

A SMALL ACOUSTIC GONIOMETER FOR GENERAL PURPOSE RESEARCH

by

Michael Pook

A dissertation

submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy in Electrical and Computer Engineering

Boise State University

December 2015

© 2015

Michael Pook

ALL RIGHTS RESERVED

BOISE STATE UNIVERSITY GRADUATE COLLEGE

DEFENSE COMMITTEE AND FINAL READING APPROVALS

of the dissertation submitted by

Michael Pook

Thesis Title: A Small Acoustic Goniometer for General Purpose Research

Date of Final Oral Examination: 15 October 2015

The following individuals read and discussed the dissertation submitted by student Michael Pook, and they evaluated his presentation and response to questions during the final oral examination. They found that the student passed the final oral examination.

Sin Ming Loo, Ph.D.	Chair, Supervisory Committee
Jim Browning, Ph.D.	Member, Supervisory Committee
H. P. Marshall, Ph.D.	Member, Supervisory Committee
Simon Y. Foo, Ph.D.	External Examiner

The final reading approval of the dissertation was granted by Sin Ming Loo, Ph.D., Chair of the Supervisory Committee. The dissertation was approved for the Graduate College by John R. Pelton, Ph.D., Dean of the Graduate College.

DEDICATION

To the four greatest temporal blessings God has bestowed upon me: my parents and grandparents. The love and support you have shown me is equaled only by my love and admiration for you.

To my grandfather, my final experiment belongs to you. You have always postulated that I will not melt in the rain. As this work is a part of me, I propose we place it outside in a storm to see if it floats away. I do believe we will finally prove your theory.

ACKNOWLEDGEMENTS

I would like to thank Dr. Sin Ming Loo for his guidance and support throughout this process and my academic career. His insistence that I keep pushing forward and genuine interest in my research and success made this work possible. I truly appreciate all that he has done for me as an advisor.

I further extend my heartfelt thanks to my committee members for their direction and time spent reviewing my work. Their guidance and insight were instrumental in improving my research.

Additionally, I would like to thank all who were involved in this research: Stephanie Johnson, Vikram Patel, Nicholas Terrell, Ali Ibrahim, Corey Warner, and Mike Clements.

I would like to offer special thanks to Josh Kiepert and Andrew Landoch for their considerable technical and moral support. The software and firmware they provided was invaluable during the testing process. More importantly, the moral support they offered kept me going when I was most inclined to leave academia.

Finally, I would like to thank my family and friends for their love and support. I am especially grateful to my parents and grandparents for their support and for easing my burden in every way they could. They suffered at least as much as I have during this process, and I would not have finished this work if not for them. For this and many other things, I am eternally grateful to them.

ABSTRACT

Understanding acoustic events and monitoring their occurrence is a useful aspect of many research projects. In particular, acoustic goniometry allows researchers to determine the source of an event based solely on the sound it produces. The vast majority of the acoustic goniometry research projects used custom hardware targeted to the specific application under test. Unfortunately, due to the wide range of sensing applications, a flexible general purpose hardware/firmware system does not exist for this research. This dissertation focuses on the development of such a system, which encourages the continued exploration of general purpose hardware/firmware and lowers barriers to research in projects requiring the use of acoustic goniometry. Simulations have been employed to verify system feasibility, and a complete hardware implementation of the acoustic goniometer has been designed and field tested. The results are reported, and suggested areas for improvement and further exploration are discussed.

TABLE OF CONTENTS

DEDICATION	iv
ACKNOWLEDGEMENTS	v
ABSTRACT	vi
LIST OF TABLES	xi
LIST OF FIGURES	xii
LIST OF ABBREVIATIONS	xvi
CHAPTER ONE: INTRODUCTION	1
1.1 Background	2
1.2 Contributions	7
1.3 Outline	8
CHAPTER TWO: PREVIOUS WORK	10
2.1 Application Research	10
2.2 Hardware Research	15
2.2.1 Audible and Higher Frequency Research	15
2.2.2 Infrasonic Research	19
2.3 Algorithm Research	29
CHAPTER THREE: ACOUSTIC GONIOMETER THEORY	37
3.1 Sound Propagation	37
3.2 Goniometer Theory of Operation	42

3.3 Sampling Theory.....	49
3.3.1 Spatial Sampling	49
3.3.2 Temporal Sampling.....	51
3.4 Filtering.....	54
3.4.1 Filtering Electrical Noise	55
3.4.2 Filtering Acoustic Noise	56
3.4.3 Acoustic Metamaterials	59
3.5 Event Detection.....	60
3.5.1 Thresholding	60
3.5.2 Fingerprinting	61
3.5.3 Fisher Statistic.....	62
3.6 Embedded Algorithm Implementation	64
CHAPTER FOUR: SIMULATIONS	66
4.1 Sensor Spacing Simulation	68
4.2 Time Delay Simulation.....	75
4.3 DOA Simulation	85
4.4 Event Detection Simulation	92
CHAPTER FIVE: IMPLEMENTATION.....	94
5.1 Firmware	95
5.1.1 Operating System.....	96
5.1.2 ADC Reader.....	99
5.1.3 Goniometer State Machine	103
5.2 Hardware.....	109

5.2.1 Platform.....	109
5.2.2 Sensor.....	117
5.2.3 Mechanical Systems.....	124
CHAPTER SIX: RESULTS	141
6.1 Data Collection	141
6.2 Laboratory Tests	145
6.2.1 Test Setup.....	145
6.2.2 Data Analysis	147
6.2.3 Acoustic Goniometer Performance Analysis.....	153
6.3 Lecture Hall Tests	154
6.3.1 Test Setup.....	154
6.3.2 Data Analysis	155
6.3.3 Acoustic Goniometer Performance.....	158
6.4 Field Tests.....	159
6.4.1 Test Setup.....	159
6.4.2 Data Analysis	163
6.4.3 Acoustic Goniometer Performance Analysis.....	172
CHAPTER SEVEN: FUTURE WORK	176
7.1 Algorithm Research	176
7.1.1 Scheduling.....	176
7.1.2 Event Handling	178
7.2 Hardware Research	181
7.2.1 Motherboard.....	182

7.2.2 Sensors	185
7.2.3 Mechanical Systems.....	185
CHAPTER EIGHT: CONCLUSIONS	187
REFERENCES	192
APPENDIX A.....	198
Building the Acoustic Goniometer	198
APPENDIX B	202
Deploying the Acoustic Goniometer	202
APPENDIX C	209
Understanding the Acoustic Goniometer Firmware	209

LIST OF TABLES

Table 4.1	Simulated error for varying sample rate (f_s) and source distance (d_s) while maintaining inter-sensor spacing ($d= 2\text{m}$), azimuth ($AZ=45^\circ$), and elevation ($EL=45^\circ$).....	73
Table 4.2	Simulated error for varying inter-sensor spacing (d) and source distance (d_s) while maintaining sample frequency ($f_s= 10\text{kHz}$), azimuth ($AZ=45^\circ$), and elevation ($EL=45^\circ$).....	74
Table 6.1	Analysis of acoustic goniometer test results shown in terms of accuracy for various DOA angles	174
Table A.1	Acoustic Goniometer Motherboard Bill of Materials	198
Table A.2	Acoustic Goniometer Sensor Board Bill of Materials	200

LIST OF FIGURES

Figure 1.1	Basic Goniometer Localization.....	5
Figure 2.1	AWSN Counter Sniper System [6].....	17
Figure 2.2	Raytheon BBN Technologies Commercial Gunfire Targeted Acoustic Goniometers.....	19
Figure 2.3	Van Lancker’s Acoustic Goniometer.....	22
Figure 2.4	Goniometer Deployed by Olivieri et al. [35].....	24
Figure 2.5	Low Pass Mechanical Filter Designed by Marcillo et al. [34]	28
Figure 3.1	Imaginary Sources Produced by Reflections when a Sound Wave Encounters an Object	40
Figure 3.2	Sound Reflections and Multipath in an Enclosed Space	41
Figure 3.3	Acoustic Goniometer Sensor Pair Diagram.....	43
Figure 3.4	Acoustic Goniometer Vector Diagram	45
Figure 3.5	Example Source Locations Explaining Azimuth and Elevation.....	47
Figure 3.6	Goniometer Sensor Data Showing Mild Multipath Distortion.....	53
Figure 3.7	Example of a Microphone Blimp (from Rode Microphones LLC)	58
Figure 4.1	Power Spectral Density Plots for Door and Gunshot Sound Events	68
Figure 4.2	Setup for First Series of Simulations	70
Figure 4.3	Simulation Validation Using Ideal Time Resolution.....	72
Figure 4.4	Door Slamming Test.....	78
Figure 4.5	Mic1-Mic2 (Red-Blue) Correlation Results	80
Figure 4.6	Mic3-Mic4 (Red-Blue) Correlation Results	81

Figure 4.7	Mic2-Mic3 (Red-Blue) Correlation Results	82
Figure 4.8	Mic2-Mic4 (Red-Blue) Correlation Results	83
Figure 4.9	Mic1-Mic3 (Red-Blue) Correlation Results	84
Figure 4.10	Mic1-Mic4 (Red-Blue) Correlation Results	85
Figure 4.11	DOA Simulation Event Source Layout.....	86
Figure 4.12	Shot Data with Source at 90° (Azimuth) with Respect to the Antenna	88
Figure 4.13	Shot Data with Source at 45° (Azimuth) with Respect to the Antenna	89
Figure 4.14	Shot Data with Source at 63.4° (Azimuth) with Respect to the Antenna .	90
Figure 4.15	Shot Data with Source at 135° (Azimuth) with Respect to the Antenna ..	91
Figure 5.1	Acoustic Goniometer Implementation Block Diagram	95
Figure 5.2	Firmware Organization	98
Figure 5.3	ADC Reader Data Analysis	103
Figure 5.4	Goniometer State Machine	104
Figure 5.5	Event Window Parts	107
Figure 5.6	Acoustic Goniometer Prototype Hardware Platform.....	114
Figure 5.7	Acoustic Goniometer Motherboard	116
Figure 5.8	Initial Prototype Infrasound Sensor	120
Figure 5.9	Final Prototype Acoustic Goniometer Sensor.....	124
Figure 5.10	Acoustic Goniometer Antenna Prototype	128
Figure 5.11	Acoustic Goniometer Enclosures.....	130
Figure 5.12	Gun Shot Data with Noise in Windy (Gusts at 20 mph) Conditions	132
Figure 5.13	Commercial and Custom Wind Screens	133
Figure 5.14	Wind Screens Tested in 20mph Gusts	137

Figure 5.15	Best Windscreen Solution Test Results	139
Figure 6.1	ADC Reader Raw Data File.....	143
Figure 6.2	Acoustic Goniometer Recorded Data	144
Figure 6.3	Laboratory (top-down view) Setup with Indicated Source (Blue) and Sensor Placement (Red)	147
Figure 6.4	Raw Data from Laboratory Test with 90° Azimuth.....	148
Figure 6.5	Raw Data from Laboratory Test with 45° Azimuth.....	150
Figure 6.6	Raw Data from Laboratory Test with 135° Azimuth.....	151
Figure 6.7	Shooting Test Data with Minimal Evidence of Multipath Reflections...	152
Figure 6.8	Lecture Hall (top-down view) Setup with Indicated Sources (Blue) and Sensor Placement (Red)	155
Figure 6.9	Raw Data from Lecture Hall Test.....	157
Figure 6.10	Goniometer Field Test Terrain.....	161
Figure 6.11	Field Test Event Source Layout.....	162
Figure 6.12	Data from High Speed Wind Tests without Sufficient Wind Screens....	164
Figure 6.13	Raw Test Data Collected from 10 Shooting Events	166
Figure 6.14	Raw Test Data Showing Variations in Magnitude Irrespective of DOA.....	168
Figure 6.15	Magnified View of Amplitude Variance Occurring Irrespective of DOA.....	171
Figure B.1	Generalized Tetrahedron Simulation Code.....	204
Figure B.2	Goniometer Configuration File.....	205
Figure B.3	Acoustic Goniometer	207
Figure B.4	Hex File Processor: Raw ADC Reader Data to CSV Converter	208
Figure C.1	GONIOMETER_DEV.h- Selecting the Operating Mode	210

Figure C.2	CCM RAM Buffer Instantiation	211
Figure C.3	ADC Initialization Variable for Setting Channels, Sample Rate, and Resolution	212

LIST OF ABBREVIATIONS

1-D DWT	One Dimensional Discrete Wavelet Transform
ADC	Analog to Digital Converter
AWSN	Acoustic Wireless Sensor Network
BGA	Ball Grid Array
BOM	Bill of Materials
BSU	Boise State University
BLMS	Block Least Mean Squares
CCM	Core Coupled Memory
DMA	Direct Memory Access
DOA	Direction of Arrival
DSP	Digital Signal Processor or Digital Signal Processing
FIR	Finite Impulse Response
FPGA	Field Programmable Gate Array
FPU	Floating Point Unit
GPS	Global Positioning System
HDL	Hardware Description Language
HSIL	Hartman Systems Integration Laboratory
HDL	Hardware Description Language
IIR	Infinite Impulse Response
IP	Intellectual Property

LE	Logic Element
LMS	Least Mean Squares
LQFP	Low Profile Quad Flat Package
MPH	Miles Per Hour
PSD	Power Spectral Density
QPC	Quadratic Phase Coupling
RAM	Random Access Memory
SOC	System on a Chip
SVD	Singular Value Decomposition
TOA	Time of Arrival
WSN	Wireless Sensor Network

CHAPTER ONE: INTRODUCTION

Acoustic events play an important role in communicating information about nearly any given environment. Whether the event is in a range perceivable by humans or not, many phenomena that are of interest for characterizing an environment produce a sound. A sound may hail the advent of an avalanche, the cracking of rock signaling the start of a landslide, or even the presence of gunfire. Understanding acoustic events and monitoring their occurrence can allow researchers to predict, prevent, and locate the source of interesting or potentially harmful phenomena. Consequently, research into acoustic events is a very popular aspect for many projects. Unfortunately, due to the specialized nature of many acoustic applications, not much exists (aside from simple microphones or microphones tuned to specific frequencies) for conducting acoustic research. This has led to the development of many specialized sensor systems with limited flexibility. While specialized equipment serves the purpose for which it was designed, this creates a barrier to research if the scientists involved do not have the expertise needed to create their own equipment. This must be expected to a certain extent for an area of research as broad as acoustic monitoring. However, more flexible hardware designed to aid in semi-specific applications could help in furthering research.

Advances in several fields of technology in recent years have created a significant increase in the availability of wireless sensor networks (WSNs) for both research and general consumer products. A WSN is a collection of sensor nodes all wirelessly connected to a central computer (and possibly each other) and deployed to monitor

phenomena over an entire area. The application of WSNs ranges from simple systems designed to protect our homes and increase health awareness to complex networks responsible for monitoring the environments in which we spend our time. Regardless of the application, WSNs typically share a common goal: to better characterize, monitor, and control a given environment. The method in which a WSN achieves this goal depends mainly on the type of environment and the characteristics or events of particular interest. For many environments, sound is an important aspect worthy of characterization and can be especially useful in event detection applications. Sudden, drastic changes in an environment are usually accompanied (either before or after) by some variation in sound (e.g., gun shots, avalanches, etc.). If the variation is in the detectable range of a sensor, changes in sound can be used to effectively identify or, in some cases, predict events. Additionally, using multiple sensors and the known speed of sound, one can use delays between detected events to not only identify the occurrence of a sound event, but also the location of its source through a process known as acoustic goniometry. Thus, due to its utility and applicability to a wide range of environments coupled with the proliferation of WSNs, acoustic wireless sensor network (AWSN) research has become an important aspect of research in many fields of study.

1.1 Background

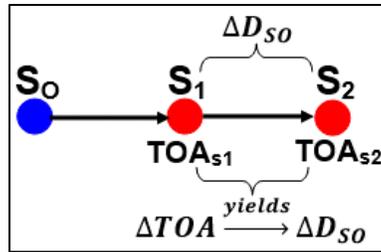
Processing sound and determining its cause has always played an important role in our understanding of the environment we occupy. Humans use the concept of acoustic goniometry countless times each day to solve a variety of problems. When we need another person's attention, we call out their name, and they acknowledge us by first detecting an acoustic event and determining the source of the commotion. Excepting

special circumstances, the individual can usually look directly at the source without needing to scan their field of vision. The same process can be applied in determining the source of danger, anticipating the arrival/departure of an object, and many other phenomena which produce sound. This ability, which comes as second nature to us, is the result of our brains automatically applying acoustic goniometry to a sensory input in order to solve a given problem.

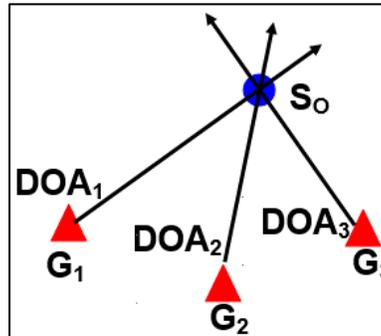
Twenty years ago, environmental sensing research and development was reserved for corporations, governments, and universities with large budgets due to the limited availability of sensors. In recent years, commercially available sensors have become abundant for many applications. The proliferation of simple sensors has made environmental sensor system research and related fields of study more accessible to researchers with backgrounds ranging from the weekend embedded systems hobbyist to the computer engineering graduate student. This increased accessibility sparked an incredible interest in environmental sensor systems and product development, which has led to a substantial growth in research. However, such growth has yet to occur for acoustic goniometer research. This useful field of study suffers from limited sensing options, which require either spending a large sum or designing a custom sensor suite. A survey of the research completed shows that the majority of goniometer projects rely on very targeted, custom built sensors designed by those conducting the research (see Chapter 2). Very often, these systems are designed by teams with limited computer engineering experience and tend toward simpler designs, which suffer from the need for large spacing between sensors (large antenna size), poor accuracy, and/or limited adaptability. Some sensor systems are commercially available but are targeted toward

very specific applications and are only available to select groups at a high cost (e.g., military equipment; see Chapter 2).

Although the process of designing a sensor system to perform such tasks is not second nature, acoustic goniometry can be performed using fairly inexpensive processors and microphones. While this may not have been the case when the first acoustic goniometers were created, modern microprocessors have become significantly more capable and power efficient. Consider the simple example of two microphones each at an unknown distance from an ideal sound point source. If a system is capable of detecting the arrival at each microphone and accurately timestamping each event, the difference in the time of arrival (TOA) between the two sensors can be calculated and used to determine the difference in distance between the sensors and point source (as illustrated in Figure 1.1 a). If, instead of two microphones, the system includes four whose relation to one another is known (spacing and geometric configuration), the calculated differences in distance from the point source between sensor pairs can be used to calculate the direction of arrival (DOA) for the sound event. Now, if the system were to be duplicated and these collections of four microphone systems (antennas) were spaced at known distances from one another, localization of the point source could be performed by a computer with knowledge of each antenna's DOA estimate (as shown in Figure 1.1 b). Explained in this manner, the process seems exceedingly simple. However, the model chosen to explain the process is highly idealized and makes assumptions that must be addressed during implementation in real hardware and software.



(a) Relationship Between ΔTOA and $\Delta Distance$ from Sensors to Source



(b) Use of Multiple Goniometers (G_{1-3}) to Locate a Source (S_0)

Figure 1.1 Basic Goniometer Localization

The first assumption made was with regards to the nature of the sound source and the path along which it traveled. While an ideal point source may work as a good model for distant or simple events, some events may occur close to the antenna or have multiple sources (e.g., close proximity avalanches, multiple objects falling simultaneously, etc.). Furthermore, outside of a lab environment, nature is rarely so kind as to give researchers a completely unobstructed path between their equipment and the phenomenon they wish to observe. Thus, multipath becomes a significant source of concern. As sound travels between the source and the antenna, it will encounter other objects (e.g., trees, buildings, ground, etc.) along its path. As the sound rebounds off these surfaces, lower amplitude copies of it will be created which may also reach the microphones. If not properly filtered out, these signals could result in erroneous event detection and incorrect DOA estimation.

The given model also does not elaborate on the complex process of detecting and identifying events. Microphones tuned to specific frequencies can be used but are costly. Inexpensive, general purpose microphones respond to a wide range of frequencies. Thus, filtering must be employed to remove signals that do not correspond to the event of interest. Given a microphone design that can pick out only the desired frequencies, thresholds must still be set to determine whether an event has occurred. Once an event is detected, the system needs some method to decide if the event was caused by the desired phenomenon or an unrelated event that produced a similar sound. Some form of pattern recognition must be used to differentiate between the desired events and those not of particular interest. This adds further complexity in the sense that a window of data must be stored in order to perform such comparisons.

Finally, the issue of accurately timestamping measurements and synchronizing clocks between microphones must also be addressed. This aspect of the system is exceptionally important with regards to accuracy, range, and the required spacing between microphones. Shorter inter-sensor spacing demands finer resolution timestamping in order to maintain accuracy over the same distance from event sources. Beyond resolution, since the calculation for DOA is based on the difference in arrival time at each microphone, the clocks for each microphone must be at least precisely synchronized as the resolution of the timestamps. This can be handled a number of ways in either hardware or firmware (e.g., military grade GPS or time synchronization algorithms).

1.2 Contributions

The ultimate goal of this dissertation is to lower barriers to research in fields of study requiring the use of acoustic goniometers. Human beings in general are curious about their environment and have an insatiable desire for knowledge. When equipped with the proper tools and sufficient resources, people explore, push boundaries, innovate, and enrich the collective knowledge of the group. However, when tools are insufficient or resources are scarce, progress can be slowed significantly or even halted completely. If a researcher does not have the requisite skills or time to build their own tools, they may be forced to abandon a field of study in favor of one that is more practical for their situation. When this occurs, potentially valuable information which may have been garnered from their research may be delayed or completely lost.

In order to serve the ultimate goal of lowering barriers to research, this dissertation focuses on the development of an inexpensive, small acoustic goniometer that is easy to deploy and capable of being adapted to meet the needs of a wide range of research applications while maintaining reasonable accuracy. Designing such a system required careful consideration of a variety of factors. The acoustic goniometer had to have the ability to work over a wide range of frequencies. Furthermore, the system needed to be simple enough for the average researcher to perform adjustments, test theories, and add capabilities. The design also had to include a structure (or example thereof) which makes deployment and use of the system as painless as possible. Each of the aforementioned design features are complex enough to consider by themselves, but each of these had to be weighed against the accuracy of the system. To achieve the best possible accuracy, the acoustic goniometer design would have had to be specifically

targeted to one application with a set frequency and be implemented on the most advanced hardware locked into a single sensor configuration on a predetermined structure. However, this would have defeated the goals of this research entirely as the resulting system would have been difficult to adapt, costly, and not well suited toward experimentation and furthering research. From the other perspective, the most flexible system with the simplest user interface would not be useful in any field of research if its accuracy is unacceptably low. Thus, a balance had to be determined and practical sacrifices had to be made on both sides.

