

SEQUENTIAL BEAMSPACE SMART ANTENNA SYSTEM

by

William Graves Tidd

A thesis submitted in partial fulfillment
of the requirements for the degree

of

Master of Science

in

Electrical Engineering

MONTANA STATE UNIVERSITY
Bozeman, Montana

May 2011

© Copyright

by

William Graves Tidd

2011

All Rights Reserved

APPROVAL

of a thesis submitted by

William Graves Tidd

This thesis has been read by each member of the thesis committee and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to The Graduate School.

Dr. Richard S. Wolff

Approved for the Department of Electrical and Computer Engineering

Dr. Robert C. Maher

Approved for The Graduate School

Dr. Carl A. Fox

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Montana State University, I agree that the Library shall make it available to borrowers under rules of the Library.

If I have indicated my intention to copyright this thesis by including a copyright notice page, copying is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for permission for extended quotation from or reproduction of this thesis in whole or in parts may be granted only by the copyright holder.

William Graves Tidd

May 2011

DEDICATION

I dedicate this to my loving fiancée Sarah. THANK YOU for all your patience in the face of perpetual dilatoriness.

ACKNOWLEDGEMENTS

I would like to thank my principle advisor, Dr. Yikun Huang, for the opportunity to investigate and pursue what could have been an insignificant white board notion. Her tutelage and foresight during this project were essential in its completion. I would also like to thank my committee chair Dr. Richard Wolff and committee members Mr. Andy Olson and Dr. Brock LaMeres for their abyss of experience and knowledge, as well as their guidance to provide an optimal path to success.

Lastly, I would like to give special thanks to Raymond Weber, Dr. Ahmed Khallaayoun, and Yufei Zhao for all their invaluable research, assistance and countless contributions to this project.

This work is supported by the Montana Board of Research and Commercialization Technology program under grant MBRCT #09-09 and by Advanced Acoustic Concepts in Bozeman, Montana.

TABLE OF CONTENTS

1. INTRODUCTION	1
1.1 Adaptive Smart Antenna Systems	1
1.2 High Resolution DOA Estimation	2
1.3 Beamforming	4
1.4 Sequential BeamSpace (SBS).....	6
1.4.1 Proposed Application.....	8
1.4.2 Advantages	9
1.4.3 Disadvantages	10
1.5 Notation.....	11
2. SYSTEM OVERVIEW.....	12
2.1 Hardware Design.....	13
2.1.1 UCA Antenna	13
2.1.2 Beamformer Board	14
2.1.3 Single Channel Receiver	17
2.2 Software Design	18
2.2.1 Firmware	19
2.2.2 MATLAB DSP.....	21
2.2.3 PC Control Interfaces	22
3. SEQUENTIAL BEAMSPACE: MODEL, SIMULATIONS, AND METHODS .	26
3.1 Data Model	26
3.2 High Resolution DOA Estimation	31
3.2.1 Element space vs. BeamSpace.....	31
3.2.2 Switched Beam vs. BeamSpace	35
3.2.3 BeamSpace Algorithms.....	36
3.2.3.1 BeamSpace Capon	36
3.2.3.2 BeamSpace MUSIC.....	37
3.2.3.3 UCA Real BeamSpace MUSIC.....	38
3.2.3.4 Simulation Study.....	39
3.2.4 Obstacles and Solutions	40
3.2.4.1 Sampling Phase Error.....	41
3.2.4.2 Time Delay Correction Filter	43
3.2.4.3 Spatial Search Methods and Simulations.....	46
3.3 Capon Beamforming	50
3.3.1 Algorithm	51
3.3.2 Simulation Study	53
3.4 Calibration	57

TABLE OF CONTENTS – CONTINUED

4. EXPERIMENTAL RESULTS	61
4.1 Lab Setup and Testing Procedure	61
4.1.1 Equipment	62
4.1.2 Procedure	63
4.2 Synchronization Validation	65
4.3 DOA Estimation	67
4.3.1 Switched Beam.....	67
4.3.2 One Source.....	68
4.3.3 Two Sources.....	70
4.3.4 Minimum Detectable Signal.....	75
4.3.5 Rough Sector Search.....	77
4.3.6 Fine Sector Search.....	79
4.3.7 DOA Estimation Summary.....	81
4.4 Beamforming	82
4.4.1 Shift and Sum Beamforming.....	82
4.4.2 Capon Beamforming	84
4.4.3 Beamforming Summary.....	87
5. FUTURE WORK	88
6. CONCLUSIONS	90
REFERENCES CITED.....	91
APPENDICES	95
APPENDIX A: VHDL Code	96
APPENDIX B: MATLAB Code	106

LIST OF TABLES

Table		Page
3.1	Wide beam windowing functions with UCA $\beta\rho = 3.05$	47
3.2	Narrow beam windowing functions with UCA $\beta\rho = 3.05$	48
4.1	SBS sensitivity according to DOA accuracy error	77
4.2	SBS RMSE accuracy result summary.....	82
4.3	Shift and sum and Capon beamforming result summary	87

LIST OF FIGURES

Figure	Page
1.1 Adaptive smart antenna system (ASAS) diagram [1].....	2
1.2 DOA estimation approaches.....	3
1.3 Beamformer output is the superposition of the antenna outputs [2]	5
1.4 Sequential beamspace (SBS) DOA estimation approach	7
1.5 SBS data acquisition flow example	8
2.1 SBS system block diagram for beamspace DOA estimation and beamspace beamforming	12
2.2 8 element uniform circular array.....	14
2.3 Beamformer photo.....	15
2.4 Beamformer block diagram	16
2.5 Single channel receiver block diagram	17
2.6 Single channel receiver photo (ADC not pictured)	18
2.7 Software interaction diagram.....	19
2.8 Commutation rate timing diagram example	21
2.9 Empirical weight measurement GUI.....	24
2.10 Sequential beamspace GUI.....	25
3.1 Element space vs. beamspace DOA accuracy and resolution simulations .	33
3.2 Element space vs. beamspace DOA probability of resolution simulations	34
3.3 Beamspace DOA accuracy simulation	40
3.4 Beamspace DOA resolution simulations	40
3.5 Unsynchronized and synchronized DOA estimation comparison	44
3.6 Time delay correction filter	45
3.7 Beamspace DOA accuracy using different windowing functions.....	47
3.8 Rough and fine sector beam fans	49
3.9 DOA using rough and fine sector searching	50

LIST OF FIGURES – CONTINUED

Figure	Page
3.10 Original transmitted signals, a sine wave and square wave.....	53
3.11 Beamspace Capon spatial pseudo-power spectrum	54
3.12 Beamformed received signals	55
3.13 Beamformers.....	56
3.14 Empirical weight calibration simulation	59
3.15 Capon spectrum calibration simulation	60
4.1 Lab setup	63
4.2 Example 1: synchronized and unsynchronized comparison.....	65
4.3 Example 2: synchronized and unsynchronized DOA comparison	66
4.4 Switched beam DOA accuracy	68
4.5 SBS DOA accuracy with 1 source at 81 KHz.....	69
4.6 SBS DOA accuracy with 1 source at 873 KHz	69
4.7 SBS spatial pseudo-power spectra with 1 source at 81 KHz	70
4.8 SBS spatial pseudo-power spectra with 1 source at 873 KHz	70
4.9 SBS DOA accuracy with 2 sources spaced by 300 KHz	71
4.10 SBS DOA accuracy with 2 sources spaced by 50 KHz	72
4.11 SBS spatial pseudo-power spectra for 2 sources	73
4.12 SBS spatial pseudo-power spectra for 2 sources	74
4.13 SBS sensitivity according to DOA accuracy error	76
4.14 Measured Hann beam patterns: 4 beam fan	77
4.15 SBS DOA accuracy using a rough sector search.....	78
4.16 SBS spatial pseudo-power spectra for a rough sector search.....	79
4.17 Measured orthogonal shift and sum beam patterns: 8 beam fan	80
4.18 SBS accuracy using a fine sector search.....	80
4.19 SBS spatial pseudo-power spectra for a fine sector search.....	81

LIST OF FIGURES – CONTINUED

Figure		Page
4.20	Spatial pseudo-power spectrum for 2 sources	83
4.21	Shift and sum beamforming patterns for 2 sources.....	84
4.22	Measured and ideal output waveforms using shift and sum beamforming	84
4.23	Capon spatial pseudo-power spectrum for 2 sources.....	85
4.24	Capon beamforming patterns for 2 sources.....	86
4.25	Measured and ideal output waveforms using Capon beamforming.....	86

ABSTRACT

This thesis proposes a design of a novel and innovative sequential beamspace (SBS) smart antenna system. The system is capable of accurate direction of arrival (DOA) estimation in beamspace and efficient beamforming. Moreover, the robust functionality of such a system includes high resolution radio frequency (RF) emitter DOA estimation and beamforming in a noisy environment in the presence of strong interference. Simulations for DOA estimation using beamspace MUSIC and beamspace Capon methods are presented in conjunction with Capon beamforming. These methods are compared and contrasted with proven element space DOA estimation techniques to demonstrate the validity and advantages of pursuing a SBS smart antenna for real-world applications. The beamspace DOA estimation accuracy, resolution, beamforming pattern, and output signal quality have been thoroughly studied and quantified. The algorithms have been tailored to utilize an 8 element uniform circular array (UCA) and an 8 channel analog beamformer (BF) operating at 5.8 GHz to gather lab-based experimental results. The simulations and experimental results show that the proposed system can achieve good performance once it is properly synchronized using a time delay correction filter. In addition, a significant decrease in hardware is realized when operating in beamspace versus element space.

CHAPTER 1

INTRODUCTION

1.1 Adaptive Smart Antenna Systems

Adaptive Smart Antenna Systems (ASAS) have been considered for many years for their advantages in saving energy and extending range. The kernel of such a system is to shape and steer the incoming or outgoing electromagnetic energy to boost such parameters as system capacity, efficiency, range, signal-to-noise ratio (SNR), throughput, etc. ASAS capitalize with these improvements through the use of three functional blocks: radio unit, beamforming unit, and adaptive processing unit. The radio unit consists of an antenna array of various geometries, e.g. linear or circular, and receivers (RX), consisting of down-converting circuitry to mix the signal to an intermediate frequency (IF) suitable for sampling and analog-to-digital converters (ADC) to perform the sampling. The beamforming unit is responsible for forming a beam in a desired direction using weights. Weights are the amplitude and phase adjustments applied to the signal, thus becoming the terminal currents of the array elements. The adaptive processing unit is responsible for calculating the correct weights for the beamforming unit using different methods and algorithms, e.g. least-mean-square (LMS) [1–3]. A general ASAS diagram is presented in figure 1.1.

The ASAS used in the completion of this work includes an 8 element uniform circular array (UCA) operating at 5.8 GHz. An 8 channel analog beamformer (BF) is used as the beamforming unit to correctly set the weights. A single channel down-converter was fabricated for this project, and the ADCs are commercial-of-the-shelf (COTS) units from National Instruments (NI). The adaptive processing unit is a com-

pilation of software executed on a personal computer (PC) and is under construction for continual improvement [3].

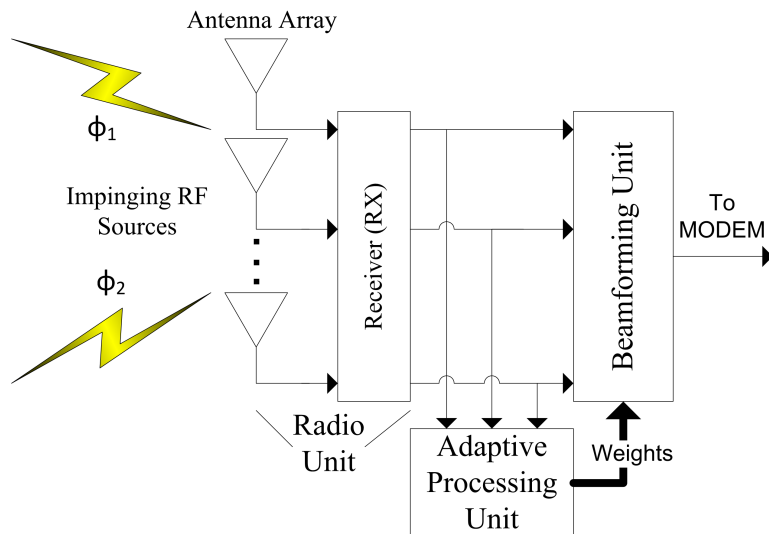


Figure 1.1: Adaptive smart antenna system (ASAS) diagram [1]

1.2 High Resolution DOA Estimation

In the context of this work, direction of arrival (DOA) estimation deals with source direction or angle finding of an electromagnetic wave impinging upon an array of antennas. This spatial spectrum estimation provides source localization and tracking capabilities used in a wide variety of military and civilian applications. Two fundamental approaches to DOA estimation use data in element space or data in beamspace for DOA computation. The classic element space DOA estimation approach is shown in figure 1.2(a), and the conventional beamspace DOA estimation approach is shown in figure 1.2(b) [3, 4]. Element space data is information taken directly from the antenna output, while beamspace data is a linear transformation of element space data, typically to a lower dimensional beamspace for less expensive computation. Element space observation data is projected into a beamspace subspace

through the use of a matrix beamformer in two ways: via software or hardware. Essentially, the transformation from element space to beamspace can be interpreted as spatial filtering via the matrix beamformer transfer function. In software this is done with a numerical matrix beamformer, where each column is a beamformer assigned to cover a sector of locations and the sources within, while attenuating out of sector sources. In hardware this is done with a matrix beamformer module capable of physically imposing amplitude and phase changes on multiple element space channels and summing together for a single output [1, 5, 6].

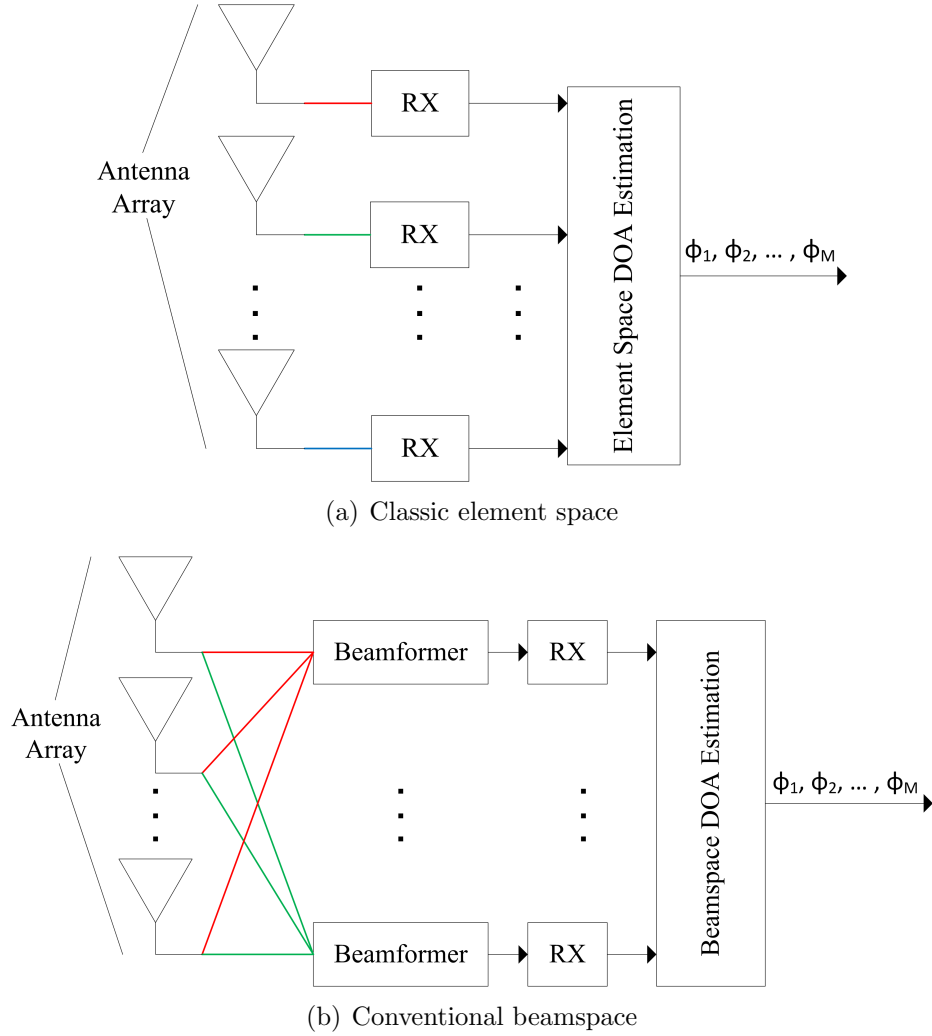


Figure 1.2: DOA estimation approaches

There are several well known, highly accurate, and high resolution (hi-res) DOA algorithms with the ability to discern two closely located sources, such as Multiple Signal Classification (MUSIC) [7] and Capon [8], that can be used in conjunction with a UCA. Through previous simulation studies at MSU presented in [4] it has been proven that MUSIC and Capon have superior accuracy and resolution results compared to other DOA algorithms. Both MUSIC and Capon were implemented in element space to provide computational flexibility and a degree of algorithm choice when confronted with closely separated sources. Also, both algorithms can be modified to compute DOA estimations with element space or beamspace data. Furthermore, using these DOA estimation algorithms to process beamspace data, collected with the new proposed technique, is a focus of this work [3, 9, 10].

1.3 Beamforming

In many cases, sources that are close or overlapping in frequency may originate from different directions. To distinguish these sources, the spatial separation can be exploited with the use of a spatial filter such as a beamformer. In a conventional beamforming system a beam is formed in a particular direction by weighting and combining the antenna array element responses to emphasize a source from a desired direction and/or de-emphasize an interferer from other directions, as shown in figure 1.3. Depending on the method, weights steer the main beam with a fixed beam pattern or a dynamic beam pattern [2, 11].

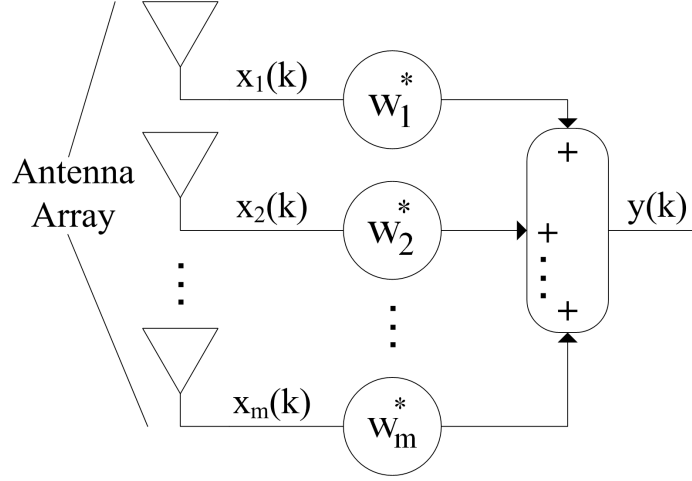


Figure 1.3: Beamformer output is the superposition of the antenna outputs [2]

There are three main beamforming method classifications. First, data independent, where the classic shift and sum or phased array method is employed. Multi-channel signals are shifted in phase and then summed together at the system output. This is the simplest beamforming method: very robust, minimal computational complexity, easy to operate, but accompanied by high side lobe levels. An extension of classic beamforming is windowing, where beam patterns are generated by window or taper functions, e.g. Chebyshev and Hann, to suppress side lobe levels by amplitude tapering. Second, statistically optimum beamforming is where an optimal spatial filter, such as Capon or maximum SNR, is used to form a beam in the desired direction and suppress power from all other directions based on the statistics of the data. Third, adaptive beamforming is a closed-loop recursive approach to calculating optimal weights for beam and null steering when data statistics are unknown. The data independent method is constrained by a fixed beam pattern with a wide half-power beamwidth (HPBW), so it suffers performance degradation from strong interference, especially if located within the main beam. Optimal beamforming requires an accurate DOA of the incoming desired signal using a separate element space DOA

estimator, and adaptive beamforming may suffer from a long convergence time or sub-optimal convergence to local maximums instead of a global maximum. Because of these issues the need arises for Capon beamforming in beamspace. This extracts an optimal Capon beamformer from beamspace data using a beamspace DOA estimator. This is different from conventional beamforming in that this is optimal beamforming stacked upon classic beamforming – beamspace data created using classic shift and sum weights. Beamforming is usually performed strictly in element space with good results, yet this work proposes to determine element space weights using beamspace beamforming methods, via the novel technique proposed in the next section [2–4].

1.4 Sequential Beamspace (SBS)

Executing an element space approach to DOA estimation requires access to each individual channel, which must be conditioned and sampled separately. For an element space approach to be successfully realized at MSU, an 8 channel receiver was developed in hardware alongside the beamformer, providing two essential and separate blocks for the ASAS: a multichannel receiver for DOA estimation and a beamformer for weighting transmitted and received radio signals [12]. In this research, a different avenue was explored, simplifying the current system and removing superfluous hardware. A novel DOA method is proposed: sequential beamspace (SBS) DOA estimation and beamforming using the 8 channel analog beamformer with a single receiver, shown in figure 1.4. The unique SBS approach presented herein is a combination of the classic element space approach and the conventional beamspace approach, combining the better aspects of both. SBS combines hi-res DOA estimation and optimal beamforming on a single hardware platform. The modularity and independence of the system allows it to be easily retrofitted to existing conventional phased

array systems in use. Also, the unit is streamlined for minimal control input/output (IO), allowing it to attach to a COTS radio with relative ease. A basic upgrade to an existing phased array system would only require a software/firmware change to realize the SBS technique, and a COTS radio would only need to provide a Transmit/Receive (Tx/Rx) signal for communication synchronization for immediate compatibility [3].

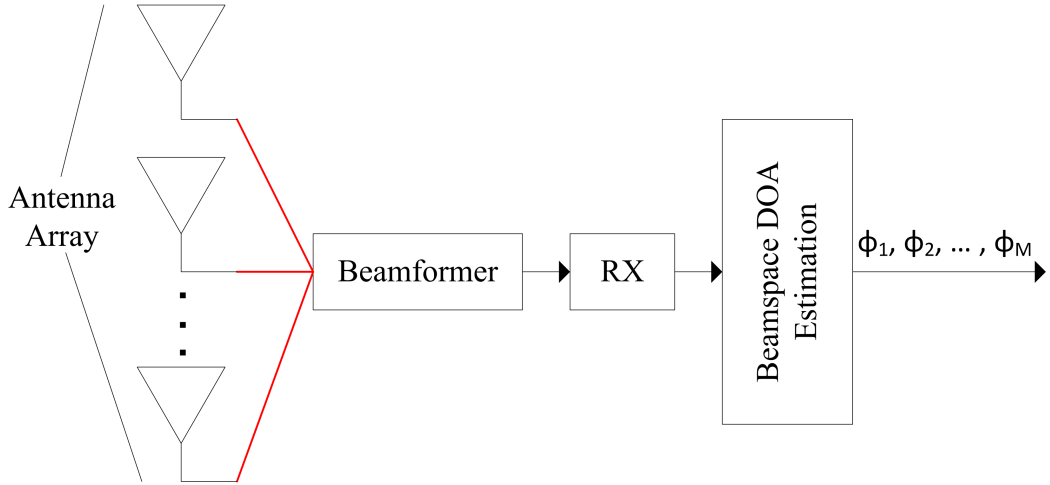


Figure 1.4: Sequential beamspace (SBS) DOA estimation approach

The SBS technique is a beamformer that utilizes beamspace data for DOA estimation and optimal beamforming, but only forms one beam at a time. The UCA with beamformer can form multiple beams, sequentially providing comprehensive 360° search coverage. This 2-D azimuth space is divided into a number of switched beams by selecting the appropriate weights, which define the bore sight pointing direction, beam shape, and side lobe level of said beams. The SBS technique then logs each single beamformed response in memory and switches to the next desired beam in a sequential manner since only one set of beamforming hardware is used. A data acquisition flow example for SBS is presented in figure 1.5. Once all beam responses are captured, a beamspace observation matrix is composed of all sampled beams and the data is processed using DOA algorithms suited for beamspace data. Popular

DOA estimation algorithms, like MUSIC or Capon, can be employed to yield hi-res DOA estimations. These DOA estimations can then be used to implement beamspace Capon beamforming to improve signal quality [3].

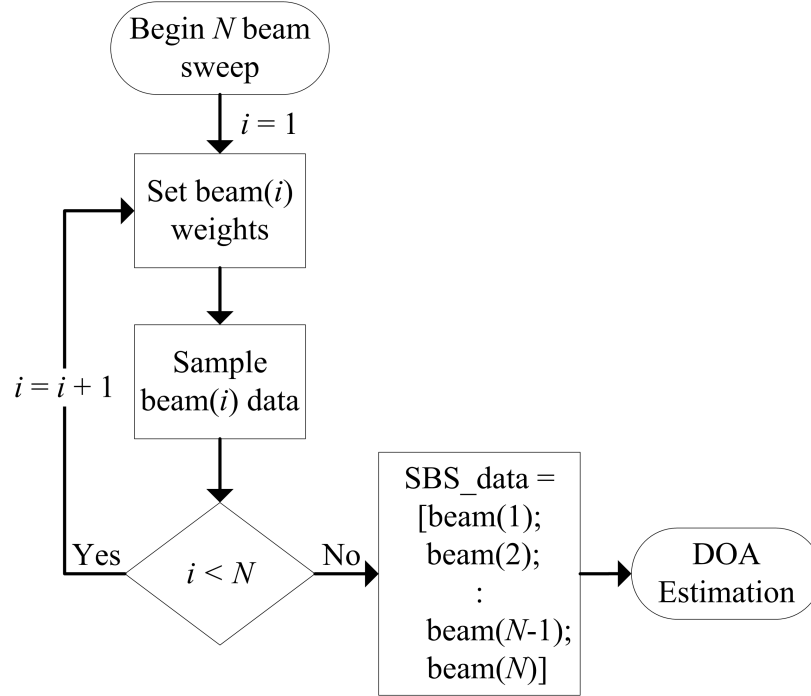


Figure 1.5: SBS data acquisition flow example

1.4.1 Proposed Application

The proposed application would upgrade the current beamformer board to a hi-res DOA estimation and beamforming device using beamspace Capon beamforming techniques. In contrast to an element space approach, the multichannel receiver would be eliminated by paring down the current eight channel receiver to a single channel receiver – the only additional hardware required. In contrast to a conventional beamspace system, a single beamformer unit is required instead of multiple beamformer units, which can be impractical for many ASAS applications. The upgrade will also consist of alterations and augmentations of the existing firmware inside the

field programmable gate array (FPGA) controller, located on the beamformer board, to provide SBS functionality.

1.4.2 Advantages

The five advantages associated with SBS can be seen best in relation to conventional beamspace and element space approaches. Conventional beamspace requires multiple sets of beamformer hardware to simultaneously form beams and collect data [13, 14]. However, the novelty of the SBS technique in hardware is that data is sampled on a single receiver at the output of a single beamformer instead of sampling data concurrently on multiple beamformers and receivers. Each beam provides a vector of beamspace data similar to how each antenna element provides a vector of element space data. First, the SBS configuration is advantageous because it eliminates the need for separate multichannel receiver hardware for hi-res DOA estimation; thus, a great portion of hardware can be saved. Essentially, the reduction in size, cost, and complexity of the hardware can be quantified by a factor, $\frac{1}{m}$, where m is the number of antenna elements or receiver channels in element space. Next, the configuration of the hardware provides an inherent beamspace manifold (spatial signature) immediately and automatically yielding beamspace data. The data no longer has to be transformed to beamspace further down the DSP chain. Third, applying optimal beamspace Capon beamforming can help suppress interference and recover original signals from other closely located signal sources and noise. Fourth, the beamformer now offers threefold functionality: communication as a highly focused active antenna using switched beam or optimal beamforming, hi-res DOA estimation, and in the near future, tracking. Last, for an m -channel beamformer the spatial filter realizes an m factor, or $10 \times \log_{10}(m)$ decibel (dB), gain advantage operating in beamspace over element space because of the summing of m channels into one,

physically being combined on the beamformer hardware. Being antenna gain this directly translates to an increase in SNR, making it ideal for working in long range or low SNR environments [3].

1.4.3 Disadvantages

Two disadvantages associated with the SBS technique include a lengthened data acquisition time and sampling phase error; both can cause a decorrelation of the observed data which is detrimental to accurate DOA estimation. The key implementation process yet most significant hazard of SBS is the sequential beamforming and data collection.

A typical element space or conventional beamspace unit may collect K samples in t seconds; however, for the same frequency or spatial resolution in SBS, collecting K samples will take $N \times t$ seconds at a minimum, assuming the beam switching time is zero. Therefore, the data acquisition time is directly proportional to the number of beams in use, N . It is important to minimize the data acquisition time to be smaller than or on the order of the channel coherence time, or the DOA estimation will be corrupted by channel fading. Moreover, if the channel is stationary (time invariant) then acquisition time is moot; however, if the channel properties are changing faster than the total acquisition time, then complete decorrelation occurs, and the SBS method will fail. This effect is something that will be studied in future work and is beyond the scope of this thesis. So, to minimize acquisition time an initial weighting configuration must be used to search 360° efficiently, by maximizing coverage and minimizing the number of beams with an acceptable variance of DOA accuracy. For example, using a popular Hamming window function to weight the channels produces an azimuth HPBW of 74° , allowing for the use of 5 beams to scan 360° while only

sacrificing 0.54° of DOA root mean square error (RMSE) accuracy, defined in eq. 3.20, at 0 dB SNR.

Next, also because of the sequential manner of operation, the data from beam to beam becomes less correlated, which manifests as sampling phase error. Signal phase is paramount in determining the correct DOA, so inclusion of sampling phase error can potentially cause severe degradation in the DOA estimation accuracy and resolution. Ultimately, the SBS signal resulting from the beam fan (all beams that sufficiently cover the desired spatial sector) must be synchronized to a single reference beam using a time delay correction filter. This is a fundamental feature of SBS which will be described in detail in chapters 2 and 3. Overall, for the purposes of this work channel effects are ignored, and only sampling phase errors are considered.

1.5 Notation

The mathematical notation used throughout this work strives to be consistent and coherent for ease of comprehension. Scalars are indicated by lower and upper-case *Italic* font, e.g. a and A , while vectors and matrices are indicated by Roman bold font, lower and uppercase respectively, e.g. $\mathbf{a}$ and $\mathbf{A}$. The superscript $(\cdot)^T$ denotes a transpose, while $(\cdot)^H$ denotes a Hermitian transpose, or complex conjugate transpose. Also, the superscript $(\cdot)^*$ denotes a complex conjugate. The symbol, $\times$, is used for multiplication and vector and matrix dimensions. Subscripts are reserved for distinguishing between two similarly named variables, which may result from applying a different algorithm or function to arrive at the same place in computation. Last, the hat symbol, $(\hat{\cdot})$, specifies an estimated variable, distinguishing between a finite sampled value taken in the lab and an exact quantity from an assumed infinite observation time [15].

CHAPTER 2

SYSTEM OVERVIEW

The system running the proposed SBS technique is an amalgam of hardware and software. The hardware has three distinct components including a UCA antenna, a beamformer board, and a single channel receiver. The software is a combination of firmware, written using very-high-speed integrated circuit hardware description language (VHDL), digital signal processing (DSP) MATLAB code, and LabVIEW Virtual Instruments (VIs) and MATLAB interfaces used for high-level control and data acquisition. The UCA and beamformer were preexisting, developed for a related project facilitated by the author, while the single channel receiver and all software components were devised and built specifically for this project. The system block diagram for SBS is shown in figure 2.1.

