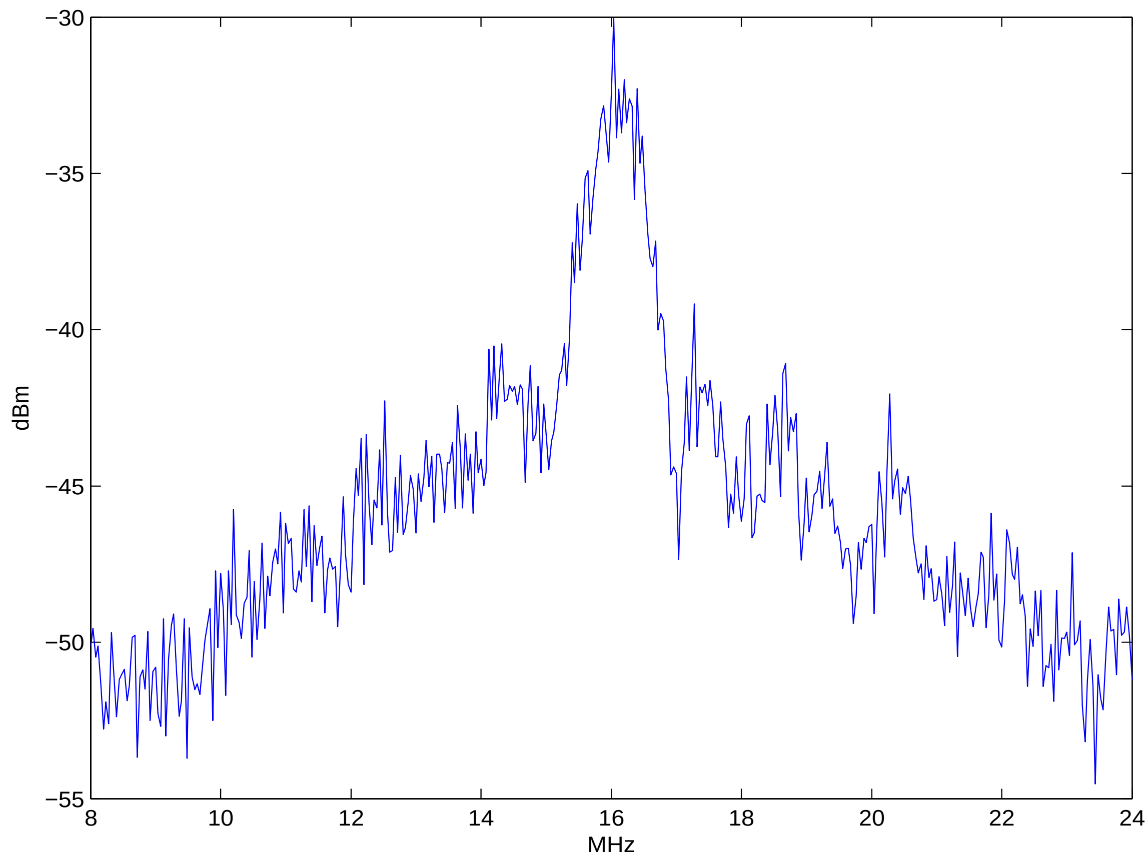

Figure 5.17: A GPS satellite spectrum from channel 2 of the Walkenhorst/Hansen receiver.

Some insight can be gained by examining the uv coverage. The number of uv tracks is six, corresponding to the six distinct baselines. Also, looking at the outer uv tracks, one can see that some of the clumps of visibilities are much smaller than their neighbors. While on other tracks, the neighboring clumps of visibilities are similar in size. This effect is due to editing out portions of the data that were corrupt. The short clumps correspond to data sets that have been heavily edited, while the long clumps correspond to sets with little or no editing.

The dirty image shown in Figure 5.23 was generated using cell averaging followed by a two dimensional FFT. The cell averaging provides some smoothing of the sidelobes, which would be more pronounced if a direct Fourier transform had been

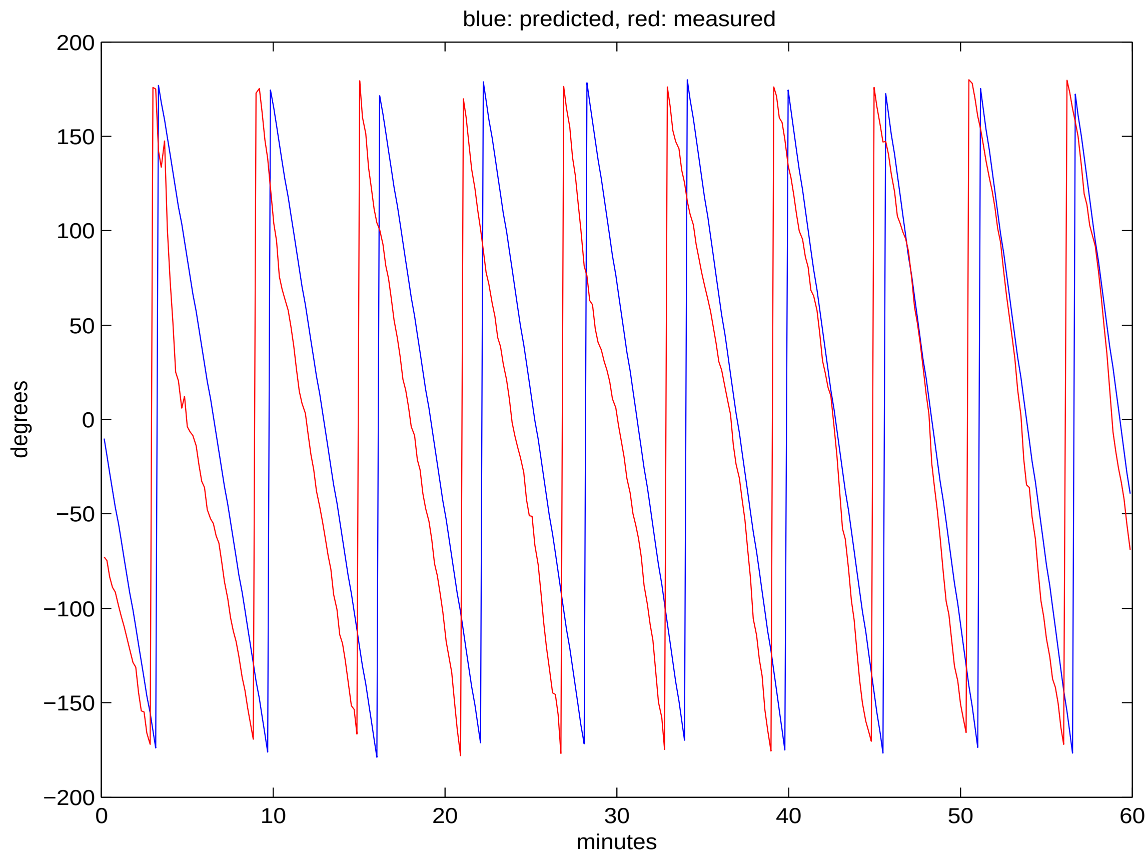

Figure 5.18: Predicted and measured phase for baseline 3,4 with DC offset removed.

used. Like most of the synthesized images in this thesis, the maximum value of the images have been normalized to have a magnitude of one.

Comparing the dirty beam and the dirty image shows that there are visible difference between the image and the beam. Given that Cassiopeia A is a point source for the VSA, the synthesized beam and the dirty image should not be noticeably different. The differences between beam and image are most likely due to calibration error in the positions of the antennas, pointing error, and the cross channel coupling discussed in Section 5.4.

This chapter has shown that, although the VSA has its limits and problems, it can make images of astronomical sources. The resolution in L band is limited

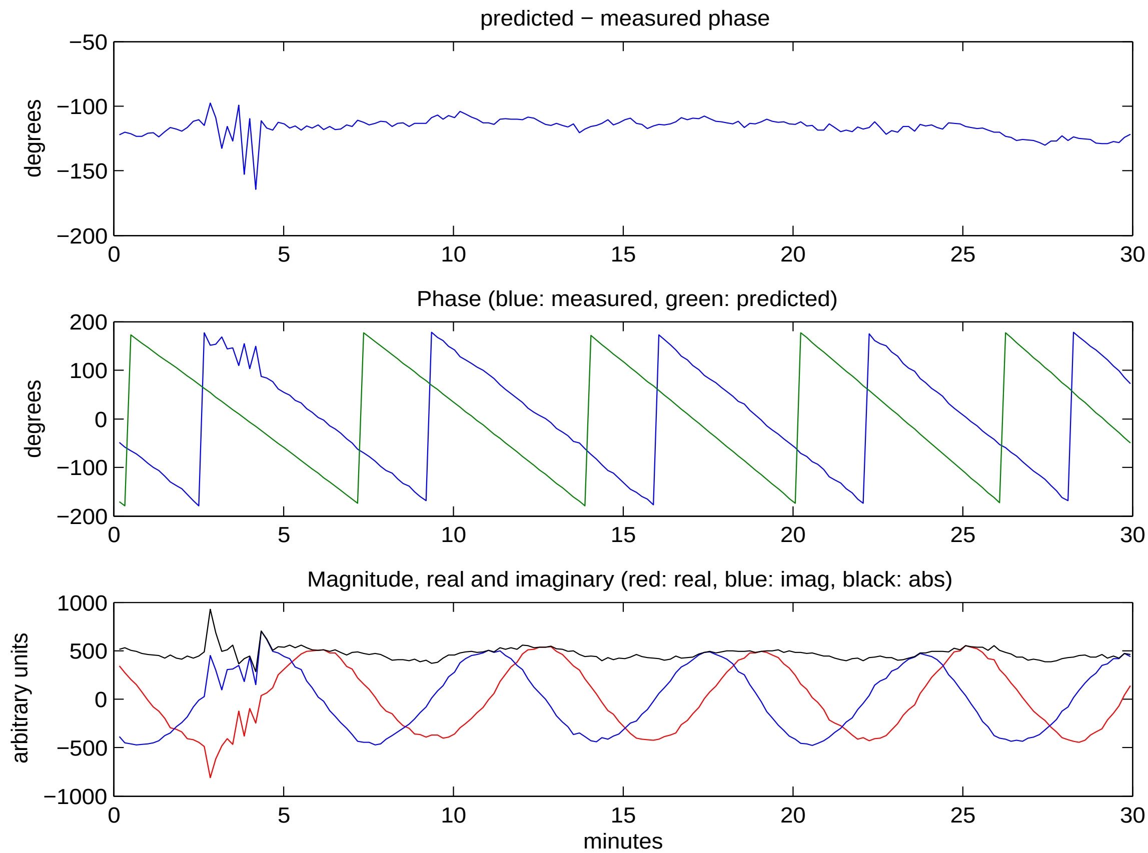

Figure 5.19: Unedited cross correlation data for baseline 1,3.

to about 10 arcmin (see Section 4.2), so most interesting phenomena in L band are unresolved.

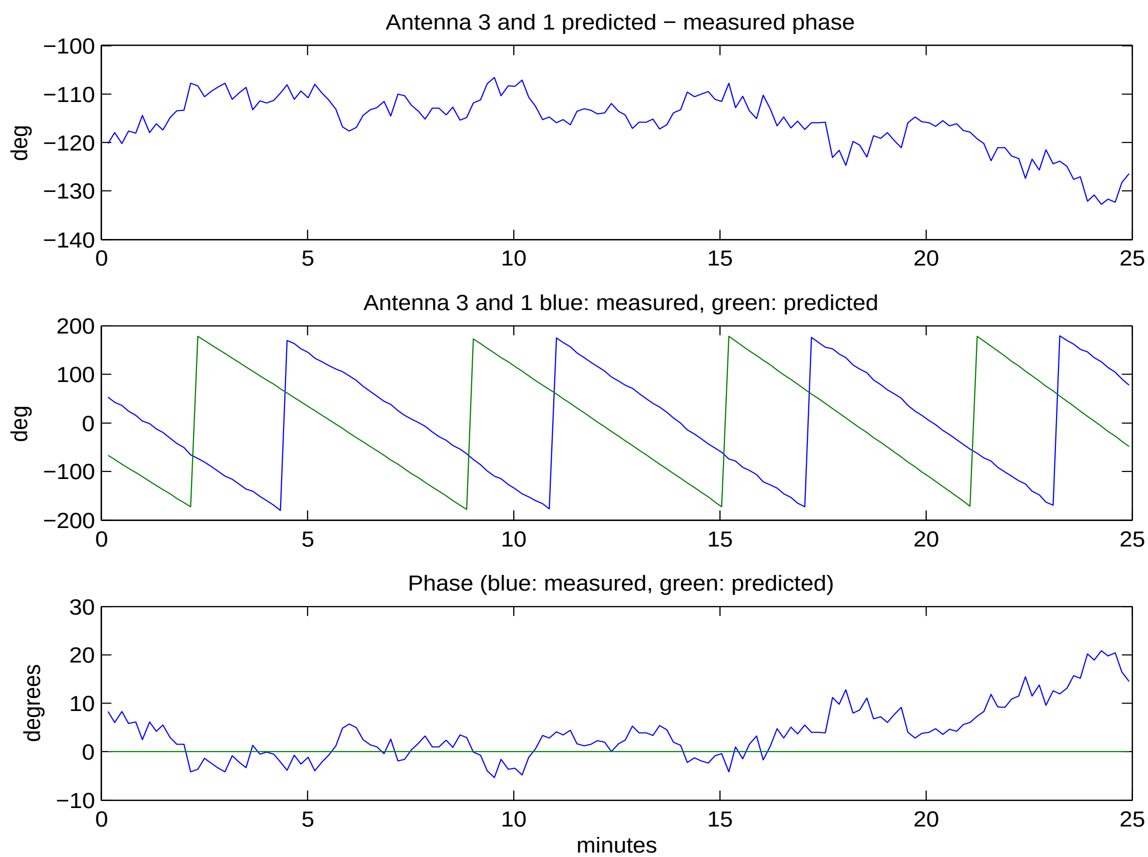

Figure 5.20: Effects of phase offset calibration for baseline 1,3.

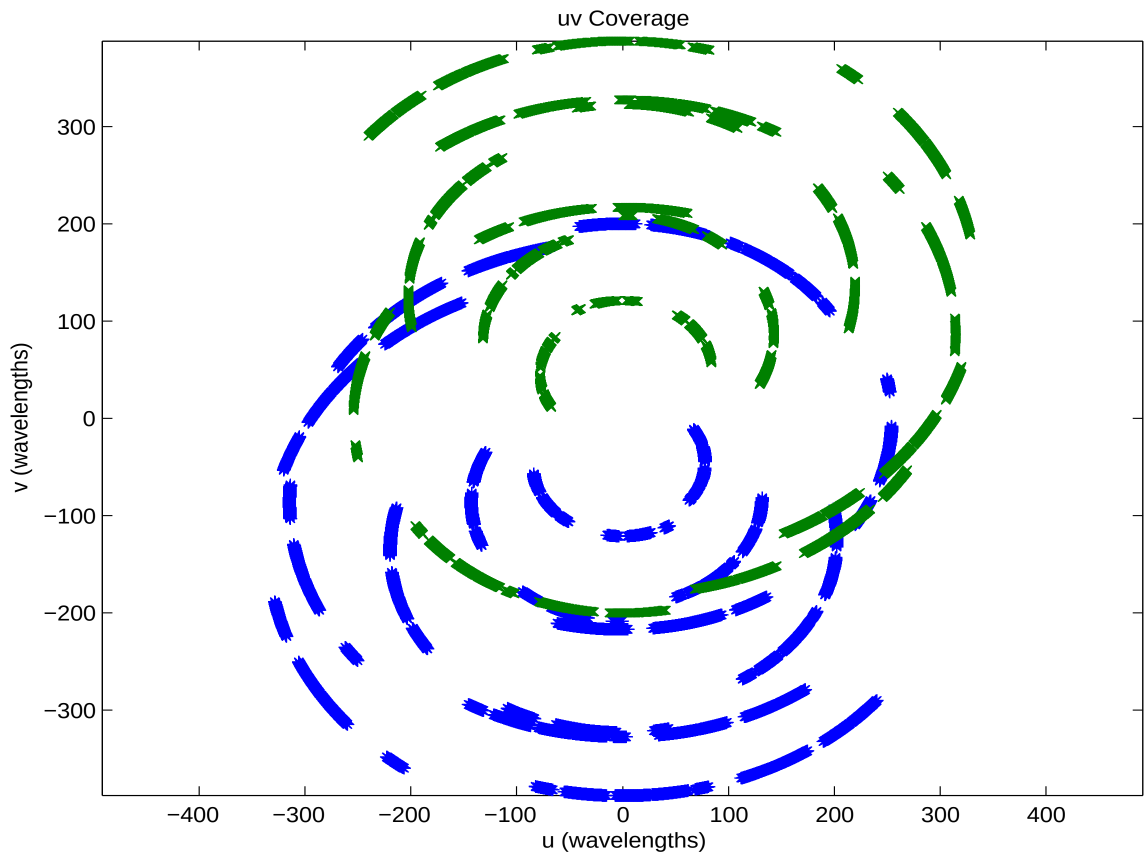

Figure 5.21: uv plane coverage for the dirty image in Figure 5.23.