Meeting all of the requirements mentioned previously was not a simple process, but the task has been accomplished. The design created for this dissertation is a small acoustic goniometer (minimum microphone spacing of 2 m) with a single centrally located processing system. The small, single processor design allows the system to maintain accurate timestamps between microphones without wasting resources on complex time synchronization algorithms/schemes or requiring costly GPS hardware. The processor includes an internal, high-speed analog to digital converter (ADC) which provides sufficiently fast sampling rates to support the small inter-sensor spacing while maintaining reasonable accuracy for the DOA calculations. The main circuit board for the goniometer is further equipped with a wireless transceiver, allowing it to be used in a network facilitating the transmission of analyzed data to researchers or enabling more accurate localization via cooperative sensing operations.

1.3 Outline

The following chapters discuss the topic pertaining to acoustic goniometry and infrasonic monitoring as they relate to the scope of this dissertation. Chapter 2 describes

the existing technology and research created by other researchers. Chapter 3 details the theory behind acoustic goniometry. Chapter 4 outlines the design of the acoustic goniometer and the simulations used to verify the implementation. Chapter 5 discusses the actual hardware and firmware implementation of the system. Chapter 6 covers the results of testing the acoustic goniometer designed for this research. Chapter 7 provides details on which areas of the research would benefit from further study. Finally, the conclusions drawn from this work are detailed in Chapter 8.

CHAPTER TWO: PREVIOUS WORK

Given the number of phenomena that produce sound, the amount of research conducted in the fields of acoustic goniometry and infrasonic monitoring is not surprising. Researchers have focused on a variety of facets including applications for acoustic goniometers and infrasonic monitoring, improvements to sensor hardware, and efficient/effective firmware design. The following sections detail both the research that has been conducted and the technology used in the process along with a discussion as to how they relate to this research.

2.1 Application Research

Although acoustic variations in the audible range can be an effective tool for many applications, certain environments can make their use difficult or impossible. Sound produced below the audible range (or infrasound) can be a precursor to or side effect of many natural and manmade phenomena (e.g., landslides, volcanic eruptions, avalanches, nuclear testing/explosions, etc.). Infrasound signals experience less attenuation over long distances than their higher frequency counterparts and have the added advantage of being discernible in environments with high audible noise pollution. Since many of the same principles used in the analysis of higher frequency sound can be applied to infrasound, the two areas of research are closely linked. Research in the use of infrasound to detect and characterize natural and manmade phenomena has been conducted extensively over the past several decades. Systems designed for or as part of this research range from a simple single microphone to networks composed of multi-

sensor nodes, which may include multiple microphones and specialized sensors targeted toward infrasound detection. The computational power of these systems varies as well and has run the gamut from simple microprocessors and digital signal processors (DSPs) to field programmable gate arrays (FPGAs) and desktop computers. This section covers these research projects in greater detail.

Some researchers have published short papers on possible applications of infrasound monitoring supported by preliminary data. The authors of [1] published a brief description of data collected on forest fires from a National Oceanic and Atmospheric Administration research outpost specializing in infrasonic monitoring. According to their work, the intensity of forest fires can be monitored via infrasound sensors (0-200 Hz). While no discussion of sensor types or processing was provided, the authors were able to determine the dominant frequencies produced by the fires and correlate the intensity of the infrasound signal to the distances to and diameters of the fires. While fire detection and generic infrasound data collection are not the primary purpose for the research detailed in this dissertation, the prototype can still be useful for this type of research. Since fires of this scale are not planned and can occur at any time, a small scale acoustic goniometer with data recording capabilities could improve the ability of researchers to continue monitoring these phenomena by easing the deployment of sensors to prime locations. Another example of monitoring/characterizing infrasound phenomena is the work of Nishiyama et al. [2]. The authors recorded the infrasonic emissions of wind near the Rocky Mountains. They propose such data could be used to monitor areas of turbulence and instability, which could affect aircraft [2]. Both cases show the potential for using acoustic goniometry or infrasonic characterization to improve the safety of

specific environments. The creation of more tools for characterizing such phenomena quickly and easily could drastically improve the rate at which research in these as well as other areas could produce useful and potentially lifesaving results.

The work of Zhu et al. focused on the characterization of infrasound emissions from various types of rock under uniaxial compression [3]. The overarching goal of their research was to identify a better means of predicting rock landslides. To this end, the team created a system with standard off-the-shelf components and focused on developing the algorithms required to analyze the signals. The sensor used in this project was a specialized infrasound sensor, which functions similarly to a condenser microphone. Output from the sensor was filtered to remove frequency components higher than 100 Hz, amplified, and sampled by an FPGA at a rate of 512 Hz. Processing and analysis was completed using a desktop computer and Matlab. Analysis of the initial data showed a significant amount of background infrasonic noise. Thus, in addition to the analog filter, digital low-pass filtering was performed on the signal in Matlab using a wavelet based denoising algorithm. Then, the short-time frequency transform of the signal was calculated and the results were analyzed to determine if such a process could produce a means for predicting landslides. The researchers concluded that their findings warranted further study, and infrasound monitoring showed promise to produce a better way to predict rock landslides. Unfortunately, their work was hampered by the necessity of designing a system in order to conduct algorithm research. The design developed by the team employed an FPGA and an expensive (relative to simple electret condenser microphones) custom infrasound sensor. A generic sensor platform with inexpensive microphones and a microprocessor could be useful in this type of research, easing

deployment and algorithm development. The prototype developed as part of this dissertation research could provide a solution that would simplify writing code and decrease the barriers to furthering research in this field of study.

Werner-Allen et al. developed a proof of concept prototype in their work for monitoring volcanic activity via an acoustic wireless sensor network (AWSN) [4]. The network topology for their field test consisted of a small set of acoustic sensors that relayed data to an aggregator node. The aggregator, in turn, forwarded the data to a computer for processing. The sensor nodes used custom sensor boards containing electret condenser microphones and filtering/amplifying circuitry connected to Mica2 motes running TinyOS (for digitization and transmission). Each node also included a GPS antenna to aid in localization as well as to maintain accurate/synchronized timestamping. After the successful completion of their project, the researchers determined that further study could prove valuable but would require increasing the size of the AWSN and creating a permanent installation on their test volcano. Their hardware worked well for data collection. However, such a setup would not work well for a project that required acoustic goniometer calculations. The inclusion of a GPS module on each node would aid in the localization process and ease deployment, because accurate notes on relative sensor positions and precise sensor placement would not be as crucial (assuming a high grade GPS were used). The downside to this approach is the cost incurred by outfitting every node with a GPS antenna and using a computer for processing rather than the embedded hardware. A system that did all the processing and storage at the aggregator node while maintaining the ability to forward data to another location would be useful in more situations.

Rud et al. did some intriguing work on the use of infrasound for non-invasive heart monitoring, which shows both the versatility of infrasound characterization of phenomena and potential for multiple applications in the health industry [5]. The goal of their work was to characterize heart murmurs using infrasound in order to explore non-invasive means of monitoring/diagnosing abnormal cardiac conditions/events. Their sensor equipment (Kardiac Infrasound Device: KID) used a specialized infrasonic sensor coupled with accelerometer measurements to perform data fusion for the monitoring of chest vibrations. Data processing for KID was performed by a computer running Matlab. The sensor system was found to be capable of accurately distinguishing normal heart behavior from abnormal behavior. However, the tests were limited in size, and the researchers determined that more data would be needed to characterize and distinguish which underlying conditions could be causing abnormal cardiac events via infrasound detection. Unless heartbeats could be detected from a significant distance and used to track individuals in an environment (more exploration would be needed), this research most likely would not benefit from goniometer technology. However, a single sensor from an acoustic goniometer could be used to monitor a patient or all sensors could be used to monitor a room full of patients if the system were properly developed. Due to the advanced acoustic sensing nature of a purely embedded implementation of an acoustic goniometer, the design would be easily repurposed to fit any number of additional applications like the research detailed in [5].

Though much of the research mentioned in this section does not pertain to goniometry specifically, it played an important work in the current research. These examples prove the utility of infrasonic monitoring of phenomena and show potential

uses for goniometers (e.g., locating rock landslides from a distance). Furthermore, much of the work presented here discussed the potential sources for infrasound noise and methods for mitigating their effects on the data.

2.2 Hardware Research

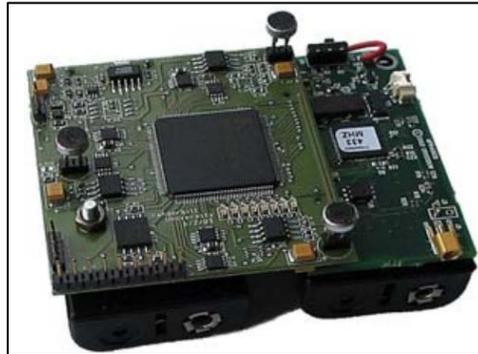
Acoustic goniometry research can be split into two main categories based on the frequency of the sound produced by the studied phenomena: research focusing on those events producing sound in and above the audible frequency range and those events producing sound below the audible frequency range. While the research conducted for this publication focused primarily on lower frequencies, the techniques used in monitoring audible and higher frequency events are still applicable, and the prototype created as part of the current work was designed to be used for as wide a frequency range as possible. Thus, some of the work done in the higher ranges is highlighted in this section as well as the research conducted below the audible range.

2.2.1 Audible and Higher Frequency Research

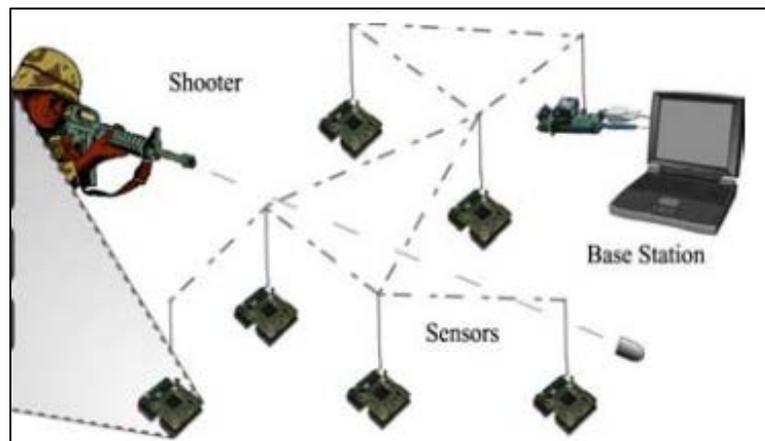
The most commonly observed and measured sound events are those which occur in and above the audible range. Some of these phenomena produce detectable levels of infrasound as well and, consequently, are related more to the current research. Systems designed to detect and locate the source of gunshots, for example, may use the audible muzzle blast, the infrasound generated by the small explosion, or the high frequency signal produced by the shockwave along the bullet's trajectory, or any combination of the three. The complexity of these systems and number of sensors varies widely. Some systems are deployed as single wearable or vehicle mounted units, while others consist of a network of sensors dispersed across an area of interest.

The authors of [6] developed an AWSN to serve as a countersniper system. The nodes consisted of Berkeley notes outfitted with simple sound sensors (microphones connected to amplifiers with adjustable gain) using Xilinx FPGAs for high speed signal processing. The wireless network of sensors was deployed throughout a target area and communicated with a single base node. The sensors perform signal processing and handle event detection on their own. When an event is detected (gun shot), the nodes forward the relevant data to the central computer for analysis. The central computer fuses the data from each of the nodes that reported the event in order to calculate the location of the shooter and the direction of the shot. Figure 2.1 shows one of their sensor nodes as well as a representation of their AWSN under test. The sensors use the initial blast from the rifle coupled with shockwaves created by the bullet along its trajectory to determine direction. Their timestamped data allows the central computer to calculate time of arrival at each location and filter out noise caused by multipath. Although this system is targeted toward shooter detection, it could be modified for more general purpose acoustic goniometry research. However, while this research shows the feasibility of goniometry using a coordinated AWSN, deployment of this network would be rather cumbersome in some situations and could require a large number of nodes depending on the desired accuracy. This implementation requires the area being monitored to be saturated with nodes in order to accurately determine the DOA for a bullet. If the bullet's trajectory does not take it through the area saturated with sensors (as shown in Figure 2.1), the system's performance suffers. Additionally, the nodes placed to monitor an area must be placed accurately in relation to one another for the computer system to correctly analyze the data. For certain applications, such a design may work well. However, the deployment of

the sensor network detailed in [6] is far too cumbersome to be useful as a general purpose research tool. Additionally, while the use of FPGAs does provide higher flexibility and greater processing capabilities, their inclusion instead of a simple high-speed microprocessor limits the number of researchers who could potentially modify the system. Thus, although the sensor network fit the needs of the research in [6] and furthered this field of study, the design is not practical for general purpose use.



(a) Sensor Node [6]



(b) Example of AWSN Deployment [6]

Figure 2.1 AWSN Counter Sniper System [6]

Given the crucial need for determining sniper location (or any source of incoming fire) on battlefields, the existence of commercial products created to serve such purposes

is to be expected. In addition to the aforementioned academic research, several corporations worldwide have developed sensor systems for tracking gunfire. An example of such a company, Raytheon BBN Technologies has a portfolio of devices that use detection of acoustic events to solve this particular problem [7-12]. Dubbed Boomerang, Boomerang Air, and Boomerang Warrior-X, Raytheon's devices use the acoustic signature of muzzle blasts and the high frequency signals from projectile shockwaves to determine gunfire events and shooter locations. Figure 2.2 shows some of their designs used for determining the DOA of gunfire. The devices range in size from large stationary antennas (Figure 2.2 b) and large mobile versions attached to vehicles (e.g., armored personnel carriers, aircraft: see Figure 2.2 a) to small shoulder mounted devices (Figure 2.2 c). Their systems use acoustic sensors, pressure sensors, or accelerometers. The technology behind the Raytheon devices is very similar to the research conducted for this dissertation. The inter-sensor spacing seen in Figure 2.2 (b) is clearly less than 2 m. However, the computer processing the data for this sensor is quite obviously (see the box near the tripod in Figure 2.2 b) larger, more expensive, and more capable than a simple embedded design. The shoulder mounted system seen in Figure 2.2 (c) uses multiple sensor types (i.e., accelerometer and microphone) and sensor data fusion to sense projectiles. It functions as part of a mobile AWSN similarly to [6] but more advanced in its execution. As with [6], this type of implementations is limited in its utility for general purpose research and is still dependent on a powerful/expensive central computer to handle the data processing. Additionally, all of these systems are tuned specifically for pinpointing the location of hostile gunfire. The filters and algorithms are specifically designed to ignore all other infrasound events. Consequently, such a system could not be

used for general research without significant modification. Finally, these products were created specifically for the military. This makes them both difficult to procure and too expensive for general purpose research.



(a) Vehicle Mounted Acoustic Goniometer



(b) Stationary Acoustic Goniometer (c) Wearable Acoustic Goniometer

Figure 2.2 Raytheon BBN Technologies Commercial Gunfire Targeted Acoustic Goniometers

2.2.2 Infrasonic Research

Although not as common as their higher frequency counterparts, research and systems targeting the infrasound range are still fairly common. Improvements in technology over the past twenty years (acoustic, processing, semiconductor, etc.) have

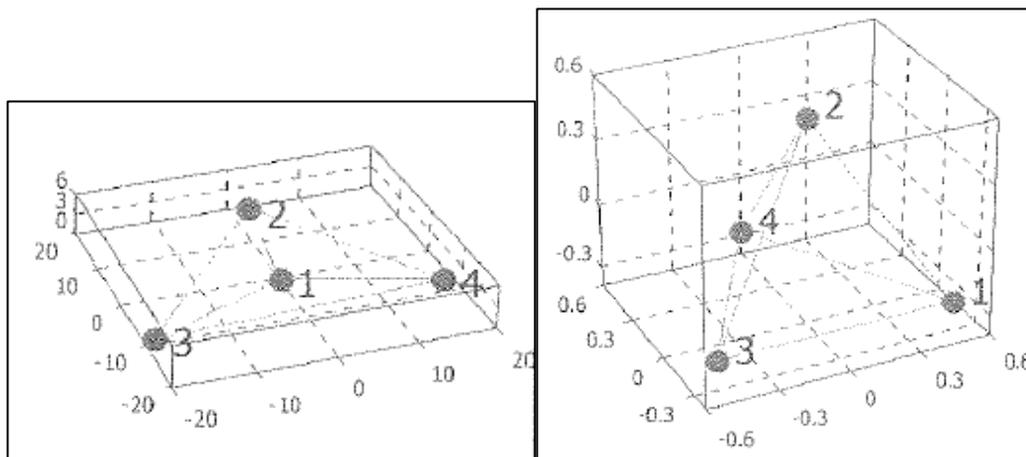
resulted in a vast increase in research pertaining to infrasonic events. The availability and decreased cost of specialized microphones, faster processors with floating point hardware, and energy efficient FPGAs has made the design and deployment of infrasonic sensors more attainable. This section details some of the important research completed in the field of infrasonic sensing and acoustic goniometry.

Van Lancker focused on the development of a generic goniometer system to track, among other events, avalanches using sound sensors (microphone and amplifier arrays similar to [6]) in his doctoral dissertation [13]. Similarly to [6], the hardware from his research also used a centralized computing station to handle some of the data processing. The author's system used a combination of well-established algorithms for this particular field to do beamforming and subspace-based localization/direction finding. Although he experimented with several quantities of microphones, each of his sensor nodes used 4 microphones whose geometry/spacing was determined based on the application and the frequency of the expected noise. For instance, avalanche detectors used an "Echo Star" pattern (Figure 2.3 b) with a spacing between 20-35 m, while a helicopter detector used a tetrahedron pattern (Figure 2.3 c) with a spacing of 1 m between microphones. He successfully designed, built, and deployed his sensor networks for several applications including a 4 year study of avalanche detection/monitoring in Switzerland [14-15]. His research spawned what is known as the ARFANG Station in Switzerland, which currently monitors avalanches in the Alps (see Figure 2.3 a). Van Lancker's work was one of the biggest boons to research in this field. His design laid the ground work for better understanding the design and implementation of such systems. However, due to progress in the abilities of inexpensive microcontrollers and FPGAs, the

technology in his work is somewhat dated. Newer technologies have changed the design tradeoffs significantly and have the potential to make smaller, less expensive designs possible. At the time of his research, microprocessors did not have the ability to sample and process data fast enough to allow for small inter-sensor spacing. Thus, the prototype needed to employ a large, expensive computer to accomplish the processing tasks within the requisite amount of time. Additionally, due to sampling frequency restraints, Van Lancker's work focused on increasing inter-sensor spacing as much as possible to improve precision, which essentially sacrificed ease of deployment for the sake of functionality. The improvements in modern microprocessors have rendered such decisions unnecessary. With ADC sample rates currently reaching into the megahertz range for inexpensive processors and well above that for FPGAs, the only limitation current hardware places on goniometry development is the speed at which the data can be processed and stored. Thus, taking advantage of modern technology can improve the ease with which researchers deploy acoustic goniometers while maintaining the same level of accuracy as previous systems. One secondary goal of the current research is to prove the validity of this statement by developing such a system.



(a) ARFANG Goniometer with Central Computer and Single Microphone Highlighted [13-15]



(b) Echo Star Pattern [13]

(c) Regular Tetrahedral Pattern [13]

Figure 2.3 Van Lancker's Acoustic Goniometer

Scott et al. studied and compared the use of single sound sensors versus multiple sensor arrays in the detection of avalanches [16-17]. The results of their study show that single sensors can be used to identify avalanches and differentiate them from other phenomena. However, such systems suffer in the presence of wind, which can lead to both false positive and negative readings. Multi-sensor arrays, in addition to providing the ability to determine the point of origin/location of an avalanche, were also found to be

far more robust to wind noise. Sensor data fusion allows the multi-sensor array systems to check data from one sensor against others in order to determine the occurrence of erroneous positive or negative readings. Although this does not eliminate false avalanche identification and is not the only means to combat wind noise, using multiple sensors does significantly reduce the effects of wind noise as well as other sources of potential error [16-17]. The research of Scott et al. provided valuable information with regards to developing an acoustic goniometer. However, their goal was never to develop a general purpose goniometer but to study a particular aspect of the design process. The work developed as part of this dissertation benefited from their research but is more ambitious in its scope.

The authors of [35] further explored the work presented by Scott et al. in [16-17] by deploying their own infrasound sensor arrays in a section of the Italian Alps where explosions are regularly used to trigger avalanches. In their work, Ulivieri et al. used a 4 sensor array with an inter-sensor spacing of 150 m. The design incorporated 3 pressure transducers and 1 infrasonic microphone in a triangular configuration with the microphone at the center (see Figure 2.4). This was a wired network of sensors connected to a centrally located computer to handle data processing and equipped with GPS antennas to synchronize time. All of the sensors were covered with aluminum boxes and buried in the snow in an effort to combat wind noise. The researchers were able to record several explosion triggered avalanches and apply goniometer theory to determine the direction of arrival for each sound event. The research group concluded that the work was a success and confirmed the feasibility of using infrasonic arrays to monitor avalanches stating that infrasonic goniometry is a very efficient means monitoring avalanches and

providing insight as to how avalanches can be better controlled. They further determined that better instrument design, array installation, environmental noise handling, and data processing were needed to further research in this field [35]. Thus, the authors of [35] seem to be indicating that the research contained in the current dissertation is precisely the work needed to further exploration in this field. One of the key difficulties they encountered was the deployment of the sensor system. The miniaturized (as compared to their 150 m spacing) prototype developed as part of the current dissertation research would provide a much simpler deployment process. Additionally, the prototype would make the real-time analysis of the data discussed by the authors much simpler due to the lower power requirements of an embedded platform as compared to the computers generally used to conduct research of this nature.