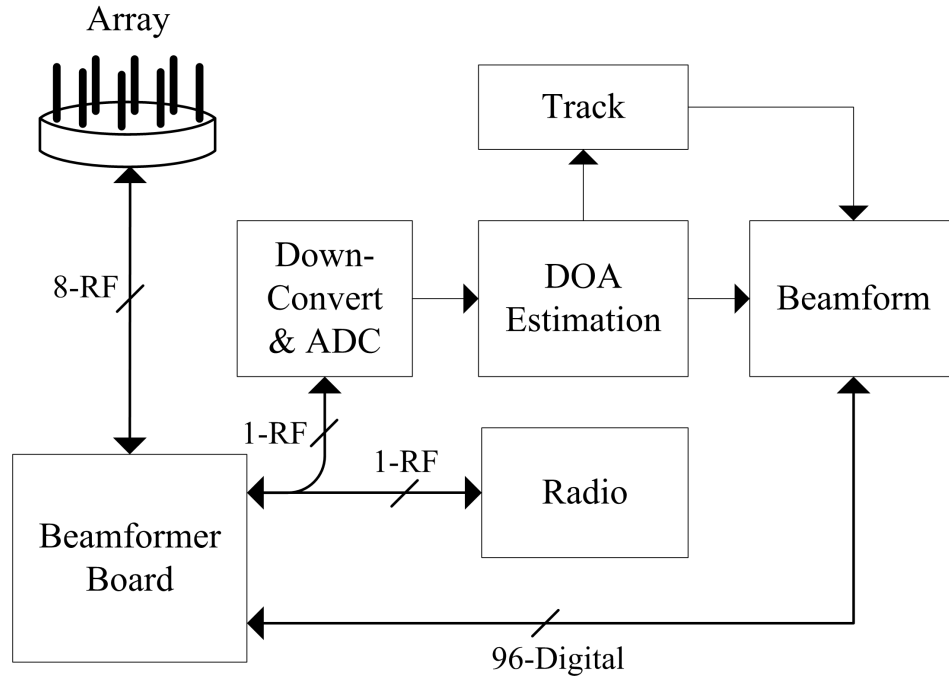


Figure 2.1: SBS system block diagram for beamspace DOA estimation and beamspace beamforming

2.1 Hardware Design

The hardware has been an ongoing project, initiated in 2004, with critical revisions and upgrades. The hardware consists of a UCA with ground skirt, a beamformer board with real-time phase and amplitude control, and a single channel receiver used for filtering and down-converting the beamspace signal to baseband. The antenna and beamformer were designed with a compact cylindric form factor in mind.

2.1.1 UCA Antenna

The UCA antenna pictured in figure 2.2(a) is an 8 element array used to transduce an RF signal into/from the air for communication purposes. The array is composed of two functional blocks: the thin-wire monopole apertures and the ground plane with ground skirt. The array samples the propagating wave field in space with minimal attenuation, distortion, and dispersion. The ground plane and skirt provide a virtual infinite ground plane and mechanical rigidity. The array has an operating center frequency of 5.8 GHz and spatially covers 360° in the azimuth plane. The array electrical size is $\beta\rho = 3.05$, where ρ is the array radius of 2.54 cm, $\beta = \frac{2\pi}{\lambda}$ is the wave number, and λ is the wavelength. Inter-element spacing is 0.375λ and the antenna element length is 0.23λ . The virtual infinite ground plane is the result of the optimal ground skirt length of 0.25λ and is crucial in forming a uniform vertical beam with a minimum upward tilt of 19° , relative to the ground plane, and an elevation HPBW of 66° . The array gain is 12-15 dBi depending on the window algorithm used in weighting the beamformer channels. The operating bandwidth of 200 MHz centered about 5.8 GHz has relatively flat gain; therefore, the array easily accommodates COTS radio channels from such standards as Wi-Fi and WiMAX. Last, the photo in figure 2.2(b)

shows the array with a weatherproof radome that protects the antenna elements while imposing minimal signal absorption and refraction [2, 16].

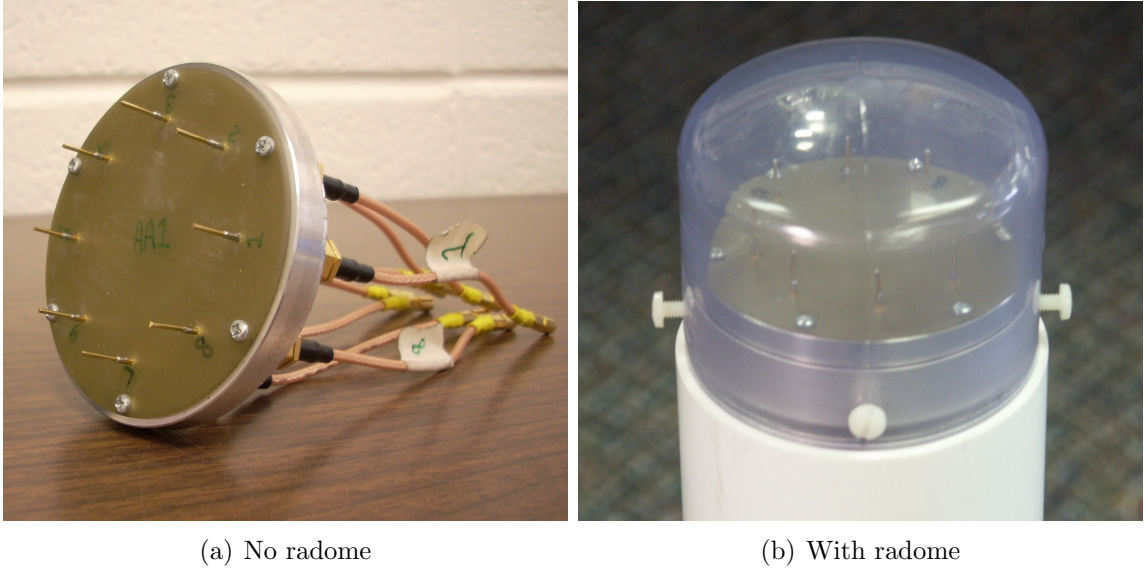


Figure 2.2: 8 element uniform circular array

2.1.2 Beamformer Board

The beamformer board, a photo in figure 2.3 and block diagram in figure 2.4, is an 8 channel digitally controlled analog beamforming system that controls phase and amplitude of each channel to form a beam in the desired direction. A beamformer usually operates in Tx or Rx mode only, but this beamformer operates in both modes. The beamforming system consists of five functional blocks: controller, analog beamforming circuitry, radio interface, power detection, and power regulation. The controlling unit provides precise digital control and ancillary functionality via an FPGA (Xilinx Spartan-3). The FPGA delivers 96 parallel control lines to eight 6-bit phase shifters (MAPCGM0002) and eight 6-bit attenuators (HMC425LP3) in no more than 100 ns of combined propagation and transmission delay time. The analog beamforming circuitry consists of an 8-way power divider/combiner and eight

subsequent channels with the following components: a variable attenuator with 31.5 dB of range and 0.5 dB resolution, a phase shifter with 360° of range and 5.6° resolution and an RMS phase error of 9° , two Tx/Rx single-pole double-throw (SPDT) switches (HMC536MS8G) with a speed of 15 ns, a power amplifier (SST11LP12) and low noise amplifier stage (TQL5000 followed by HMC320MS8G), and bandpass filters (ELB2A005). The active circuitry is duplicated eight times to provide an exact analog path to minimize phase and amplitude error. The channel path yields a radio frequency (RF) processing gain in Rx mode of 8 dB and 9 dB in Tx mode. Also, the radio interface provides signal conditioning and current driving capabilities of a radio's Tx/Rx signal for frame synchronization [2, 16].

The power detection block, seen in the far left of figure 2.3, is a modular unit that can be added to the beamformer board via a high density connector. It has a square law detector to sense RF power and an ADC to quantize and upload data to the FPGA. This power detector also provides the functionality for switched beam DOA estimation. Finally, a simple power regulation block provides ample linear regulation and energy storage using large valued tantalum capacitors, which minimize switching transients [16].

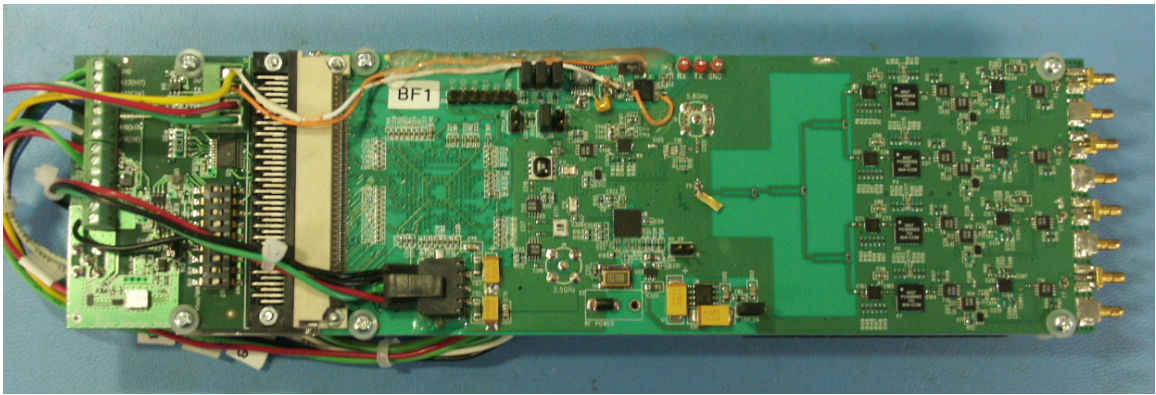


Figure 2.3: Beamformer photo

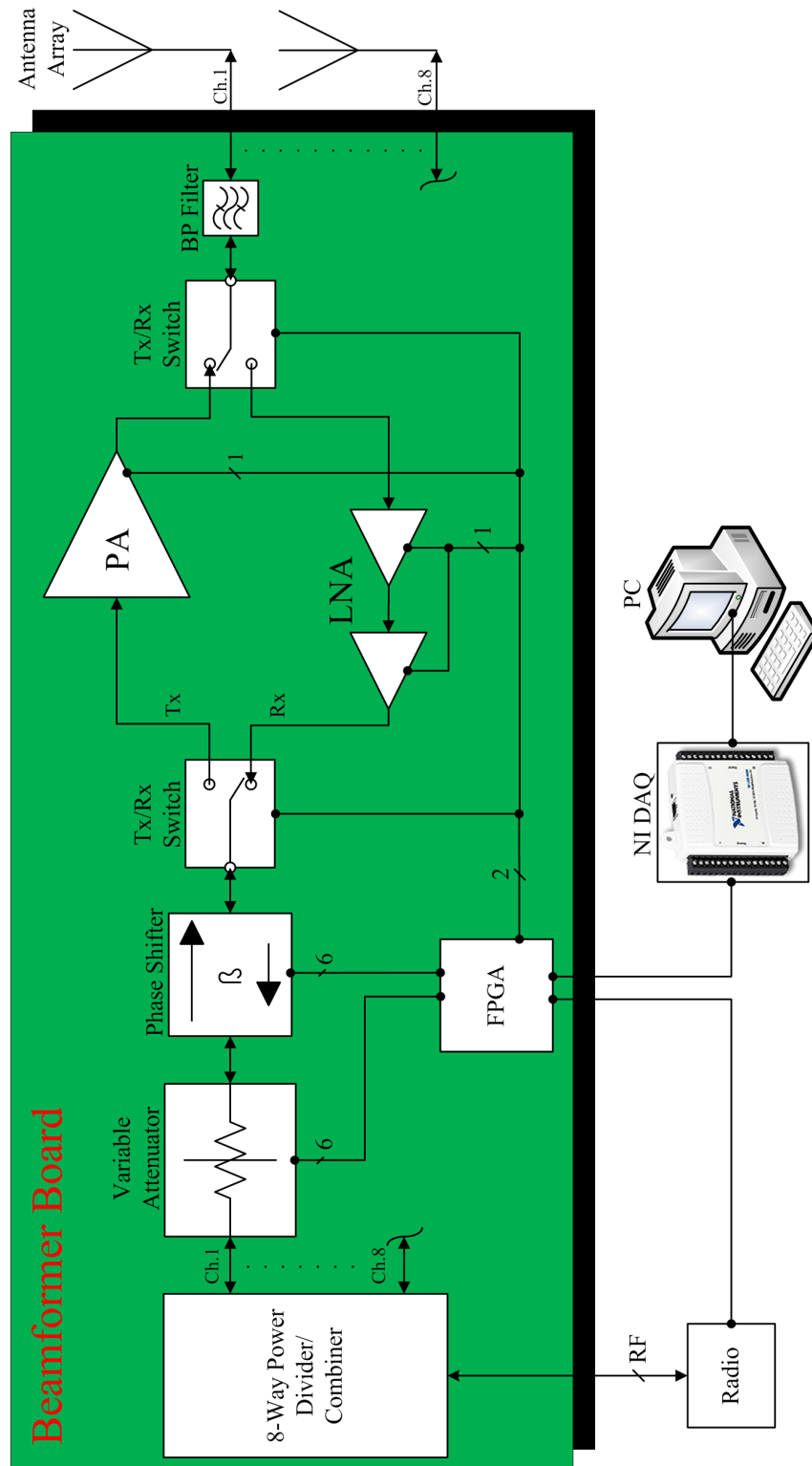


Figure 2.4: Beamformer block diagram

2.1.3 Single Channel Receiver

The single channel receiver, a block diagram in figure 2.5 and photo in figure 2.6, is a connectorized unit fabricated to quickly meet a proof of concept need. Because of its temporary role, tests were not done to characterize the receiver. The primary functions are to down-convert the signal from a high carrier frequency to a low baseband frequency, condition and filter the baseband signal, and then sample the signal for digital processing. The fundamental components include a bandpass filter, a high isolation Marki mixer (M3-0312), a voltage controlled oscillator (VCO) and phase-locked loop (PLL) development kit from Analog Devices (EVAL-ADF4106EB1), a variable gain amplifier (VGA) development kit from Analog Devices (AD8334-EVAL), that provides up to 48 dB of gain control at baseband, and 3 MHz lowpass anti-aliasing filters. The 5.8 GHz beamspace signal enters the receiver, is bandpass filtered, down-converted to 1.25 MHz, amplified, and lowpass filtered. Next, the signal is sampled by a NI Data Aquisition (DAQ) card (PCI-6133) at 2.5 mega-samples per second (MSPS) and passed to the PC for beamspace DOA estimation and beamforming analysis.

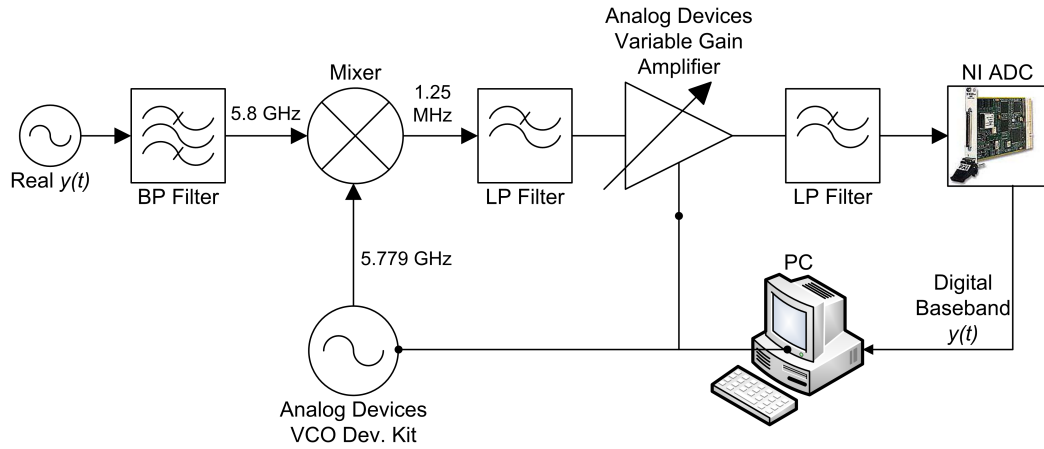


Figure 2.5: Single channel receiver block diagram

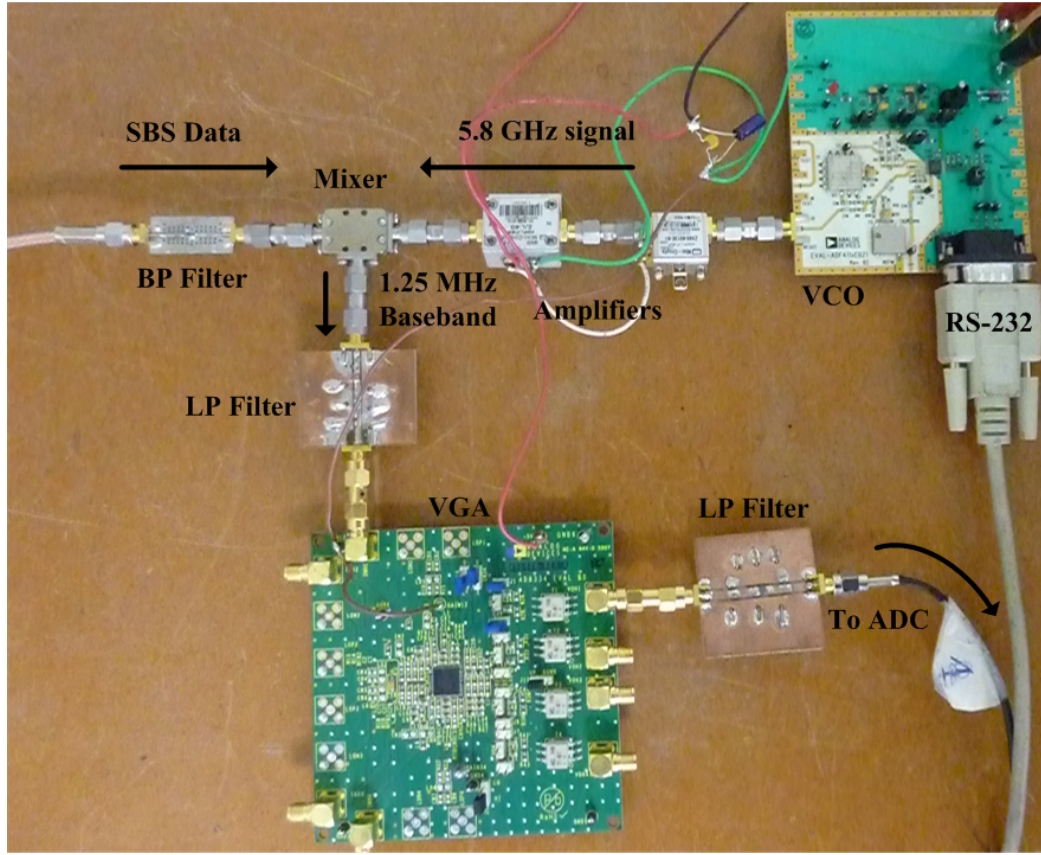


Figure 2.6: Single channel receiver photo (ADC not pictured)

2.2 Software Design

The software design spans three subcategories including firmware to intelligently control the beamformer, MATLAB DSP scripts for evaluating complex beamspace algorithms, and LabVIEW VIs and MATLAB DAQ scripts acting as PC control interfaces for streamlined interaction between user and hardware. Figure 2.7 shows the cooperation and flow between each software module.

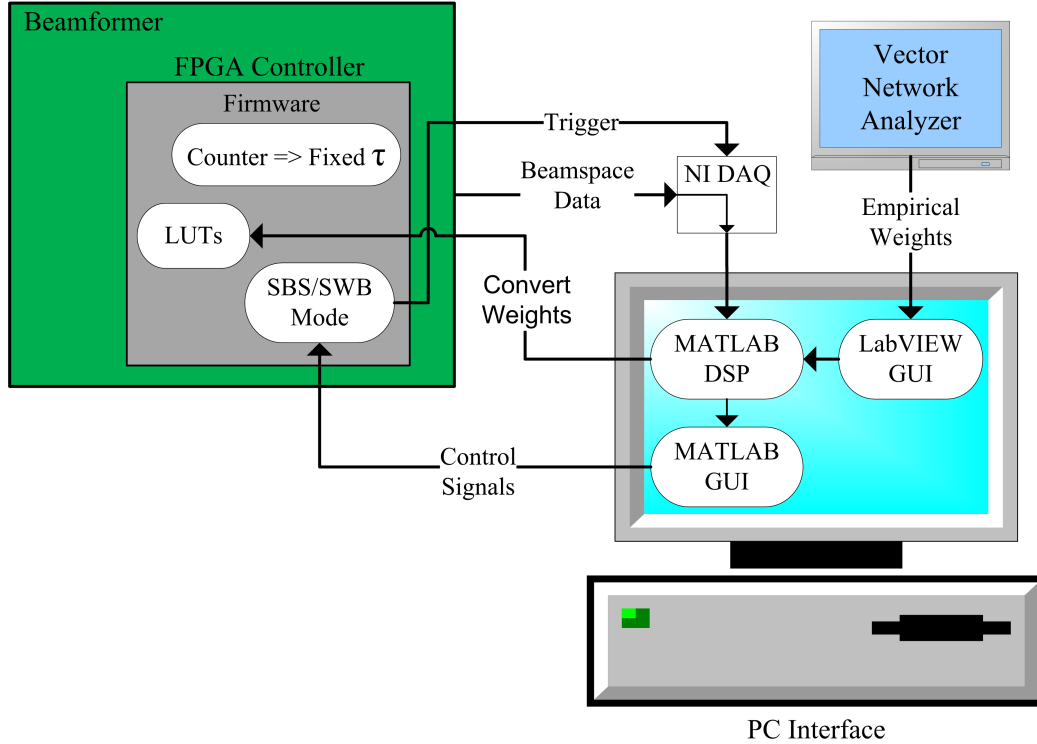


Figure 2.7: Software interaction diagram

2.2.1 Firmware

Custom VHDL software was developed as firmware to reside on the FPGA controller of the beamformer board. The significant portions of code are attached as Appendix A. The firmware performs three main functions. First, it provides a memory platform for storing look-up tables (LUT) of beamforming weights as amplitude and phase coefficients. The tables can encompass weights of numerous different window or taper functions with 3 to 16, but more if desired, beams worth of data. Next, it provides two modes of operation, switched beam (SWB) mode used for communication and beamforming and sequential beamspace (SBS) mode used for DOA estimation and beamforming. SWB mode allows an external hardware platform, e.g. radio, or external algorithm to set the current beam of operation with the indicated weights for wireless communication or beamforming. SBS mode allows for a speedy and

continuous collection of beamspace data using the indicated weights, corresponding to a specific window, and indicated number of beams.

Third, and most important for SBS mode, is the function that sets the beam commutation rate, τ , the sampling and switching time per beam. The known commutation rate is subsequently used for synchronization of data via the time delay correction filter, detailed in section 3.2.4.2. Through the use of a 20-bit internal counter the beam commutation rate is precisely fixed to control the number of samples per beam and to keep the commutation rate constant over the entire beam fan sweep. Upon triggering in SBS mode, a sweep initiates, thereby residing on each beam for a length of time equal to the commutation rate. Furthermore, the commutation rate includes a guard time at the start and end of each beam to negate the FPGA lag between beam switching, voltage transients associated the switching, and the settling time of the attenuators and phase shifters. More specifically, the commutation rate must be set longer than the desired processing sample number so a center window can be extracted from each sampled beam for processing. As an example in figure 2.8, the commutation rate is set to 2000 samples per beam while a 1000 sample window is sliced out for SBS processing, providing a 500 sample guard time on each side. Depending on the hardware, desired algorithm accuracy, sampling rate, SNR, etc; these values can shrink for increased speed and reduced acquisition time.

Finally, other firmware functionality, such as adaptive beamforming or tracking, has been experimented with and could easily be added to the platform for future improvements.

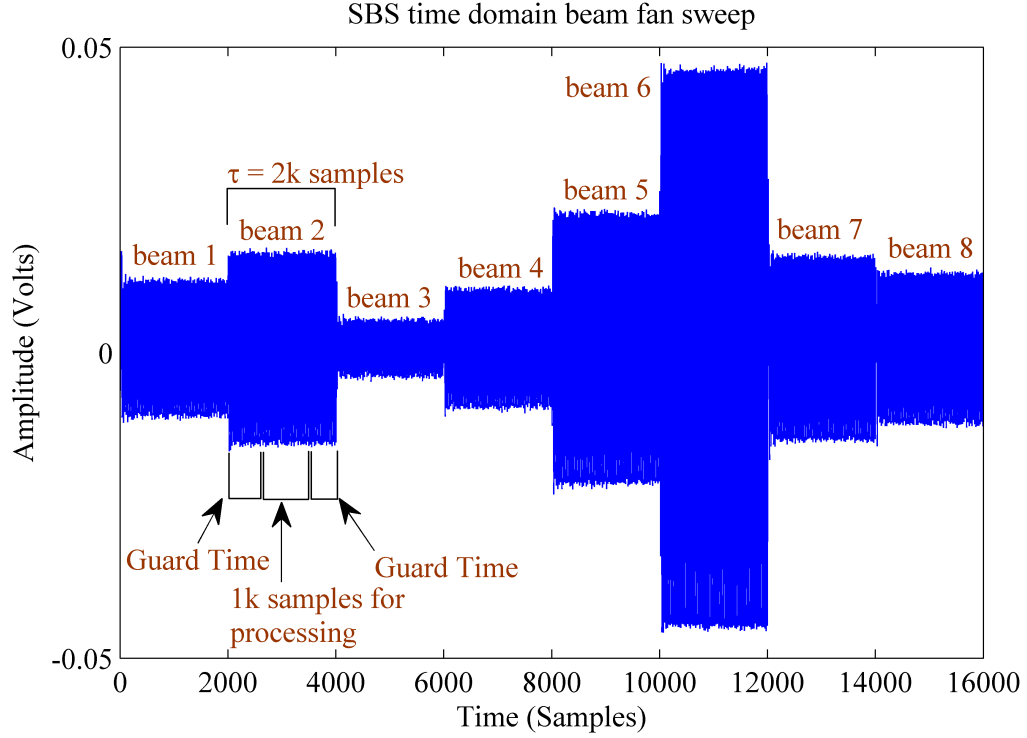


Figure 2.8: Commutation rate timing diagram example

2.2.2 MATLAB DSP

All DSP, real-time and post-processing, is accomplished in the MATLAB environment. This section briefly describes the essential, numerical MATLAB processing needed to synthesize SBS data for simulations and carry out SBS processing in the lab. The significant portions of code are attached as Appendix B.

Most importantly, for SBS simulation results to be realistic a function to create SBS data, i.e. beamspace data with a time delay between beam vectors in an arbitrary or fixed fashion, was created. The function *SamplingError.m* takes synchronized beamspace data and inserts a random or fixed commutation rate between beam vectors and returns this as a counter value, which is used to realign the data in later operations.

Four main processing steps are needed to create and process SBS data in the lab. First, the weights stored on the FPGA toggle 6-bit devices (attenuators and phase shifters), meaning they can accept 2^6 , or 64, different binary values. The weights are first created using a shift and sum method or window function in MATLAB, which are output as complex floating point decimal numbers, but need to be converted to a 6-bit number comprehensible to the FPGA. The program *weight2decimal.m* converts the complex decimal number to integer decimals, binary, and hexadecimal, all based on a 6-bit resolution. The 0-63 integer decimal numbers are an intermediate step, while the binary is key in troubleshooting software and hardware interaction, and the hexadecimal value is what is stored on the FPGA as weight coefficients. Second, a hi-res and highly accurate Capon spectrum estimator is implemented in MATLAB, further discussed in section 3.4. Each detected signal source is paired with its detected frequency of operation, and these frequencies are piped to subsequent functions to aid in synchronization. Next, a third key function is synchronizing the data with an absolute time delay correction filter through the function *wilbert.m*. Essentially, this uses the preset commutation rate, hard-coded on the FPGA, and the detected frequency to apply a corresponding phase shift to the beamspace data in frequency space. The output is an in-phase and quadrature beamspace signal that has been aligned in time and phase according to the frequency of operation. Last, multiple beamspace DOA algorithms that efficiently search and estimate the DOA of an impinging RF source are executed in MATLAB.

2.2.3 PC Control Interfaces

The final software section marries beamspace DSP algorithms with real measurement data by constructing a user friendly interface between the user, data acquisition

hardware, and software. This is accomplished with LabVIEW and the use of VIs and sub-VIs and MATLAB scripts, that all render as graphical user interfaces (GUIs).

The first GUI is key for calibration, further explained in section 3.4, by automating a process to measure the empirical weights on the beamformer. Because the phase shifters and attenuators are digitally controlled devices with a finite 6-bit resolution, error is incurred when the absolute weight is converted to an integer decimal number approximation (see *weight2decimal.m* in Appendix B). Other sources of error include component and analog path implementation effects distinct to each channel on the beamformer, such as minute, yet nontrivial differences in power distribution, component tolerances, component manufacturing, and printed circuit board (PCB) layout. By measuring the empirical weights the implementation effects can be estimated and substituted with ideal weights using a DSP calibration method. A snapshot of the user interface is shown in figure 2.9. The GUI synchronizes the beamformer, a relay test setup, and a Vector Network Analyzer (VNA) to switch between every channel and measure the phase and magnitude. The output is a *.mat* file, native to MATLAB, containing weights for all channels and every beam it is programmed to measure.

A second GUI, created with MATLAB, relates directly to SBS DOA estimation. This GUI controls the beamformer, the NI ADC (PCI-6133), and the NI DAQ (USB-6501) to collect and analyze a matrix of beamspace observation data. A soft trigger initiates the SBS mode in firmware, and then an output flag from the FPGA precisely triggers the NI ADC to begin collecting data. The length of data collection is controlled by a samples per trigger attribute. The scripts embedded in the MATLAB GUI use essential functions from the previous section, 2.2.2, to arrive at valid and consistent DOA estimations. Furthermore, the following data is output to the interface: DOA estimations of signals, the number of signals detected and their frequencies, and spatial pseudo-power spectrum plots. Sequential time domain data is also viewable.