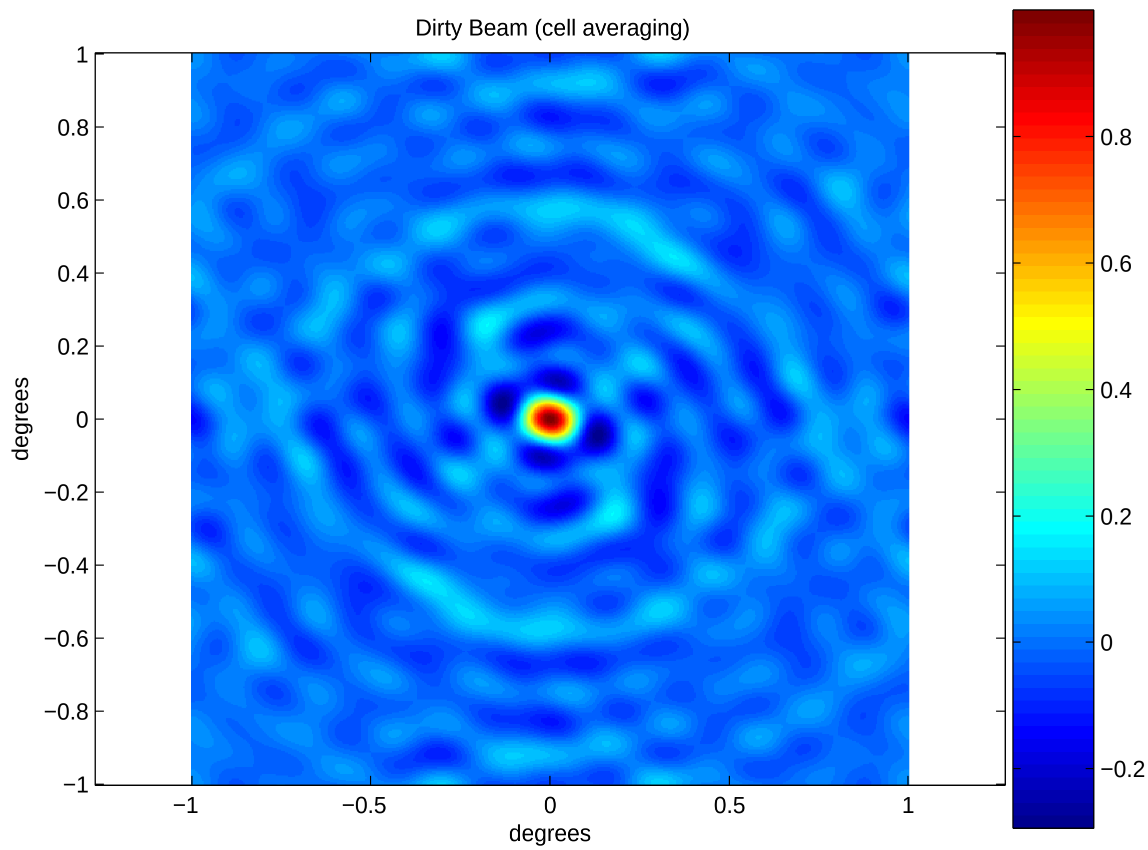

Figure 5.22: Dirty beam for the uv coverage in Figure 5.21.

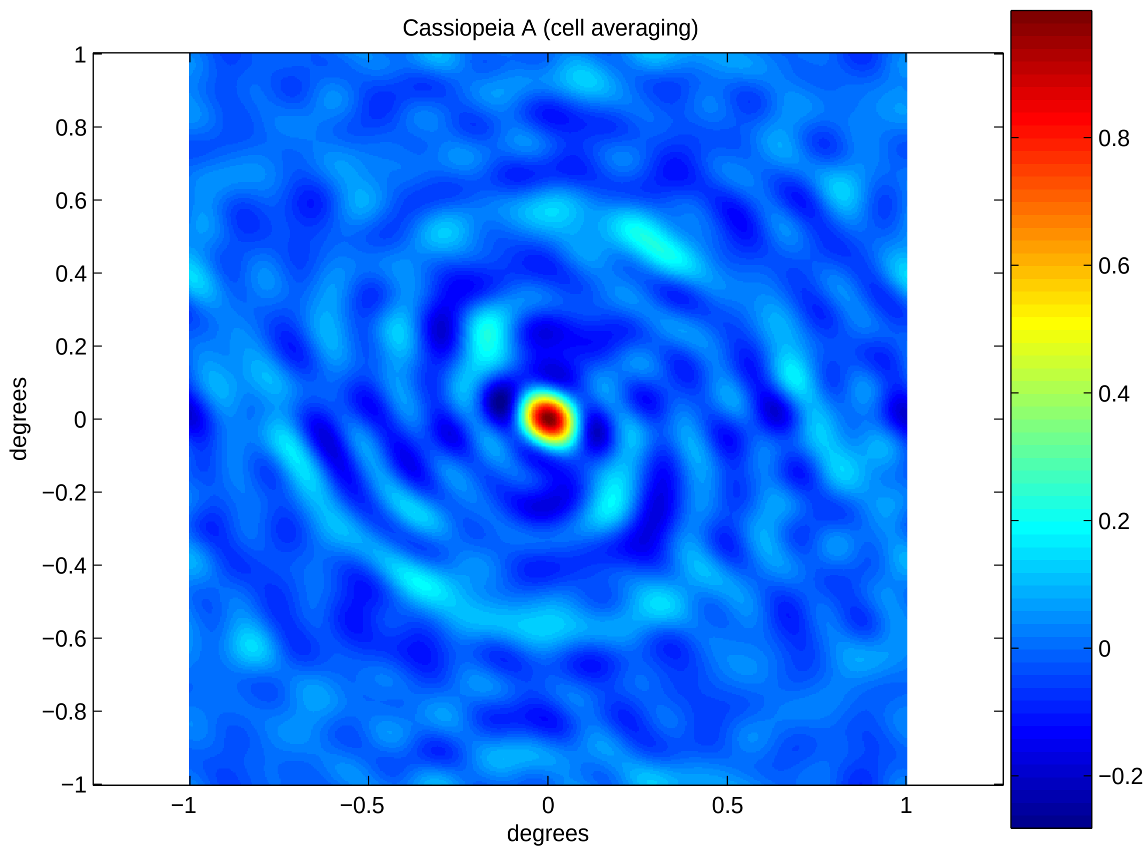

Figure 5.23: Dirty image of Cassiopeia A.

Chapter 6

RFI Experiments

6.1 Introduction

Now that the VSA is up and running, it should fulfill the purpose of its creation: RFI mitigation. As described in Chapter 1, much of the research for combating RFI in synthesis imaging is based on subspace projection. This chapter discusses a projection algorithm called cross subspace projection (CSP) [9] and describes some simulations performed to examine CSP's performance.

The VSA is not ready to implement CSP at the time of this writing. Five receiver channels are required for CSP and currently the VSA has only four channels. Our research group is working on increasing the number of receiver channels (see Section 6.4).

This chapter is structured as follows. Section 6.2 is a review of subspace projection and its use in synthesis imaging RFI mitigation. Subspace methods are familiar territory in signal processing and the reader is advised to consult [9], [29], and [30] for more detail. Section 6.3 describes the CSP algorithm. Section 6.4 describes an RFI scenario and a simulation using the CSP algorithm.

6.2 Subspace Projection

Subspace projection algorithms are based on eigenvector decomposition of correlation matrices. As shown in Section 4.3, particularly in Equation 4.9, the correlation matrices output by a synthesis array can be written as a sum of a noise correlation matrix R_η and a signal correlation matrix R_s, in other words $R = R_s + R_\eta$. For synthesis arrays, R is estimated with integration time on the order of 1 to 10 seconds. If the model of Section 4.3 is augmented to include interference, the signal vector, $\mathbf{x}$, becomes

$$\mathbf{x}[n] = \mathbf{s}[n] + \mathbf{i}[n] + \eta[n], \tag{6.1}$$

and the correlation matrix R is now written as

$$R = R_s + R_i + R_\eta. \tag{6.2}$$

The basis of subspace projection interference mitigation is constructing a projection operator P that is orthogonal to R_i, in other words $PR_i = 0$. Ideally, When P is applied to R, the filtered correlation matrix R_f becomes

$$\begin{aligned} R_f &= PRP^H \\ &= PR_sP^H + PRP^H + PR_\eta P^H \\ &= R_s + R_\eta. \end{aligned} \tag{6.3}$$

Equation 6.3 states that, ideally, the projection operator eliminates the effects of the interference, and leaves the signal and noise unmodified.

In subspace projection, the projection operator is chosen based on an eigenvector (subspace) decomposition of R. If R is estimated over a short time integration period (on the order of 1-100 msec), the astronomical signal of interest will be below the noise. In that case, an eigenvalue decomposition with the eigenvalues ordered from largest to smallest results in

$$\begin{aligned} R_k &= U\Lambda U^H \\ &= \begin{bmatrix} U_i & U_s \end{bmatrix} \begin{bmatrix} \Lambda_i & 0 \\ 0 & \Lambda_s \end{bmatrix} \begin{bmatrix} U_i^H \\ U_s^H \end{bmatrix}, \end{aligned} \tag{6.4}$$

where U_i and U_s are orthogonal subspaces called the interference subspace and signal subspace, respectively.

The eigenvalues are separated into the different subspaces based on their magnitude. The eigenvalues in the signal subspace have magnitudes approximately equal to the noise power. The eigenvalues in the interference subspace are larger than the noise power.

Typically, in signal processing, U_s is referred to as the noise subspace. Since astronomical signals typically have SNR << 0 on the short term integration time scale, the noise subspace is where the astronomical signal resides [30].

For subspace projection, the short term integration projection operator is chosen as

$$P_k = I - U_i U_i^H = U_s U_s^H,$$

and is applied as in Equation 6.3.

6.3 Cross Subspace Projection

Cross subspace projection builds on subspace projection, but the CSP projection operator is chosen differently from traditional subspace projection. CSP requires the use of one or more auxiliary reference antennas that are used to track and receive interference, one auxiliary for each interferer.

In CSP, the antennas of the synthesis array are referred to as the primary antennas. The signals from the auxiliaries and the signals from the primaries are used to estimate the cross correlation between the primaries and the auxiliaries as

$$R_{pa} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{x}_p \mathbf{x}_a^H.$$

The projection operator is then chosen using a singular value decomposition of the short time integration $R_{k,pa}$ as follows

$$\begin{aligned} R_{k,pa} &= U \Gamma V^H, \\ U_i &= [u_1 u_2 \ldots u_Q], \\ U_s &= [u_{Q+1} \ldots u_{M_p}], \\ P_k^{CSP} &= I - U_i U_i^H = U_s U_s^H. \end{aligned} \tag{6.5}$$

In Equation 6.5, Q is the number of interferers present, M_p is the number of primary antennas, and the singular values of Γ are ordered from largest to smallest [9].

CSP out performs traditional subspace projection in eliminating interference. This is because the auxiliary antennas have high SNR estimates of the interferers. These high SNR estimates provide a more accurate estimate of the interference subspace.

6.4 Simulation of CSP for the VSA

Simulations were conducted implementing CSP for the VSA. The interferer is an orbiting GPS satellite. The simulated astronomical sources are three points sources with equal flux located at the right ascension and declination of Cassiopeia A. The auxiliary antenna is located at $(25, -40, 4)$ meters East, North, and toward Zenith from the phase reference antenna (the auxiliary antenna is on the fifth floor roof of the Clyde Building).

The signal model used for this simulation is the model of Equations 6.1 and 6.2. The interference in the primary antennas is multiplied by a simulated antenna gain pattern based on the uniformly illuminated circular aperture approximation. The reference antenna tracks the interferer, and therefore has a high SNR copy of the interfering signal.

Figure 6.1 shows an image of the three simulated sources with no interference contaminating the image. The sources each have a flux density of 500 Jy, giving an instantaneous SNR of -22 dB. The observing frequency is 1575.42 MHz. The phase tracking center for this image is the right ascension and declination of Cassiopeia A. The image is made over a duration of 16 hours at 300 equal intervals (a visibility is measured approximately every 3 minutes). The simulated integration time is 10 seconds and the short term integration time is 0.1 seconds.

Figure 6.2 shows an image of the same sources that has been corrupted by the GPS satellite signal. The interferer has washed out the astronomical sources. Figure 6.3 shows the angular separation between the interferer and the phase tracking center. The interferer is impinging on the synthesis array for 128 minutes, which is the sky transit time for this particular GPS satellite. Figure 6.4 shows the interference to noise ratio at the feeds of the synthesis arrays during the 128 minutes.

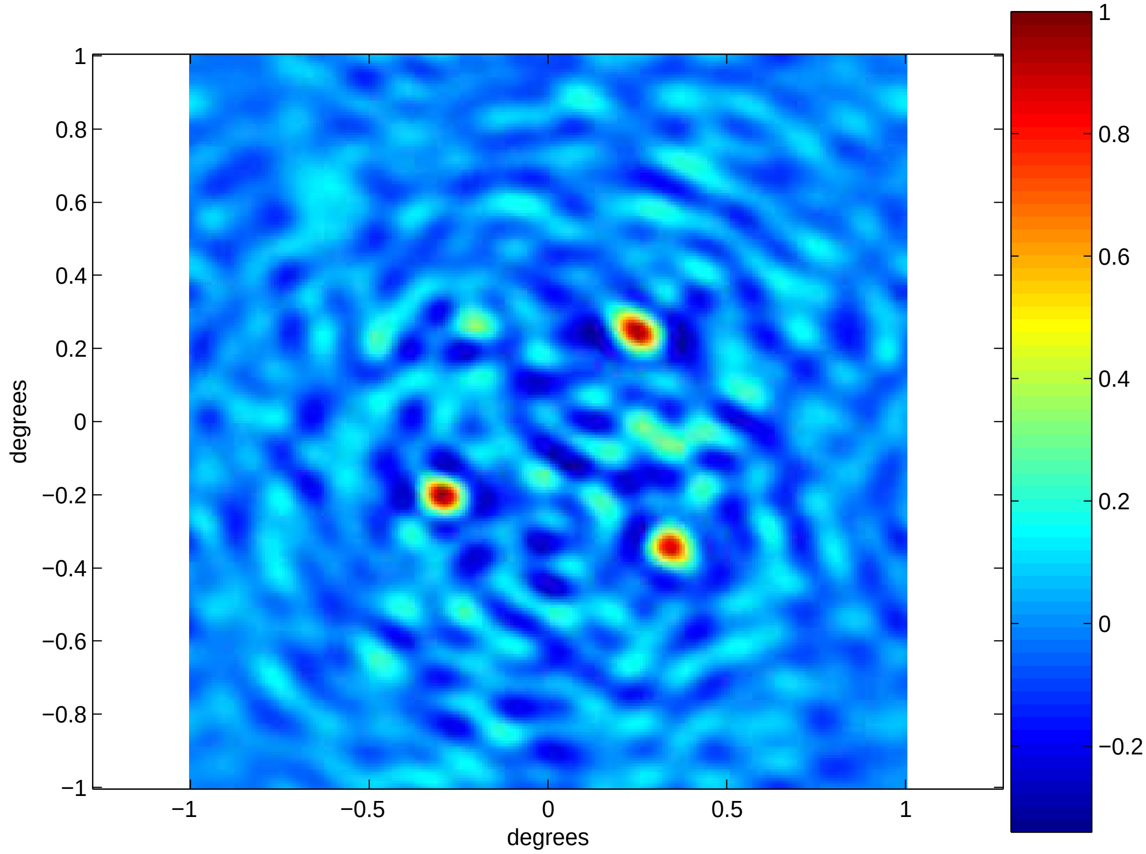

Figure 6.1: Interference-free image of three equal power point sources.

Using the CSP algorithm on the interference corrupted data results in the image shown in Figure 6.5. Comparing the CSP filtered image with the interference free image, one finds that the effects of the interference have been removed.

Implementing this algorithm on the VSA is not feasible at this time. The Walkenhorst/Hansen receiver and the DSP have only four channels. However, work has already begun in our research group on new receiver and data acquisition systems. The new systems will have 24 channels to receive and acquire data. Once these systems are in place, one of the antennas on the fifth floor can be used as an auxiliary antenna to test the CSP algorithm on real data.

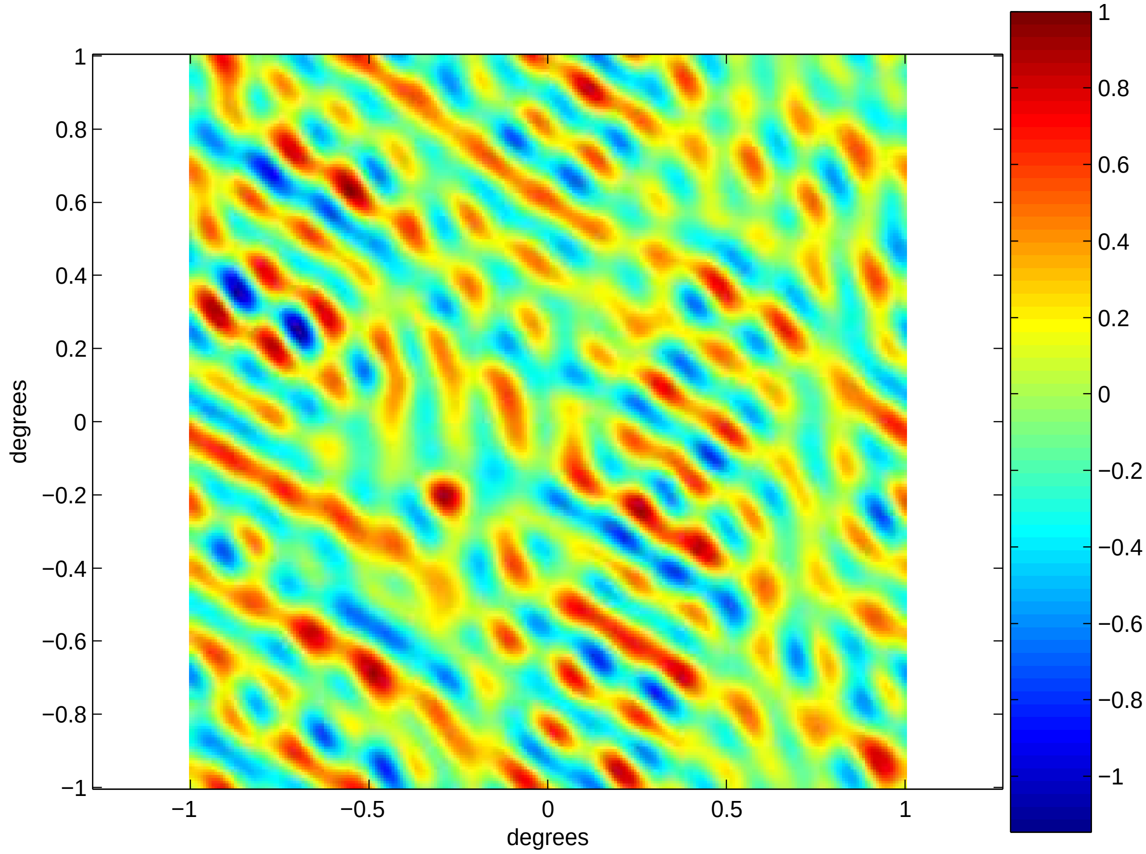

Figure 6.2: Interference corrupted image of three equal power point sources.