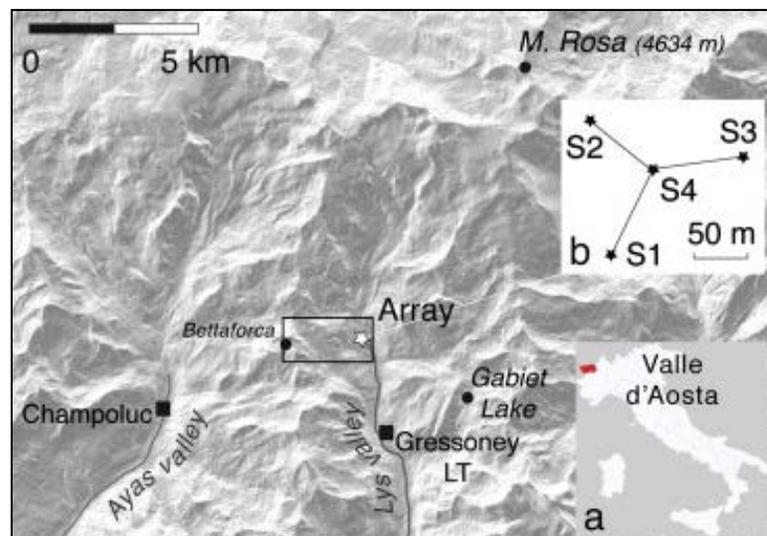


Figure 2.4 Goniometer Deployed by Olivieri et al. [35]

Some researchers focused more on the design of sensors targeted toward recording infrasound events. In his master's thesis [18], Kinnerup developed a specialized sensor for measuring infrasound between 10 mHz and 20 Hz in order to

characterize the infrasonic output from wind turbines. His research involved the use of a condenser microphone specifically designed to be sensitive in the desired frequency range connected to an amplifier circuit housed in a small custom case. Kinnerup tested and reviewed multiple circuit configurations and designs for analog filters as well as microphone vent configurations. Similarly to [16-17], Kinnerup's work focused on a single element of goniometer/audio sensing equipment as opposed to the design of an entire system. While his design is more expensive than using a simple condenser microphone and amplification, his sensor could be a useful addition to the prototype developed for this dissertation. Since the sensors in this dissertation's prototype are separate from the processing board (see Chapter 5.2), new or custom sensors can easily be swapped for the existing ones. This could be useful for applications requiring more accurate microphone selectivity or for conducting research similar to that of [18] for developing new/improved sensor designs (e.g., microphones, accelerometers, specialized hardware, etc.).

Another example of sensor development can be found in the work of Sugimoto et al. The authors of [19] created a custom infrasound sensor for monitoring the emissions from wind farms. The system used the same style of condenser microphone found in Kinnerup's research but added a sealed chamber and windscreen to improve performance. While the work did include a design for an infrasound measurement system, the focus of Sugimoto et al. appeared to be on wind reduction and improvements to previous designs. Although neither research project dealt with acoustic goniometry, their thorough analysis of multiple designs, tradeoff considerations, and noise reduction methods provided useful insight into the design of specialized infrasound sensors. As

with [18], the work of Sugimoto et al. could be easily integrated into the prototype developed for this dissertation providing a more targeted version of an acoustic goniometer for specific applications or supplying a generic test platform for further field testing and development of specialized acoustic sensors.

Similarly to the work of Kinnerup, the authors of [20] present a generic, robust system for the acquisition of infrasound data. However, their design uses a low-powered DSP, a 4 channel ADC, and 4 inexpensive electret microphones. The authors appear to have built on some of the previously discussed research completed by other scientists in order to create their device (see [4], [6], and [15]). The microphones are similar to those used in many of the previously mentioned research projects in that they do not have a guaranteed flat frequency response in the infrasound range. Additionally, their algorithm uses the same data fusion techniques described in [17]. This project had many goals in common with the research conducted for this dissertation but focused only on recording infrasound and detecting events. The research in [20] did not involve any localization of detected events (goniometry) but could have benefited from the acoustic goniometer research conducted in this dissertation research in much the same way as the work of [18] and [19].

Research has also been completed to explore technologies beyond accelerometers and microphones for infrasonic monitoring. The authors of [21] discussed the usefulness of quartz crystal resonators for use in infrasound detection. The work focuses on the use of nano-resolution sensors for monitoring oceanic and seismic events (e.g., tsunami and earthquake warning systems). The sensors have been proven useful in tracking aircraft, detecting seismic events, and predicting tsunamis. Although this work shows great

promise, the application of the technology is still relatively new. Consequently, using this technology is still difficult and expensive relative to accelerometers and microphones.

Once the technology is more commercially available, these quartz sensors could be tested on the generic platform for acoustic goniometer research developed as part of this dissertation research for localization of events or simple recording/event detection.

Exploring along a similar vein to the previously mentioned research focusing on sensor development, Marcillo et al. focused their research on the development of an inexpensive small infrasonic sensor [34]. Their work used a pressure transducer enclosed in a special mechanical filter. While pressure transducers are commonly used in acoustic sensors, the novelty of their work comes from the use of a capillary tube and backing volume as a mechanical filter for infrasound recording. As they explain in [34], a capillary tube and volume element/chamber is analogous to a resistor and capacitor (respectively) in a circuit. By combining these elements as shown in Figure 2.5, the authors effectively created a simple low pass filter where the radius and length of the capillary determine the resistance, and the dimensions (volume) of the chamber (backing volume) determines the capacitance. After testing the design, they determined the mechanical filter to be quite good at rejecting high frequency noise produced by wind (or other phenomena) with a reduction from 23 Pa to 2.02 mPa (rms) for the 0.5-2 Hz frequency range. The sensor they designed was likewise deemed a success as a low cost option for infrasonic monitoring with a flat frequency response (within 3 dB) throughout the infrasonic range [34]. The work Marcillo's group provides another option for infrasonic monitoring of phenomena and another potential sensor option for an acoustic goniometer. With regards to the work in the current dissertation, Marcillo's design could

be incorporated into the current prototype with minor adjustments. Doing so would provide a test platform for furthering research and testing goals in both the general purpose acoustic goniometer and infrasound sensor development fields. Exploration of using the mechanical filter would be especially helpful for combating wind issues in the current prototype (see Chapter 5.2.3). Marcillo's research would benefit from having an easily deployable goniometer antenna for further field testing and data collection.

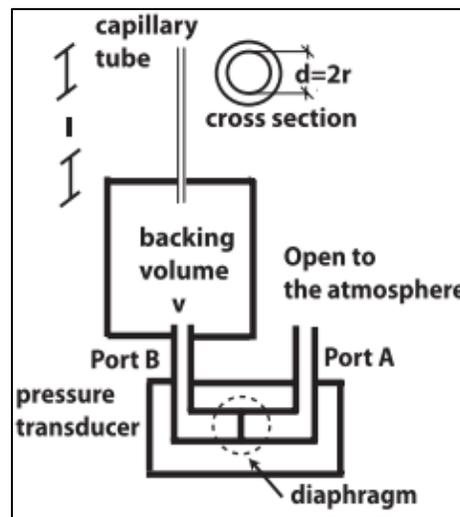


Figure 2.5 Low Pass Mechanical Filter Designed by Marcillo et al. [34]

Kogelnig et al. presented a solution for monitoring avalanches using infrasound sensors in conjunction with seismic sensors. While many researchers have weighed the benefits of each sensor technology, the work in [36] explored the use of sensor data fusion in order to combine the benefits of both types and make a better sensor system. The researchers found that infrasound sensors were better suited for monitoring aerosol fluctuations, and seismic sensors were more useful for detecting dense flow vibrations. For this reason, infrasound sensors were better at detecting the start of an avalanche while seismic sensors provided a more accurate indication of their termination. Thus, using both sensor technologies allowed them to more accurately record avalanche duration.

Additionally, the use of sensor data fusion allowed them to differentiate between the various flow regimes and more accurately characterize the type of avalanche detected [36]. While the work done by Kogelnig is outside the scope of the research presented in this dissertation, the research from [36] would clearly benefit from further study, which could only be eased by the goniometer designed for the current work. Although the sensors used for their research are not microphones (specialized infrasound and seismic sensors), the interface used for data collection on the current goniometer (see Chapter 5.2) is generic enough to sample from a diverse sensor suite.

2.3 Algorithm Research

In addition to researching applications for infrasound detection and the improvement of infrasound hardware, some researchers have focused on improving the algorithms associated with acoustic goniometry. The most applicable work explores filtering, event detection, event classification, and source localization. The following details some of the more relevant research.

Dickey and Mikhael used sensor data fusion to improve noise filtering in their research on infrasound monitoring of earthquakes [22]. They proposed a method of adaptive noise cancellation in which seismometers are used to characterize the response of infrasonic sensors to seismic activity. The data from the seismometers allows them to generate adaptive filters in order to better target the use of infrasonic sensors toward monitoring seismic activity. As part of their research, the team compared two algorithms for updating the adaptive algorithm coefficients: least mean squares (LMS) and block LMS (BLMS). The results of the comparison showed both algorithms to work well, but BLMS significantly outperformed LMS in the areas of accuracy and convergence rate.

Thus, even with the increased computational complexity, BLMS was selected as the better option. Sensor data fusion was not performed for the research conducted as part of this dissertation since the focus was more on a general purpose system and not on application specific algorithm development. However, the research in [22] indicates that sensor data fusion could be a valuable tool in the goniometry process. Using multiple microphones on each acoustic sensor of the goniometer could provide a means of data verification and/or noise reduction. The prototype developed as part of this dissertation research was designed to further this form of algorithm development (see Chapter 5.2; each sensor board has 2 microphone connections). Sensor data fusion is a vast field of study with the potential to offer much to the acoustic recording/localization process. As with other areas in need of further exploration in the field of acoustic goniometer research, sensor data fusion research in this area could benefit greatly from the general purpose platform detailed in subsequent chapters.

Bedard et al. focused their research on the collection of noise reduction schemes for infrasound sensors. These researchers gathered information on the various techniques used by other groups to combat the presence of unwanted noise sources in infrasound monitoring applications [23]. Their results show an explanation and comparison of the various techniques and detail the collection of data sets for the purpose of providing better training data for such research. While not an original work on developing algorithms nor comprehensive in its explanations, their research provides a valuable source of information on noise reduction techniques and pulls together information into a more accessible format. The research completed as part of this dissertation used the work

of [23] as a starting point for researching some of the algorithms associated with the field of acoustic recording.

Aside from noise reduction, some work has been focused on more elegant methods of detail extraction. The authors of [24] presented an FPGA implementation of a one dimensional discrete wavelet transform (1-D DWT) for infrasound data processing. According to their work, the 1-D DWT handles signal decomposition more efficiently than the FFT. Furthermore, they assert that the algorithm is capable of extracting enhanced signal information from infrasonic data in real-time. The results of their work prove the feasibility of implementing infrasound processing algorithms on an FPGA. While this algorithm has not been implemented on the prototype hardware developed for this dissertation, the 1-D DWT method is an excellent candidate for further exploration. The computational complexity would make it difficult to incorporate into a small antenna configuration with a fast sample rate using the current processor. However, the prototype could be reconfigured (larger size and slower sample rate) to further field test the algorithm and test the claim of real-time processing on simple embedded hardware.

Wang et al. built on top of the work detailed in [24]. Their research veered from the standard FFT and wavelet-based processing techniques usually employed for infrasonic signals by opting instead for a fuzzy logic approach to event classification [25]. Specifically, the authors apply Fourier and wavelet-based algorithms for feature extraction and fuzzy K-means clustering to classify earthquake events. The goal of their research was to improve infrasonic event detection and possibly provide another means of predicting earthquakes. The authors concluded that their research was successful and the algorithm showed significant promise. To further test the algorithm and viability for

further use in the field, the research in [25] could benefit from the research presented in this dissertation. As with [24], the computational complexity would demand either a slower sample rate and consequently a larger inter-sensor spacing or a more capable processor/FPGA. However, the research presented in this dissertation proves the feasibility of a generic acoustic goniometer and provides a good starting point for such an implementation.

Another point of interest related to the research of Wang et al. is the differentiation of events. Ham et al. used three neural network classifiers in their work on infrasound event classification: multi-layer feedforward perceptron trained by backpropagation, a radial basis-function network, and a partial least-squares calibration model [26]. The experimental results showed a 100% success rate for the multi-layer feedforward perceptron algorithm and slightly higher than 90% success for the other two classifiers at discerning between their 4 possible events: volcano, mountain associated waves, impulsive, and background noise (no event). The research in [26] is yet another example of work that could benefit from the availability of an easily deployable, general purpose acoustic goniometer. The prototype created as part of this dissertation work could be used to test the algorithm presented by Ham et al. and further this area of algorithm research.

Chilo et al. also focused their research in [27] on fingerprinting different events. Their work discusses the characterization of background noise, covers the features of various events, and suggests the possible use of neural networks for adaptive fingerprinting (specifically the feedforward with backpropagation algorithm described in [26]). Their work showed promise but required more research in order to make definitive

conclusions. Similarly, Ham et al. also explored the use of neural networks for infrasound event detection in [28]. In this work, the authors used parallel neural network banks to classify surf events from three different locations. They were able to achieve a success rate of 87.1% with their limited data set and claimed this to be a significant improvement over more conventional classification methods. Again, more testing was deemed desirable to prove the superiority of the algorithm more definitively. Testing on more hardware platforms and in more situations involving a more diverse set of phenomena and noise environments would greatly improve the process. In order to achieve such ambitious goals, general purpose acoustic goniometers would need to be deployed by multiple research groups. The research completed as part of this dissertation is a step in realizing this goal as it provides a means to lower the barriers to research and further explore this and other algorithms in the field of acoustic goniometry.

The authors of [29] propose using quadratic phase coupling (QPC) analysis instead of power spectral density analysis (PSD) for detecting vehicles via infrasonic monitoring. According to their research, QPC uses phase relations to better characterize infrasonic emissions from engines, thus providing better vehicle detection. They further compare the use of microphones to specialized infrasound sensors. The results of this portion of their research further showed that infrasound sensors tend to perform better over the entire infrasound range than microphones regardless of which method of analysis was employed. However, the tradeoff to using infrasound sensors is a dramatic increase in cost. Further study with a generic platform could prove useful in furthering this area of algorithm research. Since the focus of this research was on improving event detection for a very specific event (vehicle engines), research in this particular algorithm

falls outside the scope of algorithm development/testing for a general purpose acoustic goniometer. However, the research of [29] still provides another example of work being done in the area of acoustic goniometry, which could benefit from the research conducted as part of this dissertation.

Similarly to [16-17] and [20], the authors of [30] use infrasound as part of a data fusion process for event detection and source localization. However, in the case of [30], photo sensors are used in addition to the acoustic sensors to detect and determine the location of lightning. The authors created and tested a system that uses the flash of light from lightning in conjunction with the infrasonic emission of thunder to detect events. The infrasound detection system uses a GPS module for timestamping and sensor location to perform acoustic goniometry to locate the source of the lightning event. Their publication seems to suggest that the system detects events in the embedded hardware and outsources the complex localization to a remote computer (via some form of network) running Matlab and Labview. The research of [30] provides an excellent example of the type of work which would benefit most from the research presented in this dissertation. The acoustic goniometer used in [30] is large enough to warrant the use of GPS modules to accurately record the sensor positions. Furthermore, goniometry data has to be transmitted to a more powerful processing system (i.e., the remote computer) increasing the cost and complexity of the goniometer. This means the system is difficult to deploy, costly, and more complex than is necessary. The inclusion of cameras in the sensor data fusion process may be the reason a more powerful processing platform is required (depending on the cameras, algorithms, and microprocessors being used by the system). However, since the light flashes are used to verify or fine tune the goniometer

DOA calculations, the acoustic and photograph data processing can easily be separated. The prototype developed for the research in this dissertation could handle all acoustic calculations. Then, depending on the types of algorithms and cameras used, the photograph data could either be handled by the second processor of the acoustic goniometer (see Chapter 5.2) or a computer. The GPS module could be discarded, and the acoustic goniometer could be easily deployed and reconfigured to further the goals of the research. With natural phenomena like lightning, the ability to move the sensor structure closer to the source or to areas that experience the phenomena make the testing process easier thus lowering the barriers to research.

A more recent project completed by Havens et al. explored the use of an infrasound data collection array to determine not only the origin but also the velocity of an avalanche. The authors recorded the raw data from an avalanche event with an array consisting of 3 microphones in a triangular geometry with 30m inter-sensor spacing using a 24-bit ADC sampling at 100 Hz. They analyzed the data by employing the Fisher Statistic approach to event detection (see Chapter 3.5) and successfully calculated the velocity of the event to be 35.9 m/s (± 7.6 m/s). This work expanded upon knowledge in this field of research with two important breakthroughs. The Havens et al. group was the first to track the velocity time series of an avalanche from start to finish using an infrasound array positioned in an optimal placement. Furthermore, they were the first to capture a weak layer failure at the start of an avalanche using infrasound sensors [33]. The work done in [33] explored an interesting new approach to event detection and shows the potential utility of the research performed for this dissertation. Part of the reason the Havens et al. research group was the first to track the velocity and capture the

weak layer failure with infrasound sensors is that naturally occurring avalanches cannot be predicted accurately or quickly enough to facilitate optimal sensor placement for conventional goniometer arrays. Work of this nature (as well as similar fields of study) would benefit from an easy to deploy, inexpensive acoustic goniometer that could analyze data in real-time or record it for future analysis. Such a system could be deployed quickly and at multiple locations within an area known to have avalanches (e.g., the same location in Idaho described in the paper by Havens et al.). Then, the likelihood of collecting similar data with the infrasound goniometer antennas in optimal position would be far greater, thus furthering research in this field.

CHAPTER THREE: ACOUSTIC GONIOMETER THEORY

Since goniometry is derived from the combinations of two Greek words meaning “angle” and “measure,” the definition of goniometer as “a device which measures angles” is fairly obvious. An acoustic goniometer, therefore, is a device that measures angles using sound. Such a device usually includes 2 or more sensors capable of detecting sound spaced apart at a known distance. Each sensor is equipped with a synchronized clock, allowing the goniometer to detect the differences in the time at which a sound event is detected by each sensor. This information is used in conjunction with the sensor spacing and the constant speed of sound to determine the differences in distance between the sound event source and each sensor. With the aid of basic trigonometry, this result can be used to determine the direction of arrival (DOA) for the sound event. As mentioned in Chapter 1, this process of acoustic goniometry is used regularly by the average person and is usually taken for granted. However, similarly to other abilities with which humans (and many animals) are innately endowed, the process is challenging to implement in an embedded or even computer algorithm. This chapter explains the process used to perform acoustic goniometry in detail. The derivations and explanations found in this chapter are not an original work but were gleaned mainly from [13] and [31].

3.1 Sound Propagation

In order to explain the process of acoustic goniometry, a basic explanation of sound propagation and the factors that affect the transmission of a sound is helpful. Sound can be described as a mechanical wave transmitted through a medium (e.g., air,

ground, water). Vibrations from the source of a sound interact with molecules in the surrounding medium. These molecules interact with the molecules nearest to them, transmitting the sound through the medium. As the wave travels, its energy is absorbed by the medium and other objects with which the wave interacts until the sound is completely attenuated [31].

The speed and distance traveled by a sound wave is dependent upon several factors including its amplitude, its frequency, and the medium through which it travels. Stokes' law of sound attenuation (shown in Eq. 3.1) shows the specific relationship. Sound waves with higher frequencies or lower amplitude are attenuated faster than their lower frequency or higher energy counterparts. Consequently, lower frequency sounds tend to travel farther. Denser mediums will result in slower sound transmission as well as attenuate the wave at a slower rate. The effect of temperature on the speed of sound in a gas can be seen in Equation 3.2. All other things being equal, a medium with a higher temperature will allow sound to travel faster [31]. In order to accurately calculate the DOA of a sound wave, these factors must be considered for each goniometer application. Understanding how fast and far a sound will travel in the environment of interest will directly affect the accuracy of a goniometer. A system designed for deployment in a desert would be less accurate if deployed to monitor avalanches unless the change in the speed of sound were taken into account. Furthermore, a system deployed to measure sounds with higher frequency content may need to be placed closer to potential sound sources than one used to monitor infrasound activity. Depending on the accuracy requirements for a given project, making adjustments to a sensor system may be unnecessary if the effect is sufficiently small. However, the ability to adapt in the event

such changes in accuracy are significant to the given research project is necessary for a truly flexible acoustic goniometer implementation.

$$\alpha = \frac{2\eta\omega^2}{3\rho V^3} \quad \text{Eq. 3.1}$$

Where α is the rate of attenuation, η is the dynamic viscosity coefficient of the medium, ρ is the density of the medium, V is the speed of sound in the medium, and ω is the frequency of the wave.

$$c = \sqrt{\gamma r T} \quad \text{Eq. 3.2}$$

Where c is the speed of sound, γ is the specific heat (heat required to raise the temperature 1°C), r is the gas constant, and T is the temperature.

In the simplest scenario for a sound event, sound generated in open air radiates outward from the source equally in all directions. Further, if the receiver (e.g., person, sensor, etc.) is far enough from the source, a sphere or perfect point may be used to adequately model the source. Certain factors or artificially created sound can result in a directional sound wave, and spherical/point models of a source fail if the receiver is close to or contained within the source of the sound event. However, since these are special cases not usually encountered for most goniometer applications, they can be ignored for the sake of keeping the following explanation simple. As a sound wave moves away from its source, it will encounter objects with varying abilities to transmit the wave. When this happens, some of the sound will continue through the objects, but part of the wave will be reflected off the object surfaces (as shown in Figure 3.1). Any medium whose properties (density, temperature, etc.) differ from the air through which the sound is traveling will

cause this behavior (e.g., ground, trees, buildings, upper atmosphere, etc.). The reflected waves act exactly like the sound from the originating event and are capable of being reflected as well. Figure 3.1 shows this as an “imaginary source” (IS), because a goniometer analyzing the sound from this reflection would determine the DOA based on the object’s location rather than that of the source (S).

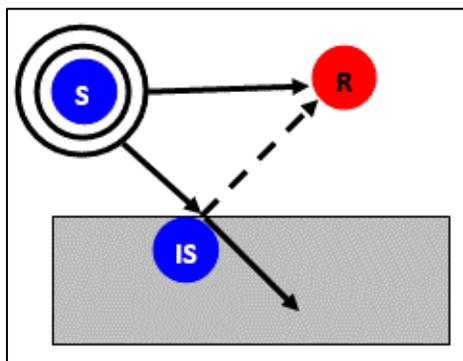


Figure 3.1 Imaginary Sources Produced by Reflections when a Sound Wave Encounters an Object

Building from the previous model and discussion of Figure 3.1, the problem presented by reflections rapidly becomes apparent. Since no environment outside of a simulation is without obstacles or at least a ground and upper atmosphere, any sound traveling to a sensor in a real-world environment most likely will arrive along with reflections of itself. These reflections can cause constructive or destructive interference depending on the path taken between the source and the receiver. If the objects causing reflections are close enough to the source or receiver, the magnitude of their effect is greater. Objects causing reflections close enough to the source may be ignored as coming approximately from the source itself, but those located closer to the receiver can create reflections that greatly impact the original sound wave. The phenomena where multiple paths are taken by a wave to reach a receiver is commonly referred to as multipath.

Figure 3.2 shows a near-worst case scenario of multipath with a sound source and receiver in a fully enclosed environment. The solid lines in the figure represent the original sound wave traveling from the source, and the circles around the source reference the idea that sound in the current model radiates outward from the source in all directions. While only three potential reflection points are shown (blue dots representing imaginary sources), the fact that this environment presents many sources of reflection should be noted. Also, only first order reflections are shown in the figure. Other paths for sound transmission exist that may require multiple reflection points to reach the receiver, and this is true for most real-world environments [31].