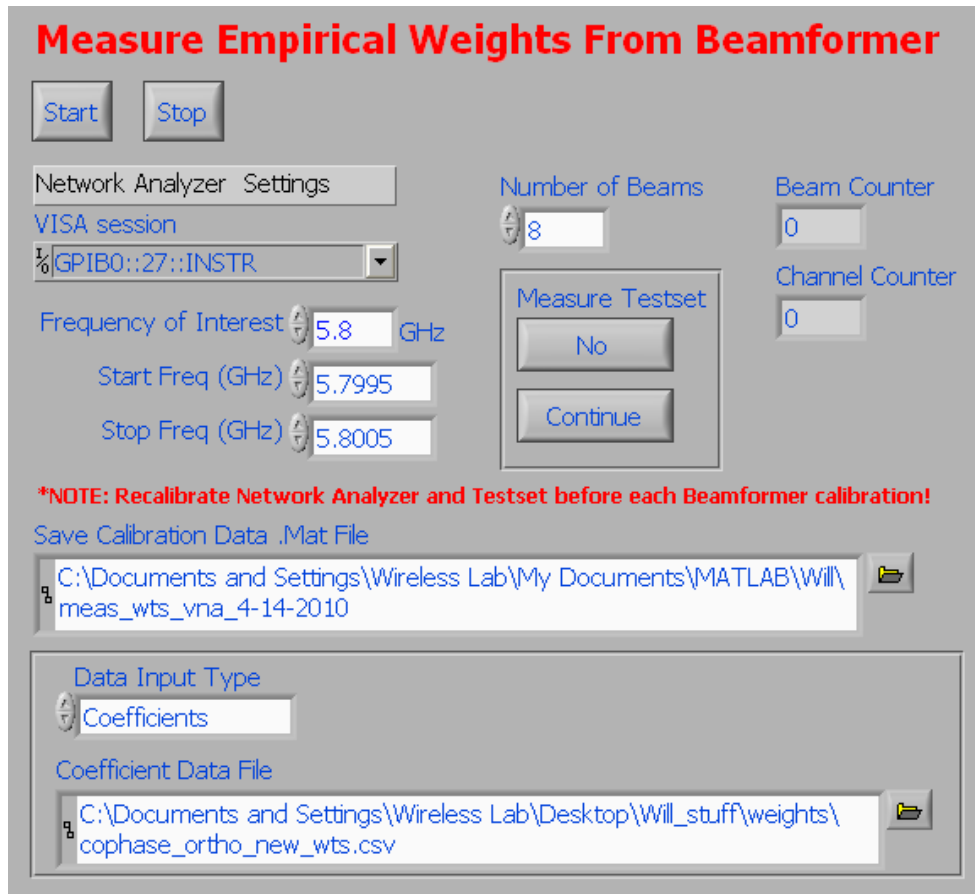


Figure 2.9: Empirical weight measurement GUI

A snapshot of the user interface is shown in figure 2.10. The upper left portion of the GUI accepts inputs for beam commutation rate, mode, sector search type, sample size, and initiating the sweep. The top plot displays time domain beamspace data on a per beam basis. The bottom plots from left to right display the DOA spatial pseudo-power spectrum and the baseband frequency spectrum, respectively. The DOA angles and frequencies of operation are displayed in the lower left portion. A useful feature allows the user to capture a single set of data, from a rough sector or fine sector sweep, and repeatedly analyze the data with different options, such as all sector or RMS voltage fine sector estimation.

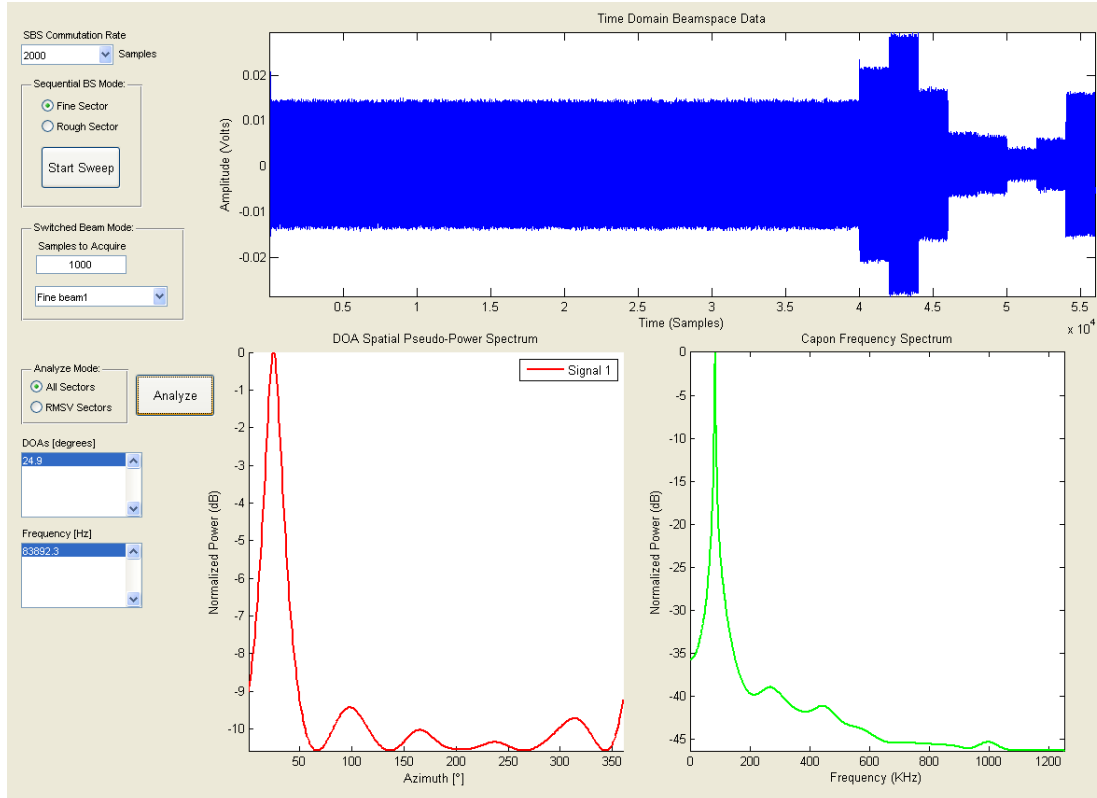


Figure 2.10: Sequential beamspace GUI

The system presented fully defines all the components needed to carry out the new SBS technique in hardware: the UCA antenna, beamformer, single channel receiver, beamformer firmware, MATLAB DSP m-files, and LabVIEW and MATLAB GUIs for user input. Countless hours of design and fine tuning ensure the successful collaboration of software and hardware, providing the foundation for this project.

CHAPTER 3

SEQUENTIAL BEAMSPACE: MODEL, SIMULATIONS, AND METHODS

Sequential beamspace has undergone rigorous simulation and comparison to other similar element space and beamspace approaches used for DOA estimation and beamforming. It is the goal of this chapter to systematically present a data model for the SBS technique and discuss results of said simulations and comparisons, which are the direct consequence of this work. The obstacles and solutions of SBS as a DOA estimation and beamforming device will be presented and dissected, including calibration to remove weighting errors and discrete phase shifting errors. Also, the feasibility of SBS will be validated by simulations and solutions herein, proving that the proposed technique can be adopted by many conventional phased array systems currently used with commercial radios, such as Wi-Fi, Wi-MAX, and LTE, as well as military radios, for hi-res RF source localization and beamforming without the addition of expensive hardware [3].

3.1 Data Model

An m element array with a circular geometry is considered. The array receives narrow band signals from M spatially separated signal sources, where $M < m$. The multipath signals are ignored. Assume an array response vector to a transmitted signal source from an azimuth direction of arrival, ϕ , and an elevation direction of arrival, θ . Calculation of the DOA of a signal is required to determine the location of the source and track its movement [3]. The discrete element space array signal is the observation data matrix, defined by the following:

$$\mathbf{x}(k) = \sum_{i=1}^M \mathbf{a}(\phi_i, \theta_i) s_i(k) + \mathbf{n}(k) \quad (3.1)$$

where $\mathbf{x}(k) \in \mathbb{C}^{m \times 1}$ at any k^{th} sample in time, and $\mathbb{C}$ represents a complex valued space. The element space spatial signature is $\mathbf{a}(\phi_i, \theta_i) \in \mathbb{C}^{m \times 1}$, of each signal, $s_i(k)$, and $\mathbf{n}(k) \in \mathbb{C}^{m \times 1}$ is an additive white Gaussian noise vector. The element space spatial signature is the UCA spatial signature at relevant arrival angles, θ and ϕ , and is defined as the following:

$$\mathbf{a}(\phi, \theta) = \begin{bmatrix} 1 \\ e^{j\beta\rho\sin(\theta)\cos(\phi - \frac{2\pi \times 1}{m})} \\ e^{j\beta\rho\sin(\theta)\cos(\phi - \frac{2\pi \times 2}{m})} \\ \vdots \\ e^{j\beta\rho\sin(\theta)\cos(\phi - \frac{2\pi \times (m-2)}{m})} \\ e^{j\beta\rho\sin(\theta)\cos(\phi - \frac{2\pi \times (m-1)}{m})} \end{bmatrix} \quad (3.2)$$

where $\beta = \frac{2\pi}{\lambda}$ is the wave number, λ is the wavelength, and ρ is the radius of the array in meters. For simplification purposes the elevation angle, from herein, is assumed to be constant at $\theta = 90^\circ$, so $\sin(\theta) = 1$. Furthermore, $\mathbf{A} \in \mathbb{C}^{m \times M}$, is the element space spatial signature matrix given by:

$$\mathbf{A} = [\mathbf{a}(\phi_1) : \mathbf{a}(\phi_2) : \dots : \mathbf{a}(\phi_M)] \quad (3.3)$$

and the signal vector, $\mathbf{s}(k) \in \mathbb{C}^{1 \times M}$, is the following:

$$\mathbf{s}(k) = [s_1(k), s_2(k), \dots, s_M(k)]^T \quad (3.4)$$

So, in matrix format the element space array signal becomes:

$$\mathbf{X}(k) = \mathbf{A}\mathbf{S}(k) + \mathbf{N}(k), \quad k = 1, 2, \dots, K \quad (3.5)$$

where K is the total number of observation samples [17].

A transformation from element space to beamspace must now be performed on the data model. The m element space channels are summed and the linear combination of the antenna data, scaled and shifted by the weight values, is simultaneously the beamformer output and the beamspace data. Therefore, the beamspace array signal as well as the beamformer output, is the following:

$$y(k) = \sum_{i=1}^m w_i^* x_i(k) = \mathbf{w}^H \mathbf{x}(k) \quad (3.6)$$

where $\mathbf{w} \in \mathbb{C}^{m \times 1}$, is a column vector of weights for a single beam [2]. The beamspace output for the entire beam fan is the following:

$$\mathbf{y}(k) = \mathbf{W}^H \mathbf{x}(k) \quad (3.7)$$

The beamformer matrix, $\mathbf{W} \in \mathbb{C}^{m \times N}$, can be generated in many ways, from window functions to adaptive algorithms; however, when using the classic shift and sum excitation method the weights are tantamount to the UCA spatial signature, constrained by specific ϕ 's. The weights are defined by:

$$\mathbf{W} = [\mathbf{a}(\phi_1) : \mathbf{a}(\phi_2) : \dots : \mathbf{a}(\phi_N)] \quad (3.8)$$

More specifically, ϕ is now each bore sight angle of N shift and sum beams. Usually, $N < m$ for dimensionality reduction, but this does not have to hold true in SBS where $N = m$ yields no loss of information. To reduce out-of-sector interference, so a signal impinging on the array at bore sight of one beam does not corrupt the side

lobe of any other beam, the following matrix is computed (guarantees orthogonality of the weighting matrix columns):

$$\mathbf{W}_{\perp} = \mathbf{W}[\mathbf{W}^H \mathbf{W}]^{(-1/2)} \quad (3.9)$$

so $\mathbf{W}_{\perp}^H \mathbf{W}_{\perp} = \mathbf{I}$ is true, where $\mathbf{I}$ is the identity matrix [17, 18]. These orthogonal weights are then used for beam formation; they are the final beamspace weights that are sent and applied to the beamformer board to generate beamspace data. Using the same transformation logic the beamspace spatial signature, or beamspace manifold, is computed from the element space spatial signature:

$$\mathbf{b}(\phi) = \mathbf{W}^H \mathbf{a}(\phi) \quad (3.10)$$

Initially, the sampled beamspace data is real, $\mathbf{y}(k) \in \mathbb{R}^{N \times 1}$, so the DC bias, or mean, is subtracted from the signal, and the Hilbert transform is used to create the analytic beamspace signal, $\mathbf{y}(k) \in \mathbb{C}^{N \times 1}$. From [19] the Hilbert transform $\mathcal{H}$, of a function $f(t)$, is given by:

$$\mathcal{H}\{f(t)\} = \frac{1}{\pi t} * f(t) \quad (3.11)$$

where $(*)$ denotes the convolution operation. Using the Fourier transform specified by $\mathcal{F}$, the Hilbert transform in frequency space is:

$$\mathcal{F}\{\mathcal{H}\{f(t)\}\} = F(\Omega)(-j \operatorname{sgn}(\Omega)) \quad (3.12)$$

where $-j \operatorname{sgn}(\Omega)$ is the continuous frequency Hilbert transformer, or sign shift function, and sgn is the signum function. Critical to SBS operation, the discrete Hilbert transformer is now defined by:

$$Ht(\omega) = -j \operatorname{sgn}\left(\frac{\omega_s}{2} - \omega\right) \operatorname{sgn}(\omega) \quad (3.13)$$

where ω_s is the sampling rate and ω is discrete frequency. Enacting this transformer on the beamspace data in the frequency domain, then transmuting back to the time domain yields in-phase and quadrature beamspace data components via minimal soft processing [19, 20]. Also, the phase correction cardinal to SBS is injected inside the Hilbert transform via the time delay correction filter. Next, to mathematically extract useful information for DOA estimations, the autocorrelation function is used to compute the spatial beamspace covariance matrix of the beamspace array signal, defined by the following:

$$\mathbf{R}_{yy} = \mathbb{E}[\mathbf{y}(k)\mathbf{y}^H(k)] \quad (3.14)$$

where $\mathbb{E}[\cdot]$ is the expectation operator. However, in the real world this is the result of a finite sampled data set and is therefore an estimate. So, computed as a time average, the sampled, spatial beamspace covariance matrix is now:

$$\hat{\mathbf{R}}_{yy} = \frac{1}{K} \sum_{k=1}^K \mathbf{y}(k)\mathbf{y}^H(k) \quad (3.15)$$

In matrix format the beamspace covariance matrix can be expanded and simplified:

$$\begin{aligned} \mathbf{R}_{yy} &= \mathbf{B}\mathbb{E}[\mathbf{s}(k)\mathbf{s}^H(k)]\mathbf{B}^H + \mathbf{W}^H\mathbb{E}[\mathbf{n}(k)\mathbf{n}^H(k)]\mathbf{W} \\ &= \mathbf{B}\mathbf{P}\mathbf{B}^H + \sigma^2\mathbf{I} \end{aligned} \quad (3.16)$$

where $\mathbf{P}$ is the source covariance matrix, and $\sigma^2\mathbf{I}$ is the noise covariance matrix. Because multipath is ignored, the source covariance matrix:

$$\mathbf{P} = \mathbb{E}[\mathbf{s}(k)\mathbf{s}^H(k)] \quad (3.17)$$

is assumed to be a constant over the beamspace acquisition time. In some radio applications where line of sight paths are not possible, multipath is a dominating effect that can cause radical swings in the source power so this effect is currently ignored. Finally, to finish the beamspace covariance matrix definition:

$$\mathbf{B} = \mathbf{W}^H \mathbf{A} \quad (3.18)$$

is the total beamspace manifold matrix for M far field sources [3, 4, 15].

3.2 High Resolution DOA Estimation

The most important objective of SBS is to provide accurate DOA estimations for desired sources (eventually this will be partnered with detecting and nulling interferers). The SBS technique is novel in the way it collects data with the same hardware it uses to beamform, yet convenient because established beamspace DOA algorithms can be used to process the data.

3.2.1 Element space vs. Beamspace

Before beamspace is vigorously pursued as a viable option for DOA estimation it must be compared with its element space counterpart to provide evidence for its feasibility. Extensive MATLAB simulations were carried out to compare the two approaches in the areas of accuracy, resolution, and probability of resolution. According to [5], DOA estimation variances can never be as small in beamspace as in element space because of the loss in source location information, via reduced di-

mension beamspace (RDBS), when transforming from element space to beamspace. Although in some cases, like when employing MUSIC, resolution SNR thresholds for DOA estimation can be lower in beamspace than in element space, but this greatly depends on the matrix beamformer used [6]. However, operating in SBS mode with $N = m$, and employing the weights from eq. 3.9 as the matrix beamformer, DOA estimation in beamspace is shown to be highly similar to and sometimes outperforms element space.

In the following simulations the UCA-Real Beamspace-MUSIC (UCA-RB-MUSIC, and for plotting this is shortened to RBS-MUSIC) algorithm from [13,21] was used on beamspace data and standard MUSIC from [7] was used on element space data. Also, orthogonal shift and sum weights from eq. 3.9 were used as the matrix beamformer. Beamspace MUSIC and element space MUSIC DOA estimation accuracy, resolution, and probability of resolution are compared. Figure 3.1(a) shows a Monte Carlo simulation study for accuracy comparison of beamspace MUSIC, element space MUSIC, and the Cramér Rao Bound (CRB). The deterministic CRB model used in this work is the best theoretical accuracy achievable at all SNR levels. The plot reflects the accuracy expressed in terms of RMSE, in degrees, of each algorithm versus increasing SNR levels. One thousand samples were taken per trial with 1000 independent trials from -10dB to 40dB SNR in steps of 1 dB. Figure 3.1(b) shows Monte Carlo simulation study results for resolution with two sources. One source is fixed at 180° , while the other source sweeps across from 120° to 180° at each SNR level, in steps of 2 dB from -10 to 40 dB. The metric for minimum source angular resolution is defined as when the DOA algorithm cannot recognize two independent signals only once out of 1000 independent trials. This means if there is a 0.1% possibility of not recognizing two separate signals, the source separation angle is irresolvable at the given SNR.

Every batch of 1000 trials was carried out at each angle, in steps of 0.25° , and SNR level [22].

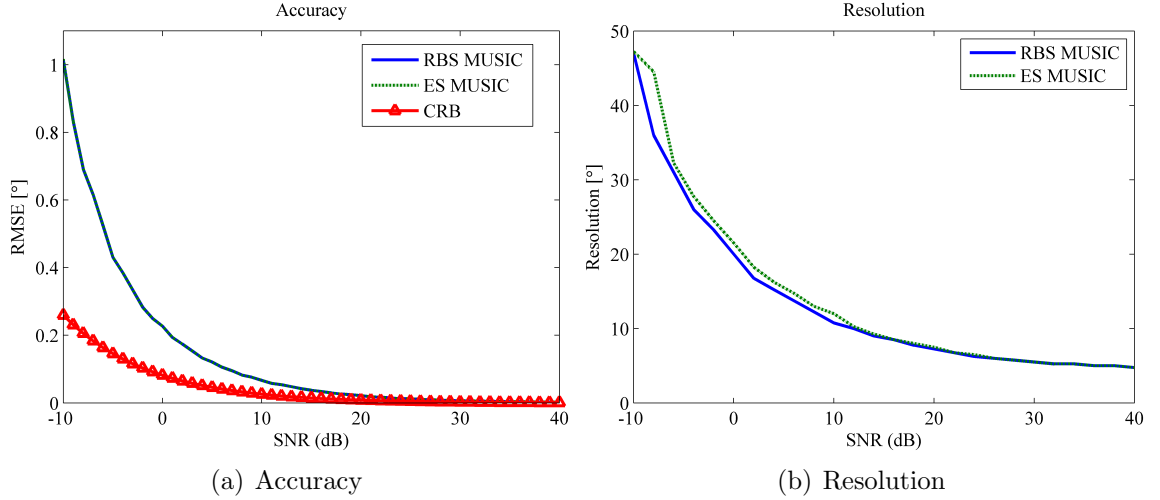


Figure 3.1: Element space vs. beamspace DOA accuracy and resolution simulations

Next, a Monte Carlo simulation study for probability of resolution was undertaken. Figure 3.2 shows probability of resolution for beamspace MUSIC and element space MUSIC DOA estimation versus SNR, where two sources are separated by 15° , i.e. one at 165° and another at 180° , 10° , and 5° . This plot is normalized to 1, with 0 being completely irresolvable and 1 being 100% resolvable. This simulation utilized 1000 independent trials at each SNR level in steps of 1 dB [22].

Figure 3.1(a) shows that the RMSE accuracy for both beamspace and element space MUSIC to be highly similar over all SNR levels. Upon closer inspection, element space MUSIC narrowly out performs beamspace MUSIC in terms of accuracy, but the variance is negligible. This result matched the remark that, in general element space MUSIC performs equal to or better than beamspace MUSIC [5]. In terms of resolution, from Figure 3.1(b), it is shown that beamspace MUSIC is better than element space MUSIC, and both methods converge to nearly the same resolution when the SNR is good ($> 13\text{dB}$). Figure 3.2 shows that element space MUSIC resolves 100%

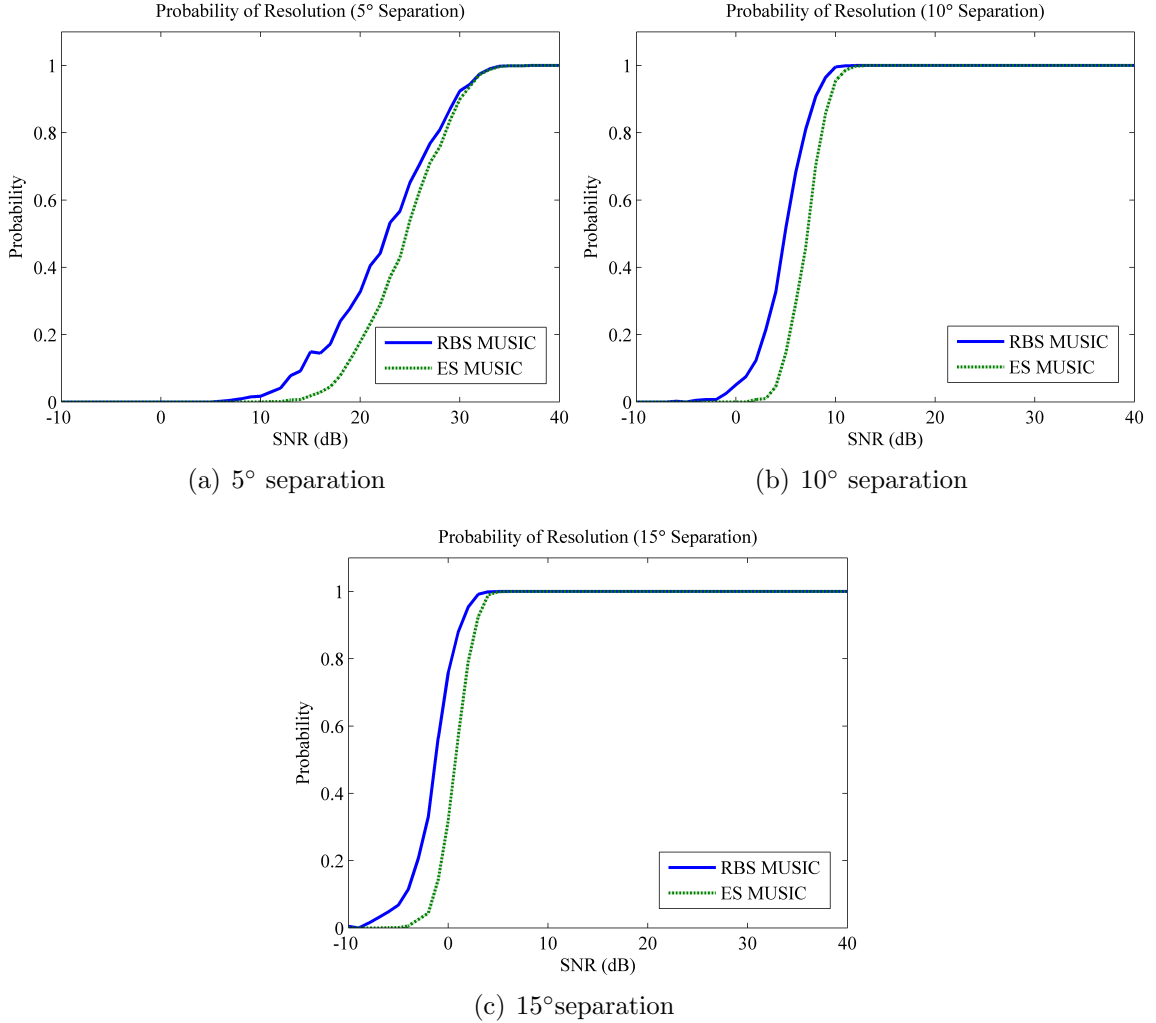


Figure 3.2: Element space vs. beamspace DOA probability of resolution simulations of the time at slightly higher SNR levels then when beamspace MUSIC resolves 100% of the time, for all three angle separations. Moreover, element space MUSIC has greater computational complexity because of a requisite complex-valued eigenvalue decomposition (EVD), whereas UCA-RB-MUSIC involves a simpler real-valued EVD, making it more attractive in both senses [13]. Because of the similar performance of beamspace compared to element space in these simulations, beamspace proves to be highly useful in DOA estimation and further research into SBS is warranted [22].

3.2.2 Switched Beam vs. Beamspace

Switched beam (SWB), or switched lobe, is a form of data independent beam-forming – typically executed with shift and sum weights, a fixed pattern, and a fixed number of beams – that can provide course DOA estimation with the help of RF power monitoring. A beam is formed in each fixed direction while a power measurement is recorded and paired with each beam, representing a fraction of the total coverage. As mentioned in chapter 2, the beamformer board has a modular power detection unit that can deliver this rudimentary yet useful functionality via bandpass filtering and diode envelope detection. The advantage of this technique lies in the simplicity and speed; forming a beam according to preset weights and then saving and comparing received power values has little computational baggage. SWB measurement speed relies on the beam switching time and the detector response time. The present system can switch beams as fast as a bit change on the FPGA controller, 100 ns, plus a short settling time associated with the phase shifters and attenuators, ≈ 180 ns, plus the detector pulse response time, 15 ns. Assuming the same sample length and ignoring the power detector response time, this is the same time it would take to switch beams in SBS. However, the speed of the SWB system is indirectly proportional to the accuracy. For example, forming 8 beams only yields an RMSE accuracy of $\frac{360^\circ}{2N} \times \frac{1}{\sqrt{3}} = \frac{22.5^\circ}{\sqrt{3}} \approx 13^\circ$, while an RMSE accuracy of 1° would require 104 beams, drastically decreasing speed and increasing acquisition time. However, sub-degree accuracy is theoretically achievable with $N = m \Rightarrow 8$ beams, using the current hardware and SBS technique. Tracking in SWB can be implemented more elegantly, such as switching between 3 beams, the present beam and each adjacent beam. This exemplifies another downfall with SWB compared to SBS, the ability to discover and track only a single friendly source [16].

3.2.3 Beamspace Algorithms

Three spectral-based DOA algorithms have been studied in simulation and implemented in SBS. Two popular algorithms, MUSIC from [7] and Capon from [8], as well as a derivative of MUSIC, UCA-RB-MUSIC from [13, 21], all operate in beamspace. For all three algorithms, a search technique is used to find the location of the maximum peak of the spatial pseudo-power spectrum $S(\phi)$, which is the DOA estimation:

$$\hat{\phi} = \arg \max_{\phi} (S(\phi)) \quad (3.19)$$

Also, the RMSE of the estimated detected arrival angle is defined as the following:

$$\text{RMSE} = \sqrt{\sum_{d=1}^D \frac{(\hat{\phi}_d - \phi_d)^2}{D - 1}} \quad (3.20)$$

where $\hat{\phi}_d$ is the estimated DOA at each trial, ϕ_d is the true DOA of the signal source at each trial, and D is the total number of the trials [3, 4].

3.2.3.1 Beamspace Capon: Capon DOA estimation employs the Capon beamformer technique, discussed in greater detail in section 3.3, also known as the Minimum Variance Distortionless Response (MVDR) filter. The Capon DOA method resolves the power of sources spaced closer than a fixed beamwidth, unlike switched beam DOA above. The spatial spectrum is found by computing a set of optimal weights with Lagrange multipliers and beamforming in all directions. This results in a sharp spatial bandpass filter with high resolution, from high out-of-band suppression. Using eqs. 3.10 and 3.15, the normalized Capon spatial pseudo-power spectrum is the following:

$$S_{CAP}(\phi) = \frac{1}{\mathbf{b}^H(\phi) \hat{\mathbf{R}}_{yy}^{-1} \mathbf{b}(\phi)} \quad (3.21)$$

[3, 15, 18]

3.2.3.2 Beamspace MUSIC: MUSIC is categorized as a subspace-based algorithmic DOA estimation solution. Assuming spatially white noise, further factorization of eq. 3.16 is needed to arrive at the MUSIC spatial pseudo-power spectrum. The beamspace covariance matrix can be written in terms of its eigenvalue decomposition:

$$\mathbf{R}_{yy} = \mathbf{U}_{bs,s} \mathbf{\Lambda}_s \mathbf{U}_{bs,s}^H + \mathbf{U}_{bs,n} \mathbf{\Lambda}_n \mathbf{U}_{bs,n}^H \quad (3.22)$$

where $\mathbf{\Lambda}_n = \sigma^2 \mathbf{I}$ is the eigenvalues matrix of the beamspace array noise, $\mathbf{U}_{bs,s}$ is the beamspace signal eigenvectors, and $\mathbf{U}_{bs,n}$ is the beamspace noise eigenvectors. All noise eigenvectors are orthogonal to $\mathbf{B}$, so the columns of $\mathbf{U}_{bs,s}$ span the space of $\mathbf{B}$, while the columns of $\mathbf{U}_{bs,n}$ span the nullspace of $\mathbf{B}^H$. The following projection operators exhibit the orthogonal nature between the signal subspace and the noise subspace, i.e. the signal and noise are linearly independent:

$$\mathbf{\Pi}_{bs} = \mathbf{U}_{bs,s} \mathbf{U}_{bs,s}^H \quad (3.23)$$

$$\mathbf{\Pi}_{bs}^\perp = \mathbf{U}_{bs,n} \mathbf{U}_{bs,n}^H \quad (3.24)$$

where $\mathbf{\Pi}_{bs} + \mathbf{\Pi}_{bs}^\perp = \mathbf{I}$. Because all the eigenvectors not belonging to the impinging signals, i.e. noise eigenvectors, are orthogonal to $\mathbf{B}$, the estimated noise space multiplied by the ideal beamspace manifold yields the following:

$$\hat{\mathbf{U}}_{bs,n}^H \mathbf{b}(\phi) = 0, \quad \phi \in \{\phi_1, \dots, \phi_M\} \quad (3.25)$$

Therefore, applying eq. 3.24 to an estimated noise dataset, the normalized beamspace MUSIC spatial pseudo-power spectrum becomes:

$$S_{MUS}(\phi) = \frac{1}{\mathbf{b}^H(\phi) \hat{\mathbf{\Pi}}_{bs}^\perp \mathbf{b}(\phi)} \quad (3.26)$$

[3, 15, 17]

3.2.3.3 UCA Real Beamspace MUSIC: UCA-RB-MUSIC is a novel extension of beamspace MUSIC that typically employs the use of a real-valued beamformer with reduced dimension; however, in the context of this work it is used with a $N = m$ beamformer, from eq. 3.9, with shift and sum weights as the original components. Three advantages accompany UCA-RB-MUSIC, including the use of real-valued matrix computations, meaning real-valued EVDs, an inherent forward/backward average that improves estimation in correlated source scenarios, and the ability to enlist the help of Root-MUSIC from [23]. In this work, the first advantage is the focus of the UCA-RB-MUSIC implementation. The use of a real-valued covariance matrix and real-valued beamspace manifold exemplify the distinction between MUSIC and UCA-RB-MUSIC. The new real-valued covariance matrix is the following:

$$\mathbf{R} = \Re\{\mathbf{R}_{yy}\} = \mathbf{B}_r \mathbf{P}_r \mathbf{B}_r^T + \sigma^2 \mathbf{I} \quad (3.27)$$

where $\mathbf{P}_r = \Re\{\mathbf{P}\}$, subscript $(\cdot)_r$ denotes a real-valued space, and $\Re\{\cdot\}$ is the real number operation that takes the real part of a complex value. A real-valued EVD of $\mathbf{R}$ will yield beamspace subspaces, signal and noise, composed of orthonormal

eigenvectors. Thus, the real-valued EVD of $\hat{\mathbf{R}}$ returns the real-valued estimated noise space, $\hat{\mathbf{G}}$, and the real-valued estimated signal subspace. Finally, the normalized UCA-RB-MUSIC spatial pseudo-power spectrum is:

$$S_{RBM}(\phi) = \frac{1}{\mathbf{b}_r^T(\phi)\hat{\mathbf{G}}\hat{\mathbf{G}}^T\mathbf{b}_r(\phi)} \quad (3.28)$$

[13, 21].