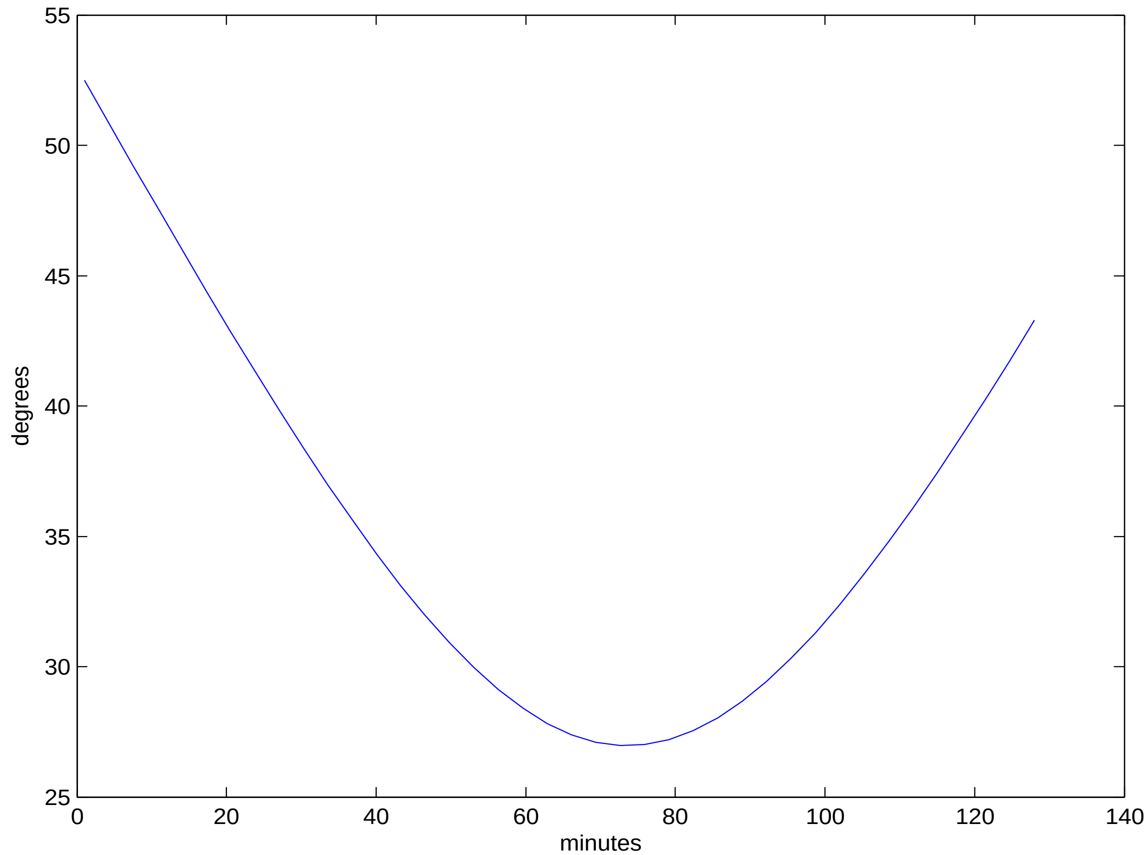

Figure 6.3: The angular separation between the phase tracking center and the interferer.

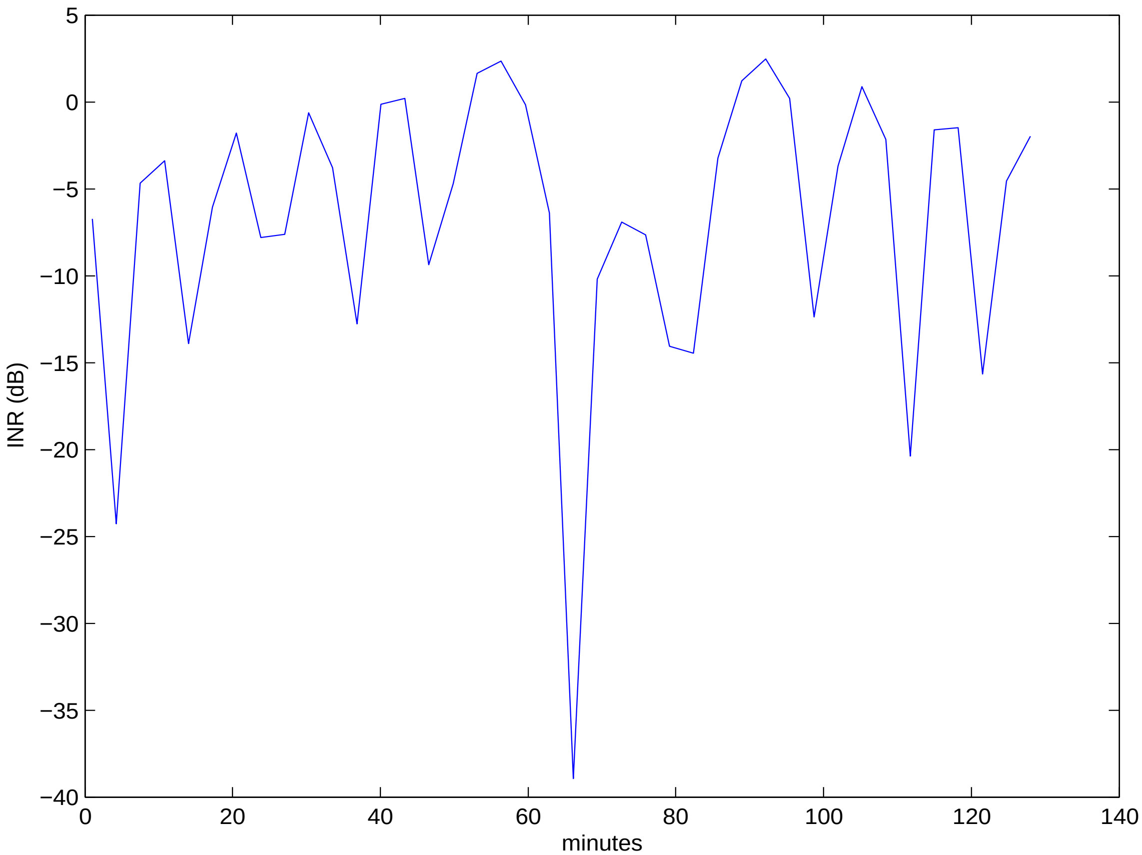

Figure 6.4: Interference to noise ratio in the feeds of the synthesis array antennas.

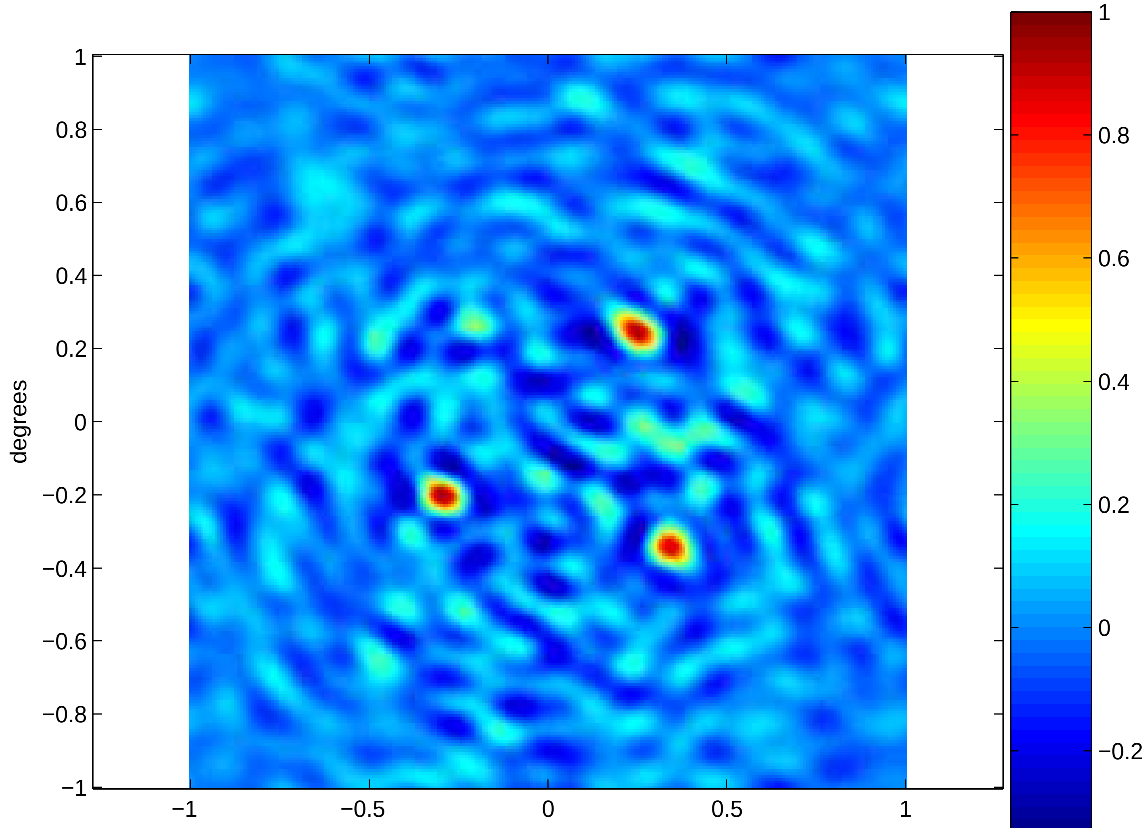

Figure 6.5: An image filtered using CSP.

Chapter 7

Conclusions and Future Work

This chapter provides additional ideas for future experiments to be performed with the new array, some necessary improvements to the array, and some conclusions.

7.1 Imaging a high SNR Source

An experiment that should be run is to image an orbiting GPS satellite. While of little interest to the radio astronomy community, this experiment would help iron out the kinks in the VSA (such as the absolute amplitude calibration). This experiment can be done at any time of the day because there are always 7 or 8 GPS satellites in the sky above Provo. GPS satellites sweep out their arcs more quickly than astronomical phenomena so this experiment could be run more than once a day if necessary.

7.2 Imaging Other Astronomical Phenomena

Another experiment that should be performed with the VSA is to image Cygnus A. Cygnus A emits continuum radiation in L-band at approximately the same power levels as Cassiopeia A. Our NRAO collaborators suggested the VSA be tested on Crab Nebula and Virgo A as well . These are lower power sources than Cassiopeia A and Cygnus A. The Crab Nebula is 3 times weaker and Virgo A is 10 times weaker than Cygnus A. [1]

[1] These sources were suggested by Dr. Rick Fisher at NRAO. Dr. Fisher also provided the information on the flux of these sources.

The Crab Nebula is approximately 18 arcmin in diameter and should be resolved by the VSA. The sun and moon should also be resolved by the VSA. Both are approximately 30 arcmin in diameter.

7.3 RFI Scenarios

There are numerous RFI sources and scenarios under which to image the sources listed above. A small number are covered in the paragraphs that follow.

There is a dipole antenna mounted on the North tower on top of the Clyde building. This antenna can easily be connected to a signal generator and can provide an RFI source at any time, at any frequency in the the Walkenhorst/Hansen receiver band, at any power level, and for any duration. This is a controllable RFI source that can be used to test RFI mitigation algorithms for conceivably any imaging source and scenario.

Another experiment would be to image an orbiting satellite, say GPS or GLONASS, and use another satellite as an interferer. As mentioned before, there are always an abundance of GPS satellites in the sky, and three to five GLONASS satellites in the sky above Provo. The GLONASS satellites use different operating frequencies around 1600 MHz (see [31]) and all GPS satellites transmit at 1575.42 MHz. This is a good experiment to work out the bugs in the array and to test RFI algorithms.

An exciting experiment would be to image an astronomical source that emits continuum radiation, like Cassiopeia A and Cygnus A, at or near the frequency broadcast by GLONASS. As mentioned before, there are usually only three to five GLONASS satellites in the sky at any one time. As such, it is probable that one satellite could be chosen which crosses the sky near a astronomical source of interest, without having to worry about the other visible satellites as a source of interference. The observing frequency could be tuned to provide high or low levels of interference. For high levels of interference, the observing frequency should be set right at the operating frequency of the satellite. For lower levels of interference, the observing frequency could be tuned to the sidelobes of the satellite.

The experiment described in the previous paragraph could also be carried out using a GPS satellite. However, extra care must be taken. There are typically 7 or more GPS satellites in the sky above Provo at any time and all of them use the same RF. This could be a help, if more RFI is desired, or a hinderance, if one desires to limit the sources of RFI.

7.4 Improvements to the Array

This subsection is devoted to describing some of the improvements that should be made to the array: a new receiver, a new data acquisition system, and a fifth antenna. Other improvements (not listed in this section) are discussed in Appendix C

The first improvement to the array is already underway (as mentioned in Section 6.4): the number of receiving and data acquisition channels of the array are being increased. The author is pleased that the receiver is currently being redesigned. A side effect of this new receiver is increased channel isolation, a problem with the current receiver system.

With one more receiver channel, a fifth antenna could be added to the array. A fifth antenna placed between antennas 1 and 2 would make the sampling of the uv plane more even and thereby lower the sidelobe levels in the synthesized beam. A fifth antenna also increases the number of independent baselines from six to ten.

Extra channels and more data acquisition will also allow for testing of mitigation algorithms that require one or more auxiliary antennas. With the current 4 channel system, using one auxiliary would require that one of the synthesis antennas be used as an auxiliary. This is undesirable as it reduces the number of independent baselines to three.

Another recommendation is to increase the observing frequency of the VSA. With higher observing frequencies (for example, 5 or 8 GHz), the baselines of the VSA become larger and the resolution of the array is increased. With higher resolution, the VSA will be able to resolve astronomical sources and display structure in an image.

7.5 Conclusions

The results of Chapter 5 show that the VSA is capable of making radio synthesis images. The array is in place and equipped with software for data analysis and image generation. Some fixes and improvements to the VSA will make it a versatile tool for imaging and for RFI mitigation experiments.

APPENDIX

Appendix A

Operating Procedures and Guidelines

This appendix provides array operating procedures and general guidelines for operating the Very Small Array (VSA) synthesis imaging expansion. It is provided as a resource for those who will use the VSA, as such the tone is somewhat informal.

A.1 Outdoor Equipment

In the feed of each antenna, there are two Mini-Circuits ZEL-1217LN LNAs as shown in Figure A.1. These LNAs run on 15 VDC. One of the small grey cables running out to each antenna provides this 15 V.

For each of the four antennas there are two positioners: one for azimuth and one for elevation. Figures A.2 and A.3 show photographs of the azimuth and elevation positioners for antenna 3. These are custom positioners built by M^2 Inc. The positioners are guaranteed to have 0.5 degrees of pointing accuracy. Each positioner has one black cable running out to it. The black cable connects to the controller boxes located in our lab. The black cable contains two power cables and two sense lines. For more information see the M^2 manuals.

There are coaxial signal cables coming from each antenna. These are thick black cables labelled with PE-B3199. Each signal cable is split into two 210 ft section. Each antenna also has a LNA repeater box that connects the two sections of signal cable. The repeater box contains one Mini-Circuits ZEL-1217LN. The other small grey cable coming from the repeater boxes into the building provides the 15 VDC for the repeater box. Figure 5.6 shows a photograph of the controller boxes and the

Figure A.1: The LNAs in the feed of antenna 2.

Figure A.2: The azimuth positioner for antenna 3.

Figure A.3: The elevation positioner for antenna 3.

LNA/repeater power supplies. Figure A.4 shows a photograph of the repeater box for Antenna 2.

The cables heading out to antenna 2 are shown in Figure A.5. All of the cables from the four antennas enter the building through an outdoor hatch shown in Figure A.6.

A.2 Equipment Check

It is important that you regularly check the functionality of the equipment that makes up the VSA. A good way to do this is to observe an orbiting GPS satellite with each antenna. Use the Hansen receiver to down convert the signal and observe the spectrum on the spectrum analyzer. This kind of observation checks the signal path integrity from the the LNAs through to the final baseband low pass filter. The spectrum from each antenna should look similar in structure and be at similar power levels.

Figure A.4: The repeater box for antenna 2.

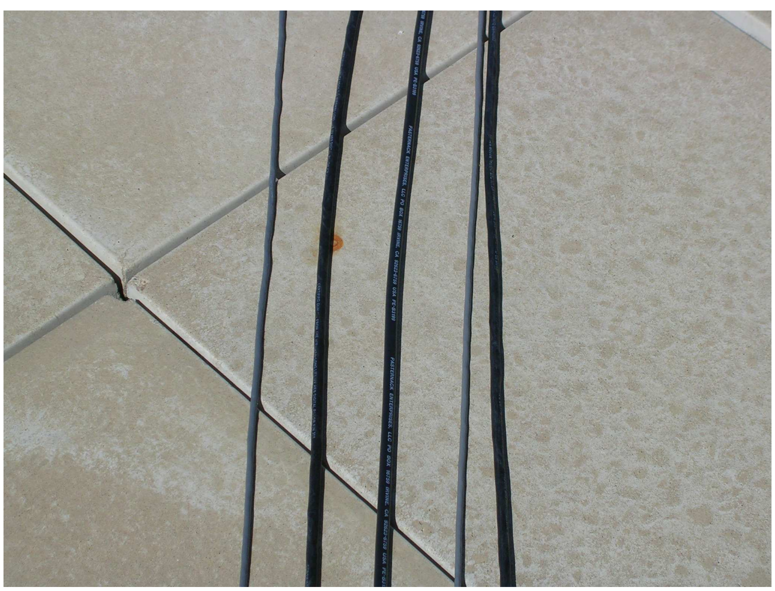

Figure A.5: The cables for antenna 2.