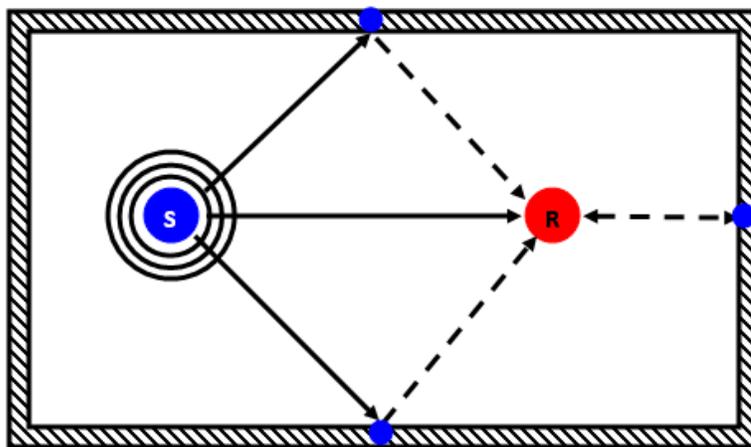


Figure 3.2 Sound Reflections and Multipath in an Enclosed Space

Multipath creates a challenging problem for an acoustic goniometer. Since a goniometer uses multiple sensors at different locations to perform its operations, the reflected sound waves arriving at one sensor may be different than those arriving at another. The waves traveling along the straightest, least obstructed path will arrive soonest and have the greatest amplitude since they traveled the shortest distance and encountered fewer obstacles [31]. The result in a very poor environment could be a set of signals for a sound event that cannot be correlated to one another, which would make the

goniometer calculations impossible (a practical example of this is explored in Chapter 4.2; Figure 4.4). As the path between the event source and the sensors becomes more obstructed or the goniometer sensors are placed farther apart, the likelihood of reflections causing problems increases drastically. In order to work in a practical environment, an acoustic goniometer needs to be capable of handling a reasonable amount of multipath.

3.2 Goniometer Theory of Operation

Consider the simple acoustic sensor system as shown in Figure 3.3 where the two sensors (S_1 and S_2) are represented by red dots spaced apart by some distance (d). Now, suppose an event occurs producing sound at the location indicated by the blue dot (S_0). For ease of explanation, assume the path between the sensors and the sound source is unobstructed. Further, given a large enough distance between the source and the sensors (represented by d_{s1} and d_{s2}), note that even a sound source with multiple related origins (e.g., an avalanche) can be modeled as a perfect point source whose sound radiates outward equally in all directions. The only remaining difficulty with this model is the presence of reflected signals (see r_1 and r_2 in Figure 3.3). Reflections occur when a sound event reverberates off an object along its path and can make the process of event detection more difficult by masking the fingerprint of a given event. The practical ramifications of this problem will be discussed in greater detail in Chapter 4. For the sake of explaining goniometry by itself, assume reflections are not present in the current model (exclude r_1 and r_2).

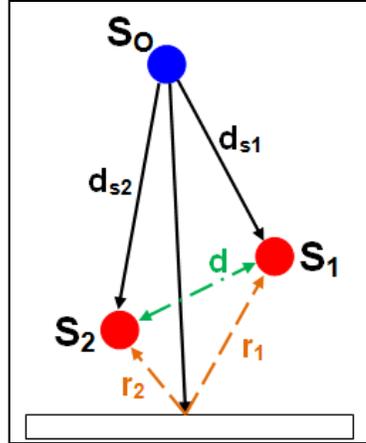


Figure 3.3 Acoustic Goniometer Sensor Pair Diagram

Under the conditions given above, the time required to travel from S_0 to S_1 can easily be calculated by the casual observer using Equation 3.3 where c is the constant speed of sound. The time required for the sound to travel from the source to the second sensor can be defined in a similar fashion. However, since the goniometer is trying to calculate the direction from which the sound event originated, it obviously has no knowledge of d_{s1} or d_{s2} . Equation 3.3 can be modified instead to give the relationship between the difference in the distances between the sensors and the source (Δd_{12}) and difference in propagation times (Δt_{12}) from the source to the sensors (as shown in Equation 3.4).

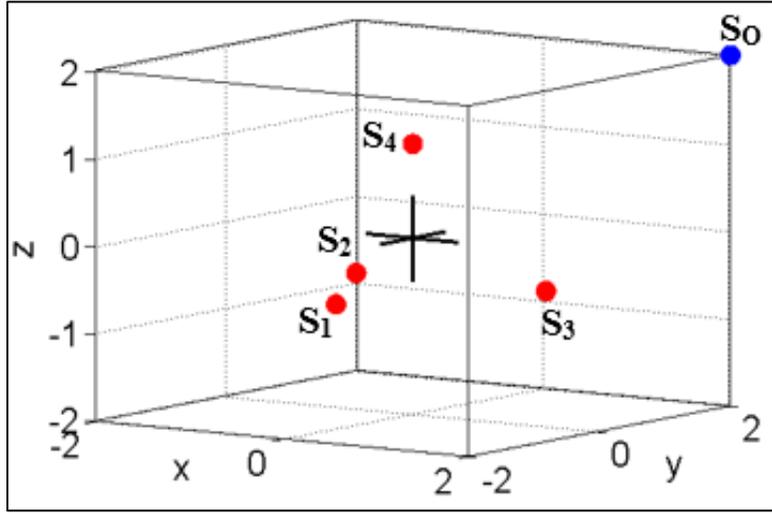
$$t_1 = \frac{d_{s1}}{c} \quad \text{Eq. 3.3}$$

Where t_1 is the time taken for the sound wave to reach S_1 , d_{s1} is the distance from the source to S_1 , and c is the speed of sound.

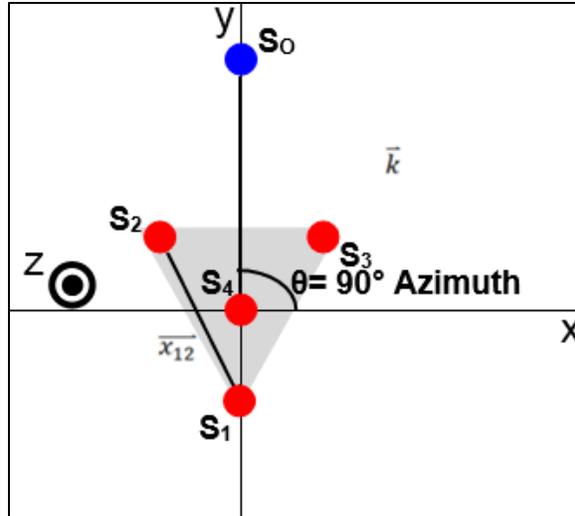
$$\Delta t_{12} = t_2 - t_1 = \frac{d_{s2} - d_{s1}}{c} = \frac{\Delta d_{12}}{c} \quad \text{Eq. 3.4}$$

Where t_2 is the time taken for the sound wave to reach S_2 , and d_{s2} is the distance from the source to S_2 .

In order to continue with the derivation, a more complex goniometer model with a coordinate system must be considered (see Figure 3.4). The new model shows a goniometer with 4 sensors where S_4 is located along the positive z-axis, and the other 3 sensors are located in a plane parallel to the one defined by the x and y coordinate axes in the negative z-axis. The sensors are arranged such that the center of the goniometer is located at the origin. The wave vector (\vec{k}) is defined as the sound event's path of travel from the source to the center of the goniometer. Thus, the vector defining the path from S_1 to S_2 (\vec{x}_{12}) is related to the normalized wave vector (\vec{n} : Equation 3.5) by Equation 3.6. Although the time required to travel from the source to the respective sensors (t_1 and t_2) is not known by the goniometer, this difference can be obtained by simply using the difference between each sensor's timestamp of the sound event.



(a) Sensor Orientation 3D View (origin marked with )



(b) Top-Down View with Vectors

Figure 3.4 Acoustic Goniometer Vector Diagram

$$\vec{n} = \frac{\vec{k}}{|\vec{k}|} \quad \text{Eq. 3.5}$$

Where \vec{k} is the sound wave's path of travel from the source to the center of the goniometer, and \vec{n} is the normalized vector.

$$\Delta t_{12} = \frac{\vec{n} \cdot (\vec{x}_2 - \vec{x}_1)}{c} = \frac{\vec{n} \cdot \vec{x}_{12}}{c} \quad \text{Eq. 3.6}$$

Where Δt_{12} is the difference in arrival times at the sensors, x_1 and x_2 represent the positions of S_1 and S_2 (respectively), and c is the speed of sound.

Now, if Equation 3.6 is used for each of the sensor pairs, a system of equations may be defined to solve for \vec{n} (see Equation 3.7). In addition to giving the distances between sensors pairs, the D matrix also determines the coordinate system used by the acoustic goniometer. Since D is a matrix of vectors rather than magnitudes, the direction

(polarity) of the vectors determine the coordinate axes. In order to solve Equation 3.7 for \vec{n} , the inverse of the matrix D must be calculated. This can be done simply if D is invertible. However, this only occurs if three sensor pairs are used (assuming the calculations are taking place in 3-D space). Otherwise, a pseudo-inverse of D must be calculated using Singular Value Decomposition (SVD). Equation 3.8 shows the inverse calculation and the final solution to solving Equation 3.7 for \vec{n} . In Equation 3.8, D_p represents the pseudo-inverse of D .

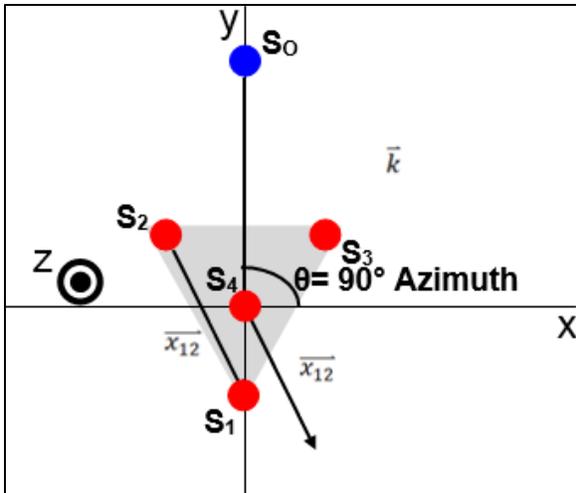
$$T = \frac{\vec{n} \cdot D}{c} \quad \text{Eq. 3.7}$$

Where D is a matrix containing the vectors between sensor pairs, and T is a matrix containing the time delays between sensor pairs

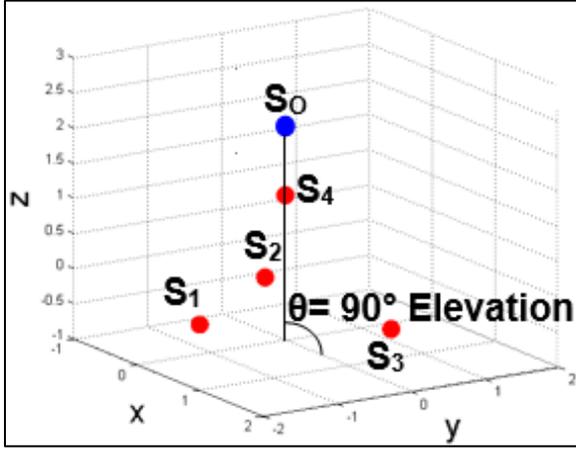
$$\vec{n} = c \cdot (D^t \cdot D)^{-1} \cdot D^t \cdot T = c \cdot D_p \cdot T \quad \text{Eq. 3.8}$$

Where D^t is the transpose of D , and D_p is the pseudo-inverse of D .

Once the normalized wave vector is calculated, the DOA azimuth (A) and elevation (E) can be determined using Equation 3.9 and Equation 3.10, respectively. The equation for azimuth provides the angle with respect to the the x-axis of the goniometer's coordinate system. Thus, a sound wave traveling along the y-axis would correspond to an azimuth of 90° (see Figure 3.5 a). The elevation equation provides an angle with respect to the y-axis of the goniometer's coordinate system. Therefore, a sound wave traveling directly down the z-axis would correspond to an elevation of 90° (see Figure 3.5 b). The choice for these conventions can change depending on the needs of a given project and can be adjusted easily with minor modifications to Equation 3.9 and 3.10.



(a) Source Location at 90° Azimuth 0° Elevation



(b) Source Location at 0° Azimuth 90° Elevation

Figure 3.5 Example Source Locations Explaining Azimuth and Elevation

$$A = 90^\circ - \tan^{-1} \frac{n_x}{n_y} \tag{Eq. 3.9}$$

Where A is azimuth, and n_x/n_y are the first and second elements of the normalized wave vector.

$$E = -\tan^{-1} \frac{n_z}{\sqrt{n_x^2 + n_y^2}} \quad \text{Eq. 3.10}$$

Where E is elevation, and n_z is the third element of the normalized wave vector.

The importance of antenna geometry to acoustic goniometry is made fairly clear in the work of Van Lanker. In his dissertation, Van Lanker explored the use of various geometries and made recommendations for particular geometries depending on the type of phenomena and associated frequency being monitored [13]. While the goal of the current dissertation was not to find the best geometry for any particular application, providing some information on constraints of selecting a geometry is relevant to a theory of acoustic goniometer operation. A simple analysis of Equations 3.8-3.10 quickly yields the most important constraint on selecting an antenna geometry. A goniometer attempting to locate the source of a sound occurring in 3-D space must have an antenna geometry, which occupies 3-D space. In other words, an antenna that has no sensors whose positions vary in the vertical axis will not be capable of calculating the elevation for a DOA vector. Similarly, a geometry that resides completely in a plane of either the x-axis or y-axis would be unable to calculate the azimuth for a DOA vector. To understand this, consider Equation 3.10. If the z-axis term of the normalized vector \vec{n} is zero, the calculated elevation will likewise be zero. One way this is guaranteed to always occur is for the differences in z-axis distances between the sensor pairs to be zeros (all sensors in the same z-plane). Thus, if all of the sensors of the goniometer antenna are aligned in the z-axis, the calculated elevation from the goniometer will always be zero. A similar

analysis holds true with regard to the calculated azimuth for Equation 3.9 and an antenna whose sensors are all aligned in either a common x-plane or y-plane.

3.3 Sampling Theory

In theory, the equations discussed previously in Section 3.2 provide perfect accuracy. However, the key limitations of an actual implementation of the acoustic goniometer are the finite precision math imposed by a microprocessor and finite precision time. A truly continuous time analysis of audio signals is not possible for real hardware. Thus, sampling techniques must be employed to create a discrete time representation of the signal. Since the accuracy of the acoustic goniometer hinges on the ability to determine differences in TOA for sensor pairs, sampling theory in the spatial and time domains is a requisite for designing and using an acoustic goniometer.

3.3.1 Spatial Sampling

The determination of inter-sensor spacing for an acoustic goniometer is determined in part by the wavelength of the sound produced by the phenomena being monitored. According to the Nyquist-Shannon sampling theorem, the condition expressed in Equation 3.11 must be met in order to minimize error in the correlation process caused by spatial aliasing [13]. Since the worst case scenario according to Equation 3.11 is the one in which the sound wave approaches at 90° with respect to the sensors, the condition can be simplified to state that the inter sensors spacing must be less than half the wavelength of the expected sound (see Equation 3.12).

$$d < \frac{\lambda}{1 + \sin \theta} \quad \text{Eq. 3.11}$$

Where d represents the inter-sensor spacing, λ represents the wavelength of the sound wave, and θ represents the angle at which the sound wave approaches the sensors.

$$d < \frac{\lambda}{2}, \{for \theta = 90^\circ\} \quad \text{Eq. 3.12}$$

The other factor used to determine inter-sensor spacing is the sampling speed of the acoustic goniometer. The minimum inter-sensor spacing is dependent on how fast the hardware can sample and process data. The maximum difference in the arrival time (Δt) of a sound wave between two given sensors is directly proportional to the space between them (d) and can be seen in the inequality expressed in Equation 3.13. It follows logically from a rearrangement of the terms (see Equation 3.14) how the minimum distance between sensors could be determined. However, the value calculated as the minimum in Equation 3.14 can only be practically identified as the theoretical minimum insofar as using this minimum as the actual inter-sensor spacing would severely limit the accuracy of the acoustic goniometer as to make it unusable in almost all research situations. Since the maximum difference in the arrival time was used as the starting criteria, choosing the minimum calculated Equation 3.14 would allow at most only one sample period of difference between the TOAs of the two sensors. Thus, the sensor pair would act as a binary sensor indicating only whether the DOA was one particular direction: directly along the line between the two sensors. Increasing the distance above the minimum improves the resolution of the system to determine the DOA. Thus, determination of the

actual distance needed between sensors would depend on the requirements of the system and careful consideration of the design tradeoffs. Experimental results illustrating this process can be found in Chapter 4.

$$\Delta t < \frac{d}{c} \quad \text{Eq. 3.13}$$

Where Δt is the difference in the arrival time, d is the distance between sensors, and c is the speed of sound.

$$d > c \cdot \Delta t \quad \text{Eq. 3.14}$$

3.3.2 Temporal Sampling

The temporal sampling process can be split into two main parts. The first is the speed at which the system samples. The second is the number of samples used for data analysis or the length of a data event window. Both aspects of sampling greatly affect the operation and accuracy of the goniometer.

3.3.2.1 Sampling Frequency

Previous discussions have made clear the importance of the sample rate for an acoustic goniometer. This one feature of the system affects many aspects of the sensor's operation and setup. According to the Nyquist-Shannon sampling theorem, the minimum sampling frequency required to analyze the sound wave is twice the maximum frequency of interest. However, as with the discussion on inter-sensor spacing in Section 3.3.1, the theoretical minimum sampling frequency is hardly adequate for a practical implementation. The Nyquist-Shannon sampling theorem applies to the recreation of a signal from a minimum number of samples. In theory, if the sound wave's signal from a

sensor could be recreated, this would allow for the use a correlation technique to determine the time delay between it and the signal from another sensor. However, using this approach would be challenging in an embedded system responsible for monitoring events in real-time as it would require the use of more advanced filtering methods to recover the signal and still not provide a perfectly accurate measure of the arrival time. Since the acoustic goniometer is expected to detect randomly occurring events and then analyze them for a DOA, faster sampling improves the resolution of the goniometer's TOA determination and can directly affect the accuracy of the system. As a simple example, consider a system with a 1 Hz sample rate where the actual difference in arrival time between two sensors is 10 ms. The calculated difference in TOA between the two sensors on such a system would be 1 s, creating an error of 990 ms. Increasing the sample rate to 2 Hz would result in a smaller error of 490 ms.

3.3.2.2 Event Windowing

In addition to sampling frequency, event windowing is another part of temporal sampling that greatly affects a system. As discussed in Section 3.1, multipath can distort a signal, making it difficult or impossible to correlate it to signals from other sensors on the acoustic goniometer antenna. Figure 3.6 shows a set of sample signals from acoustic goniometer sensors that exhibit mild multipath distortion. The time scale (x-axis) for the figure is measured in number of samples. The amount of time between samples is 94 μ s (or the inverse of the sample frequency: 10.638 kHz). Thus, the time at Tick 1 (or sample 1) is the equivalent of 94 μ s, the time at Tick 1000 is 94 ms, and any time can be gained by multiplying the number of ticks (samples) by 94 μ s. The actual arrival of the sound wave at each sensor can be seen around the 125th tick mark where the data from both

sensors shows a small upward peak followed by a severe downward peak. This is followed by a series of smaller minima and maxima terminated with a miniature event shortly after the 2000th tick mark. These smaller peaks and delayed miniature event are the product of multipath distortion. Delayed versions of the initial event with smaller magnitudes add (or subtract) to the actual sound signal. The sensors used to collect this data were spaced within 2 m of one another in order to mitigate the effect of multipath, which could prevent correlation. However, close examination of the signals shows that, although they are quite similar, minor differences exist. The minor distortions between the two sensors may not prevent correlation in scenario presented by Figure 3.6, but the differences would be sufficient to affect the accuracy of the calculated difference in TOA. A simple method for counteracting this problem is to limit the numbers of samples used during the correlation process.

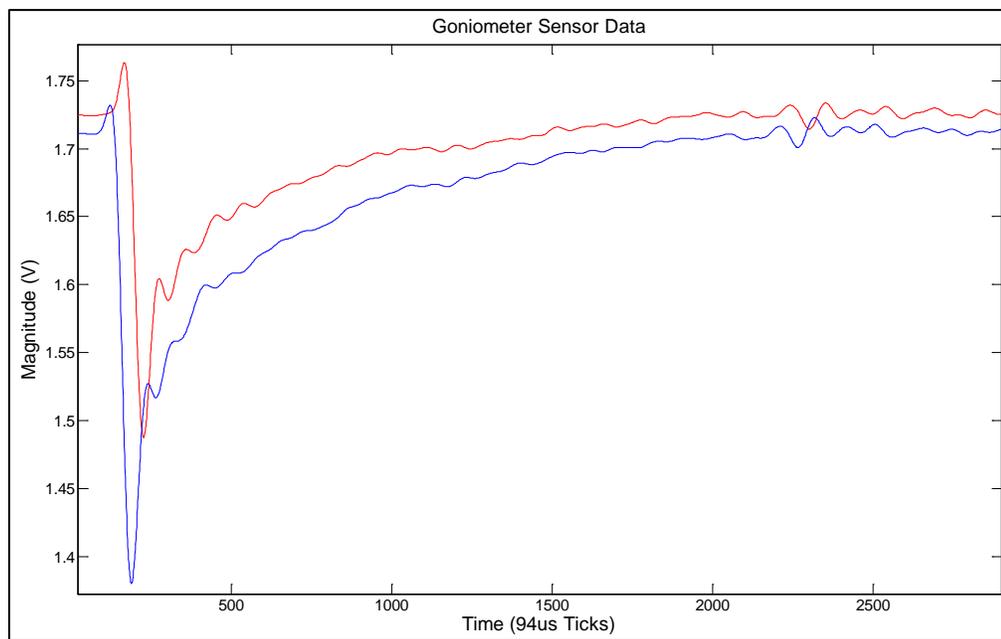


Figure 3.6 Goniometer Sensor Data Showing Mild Multipath Distortion

Choosing a smaller window of the sound event to analyze (or windowing the data) is beneficial to the system in several ways. First, doing so can provide the correlation algorithm with only the initial peaks of a sound event. In the case of the data from Figure 3.6, the selected window could be as small as the first 300 samples. This would include only the first four peaks, which show mostly ideal correlation. Since the goal of the correlation algorithm is to determine the difference in TOA as opposed to the degree of correlation between the two signals, limiting the correlation to the beginning of the event accomplishes the task more accurately. Second, in an embedded implementation limiting the number of necessary calculations increases the system efficiency allowing for faster sampling rates. Reducing the number of samples used during correlation is one potential way to achieve this goal. However, care must be taken that an adequate number of samples is still used for a window size. Similarly to incorporating too much data, decreasing the window size too much can cause inaccuracies and even failure to correlate at all.