3.2.3.4 Simulation Study: Next, UCA-RB-MUSIC, beamspace MUSIC, and beamspace Capon DOA estimation accuracy, resolution, and probability of resolution are compared. Figure 3.3 shows a Monte Carlo accuracy comparison between all three algorithms and the CRB. Figure 3.4 shows Monte Carlo simulation results for resolution with two sources. The simulation was conducted exactly as the beamspace vs. element space study in section 3.2.1, using the same parameters: SNR sweep, number of trials, and metric for source separation angle resolution. From figure 3.3(a), the accuracies completely coincide with minute negligible differences. Upon zoom-in, figure 3.3(b) exhibits the minuscule differences in RMSE accuracy between beamspace MUSIC and beamspace Capon, at 5 dB SNR. Also, examining figure 3.4(a), UCA-RB-MUSIC has the best resolution results while beamspace Capon has expected results; it is unable to resolve as closely spaced sources as beamspace MUSIC. A probability of resolution example at 10° angular separation is given in figure 3.4(b), further supporting the results from figure 3.4(a). These results also correlate nicely with the element space results; thus, underpinning the claim of successful theoretical implementation of all three algorithms for beamspace DOA estimation [22].

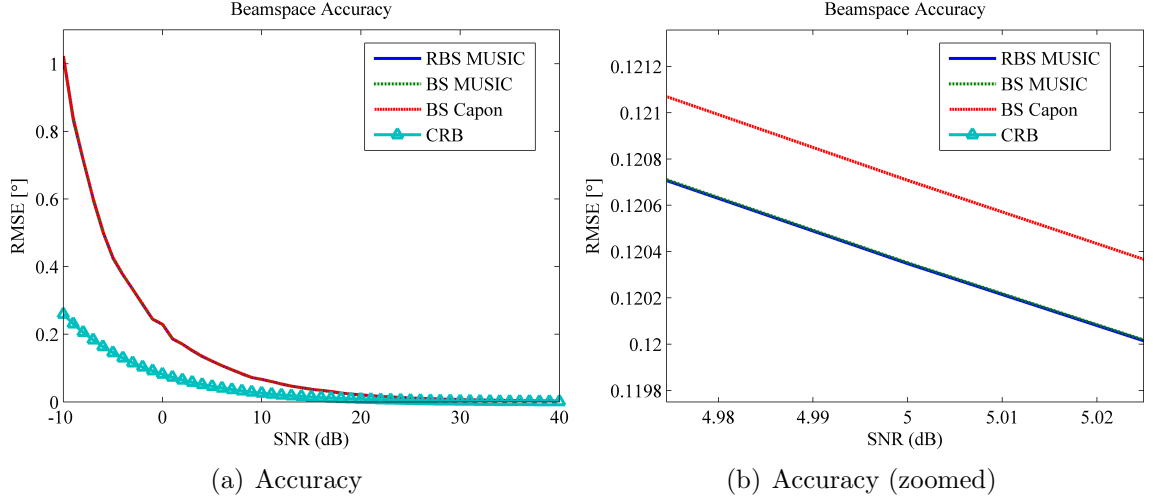


Figure 3.3: Beamspace DOA accuracy simulation

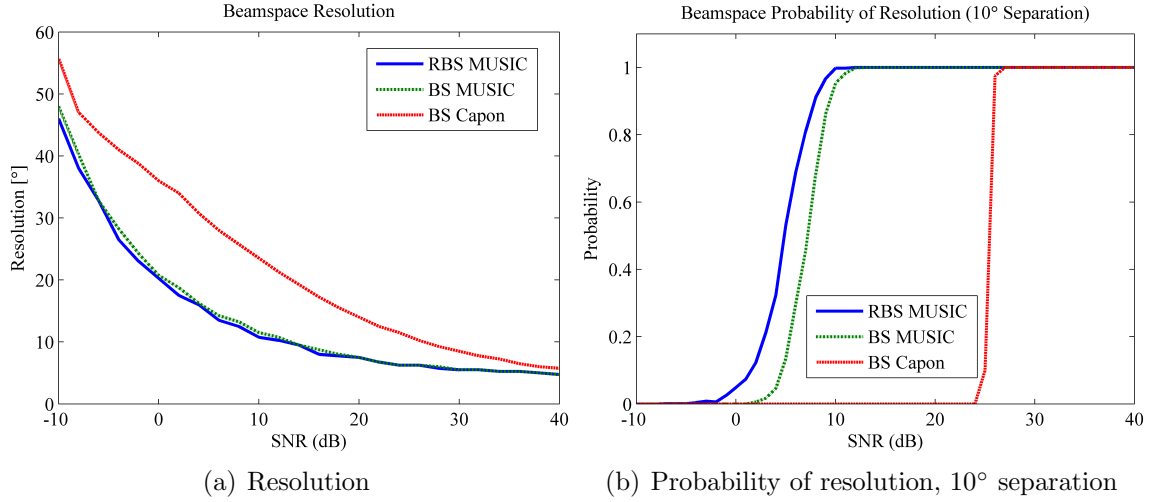


Figure 3.4: Beamspace DOA resolution simulations

3.2.4 Obstacles and Solutions

SBS presents some challenges, but once these obstacles are overcome with the presented solutions all advantages of SBS can be realized. From chapter 1, SBS suffers from a synchronization issue where the phase front of each vector of beamspace data is no longer aligned. This phenomenon is a consequence of sampling phase error, which must be corrected for accurate DOA estimations. Also, some interesting tech-

niques can be implemented to minimize the total acquisition time without sacrificing accuracy with a dual search technique using a rough sector search and then a fine sector search.

3.2.4.1 Sampling Phase Error: The beamspace simulations were accomplished by transforming synchronized element space data directly into beamspace data by applying a numerical beamformer matrix. In element space, the freedom exists to sample all antenna outputs concurrently; whereas, in SBS the beamformer can only form one beam in one direction at a time. Because the data is sampled in a sequential manner, an arbitrary time delay exists between each vector of beamspace data. In the present system the time delay is settable because of intelligent resources on the beamformer board, i.e. counter on the FPGA controller. Basically, transforming element space data directly to beamspace, to use as SBS data, gives the illusion that all beams are formed simultaneously and therefore provides erroneous data. To generate a more realistic data set, a time shift must be incorporated between beam vectors in the beamspace observation matrix (see *SamplingError.m* in Appendix B). An element space data matrix, consisting of 1000 samples and directly transformed into ideal conventional beamspace, can be modeled as having the following sample indexes:

$$\mathbf{Y}_{CBS}(k) = \begin{bmatrix} 1 : 1000 \\ 1 : 1000 \\ \vdots \\ 1 : 1000 \end{bmatrix} \quad (3.29)$$

This provides complete synchronization, and no phase error is introduced from sequential sampling, i.e. all row vectors start and end at the same point. A more accurate and complete model for SBS data is the following:

$$\mathbf{Y}_{SBS}(k) = \begin{bmatrix} k_1 : k_1 + 999 \\ k_2 : k_2 + 999 \\ \vdots \\ k_N : k_N + 999 \end{bmatrix} \quad (3.30)$$

where:

$$k_1 + 999 < k_2, \dots, k_{N-1} + 999 < k_N \quad (3.31)$$

This incorporates a time delay between each beamspace row vector, of at least the sampling length, to emulate sequential beamforming and sampling. To clarify, k_i represents a sample time index specifying the start of a particular beam number, from 1 to N . In this representation all beams are chronological in order and none of the sampling time intervals overlap one another. The time difference between k_2 and k_1 is the duration to collect a given number of samples and a short, yet nontrivial in terms of phase error, switching time. Again, this time duration is the beam commutation rate, τ . Furthermore, to acquire a complete beamspace matrix composed of all beamspace vectors the implicit assumption is that the radio channel is stationary over the entire acquisition interval, i.e. the total sampling and switching time for all beams. This is defined as the beam commutation rate multiplied by the total number of beams, $\tau \times N$ [3].

A further mathematical perspective on sampling phase error is how it ultimately affects the covariance matrix of eq. 3.15. An ideal beamspace covariance matrix,

where all beams have been collected concurrently, would have the following value for off-diagonal elements, ij :

$$r_{ij} = \frac{1}{K} \sum_{k=1}^K y_i(k) y_j^*(k) \quad (3.32)$$

where i is the row index and j is the column index. These indexes can also be considered the beam indexes, where i or $j = 1, 2, \dots, N$. The diagonal entries, average beam powers or variances, occur when $i = j$. With the introduction of sampling phase error in the SBS approach, the SBS covariance matrix now resembles the following:

$$r_{s,ij} = \frac{1}{K} \sum_{k=1}^K y_i(k - (i-1)\tau) y_j^*(k - (j-1)\tau) \quad (3.33)$$

Using a fixed commutation rate, every subsequent beam is collected an integer multiple of τ behind the initial beam; therefore, the delay is the previous beam index multiplied by τ . Contrarily, with an arbitrary commutation rate the $(i-1)\tau$ and $(j-1)\tau$ terms become a random delay, T_i and T_j respectively. The difficult issue is determining when and how to differentiate between phase injected by the signal spatial signature or unwanted phase inherent to sequential sampling.

3.2.4.2 Time Delay Correction Filter: Next, a simulation was carried out with this realistic model to verify or refute the beamspace DOA estimation accuracy in the presence of sampling phase error. Also, the solution that compensates for the sampling error is presented, and justified by the simulation results.

Clearly, from the unsynchronized blue trace in figure 3.5, it can be seen that the phase error introduced by an arbitrary sequential commutation rate is detrimental to the DOA estimation accuracy and resolution. Errors on the order of tens of degrees are

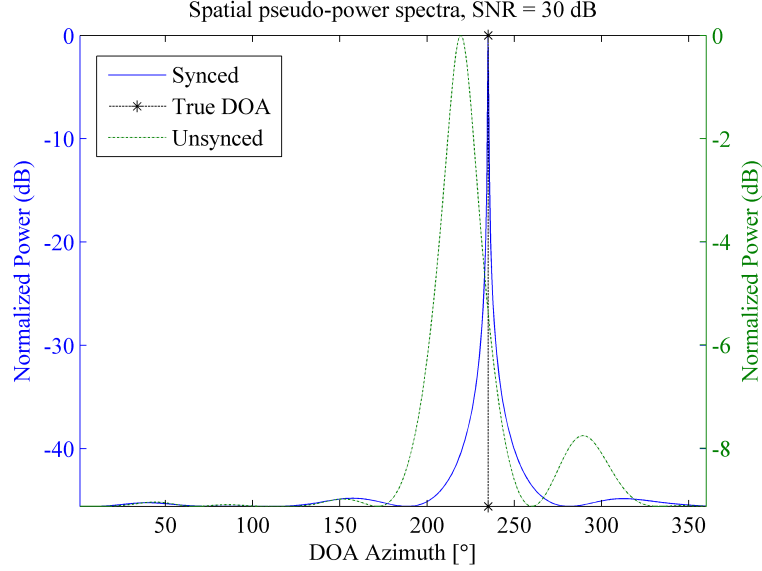


Figure 3.5: Unsynchronized and synchronized DOA estimation comparison

fairly consistent, 15° in figure 3.5, and the noise floor increases drastically, as much as 37 dB, which relates to the potential angular resolution between sources. Specifically, the extent of accuracy and resolution error depends heavily on the arbitrariness of τ . Since all beams are sampled sequentially, the collected beam vectors must be shifted to align in time and phase. This is accomplished by applying a time delay correction filter to the incoming signal, shown in figure 3.6. The filter employs the general discrete-time Fourier time shift property defined by:

$$x(k - \tau) \xleftrightarrow{\mathcal{F}} e^{-j\omega\tau} X(\omega) \quad (3.34)$$

Thus, the following is true:

$$\mathcal{F}\{x(k - \tau)\} = X(\omega)e^{-j\omega\tau} \quad (3.35)$$

[24]. For this filter to work properly the commutation rate must not be arbitrary but set prior to data collection. The process to align beamspace data can be divided into

two separate tasks. First, set the beamformer commutation rate via precise FPGA control. Second, use this known rate to implement a time delay correction filter to synchronize the data. Overall, the time delay correction filter implements a Hilbert transform on the real data and applies a pure time delay in frequency space over all samples.

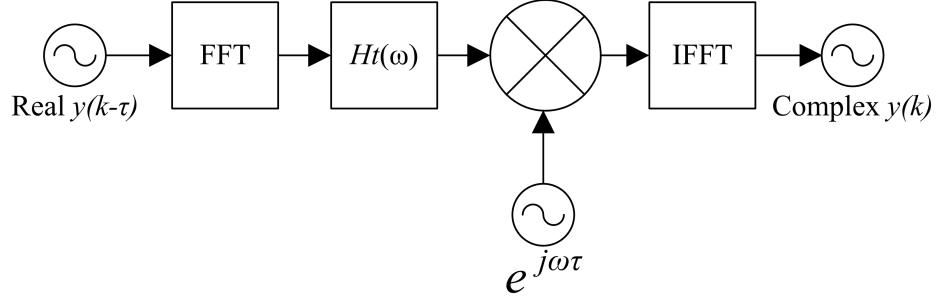


Figure 3.6: Time delay correction filter

Examining figure 3.6, the collected beamspace signal is converted to frequency space using the Fast Fourier Transform (FFT) and multiplied by the Hilbert transformer $Ht(\omega)$. The signal is then mixed with a complex delay-compensated signal, where τ is the commutation rate in terms of the sampling period, and ω is the discrete frequency of operation for each detected signal source in radians/second. The frequency term, ω , is an estimated value determined by a Capon spectrum estimator, discussed more in section 3.4. Next, the phase corrected signal is processed by the Inverse Fast Fourier Transform (IFFT). Finally, the output is an analytic – real and imaginary – temporal signal that has been synchronized – matched in time and phase – to a single reference beam. Examination of the synchronized trace (green) in figure 3.5 indicates that no accuracy or resolution performance has been compromised. The time delay correction filter eliminates all sampling phase error so the signal can now be processed using hi-res beamspace DOA estimation algorithms [3].

To completely “shift-out” the sampling phase error, certain assumptions have been made and highlighted throughout this work. Summarily, for the time delay correction filter to work properly, the mean source transmit power, distance, and DOA are assumed to be constant over the acquisition interval. Basically, this assumes no multipath, i.e. channel fading, or assumes the acquisition interval to be shorter than the channel coherence length. Also, because noise is a random variable it cannot be shifted; however, the mean noise power is assumed to be constant, meaning the SNR is constant over the acquisition interval.

3.2.4.3 Spatial Search Methods and Simulations: A technique to increase the flexibility and efficiency of SBS when spatially searching the array circumference for signal sources, is to bifurcate the searching process into a rough sector search and a fine sector search. This technique is in lieu of, or alongside, the standard all sector search used in previous simulations where all 360° search space is segmented into fine sector beams.

The rough sector search helps minimize the acquisition time to combat a fast fading channel, mentioned in section 1.4.3. Using a specific windowing function with lower side lobes and a larger HPBW, e.g. Hann or Blackman, to compose the beamformer matrix, the search space can be effectively covered with fewer beams. Table 3.1 lists some windowing functions, and their attributes when used with the UCA dimensions in section 2.1.1, suitable for wider beam creation.

Next, a simulation was run to compare the DOA accuracy, in RMSE degrees, between windowing functions when the minimum beam number is used to cover 360° , indicated by the minimum beam fan in table 3.1. The Monte Carlo simulation in figure 3.7 is composed of 1000 trials at each SNR level from 0 to 40 dB SNR. The UCA-RB-

Window Function	Side lobe max (dB)	HPBW($^{\circ}$)	Min beam fan
Triangular	-20.2	62	6
Hamming	-28.4	74	5
Hann	-30.9	87	5
Blackman	-45.3	98	4
Blackman-Harris	no side lobes	126	3

Table 3.1: Wide beam windowing functions with UCA $\beta\rho = 3.05$

MUSIC algorithm, used in previous simulations, was used with all windows to ensure the highest success. Also, the orthogonal shift and sum weights with an all sector beam fan of $N = 8$, were used for comparison as an indicator of low DOA estimation error. All windows, except for the Blackman window, achieve a RMSE accuracy below

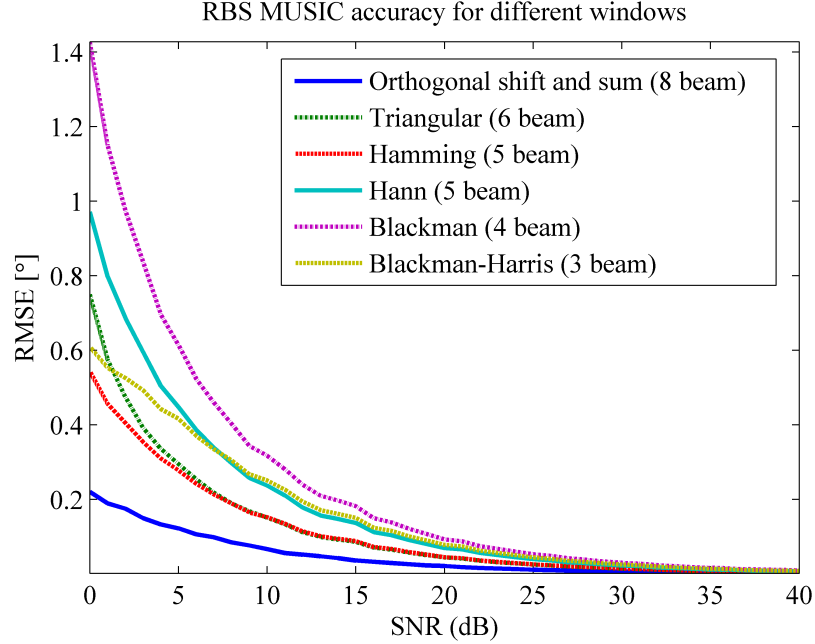


Figure 3.7: Beamspace DOA accuracy using different windowing functions

0.5° at a modest SNR of 5 dB, which would be necessary for a simplistic modulation scheme like binary phase-shift keying (BPSK) in a communications channel. At high SNR levels (> 15 dB), the RMSE accuracy for all windows does not exceed 0.2° and continues to converge toward the all sector orthogonal shift and sum window, used

as a benchmark in this plot. This simulation proves that most of these windowing schemes would be suitable for rough sector searching.

Once the rough DOA of the signal is ascertained, the fine sector search is implemented to evaluate the desired rough sector with closely spaced, narrower beams for more accurate DOA estimation. This is a zooming technique where the narrow beams are formed using a different set of windowing functions, e.g. shift and sum, orthogonal shift and sum, and Chebychev, yielding HPBW close to 45° . The attributes for these windowing functions when used with the UCA dimensions in section 2.1.1, are shown in table 3.2. Only a single rough sector is beamformed across, so the size of the beam fan can be increased to gather more observation data while fewer beams of the beam fan are needed to cover the rough sector; ergo, acquisition time and computational burden can be kept to a minimum while a DOA with a high degree of accuracy can be estimated.

Window Function	Side lobe max (dB)	HPBW($^\circ$)	Min beam fan
Shift and Sum	-8	42	8
Orthogonal S&S	-10.4	42	8
Chebychev	-8.1	47	8

Table 3.2: Narrow beam windowing functions with UCA $\beta\rho = 3.05$

A scenario simulation was completed to compare DOA estimation when detecting two sources located within the same rough sector. Two sources are impinging on the same rough sector, azimuthally at 60° and 120° , and are first detected with a rough sector search using a 3 beam Blackman-Harris window beam fan, shown in figure 3.8(a). Next, they are detected with a fine sector search using an 8 beam orthogonal shift and sum window beam fan. However, only 3 beams are needed to cover the 126° rough sector because the beams are spaced apart by 45° with a HPBW of 42° , yielding a coverage area of 132° . The reduced, 3 beam, fine sector beam fan is shown

in figure 3.8(b). These DOA results are compared to an 8 beam all sector orthogonal shift and sum beam fan, having a longer acquisition time and greater computational burden than either individual search. The DOA results are shown in figure 3.9.

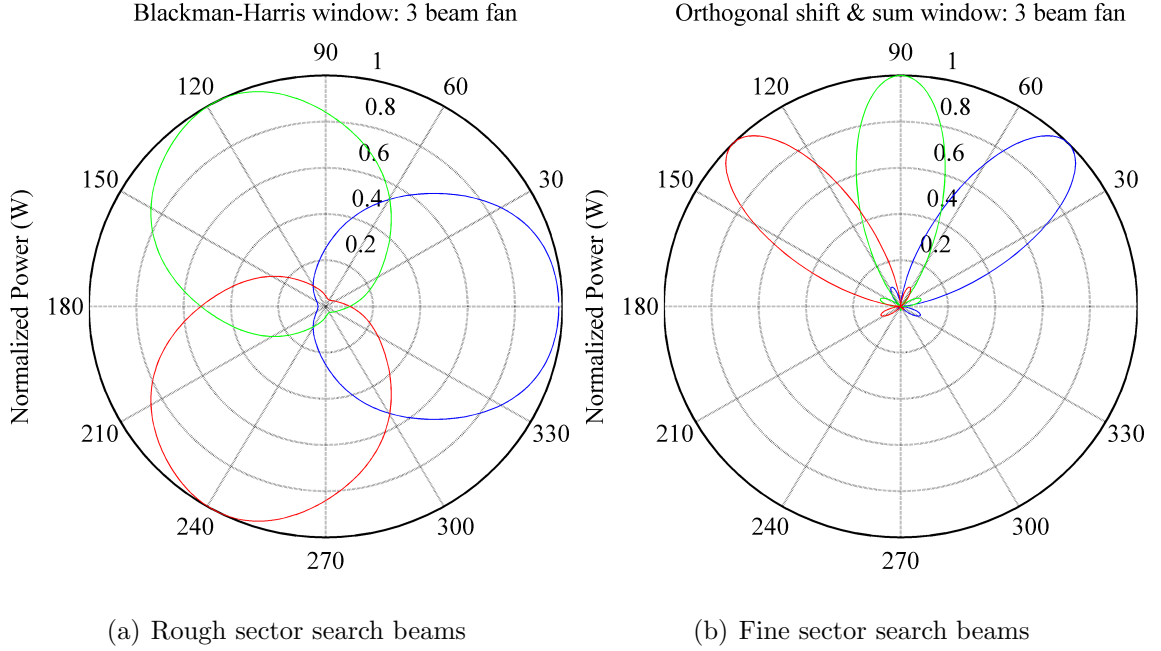


Figure 3.8: Rough and fine sector beam fans

The DOA pseudo-power spectra in figure 3.9 were all found using beamspace Capon DOA estimation at an SNR of 20 dB. The rough sector provides just enough definition to indicate a rough direction that the fine sector search can then zoom upon to accurately estimate the location of the two sources. Both searches combined take 75% of the time to complete compared to the all sector search, yet being individual searches they each finish in 37.5% of the all sector search. This is the beauty of dividing the searches; it gives the channel fading, time to recover. Also, one way to quantify the reduction in computational burden is to observe the complexity of a matrix inversion on the beamspace covariance matrix, when dealing with the Capon DOA method. It is well known that the complexity of a matrix inversion goes up

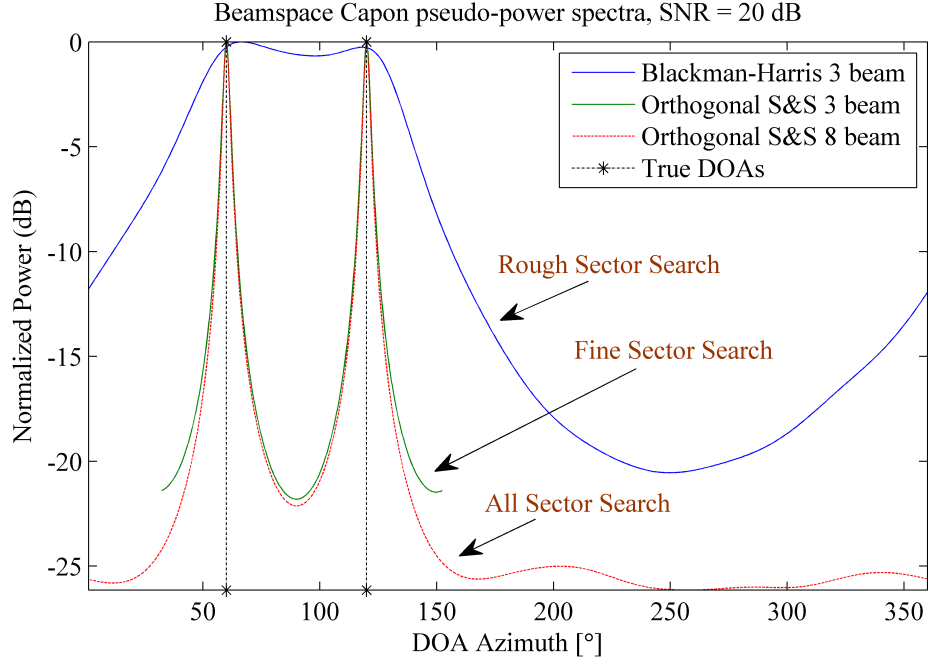


Figure 3.9: DOA using rough and fine sector searching

as the dimension cubed, $O(n^3)$ for an $n \times n$ matrix, where $O()$ is known as the Landau notation. The rough and fine sector searches effectively reduce the size of the beamspace covariance matrix, and in this scenario they each have a processing complexity of $O(27)$ Floating point Operations (flops) or $O(54)$ flops combined. However, the single all sector search has a processing complexity of $O(512)$ flops. Finally, from figure 3.9, there is minimal lifting in the noise floor, but the accuracy experiences no apparent degradation, proving the favorable application of the rough and fine sector spatial search methods [25].

3.3 Capon Beamforming

For beamforming in SBS, beamspace Capon beamforming – a method of optimal beamforming using beamspace information – combined with element space Capon beamforming is investigated and implemented in simulations.

3.3.1 Algorithm

The key to beamspace Capon beamforming, exactly as in element space, is to minimize the output power of the system except in the desired signal direction by finding the optimal set of weights. In element space an optimal combination of weighted channels is found, while in beamspace an optimal combination of weighted beams is found. Therefore, the quadratic cost function and linear constraint are the same, yet the Capon weights are computed using the beamspace manifold [3, 15, 25]. The beamformer average output power is:

$$P(\mathbf{w}_{es}) = \frac{1}{K} \sum_{k=1}^K |y(k)|^2 = \mathbf{w}_{es}^H \hat{\mathbf{R}}_{xx} \mathbf{w}_{es} \quad (3.36)$$

where $\hat{\mathbf{R}}_{xx} = \frac{1}{K} \sum_{i=1}^K \mathbf{x}(k) \mathbf{x}^H(k)$, is the element space sample covariance matrix. The subscripts $(\cdot)_{es}$ and $(\cdot)_{bs}$ have been enlisted to distinguish between element space weights (all previously discussed weights) and the beamspace weights, created with beamspace Capon beamforming. The proposed Capon optimization problem is then [15]:

$$\min_{\mathbf{w}_{es}} \mathbf{w}_{es}^H \hat{\mathbf{R}}_{xx} \mathbf{w}_{es} \quad \text{subject to} \quad \mathbf{w}_{es}^H \mathbf{a}(\phi) = 1 \quad (3.37)$$

More specifically, from [25] the constraint in element space is equivalent to the constraint in beamspace:

$$\mathbf{w}_{bs}^H \mathbf{b}(\phi) \equiv \mathbf{w}_{es}^H \mathbf{a}(\phi) = 1 \quad (3.38)$$

This is also further supported by eq. 3.40. Once the Capon DOA estimation is accurately determined from eq. 3.21, the beamspace spatial signature from eq. 3.10, as-

sociated with the given arrival angle, can be used to determine the Capon beamspace weights [18]:

$$\mathbf{w}_{bs,cap} = \frac{\hat{\mathbf{R}}_{yy}^{-1} \mathbf{b}(\phi)}{\mathbf{b}^H(\phi) \hat{\mathbf{R}}_{yy}^{-1} \mathbf{b}(\phi)} \quad (3.39)$$

Typical in conventional beamforming systems, since a beam fan observation data matrix is available instantaneously, beamforming is done in post-processing signal conditioning to arrive at the optimal beamformer scalar output, $y_{cap}(k)$; however, this cannot happen in SBS since beamspace data is available sequentially. In SBS, once all the beamspace data is collected, the optimal beamformer output is provided by transforming the beamspace Capon weights to element space Capon weights and applying these directly to the beamformer board. Once the element space Capon weights are computed, this method provides a real-time optimal beamformer output. Also, the original element space matrix beamformer has orthogonal column weights because $\mathbf{W}_{es} = \mathbf{W}_{\perp}$ is used, so the element space matrix beamformer is unitary. This truth is used for simplification to arrive at the element space Capon weights needed to weight the beamformer hardware. The beamspace Capon weights are further simplified to the following:

$$\mathbf{w}_{bs,cap} = \frac{(\mathbf{W}_{es}^H \hat{\mathbf{R}}_{xx} \mathbf{W}_{es})^{-1} \mathbf{W}_{es}^H \mathbf{a}(\phi)}{\mathbf{a}^H(\phi) \mathbf{W}_{es} (\mathbf{W}_{es}^H \hat{\mathbf{R}}_{xx} \mathbf{W}_{es})^{-1} \mathbf{W}_{es}^H \mathbf{a}(\phi)} = \mathbf{W}_{es}^H \mathbf{w}_{es,cap} \quad (3.40)$$

So, the element space Capon weights found via beamspace Capon beamforming are defined by:

$$\mathbf{w}_{es,cap} = \mathbf{W}_{es} \mathbf{w}_{bs,cap} \quad (3.41)$$

[25]. Finally, the Capon beamformer output, optimal because of the know DOA estimation statistics, is defined by the beamspace or element space Capon weights:

$$\begin{aligned}
 y_{cap}(k) &= \mathbf{w}_{bs, cap}^H \mathbf{y}(k) \\
 &= (\mathbf{W}_{es}^H \mathbf{w}_{es, cap})^H \mathbf{W}_{es}^H \mathbf{x}(k) \\
 &= \mathbf{w}_{es, cap}^H \mathbf{x}(k)
 \end{aligned} \tag{3.42}$$

3.3.2 Simulation Study

Using estimated DOA locations to apply Capon beamforming to the receiver can produce a highly correlated waveform to the transmitted signal. This technique allows accurate beam formation in the direction of the detected signal and null placement in the direction of the interference, to reduce noise and suppress interferers. In SBS, beamspace Capon beamforming estimates the DOA and calculates beamspace Capon weights; then, the weights are transformed to element space weights and sent to the beamformer. The beamformer output is a real-time Capon-filtered signal.

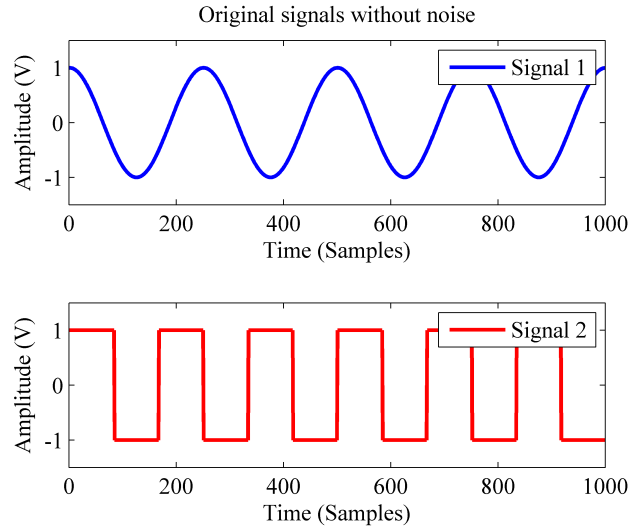


Figure 3.10: Original transmitted signals, a sine wave and square wave

Figure 3.10 shows the original signals used in this simulation, a sine wave and a square wave impinging upon the array at 155° and 115° , respectively. These were chosen solely for their distinct visual differences. The signals were then subjected to additive white Gaussian noise with an SNR of 10 dB, processed using beamspace Capon DOA estimation, and finally analyzed with various beamforming methods. The estimated Capon pseudo-power spectrum, detecting both sources simultaneously, is shown in figure 3.11.