Figure A.6: Antenna cables enter the building.

A.3 Pointing Calibration

At least once a month, inspect the pointing calibration of the dishes. Also, inspect the antennas and their pointing after wind storms. Elevation pointing is easily inspected with a level. Check the azimuth by pointing at the Southwest corner of the Carillon Bell Tower. (Do NOT perform this check on antenna four without a safety harness.) When the dishes are pointed at the bell tower, the controller box readings should be as follows.

- Antenna 1: 5.6881 degrees East of North
- Antenna 2: 0.8150 degrees East of North
- Antenna 3: 6.2529 degrees East of North
- Antenna 4: 5.6493 degrees East of North

The measurements made with the surveying total station are shown in Table A.1. V is the vertical angle, in degrees. V increases downward, with 90° being parallel to the ground. HR is the horizontal angle, in degrees, and R is the range in feet. The

Table A.1: Results of surveying.

Antenna	Bearing/Range	Measurement 1	Measurement 2
A2	V	89°59′49”	89°59′53”
	HR	359°59′52”	359°59′52”
	R	177.940 ft	177.965 ft
A3	V	90°02′15”	90°02′20”
	HR	97°37′12”	97°37′35”
	R	151.945 ft	151.940 ft
A4	V	89°58′39”	89°58′34”
	HR	90°04′47”	90°04′47”
	R	237.930 ft	237.930 ft
Bell Tower	HR	276°06′56”	NA

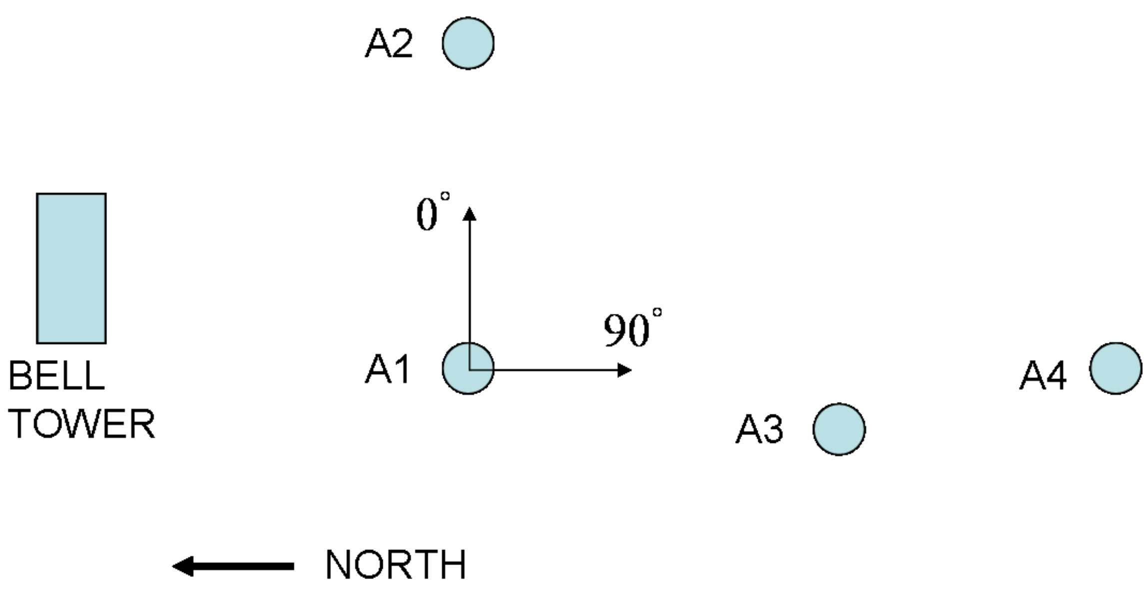

Figure A.7: The coordinate system for surveying.

Table A.2: Weather thresholds for the VSA.

Weather Condition	Threshold
steady wind	wind speed > 6 mph
gusting wind	wind speed > 11 mph
precipitation	any

reference coordinate system for Table A.1 is shown in Figure A.7. Notice that antenna 1 is at the coordinate origin and antenna 2 is at 0°. The relative positions of objects in Figure A.7 are only approximate and distances are not to scale.

A.4 Weather Conditions

It is important that you do not run the array, i.e. steer the antennas, in adverse weather conditions. Calm weather is a must for using the VSA. Table A.2 lists guidelines for quantifying weather that is not calm. As a general rule, shut off the power to all of the array equipment during bad weather.

Don't run the array when there is precipitation of any kind. Operating the array during rain, snow, hail, etc runs the risk of exposing the electronics to water, which is inherently unhealthy for electronics. Also, if there has been precipitation, wait until the equipment is dry before using it again. One day's wait is usually sufficient.

Do not operate the array in strong winds. The satellite dishes act like large sails that catch the wind. This is hard on the motors which have to work harder to move the antennas. This is also hard on the positioner's electronics since more amps are flowing. In addition, strong winds can bend or break the antennas' parts and various supporting structures. If there is a steady wind, the wind speed should be less than 6 mph. If there are gusts, the wind speed should be less than 11 mph.

The software that steers the antennas to track celestial objects will automatically stow the antennas if the wind gust condition is met.

A.5 Using the Pentek DSP

The author hopes that the data acquisition system for the DSP will be migrated away from the Pentek DSP. If you have to use the DSP bear in mind that the DSP is finicky!

You must warm up the DSP before you use it. I have provided a Matlab script that will accomplish this: dspwarmer.m. Before using the script, there should be no input cables on any of the DSP SMA terminals. Turn on the DSP card cage, open the script, set the parameters at the top of the file, and run the script.

Set the DSP code to run for 30 minutes intervals using the Matlab script collectdata.m (see Section B.1). collectdata.m will reset the DSP after each interval. This provides a measure of protection against DSP malfunctions during the time it is acquiring data. (The DSP does malfunction often enough to warrant this.) Only thirty minutes of data is lost. I have written Matlab scripts and batch files to perform these tasks. For further information, see the Appendix B which describes the code for this thesis.

The repeaters should not be used until the DSP is repaired. The DSP crashes when the input power is too high. Having the repeaters on is within the input specifications of the DSP, but the DSP still crashes.

A.6 Checklist

This section provides a checklist of the things you should do before you start the DSP correlator.

1. Check the weather forecast. If there is going to be bad weather, reschedule your observation.

2. Use the atomic clock sync program to set the host PC's internal clock.

3. Warm up the DSP using the Matlab script dspwarmer.m (C:\ti\myprojects\astronomy). This takes about 20 minutes.

4. Inspect the antennas. Look for twisted or taut cable and any kind of damage to the antennas or the positioners.

5. Attach the signal and LO cables to the receiver. Use a high frequency coax for LO1.

6. Turn on the LNA and repeater power supplies (see Figure 5.6). The power supplies should be set to 15 V. When the LNAs and the repeaters are on, they should draw approximately 0.15 amps. If the repeaters are not being used, the LNAs draw about 0.1 amps.

7. Turn on the azimuth and elevation positioner control boxes. The azimuth and elevation control boxes for one antenna must be turned on simultaneously.

8. Turn on the Local Oscillators. Both LOs should have their amplitude set to 0.0 dBm. Set the LO frequencies to receive the proper RF. The LOs are set based on the following equation:

$$\mathrm{LO1} = \mathrm{RF} + \mathrm{LO2} + \mathrm{DSPFC},$$

   where DSPFC is the DSP center frequency. To Illustrate how to set the LOs, consider the following example. A typical setting for the DSP center frequency is 18 MHz. Usually LO2 is set to 822 MHz. In this case, to receive an RF frequency of 1575.42 MHz (GPS operating frequency), you would set LO1 to 2415.42 MHz.

9. Open another instance of Matlab and run controlcenter.m (controlcenter.m is located in D:\MATLAB701\work\Observe). Make sure the serial ports are controlling the right antennas. Make sure the azimuth and elevation controllers are both communicating with the serial ports by changing the azimuth and elevation speeds. If the speeds don't change for a particular antenna, this means that the serial communication is faulty for one of the controllers. Shut off azimuth and elevation control boxes of the antenna that isn't working. Turn off controlcenter.m. Wait five seconds and turn the control boxes on again (once

again, the azimuth and elevation controllers must be turned on simultaneously). Run controlcenter.m again. Test the serial communication again by changing the azimuth and elevation speeds.

10. Input the astronomical right ascension and declination and start tracking the astronomical phenomenon. To track a source with a negative declination, only negate the number in the Deg declination text input box.

11. Set the parameters in wthrtime.txt and run the weather recorder software wthrcdr.exe (both are found in D:\MATLAB701\work\Observe).

12. Turn on the Walkenhorst/Hansen receiver.

13. Press "RF ON" on the LOs.

14. Use the Matlab script collectdata.m (C:\ti\myprojects\astronomy) to run the DSP correlator code. Make sure the LEDs on the front panel indicate that the correlator is running.

15. When your observation is complete, point the antennas at elevation 90° (straight up). Shut down the equipment you turned on. Save your data to some sort of removable media along with a text file describing the observation: start and stop times, observing frequency, source location, and any other pertinent information.

A.7 Antennas Tracking

Figure A.8 shows antenna 4 (Rachel) of the VSA. This picture is taken facing South. Antenna 4 is located on the Southwest corner of the Clyde Building. Antenna 4 is tracking Cassiopeia A in the early morning April 2005. West Mountain can be seen in the background.

Figure A.9 shows a photograph of antenna 3 (in the foreground) and antenna 1 in the background. This photograph was taken facing North. The two antennas are tracking Cassiopeia A in the early morning April 2005.

Figure A.8: Antenna 4 (Rachel) tracking Cassiopeia A in the early morning, April 2005.

Figure A.9: Antenna 3 (Rebekah), in the foreground, and antenna 1 (Eve), in the background, tracking Cassiopeia A in the early morning, April 2005.

Appendix B

VSA Software

I have tried to be a conscientious programmer by putting meaningful comments in my code. I consider the code I have written to be well commented.

B.1 Code for the DSP

This section describes the software found in the directory C:\ti\myprojects\astronomy on the host PC (computer name: ASTRONOMY).

B.1.1 C Source

This appendix gives a brief overview of the source files are used to program the DSP as a four channel correlator. Look at the files for more detail. The source files are organized into a Code Composer Studio workspace called astronomy. I have assumed that the reader has read through section 2.4 of Andrew Poulsen's thesis [8] prior to reading this section. Reading Appendix D might also be helpful.

ProcessorA.c, ProcessorB.c, ProcessorC.c, ProcessorD.c, astronomy.h, DmaTransfer.h, GlobalBufferSize.h, astronomy.cmd

ProcessorA.c, ProcessorB.c, ProcessorC.c, and ProcessorD.c are the main() C source files for each processor of the astronomy workspace. These files create an internal data passing chain and correlator shown in Figure B.1. Data is sampled from the front panel, down converted, and sent to the IO bififos of each processor.

Each IO bififo is configured to fill up a ping-pong buffer in the SBSRAM of its processor. The ping-pong structure allows the processor to work on one part of

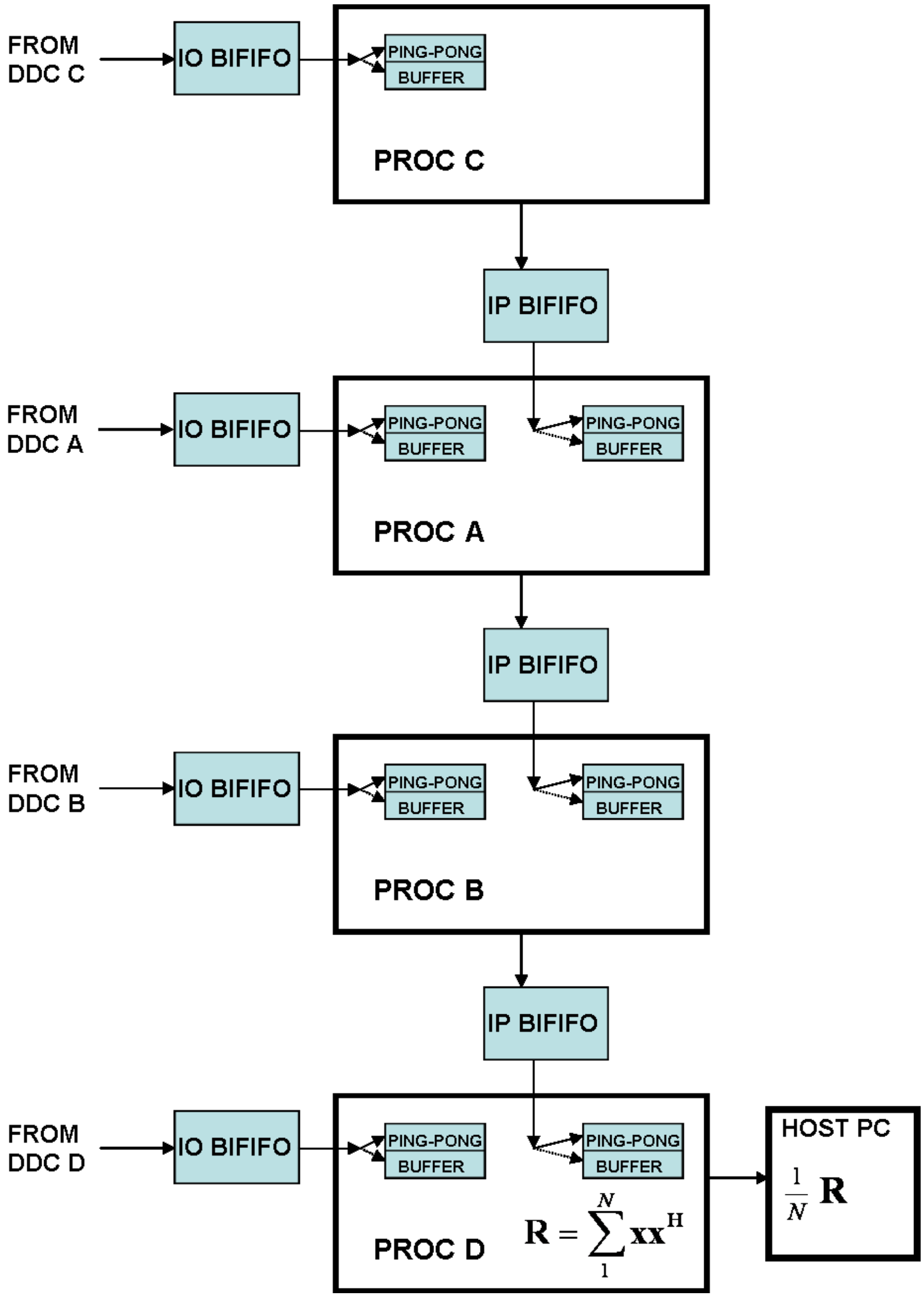

Figure B.1: Inter-processor setup for the correlator. The inter-processor bififo between processors C and D is not shown.

the buffer while the other half is being filled up. The processing must be completed before the other buffer is full.

Data is passed from processor C to A to B to D. Processor D is where the correlations are computed and sent to the host PC. The host PC software is responsible for the $1/N$ normalization.

The astronomy.cmd file tells the CCS linker how to place code and data into the memory available to the microprocessors. The header files astronomy.h, GlobalBufferSize.h, and DmaTransfer.h, contain important parameters, flags, and information for creating the four channel correlator. See each header file for more detail.

Tunable.h

This header file contains parameters that are changed regularly. These include the program run time, the length of time that correlations are estimated, the processing bandwidth, and the DSP center frequency.

CorrMatrix.c, CorrMatrix.h

These files contain the functions that compute the correlations. These functions are optimized for speed.

LoopTimimg.c, LoopTiming.h

These files contain a function to compute loop counters based on input parameters such as the program run time and the length of each correlation interval.

ReceiverBoardInit.c, ReceiverBoardInit.h

These files contain functions used to set parameters in the DDRs and to synchronize the sampling between processors.

dsp2host.c, dsp2host.h

These files contain functions for passing data from the DSP to the host PC using the streaming API protocols.

ProcessorA.mak, ProcessorB.mak, ProcessorC.mak, ProcessorD.mak

These are the make files used by Code Composer Studio to build the executable files that run on the DSP hardware. These files contain compiler options and linker build options for the all of the source files in the astronomy workspace. Some of the source files described above are custom compiled for speed while others are not. Therefore, the make files are part of the source files.

apihost.c, byuapitest.dsp, byuapitest.dsw

These are the C source file, Visual C++ project file, and Visual C++ workspace file for creating the streaming API called CorrAPI.exe. CorrAPI.exe must be compiled and linked using Microsoft Visual C++. The Debug directory in the astronomy folder is created by Visual C++ and should be left alone. The streaming API is responsible for handling the data passed from the DSP to the host PC. Once again, the project and workspace files contain build options and compiler dependencies necessary to create CorrAPI.exe.

B.1.2 Intermediate and Output Files

The following files are created when the source files are compiled and/or linked. These files can be deleted to save disk space if they are not being used.

ProcessorA.out, ProcessorB.out, ProcessorC.out, ProcessorD.out

These are the executable files generated by the Code Composer Studio compiler and linker that run on the DSP hardware.

ProcessorA.map, ProcessorB.map, ProcessorC.map, ProcessorD.map

The map files are generated by the CCS linker and show how the memory in the DSP is being used.

filename.obj, filename.ncb, filename.ilk, filename.log, filename.plg, filename.opt

Filenames with these extensions are intermediate files generated by the compilers and/or linkers while compiling and building executables.

CorrData

The files that begin with CorrData are the data files created by the streaming API. The CorrData files contain the correlations computed by the DSP and other data parameters. The format of the CorrData files is covered in the comments of data2corr.m (discussed below).

B.1.3 Helper Code

This subsection describes various matlab scripts, CCS customization files, and batch files written to automate the processes of writing DSP code and running it on DSP hardware.

filename.gel

Gel files are customization files for Code Composer Studio. These files make the code generation process less tedious. Some of the gel files make calls to the batch files discussed below.