3.4 Filtering

An important part of any sensing application is the filter or filters used to prepare the data for analysis. The purpose of a filter is to distort signals in ways that are useful to the system of which it is apart. This task commonly involves noise removal but can also involve amplifying/removing particular frequency components.

Noise is a major concern for most electrical systems and can be an equally distressing problem for the analysis of audio data. Sources for electrical noise can include anything from accidental (yet unavoidable) antennas in the PCB layout and cabling to imperfect electrical connections and components. The most likely contributors to acoustic

noise are sound events, which are not of interest for a given research application (e.g., wind and phenomena, which share the same frequency range as the one of interest).

3.4.1 Filtering Electrical Noise

Electrical noise can be dealt with using analog or digital filters which range in complexity and are only limited by system resources and the abilities of the engineer. Analog filtering is an attractive option for systems like the acoustic goniometer where miniature size is not the primary concern. While smaller sizes are desirable for ease of deployment, a few extra components for analog filtering add at most a few inches to a board design. The key advantage to using analog filters is that they save processor resources as compared to their digital counterparts. The drawbacks to analog filtering are the precision of the parts required, the increased cost, and the complexity of the designs required to achieve the desired results. Digital filters provide more ideal solutions (limited only by the floating point precision of the processor), are simpler to implement, and decrease board size and cost. Unfortunately, as their complexity increases, so too does their demand for system resources. In a system that must process larger amounts of data in real-time, complex filtering is costly or even impossible. Consequently, the best solution for an acoustic goniometer is a hybrid between the two methods. Simple analog filters can be used to adjust the selectivity of the microphone (i.e., limit the response to the frequency range of interest) and remove some electrical noise. Digital filters can then be used to clean the signal as necessary for the rest of the goniometer process.

Electrical noise may also be mitigated by taking appropriate steps in the design of the goniometer antenna. Multi-layered PCBs with clean ground and power planes (e.g., no traces or components, properly spaced vias, etc.) can be used to increase the system's

decoupling capacitance in order to reduce noise from the system's supplies. Additionally, heavily shielded coaxial cable can be used to route both power and sensor signals between the sensor boards and the goniometer motherboard. Finally, careful PCB layout can be used to shorten trace lengths on the important sensor analog traces traveling to the processor for sampling. This effectively eliminates trace antennas and keeps the sensor signal free from high frequency electrical noise. Simple design considerations such as these greatly reduce the presence of electrical noise within a system and profoundly affect performance.

3.4.2 Filtering Acoustic Noise

Acoustic noise poses a more serious problem to the acoustic goniometer than its electrical counterpart. While electrical noise is usually limited to frequency ranges higher than those of interest in this research and is easily controlled by adequate planning in the system design, acoustic noise often occurs at the same frequency range as the sound from the phenomena under study, can be similar in amplitude to the signal of interest, and is difficult or impossible to remove without degrading or even losing the signal of interest. Acoustic noise which occurs outside the frequency range of the sound produced by the phenomena under study can be dealt with similarly to electrical noise. However, acoustic noise which does not fit this criteria is indistinguishable from valid sound data and can not be filtered by standard means. Multiple mechanical options exist for combatting such noise depending on how the noise is produced and how closely related it is to the sound of interest.

The most common source of acoustic noise in a system which is deployed in outdoor environments is wind. A microphone is a mechanical sensor which can use any

number of technologies (e.g., electromagnetic induction, capacitance change, piezoelectricity, etc.) to convert pressure variations (sound waves) into a corresponding electrical signal. Since the sensor depends on a mechanical system which effectively measures vibrations, wind can actuate the sensor causing an erroneous signal or noise. The simplest means of combatting wind noise is to enclose each microphone in an acoustic foam windscreen or a blimp. Windscreens use acoustic foam to isolate the microphone from direct contact with the environment and can work well to mitigate wind noise. The foam is designed to minimally attenuate sound while blocking wind from directly actuating the sensor. However, care still has to be taken in selecting a windscreen in order to ensure the sound of interest is not overly dampened as well. A blimp employs an acoustically transparent material stretched over a frame to create a local area of stagnant air around a microphone in order to protect the sensor from direct actuation by wind (see Figure 3.7). These are usually more expensive options due to their size and requirements for special material. If poor materials are selected for the construction of a blimp, the device can act like the diaphragm of a speaker making the wind noise even worse. Although the aforementioned methods are the most common, other options have been explored for combatting wind noise. Some have attempted to characterize wind noise (in addition to other sources of background noise) in order to remove it digitally. However, due to the variability of wind depending on the surrounding environment, these attempts have had little success in generalized situations [26-27]. A more promising technique involves burying the sensors under earth or snow to create a natural windscreen. However, these materials suffer from the disadvantage that the filtering material significantly affects the speed of sound requiring more complex algorithms for

goniometer calculations. No perfect solution for combatting wind noise exists, but the options previously discussed may mitigate the effects enough for acoustic goniometers to function in many environments.



Figure 3.7 Example of a Microphone Blimp (from Rode Microphones LLC)

Another common source of acoustic noise in any environment is the presence of phenomena other than the one of interest that produce sound in the same frequency range. While still a challenging problem, filtering unwanted events can be easier than removing wind noise. Wind noise signals have random shapes and are unpredictable. Many events which cause unwanted noise in a known environment can either be predicted prior to a research project or produce a signal with their own distinct amplitude and shape. In the case of the former, a simple algorithm can be created to ignore scheduled events that are of no interest. For the latter case, a form of adaptive or smart filtering can be employed to ignore unwanted events. One such method involves fingerprinting and identifying both useless events and the one of interest (see Section 3.5.2). This data can be used by a system to actively select events of interest in noisy environments while filtering out all other stimuli.

3.4.3 Acoustic Metamaterials

A metamaterial is a special type of material that can be used to manipulate, control, and direct phenomena that travel in waves. Metamaterials use conventional materials arranged in repeating patterns at scales smaller than the wavelengths of their target phenomena. They derive their abilities from the size, shape, and orientation of the materials from which they are comprised rather than from the materials themselves [32].

Acoustic metamaterials specifically focus on influencing sound waves and could be useful for certain acoustic goniometer applications. An acoustic metamaterial engineered to direct sound while blocking wind could be used to guide sound waves to a microphone set in a sound/wind proof enclosure. Such a design could effectively eliminate the problem of wind noise and allow for deployment in turbulent environments. An acoustic metamaterial could also be designed as a mechanical audio filter to allow only specific frequencies of interest to reach the microphone or to amplify particular frequencies. Such a filter would improve the selectivity of the acoustic goniometer, allowing the system to more easily limit its calculations to events of interest. Depending on the frequencies contained in the sound of interest, acoustic metamaterials could be made small enough to still maintain ease of deployment for an acoustic goniometer. A design incorporating such a material would have to either encase the microphones within the metamaterial or ensure the metamaterial blocked the only physical input into the microphones. While this should be possible, pursuing either of these options would present challenges that should be carefully considered.

Unfortunately, acoustic metamaterial research is an emerging field of study which will require further exploration before materials such as those previously described will

be commercially available. Designing and implementing an acoustic metamaterial for any of the aforementioned purposes would require a specialized skill set and a significant amount of time and resources. Thus, while acoustic metamaterials show promise for improving acoustic goniometry, the technology is not yet ready for deployment on an actual prototype.

3.5 Event Detection

Most of the previous discussion in this chapter dealt with the process of collecting data, filtering data, and computing the direction of arrival based on differences in arrival times. In order to make any of these processes useful, an acoustic goniometer must first be capable of detecting the sound events from the phenomena under observation. Many papers and books have been devoted to the topic of event detection (e.g., [39-43]), making a full description of all possible options superfluous for this publication. However, a few more common options used regularly in the work related to this field are described in this section.

3.5.1 Thresholding

The simplest of the method employed for event detection in goniometry research (or any sensing application) is the threshold approach. As the name implies, a threshold value is selected at which any spike in the data, which drops below or rises above the selected value, triggers the detection of an event. This process does not require previous knowledge (characterization) of the event before deploying the sensor system in order to work, but it can benefit from the knowledge. Even though previous characterization of an event is an absolute requirement with many more complicated methods (e.g., fingerprinting) while it is only a bonus for thresholding, this algorithm suffers from the

inability to distinguish between different types of events. This makes it less suitable for environments with many sound sources that are not of interest yet could still trigger an event. Additionally, thresholding is more susceptible to false detections due to noise. These drawbacks aside, the algorithm is the quickest means by which a researcher can deploy and test a sensor system. If the goal of the research at hand is not to determine the best possible algorithm for a specific application, thresholding supplies a quick valid way to test more pertinent theories and prove any other portions of a design.

3.5.2 Fingerprinting

Another method for detecting events that is quite popular in the realm of acoustic goniometer research is the fingerprinting method. Fingerprinting uses pattern recognition and the characterization of events (including both those of interest and others which may be considered noise) to match specific features of an event to the data recorded by the acoustic sensors. While fingerprinting can be as simple as differentiation of one or more thresholds, the use of more complex features in a fingerprint are often more likely to be employed (e.g., slope, frequency content, etc). Fingerprinting can be used in conjunction with other algorithms to improve the accuracy or provide a more automated solution. Some implementations use neural networks to create learning algorithms whose performance improves over time, employ wavelet analysis for better feature selection, or take advantage of data from multiple sensor types to create a more unique fingerprint (see Chapter 2.3). As more algorithms are included (or the complexity of the fingerprinting algorithm increases), the resources needed to run the algorithm can become prohibitive. For powerful computers, this is less of an issue, but embedded implementations must weigh the tradeoffs of increased event classification accuracy against diminished sample

rates and decreased DOA calculation accuracy. Using fingerprinting for event detection provides a much better tolerance to noise and event selectivity than many other algorithms (including thresholding). However, it still suffers from the need to fully characterize an event before it can be properly employed. While this may be a perfectly acceptable tradeoff for a research application with a known event and fingerprint, this does not make fingerprinting ideal for testing a general purpose design.

3.5.3 Fisher Statistic

Another method employed by specifically by sensor arrays (i.e., sensor systems comprised of 2 or more of the same sensor) to perform event detection is the use of the Fisher Statistic (F-Detector) [33, 37-38]. The Fisher Statistic is used to quantify signal coherence and can be calculated by dividing the power of a beam by the average difference in power of the beam and individual channels. Equation 3.15 shows the formula for calculating the Fisher Statistic in the frequency domain, while Equation 3.16 and Equation 3.17 provide a definition of the terms. The Fisher Statistic calculation assumes a perfectly correlated signal from one perfect point source and perfectly uncorrelated noise [38].

$$F(\omega, s) = \frac{E(\omega, s)}{E(\omega) - E(\omega, s)} \cdot (N - 1) \quad \text{Eq. 3.15}$$

$$E(\omega, s) = \left| \frac{1}{N} \sum_{j=1}^N A_j(\omega) \cdot e^{(-i\omega s \cdot r_j)} \right|^2 \quad \text{Eq. 3.16}$$

$$E(\omega) = \left| \frac{1}{N} \sum_{j=1}^N A_j(\omega) \right|^2 \quad \text{Eq. 3.17}$$

The number of sensors is represented by N with position vectors r_j . The amplitude information from the Fourier transform is contained in $A_j(\omega)$, and the slowness vector (gives the possible direction from the sensor to the source) is represented by s [33].

Ideally, the Fisher Statistic calculations would be performed for every possible slowness vector with each peak indicating a possible event. However, more realistic implementations choose a step size and perform the calculations, and the maximum peak is selected as the most likely event. This approach is limited in its potential accuracy by the step size chosen for the slowness vector, which is determined by the requirements of the given research application and the capabilities of the hardware. The biggest drawback to the Fisher Statistic approach lies in its computational complexity. Researchers who have employed this algorithm have done so on fairly powerful computing platforms with sufficient system resources and fast processor clock rates. On such systems, the cost of completing such a significant number of calculations for post-processing or even real-time is insignificant compared to the available resources. However, implementing the same algorithm on an embedded system to analyze data in real-time requires more careful consideration. While implementing an F-Detector algorithm for an acoustic goniometer is not by any means impossible on all embedded hardware, the limited resources available could require the designer to sacrifice sampling speed (consequently DOA calculation accuracy) for the sake of implementing the algorithm. Certain higher end processors, FPGAs, or SoCs could implement this algorithm in real-time with little difficulty or negative effect on the sampling rate and DOA accuracy. However, many of the more

limited inexpensive processors would not be able to handle both the DOA calculations and the F-Detector algorithm at once.

3.6 Embedded Algorithm Implementation

Since the primary goal of this work is to lower barriers to further research in the field, a discussion of implementing different/more advanced algorithms on embedded hardware (such as the hardware developed for this dissertation: see Chapter 5) is pertinent. Previous discussions have made reference to the limited capability of embedded systems to implement complex algorithms. As a general rule, embedded platforms tend to be more limited in resources (system clock speed, memory, floating point resolution, etc.) than the computers used to develop high performance algorithms. Consequently, if the implementation of such algorithms on a given embedded platform is even possible, careful planning and some sacrifices may be required.

Consider, as an example, the use of advanced filtering techniques to remove electrical or acoustic noise. Advanced filtering techniques can require multiple math operations involving floating point numbers. These calculations are time consuming for embedded hardware. Performing these calculations on all data could have the effect of reducing the goniometer's sample rate (DOA accuracy). One possible way to avoid this issue is to selectively filter only the important data. Inserting a filter on the data after an event is detected reduces the number of calculations and could allow the system to maintain the same sampling rate while still using the advanced filter. Furthermore, selecting an appropriate type of filter can also reduce the number of required calculations. For instance, implementing the infinite impulse response (IIR) version of a digital filter

as opposed to its finite impulse response (FIR) counterpart would require significantly fewer calculations and reduce the strain on the system.

As another example, more advanced event detection methods could be employed on embedded hardware depending on the system's capabilities. Fingerprinting using a simple pattern could be performed in a fairly straightforward manner on many platforms. However, as the complexity of the fingerprint increases (number of features used to create the fingerprint) so too do the number of calculations. One possible method for reducing the number of calculations is to use fingerprinting in conjunction with a simple event detection method (e.g., thresholding). The simple method can be used to identify events which can then be differentiated based on fingerprints. Thus, the fingerprint calculations would only be performed on pre-screened data as opposed to all sampled data.

The key to implementing any algorithm properly on an embedded system is to understand the limitations of the hardware and adjust the code appropriately. Making decisions that reduce calculations, limit or eliminate the need for floating point operations (in processors without dedicated floating point hardware), and reduce memory usage can allow for the implementation of more advanced algorithms or even just improve system performance (e.g., allow for faster sample rates). Writing the algorithm in this fashion may reduce its effectiveness or make it more difficult for other programmers to understand (slowing collaborative projects), but this sacrifice must be made in many cases just to make the implementation possible.

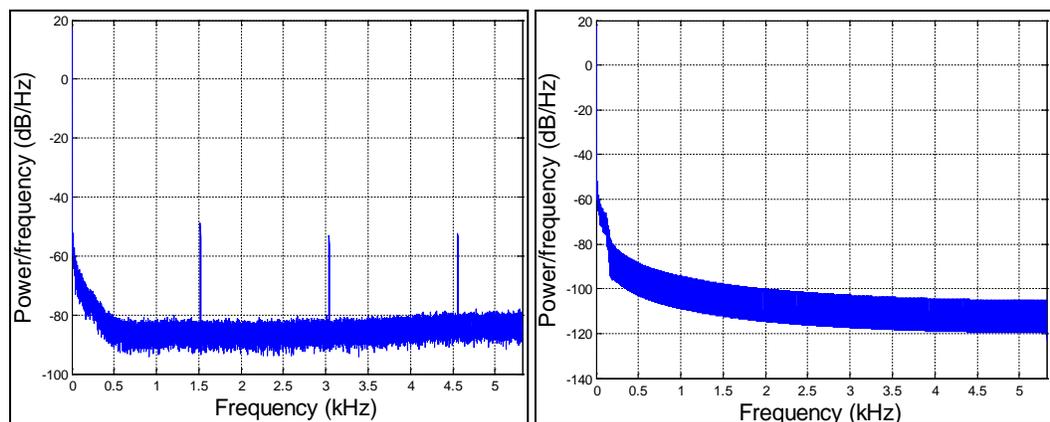
CHAPTER FOUR: SIMULATIONS

Completion of this research required a significant amount of simulation coupled with actual hardware and firmware development. The following subsections detail the simulations of the acoustic goniometer. Simulations were performed both with ideal data/conditions to prove the theory behind the design and with data gathered by actual hardware in realistic conditions in order to prove the feasibility of a fully embedded implementation.

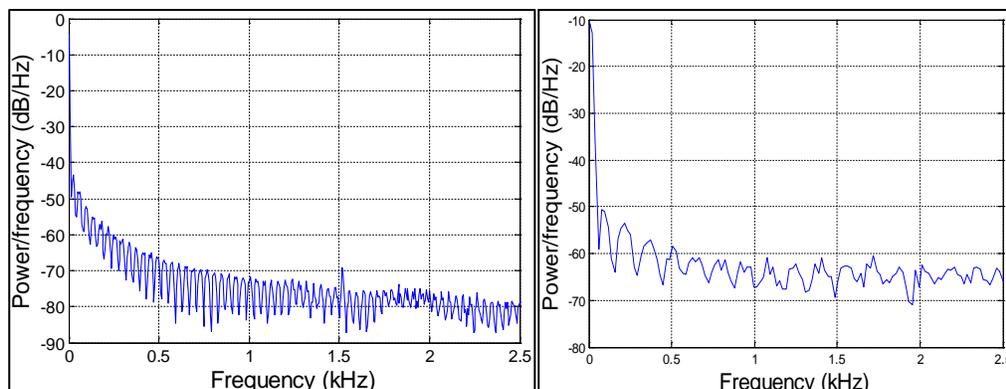
The design was split into more manageable blocks before simulation was attempted. The first simulation series dealt with the feasibility of DOA calculation given a significantly reduced inter-sensor spacing and explored the effect of various parameters on the error in the calculated DOA. The second simulation series verified correlation algorithms for determining the difference in arrival time. The third simulation series combined the concepts tested in the first two simulations in order to prove the feasibility of performing DOA calculations given data from an actual goniometer antenna. The final series of simulations tested the event detection algorithms.

Some of the simulations used data gathered from actual sensors. Readily available infrasound sources were used to prove the validity of the models. Although both contain audible elements to their frequency content, the closing of a door in an office building and gunfire contain detectable levels of infrasound. Even though the techniques used in monitoring audible sound events would be a valid means to prove the goniometer models, Figure 4.1 shows the frequency content for these two events in order to prove that only

sounds in the infrasonic range were used to test the models. Figure 4.1 (a) and (b) show the Power Spectral Density (PSD) plots for a data set containing multiple gunshots before and after digital filtering, respectively. Notice that the largest frequency component is closest to 0 Hz in both cases. Some higher frequency components can be seen in Figure 4.1 (a), but the analog filter on the output of the microphone has kept them below -40 dB/Hz. While these frequencies could be part of the response of the microphone to the audible components of the events, electrical noise in the system could also be a contributing factor. In Figure 4.1 (b), these components have been entirely removed by the digital filter used to remove noise before processing. Figure 4.1 (c) shows the spectral content for a single shot event. The plot was adjusted to focus on the lower frequencies in order to show that the majority of the content is below 200 Hz. Finally, Figure 4.1 (d) shows the PSD plot for a single door closing event. Again, the largest frequency components are in the infrasound range. Thus, Figure 4.1 shows that the data used for the simulations to test the models was valid infrasound data. The following sections detail the setup and results for each simulation.



(a) Multiple Shot Raw Sensor Data (b) Multiple Shot Filtered Sensor Data



(c) Single Shot Data

(d) Door Event Raw Data

Figure 4.1 Power Spectral Density Plots for Door and Gunshot Sound Events

4.1 Sensor Spacing Simulation

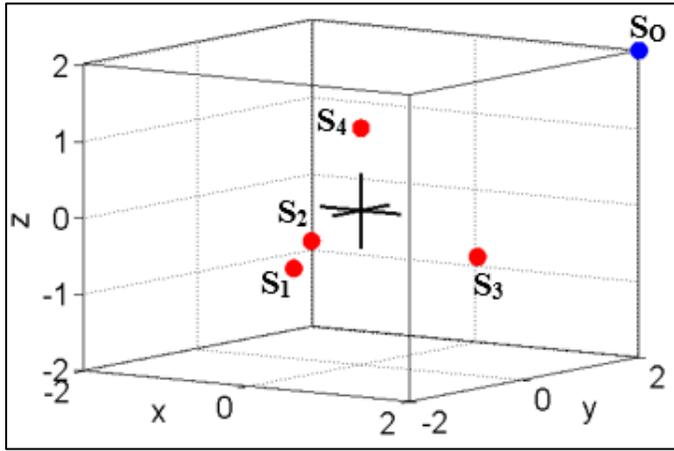
In order to prove the viability of this research, simulations were created that would test the most questionable and critical aspect: the effect of inter-sensor spacing on DOA calculation and source localization. Although completion of the other aspects of this research was no small task, they had largely been proven by previous research (see Chapter 2): DOA algorithms, feasibility of characterizing the sound fingerprint of an event, pattern recognition, etc. Simulations were created in stages using Matlab and evolved as more questions presented themselves. All simulations assumed an ideal sound point source, double precision floating point arithmetic (64 bit), and an ambient temperature of 20° C yielding a constant speed of sound (c) equal to 340.29 m/s. Some simulations used ideal time resolution to prove algorithms. However, these are clearly indicated, and more realistic restrictions were placed on the simulations used to verify feasibility. This section details the setup and results of simulation.