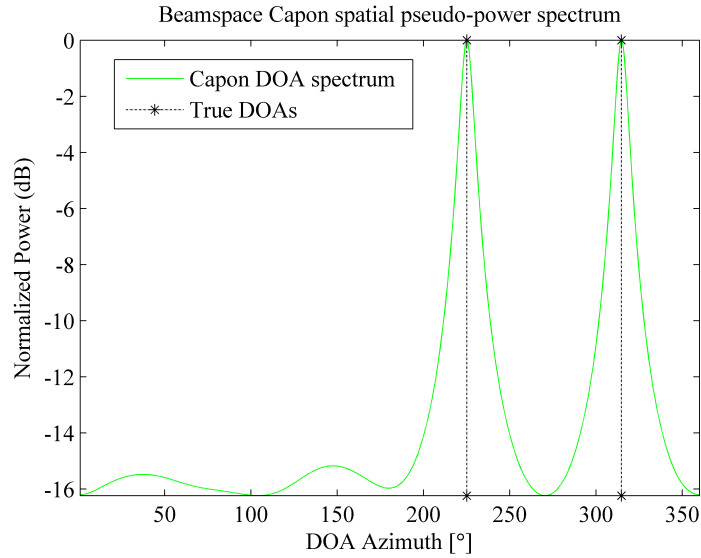


Figure 3.11: Beamspace Capon spatial pseudo-power spectrum

Figure 3.12 compares the output waveforms between classic beamforming (shift and sum method) and Capon beamforming, using the final element space weights. The location of the sources suggests that the logical application of a shift and sum beamforming algorithm would utilize two beams pointing in the direction of each source. Therefore, the shift and sum weights generate two beamformers shown in figure 3.13(a). The original sources, a sine wave and a square wave, have a respective correlation coefficient of 0.89 and 0.97 to the shift and sum beamforming output in figure 3.12(a), with 1 being the same.

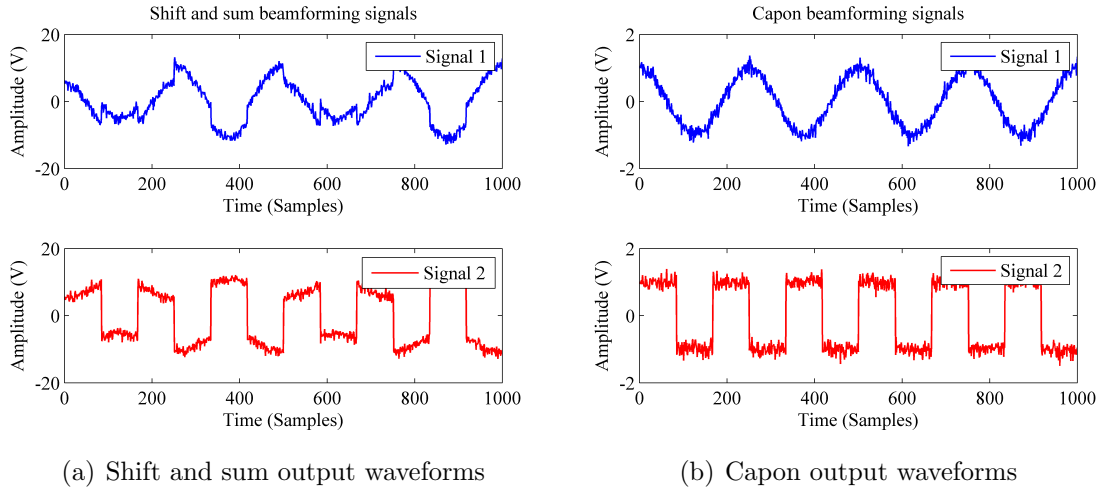
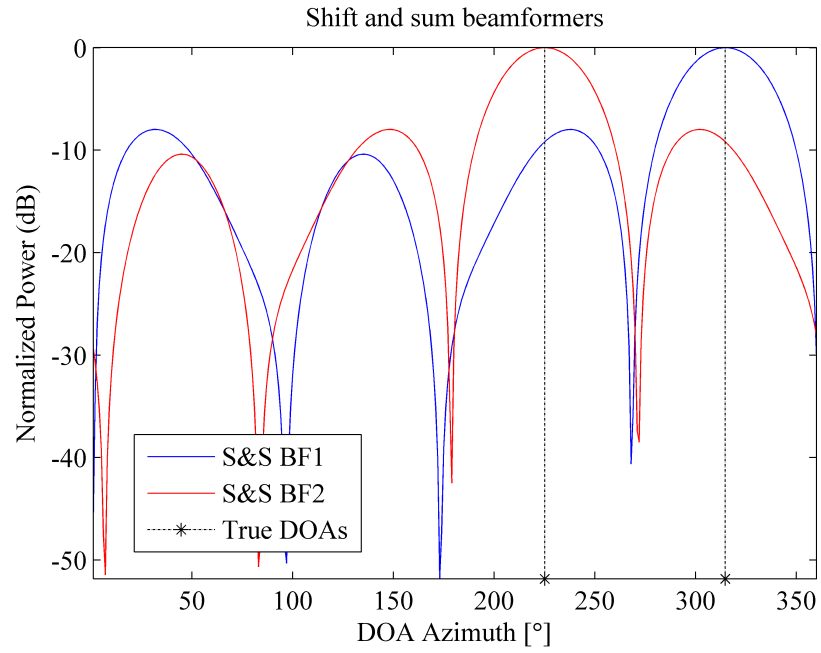
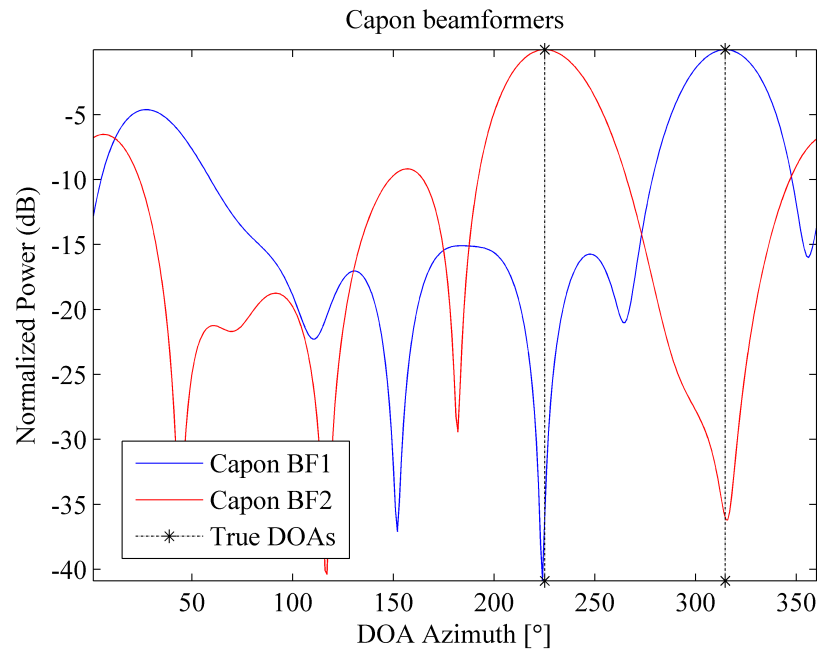


Figure 3.12: Beamformed received signals

Figure 3.12(b) shows the waveforms extracted from the interference of the adjacent signal, via the application of Capon beamforming. Looking at the Capon beamformers in figure 3.13(b), when beamforming in one direction a null is strategically placed at the other direction to mitigate its interference. With Capon beamforming, a set of optimal weights is paired with each detected DOA providing a new output stream for each direction. Both waveforms in figure 3.12(b) show excellent similarity to the original signals in figure 3.10. Their correlation coefficients with respect to the original signals are both 0.99. Clearly, the waveforms in figure 3.12(a) are more of a mixture of both sources than the waveforms in figure 3.12(b). This simulation proves the feasibility of calculating the beamspace Capon weights, using eq. 3.39, then converting to element space Capon weights, using eq. 3.41, to filter the output and ultimately improve signal quality. It also demonstrates the superiority of Capon beamforming to classic shift and sum beamforming [3,4].



(a) Shift and sum beamformers



(b) Capon beamformers

Figure 3.13: Beamformers

3.4 Calibration

The SBS calibration methods strive to remove error caused by two main sources. One, the phase shifters and attenuators that set the weights on the beamformer board are discrete devices. Because of this quantization and other implementation effects, e.g. signal path and part differences mentioned in section 2.2.3, exact phase and amplitude is unattainable, so a calibration method must be employed to compensate for the disparity between empirical weights and ideal weights. Second, error induced by quantized phase shifts, when synchronizing beamspace vectors with the time delay correction filter, needs to be minimized by accurately estimating the frequency of operation. One possible solution, implemented in this work, incorporates a linear array Capon spectrum estimator to reduce discrete phase shift error. A third origin of error is mutual coupling – signal crosstalk and interference – that occurs between antenna elements and adjacent channels on the beamformer printed circuit board (PCB). Error as a result of mutual coupling is more difficult to eliminate and is ignored in this work, but there is an ongoing effort to adapt a recursive batch least squares technique used in element space over to the beamspace realm.

The empirical weight compensation is the first calibration technique in use. The logistics, hardware, and software were discussed previously in section 2.2.3. A mathematical explanation of the calibration ensues. The ideal beamspace array signal, from eq. 3.7, and ideal spatial signature, from eq. 3.10, assume the element space data and spatial signature have been subjected to an ideal weighting matrix, $\mathbf{W}_\perp$. In actuality, the beamspace data is created with empirical weights on the beamformer board. If an uncalibrated beamspace covariance matrix, from eq. 3.15, is used with the ideal beamspace spatial signature for DOA estimation, large detection errors will

occur. If the empirical weights, $\mathbf{W}_{emp}$, are measured with the VNA via the method from section 2.2.3, they can be removed from the measured beamspace data:

$$\mathbf{x}_{meas}(k) = \mathbf{W}_{emp}^{-T} \mathbf{y}_{meas}(k) \quad (3.43)$$

Then, using the ideal matrix beamformer from eq. 3.9, a calibrated beamspace array signal can be used for beamspace DOA estimation:

$$\mathbf{y}_{cal}(k) = \mathbf{W}_{ideal}^H \mathbf{x}_{meas}(k) \quad (3.44)$$

A simulation of the RMSE accuracy when searching a 360° space every 4.5° is presented in figure 3.14. The output is a result of using empirical weights, measured in the lab, inserted into the simulation to show the DOA error induced from not calibrating the weights with the proposed method. Identical data sets, operating at an ideal frequency, i.e. frequency located at the center of an FFT bin, centered at 5.8 GHz, subjected to a 30 dB SNR ratio and having an SBS commutation rate of 2000 samples were used. With and without empirical weight calibration the RMSE accuracy is 0.02° and 13.92° , respectively.

The next calibration technique employs the use of a Capon spectrum estimator for hi-res and highly accurate signal source frequency of operation estimation. The detected frequency is passed to the time delay correction filter to minimize any quantization errors associated with the discrete phase shift. The term $\omega\tau$, within the time delay correction filter, causes a shift in the phase spectrum of the observation data sequence and has a resolution only as fine as the number of samples captured. To minimize phase shift error, the granularity of ω is increased by estimating the spectrum using a Capon based algorithm. First, samples are captured in an omnidirectional antenna mode, and then the data vector is formatted to artificially emulate

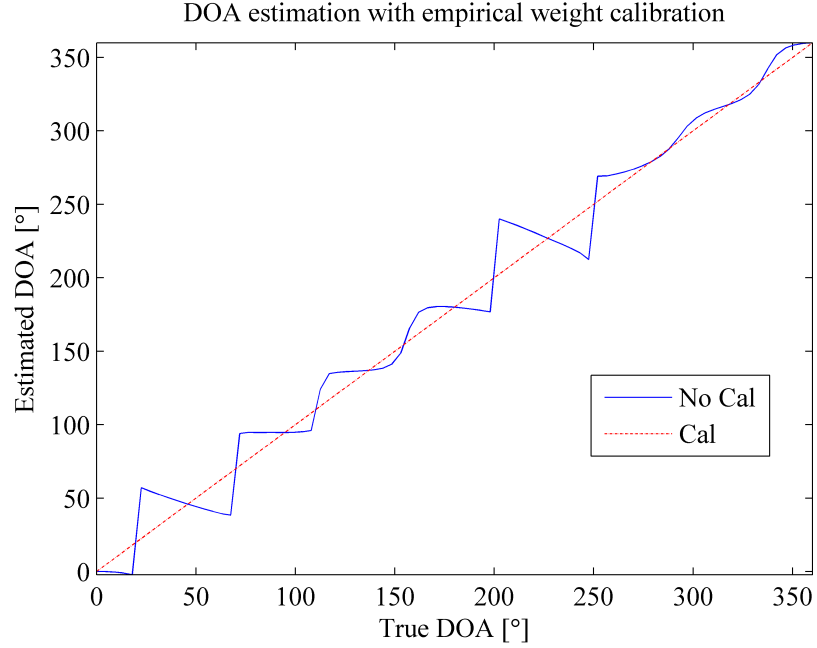


Figure 3.14: Empirical weight calibration simulation

a large linear array. The frequency resolution and accuracy increase as the size of the array grows, but this is also proportional to the computational complexity because for every new linear element the covariance matrix dimensions grow by one. The Capon estimator uses the inverse of the covariance matrix; therefore, the complexity increases by the power of three. The simulation and lab-based systems use a 31 element linear array to compute the estimated frequencies.

Again, a simulation of the RMSE accuracy when searching a 360° space every 4.5° is presented in figure 3.15. The outputs are the calibrated result, using the Capon estimated frequency for ω , versus the uncalibrated result, using the FFT frequency vector for ω . Identical data sets, operating at a random discrete frequency, 191.9606 KHz , applied to a 5.8 GHz carrier, subjected to a 30 dB SNR ratio and having an SBS commutation rate of 2000 samples were used. With and without phase shift calibration the RMSE accuracy is 0.02° and 20.21° , respectively. This is one example

at one frequency, and the RMSE accuracy for the uncalibrated data will cyclically fluctuate as the operation frequency trends to and away from the center of an FFT bin, which depends on the sample rate and the sample number.

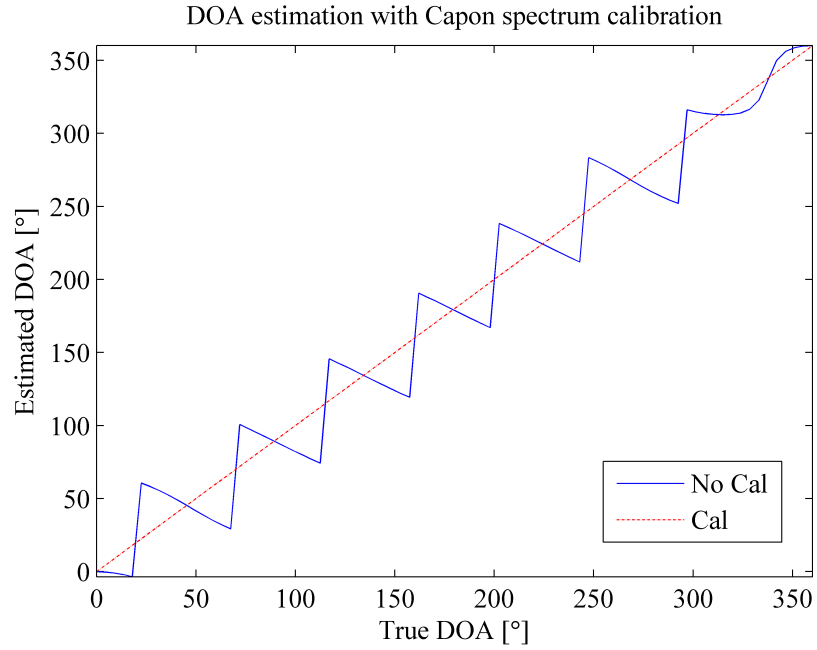


Figure 3.15: Capon spectrum calibration simulation

The simulations prove the need for empirical weight and phase shift calibration by radically reducing the DOA RMSE accuracy in both cases by more than an order of magnitude.

CHAPTER 4

EXPERIMENTAL RESULTS

The SBS technique was implemented in the lab, relying on resources from sections chapter 2, to validate the operation and advantages while using a real system and signals. Numerous lab-based tests have been performed to support the SBS technique with a single beamformer. First, the lab resources and procedure will be discussed. Then, test results are presented that exhibit the necessity of synchronization via the time delay correction filter, accurate DOA estimation with one and two signal sources, DOA estimation using rough and fine sector searches, and shift and sum and Capon beamforming; all, while under the umbrella of SBS [26].

Note, all experiments are proof of concept, thus utilizing continuous wave (CW) signal sources with an SNR of 30 dB or greater. Also, the receiver samples at a rate of 2.5 MSPS, down-converting the 5.8 GHz CW signal to baseband and effectively operating within 1.25 MHz of bandwidth. All specified frequencies are referenced to baseband, being originally down-converted from a 5.8 GHz carrier. Last, all subsequent experiments are done using the UCA-RB-MUSIC algorithm, detailed in section 3.2.3.3, unless otherwise specified.

4.1 Lab Setup and Testing Procedure

This section summarizes the lab environment, specifically the equipment available for testing and the procedure that is followed to get acceptable SBS results.

4.1.1 Equipment

The Communications Lab at MSU is equipped with many pieces of sophisticated lab equipment to carryout tests to prove the concept and operation of SBS. The central area of test is an anechoic chamber, used to mitigate RF signal reflections with microwave absorber foam, equipped with a stepper-motor based turntable for precise rotational positioning of the UCA and beamformer. The opposite side of the chamber holds a calibrated horn antenna for transmitting an RF signal from 4.9 to 7.05 GHz. The antenna positioner is controlled with an NI DAQ card (PCI-6013) and various LabVIEW and MATLAB based programs. A key piece of instrumentation is the 8720D Hewlett Packard VNA, used for beam calibration and empirical weight measurement on the beamformer board. A 68369A/NV Anritsu signal generator is utilized to generate a 5.8 GHz sine wave that is cabled to the horn antenna for transmitting. The R3273 Advantest spectrum analyzer provides the ability to measure energy at a frequency of interest for displaying beam patterns. All equipment is connected in parallel via a General Purpose Interface BUS (GPIB) and routed to a central desktop PC for ease of control and measurement data capture. Also, various Tektronix oscilloscopes are available for precise capture and triggering of time waveforms, essential to the debugging process. Finally, two more NI DAQ modules, mentioned in section 2.2.3 (USB-6501 and PCI-6133), are PC operated to toggle control switches on the beamformer and sample beamspace data at a rate of 2.5 MSPS. The harmonious handshaking of said equipment provides a complete environment, shown in figure 4.1, for validating the SBS technique.

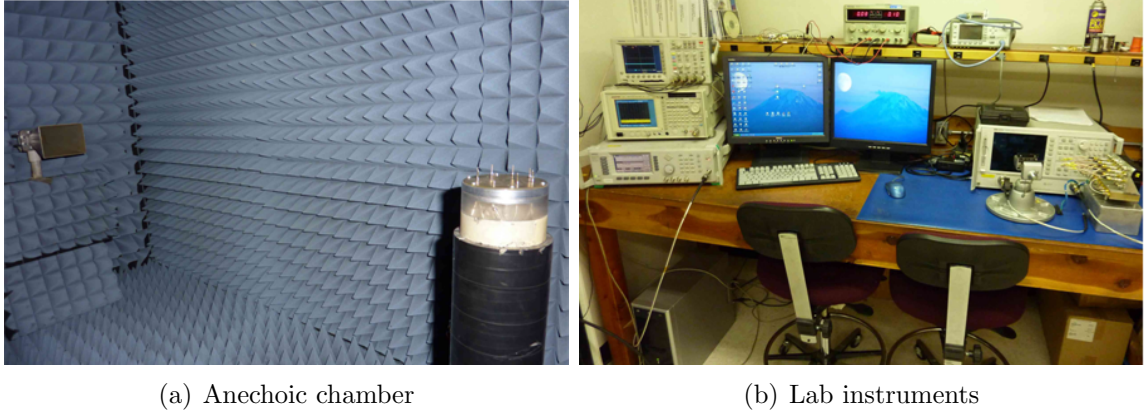


Figure 4.1: Lab setup

4.1.2 Procedure

Before lab results are presented, a chronological explanation of the lab procedure, including hardware calibration and setup, data capture, and software processing must be described. The following steps all contribute to the final objective, an accurate DOA estimation via SBS techniques.

Initially, the 8 orthogonal shift and sum beams, or the all sector beam fan, must be calculated according to the UCA electrical size, 3.05, and center frequency, 5.8 GHz, using eqs. 3.8 and 3.9, and calibrated. The calibration procedure using an 8 channel, high isolation relay test setup and VNA is detailed in [27,28]. Essentially, the process reduces the disparity between ideal weight amplitude and phases and the unavoidable approximations realized on the beamformer board because of quantization effects. The method dithers through the phase shifter and attenuator settings to find optimal settings, offering the least amount of error. However, the calibration process must be augmented for accurate DOA capabilities, and once the weights are calibrated they are measured with the VNA. The measured empirical weights are saved on the PC and used to calibrate the beamspace data later in the DSP chain, described in section 3.4. A distinction must be made between the calibrated weights, which are discrete

calibrated weight settings, and the empirical weights, which are analog measurements of these calibrated weights once they are set on the beamformer.

Next, the calibrated weights are converted to a hexadecimal equivalent and inserted into the SBS firmware, see section 2.2.2. Up to this point, the same process is done with a 4 beam Hann window beam fan used for rough sector searching. The orthogonal shift and sum beam fan is used for all sector searching or a portion of the beam fan is used for fine sector searching. The updated version of SBS firmware is now flashed onto the controller FPGA on the beamformer board.

The beamformer system is placed inside the anechoic chamber with all necessary connections: power, RF, and DAQ IO. The unit is placed on the antenna positioner for automated positioning. The RF signal path from the beamformer is routed out of the chamber to the single channel receiver and NI ADC for sampling. The Anritsu signal generator excites a 5.8 GHz signal in the chamber, which is transmitted from the horn antenna. The SBS GUI, from section 2.2.3, allows the user a multitude of functions to evaluate beamspace data. When toggled, the beamformer initially captures a large amount of samples, 20k - 40k, in omni-directional mode for frequency spectrum detection and then commutates beams, collecting a minimal amount of samples, 2000 per beam, in a single 360° sweep. The beamspace matrix is saved in the MATLAB environment, calibrated with the stored empirical weights, filtered through the time delay correction filter using the estimated frequency for calibrated phase shifts, and finally evaluated with beamspace DOA algorithms. The result is an accurate DOA estimation for a signal of interest, and all data is saved for research purposes and post-processing analysis.

4.2 Synchronization Validation

The first issue is to reinforce the necessity of the time delay correction filter, as applied to real SBS data, to realign the phase fronts of the beamspace data. The filter synchronizes the phases of each beam vector as if they were all sampled simultaneously using an N number of beamformers. This process reduces, to a high degree, the phase error induced from sequential beam sampling.

A signal, approximately 100 KHz, strikes the array head from an azimuth arrival angle of 333° . Examining figure 4.2, the synchronized data predicts the correct DOA with a noise floor minimum of -12 dB while the data corrupted by the sampling phase error exhibits a main peak at 313° and a false peak at 15° , with a noise floor minimum of -3 dB. An accuracy error of 20° and an increase in the noise floor of 9 dB (relating to a decrease in resolution) is unacceptable for this type of hi-res DOA algorithm and would destroy emitter localization or concomitant beamforming abilities.

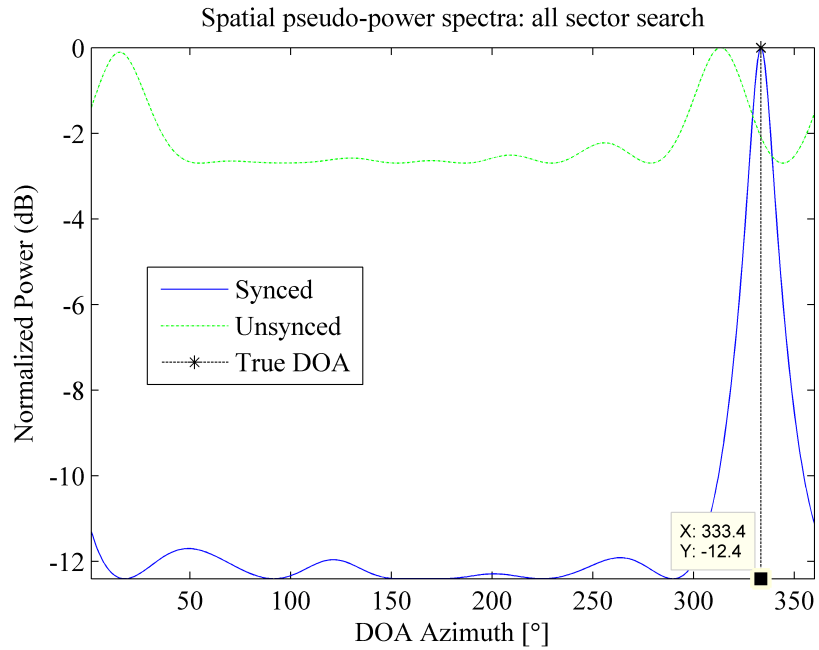


Figure 4.2: Example 1: synchronized and unsynchronized comparison

In a second example, a signal at approximately 100 KHz arrives at 235° ; subjected to an all sector and rough sector search the results are shown in figure 4.3. In figure 4.3(a), the synchronized DOA estimation again predicts the correct arrival angle with a noise floor minimum near -11 dB. The unsynchronized DOA estimation predicts a DOA of 225° , yielding an accuracy error of 10° , while the noise floor increases by 3 dB. This same setup and positioning was then used to collect rough sector search data, to ensure to same outcome when using less beams. Figure 4.3(b) shows the proper DOA prediction using rough sector synchronized data, while the unsynchronized spatial spectrum is wildly inaccurate.

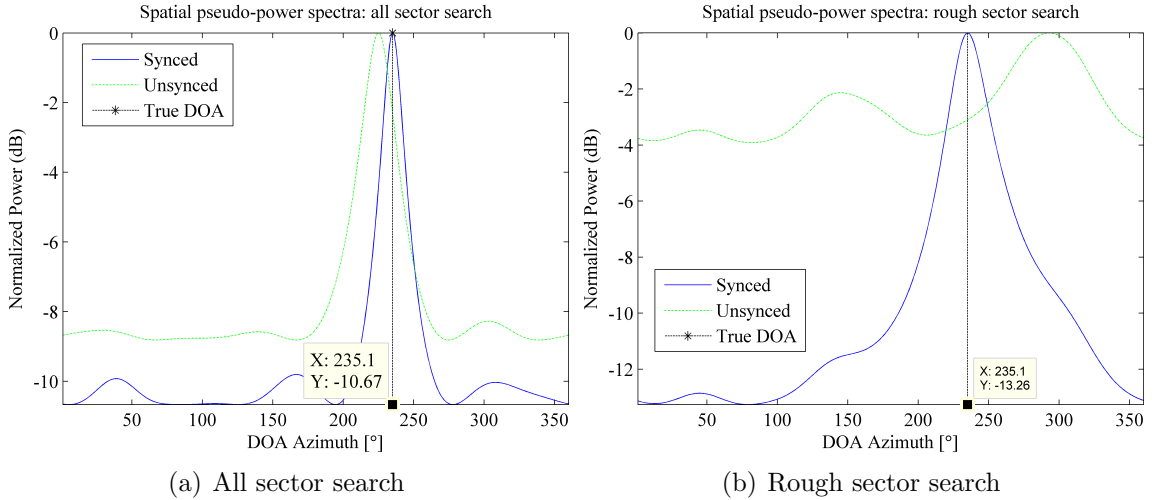


Figure 4.3: Example 2: synchronized and unsynchronized DOA comparison

These examples are a sampling of the corruption that occurs when SBS data is not synchronized using the time delay correction filter. Also, they offer a qualitative perspective on the type of favorable results produced when the data is synchronized. Shown by these experiments, the time delay correction filter is an imperative step in the SBS process.

4.3 DOA Estimation

Accurate DOA estimation is the most important metric to verify the SBS technique in the lab. SWB accuracy results from the lab are shown to provide a basic benchmark of DOA estimation abilities. Also, one source and two source experiments characterize the accuracy and resolution performance of SBS and some basic sensitivity specifications are given. Furthermore, rough and fine sector search experiments are carried out to verify this useful spatial search method.

Note, a measurement technique was developed involving the automated acquisition and processing of beamspace data over 360° every 4.5° . This technique is used in many subsequent tests and is entitled the “full-360” test. The impetus for this test was to minimize the error induced by the open-loop antenna positioner while spatially compiling DOA RMSE accuracy data. The full-360 test begins a rotation, takes the first measurement, and zeros this point so all measurements are relative to the first. This test is carried out with a MATLAB script, involving numerous other custom function calls, called *automated_positioner.m*. The significant portions of code are attached in Appendix B.

4.3.1 Switched Beam

A SWB technique was executed on the beamformer board by measuring the RMS voltage of each beam, using an 8 beam sweep, and estimating the DOA as the pointing direction of the beam with the largest RMS voltage. A comparison of the measured RMSE accuracy with the theoretical results from section 3.2.2 can be made. A full-360 experiment was completed, and the RMSE accuracy was found. The measured RMSE accuracy was 13° , completely corroborating the theoretical results. The lab

measurements are shown in figure 4.4. This test validates the measurement methodology and provides a minimum level of performance for comparison.

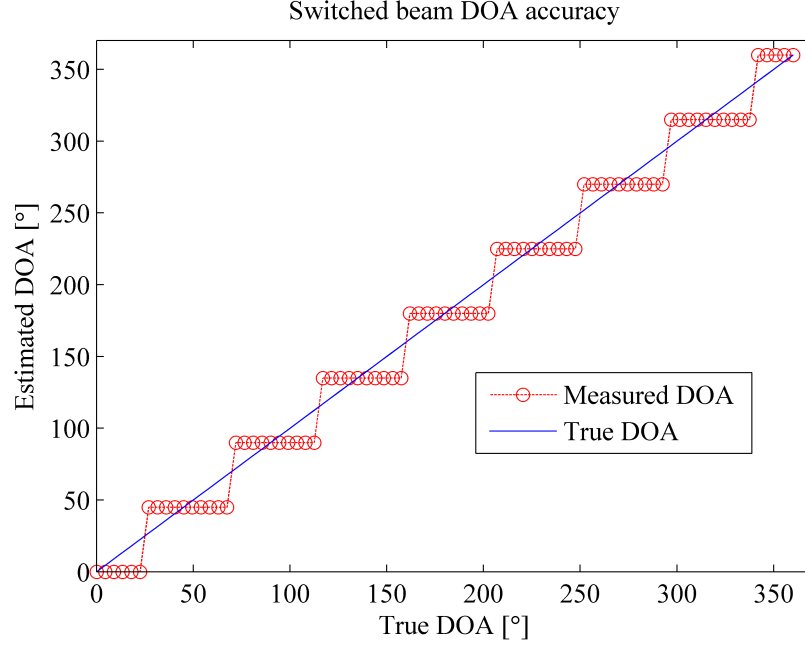


Figure 4.4: Switched beam DOA accuracy

4.3.2 One Source

Measurements were made to characterize the DOA accuracy using the standard all sector search with one signal source. Two experiments were conducted using a low frequency and a high frequency source to adequately span the baseband bandwidth. Both experiments were full-360 tests. The low frequency, approximately 80 KHz, results are shown in figure 4.5 while the high frequency, approximately an order of magnitude greater at 870 KHz, results are shown in figure 4.6. The RMSE accuracy is 1.6° and 1.8° for each respective signal.