StartSwifnet.bat, startswiftnet.m

The batch file starts the Swiftnet control panel (minimized). The Matlab script file runs StartSwifnet.bat from the Matlab command prompt.

AstronomyQuickStart.bat

This file can be used to quickly start the Swiftnet control panel and load the astronomy workspace into Code Composer Studio. The DSP must be turned on when you run this batch file.

RunStream.bat

This batch file starts the streaming API (CorrAPI.exe) for the correlator code. It is called by Matlab code and by the gel customization files.

EasyReset.bat, dspreset.m

The batch file resets all four processors on the DSP. Note that this is a software reset. dspreset.m is used to call EasyReset.bat from the Matlab command prompt.

MakeAll.bat, makeall.m

This file compiles all of the C source files for the correlator code using the CCS compiler. Then it builds the four executable .out files for the four processors using the CCS linker. makeall.m calls MakeAll.bat from the Matlab command prompt.

BuildTunable.bat, buildtunable.m

This batch file compiles only the C source files that are dependant on Tunable.h. Then it also builds the four executable .out files. Like MakeAll.bat, this batch file also calls the CCS compiler and linker. buildtunable.m calls BuildTunable.bat from the Matlab command prompt.

LoadNRun.bat, loadnrun.m

This file loads each .out file on the appropriate processor and runs the code. loadnrun.m calls LoadNRun.bat from the Matlab command prompt.

DeleteCorrData.m, clean_folder.m

DeleteCorrData.m is a Matlab script file that deletes all of the CorrData files in the astronomy folder. Be sure you don't need any of the CorrData files before you run this script file. clean_folder.m is a script that deletes the intermediate and output files in the astronomy folder. As with DeleteCorrData.m, be sure you don't need any of the intermediate files before you run clean_folder.m.

write_Tunableh.m

This Matlab function can be used to write the header file Tunable.h. This is useful for making on the fly changes to the correlator from Matlab. The inputs to this function are the program run time, the correlation time, the DSP center frequency, the decimation rate (which controls the bandwidth), and a switch for flipping the spectrum.

dspwarmer.m

This Matlab script warms up the DSP. This script should be run before using the DSP to compute correlations. Once the DSP is warmed up, you can test it using the four-way splitter as shown in Figure B.2. See the paragraphs on AnalyzePhase.m, in the next subsection, for information on testing the four channel correlator code.

collectdata.m

This Matlab script is used to collect data. The script can be programmed to run the correlator code for a specific number of intervals at a specific run time. Typically, when making astronomical observations, you set the run time to 30 minutes and the run time to however many hours the astronomical source is visible.

test_timing.m

This script was useful in code development to confirm that LoopTiming.c was working.

filename.scc

These are Visual Source Safe files.

B.1.4 Initial Data Analysis Code

Most of the code for analyzing the data from the correlator is described in Section B.3. However, the analysis code described in this subsection resides in the astronomy folder and is used for initial data analysis, and so it is covered in this section.

data2corr.m

This is a Matlab function which takes a CorrData file name as an input and returns the normalized correlation matrices and other code parameters. These parameters include, but aren't limited to, the decimated sample rate, the start time, the time per correlation, and the last two buffers of sampled data from each channel. The comments in data2corr.m describes the format of the CorrData data files.

AnalyzePhase.m, mavgdws.c, mavgdws.dll

AnalyzePhase.m is a Matlab script used to perform initial data analysis of the correlations. mavgdws.c is a Matlab mex file that is called by AnalyzePhase.m. mavgdws.c must be compiled using mex before AnalyzePhase.m will work (see mavgdws.c file for details on compiling the mex file).

AnalyzePhase.m calls data2corr.m and plots some of the data as an aid to determine if the observation was successful. Some of the plots this script generates are shown in Figures 5.10, 5.11, and 5.12.

This script can be used to test the correlator code to determine if the data is valid and is synchronized properly. Attach the four-way splitter to the DSP as shown in Figure B.2. Attach a signal generator to the input of the splitter and set the generator to the DSP center frequency. Set the generator to output an AM modulated signal. Set the AM modulation frequency to at least one-hundredth of the sampled bandwidth. Set collectdata.m to collect a small block of data. The phase from the test should look something like Figure B.4, and the time data should look something like Figure B.3.

Figure B.3 shows that the data appears to be synchronous. The phase data confirms this assumption since each cross correlation is within 25 degrees of zero. See Chapter 5 and the AnalyzePhase.m script file for more details.

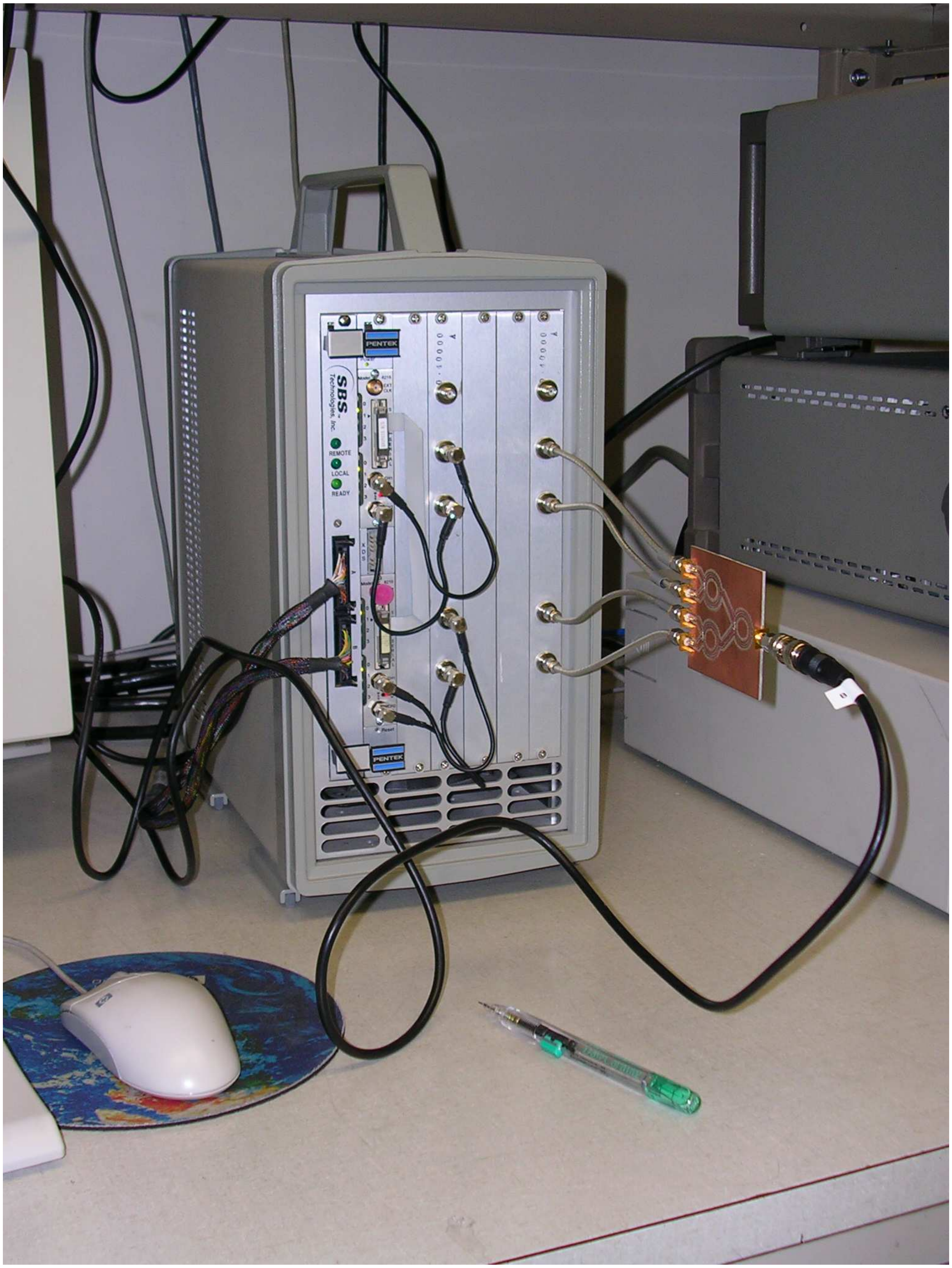

Figure B.2: DSP with four-way splitter on the inputs to each channel.

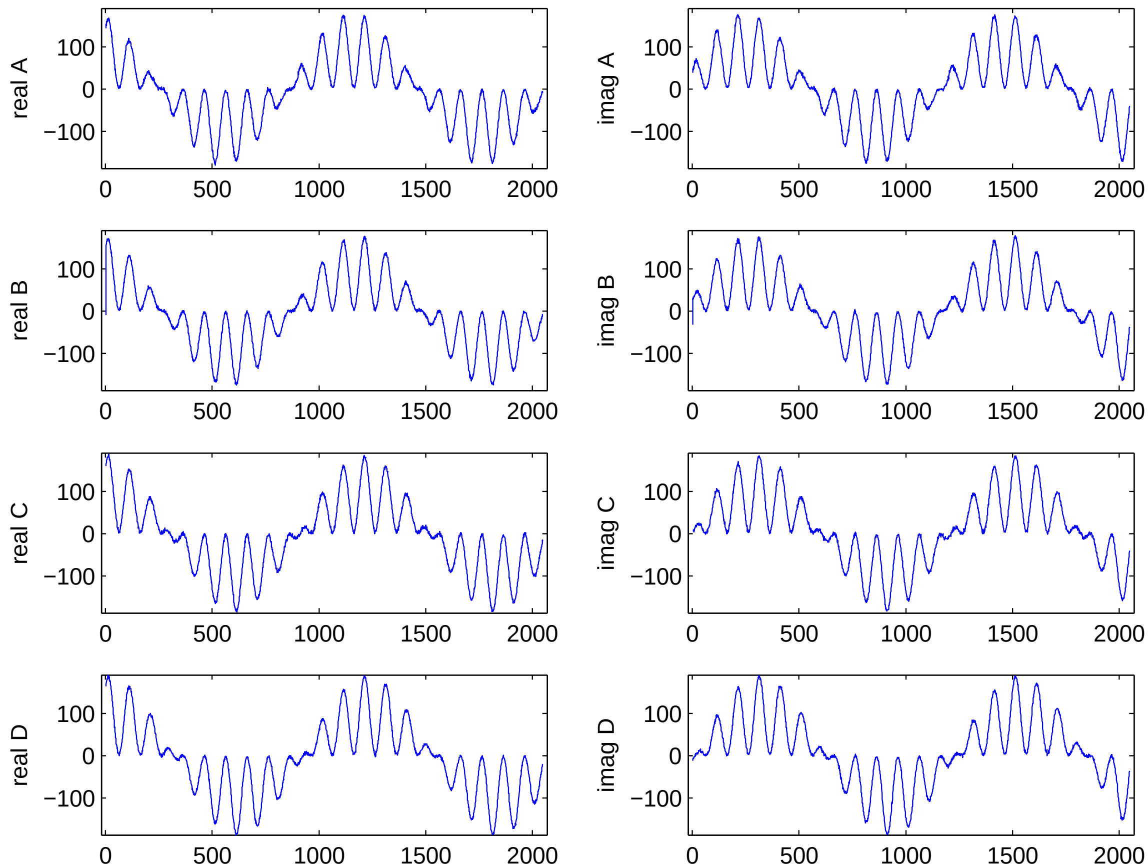

Figure B.3: The last two buffers of sampled data for each channel using a signal generator as an input.

B.2 Observation Code

This section describes the software found in the folder D:\Matlab\work\Observe on the computer named ASTRONOMY. If you would like more detail on a particular piece of code than what is presented in this section, look at the specific source file for more information. The source code is well commented. For the Matlab functions you can type ”help filename” at the Matlab prompt.

B.2.1 Astro and satellite

The code in the Observe directory uses functions from the directories Astro and satellite. Therefore, *the Astro and satellite directories must be in the Matlab path!*

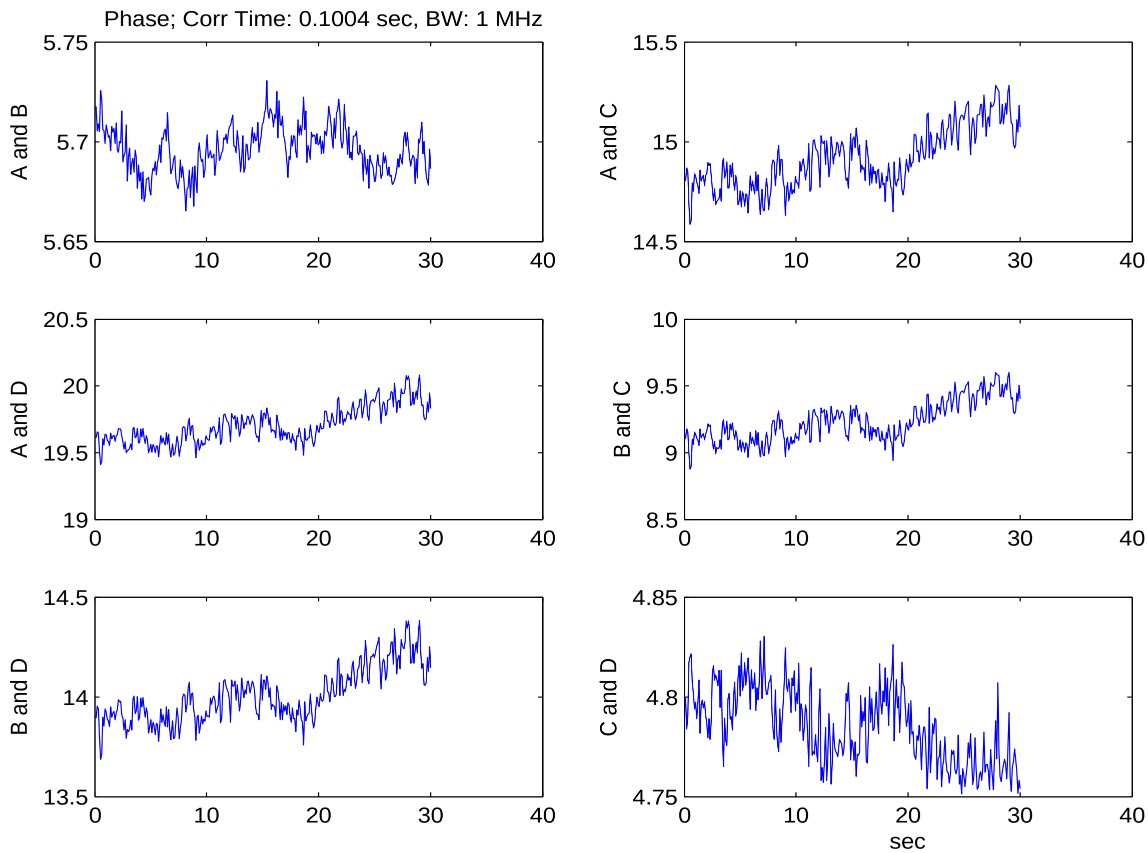

Figure B.4: Phase between channels of the correlator using a signal generator as an input.

The code in these directories will not be covered in this thesis. You should read the ReadMe.txt files in each of these directories for information on the software found in these directories.

B.2.2 Tracking and Steering Software

controlcenter.m, controlcenter.fig

These two files comprise the GUI interface for controlling the antennas. The GUI can be used to point the antennas and track astronomical sources. To run the GUI, type controlcenter at the Matlab command prompt. A picture of the control-center GUI is shown in Figure 5.8.

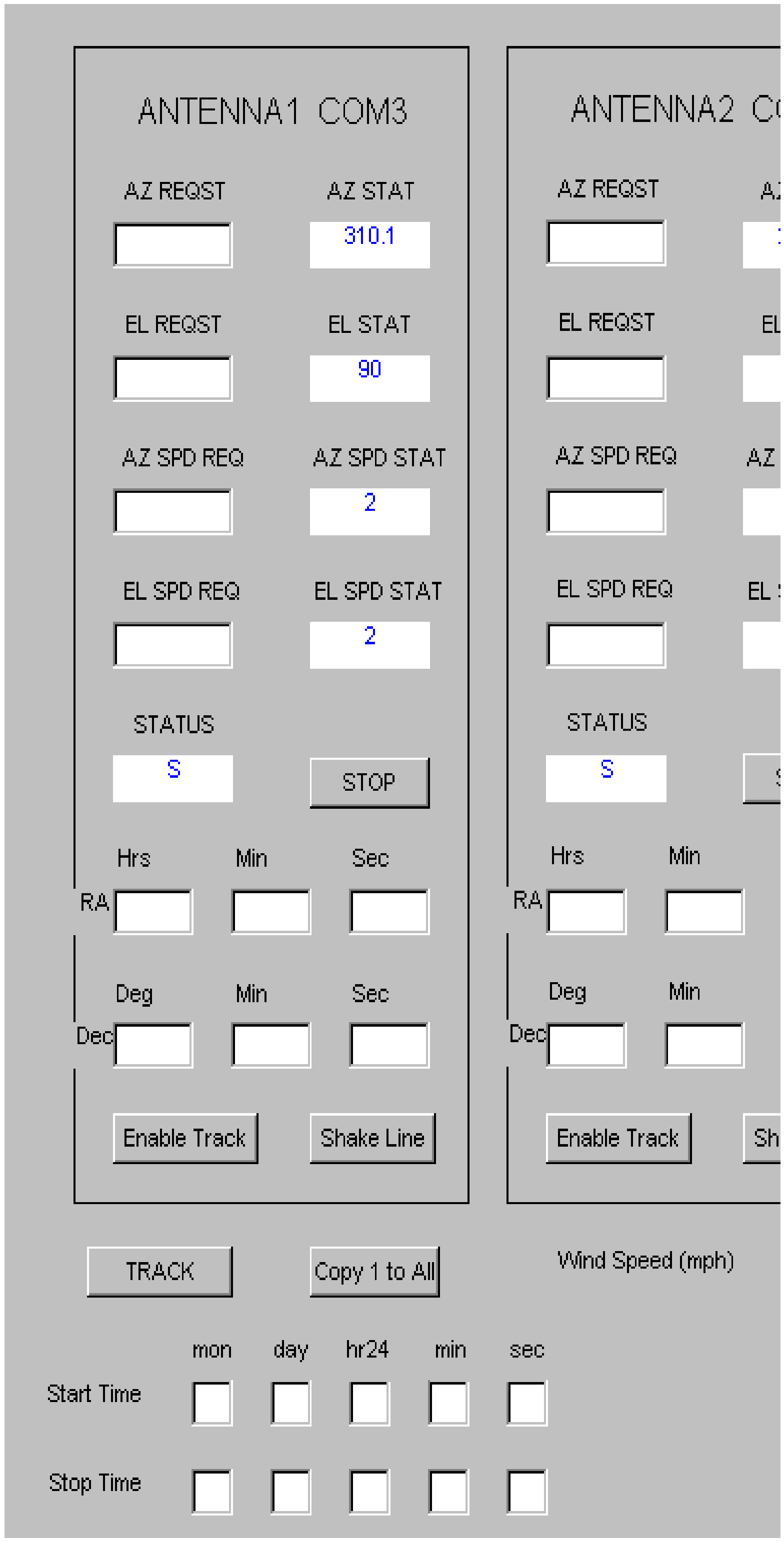

Figure B.5: A close up of the controlcenter GUI.