The code for each simulation is split into two main parts (with a total of 80 lines of code). The first part sets the locations for all microphones and the sound source. This information is then used to calculate the time it would take for the sound to travel from

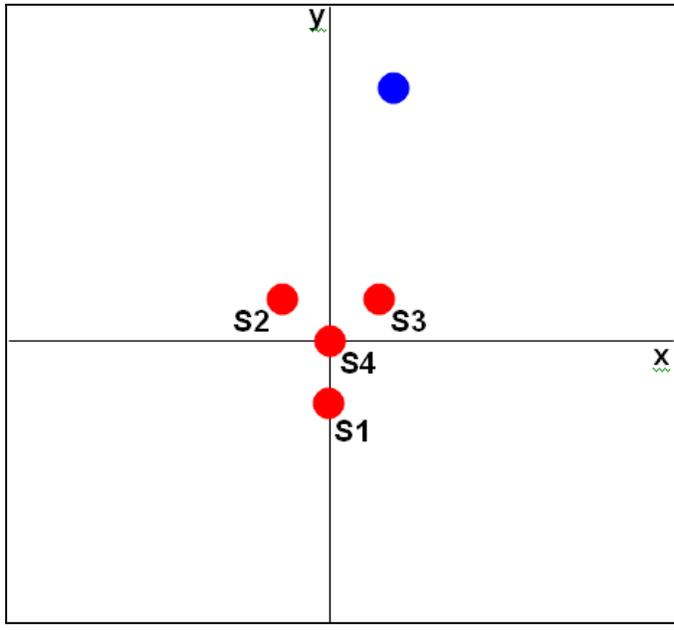
the source to each microphone (time of arrival or TOA). The second part of each simulation handles the calculations that would be performed by the goniometer. For the purpose of simplification and not re-proving the concepts developed as part of previous work, the detection of a sound event, classification as an event of interest, and correlation of events between microphones was taken for granted for this particular set of simulations (see Sections 4.2-4.4 for further discussion/simulation of these aspects of acoustic goniometry). The goniometer section of the simulation uses the TOA calculations to obtain the difference in arrival times. The limitations of the ADC sampling rate are imposed here by rounding using a ceiling function according to the predetermined resolution. Since the ADC on the hardware can only sample in intervals of approximately 0.0001 s, the code similarly handicaps the simulation (e.g., an event which occurs at 0.000143 s is recorded by the simulation as occurring at 0.0002 s). After this point, the rest of the calculations are performed as the real hardware would complete them. The DOA result is given by azimuth and elevation, and a vector is graphed using the magnitude known by the first part of the simulation in order to verify the validity of calculation. The predetermined azimuth and elevation from the first part of the simulation are also used to calculate error and act as a second check for the simulation.

The setup for the first series of simulations can be seen in Figure 4.2. The microphones are represented by the four red dots and are configured in a regular tetrahedral antenna formation with a spacing of 2 m. The center of the antenna is located at the origin of the coordinate system to ease calculations. The ideal sound point source is represented by the blue dot and was moved to multiple locations to verify the algorithm. In Figure 4.2, the source is located excessively close to the sensor array in order to make

the antenna geometry visible. The simulation uses the location of the source (S_0) to calculate the arrival time at each sensor (S_1 - S_4). The arrival times are used as inputs to the simulated goniometer which proceeds to calculate the estimated DOA based solely on the arrival times provided (same way in which the real hardware would operate).



(a) Sensor Orientation 3D View (origin marked with )



(b) Sensor Orientation Top-Down view

Figure 4.2 Setup for First Series of Simulations

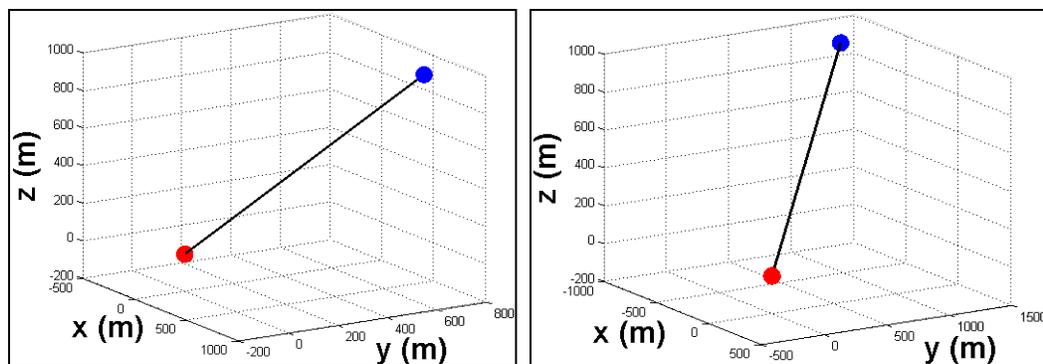
The results of the first simulations can be seen in Figure 4.3. Due to the distance between the source and the microphones, some of the dots representing the microphones are merged together in some of these figures. The first simulations illustrated in Figure 4.3 were performed with idealized time resolution in order to prove the validity of the model. The error was calculated for each simulation by comparing the DOA (azimuth and elevation) calculated by the simulated goniometer to the actual DOA determined by the source's placement. Equation 4.1 and Equation 4.2 show the formulas used to calculate the error for azimuth and elevation, respectively. The total error was recorded as the sum of these two calculations. The error calculations were performed with respect to 180° since a calculation placing the DOA as coming from the opposite direction from the expected would be the worst possible error. The source can be seen in three separate locations, and the DOA is calculated with an error of less than 0.05% in all cases. The small amount of error present is due to the finite representation of the floating point numbers imposed by Matlab (identical to those imposed by the actual hardware). Thus, the simulation model is valid for testing the microphone spacing and time resolution (ADC sampling rate) parameters.

$$Error_{AZ} = \frac{|AZ_S - AZ_A|}{180^\circ} \quad \text{Eq. 4.1}$$

Where AZ_S is the azimuth calculated by the simulation and AZ_A is the actual azimuth determined by the location of the source.

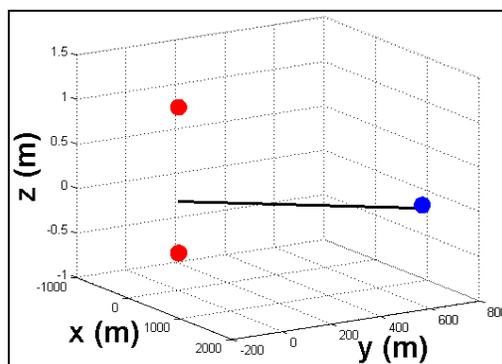
$$Error_{EL} = \frac{|EL_S - EL_A|}{180^\circ} \quad \text{Eq. 4.2}$$

Where EL_S is the elevation calculated by the simulation and EL_A is the actual elevation determined by the location of the source.



(a) Source at (680, 680, ~961.7)

(b) Source at (-680, 1115, ~961.7)



(c) Source at (1450, 680, 0)

Figure 4.3 Simulation Validation Using Ideal Time Resolution

The second set of simulations was performed with varying restrictions placed on the time resolution for a fixed microphone spacing of 2 m. The purpose of this test was to verify the effect of time resolution on DOA calculation accuracy. The results for these simulations can be seen in Table 4.1. For each of the simulations, the azimuth and elevation of the sound point source were held constant at 45 degrees. From these results, the relationship between time resolution (sampling rate) and accuracy can be clearly seen. As the resolution is increased, the model is capable of achieving more accurate results. This outcome was expected since the increased time resolution allows the model to better

determine the differences in TOA for each microphone pair. Although only three distances are shown in Table 4.1, this relationship holds true for all other distances as well.

Table 4.1 Simulated error for varying sample rate (f_s) and source distance (d_s) while maintaining inter-sensor spacing ($d=2\text{m}$), azimuth ($AZ=45^\circ$), and elevation ($EL=45^\circ$)

Sample rate (f_s)	10kHz		
Source Distance (d_s)	AZ ($^\circ$)	EL ($^\circ$)	Error (%)
1360m	43.710	43.690	1.445
2400m	43.710	43.690	1.445
5200m	43.710	43.690	1.445

Sample rate (f_s)	100kHz		
Source Distance (d_s)	AZ ($^\circ$)	EL ($^\circ$)	Error (%)
1360m	44.891	45.110	0.122
2400m	44.890	45.122	0.129
5200m	44.891	45.110	0.122

Sample rate (f_s)	1MHz		
Source Distance (d_s)	AZ ($^\circ$)	EL ($^\circ$)	Error (%)
1360m	44.990	44.987	0.013
2400m	44.990	44.987	0.013
5200m	44.991	45.011	0.011

The next set of simulations was designed to verify and understand the magnitude of the effect inter-sensor spacing on DOA calculation accuracy. Thus, inter-sensor spacing (d) was varied while the time resolution was held constant ($f_s=10\text{ kHz}$), and the DOA was kept constant at 45 degrees for both azimuth and elevation. As with the previous set of simulations, d_s was varied to confirm that the relationship between d and DOA accuracy was independent of the distance from the sound source. The results of this third set of simulations can be seen in Table 4.2. As expected, the inter-sensor spacing does not have a readily predictable effect on calculation accuracy (given a reasonable f_s).

Varying d moves the sensors in relation to the sound source as well as one another. If the sensors move such that their TOA is closer to a multiple of $100\mu\text{s}$ ($1/f_s$ or T_s), the rounding effect caused by the finite time resolution is mitigated. This results in smaller errors in the final calculation. However, if the change in position moves the TOA farther from a multiple of T_s , the result is an increase in error. The importance of the inter-sensor spacing is its effect on the required sample rate. Shorter distances for d require faster sample rates (time resolution: f_s) in order to maintain the same amount of error.

Table 4.2 Simulated error for varying inter-sensor spacing (d) and source distance (d_s) while maintaining sample frequency ($f_s= 10\text{kHz}$), azimuth ($AZ=45^\circ$), and elevation ($EL=45^\circ$)

Sensor Spacing (d)	1m		
Source Distance (d_s)	AZ ($^\circ$)	EL ($^\circ$)	Error (%)
1360m	42.876	47.124	2.361
2400m	43.488	43.488	1.680
5200m	42.327	47.673	2.972

Sensor Spacing (d)	2m		
Source Distance (d_s)	AZ ($^\circ$)	EL ($^\circ$)	Error (%)
1360m	43.710	43.690	1.445
2400m	43.710	43.690	1.445
5200m	43.710	43.690	1.445

Sensor Spacing (d)	3m		
Source Distance (d_s)	AZ ($^\circ$)	EL ($^\circ$)	Error (%)
1360m	44.891	45.109	0.121
2400m	44.891	45.109	0.121
5200m	44.179	45.821	0.912

Sensor Spacing (d)	4m		
Source Distance (d_s)	AZ ($^\circ$)	EL ($^\circ$)	Error (%)
1360m	44.538	44.538	0.513
2400m	44.450	45.550	0.611
5200m	44.450	45.550	0.611

Sensor Spacing (d)	5m		
Source Distance (d_s)	AZ (°)	EL (°)	Error (%)
1360m	44.829	45.171	0.19
2400m	44.829	45.171	0.19
5200m	44.442	45.558	0.62

Sensor Spacing (d)	6m		
Source Distance (d_s)	AZ (°)	EL (°)	Error (%)
1360m	44.892	45.108	0.12
2400m	44.613	45.387	0.43
5200m	44.892	45.108	0.12

4.2 Time Delay Simulation

The ability to calculate the DOA of a sound hinges upon the ability to accurately measure the differences in arrival time between sensors on the antenna. The most basic method for calculating delay would be to identify the event at each sensor and record its arrival time. However, event identification becomes a point of concern. The system must not only be capable of sensing a given event (via simple threshold, fingerprinting, etc.) but must also determine if the events detected at each microphone come from the same source. One way to achieve this is by employing cross correlation of the signals recorded from the microphones to match like events. The index of the maximum value in the correlation result corresponds to the phase shift between the signals (the delay time). Due to its simplicity, this method was the basis for the first simulation used to try calculating sensor delays. Equation 4.3 is the implementation of a cross correlation function where k defines the length of signals being convolved. Although Matlab has a built in correlation function, the simulations use a custom function which could easily be implemented in C or Verilog.

$$(f * g)[n] = \sum_{m=-k}^k f^*[m] \cdot g[m + n] \quad \text{Eq. 4.3}$$

Where f and g represent functions/signals, m is the index for the summation, k is the limit for the summation, and n is the index of the current value being calculated.

Cross correlation may also be performed in the frequency domain using Equation 4.2. The downside to this method lies in its computational complexity. Although several algorithms have been created in C and Verilog for performing the Discrete Fast Fourier Transform (DFFT), the code is far more lengthy, requires more time to execute, and requires floating point hardware to achieve the accuracy required for DOA calculations. The major benefit gained by employing this method of correlation is its robust performance in the presence of noise [13]. While noise may be filtered prior to the cross correlation stage to mitigate the issue, the potential for less signal filtering made this option attractive.

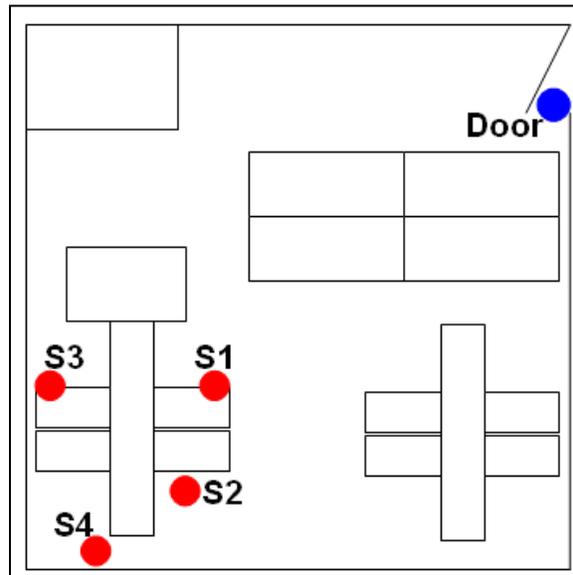
$$(f * g)[n] = \text{Real}(F^{-1}(F(f[n]) \cdot F^*(g[n]))) \quad \text{Eq. 4.4}$$

Where f and g represent functions/signals, F denotes the DFFT, and F^{-1} is the inverse DFFT.

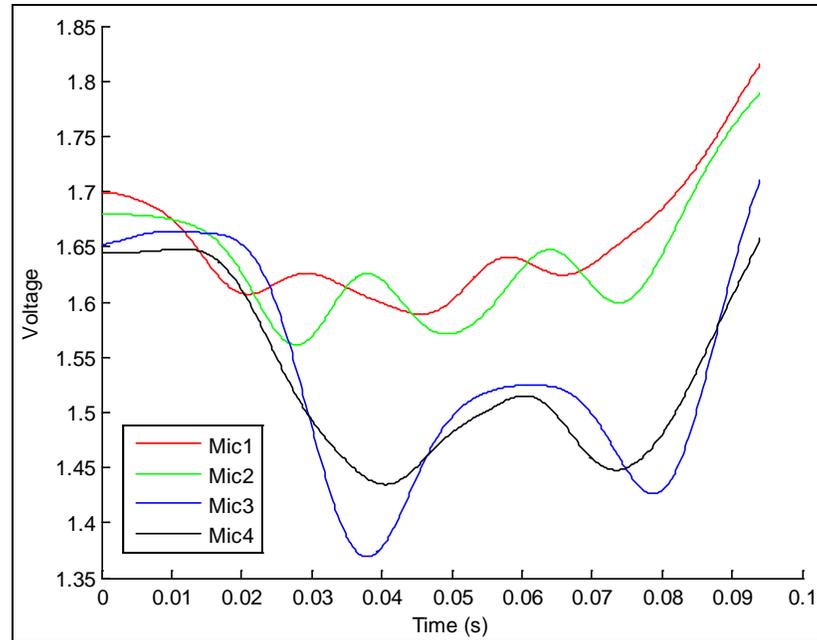
Another method that can be used to calculate the delay involves picking similar features of each signal and comparing their timestamps. These features may be any characteristic of the lines ranging from simple minima and maxima to the more complex local slopes and rate of change in slopes. Although this seems more straightforward than

cross correlation, automatic feature picking is a more difficult process when implemented in embedded hardware. This option in its simplest form (local minima and maxima) was explored as part of the first round of simulations.

Test data for these simulations was acquired using actual microphones (for a description of the hardware, refer to Chapter 5.2) arranged in a random pattern as shown in Figure 4.4 (a). The distances from sensors 1 through 4 (S1-S4 or Mic1-Mic4) to the sound source were approximately 7.335 m, 8.809 m, 10.745 m, and 9.018 m, respectively. Since localization was not the goal of this simulation, the tetrahedral configuration with exact inter-sensor spacing was not deemed necessary. The infrasound generation mechanism used to stimulate the sensors was a slammed door. The data collected by this experiment can be seen in Figure 4.4 (b).



(a) Laboratory Setup



(b) Test Results

Figure 4.4 Door Slamming Test

A brief examination of these signals reveals a disturbing apparent lack of correlation between some pairs. Although Mic1 and Mic2 show a clear relationship, and the same can be said to some extent for Mic3 and Mic4, the former pair and latter pair do not have nearly as many common features. Between time 0s and 0.07s, Mic1 and Mic2 each show 3 distinct minima and 2 peaks. However, Mic3 and Mic4 have only 2 minima and 1 peak for the same time period. Since the microphones do not respond to normal audible frequencies, and only one infrasound source was present for the experiment, this lack of obvious correlation can be blamed on multipath signal propagation. Mic1 and Mic2 were located closer to the sound source and had less interference between them and the source. However, Mic3 and Mic4 were farther from the source and placed closer to walls. Thus, this pair was more heavily affected by the delayed, reflected signals traveling along other propagation paths (as mentioned in the discussion of Figure 3.3).

While at first glance this may seem problematic, the less-than-ideal signal correlation for these test signals provided a good worst-case scenario for testing the time delay algorithms.

The best test cases in the dataset seen in Figure 4.4 (b) for the correlation algorithm are the Mic1-Mic2 and the Mic3-Mic4 sensor pairs. The Mic1-Mic2 set was used for preliminary verification with non-ideal signals (the correlation algorithm having previously been verified using ideal, binary signals). The results of the correlation can be seen in Figure 4.5. The upper graph in the figure shows the output from the correlation function described in Equation 4.3. The maximum value indicates the point of maximum correlation and the time delay between the two signals. In the case of the Mic1-Mic2 pair, the calculated delay using Equation 4.3 is 4.6 ms. A quick calculation places Mic2 1.57 m farther from the door than Mic1 (actual difference is 1.474 m: ~6.5% error). Using Equation 4.4 gives in slightly poorer results with an error of approximately 15.4%. The lower graph in Figure 4.5 shows the filtered signals with marked local minima and maxima. Starting from the first major local minimum, an average of the delays between these features gives a time delay of 6.6 ms for a calculated travel distance of 2.25 m (~52.6% error).

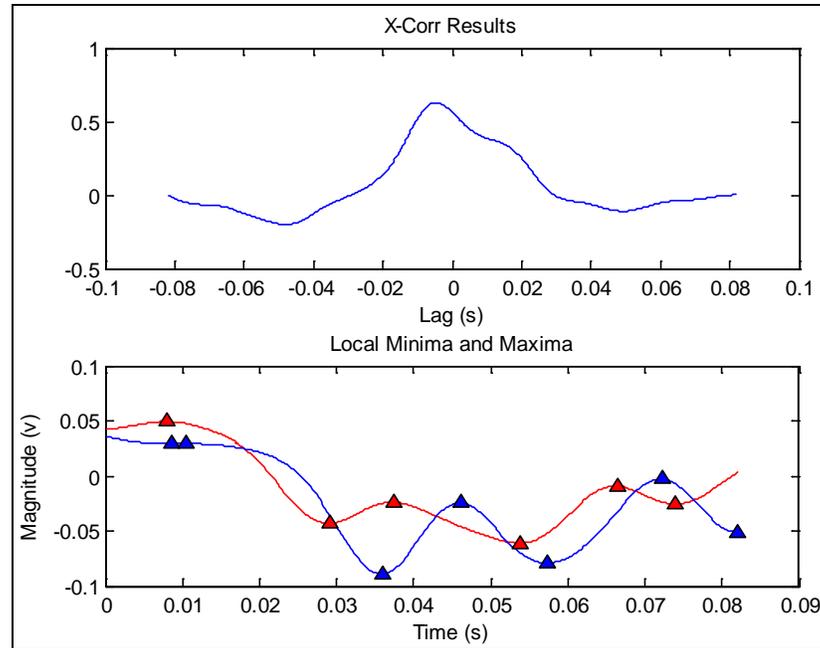


Figure 4.5 Mic1-Mic2 (Red-Blue) Correlation Results

The results from the cross correlation technique for the Mic3-Mic4 pair were less ideal (see Figure 4.6). According to the results of both correlation methods, the sound event occurred at near the same time resulting in less than 0.28 m of difference between the two source-sensor distances (greater than 84% error). The difficulty with correlating these two signals can be easily identified by observing the phase difference in the lower graph of Figure 4.6. Although Figure 4.4 seems to suggest that the two signals are quite similar, the phase shift between the two is definitely not constant. Prior to the first major local minimum (see between time 0.04 and 0.05 s), the signal from Mic3 appears to lag Mic4. However, from that local minimum to the subsequent maximum, Mic3 leads Mic4. After this, Mic3 again appears to lag Mic4. Consequently, cross correlation does not work well for calculating the time delay between these two signals. If instead, the difference between the two first major local minima is used, the calculated difference in

distance is 0.88 m (50.2% error). This is only marginally better but is the best possible result for the given signals using the previously discussed methods.

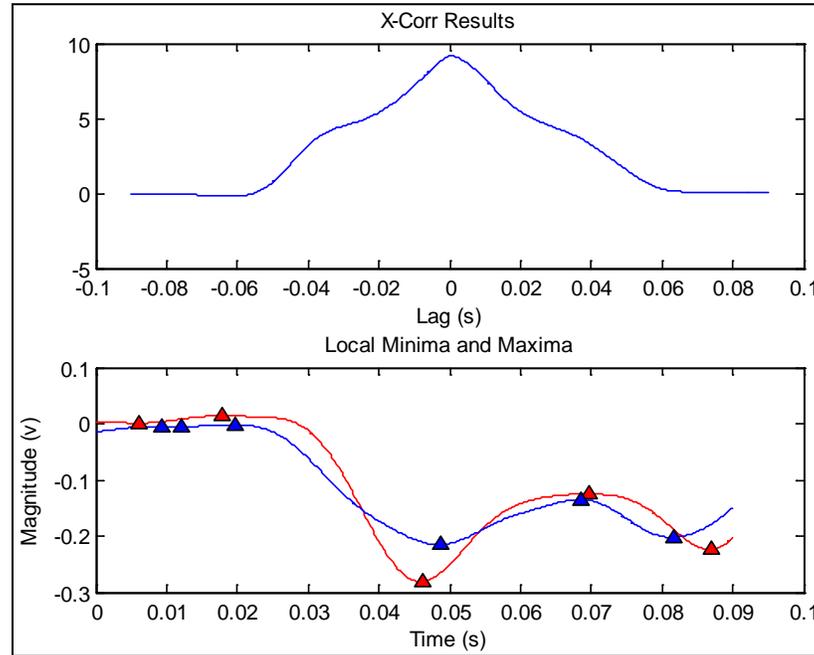


Figure 4.6 Mic3-Mic4 (Red-Blue) Correlation Results

Despite the differences between the later portions of the signal, the cross correlation method worked relatively well on the Mic2-Mic3 pair. The results in Figure 4.7 show a lag of 5.8 ms between the two signals resulting in a calculated distance difference of 1.97 m (~2% error). The method described by Equation 4.4 yields much poorer results with an error of 43.3%. This discrepancy could likely be attributed to the differences between the frequencies contained in the two signals. The Mic3 signal is quite obviously missing the higher frequency component present in the Mic2 signal. Consequently, given that the correlation takes place in the frequency domain, the application of Equation 4.4 to this data set should be expected to yield poorer results. Looking at the local extrema produced only marginally better results than the method described by Equation 4.4. Since the local maxima do not match between the two signals,

an average of the difference between the local minima may be used. This results in a 7.6 ms delta corresponding to a distance of 2.59 m or 30.2% error.