Spatial pseudo-power spectra are shown in figures 4.7 and 4.8 to provide the DOA spectrum shape for single source measurements in SBS mode. Also, the average

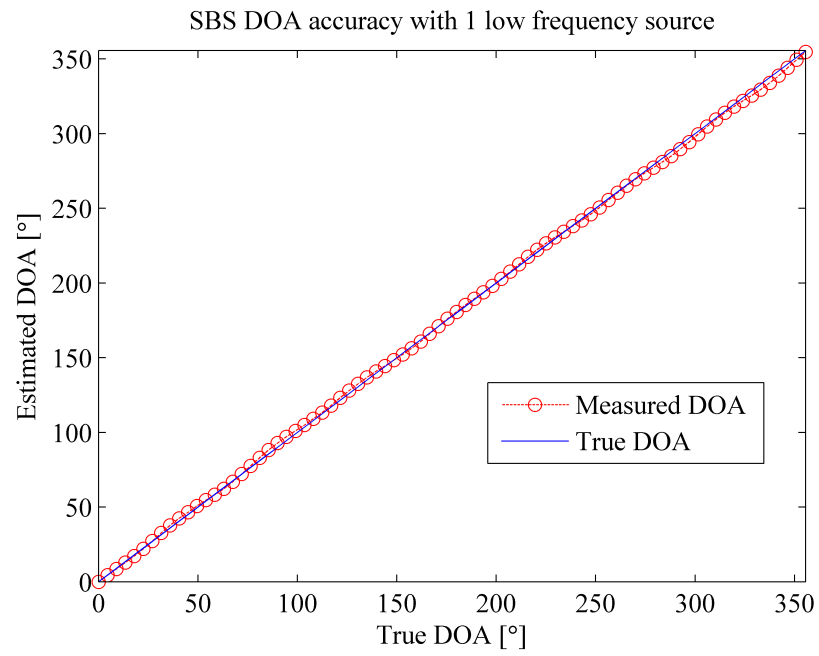


Figure 4.5: SBS DOA accuracy with 1 source at 81 KHz

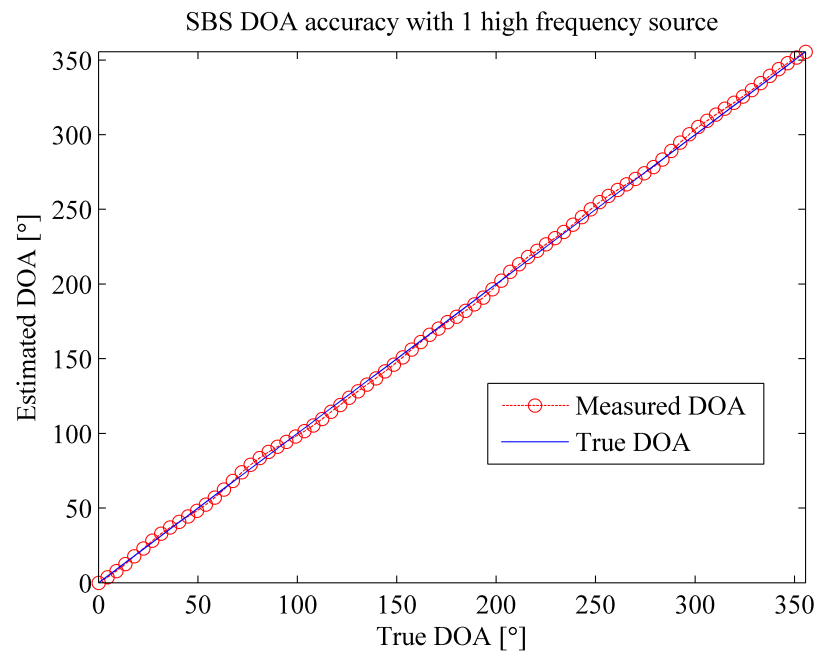


Figure 4.6: SBS DOA accuracy with 1 source at 873 KHz

minimum noise floor value for each 360° sweep, low and high frequency, were -8.5 dB and -8.2 dB, respectively.

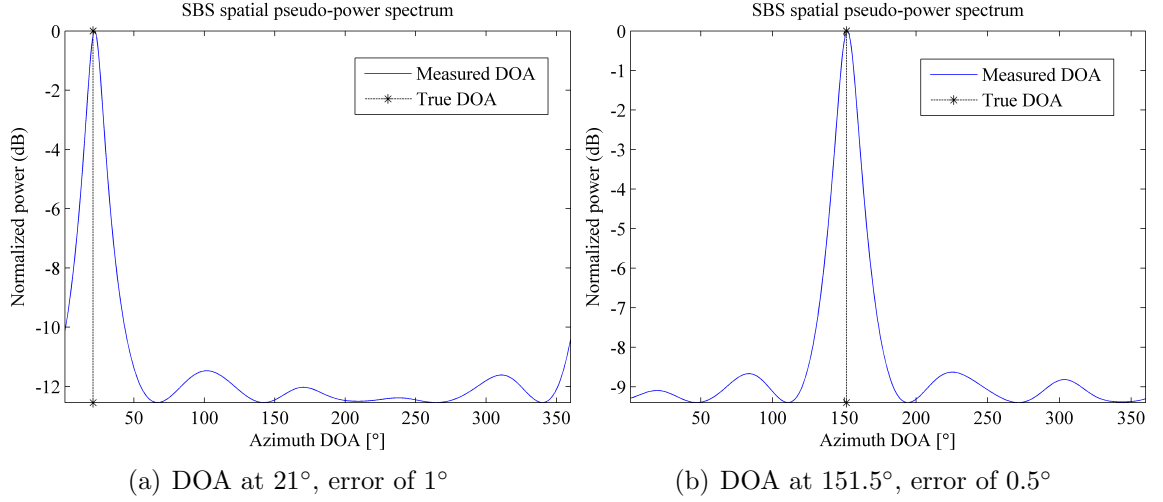


Figure 4.7: SBS spatial pseudo-power spectra with 1 source at 81 KHz

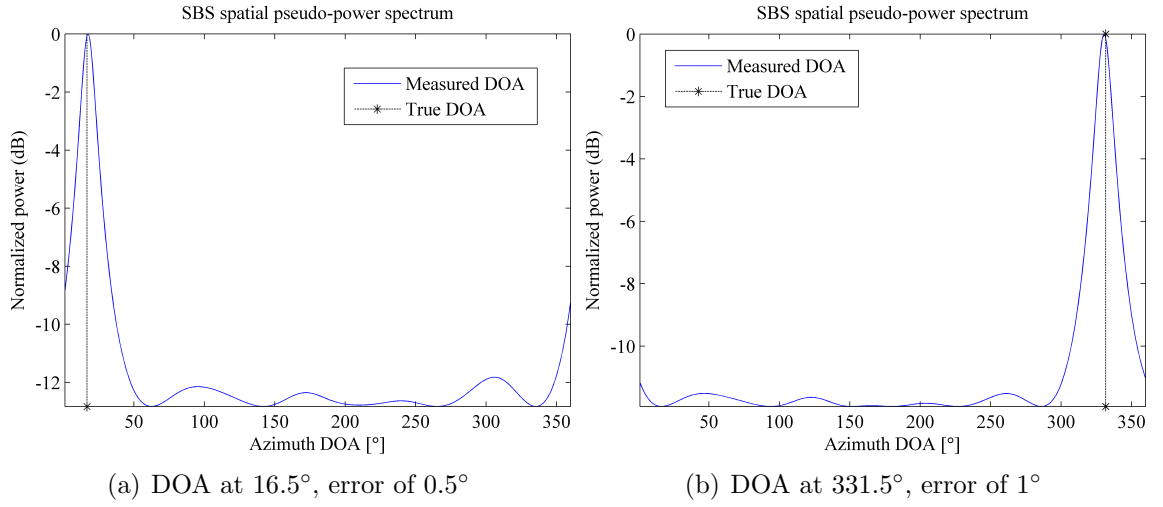


Figure 4.8: SBS spatial pseudo-power spectra with 1 source at 873 KHz

4.3.3 Two Sources

Measurements were taken at various frequencies to quantify the accuracy and resolution when two signal sources impinge upon the array simultaneously with ap-

proximately the same power levels. Two full-360 tests were completed, one when signals were widely separated in frequency and one when they were close in frequency. The RMSE accuracy is 2.8° and 2.3° for two sources when they are widely separated in frequency, approximately 300 KHz. This test result is shown in figure 4.9, where an angular resolution of 80° is achieved. The RMSE accuracy is 3.7° and 2.6° for two sources when they are close in frequency, approximately 50 KHz. The test result is shown in figure 4.10, also where an angular resolution of 80° is achieved. The decline in RMSE accuracy with two signal sources close in frequency was expected and a subject of previous research at MSU in [29].

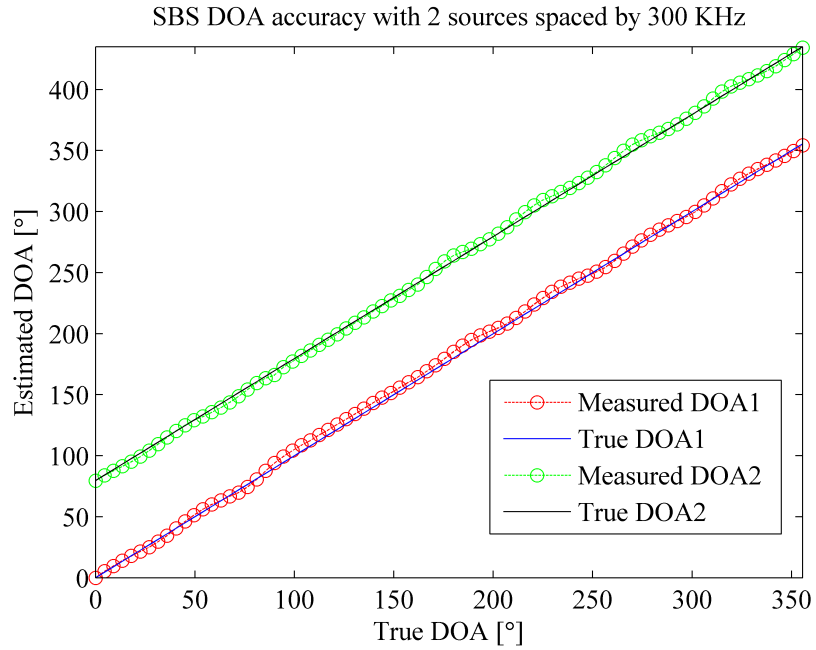


Figure 4.9: SBS DOA accuracy with 2 sources spaced by 300 KHz

The following two experiments show the spatial pseudo-power spectrum with two signal sources at different resolutions. Each source is toggled on while the other is toggled off, and the DOA is estimated. The true DOAs for both sources are assumed to be when they are activated as the single source. Figure 4.11 shows the progression

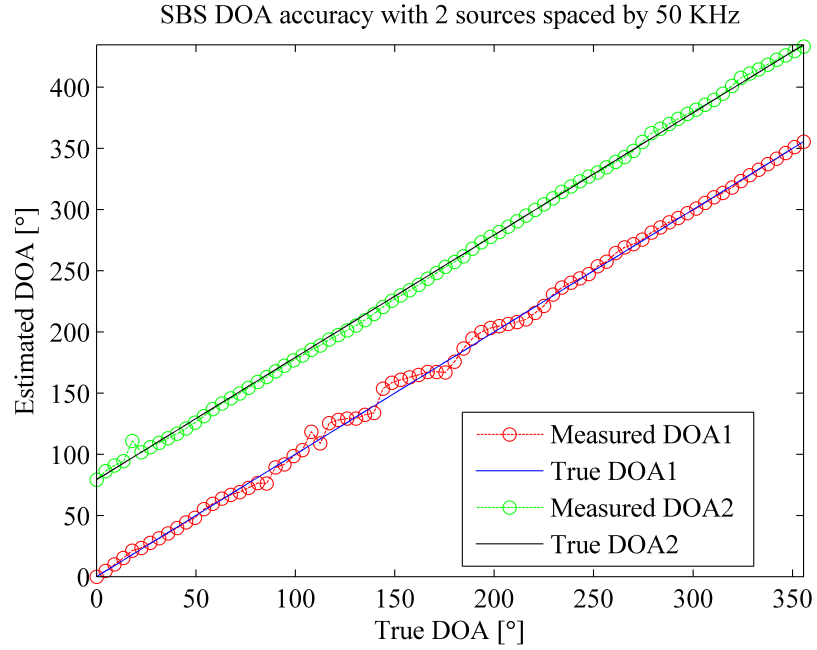


Figure 4.10: SBS DOA accuracy with 2 sources spaced by 50 KHz

of the first experiment as each signal source is toggled on and off, and finally both are toggled on with minimal accuracy error. This was performed at an angular resolution of 60° . Figure 4.12 shows the progression of the second experiment as each signal source is toggled on and off, and finally both are toggled on at an angular resolution of 97° .

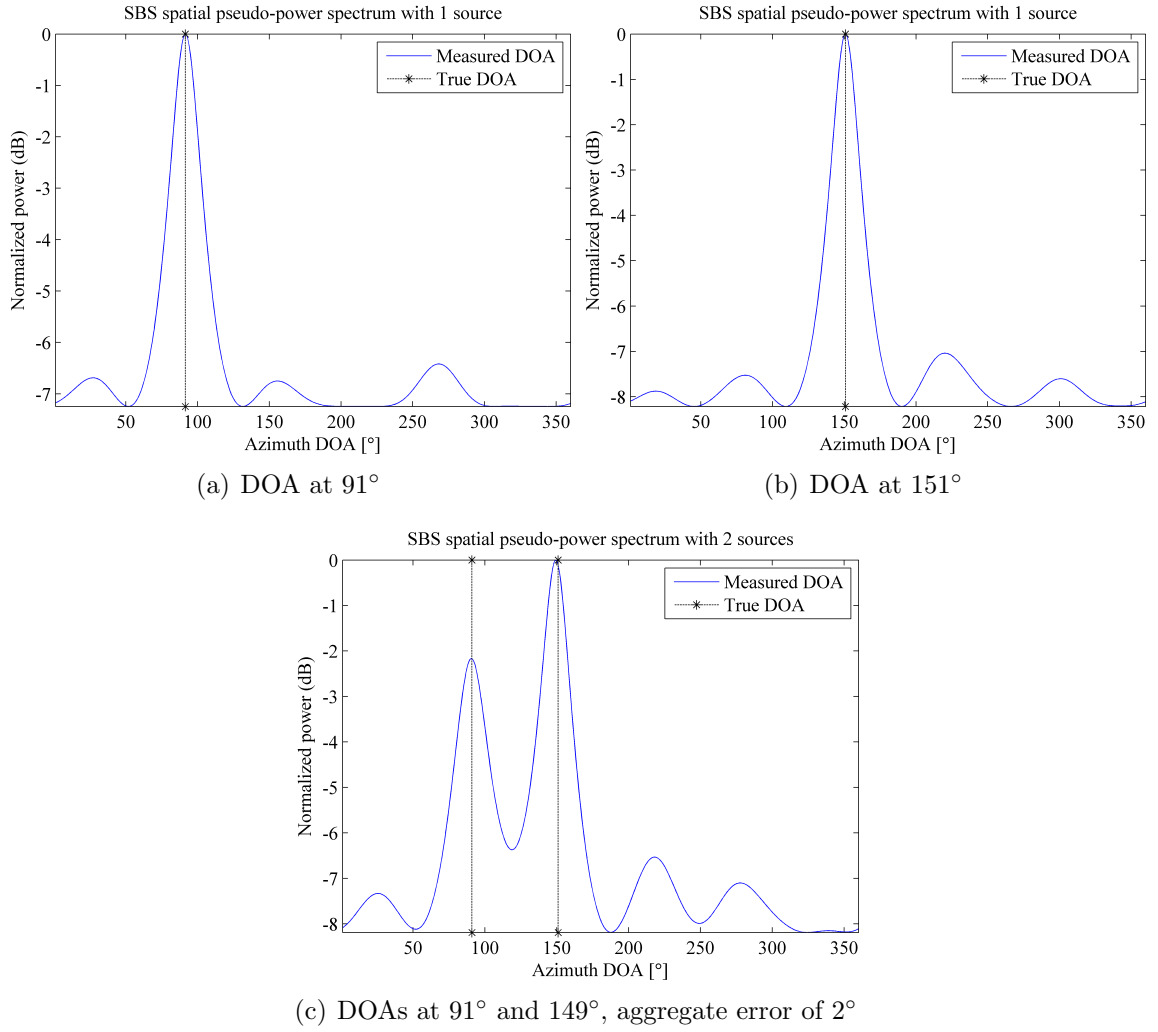


Figure 4.11: SBS spatial pseudo-power spectra for 2 sources

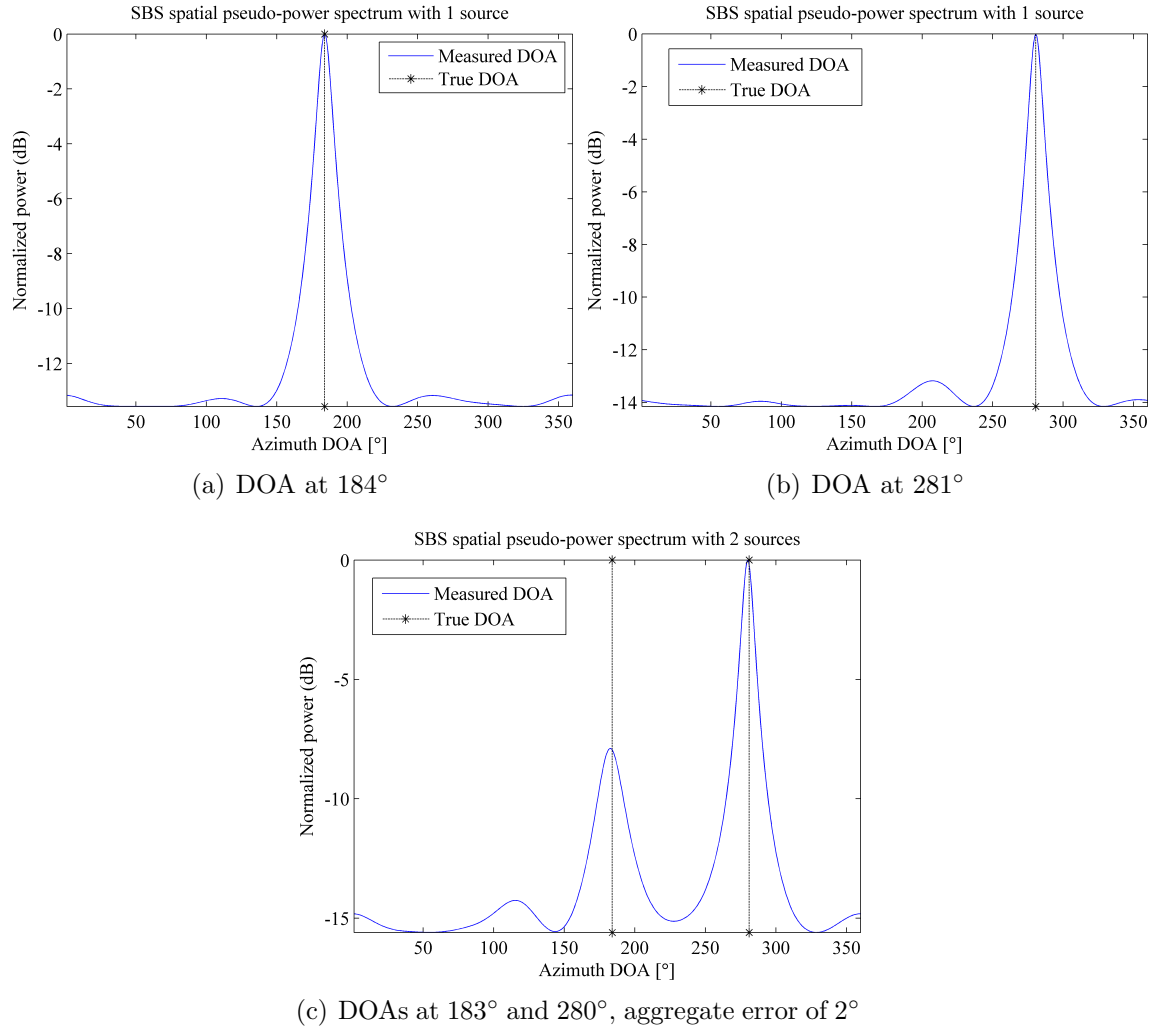


Figure 4.12: SBS spatial pseudo-power spectra for 2 sources

4.3.4 Minimum Detectable Signal

Next, an effort was made to measure the accuracy error as the power level of the impinging signal source drops to very low levels, i.e. characterize the minimal detectable signal (MDS) or sensitivity of the SBS system. The UCA was positioned so the signal from the horn antenna would arrive at approximately 23° , halfway between the first and second antenna elements. This was verified with the first DOA measurement, being equal to 23° , which corresponded to a power level of -78 dBm at the input of the UCA. With a carrier frequency of 5.8 GHz, the link budget is the following: -30 dBm out of the signal generator, 14 dB cable attenuation, 15 dB horn antenna gain, and 49 dB of free space path loss in the anechoic chamber for a total path loss of 48 dB. The output power of the signal generator was monotonically decreased until the accuracy error exceeded a certain limit. The limit is based on the previous RMSE accuracy measurements for one source, so error above 2° is considered unsatisfactory. Figure 4.13 shows the spatial pseudo-power spectra as the power decreases, and table 4.1 summarizes the results, showing the sensitivity to be approximately -111 dBm at the input to the UCA. Furthermore, results from [30] for a similar test using identical UCA hardware, yet in element space, measured an MDS of -90 dBm. These results help shore the claim in section 1.4.2 that SBS would experience at least an 8 factor (9 dB) gain advantage over a similar element space approach since SBS uses beamspace data for DOA estimation.

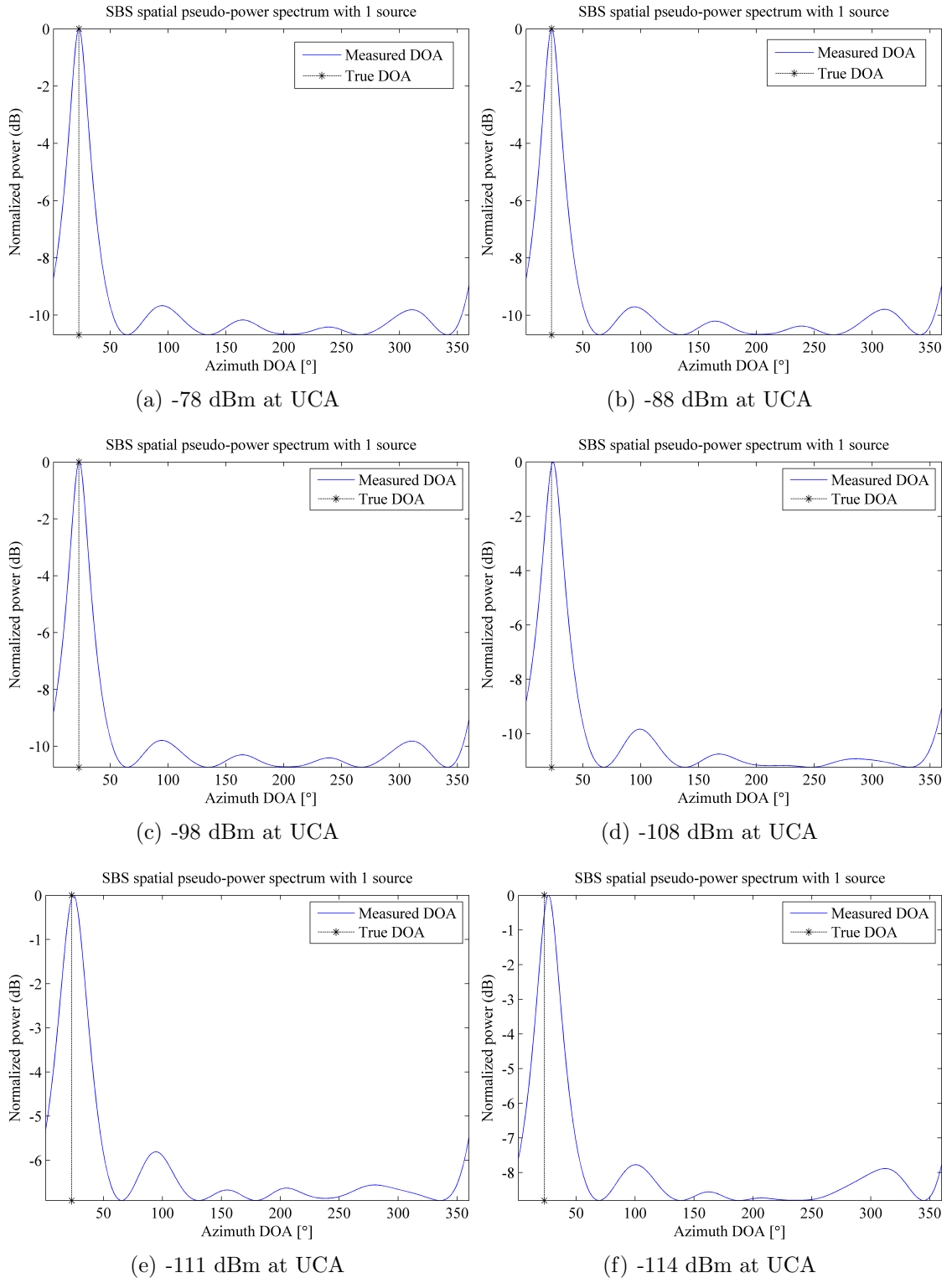


Figure 4.13: SBS sensitivity according to DOA accuracy error

RX Power (dBm)	True DOA [$^{\circ}$]	Measured DOA [$^{\circ}$]	Error [$^{\circ}$]
-78	23	23.0	0.0
-88	23	23.1	0.1
-98	23	23.2	0.2
-108	23	24.2	1.2
-111	23	24.5	1.5
-114	23	26.6	3.6

Table 4.1: SBS sensitivity according to DOA accuracy error

4.3.5 Rough Sector Search

A beam fan composed of 4 beams, using a third order polynomial Hann window, was calibrated for the beamformer board and implemented as a rough sector search scenario. The measured beam patterns are shown in figure 4.14 with a mean HPBW of 101° . A full-360 test, results shown in figure 4.15, was completed using one signal source at an approximate frequency of 560 KHz. The RMSE accuracy is 11.5° , performing better than SWB at 13° .

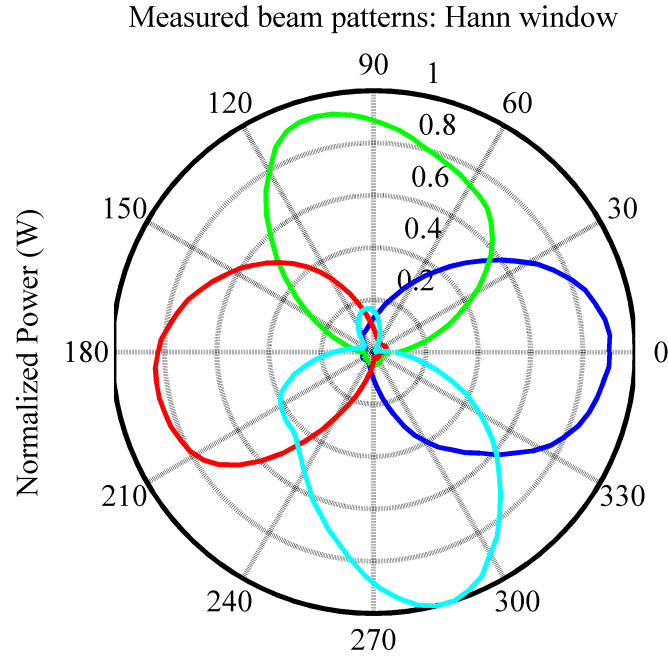


Figure 4.14: Measured Hann beam patterns: 4 beam fan

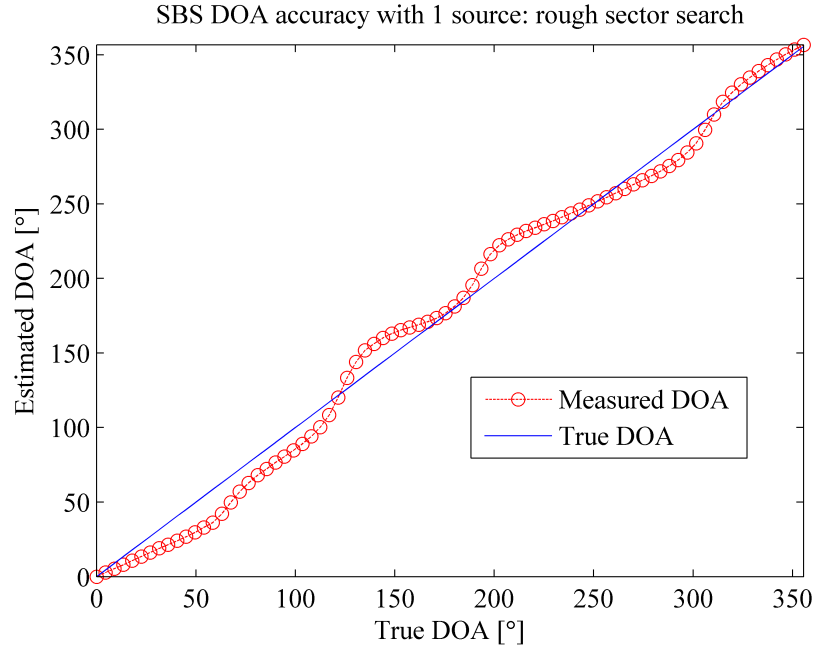


Figure 4.15: SBS DOA accuracy using a rough sector search

Spatial pseudo-power spectra are shown in figure 4.16 as an example of the DOA spectrum output using rough sectors. The beamwidth widens, increasing the potential angular separation between sources, and the accuracy decreases compared to the all sector search; however, the acquisition time halves, and the accuracy is still better than SWB.