For each antenna the GUI has a column like the one shown in Figure B.5. At the top of the column you can see which serial port the antenna is attached to. In this case, antenna 1 is controlled by serial port number 3. The text display boxes labelled AZ STAT and EL STAT display the current azimuth and elevation of the antenna in degrees.

The input text boxes labelled AZ REQST and EL REQST allow the user to input an azimuth and elevation heading in degrees. For all of the text input boxes in the controlcenter GUI, type the number and press enter. For example, to point the antenna to an elevation of 90 degrees, type 90 in the EL REQST text input box and press enter.

The text input boxes labelled AZ SPD REQ and EL SPD REQ allow the user to change the positioner speeds for the azimuth and elevation positioners, respectively. Valid speed settings are 1 through 9, with 1 being the slowest and 9 being the fastest. The text display boxes labelled AZ SPD STAT and EL SPD STAT show the current speed for each positioner. The text display box labelled STATUS displays S (for stopped) when the antenna is not moving, and M when the antenna is moving. The STOP button will stop the antenna when it is moving.

The text input boxes labelled with RA (right ascension) and Dec (declination) are for entering an astronomical source location. To input a source with a negative declination, only negate the number you enter in the Deg text input box of the declination. Only enter integers into these text boxes. Not all of the antennas must track the same source. Each antenna can track a different source if desired. If you want all of the antennas to track the right ascension and declination in antenna 1's RA and Dec boxes, press the Copy 1 to All button.

To command an antenna to track a source, you must press that antenna's Enable Track button. Then press the TRACK button. Only those antennas with Enable Track pressed will track the source, the antennas that are not enabled will remain motionless. While the antennas are tracking, the GUI refreshes the weather information and will stow the antennas if the wind speed is greater than 10 mph. To stop the tracking, press the TRACK button again.

Start Time and Stop Time are used to start and stop tracking at a particular time. To use the Start Time and Stop Time, you must input the start and stop times before you press the TRACK button.

The Shake Line button is used to send a test signal to the antenna controllers in the event that the serial communication is not working. The Update Weather button refreshes the Wind Speed and Temperature text display boxes to reflect the current weather conditions (weather software is discussed in subsection B.2.3). Track offset is used to increase the tracking accuracy of the M^2 positioners. For more information, see the M^2 antenna manuals and the controlcenter.m file.

astrotracker.m, astropar.txt

These two files were created to track an astronomical source without a GUI. They aren't working properly yet. Fix them or use controlcenter.

satellitetracker.m (or satellitetracker.exe) and satpar.txt

satellitetracker.m can be used to track a single GPS or GLONASS satellite for a specified length of time. This file uses functions and data files from the satellite directory. To use satellitetracker.m, run the script update_satellite_params.m. This script queries a web page for up to date orbital parameters on GPS and GLONASS satellites. These orbital parameters change frequently. I recommend running update_satellite_params every day you use satellitetracker.m. Use the abovehorizon.m and glo_abovehorizon files to pick a satellite to track. Set the parameters in satpar.txt These parameters are repeated below.

GPS_ONE_OR_GLO_ZERO 1
SATELLITE_NUMBER 6
MINUTES_TO_TRACK 20
TRACKING_OFFSET 0.8
STOW_ONE_OR_NO_STOW_ZERO 1
SERIAL_ANT1 0
SERIAL_ANT2 0
SERIAL_ANT3 5

SERIAL_ANT4 0

START_TIME 0.0

The first parameter selects either GLONASS (zero) or GPS (one). Satellite number is the abovehorizon satellite index. MINUTES_TO_TRACK is the number of minutes to track the satellite. TRACKING_OFFSET is an offset in degrees added to the actual azimuth and elevation to increase the performance of the M2 controllers. It should be set to 0.7 or 0.8. The stow parameter selects whether or not to stow the antennas after the track (point the antennas up). The SERIAL_ANT parameters select the serial port that an antenna is controlled by. Set SERIAL_ANT to zero if you don't want a particular antenna to track. Start time is a Matlab serial date number for the time you want to start tracking. If it is set to 0.0, tracking begins when you run satellitetracker.m. satellitetracker.exe is a stand alone executable version of the Matlab function (you can run it without Matlab).

The following files are created by Matlab for the .exe file and should be left alone unless you make changes to satellitetracker.m and recompile satellitetracker.exe: satellitetracker_main.c, satellitetracker_mcc_component_data.c, and satellitetracker.ctf. The directory satellitetracker_mcr is created when you run satellitetracker.exe for the first time. This directory should also left alone unless you make code changes.

B.2.3 Weather Software

getwebwth.m

This Matlab function retrieves current weather conditions for BYU campus from the internet. There is a weather station on top of the Eyring Science Center. The data from this station is posted to a web page and is updated every second. getwebwth.m is used by controlcenter.m and wthrcdr.m (discussed below).

wthrcdr.m (or wthrcdr.exe) and wthrtime.txt

wthrcdr.m records the weather for the time and interval length specified in wthrtime.txt. wthrcdr.exe is an executable version of the .m file which can be run without Matlab.

The following files are generated by Matlab when you compile whtrcdr.exe with Matlab's mcc: wthrcdr_main.c, wthrcdr_mcc_component_data.c, and wthrcdr.ctf. The directory wthrcdr_mcr is also created when you run wthrcdr.exe.

crweather.m

This function is used to process and display the weather data from the weather.dat file created by wthrcdr.m (or wthrcdr.exe).

B.2.4 Management Scripts for Executables

DeleteExecFiles.m

This script deletes all of the executable files and auxiliary files generated by Matlab's mcc when creating the satellitetracker.exe and wthrcdr.exe. If you are making changes to satellitetracker.m and wthrcdr.m, run DeleteExecFiles.m before you use mcc to create satellitetracker.exe and wthrcdr.exe.

GenerateExecFiles.m

This script creates executable files for wthrcdr.m and satellitetracker.m. Run this script after making code changes to the .m files so that the changes are reflected in the executables.

B.3 Analysis and Image Generation Code

This section describes the Matlab code used to perform data analysis and calibration. This section also covers the code that generates an image from measured visibilities. See also Section 5.4.

B.3.1 Making an Image

For convenience, I have repeated the major steps to generate an image below.

1. Measure the visibilities by estimating correlations.

2. Edit, calibrate, and phase rotate the data.

3. Compute u and v from the antenna locations and the source location.

4. Fourier transform the visibilities.

B.3.2 Initial Analysis

After several hours of observing an astronomical source, there will be a number of CorrData files in the C:\ti\myprojects\astronomy directory. Copy these files to some sort of removable media (like a zip disk) into a directory of their own. Create a new text file called ExperimentalSetup.txt in that same directory. In this text file, you should record the start and stop dates of observation, the observing frequency (I am assuming that the observations were all made at the same frequency), the LO settings, the source right ascension and declination, and any other pertinent information about the observation. Place a copy of AnalyzePhase.m, data2corr.m, and mdwsavg.c in this folder. (This is a good idea even if you have this software on your own computer already.) Finally, place the weather.dat file for the observation in this folder as well. Copy the folder to your computer.

Use the AnalyzePhase.m script to determine which CorrData files, if any, should be deleted entirely. See Sections 5.4.3 and B.1.4 for ideas on how to identify corrupted data. Create a .txt file in the directory you are working in called FlaggedData.txt. When you find files with suspicious or corrupted data, make a note of it in the text file. If an entire CorrData file is bad, either delete it, or rename it with the word "FLAGGED" at the front of the file name.

B.3.3 cruncher.m

cruncher.m is the Matlab script you use to do in depth analysis of the data. It is based on the assumption that you are observing a point source, and therefore the phase between antennas is completely predictable.

To use cruncher.m you must create two .mat files in the directory: centerfreq.mat and FlagMap.mat. Create a variable named fc and set it equal to your observing frequency. Save this variable, and only this variable, in the centerfreq.mat file. Create a variable fd as a 1 x 3 array of zeros and store only this variable in FlagMap.mat.

At the top of cruncher.m, there are several parameters you must enter. Enter the name of the directory with the CorrData and .mat files. Enter the right ascension and declination of the source the array tracked. You must also enter the number of seconds to integrate for each correlation matrix. The top of cruncher.m should look as follows.

```
% Specify the directory.
dirstr = 'May18'; cd([Matlabroot '\work\CasData']); cd(dirstr);


% Position of Cas A.
ra = (23+21/60+13/3600)*360/24;
dec = 58+32/60+35/3600;
AvgT = 10; % average for AvgT seconds.
```

When you run cruncher.m, for each CorrData file, it generates figures for each baseline like the one shown in Figure 5.19. It is shown again in Figure B.6. cruncher.m will process all of the CorrData files in the directory and pause between each file. This particular file has 30 minutes worth of data.

The first five minutes of this CorrData file contains some anomalous data. To remove this data, you must modify the fd variable of the FlagMap.mat file. When you see data you wish to edit, press control-c to stop cruncher.m. You then need to add a row to the fd variable. The first element of the row is the start_time_num variable for this CorrData file. The second and third elements are fractions.

The first fraction is how much of the data you wish to eliminate from the front of the set. For the data in Figure B.6, we wish to eliminate the first five minutes of data. At the Matlab prompt, you would type:

```
>> fd = [fd; [start_time_num 5/30 0]];
```

Once you have modified the fd variable, save the variable to the FlagMap.mat file as shown below.

```
>> save('FlagMap','fd');
```

The second fraction specifies how much data to eliminate from the end of the set. If you wanted to eliminate the last 10 minutes of data, you would type:

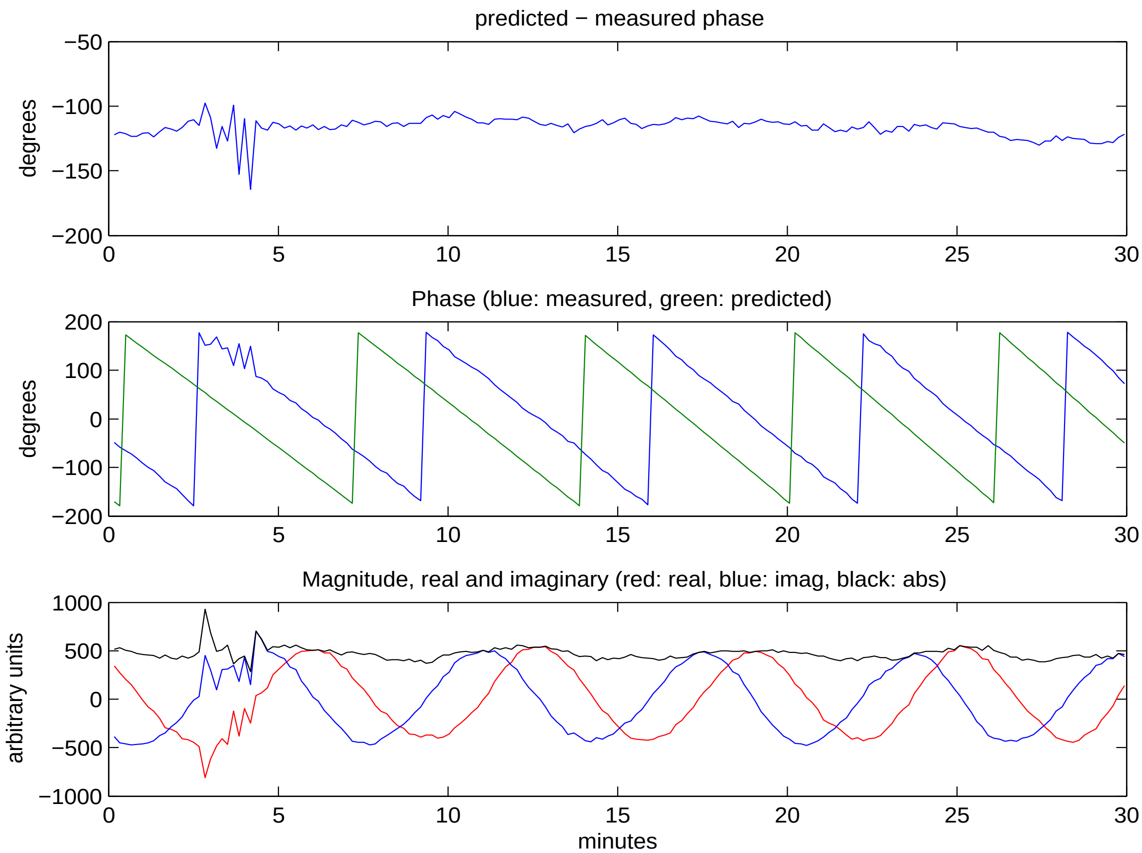

Figure B.6: Unedited cross correlation data for baseline 1,3.

```
>> fd = [fd; [start_time_num 0 20/30]];
>> save('FlagMap','fd');
```

If you only want to clip data from the front, set the third fraction to zero, as in the first example. Likewise, if you only wish to clip data from the end of the data set, set the second fraction to zero, as in the second example.

The results of editing the data from Figure B.6 are shown in Figure B.7. Notice the data set only spans 25 minutes now instead of 30. This is because the first five minutes have been removed.

cruncher.m also performs phase calibration. In Figure B.7, the difference between the measured and predicted phase hovers around -120 degrees. cruncher.m assumes that every baseline in each data set has some bulk phase shift that needs to

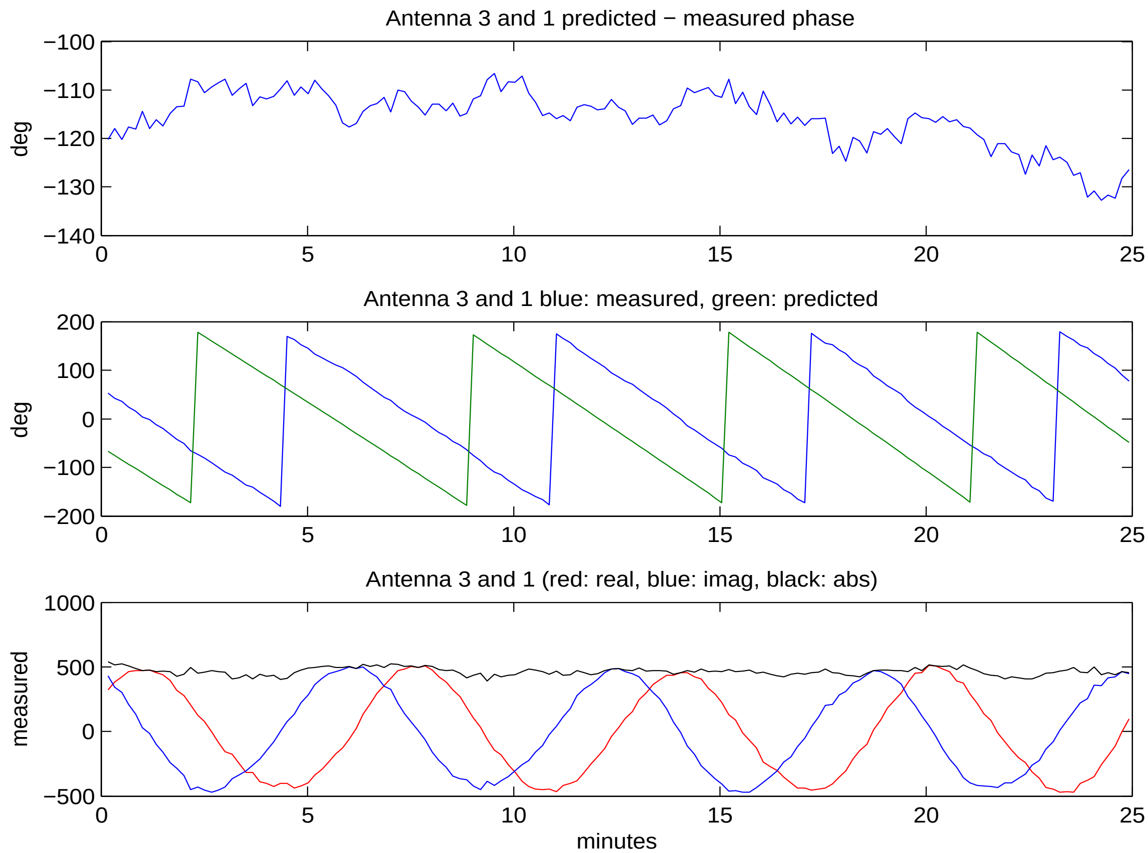

Figure B.7: Edited cross correlation data for baseline 1,3.

be removed. cruncher.m estimates this phase shift for each baseline in a data set and phase rotates the data to eliminate the shift.

You should use cruncher.m until all of the CorrData files in the directory are properly edited, as described above. cruncher.m outputs a .mat beginning with the word Data. The .mat file contains variables called u, v, and Vis, which are u, v, and the corresponding measured visibilities for one CorrData file.