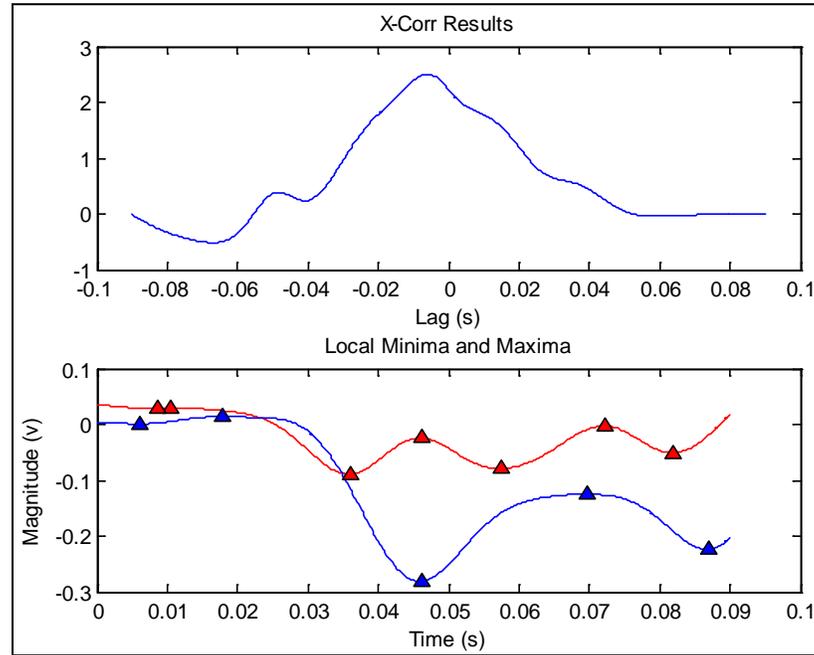


Figure 4.7 Mic2-Mic3 (Red-Blue) Correlation Results

The Mic2-Mic4 pair suffered from very similar issues to those described in the discussion of the Mic2-Mic3 pair. The two methods of correlation described by Equation 4.3 and Equation 4.4 yielded errors of 2.3% and 30.3%, respectively. Using the feature method of time delay estimation proved quite difficult in this case due to the dissimilarities betwixt the two signals (see Figure 4.8). The results from an attempt at using local minima yielded an entirely unacceptable 974.6% error.

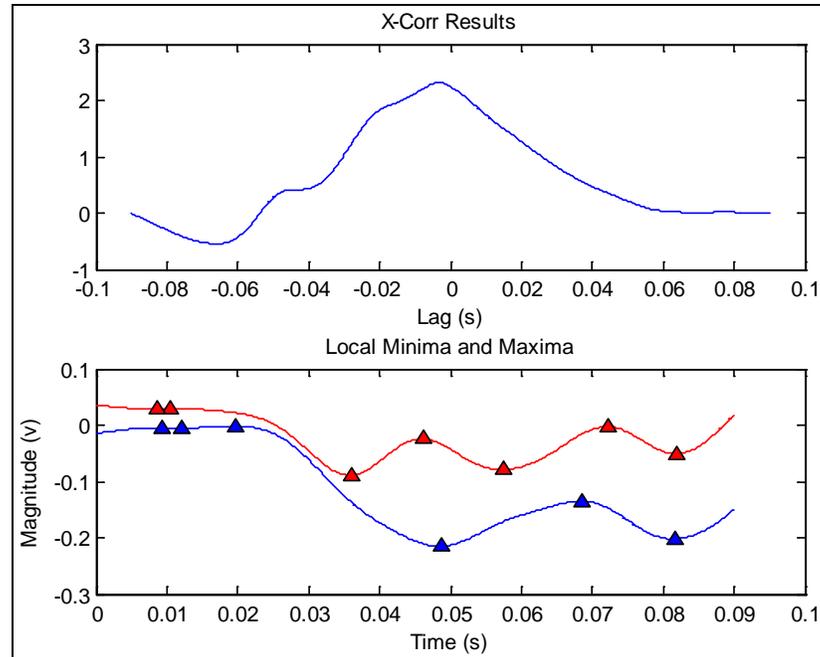


Figure 4.8 Mic2-Mic4 (Red-Blue) Correlation Results

The results for the Mic1-Mic3 pair were reasonably accurate. The standard correlation method placed the difference in their distances from the source at 3.61 m (5.9% error). Notice, however, the graph of the correlation results in Figure 4.9. The peak value (indicating the phase difference) is only slightly higher than the second largest peak. Aside from the obvious visual difference between the two signals, this feature of the cross correlation graph indicates that the signals do not match well. In an embedded implementation of this algorithm, a confidence factor may be used to describe the quality of correlation between two signals in order to determine if the calculated delay is useable in further calculations. Such results as those shown in Figure 4.9 would have a low confidence factor and could be excluded from the final calculations despite the low error. The results from Equation 4.4 were not nearly as good (21.3% error) as those produced by the standard correlation method. However, using the local minima yielded the poorest results with an error of 47.5%.

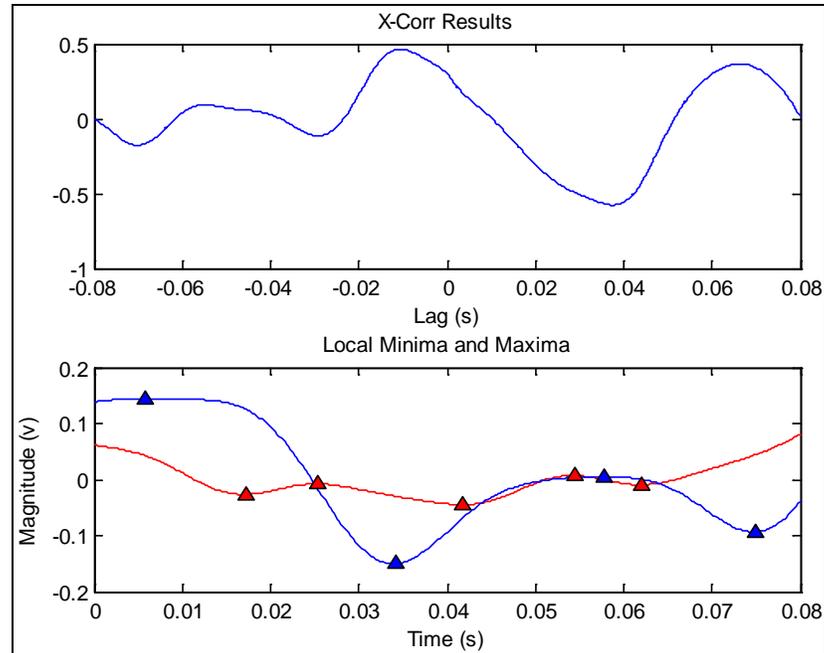


Figure 4.9 Mic1-Mic3 (Red-Blue) Correlation Results

The standard cross correlation method placed Mic4 1.70 m farther from the sound source than Mic1 (1.01% error). Although not as bad as the results seen in the cross correlation graph of Figure 4.9, the results for the Mic1-Mic4 pair (Figure 4.10) do not show as clear a correlation as the other sensor pairs. As with the Mic1-Mic3 pair, this is again due to the obvious dissimilarities in the signals. However, in this case, the confidence factor that might be used in an embedded algorithm could still be high enough to justify using this sensor pair. The frequency domain correlation again performed worse, producing an error just less than 9.2%. Using the local minima to calculate the delay resulted in 175% error.

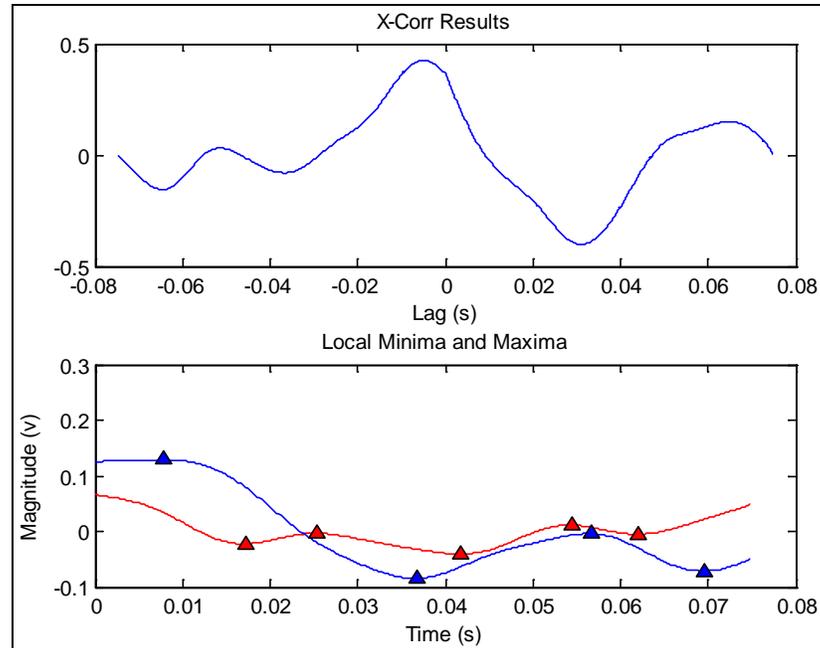


Figure 4.10 Mic1-Mic4 (Red-Blue) Correlation Results

The results of the simulations showed the standard correlation method to be far superior to either of the other tested options. None of the errors produced from the correlation method described by Equation 4.3 exceeded 6% even under less than ideal circumstances. The frequency domain correlation consistently produced poorer results. Since the signals contained little noise after a simple digital filter was applied, this was expected. Perhaps, under conditions in which noise seriously degrades the signal, the frequency domain calculation would be the more reliable choice. Feature based time delay calculations performed exceptionally poorly in these tests. Granted, this could be due to the simplicity of the features selected for the process. More complex features may produce better results but would increase the complexity of the implementation.

4.3 DOA Simulation

Although the results from the previous simulations showed promise, proving the feasibility of actual hardware to collect useful data that could produce reasonably

accurate DOA calculations was still a requisite step for moving forward. In order to accomplish this feat, data was collected using prototype hardware outside of the lab environment (for a description of the hardware, refer to Chapter 5.2 Hardware). The microphones were arranged in a regular tetrahedral pattern with approximately 2.032 m spacing between each microphone in the same orientation shown in Figure 4.2. Shots were then fired from an AR-15 using .223 caliber ammunition at 4 pre-determined and measured locations around the tetrahedral antenna (as shown in Figure 4.11). The data was logged at a rate of approximately 10.6 kHz, and Matlab was used to perform time delay calculations (via the standard cross correlation method: Equation 4.3) and DOA calculations.

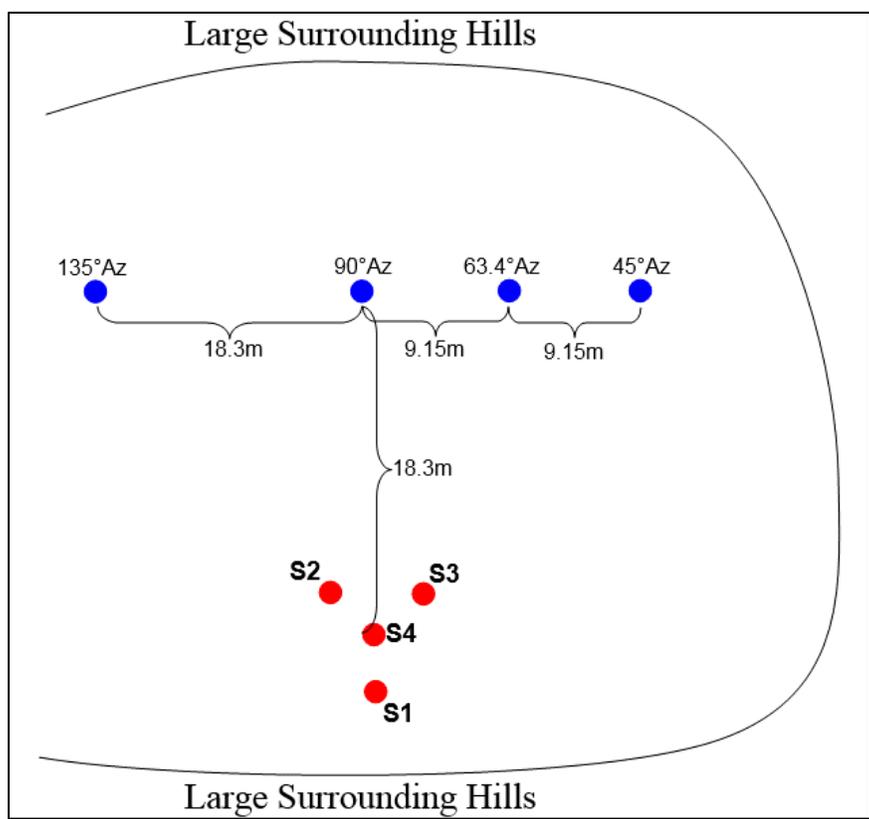


Figure 4.11 DOA Simulation Event Source Layout

The first shot was fired from a position approximately 18 m away from the phase center of the antenna at an azimuth and elevation of about 90° and 0° , respectively. Data collected for the first shot can be seen in Figure 4.12. As with Figure 3.6 and all data collected with the acoustic goniometer, the time scale (x-axis) for the figure is measured in number of samples. The amount of time between samples is 94 μs (or the inverse of the sample frequency: 10.638 kHz). Thus, the time at Tick 1 (or sample 1) is the equivalent of 94 μs , the time at Tick 1000 is 94 ms, and any time can be gained by multiplying the number of ticks (samples) by 94 μs . Although the magnitudes are not equal, determining the order of arrival can still be accomplished by analyzing the figure. The event appears to register first at microphones 2 and 3 almost simultaneously. Microphone 4 detects the shot shortly after the first two, and the event is seen by microphone 1 last. Given the orientation of the microphones (see Figure 4.2) and the location of the shot, this was the expected response of the system. The cross correlation algorithm agreed closely with the visual analysis, and the DOA calculation determined the source to be originating from a direction with an azimuth of 95.6° and an elevation of 2.3° . Assuming the shot was fired exactly from the desired location, the error for the azimuth and elevation calculations would be 3.1% and 1.3%, respectively.

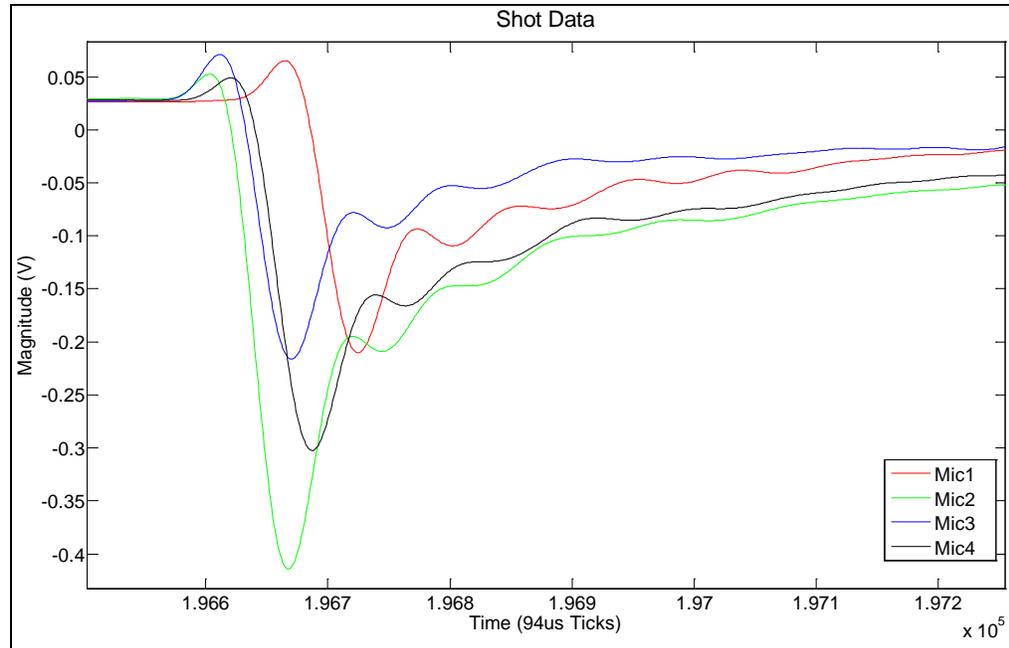


Figure 4.12 Shot Data with Source at 90° (Azimuth) with Respect to the Antenna

The second shooting position was selected to be approximately 26 m away from the phase center of the antenna at an azimuth and elevation of around 45° and 0°, respectively. Results from this test can be seen in Figure 4.13. At this DOA, the event registers first at microphone 3. Then, it registers at microphones 2 and 4 followed closely by microphone 1. Again, the cross correlation and DOA algorithms performed well with a calculated azimuth and elevation of 44.2° and -6.1°, respectively. This result has an error of approximately 0.4% and 3.4% for azimuth and elevation, respectively.

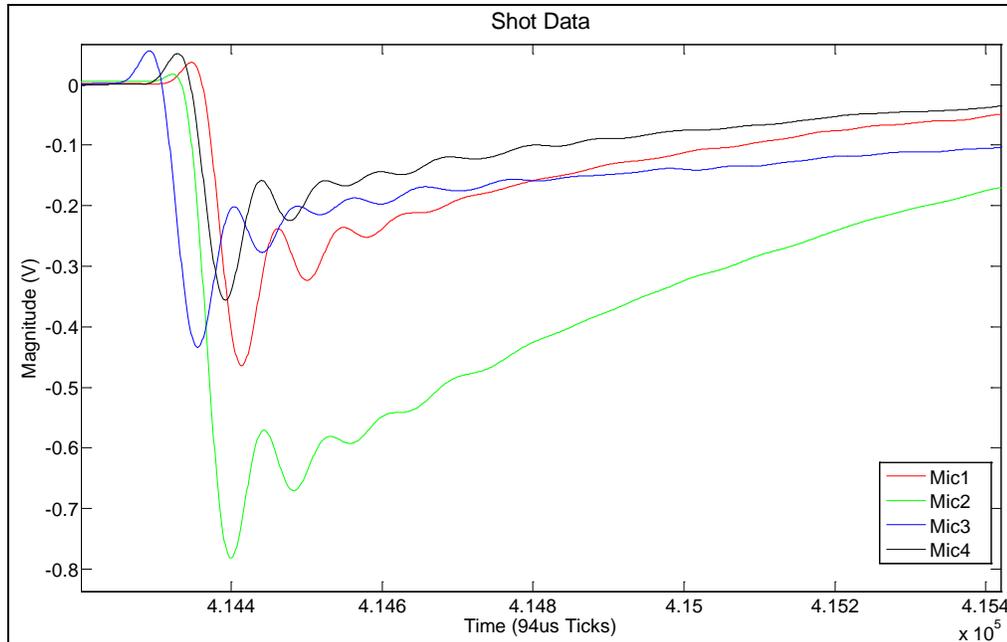


Figure 4.13 Shot Data with Source at 45° (Azimuth) with Respect to the Antenna

For the third part of this test, a location 20.4 m at an azimuth of approximately 63.4° and an elevation of 0° from the phase center of the antenna was selected for the shot. The results can be seen in Figure 4.14. The order in which the event triggered at each microphone for this shot is identical to the order seen in the previous test. However, the differences in TOA between each microphone (excepting the TOA for microphones 2 and 4) have changed slightly. This is somewhat apparent visually in Figure 4.14 but was more prominent in the results for the cross correlation. The final DOA calculation placed the azimuth at 65.2° and the elevation at -6.1°, which corresponds to an error of 1% and 3.4%, respectively.

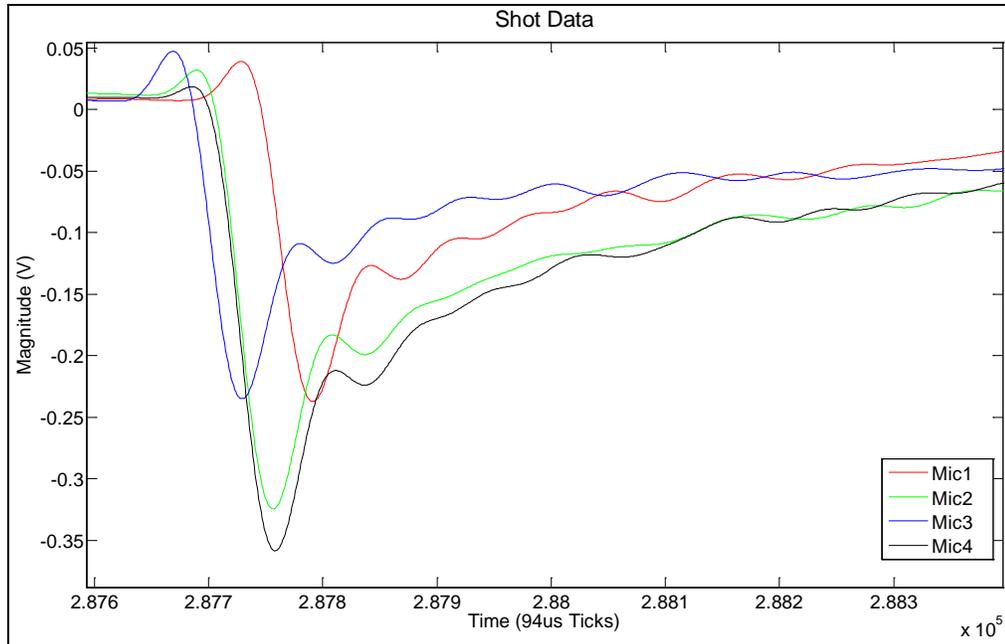


Figure 4.14 Shot Data with Source at 63.4° (Azimuth) with Respect to the Antenna

In order to prove that the system is not limited to one quadrant, the fourth shot location was selected to be 26 m away from the phase center of the antenna at an azimuth and elevation of 135° and 0°, respectively. The results (shown in Figure 4.15) are very similar to those seen in Figure 4.13. The key difference is that microphones 2 and 3 have traded places in the order of event detection. Given the orientation of the antenna with respect to the two shots, this was the expected result. The final DOA calculation for the test depicted in Figure 4.15 determined an azimuth of 129.2° and an elevation of -1.6° with a corresponding error of 3.2% and 0.9%, respectively.