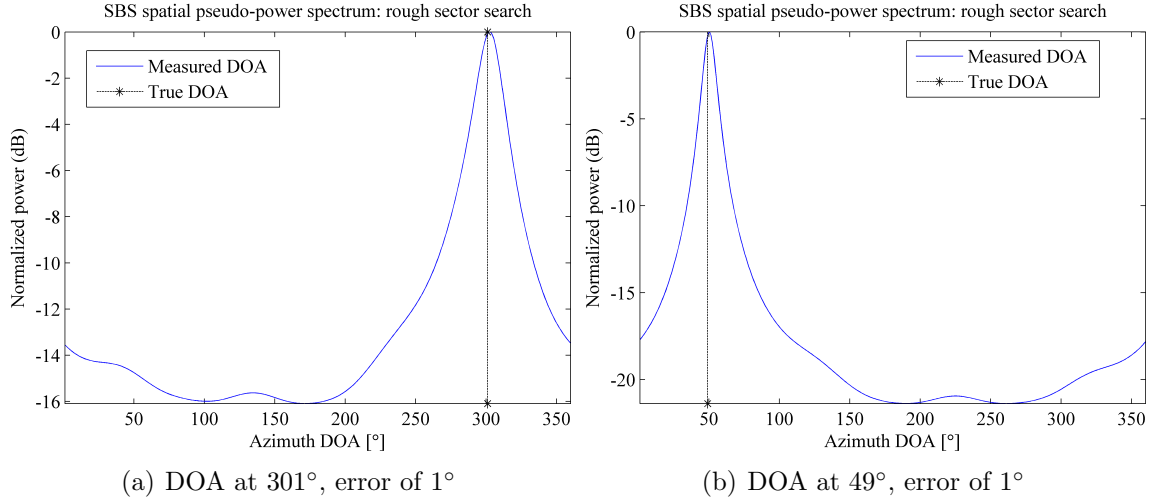


Figure 4.16: SBS spatial pseudo-power spectra for a rough sector search

4.3.6 Fine Sector Search

A fine sector search experiment was carried out using the 3 best beams, depending on the rough DOA estimation, of the all sector orthogonal shift and sum beam fan, measured and shown in figure 4.17. All 8 beams have a mean HPBW of 37° . The full-360 test results are presented in figure 4.18. One signal source, at an approximate frequency of 790 KHz, experiences an RMSE accuracy of 2.2° . The RMSE accuracy is slightly worse than the all sector method, but this is reasonable considering 5 beams that contain DOA information are essentially thrown away, which is equivalent to reducing the size of the beamspace covariance matrix. Also, the initial rough sector search in this experiment used the RMS voltage SWB method for ease in implementation; however, the measured RMSE accuracy for rough sector searching proves that it could be supplemented for the SWB and yield better results.

Spatial pseudo-power spectra are shown in figure 4.19 as an example of the DOA spectrum output using fine sectors. The beamwidth narrows, decreasing the potential angular separation between signal sources, but the accuracy decreases slightly, on the

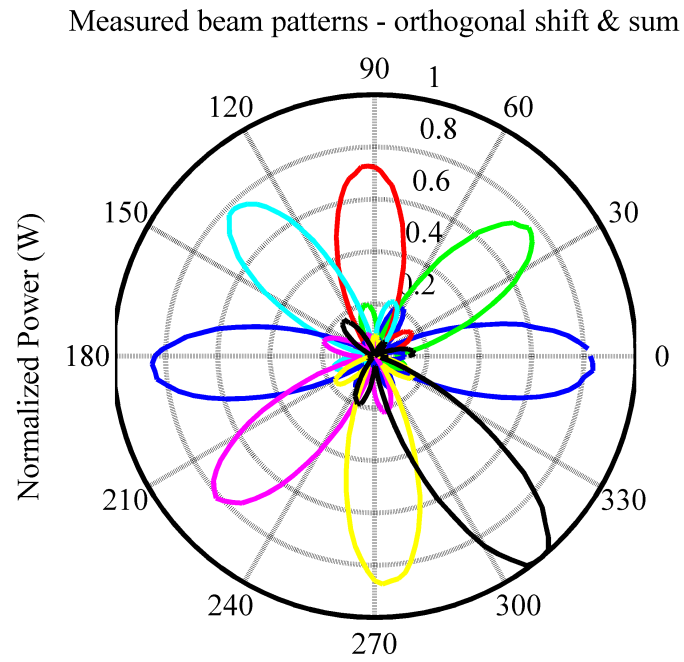


Figure 4.17: Measured orthogonal shift and sum beam patterns: 8 beam fan

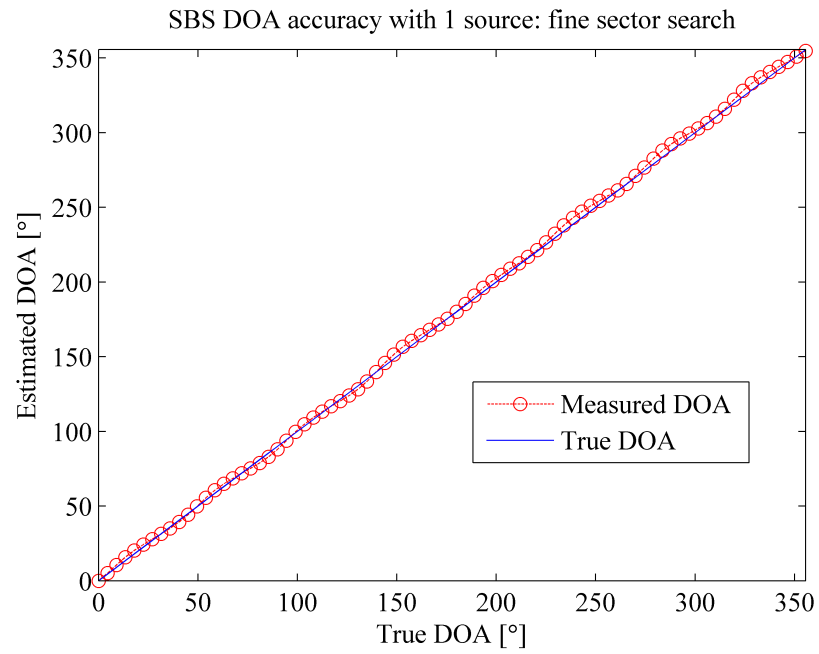


Figure 4.18: SBS accuracy using a fine sector search

order of 0.5° RMSE, compared to the all sector search. However, the acquisition time is 37.5% of the all sector search, and the computational complexity is also greatly reduced.

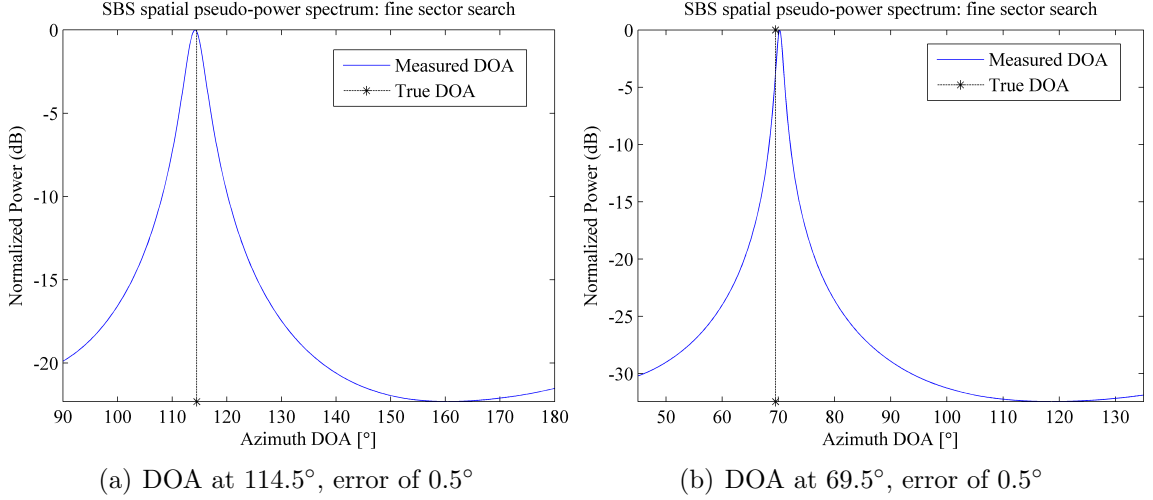


Figure 4.19: SBS spatial pseudo-power spectra for a fine sector search

4.3.7 DOA Estimation Summary

The DOA estimation test results clearly assert the superiority of SBS over SWB, and its ability to accommodate two sources, its robustness with low signal power levels, and its flexibility using rough and fine sector searches. The measured RMSE accuracy results are excellent lab results for three reasons. First, the antenna positioner still injects some small accuracy error from gear slop and the discreteness of the stepper-motor and driver. Second, because of the nearness of transmit and receive nodes in the anechoic chamber the horn antenna has a measurable aperture size in relation to the UCA, so it does not resemble an ideal point source. Last, mutual coupling has not been removed, which is a subject of future work. These reasons support an assumption of accuracy tolerance no smaller than $\pm 1^\circ$. Furthermore, the results do not quite reach theoretical sub-degree accuracy levels; however, they

do compare to similar element space lab-based results found in [4, 30]. The full-360 test results are summarized in table 4.2, and for tests with two sources, frequency is specified as spectral separation.

Method	# of sigs	Freq (KHz)	RMSE-1 [°]	RMSE-2 [°]	Res. [°]
SWB	1	NA	13.0	NA	NA
all sector	1	80	1.6	NA	NA
all sector	1	870	1.8	NA	NA
all sector	2	300	2.8	2.3	80
all sector	2	50	3.7	2.6	80
rough sector	1	560	11.5	NA	NA
fine sector	1	790	2.2	NA	NA

Table 4.2: SBS RMSE accuracy result summary

4.4 Beamforming

Basic beamspace beamforming tests were carried out within the lab environment to validate SBS and its ability to efficiently beamform once a friendly signal source DOA is found. The end result sends element space weights to the beamformer board; however, this is considered beamspace beamforming because the element space weights are found via beamspace techniques. Two beamforming techniques are evaluated, classic shift and sum beamforming and optimal Capon beamforming. As a basis of evaluation, two metrics will be presented; the measured beam pattern and the output waveform correlated with the original transmitted waveform.

4.4.1 Shift and Sum Beamforming

Two signal sources, approximately 100 KHz and 400 KHz, transmitting approximately the same power were arranged to strike the array from arrival directions of approximately 225° and 315° , respectively. The frequency spread was chosen to

easily distinguish between signal sources when correlating the signals to an ideal transmitted signal. The spatial pseudo-power spectrum, in figure 4.20, estimated the DOAs as 230° and 314° . Next, the beamspace weights were converted into element space weights and calibrated for the beamforming board using the VNA and relay test setup. The weights were then flashed to beamformer FPGA, and the patterns measured in the anechoic chamber. The beam pattern results are shown in figure 4.21.

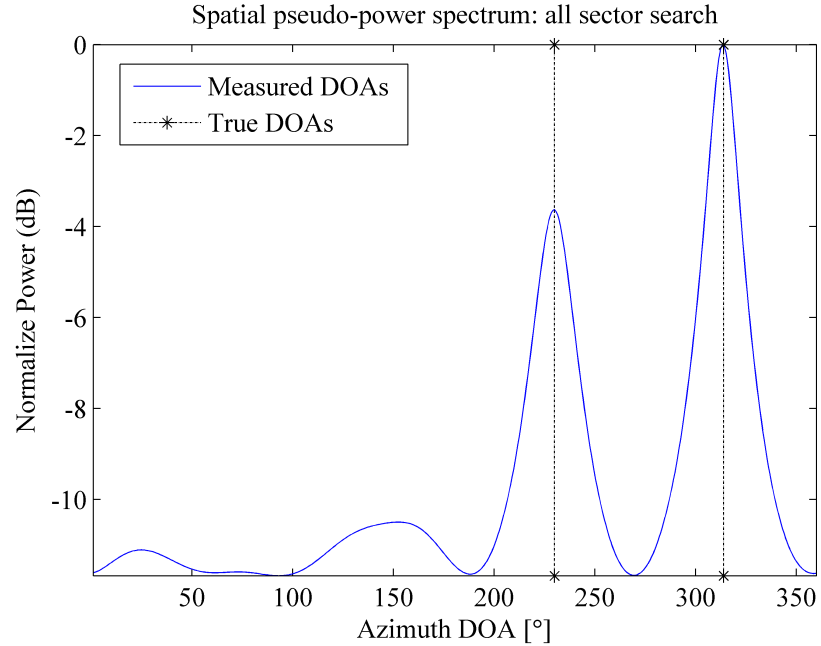


Figure 4.20: Spatial pseudo-power spectrum for 2 sources

To quantify the success of the beamforming, both output time waveforms were recorded when the shift and sum beam was activated in the direction of the corresponding signal source. The waveform results are shown in figure 4.22. The measured shift and sum waveforms, 100 KHz and 400 KHz, have correlation coefficients of 0.99 and 0.68, compared to the ideal software generated transmitted waveforms, respectively. A correlation coefficient of 1 means identical waveforms.

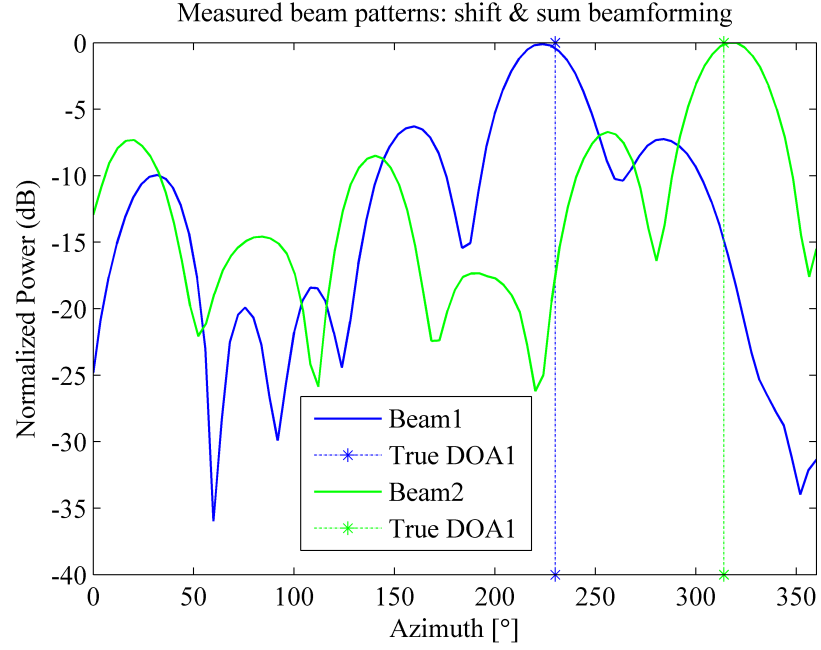


Figure 4.21: Shift and sum beamforming patterns for 2 sources

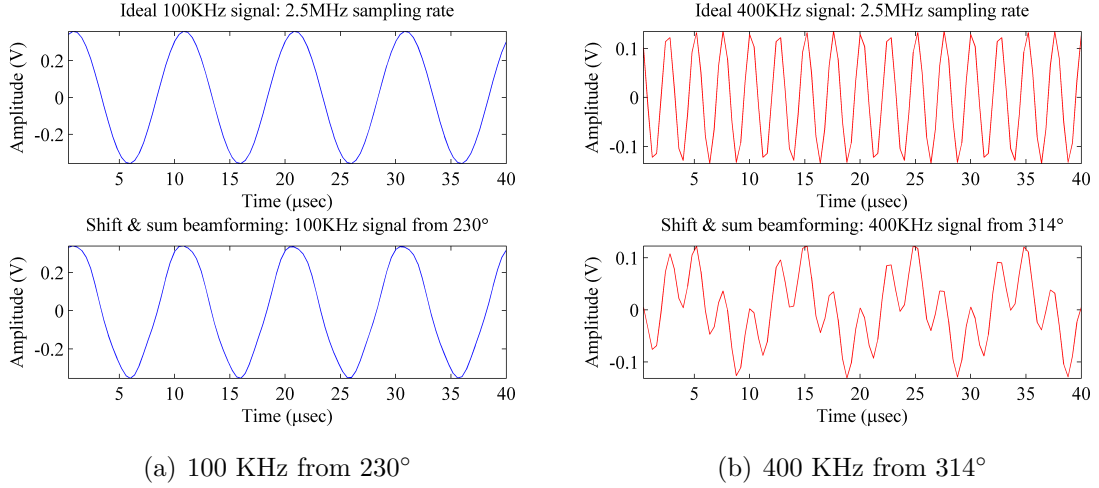


Figure 4.22: Measured and ideal output waveforms using shift and sum beamforming

4.4.2 Capon Beamforming

The same scenario from the shift and sum beamforming experiment was then reproduced using Capon beamforming. The Capon spatial pseudo-power spectrum, shown in figure 4.23, measured the DOAs as 229° and 314°, using the beamspace

Capon DOA algorithm. The calculated beamspace weights were converted into element space weights and calibrated for the beamforming board using the VNA and relay test setup. This technique produced poor beam pattern results, possibly because of the high SNR levels. In the presence of high SNR, Capon tends to place more significance on nulling the foe, i.e. interferer, than beaming toward the friend. Ideal Capon weights for the measured DOAs were then calculated using MATLAB, calibrated, and flashed to the beamformer FPGA, and the patterns were measured in the anechoic chamber. The Capon beam pattern results are shown in figure 4.24. Examining the beam patterns, notice the advantage of Capon beamforming with two signal sources; the method works to form a null at the DOA of the signal it is not beamforming toward, because it perceives the adjacent signal as interference.

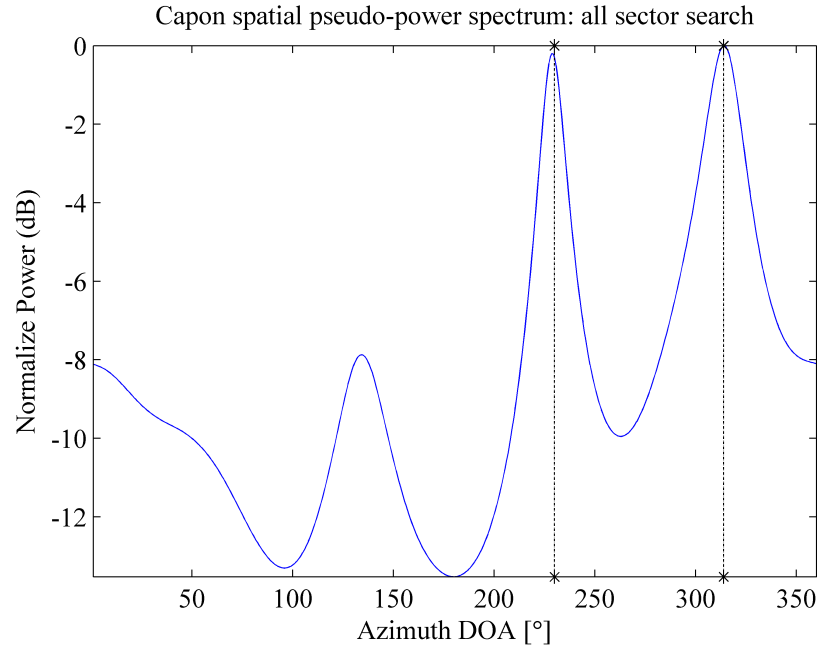


Figure 4.23: Capon spatial pseudo-power spectrum for 2 sources

Again, both output time waveforms were recorded when the proper Capon beam was activated in the direction of the corresponding signal source. The waveform

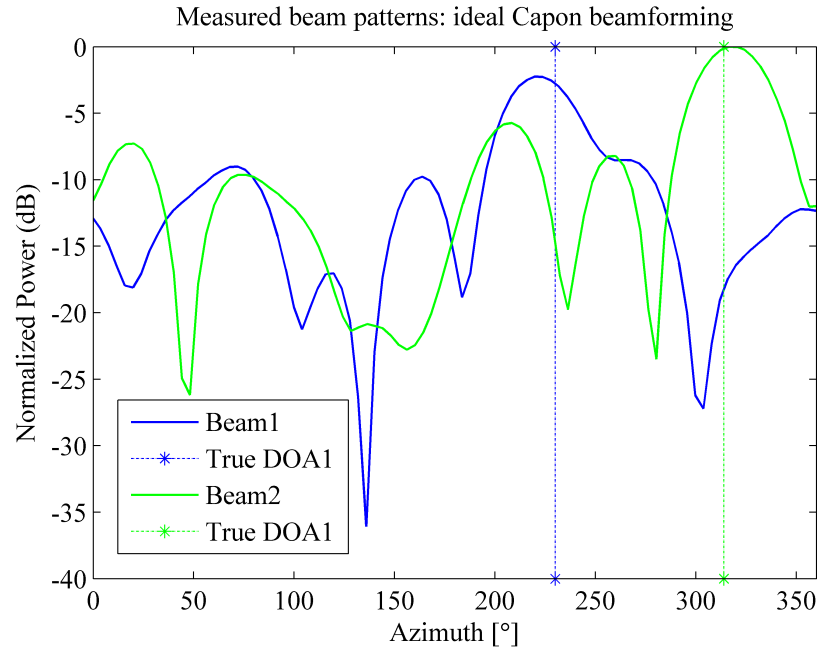


Figure 4.24: Capon beamforming patterns for 2 sources

results are shown in figure 4.25. The measured Capon waveforms, 100 KHz and 400 KHz, have correlation coefficients of 0.99 and 0.88 compared to the ideal software generated transmitted waveforms, respectively.

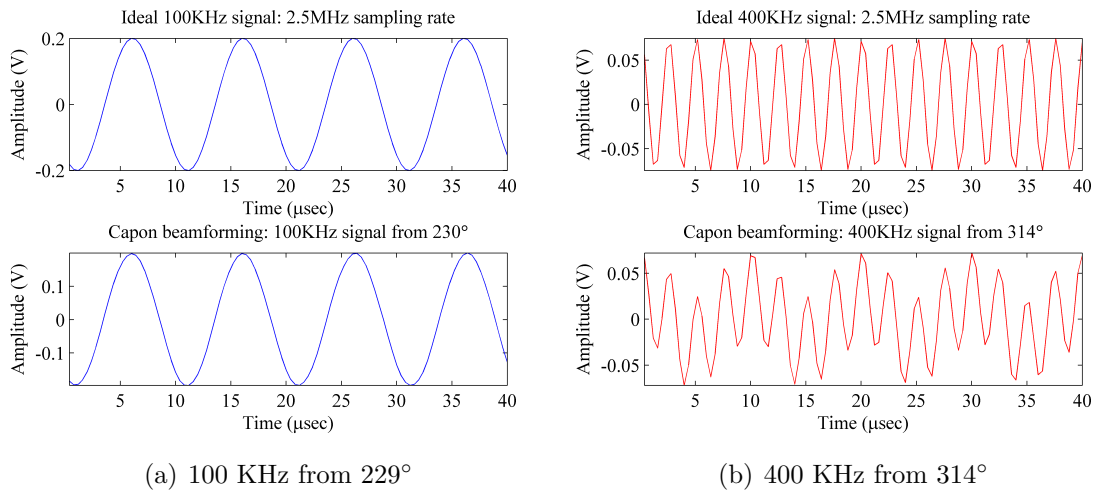


Figure 4.25: Measured and ideal output waveforms using Capon beamforming

4.4.3 Beamforming Summary

High correlation coefficients for both exercises, as well as the relative shape of the beamforming patterns points to the success of these experiments. As expected, Capon beamforming, being the statistically optimal beamforming choice, outperformed shift and sum beamforming. Moreover, these experimental results concur with the theoretical simulations in section 3.3.2. Most likely, both beamforming techniques could be implemented on the beamformer in real-time via the benefit of more extensive calibration, e.g. mutual coupling compensation. The beamforming results are summarized in table 4.3. Note, the null level is specified at the DOA of the opposite signal source.

Method	DOA [°]	Pattern Peak [°]	Null (dB)	CorrCoeff
S&S	230	224	-14.9	0.99
S&S	314	316	-17.6	0.68
Capon	229	223	-18.2	0.99
Capon	314	316	-13.7	0.88

Table 4.3: Shift and sum and Capon beamforming result summary

CHAPTER 5

FUTURE WORK

This work is a continuation of MSU's smart antenna research group, and it spurs many interesting studies for future work. One essential unknown is how the SBS data acquisition rate compares to the coherence channel length for a real-world radio propagation channel. The assumption was made in this work that the channel is stationary over the entire SBS acquisition time; however, a complete statistical coherence study needs to take place to fully characterize these parameters. Essentially, what type of fading channel will the SBS acquisition rate, $\tau \times N$, accommodate? Can the SBS technique acquire data fast enough to provide quality DOA estimations in multipath environments, e.g. Ricean channel? What is the effectiveness of SBS for wideband versus narrow band channels? These are the questions that need to be answered to fully validate the SBS technique with live communication systems.

As mentioned in section 3.4, the calibration process does not compensate for mutual coupling between antenna elements and adjacent RF channels on the beamformer board. An effort to remove the mutual coupling by adapting and updating the present element space recursive batch least squares method to beamspace is key for better SBS DOA estimation with multiple signal sources. Also, as an extension to the frequency calibration, it would be best to structure this so the system is frequency agnostic, especially when transitioning to wideband communications signals.

An update of the UCA structure and its abilities would be a good candidate for future work. The monopole elements may be upgraded to collinear dipoles, or monopoles, to effectively achieve a higher gain with the same omni-directional pattern, still being suitable for array applications. Also, investigation into elevation beam

control and steering would greatly add to the functionality, flexibility, and efficiency of a smart antenna array.

Improved tracking functionality in SBS mode would be an intriguing tangent to pursue. Basic Kalman filters have been implemented in element space for tracking purposes without the use of control input signals. A linear Kalman filter or a nonlinear extended Kalman filter given knowledge about target movements, as control signals to update the filter, while using SBS DOA estimations could be a powerful combination for node localization and tracking.

The future work embodies a fully autonomous adaptive beamforming “black box” that can interface with numerous radio standards using real-time SBS DOA estimation and beamforming capabilities, constantly optimizing beam patterns to steer toward desired users and null interference.

CHAPTER 6

CONCLUSIONS

A novel SBS smart antenna system using a single compact beamformer with UCA has been presented in this thesis. The SBS system could be adopted by conventional phased array systems currently in operation to provide DOA estimation capabilities, with accuracy on the order of $\pm 1^\circ$, partnered with beamspace shift and sum beamforming. Also, a unique dual search technique in beamspace has been proposed using rough and fine sector searches to reduce data acquisition time and computational complexity. Multiple SBS units could be linked to provide accurate RF emitter localization and tracking or efficient communication in an ad hoc networking scheme. The proposed system and techniques have been thoroughly investigated via relevant literature, simulation, and lab-based experiments. Simulations and experiments have proven that the SBS system can perform equally as well as more complex element space or conventional beamspace DOA estimation approaches. Also, the system uses a reduced set of hardware compared to its alternatives and provides multi-functionality in a single unit, e.g. DOA estimation, beamforming, and high antenna gain. The aim of this thesis to research and develop the SBS technique and provide proof of concept experimental support has been met.

REFERENCES CITED

- [1] Zhizhang Chen, Gopal Gokeda, Yiqiang Yu. *Introduction to Direction-of-Arrival Estimation*. Artech House, 2010.
- [2] B.D. Van Veen, K.M. Buckley. Beamforming: a versatile approach to spatial filtering. *ASSP Magazine, IEEE*, 5(2):4–24, April 1988.
- [3] William G. Tidd, Yufei Zhao, Yikun Huang. A compact beamspace doa estimation and beamforming communication device. *2011 IEEE Aerospace Conference*, pages 1–8, Big Sky, MT, January 2011.
- [4] Raymond J. Weber. Development of a cognitive array system. Master’s Thesis, Montana State University, April 2010.
- [5] Petre Stoica, Arye Nehorai. Comparative performance study of element-space and beam-space music estimators. *Circuits, Systems, and Signal Processing*, 10:285–292, 1991.
- [6] X.L. Xu, K.M. Buckley. Statistical performance comparison of music in element-space and beam-space. *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*, pages 2124–2127 vol.4, May 1989.
- [7] Ralph O. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation*, AP-34(3):276–280, March 1986.
- [8] J. Capon. High-resolution frequency-wavenumber spectrum analysis. *Proceedings of the IEEE*, 57(8):1408–1418, August 1969.
- [9] William G. Tidd, Raymond J. Weber, Yikun Huang. Cognitive array - a new approach. *National Radio Science Meeting*, University of Colorado at Boulder, January 2010.
- [10] William G. Tidd, Raymond J. Weber, Yikun Huang, Yufei Zhao. A rf source localization and tracking system. *2010 Military Communications Conference*, pages 267–272, San Jose, CA, November 2010.
- [11] Jerome A. Gansman, Michael D. Zoltowski, James V. Krogmeier. Multidimensional multirate doa estimation in beamspace. *IEEE Transactions on Signal Processing*, 44(11):2780–2792, November 1996.
- [12] Ahmed Khallaayoun, Andy Olson, Michael D. Panique, Yikun Huang. An adaptive smart antenna testbed for wimax radio. *IEEE Mobile WiMAX Symposium*, pages 209–213, Napa Valley, CA, July 2009.
- [13] Michael D. Zoltowski, Cherian P. Mathews. Eigenstructure techniques for 2-d angle estimation with uniform circular arrays. *IEEE Transactions on Signal Processing*, 42(9):2395–2407, September 1994.

- [14] Chao Sun, Yi xin Yang. On beampattern design for beamspace music. *Acoustical Science and Technology*, 25(1):2–8, 2004.
- [15] H. Krim, M. Viberg. Two decades of array signal processing research: the parametric approach. *Signal Processing Magazine, IEEE*, 13(4):67–94, July 1996.
- [16] Huang Yikun, William G. Tidd, Richard S. Wolff, Andy Olson. A compact smart antenna for wimax radio. *Mobile WiMAX Symposium, 2009. MWS '09. IEEE*, pages 169 –173, Napa Valley, CA, July 2009.
- [17] Harry L. Van Trees. *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*. Wiley-Interscience, 2002.
- [18] Nicholas J. Roseveare, Mahmood R. Azimi-Sadjadi. Capon beamspace beamforming for distributed acoustic arrays. *Proceedings of SPIE, the International Society for Optical Engineering*, 6562:65620K–1 – 65620K–8, April 2007.
- [19] V. Cizek. Discrete hilbert transform. *Audio and Electroacoustics, IEEE Transactions on*, 18(4):340 – 343, December 1970.
- [20] Mathias Johansson. The hilbert transform. Master’s Thesis, Växjö University, 1999.
- [21] Cherian P. Mathews. *Signal Subspace Techniques for Source Localization with Circular Sensor Arrays*. Doctoral Dissertation, Purdue University, January 1994.
- [22] William G. Tidd, Yufei Zhao, Yikun Huang. Sequential beamspace doa estimation. *Paper Accepted by 2011 IEEE International Symposium on Antennas and Propagation*, pages 1–4, Spokane, WA, July 2011.
- [23] B.D. Rao, K.V.S. Hari. Performance analysis of root-music. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 37(12):1939 –1949, December 1989.
- [24] John G. Proakis, Dimitris G. Manolakis. *Digital Signal Processing*. Practice Hall, edition fourth, 2007.
- [25] C.-I.C. Nilsen, I. Hafizovic. Beamspace adaptive beamforming for ultrasound imaging. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, 56(10):2187 –2197, October 2009.
- [26] William G. Tidd, Yikun Huang. An experimental evaluation of a sequential beamspace smart antenna system. *Paper Submitted to 2011 Military Communications Conference*, pages 1–6, Baltimore, MD, November 2011.
- [27] Dwayne A. Folden. Automated antenna test and characterization facility. Master’s Thesis, Montana State University, 2007.

- [28] Michael David Panique. Design and evaluation of test bed software for a smart antenna system supporting wireless communications in rural areas. Master's Thesis, Montana State University, 2008.
- [29] Raymond J. Weber, Yikun Huang. A wideband circular array for frequency and 2d angle estimation. *2010 IEEE Aerospace Conference*, Big Sky, MT, March 2010.
- [30] Ahmed Khallaayoun. *High Resolution Direction of Arrival Estimation Analysis and Implementation in a Smart Antenna System*. Doctoral Dissertation, Montana State University, 2010.