B.3.4 gather.m

At this point your directory should have a Data.mat file for all of the CorrData files in the directory. Use the gather.m Matlab script to place all of the data in the Data.mat files into one .mat file. At the top of gather.m, set the directory name to the proper directory as shown below.

```
% Specify the directory.
clear all;
dirstr = 'Apr27';
```

After you run gather.m, there should be a DataAll.mat file in the directory. This .mat file will have u, v, and Vis for all of the CorrData files in the directory. These variables are suitable for making an image using the functions in the ToolImag directory which is described in the next subsection. If you are making an image at more than one observing frequency, place the DataAll.mat files from the different directories into a new directory. Use gatherAll.m to generate imaging variables from the multiple DataAll.mat files. gatherAll.m will create a single DataAll.mat file.

B.3.5 Tools for Imaging

In addition to AnalyzePhase.m and data2corr.m (see Section B.1.4), a number of Matlab functions were written to facilitate generating an image. These functions are found in the directory called ToolImag. As with the other code described in this appendix, look at the actual source files for more detailed information. For the Matlab functions, you can type "help functionname" at the Matlab command prompt, where functionname is the name of the Matlab function.

make_ToolImag.m

This Matlab script compiles the mex C files in this directory. You should run this script before using the software in this directory.

cellavg.m

This function uses cell averaging to interpolate visibilities to a rectangular grid. Then fft2 is used to create a synthesized image.

cxfrmdir.m

This function uses a direct transform to create a synthesized image from visibilities. This function writes, compiles, builds, and runs C code to perform the transform. On a Windows PC, this requires that you download and install the lcc compiler. Look at the Matlab help for this function to find the url to download lcc. On a Unix machine, this function uses gcc.

enu2uvw.m

Given a synthesis array's antenna positions in meters East, North, and Up, this file finds the corresponding u, v, and w in the uv plane.

mfind.c, mfind.m, mfind.dll

mfind.c is a mex file used by cellavg.m. mfind.m contains help on mfind.c, and mfind.dll is the compiled application extension used by Matlab when the mfind function is called. .dll is the Windows extension. The extension is different for other platforms. For example, the extension on a Linux PC platform is .mexglx.

mxfrm.c, mxfrm.m, mxfrm.dll

mxfrm.c is a mex function for generating a synthesized image from visibilities using a direct Fourier transform. This function is not as fast as cxfrmdir.m, however, mxfrm.c doesn't require that you install the lcc compiler. mxfrm.dll is the compiled application extension used by Matlab when mxfrm is called.

uv_prune.m

uv_prune.m is a Matlab function that generates a GUI for manually deleting visibilities from the uv plane. Areas in the uv plane with a high density of visibilities tend to increase sidelobe levels in the dirty image. The purpose of deleting visibilities is to reduce sidelobes. See the help for this function (type "help uv_prune" at the Matlab command prompt).

Appendix C

Recommended Improvements

This appendix documents some of the problems with the VSA and suggests some improvements to overcome the problems. The appendix also lists some improvements to the array that aren't essential, but would enhance the arrays performance. The recommendations in this appendix are in addition to those discussed in Chapter 7.

Our research group has already begun work on new receiver and data acquisition systems. These new systems will solve a number of the problems discussed below.

C.1 Problems

C.1.1 Cross Talk

This subsection presents the full results of the cross talk experiment introduced in Section 5.4. The receiving channel is channel 3. The channels not being measured were terminated with 50 Ohm terminators. For this particular experiment, the coupling between channels 2 and 3 is the worst. The cross coupling causes phase drift in the baselines with receiver channels close to each other.

As you can see from Figures C.1, C.2, C.3, and C.4, the channels have some significant cross talk. The adjacent channels have only 30 to 35 dBm of difference from the channel that is actually receiving the signal.

The new receiver system, mentioned above, will be connectorized and should solve the cross talk problems with our array.

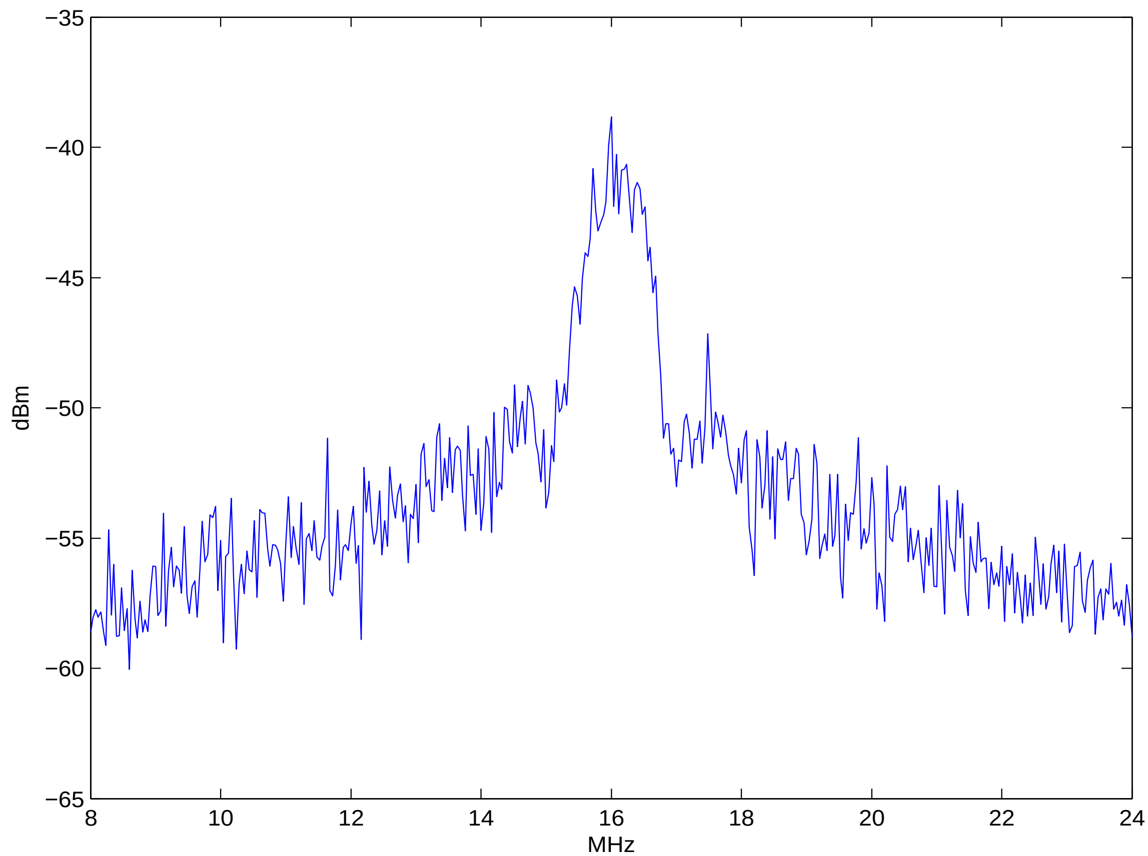

Figure C.1: A GPS satellite spectrum from channel 1 of the Walkenhorst/Hansen receiver.

C.1.2 Antenna Positions

Once the cross talk problems are solved, accurately determining the antenna positions should be much easier. With the new system, observing some satellites and some astronomical point sources should provide stable phase measurement from which the locations of the antennas can be accurately determined.

C.1.3 Pentek DSP Issues

The Pentek DSP needs to be repaired or replaced. As mentioned above, our group is in the process of building a new data acquisition system which elimates the need to use the DSP. However, the DSP is a powerful real time data processing

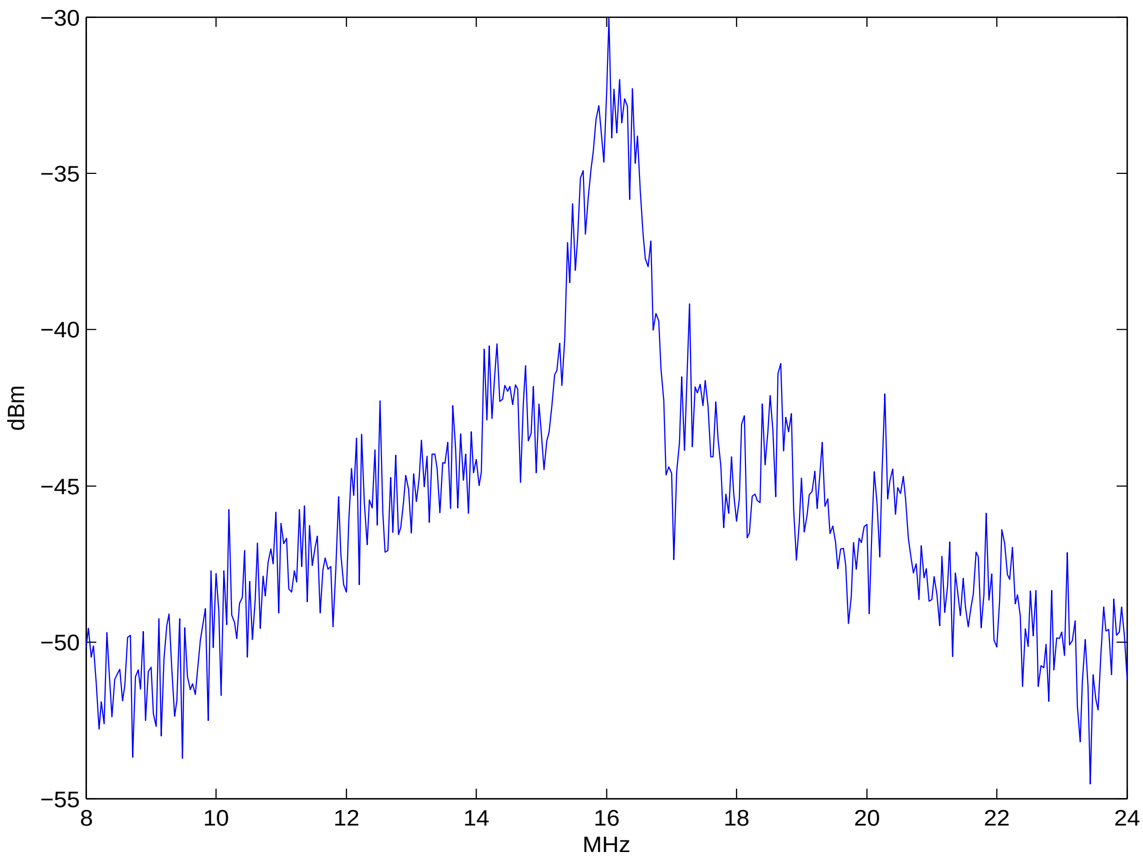

Figure C.2: A GPS satellite spectrum from channel 2 of the Walkenhorst/Hansen receiver.

instrument, and I hope that the problems it has will be accurately diagnosed and repaired. Until that happens, be aware of the following problems that I have observed while using the DSP.

The DSP must be warmed up. That is, it must be slowly coaxed into operating properly. You can not just turn on the DSP and run your code.

The DSP also seems to fail when the input power on the front channels is high. *This behavior effectively prevents you from using the repeaters when you acquire data with the DSP.* Figure 5.12 shows what (usually) happens when the input power is too high. I suspect that this is a problem with one or both of the P6216 receiver boards. The correlation between channels A and B looks just fine, which means that the data

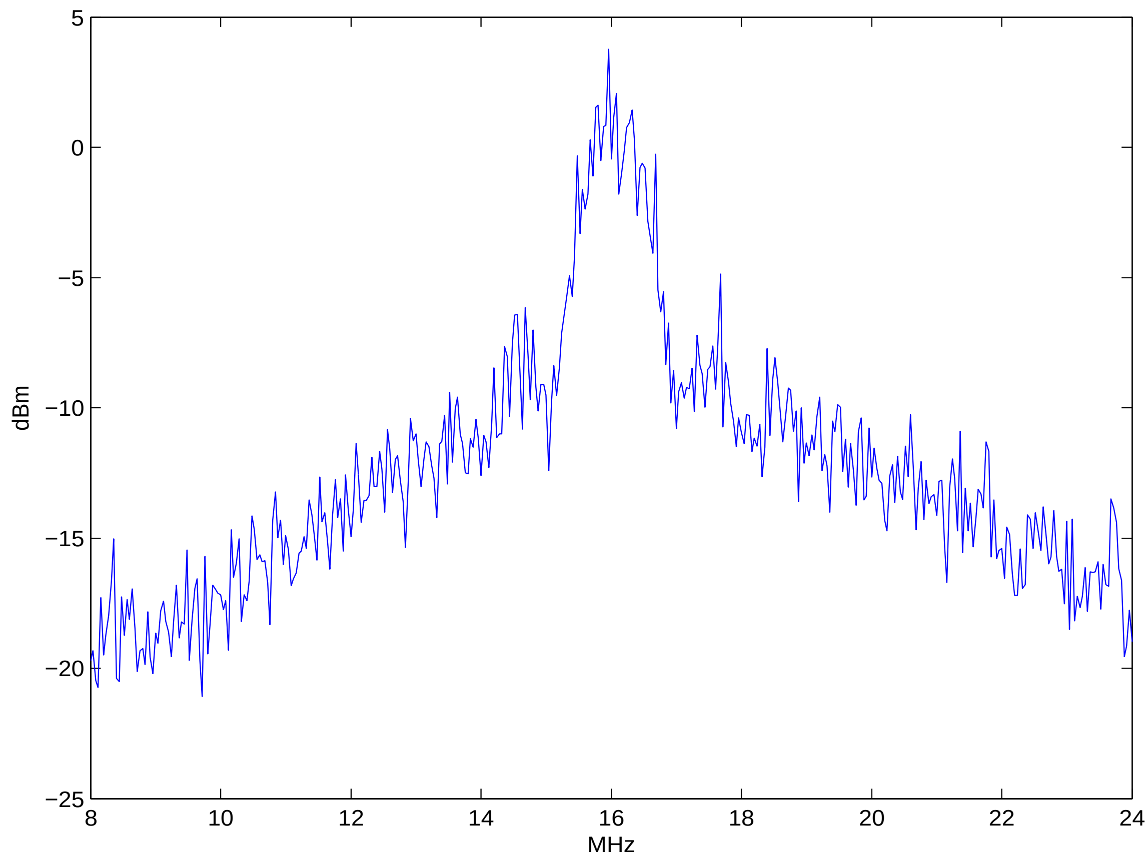

Figure C.3: A GPS satellite spectrum from channel 3 of the Walkenhorst/Hansen receiver.

is being passed to processor D properly, and the correlation code is working. In order for the data chain to operate, each processor must pass data at the appropriate time. In order for a processor to pass data and run the rest of its code, the processor's associated receiver must be filling up the processor's IO bififo. So something is filling the IO bififos. From Figure 5.12, it appears that channels A and B are sampling data properly, in other words, the correlation between A and B doesn't look like noise. The other correlations include data from either processor C or processor D. I suspect that channels C and D are not sampling properly. Bear in mind though that this behavior is not usually observed when the repeaters aren't being used.

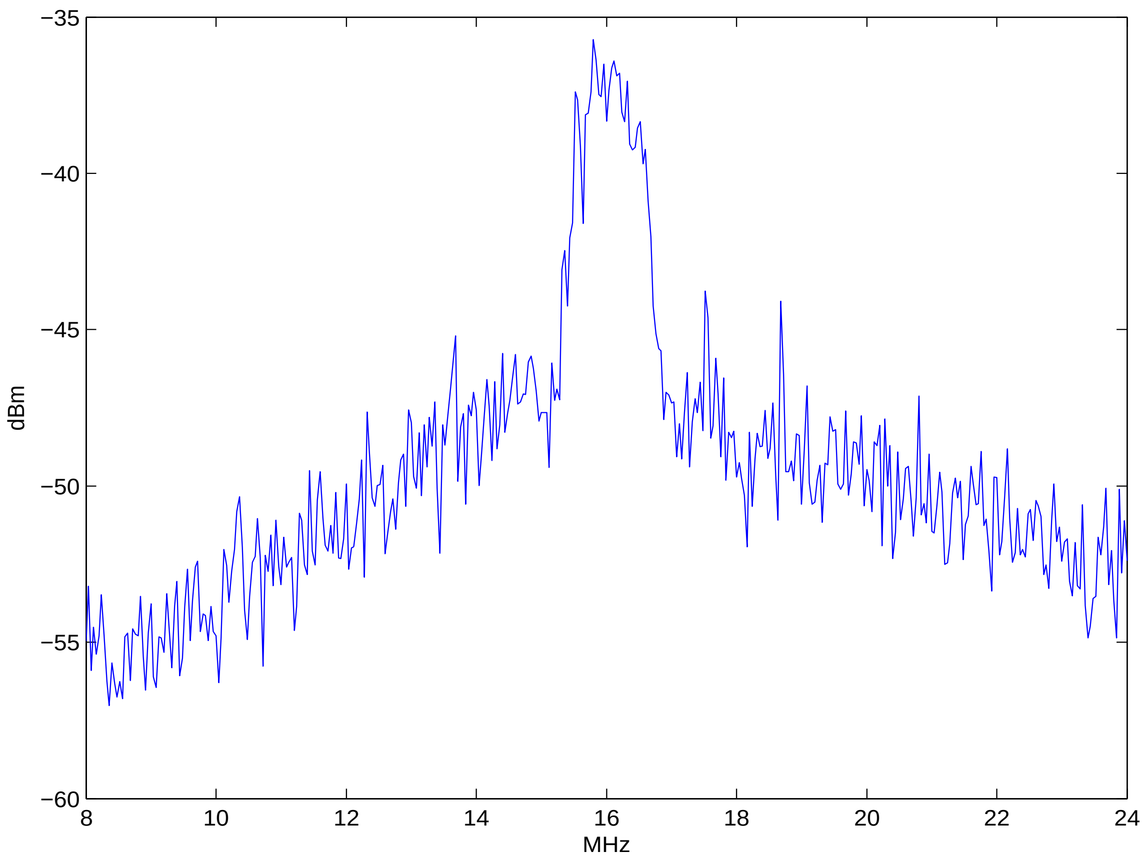

Figure C.4: A GPS satellite spectrum from channel 4 of the Walkenhorst/Hansen receiver.