APPENDICES

APPENDIX A

VHDL CODE

A.1 Beamformer Firmware

```

1  _____
2  — Company:
3  — Engineer: William Tidd
4  —
5  — Create Date:      12:43:29 08/31/2010
6  — Design Name:
7  — Module Name:      beam_switch – Behavioral
8  — Project Name:
9  — Target Devices:
10 — Tool versions:
11 — Description:
12 —
13 — Dependencies:
14 —
15 — Revision:
16 — Revision 0.01 – File Created
17 — Additional Comments:
18 —
19 — Last modified 4-20-2011
20 —
21  _____
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24 use IEEE.STD_LOGIC_ARITH.ALL;
25 use IEEE.STD_LOGIC_UNSIGNED.ALL;
26
27 — Uncomment the following library declaration if using
28 — arithmetic functions with Signed or Unsigned values

```

```

29  —use IEEE.NUMERICSTD.ALL;
30
31  — Uncomment the following library declaration if instantiating
32  — any Xilinx primitives in this code.
33  —library UNISIM;
34  —use UNISIM.VComponents.all;
35
36  entity beam_switch is
37      Port ( clk : in  STD_LOGIC;
38            data : in  STD_LOGIC_VECTOR (5 downto 0);
39            flag_out_pin : out  STD_LOGIC;
40            control_pins : out  STD_LOGIC_VECTOR (0 to 95));
41  end beam_switch;
42
43  architecture Behavioral of beam_switch is
44
45  —signals
46  signal counter : std_logic_vector (19 downto 0);
47  signal search : std_logic;
48  signal mode : std_logic;
49  signal cycles : std_logic;
50  signal flag : std_logic;
51  signal beam_select : std_logic_vector (3 downto 0);
52  signal beam_count : std_logic_vector (3 downto 0);
53  signal control_data : std_logic_vector (0 to 95);
54  signal beamf1 : std_logic_vector (0 to 95);
55  signal beamf2 : std_logic_vector (0 to 95);
56  signal beamf3 : std_logic_vector (0 to 95);
57  signal beamf4 : std_logic_vector (0 to 95);
58  signal beamf5 : std_logic_vector (0 to 95);
59  signal beamf6 : std_logic_vector (0 to 95);

```

```

60 signal beamf7 : std_logic_vector (0 to 95);
61 signal beamf8 : std_logic_vector (0 to 95);
62 signal beamr1 : std_logic_vector (0 to 95);
63 signal beamr2 : std_logic_vector (0 to 95);
64 signal beamr3 : std_logic_vector (0 to 95);
65 signal beamr4 : std_logic_vector (0 to 95);
66 signal omni : std_logic_vector (0 to 95);
67 signal default : std_logic_vector (0 to 95);
68
69 —begin behavioral
70 begin
71
72 —assign data
73 search <= data(5);           —search selection for rough or fine search
74 mode <= data(4);           —mode selection (manual beam selection
                             or auto beam switch)
75 beam_select <= data(3 downto 0); —manual beam selections (up to 16
                             beams)
76
77 —fine search beams (orthogonal weights)(vna calibration 8/20/2010)
78 beamf1 <= x"F43DEDF53D8BF10D2AF40D08"; —beam 1 (0/360 deg)
79 beamf2 <= x"D63F59CFFF69F1CF29CC2F14"; —beam 2 (45 deg)
80 beamf3 <= x"F68D33F66D11F36D30F7AD8E"; —beam 3 (90 deg)
81 beamf4 <= x"D23F39CC0F76F1CF0ACC1F08"; —beam 4 (135 deg)
82 beamf5 <= x"F42D98F6BE35F28D57F7FDF5"; —beam 5 (180 deg)
83 beamf6 <= x"DDBF64DB7F4ECD5F4DD38F20"; —beam 6 (225 deg)
84 beamf7 <= x"F54D4BF50DE7F7AD07F7DCA6"; —beam 7 (270 deg)
85 beamf8 <= x"D63F4CCC0F4AD1DF76CC0F76"; —beam 8 (315 deg)
86 omni <= x"F74F74F72F72F6FF32F73F32"; —omni-directional beam
87 default <= x"FC0FC0FC0FC0FC0FC0FC0FC0"; —default — atten all 1s, phase
                             shifters all 0s

```

```

88
89  —rough search beams (hann weights, 3rd order)(vna calibration
    4/19/2011)
90  beamr1  <= x"F2EBB6EFAF7CE78BF5EAAF6A"; —beam 1 (0/360 deg)
91  beamr2  <= x"EABF6BE28B31E34F78E77AF1"; —beam 2 (90 deg)
92  beamr3  <= x"EB9B33E66F68E24B2FE76F76"; —beam 3 (180 deg)
93  beamr4  <= x"E0FF10E8CB88E7AF7CF3DAC6"; —beam 4 (270 deg)
94
95  —begin process
96  process(clk, search, mode, beam_select)
97
98  begin
99
100  if (clk'event and clk = '1') then  —do everything on every rising edge
    of each clock cycle
101
102    — mode has to stay high longer than sweep time of all beams or data
    will be truncated
103    if (mode = '1' and cycles = '0') then  —switching/sampling mode (
    SBS mode)
104        counter <= counter + 1;  —increase counter by 1
105
106        case beam_count is
107            when "0000" => —when beam_count is "0000" use large counter
    value for omni freq detection
108
109            case counter is
110                when "11000011010100000000" =>  —40,000 samples
    /beam
111                beam_count <= beam_count + 1;  —increase
    beam count by 1

```

```

112         counter <= "00000000000000000000";  --reset
           counter
113     when others =>
114     end case;
115
116     when others => --when beam_count is others use small counter
           value for beamspace data
117
118     case counter is
119     when "00001001110001000000" =>          --2,000 samples/
           beam
120         beam_count <= beam_count + 1;          --increase
           beam count by 1
121         counter <= "00000000000000000000";  --reset
           counter
122
123         if search = '0' then  --inverted on TX/
           RX_INVERT (search = '1'), fine search
124
125         case beam_count is
126         when "1000" =>          --sweep through 9
           beams (including omni for freq
           detection)
127         cycles <= '1';  --set cycles high
           for single sweep
128         flag <= '0';  --turn off flag
           after
129         when others =>
130         end case;
131

```

```

132         elsif search = '1' then —inverted on TX/
           RX_INVERT (search = '0'), rough search
133
134         case beam_count is
135             when "0100" =>           —sweep through 5
               beams (including omni for freq
               detection)
136             cycles <= '1'; —set cycles high
               for a single sweep
137             flag <= '0'; —turn off flag
138             when others =>
139                 end case;
140
141         end if;
142
143         when others =>
144             end case;
145
146     end case;
147
148     if search = '0' then —inverted on TX/RX_INVERT (search = '1'),
       fine search
149
150     case beam_count is —fine search with orthogonal co-phasal
       weights
151         when "0000" => control_data <= omni; flag <= '1'; —
               turn on flag to start sampling (hardware trigger)
152         when "0001" => control_data <= beamf1;
153         when "0010" => control_data <= beamf2;
154         when "0011" => control_data <= beamf3;
155         when "0100" => control_data <= beamf4;

```

```

156         when "0101" => control_data <= beamf5;
157         when "0110" => control_data <= beamf6;
158         when "0111" => control_data <= beamf7;
159         when "1000" => control_data <= beamf8;
160         when others => control_data <= omni; —default to omni
161     end case;
162
163     elsif search = '1' then —inverted on TX/RX_INVERT (search =
        '0'), rough search
164
165         case beam_count is —rough search with window weights
166             when "0000" => control_data <= omni; flag <= '1'; —
                turn on flag to start sampling (hardware trigger)
167             when "0001" => control_data <= beamr1;
168             when "0010" => control_data <= beamr2;
169             when "0011" => control_data <= beamr3;
170             when "0100" => control_data <= beamr4;
171             when others => control_data <= omni; —default to omni
172         end case;
173
174     end if;
175
176     elsif mode = '0' then —manual beam select mode
177         —reset all internal variables
178         counter <= "00000000000000000000";
179         beam_count <= "0000";
180         cycles <= '0';
181         flag <= '0';
182
183     if search = '0' then —inverted on net: TX/RX_INVERT (search =
        '1')

```

```

184
185         case beam_select is  --orthogonal co-phasal weights
186             when "0000" => control_data <= beamf1;
187             when "0001" => control_data <= beamf2;
188             when "0010" => control_data <= beamf3;
189             when "0011" => control_data <= beamf4;
190             when "0100" => control_data <= beamf5;
191             when "0101" => control_data <= beamf6;
192             when "0110" => control_data <= beamf7;
193             when "0111" => control_data <= beamf8;
194             when "1000" => control_data <= omni;
195             when others => control_data <= default;  --default to
                all 1s and 0s
196         end case;
197
198     elsif search = '1' then --inverted on net: TX/RXINVERT (search
        = '0')
199
200         case beam_select is --window weights
201             when "0000" => control_data <= beamr1;
202             when "0001" => control_data <= beamr2;
203             when "0010" => control_data <= beamr3;
204             when "0011" => control_data <= beamr4;
205             when others => control_data <= default;  --default to
                all 1s and 0s
206         end case;
207
208     end if;
209
210 end if;
211

```

```
212 end if;
213
214 end process;
215
216 —send phase/magnitude data and flag bits out
217 control_pins <= control_data;
218 flag_out_pin <= flag;
219
220 end Behavioral;
```

APPENDIX B

MATLAB CODE

B.1 Weight Conversion for Beamformer

```

1  % Weight2decimal – Beam weight conversion from complex decimal to 6-bit
   % decimal integer,
2  % binary, and hexadecimal
3  % [wts_all ,wts_mag, wts_ang, bin_wts_all, hex_wts_all] = ...
4  % weight2decimal(weights, filename,
   % path)
5  %
6  % Parameters (in order):
7  % weights: ideal beamspace excitation weights in floating point
8  % filename: name of output as a string
9  % path: directory path to output file as a string
10 %
11 % Returns (in order):
12 % wts_all: interleaved magnitude and angle weights
13 % wts_mag: magnitude weights
14 % wts_ang: angle weights
15 % bin_wts_all: all interleaved binary weights
16 % hex_wts_all: all interleaved hexadecimal weights
17 % csv file: interleaves weights and writes to csv file for
   % beamformer use
18 %
19 % Note: Weights must be entered in column form. If path is left empty
   % file
20 % outputs to operating directory. If filename is left empty no output
   % file
21 % is created.
22 %
23 % Revision 1/26/2011 by William Tidd, aka output_decimal

```

```

24 %
25 function [ wts_all , wts_mag, wts_ang, bin_wts_all, hex_wts_all ] = ...
26                                     weight2decimal( weights , filename ,
27                                                     path)
28
29 % Number of antenna elements and beams
30
31 [m, beams] = size( weights );
32
33 % Magnitude to decimal
34
35 wts_max = max(max(abs( weights )));
36 wts_mag = round(20*log10(abs( weights )/wts_max)*2 + 63);
37
38 % Angle to decimal (Phase Shifters 0 to -360 degrees)
39
40 wts_ang = abs(round(angle( weights )/pi*32 - 32));
41 wts_ang( wts_ang > 63 ) = wts_ang( wts_ang > 63 ) - 64;
42
43 % Interleave magnitude and angle weights for csv file and beamformer
44
45 wts_all = zeros( beams, 2*m );
46
47 for k = 1:beams
48     n = 1;
49     for i = 1:2:2*m
50         wts_all(k, i) = wts_mag(n, k);
51         wts_all(k, i+1) = wts_ang(n, k);
52         n = n + 1;
53     end
54 end
55
56 % Check existance of path
57
58 if ~exist( 'path' , 'var' )
59     path = pwd;
60 end

```

```

54
55 % Check existance of file name
56 if exist('filename','var')
57     csvwrite([path '\ ' filename '.csv'], wts_all);    % write to CSV File
58 end
59
60 % Decimal integer to binary and hex weights
61 ICs = 2*m;
62 bit_size = 6;
63 bin_wts_all = cell(beams,1);
64 hex_wts_all = cell(beams,1);
65 bin_wts = [];
66
67 for k = 1:beams
68     % Decimal integer to binary
69     for i = 1:ICs
70         bin = dec2bin(wts_all(k,i),bit_size);
71         bin_wts = [bin_wts bin];
72     end
73
74     % Decimal integer to hex
75     hex_wts = [dec2hex(bin2dec(bin_wts(1:48))) dec2hex(bin2dec(bin_wts
        (49:96)))]);
76
77     % Weights for all beams
78     bin_wts_all(k) = {bin_wts}; bin_wts = [];
79     hex_wts_all(k) = {hex_wts}; hex_wts = [];
80 end

```

B.2 Sampling Phase Error

```

1  % SamplingError – function that randomizes beamspace samples to simulate
2  % random beam switching/sampling which manifests as sampling phase error
3  % [y_t, counter] = SamplingError(y_t, Samp)
4  %
5  % Parameter :
6  %     y_sync: synchronized beamspace analytic signal
7  %     Samp: desired sample length for unsynchronized y_sync
8  %     Samptype: randomized, 'random', or fixed, 'fixed' as a character
      string
9  %     CommRate: commutation (sampling/switching) rate per beam in sample
      length
10 %
11 % Returns (in order):
12 %     y_unsync: unsynchronzied beamspace signal (sampling phase error)
13 %     counter: sample counter in time (beam commutation rate)
14 %
15 function [y_unsync, counter] = SamplingError(y_sync, Samp, Samptype,
      CommRate)
16
17 % Input arguments
18 if nargin < 4
19     error('Must have 4 inputs');
20 end
21
22 % Commutation rate vs. sample length
23 if Samp > CommRate
24     error('Commutation rate must be equal or larger than desired sample
      length');

```

```

25 end
26
27 % Beams
28 beams = min(size(y_sync));
29
30 % Random or fixed sample error
31 switch lower(Samptype)
32     case 'random'
33
34         % Random switching time up to the length of CommRate
35         inc = [1 randi(CommRate,1,beams - 1)];
36         start = @(i,finish) inc(i) + finish;
37     case 'fixed'
38
39         % Fixed sampling/switching at the length of CommRate
40         inc = [0 (1:beams - 1)*CommRate] + 1;
41         start = @(i,finish) inc(i);
42     otherwise
43         error('Sample error type not known, random or fixed');
44 end
45
46
47 % Initialize
48 y_unsync = zeros(beams,Samp);
49 Start = zeros(1,beams);
50 Finish = 0;
51
52 % Create indexes and unsynchronized beamspace data vectors
53 for i = 1:beams;
54
55     % Indexes

```

```
56     Start(i) = start(i, Finish);
57     Finish = Samp - 1 + Start(i);
58
59     % Assign and set length of unsynchronized data
60     y_unsync(i, 1:Samp) = y_sync(i, Start(i):Finish);
61
62     % FPGA Counter Output
63     counter = Start;
64 end
```

B.3 Time Delay Correction Filter

```

1  % Wilbert - Hilbert using FFT with time delay correction filter
2  % [y_fixed, tau_delay] = wilbert_v2(y_t, Ts, counter, Op_freq)
3  %
4  % Parameter (others are set internally):
5  %     y_t: beamspace analytic signal
6  %     Ts: sampling period
7  %     counter: number of samples waveform is delayed
8  %     Op_freq: operation frequency estimated by MVDR algorithm
9  %
10 % Returns (in order):
11 %     y_fixed: analytic beamspace signal with absolute time delay
12 %     tau_delay: time delay of each beam vector in terms of sampling rate
13 %
14 function [y_fixed, tau_delay] = wilbert_v2(y_t, Ts, counter, Op_freq)
15
16 % Transpose to column vector if necessary
17 if size(y_t,1) < size(y_t,2)
18     y_t = y_t.';
19 end
20 [rows, cols] = size(y_t);           % Columns and rows
21
22 % FFT
23 P = nextpow2(rows);                 % Next power of 2 samples for optimum
                                     % FFT time
24 nfft = 2^P;                         % Pad FFT with zeros
25 yfft = fft(y_t, nfft);              % FFT on columns
26
27 % Create h(hilbert) vector

```

```

28  h = zeros(nfft,1);           % Zeros i = r/2+2 to r
29  if 2*fix(nfft/2) == nfft     % Even
30      h([1 nfft/2+1]) = 1;     % Ones i = 1 and r/2+1
31      h(2:nfft/2) = 2;         % Twos i = 2 to r/2
32  else                          % Odd
33      h(1) = 1;                % Ones i = 1
34      h(2:(nfft+1)/2) = 2;     % Twos i = 2 to (r+1)/2
35  end
36  h = h(:,ones(1,size(yfft,2))); % Copy hilbert vector across all columns
37
38  % Element Multiply
39  hyfft = yfft.*h;
40
41  % Time delay in terms of sampling rate
42  tau_delay = counter*Ts;
43
44  % Apply delay correction to each column with best frequency
45  hy_best = zeros(nfft,cols);
46  for i = 1:cols
47      hy_best(:,i) = hyfft(:,i).*exp(-1i*2*pi*Op_freq*tau_delay(i));
48  end
49
50  % IFFT and transpose to row vectors
51  y_fixed = ifft(hy_best).';

```

B.4 “full-360” Automated Positioner

```

1  % Automated_Positioner – script to automate and run "full-360" tests for
2  % Beamspace DOA RMSE accuracy collection
3  %% Setup DAQs
4  hw = daqhwinfo('nidaq');
5
6  %%Stepper Motor DAQ PCI-6013
7  % Find device PCI-6133 for analog acquisition
8  index = strcmp(hw.BoardNames,'PCI-6013');
9  device_motor = cell2mat(hw.InstalledBoardIds(index));
10
11 % Create a digital output object, set channel 0:4, port0, and attributes
12 dio_motor = digitalio('nidaq',device_motor);    % Create digital line
           object 2
13 addline(dio_motor,0:7,0,'out');                % Addline
14 dio_motor.line.LineName(1) = 'ON_OFF';          % Name channel
15 dio_motor.line.LineName(2) = 'Direction';       % Name channel
16 dio_motor.line.LineName(3) = 'MS1';             % Name channel
17 dio_motor.line.LineName(4) = 'MS2';             % Name channel
18 dio_motor.line.LineName(5) = 'CLOCK';           % Name channel
19 putvalue(dio_motor,0);                          % Initial output low
20
21 %%Analog Input and Digital Output DAQ PCI-6133
22 % Find device PCI-6133 for analog acquisition
23 index = strcmp(hw.BoardNames,'PCI-6133');
24 device_anal = cell2mat(hw.InstalledBoardIds(index));
25
26 % Create an analog input object, set channel 0, and attributes
27 ai = analoginput('nidaq',device_anal);          % Create input analog object

```

```

28  addchannel(ai,0); % Analog Channel
29  set(ai,'SampleRate',2.5e6); % Sample Rate
30  set(ai,'TriggerType','HwDigital') % Trigger Type
31  set(ai,'HwDigitalTriggerSource','PFI1'); % Trigger Channel
32  set(ai,'TriggerCondition','PositiveEdge'); % Trigger Condition
33  set(ai,'TriggerConditionValue',0.5) % Trigger Condition value
    (0.5V)
34  set(ai,'SamplesPerTrigger',samples); % Samples to acquire (
    default)
35  set(ai.channel,'InputRange',[-1.25 1.25]); % Input Range
36
37  % Create a digital output object, set channel 0, port0, and attributes
    with PCI-6133
38  dio1 = digitalio('nidaq',device_anal); % Create digital line object
    1
39  addline(dio1,0,0,'out'); % Addline
40  dio1.line.LineName = 'SweepTrigger'; % Name channel
41  putvalue(dio1,0); % Initial output low (no
    sweep)
42
43  %%Digital Output DAQ USB-6501
44  % Find device USB-6501 for digital generation
45  index = strcmp(hw.BoardNames,'USB-6501');
46  device_dig = cell2mat(hw.InstalledBoardIds(index));
47
48  % Create a digital output object, set channel 0:4, port0, and attributes
49  dio2 = digitalio('nidaq',device_dig); % Create digital line object
    2
50  addline(dio2,4,0,'out'); % Addline
51  dio2.line.LineName(1) = 'SectorType'; % Name channel
52  putvalue(dio2,0); % Initial output low

```

```

53 if handles.sectors == 9
54     putvalue(dio2.Line(1),1);           % Default fine search
55 elseif handles.sectors == 5
56     putvalue(dio2.Line(1),0);         % Default rough search
57 end
58
59 %% Stepper Motor Parameters
60
61 % Allocate
62 step_array = [];
63
64 % Inputs
65 speed = 100;           % motor speed in hertz
66 step_dist = 4.5;       % step distance per measurement
67 ON_OFF = 1;           % stepper on = 1, off = 0
68 mov_dir = 0;          % CCW = 0, CW = 1
69
70 % Step Type: Auto step - Eighth step
71 m1 = 1;
72 m2 = 1;
73 step_size = 0.1125;
74
75 % Number of steps for motor to turn
76 num_steps = ceil(step_dist/step_size);
77
78 % Inputs
79 clock_high = (2^4)*1 + (2^3)*m2 + (2^2)*m1 + (2^1)*mov_dir + (2^0)*
    ON_OFF;
80 clock_low = (2^4)*0 + (2^3)*m2 + (2^2)*m1 + (2^1)*mov_dir + (2^0)*
    ON_OFF;
81

```

```

82  if ON_OFF == 0
83      num_steps = 1;
84  end
85
86  for clock_loop = 1:num_steps
87      step_array = [step_array clock_high clock_low];
88  end
89
90  %% Capture data
91  total_dist = 360; % Total distance in degrees
92  total_moves = ceil(total_dist/step_dist); % Total measurements
93
94  for i = 1:total_moves
95
96      % Move motor
97      for steps = 1:2*num_steps
98          putvalue(dio_motor, step_array(steps));
99          pause(1/speed);
100  end
101
102  % Take measurement
103  bsdata{i} = take_measurement(ai, dio1, handles);
104
105  % Analyze
106  handles.bsdata = bsdata{i};
107  [bestdoa{i}, Op_freqs(i), x{i}, y{i}, wes_cap{i}, wes_ss{i}] =
      LV_Beamspace_DOA(handles);
108  bestdoas(i) = bestdoa{i}{1};
109
110  % Adjust according to direction
111  if mov_dir == 1

```

```

112     meas_ang(i) = bestdoas(i) - bestdoas(1) + 360; % measured angle
113     true_ang(i) = num_steps*step_size*(i - 1); % true angle
114 else
115     meas_ang(i) = bestdoas(i) - bestdoas(1); % measured angle
116     true_ang(i) = -num_steps*step_size*(i - 1); % true angle
117 end
118
119 % Difference angle
120 diff_ang(i) = meas_ang(i) - true_ang(i);
121 diff_ang(diff_ang < -180) = diff_ang(diff_ang < -180) + 360;
122 diff_ang(diff_ang > 180) = diff_ang(diff_ang > 180) - 360;
123
124 % RMSE
125 std_diff_ang(i) = sqrt(mean(diff_ang(1:i).^2));
126 disp(['RMSE DOA Accuracy is = ', num2str(std_diff_ang(i)) ' degrees'
127     ]);
128
129 % New measured angle
129 new_meas_ang(i) = diff_ang(i) + true_ang(i);
130
131 % Floor minimums
132 floor_mins(i) = min(y{i}{1});
133
134 % Plots
135 h1 = figure(1);
136 set(h1,'position',[560 420 560 420]);
137 plot(1:i,new_meas_ang,'ro--',1:i,true_ang,'b');
138 legend('Measured DOA','True DOA','location','best');
139
140 h2 = figure(2);
141 set(h2,'position',[1 1 560 420]);

```

```

142     plot(x{i}{1},y{i}{1});
143     title('Measured DOA Spatial Psuedo-Power Spectrum');
144
145     h3 = figure(3);
146     set(h3,'position',[1 420 560 420]);
147     plot(floor_mins);
148     title('Measured DOA Floor');
149 end
150
151 % Save file
152 save full360data.mat
153
154 % Turn off motor
155 putvalue(dio_motor,0);           % Initial output low

```

B.5 Beamspace Parsing and Processing

```

1  % Function to process beamspace data in SBS_v1 GUI
2  % LV_BEAMSPACE_DOA
3  %
4  % [bestdoas, Op_freqs, x, y, wes_cap, wes_ss] = LV_Beamspace_DOA(handles
   )
5  %
6  function [bestdoas, Op_freqs, x, y, wes_cap, wes_ss] = LV_Beamspace_DOA(
   handles)
7  %% Extract data
8  inCh = handles.bsdata;
9  beams = handles.sectors - 1;
10 CR = handles.CommRate;
11 clks_omni = handles.clks_omni;
12 analyzemode = handles.analyzemode;
13 plotswitch = handles.plotswitch;
14
15 %% Constants
16 c = 2.998e8;           % speed of light
17 f = 5.8e9;             % center frequency of operation
18 lamda = c/f;          % wavelength
19 rho = .02515;          % radius for 2" UCA
20 beta = 2*pi/lamda;     % wavenumber
21 betaRho = beta*rho;    % antenna size
22 xi = betaRho*sin(pi/2); % elevation dependence
23 m = 8;                 % antenna elements
24 Ts = 1/2.5e6;          % sampling period
25
26 %% Generate Orthogonal Cophasal Weights(in Columns)

```

```

27  if beams == 8
28      angVal = linspace(0,360 - 360/beams,beams);
29      bno = zeros(m,beams);
30      for angInd = 1:beams;
31          bno(:,angInd) = exp(1i*xi*cos(angVal(angInd)*pi/180 - 2*pi/m*(1:
              m)));
32      end
33      wts = bno*(bno'*bno)^(-1/2);
34  elseif beams == 4
35      angVal = linspace(0,360 - 360/beams,beams);
36      load rough_wts
37      wts = hann_ideal;
38  end
39
40  %% Notch fiter - 16th order bandstop filter (cancels receiver noise at
      200 KHz)
41  [b,a] = butter(8,[.175/1.25 .225/1.25],'stop');
42  inCh = filter(b,a,inCh);
43
44  %% Organize Samples
45  clks = CR;                                % clock cycles (samples) per beam
46  samps_omni = 0.9*clks_omni;                % omni-directional clock cycles (samples
      ), 90% of clks_omni
47  samps = 0.5*clks;                          % beam samples
48
49  % Eliminate first few distorted samples
50  elim_samps = (clks_omni - samps_omni)/2;    % eliminated samples, 5% of
      clks_omni
51  inCh(1:elim_samps) = [];                    % eliminate samples
52
53  % Sample indexes

```

```

54 start = samps_omni + elim_samps + (clks - samps)/2;
55 s = zeros(1,beams);
56 for i = 0:beams-1
57     s(i+1) = 1 + start + i*clks;
58 end
59
60 % Counter to match fixed commutation rate
61 counter = s - start;
62
63 % Compose beamspace data and remove DC component
64 a = zeros(samps,beams);
65 for i = 1:beams
66     a(:,i) = inCh(s(i):s(i) + samps - 1);
67     a(:,i) = a(:,i) - mean(a(:,i));
68 end
69
70 %% Choose operation mode
71 switch lower(analyzemode)
72     case 'allsectors'
73         space = [1 360]*pi/180;
74
75     case 'rmsvsectors'
76         % RMSV Rough Sector
77         RMSV = sqrt(mean(a.^2));
78         [~,rmsv_ind] = max(RMSV);
79
80         % Calculate fine Sector
81         fine_sector = [rmsv_ind - 1 rmsv_ind rmsv_ind + 1];
82         fine_sector(fine_sector > beams) = fine_sector(fine_sector >
            beams) - beams;

```

```

83         fine_sector(fine_sector < 1) = fine_sector(fine_sector < 1) +
            beams;
84
85         % Span and space
86         span = angVal(fine_sector);
87         space = unwrap(angle(exp(1i*span*pi/180)));
88
89         case 'switchedbeam'
90             % RMSV Rough Sector
91             RMSV = sqrt(mean(a.^2));
92             [~,rmsv_ind] = max(RMSV);
93             bestdoas = angVal(rmsv_ind);
94
95         otherwise
96             error('Search type not known');
97     end
98
99     %% Capon frequency detect
100    x_t = inCh(1:samps_omni);    % select samples from omni-beam
101    x_t = x_t - mean(x_t);      % subtract DC component
102    [Op_freqs, signals, xfreq, yfreq] = MVDRf(x_t, samps_omni, Ts);
103
104    % If switched beam mode, end after frequency detection
105    if strcmp(analyzemode, 'switchedbeam');
106        x = 0; y = 0; wes_cap = 0;
107        return;
108    end
109
110    %% Empirical weight calibration
111    if beams == 8
112        load meas_wts_vna_4-14-2011;    % Load latest calibration file

```

```

113     WIS = norm_cal_wts;           % Assign empirical weights
114 elseif beams == 4
115     load meas_hamming_hann_wts_vna_4-19-2011;
116     WIS = cal_measure_weights(:,5:8);
117 end
118 y_cal = wts'*(WIS.')\((a.')).'; % Remove empirical weights
119
120 % Data, weights, and counter for fine sector search
121 if strcmp(analyzemode,'rmsvsectors')
122     y_cal = y_cal(fine_sector,:); % Certain beamspace rows
123     wts = wts(:,fine_sector);    % Certain weight columns
124     counter = counter(fine_sector); % Certain counter values
125 end
126
127 %% Wilbert - Delay correction
128 Op_freqs = sort(Op_freqs,'ascend');
129 for i = 1:signals
130     y_fixed{i} = wilbert_v2(y_cal,Ts,counter,Op_freqs(i));
131 end
132
133 %% DOAs
134 for i = 1:signals
135     [x{i},y{i},doas,bestwts,Rcov] = DOAs(y_fixed{i},wts,m,xi,signals,
        space,'rbs_music');
136     bestdoas{i} = round(doas*10)/10;
137 end
138
139 %% Capon beamforming
140 for i = 1:size(bestwts,2)
141     wbs_capon(:,i) = (inv(Rcov)*bestwts(:,i))/((bestwts(:,i))'*inv(Rcov)*
        bestwts(:,i));

```

```

142     wes_cap(:, i) = wts*wbs_cap(:, i);
143     wes_ss(:, i) = wts*bestwts(:, i);
144 end
145
146 %% Plots
147 if strcmp(plotswitch, 'on')
148     color = 'rg';
149     cla(handles.axes2, 'reset');
150     hold(handles.axes2, 'on');
151     for i = 1:signals
152         plot(handles.axes2, x{i}, y{i}, color(i), 'linewidth', 2);
153     end
154     hold(handles.axes2, 'off');
155     axis(handles.axes2, 'tight');
156     xlabel(handles.axes2, 'Azimuth [\circ]');
157     ylabel(handles.axes2, 'Normalized Power (dB)');
158     title(handles.axes2, 'DOA Spatial Pseudo-Power Spectrum');
159     if signals == 1
160         legend(handles.axes2, 'Signal 1');
161     else
162         legend(handles.axes2, 'Signal 1', 'Signal 2');
163     end
164
165     plot(handles.axes3, xfreq/1e3, yfreq, 'g', 'linewidth', 2);
166     axis(handles.axes3, 'tight');
167     title(handles.axes3, 'Capon Frequency Spectrum');
168     xlabel(handles.axes3, 'Frequency (KHz)');
169     ylabel(handles.axes3, 'Normalized Power (dB)');
170 end
171
172 %% Save file

```

```
173 if strcmp(plotswitch,'on');  
174     save('C:\Documents and Settings\Wireless Lab\My Documents\MATLAB\  
        Will\bsdata_all.mat');  
175 end
```