C.1.4 Absolute Gain Calibration

Once the new receiver system is in place, absolute gain calibration should be performed using both satellite and astronomical sources. Among other things, this will aid in characterizing the array's sensitivity in terms of Janskys. This is how radio astronomers characterize the sensitivity of image synthesis arrays. Calibrating the gain will also allow us to put units on the data we observe with the VSA.

C.2 General Improvements

This section discusses some general improvements to the array. These improvements are not critical like the ones mentioned above.

C.2.1 Fiber Optic Cable

An especially marked improvement to the array would be to use fiber optic cable for carrying the signal from the dishes into the building. Fiber optic cable is decidedly lower loss than the current coaxial cable being used at the time of this writing. With fiber optic cable, there would be no need for the repeaters. In the past, we have spoken to the astronomy department about putting an antenna on top of the Eyring Science Center. If another antenna is added to the array that is not on top of the Clyde Building, the signal from that antenna will have to be carried on fiber.

With the coaxial cable that is currently in place, observing at higher frequencies (3 - 8 GHz) increases the amount of insertion loss incurred getting the signal into the building. The coaxial cable insertion loss is 7 dB/100 ft at 1.6 GHz. At 4 GHz, the loss is 11.5 dB/ 100 ft. With fiber optic cable, there would not be any (noticeable) loss observing a higher frequencies. Also, the transmission properties of fiber optic cable are less sensitive to temperature change than the current coaxial cable.

C.2.2 A New Host PC

Running the array would be much more convenient if the host PC were a faster computer (something with a processor faster than 900 MHz) and if the host PC was running Windows XP instead of Windows NT 2000. In my mind, this change will need to be made eventually anyway. The host PC we have now is fast becoming a dinosaur and Microsoft pulled their support for Windows NT a long time ago.

C.2.3 Software Improvements

Another improvement to the array would be to write the tracking software in C++, Java, or LabVIEW. The GUI created by controlcenter.m performs two main functions: it computes positions of sources based on the PC's clock, and it manages the serial ports that connect to the M^2 controller boxes. A new control center could also have some kind of internet connectivity to allow a user to control the antennas from her desktop PC.

Additional imaging software could expedite the process of generating images. A software GUI for analyzing phase, calibrating the data, and plotting the uv coverage would be a vast improvement over the script files I have written to do the same things.

Some software should also be written for weighting the visibilities such as density weighting and taper weighting (see [23]). Also, software should be written to perform deconvolution. Deconvolution methods currently in use by radio astronomers are based on CLEAN and Maximum Entropy Method (see [24]). There is no weighting or deconvolution software for the VSA as of this writing.

Appendix D

DSP Technical Literature

D.1 Introduction

This appendix reviews the technical documents one should have access to and be familiar with to program the Pentek DSP platform used by the Radio Astronomy research group. Some literary license has been taken by the author in writing the review, mostly in the informal tone which it is written in. The review is intended for those who will be working with the DSP code after I leave. My hope is that your learning curve will not be as steep as mine was and that you will be able to wade through the technical literature described below faster than I did.

D.1.1 Assumptions

In writing this review, some assumptions were made about the background of the DSP Programmer. I assume you are comfortable with the C programming language. You should also be familiar with concepts and terms associated with microprocessors such as interrupts, internal and external memory, data and program memory (Harvard Architecture), data and control registers, and cache. Some familiarity with code generation software such as assemblers, compilers, linkers, and integrated development environments (IDEs) is also helpful.

You should read Section 2.4 of Andy Poulsen's Master Thesis [8].

D.1.2 Our Hardware

It is also important to remember the technical specifications of the hardware we use. We use a Pentek P4291 Quad Processor DSP board, with Option 330 installed. The board is attached to two Pentek P6216 DDR boards.

D.1.3 Finding Documentation

This review should not be construed to be comprehensive, and you should know where to find documentation when the need arises. Both Texas Instruments (TI) and Pentek maintain websites with current revisions of the documentation reviewed in this appendix. You should check the websites periodically and make sure the documentation you are using is the current revision.

Pentek's web address is: http://www.pentek.com. You will need to register with Pentek to obtain any useful documentation from the website. Click the register link on Pentek's home page and register with Pentek. Pentek's web pages provide documentation on the hardware of the P6216 and P4291, and provide documentation for ReadyFlow, Pentek's board support software for their DSP boards.

TI's DSP support web address is: http://dspvillage.ti.com. The microprocessors used in the P4291 are TMS320C6701 Floating Point processors. Navigate to the web pages describing the C6000 family of Digital Signal Processors. These pages provide a wealth of information describing the hardware of the TMS320C6000 family microprocessors and how to program the TMS320C6000 family.

D.1.4 Appendix Structure

I have taken a sequential approach in structuring the sections that follow. By sequential, I mean starting at the analog inputs to the DSP and ending at the host PC. During the review, I will point out certain pages, section and chapters which should be read instead of merely skimmed. I also give the current revision of the document (at the time of this writing) and in some cases the Adobe Acrobat file name of the document.

D.2 P6216: Digital Drop Receiver

Operating Manual: Pentek Model P6216 Rev. B You should read the first chapter of the operating manual for the P6216 Digital Down Converter. The first chapter provides an overview of the ADCs and their specifications. Read through the table of contents of the operating manual as well. The operating manual contains the specifications for the analog inputs, and the DDRs. It also describes the down conversion process. The appendices of the operating manual include the data sheets describing the ADCs of the P6216. The Acrobat file name for this document is: 80062160.pdf.

Model P6216 Ready Flow Board Support Libraries Rev. 4 This document describes the functions used in a C program to perform such things as setting decimation rate, setting the DSP center frequency, switching between complex and real down-conversion, and so forth. This document does not need to be read in any great detail. It is a programming reference, and you should look through the table of contents to see what functions are provided by Pentek. 80162160.pdf.

D.3 P4291: Quad Processor DSP Board

D.3.1 Hardware Documentation

Operating Manual: Models P4290 and P4291 Rev. D This document describes the Quad Processor DSP board used in this project. You should read Chapter 1 to get an overview of the P4291. You should also read Chapter 3 to learn how the memory is organized and managed on the P4291. Read Section 4.1 to familiarize yourself with the memory map used for the DSP board. Look over the rest of Chapter 4. You should read Sections 6.1, 6.2, 6.3, and 5.5 to learn how to use the Bidirectional FIFOs (First In, First Out Registers). Pay particular attention to section 6.5.5, as it documents a flaw in the bififo hardware. The Acrobat file name for this document is: 801.42900.pdf.

D.3.2 Software Documentation

Software Support Documentation Model P4290/ Model P4291 Ready Flow Board Support Librarys Rev. 2.3 This document describes C functions written by Pentek that are used quite frequently in DSP code for the P4291. In particular, this document describes C functions for using the bififos and the DMA (Direct Memory Access Controller). This document does not need to be read and digested, but you will refer to it quite often in your DSP programming. The Acrobat file name for this document is: 801.42900.pdf.

D.4 TMSC320C6701 Floating Point Processor

D.4.1 Introduction

To begin understanding the C6000 family Processor, you should read the following chapters and sections of **Digital Signal Processing Implementation using the TMS320C6000 DSP Platform** [32]: 1, 2, 3.1, 3.2, 4, 8.5, and 8.6. Chapter 2 is especially helpful because it introduces the processor's architecture. In particular, the chapter introduces the CPU, memory, and interrupt structure of the C6000 family processor. Chapter 2 also introduces the DMA. Chapter 3 provides an introduction to the software code generation tools for the C6000, and describes the process of turning C code into an executable file that will run on a C6000 processor.

D.4.2 Hardware Documentation

The DSP programmer should read Chapters 1 and 2, and Section 8.1 of **TMS320C62X/C67X CPU and Instruction Set Reference Guide**, Literature Number: SPRU189F.pdf, revised October 2000. Chapters 1 is an introduction to TI's DSP families. Chapter 2 describes, in detail, the CPU architecture, and Section 8.1 provides an overview of the interrupts of the C67x processors.

Read the first 3 pages of **TMS320C6701 Floating-Point Digital Signal Processor**, Literature Number: SPRS067F.pdf, revised March 2004. This is the Processor's datasheet and describes all of the bells and whistles on the C6701.

The DSP Programmer should read Chapter 2 of **TMS320C620x/C670x DSP Program and Data Memory Controller/Direct Memory Access (DMA) Controller Reference Guide**, Literature Number: SPRU577.pdf, revised July 2003. This Chapter describes, in detail, the operation of the DMA. Reading this chapter will help the programmer understand the DMA C functions documented in ReadyFlow P4290/P4291.

D.4.3 Software Documentation

Texas Instruments C6X Processor Ready Flow Board Support Libraries, Rev: 1.1, 80162670.pdf. This document is primarily a programming reference. It documents the C functions used for operations such as: setting up, enabling, and mapping interrupts, and reading and writing to registers. The interrupt functions are of note because with Option 330 installed, the interrupts on the P4291 are mappable and must be mapped to the CPU to be used at all.

TMS320C6000 Optimizing Compiler Users Guide.The literature number is SPRU187K.pdf , October 2002. You should read Chapter 1 to get an overview of the code generation process. You should also read Sections 2.1 - 2.4 to learn how to invoke compiler and manipulate the process of compiling C code. Read Chapter 3.1 to learn what kind of optimization the compiler performs on your code. Skim through the rest of Chapter 3. Read Chapter 5 to learn about the linking process. Chapter 5 of the Compiler User's Guide and Chapter 7 of the Assembly Language Tools User's Guide are essential to learning what memory sections are created by the compiler and where those sections are placed in memory. You should also look through chapters 7, 8, and 9 to learn how C is implemented for the TMS320C6000 family processors.

TMS320C6000 Assembly Language Tools User's Guide, Literature Number SPRU186M.pdf, Revised March 2003. This document contains a great deal of information essential to programming the DSP. Read Chapters 1 and 2. Chapter 1 provides an overview of the code generation process. Chapter 2 explains the concept of memory sections and discusses the Common Object File Format (COFF). COFF is the output format for the executable files that run on DSP Hardware.

Chapter 7 should be read, reread, and committed to memory. Chapter 7 details how the linker works. The linker is responsible for placing code and data in the different types of memory on the P4291. This chapter teaches you how to write a linker command (.cmd) file. The .cmd file is used to control the linker's behavior, the linking process, and the memory allocation process. Pay particular attention to the MEMORY and SECTIONS directives explained in this chapter. These two directives allow the programmer to place the different parts of the executable code and data into the different types of memory.

D.5 Transferring Data to the Host PC

D.5.1 Swiftnet Application Program Interface

Developer's Guide: Swiftnet Application Programming Interface This document describes Pentek's prewritten C functions for moving data between the host PC and the DSP Board. Read Chapters 1 and 2 to familiarize yourself with the therory of operation and concepts of an Application Program Interface (API). Read Chapter 5 also as it describes the real time streaming API functions which are the ones used most frequently in the DSP code of the Radio Astronomy Project.

Appendix E

Bibliography Addendum

This appendix includes urls that would not display properly in the bibliography because they are extremely long.

The url for [1] is:

http://ims2.co.utah.ut.us/website/Realtime%20Parcel/
RealTimeGIS.asp?left=1958427.3604822952&bottom=696752.9183559765
&right=1959368.8178292983&top=697597.8159750819&cmd=zoomin
&clickx=0&clicky=0&img=4&dat=2%2F8%2F2005&pht=1&scl=100&par=
&buf=0&cty=Utah+County&qt

The url for [2] is:

http://ims2.co.utah.ut.us/website/Realtime%20Parcel/
RealTimeGIS.asp?left=1958610.8239652996&bottom=696790.0938512167
&right=1959193.8033224817&top=697297.0324226803&cmd=zoomin
&clickx=0&clicky=0&img=4&dat=2%2F8%2F2005&pht=1&scl=100&par=
&buf=0&cty=Utah+County&qtr=0&sec=1&twn=3&rng=3W

Bibliography

[1] "Real time parcels", World Wide Web, See Appendix E for the full url.

[2] "Real time parcels", World Wide Web, See Appendix E for the full url.

[3] A. R. Thompson, J. M. Moran, and Jr. G. W. Swenson, *Interferometry and Synthesis in Radio Astronomy*, chapter 1, John Wiley & Sons, Inc., 2nd edition, 2001, 1-45.

[4] "Nrao very large array", `http://www.vla.nrao.edu/`.

[5] "Radio astronomy for teachers", `http://www.aoc.nrao.edu/epo/teachers/RAFT/`.

[6] "Srt - interferometery", `http://web.haystack.mit.edu/SRT/interferometer.html`.

[7] W. Dong, "Time blanking for gbt data with radar rfi", Master's thesis, Brigham Young University, August 2004.

[8] A. J. Poulsen, "Real-time adaptive cancellation of satellite interference in radio astronomy", Master's thesis, Brigham Young University, August 2003.

[9] B. D. Jeffs, L. Li, and K. F. Warnick, "Auxiliary antenna-assisted interference mitigation for fadio astronomy arrays", *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 439–451, February 2005.

[10] B. G. Clark, "Coherence in radio astronomy", in *Synthesis Imaging in Radio Astronomy II*, G. B. Taylor, C. L. Carilli, and R. A. Perley, Eds., vol. 180 of *A Series of Books on Recent Developments in Astronomy and Astrophysics*, lecture 1, pp. 1–10. Astronomical Society of the Pacific Conference Series, 1999.

[11] D. McConnell, "Basics of interferometry", http://www.atnf.csiro.au/whats_on/workshops/synthesis2003/talks/basics.pdf, 2003.

[12] R. A. Perley, "Basics of interferometry", http://www.aoc.nrao.edu/events/synthesis/2002/PerleyFundamentals.pdf, 2002.

[13] J. A. Kong, *Electromagnetic Wave Theory*, chapter 4, pp. 509–510, EMW Publishing, Cambridge Massachusetts, USA, 2000.

[14] A. R. Thompson, "Fundamentals of radio interferometry", in *Synthesis Imaging in Radio Astronomy II*, G. B. Taylor, C. L. Carilli, and R. A. Perley, Eds., vol. 180 of *A Series of Books on Recent Developments in Astronomy and Astrophysics*, lecture 2, pp. 11–36. Astronomical Society of the Pacific Conference Series, 1999.

[15] S. L. Salas, *Salas and Hille's Calculus: One and Several Variables*, chapter 12, pp. 801–802, John Wiley & Sons, Inc., seventh edition, 1995.

[16] A. R. Thompson, J. M. Moran, and Jr. G. W. Swenson, *Interferometry and Synthesis in Radio Astronomy*, chapter 3, John Wiley & Sons, Inc., 2nd edition, 2001, 68-85.

[17] F. Vincent, "Positional astronomy - index", http://star-www.st-and.ac.uk/~fv/webnotes/index.html, 2003.

[18] D. E. Gary, "Radio astronomy lecture #6", http://physics.njit.edu/~dgary/728/Lecture6.html, 2004.

[19] A. R. Thompson, J. M. Moran, and G. W. Swenson Jr., *Interferometry and Synthesis in Radio Astronomy*, chapter 5, John Wiley & Sons, Inc., 2nd edition, 2001, 122-167.

[20] D. S. Briggs, F. R. Schwab, and R. A. Sramek, "Imaging", in *Synthesis Imaging in Radio Astronomy II*, G. B. Taylor, C. L. Carilli, and R. A. Perley, Eds., vol. 180 of *A Series of Books on Recent Developments in Astronomy and Astrophysics*,

lecture 7, pp. 127–149. Astronomical Society of the Pacific Conference Series, 1999.

[21] B. T. Walkenhorst, "Development of a radio telescope receiver for research in radio frequency interference mitigation", Master's thesis, Brigham Young University, August 2001.

[22] C. K. Hansen, "Beamforming techniques and interference mitigation using a multiple feed array for radio astronomy", Master's thesis, Brigham Young University, April 2004.

[23] T. Cornwell, R. Braun, and D. S. Briggs, "Deconvolution", in *Synthesis Imaging in Radio Astronomy II*, G. B. Taylor, C. L. Carilli, and R. A. Perley, Eds., vol. 180 of *A Series of Books on Recent Developments in Astronomy and Astrophysics*, lecture 8, pp. 151–170. Astronomical Society of the Pacific Conference Series, 1999.

[24] J. M. Wrobel and R. C. Walker, "Sensitivity", in *Synthesis Imaging in Radio Astronomy II*, G. B. Taylor, C. L. Carilli, and R. A. Perley, Eds., vol. 180 of *A Series of Books on Recent Developments in Astronomy and Astrophysics*, lecture 9, pp. 171–186. Astronomical Society of the Pacific Conference Series, 1999.

[25] D. M. Pozar, *Microwave Engineering*, chapter 10, pp. 547–559, John Wiley & Sons, Inc., second edition edition, 1998.

[26] C. K. Hansen, K. F. Warnick, B. D. Jeffs, J. R. Fisher, and R. Bradley, "Interference mitigation using a focal plane array", future publication for Radio Science.

[27] A. V. Oppenheim and A. S. Willsky with S. H. Nawab, *Signals and Systems*, chapter 7, pp. 514–520, Prentice-Hall, Inc, 1997.

[28] E. B. Fomalont and R. Perley, "Calibration and editing", in *Synthesis Imaging in Radio Astronomy II*, G. B. Taylor, C. L. Carilli, and R. A. Perley, Eds., vol. 180

of *A Series of Books on Recent Developments in Astronomy and Astrophysics*, lecture 5, pp. 79–110. Astronomical Society of the Pacific Conference Series, 1999.

[29] J. Raza, A-J Boonstra, and A-J van der Veen, "Spatial filtering of rf interference in radio astronomy", *IEEE Signal Processing Letters*, vol. 9, no. 2, pp. 64–67, Februrary 2002.

[30] A. Leshem, A-J. van der Veen, and A.J. Boonstra, "Multichannel interference mitigation techniques in radio astronmy", *Astrophysics Journal Supplement*, vol. 131, no. 1, pp. 355–374, November 2000.

[31] "Glonass", `http://www.glonass-center.ru/`.

[32] N. Dahnoun, *Digital Singal Processing Implementation using the TMS320C6000 DSP Platform*, Prentice Hall, 2000.