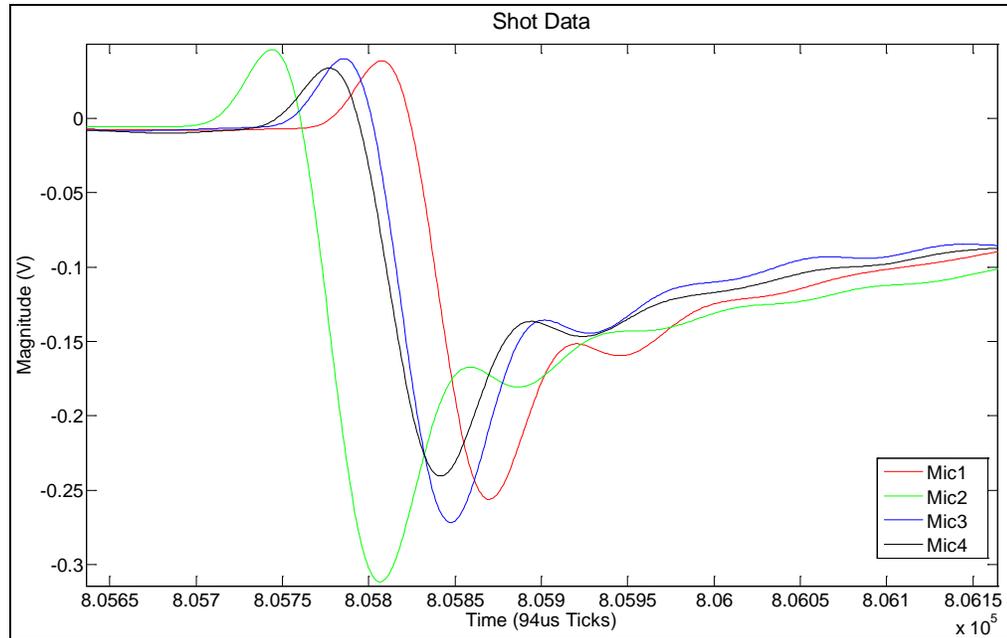


Figure 4.15 Shot Data with Source at 135° (Azimuth) with Respect to the Antenna

Although the errors for each of the tests were acceptable, they were larger than the simulated error. This could easily be explained by two factors. First, since the data was gathered in a non-controlled environment with no survey equipment, the locations of the shots with respect to the antenna were not precise. The angle measurements taken during the experiment were approximations. Also, the terrain was uneven leading to some shot locations being below or above the phase center of the antenna. This would make the assumed elevation of 0° incorrect by approximately $\pm 2^\circ$ for the majority of the tests. Second, the antenna was assembled using a temporary structure, and the microphones were affixed to the structure with tape. Due to the amount of give in the tape and the logistics of creating an easy-to-assemble/disassemble structure, the microphone spacing could only be approximated. These two problems presented by real-world implementation could have easily caused the increase in error for the tested prototype. In order to decrease the error, the final implementation of the goniometer would need to be made from precisely machined parts to ensure accurate sensor spacing.

4.4 Event Detection Simulation

Depending on the restraints of a given research project, event detection is one of the more difficult aspects of the acoustic goniometer design. While many publications on the topic are readily available, putting the theory into practice is never a simple matter. The simplest method is to analyze the signal from the sensors in search of peaks (minima or maxima) as defined by a predetermined threshold (thresholding: see Chapter 3.5.1). While effective at finding events, this method suffers from its susceptibility to false positive readings caused by noise and the inability to distinguish between events of interest. Even though the false positive detections can be mitigated using analog or digital filtering techniques, distinguishing between events is not possible for this method without either significant modification or very simple events whose distinguishing characteristics are well defined differences in amplitude. Another alternative is the application of some form of event fingerprinting (see Chapter 3.5.2). An implementation of such an algorithm could be simply the use of correlation between the signal and a known fingerprint for a predetermined event, or the algorithm could be more complex including the use of a neural network as in [26] and [28].

While the current research was developed to be adaptable for a wide range of applications, tests of the system focused on readily available sources of infrasound (e.g., door closings and gunshots). Thus, more complicated methods of event detection and classification (e.g., finger printing, Fisher statistic, etc.) were not employed. Due to the benefits gained in the areas of accuracy and characterization, a more complex event detection algorithm should be created for a system deployed for monitoring in a specific research application. The current research utilizes a simple thresholding scheme in

conjunction with a predetermined window length to identify events. The data used to test the algorithm is the same data acquired for the DOA simulations of Section 4.3.

Consequently, the thresholds are currently targeted toward detecting gun fire and would need to be adjusted should the system be deployed to monitor other types of events. For the data currently under test (gun shots), the simple thresholding technique has yielded 100% accuracy in low ambient noise conditions. In the presence of wind or other events producing sound in the same frequency range as the event of interest, the accuracy suffers dramatically, and many false events are detected (see Chapter 5 for more details).

CHAPTER FIVE: IMPLEMENTATION

The simulation results discussed in Chapter 4 provide reasonable proof of the current research's feasibility. However, an actual hardware implementation of the system is the most effective means of proving the design. While the simulation code was limited in every possible way to the constraints of an embedded system (fixed floating point accuracy, C implementable coding techniques, etc.), actual system implementation always poses unexpected challenges not encountered in the ideal environment of a simulation. The following sections detail the design of the acoustic goniometer firmware and hardware as well as the challenges encountered during the design process.

For the sake of discussing the design of the system, the acoustic goniometer can be split into two main components: firmware and hardware. Each of these can be further subdivided into smaller components/modules which carry out the task of collecting data, detecting events, and determining the location of event sources (DOA calculation). Figure 5.1 shows a basic block diagram of the acoustic goniometer components. The hardware acts as the conduit for the firmware to the outside world for both collecting information and providing feedback/data. Hardware includes the platform (processor and associated circuitry), the sensors, and the mechanical systems. The firmware collects, analyzes, and stores data provided by the hardware. To accomplish this, individual algorithm modules work in concert to schedule system tasks (operating system), manage/store incoming data (ADC Reader), and provide/record data analysis (event detection, event windowing, and goniometer state machine).

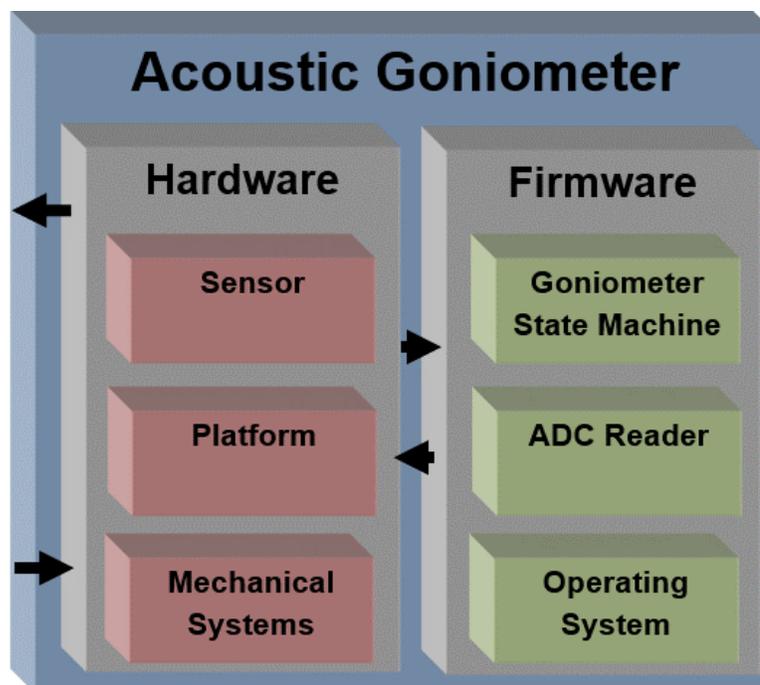


Figure 5.1 Acoustic Goniometer Implementation Block Diagram

5.1 Firmware

The design of the acoustic goniometer firmware was modeled in the simulation code. However, the two implementations differ in one very important way: the simulations ran on a computer analyzing static data stored in an array, while the hardware has to be capable of performing in real-time. As such, the simulations could be broken down into individual functions called at will by the researcher and stopped for debugging or adjustment at any point during the course of the simulation. The embedded implementation, however, had to be designed to run without interruption, switching automatically between tasks as events are detected all within a limited amount of time due to the real-time processing requirements demanded for the sensing task. Thus, although the simulations were able to prove feasibility and test potential algorithms, actual implementation of the goniometer on embedded hardware required careful

development of low-level hardware drivers, support devices, scheduling, and memory management.

5.1.1 Operating System

While creating a simple scheduler and developing low level hardware drivers for this project would have been possible, doing so would have required a significant investment of time. In an effort to ease development and spend more time on developing the acoustic goniometer, the decision was made to make use of an embedded operating system. Real-time operating systems (RTOS) for embedded systems can provide a reliable scheduling system, integrated hardware support, and a more intuitive application programming interface (API) than directly accessing processor registers. For this research, ChibiOS [44] was selected primarily due to its ease of deployment on the selected hardware. At the time of this project's inception, a port of ChibiOS for the chosen processor already existed. Thus, use of the operating system only required the creation of a board file to specify the pin configuration and available peripherals. ChibiOS provides a full featured scheduling solution in a preemptive multithreaded environment with 128 priority levels and configurable software timers. The operating system handles context switches in 1.2 microseconds and provides thread-safe memory allocations along with support for counting semaphores and mutexes [44].

Ease of deployment and standard RTOS features aside, numerous other attributes of ChibiOS made it attractive for use in this project. First, ChibiOS offers full direct memory access (DMA) support for the on-chip ADC's. This feature allows the hardware to fill the goniometer data storage buffer while the firmware processes data interrupting the system only once a sufficient amount of data is ready for processing. Second, a

complete File Allocation Table (FAT) file system library is available along with Secure Digital High Capacity (SDHC) support. Given the importance of data storage for the goniometer, these two features greatly simplified the process of saving data (both raw and processed) to the SD card. The support of SDHC cards further allows the use of larger capacity SD cards, which enables the raw data logging processor to run for longer intervals without the need for interaction on the part of the researcher. Finally, ChibiOS has low level drivers for all standard communication protocols, including Universal Synchronous Asynchronous Receiver Transmitter. The USART provided an adequate means for monitoring the goniometer's output in real-time.

Even though ChibiOS provides the aforementioned tools, using these tools to create a functioning system is still the responsibility of the firmware engineer. The useful features of the operating system must be arranged and managed such that efficiency is maintained and tasks run on a proper schedule. Modularity is also a concern for the sake of future modification and to maintain the ability to easily port the goniometer to more capable hardware as part of future research. The current firmware uses 6 threads to complete goniometer operations: *application (main)*, *data manager*, *scheduler*, *sensor management*, *storage*, and *communication*. These threads run on top of the device drivers (communication protocols, system timers, etc.) provided by ChibiOS. The modular design of the firmware can be seen more clearly in Figure 5.2. Modules resting on top of others in the figure represents dependence. Higher level modules (e.g., the application module) are dependent on those beneath them. The separation between layers allows the threads to be easily moved to other hardware as long as new device drivers with identical

APIs are written to function on the new hardware. Thus, portability is maintained by the firmware's modular design.

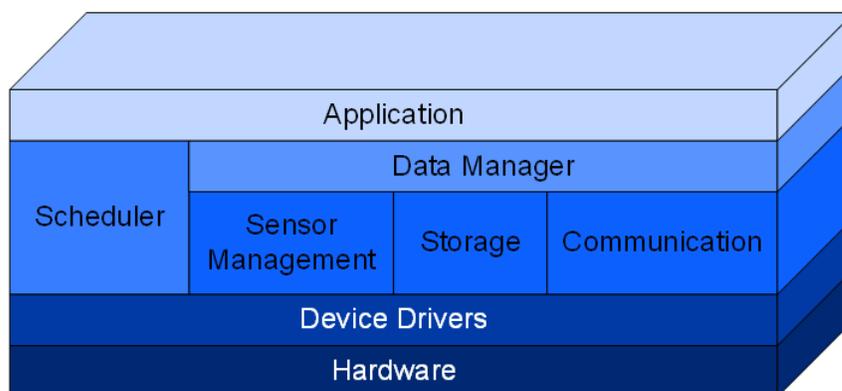


Figure 5.2 Firmware Organization

Each of the threads performs essential system tasks and is given priority based on its level of importance to the operation of the goniometer. The scheduler thread has the highest priority and is responsible for handling the system's timers and periodic tasks. The data manager and sensor management threads use the scheduler to register periodic tasks and depend on it to call time critical functions on time. The sensor management thread handles the sensor specific tasks for the goniometer. The system's ADCs are set up by this thread as part of an initialization routine to use the DMAs to fill the data buffers automatically at a rate of 10.6 kHz. The storage thread manages writing the data to the SD card, and the communication thread is responsible for processing any incoming data and sending outgoing data using the proper communication protocols. The data manager thread oversees the transfer of data from the sensors to the storage and communication threads. Finally, the application thread initializes all other threads and monitors their outputs for any errors. In the event errors occur, the application thread will

notify the researcher via several user interfaces (SD card system log file, LED flash pattern, and error messages on the system's debug USART).

The acoustic goniometer motherboard runs two different versions of this firmware on its twin processors (see Section 5.2 for an explanation of the hardware). One version of the firmware runs a simple data logging algorithm to record all raw data sampled by the sensors in order to allow for verification of the goniometer calculations independently of the embedded hardware. This version is referred to as the ADC Reader. The other version runs the acoustic goniometer sensor algorithms to detect events and determine the direction of arrival for the sound event. This version is designated as the Goniometer State Machine.

5.1.2 ADC Reader

The ADC Reader firmware is a simple data logger that takes raw data from the goniometer data buffers and writes them to the SD card. However, the simplicity of its task is a poor measure of the complexity of its implementation. The fast sampling rate of the ADCs creates several significant issues that had to be addressed in order for the system to function without jeopardizing the integrity of the data. The first issue is the time required to write data to the SD card. Writing data to an SD card takes a significant amount of time for an embedded system. Since each write operation has a certain amount of required setup and communication time, minimizing the frequency of the writing operations was deemed necessary. To this end, a minimum of 2 kB worth of data per channel (8 kB total) for each write operation was selected. Assuming a class 10 card is used on the current goniometer hardware, writing 8 kB of data can take more than 80 ms. During the time data is being written to the card, the goniometer cannot process incoming

ADC data. Thus, the system needed a method for storing at least 80 ms worth of data from all 4 ADC channels (assuming 8 kB data transfers), and the amount of data written to the card for each sample had to be minimized.

The problem of storing data accrued during SD card writes was the simplest to address but still required careful implementation. Maintaining the integrity of the ADC Reader data is of the utmost importance since an analysis of this data is the only means troubleshooting changes to the goniometer algorithm under laboratory/field test conditions and may be the only means of verifying the accuracy of the DOA calculation under conditions where the source location of a sound event is not known to the researcher a priori. As such, providing adequate buffer space to store incoming data during the SD card write operations is important. During the 80 ms used to perform an 8 kB write operation, a little more than 6 kB worth of data can be collected. Additionally, since other system processes may occur at the same time as the SD card writes, which can increase the time incoming data is not processed, that number only specifies the minimum amount of needed storage. Thus, 8 kB of storage was selected as the initial amount. Unfortunately, that amount of data is too large for a single array to fit in the normal program RAM space allocated by the linker script used by ChibiOS. Thus, the buffers were allocated to a special section of the processor's memory designated CCM (Core Coupled Memory) RAM. Unfortunately, data stored in the CCM RAM cannot be passed directly to the drivers provided for ChibiOS, which handle SD card writes. This necessitated copying data to be written to the SD card to an intermediate buffer before writing it to the card. Copying such a large amount of data requires a significant amount of time necessitating an even greater increase in the storage space. Consequently, the

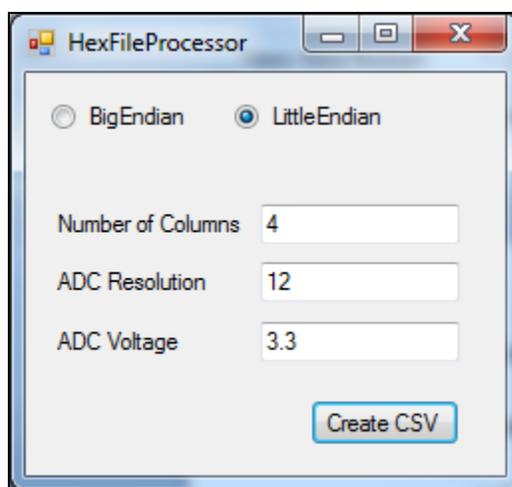
current goniometer storage buffer is a little over 56 kB. While this may seem like a ludicrous amount of data to store, testing the system with various sized buffers has shown this amount to be near the minimum for normal operating conditions. Once this was determined, the minimum size of data written to the SD card was revisited in the hopes of avoiding the CCM RAM. However, using smaller storage and more frequent writing operations was found to be an inadequate solution.

As mentioned previously (see Chapter 4), the sampling speed of the ADCs directly affects the accuracy of the acoustic goniometer. As such, a key goal of the design is to sample as quickly as possible. Thus, although ideally one would want to store the sampled voltages in a human readable file using ASCII characters, formatting the data is not possible since it wastes valuable processor clock cycles. In order to store the massive amounts of data gathered by the DMA sampled ADCs, the measurements are stored in a raw unsigned short integer format to a text file with no formatting (commas, spaces, carriage returns, etc.). The data is written in the order of sampling resulting in a scheme where every n^{th} short integer starting from the first represents a value from sensor 1, every n^{th} short integer starting from the second represents a value from sensor 2, etc (where n represents the number of sensors in the antenna). As an illustration of this order, Figure 5.3 (a) shows a representation of the data in human readable ASCII format using a standard hexadecimal representation of the sampled values. Unfortunately, raw data written to a text file is not easy to analyze since it is neither in a readable format nor a format which is recognized by any standard data analysis program. Thus, a simple raw data file processing program (known as Hex File Processor) was created to convert the raw data text files into a human readable comma separated value (.csv) format. The

program (shown in Figure 5.3 b) was designed with the flexibility of the target acoustic goniometer design in mind. The simple interface allows the researcher to select the endianness of their data (defaulted to little endian), the number of desired columns (number of sensors in the antenna; defaulted to 4), the resolution of their ADC (defaulted to 12 bit), and the maximum voltage (assumes 0V for minimum; defaulted to 3.3V maximum). Once the values are selected, clicking the “Create CSV” button opens a dialogue allowing the researcher to specify the raw data file, and a formatted data file is created with the same name. The converted file can be viewed in any program that recognizes the CSV extension and format. The processed data is represented in volts and can be analyzed using Matlab (as in the current research) or any similar mathematical analysis program. Figure 5.3 (c) shows a sample output file as viewed in Microsoft Excel.

0x1040	0x1045	0x1042	0x1044	0x1041	0x1044	0x1049
Sensor1	Sensor2	Sensor3	Sensor4	Sensor1	Sensor2	Sensor3

(a) Raw Data Represented In Readable ASCII Characters



(b) Hex File Processor: Raw ADC Reader Data to CSV Converter

	A	B	C	D
1	Sensor0	Sensor1	Sensor2	Sensor3
2	3.295972	3.295972	3.299194	3.297583
3	3.297583	3.298389	3.297583	3.297583
4	3.299194	3.298389	3.295166	3.299194
5	3.299194	3.296777	3.295972	3.299194
6	3.295972	3.296777	3.295166	3.299194
7	3.299194	3.293555	3.295972	3.298389
8	3.295166	3.29436	3.292749	3.298389
9	3.296777	3.299194	3.298389	3.297583

(c) Processed Data (viewed in Microsoft Excel)

Figure 5.3 ADC Reader Data Analysis

5.1.3 Goniometer State Machine

As mentioned earlier in this chapter, the main source of challenge associated with implementing the acoustic goniometer in hardware as compared to Matlab simulations is the real-time nature of the embedded implementation. The embedded system cannot wait for input from the researcher to validate its event detection or correlation results and must decide on its own when to move between steps in the calculation. Each step in the goniometer's process must wait to process data until it has sufficient/valid data. In turn, each step must either trigger the next in line or restart the process to detect and process future events. The process of determining the direction of arrival for a sound event on the embedded platform has been split into three steps (or states): event detection, event windowing, and DOA calculation. Figure 5.4 shows the states for the goniometer state machine. Once an event is detected, the second state stores a window of the data from each sensor. This window of data is then used by the third state to calculate the DOA for the given sound event.

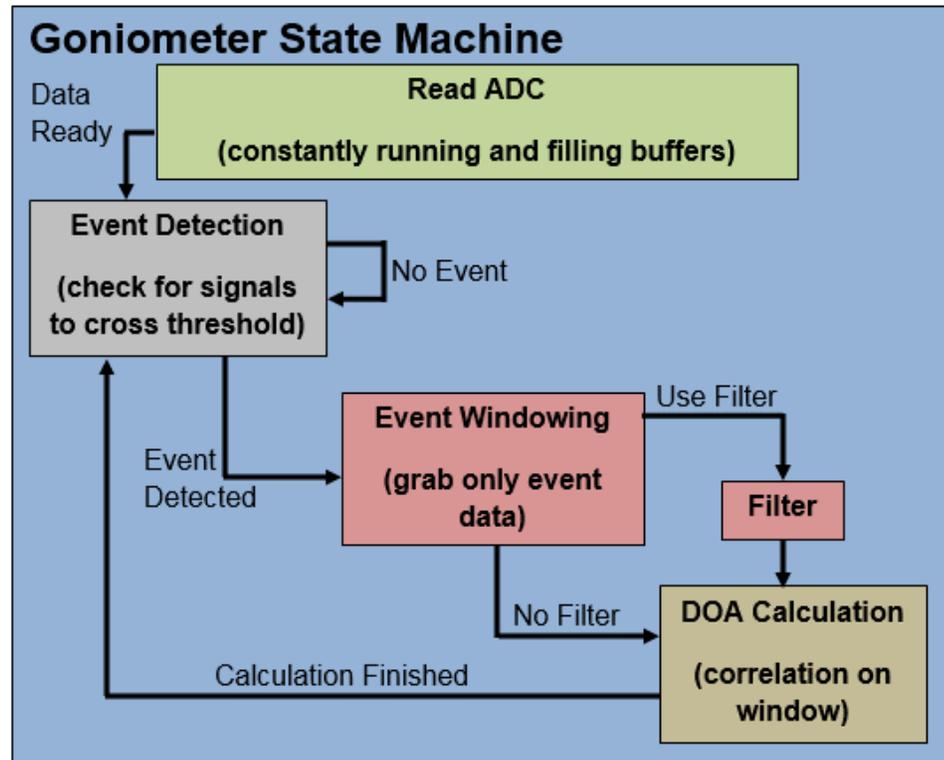


Figure 5.4 Goniometer State Machine

5.1.3.1 Event Detection

Every decision made during the design and implementation of an embedded system is a balance of tradeoffs, and the event detection stage is a shining example of this rule. This portion of the goniometer state machine had a considerable number of implementation options each with their own set of advantages and disadvantages. The simplest algorithm to implement is a threshold event detection scheme. The advantages of this design are its low cost in terms of processing time and the ease of implementation. However, this solution suffers from the inability to distinguish between complex events. Unless amplitude is the only feature that marks the difference between signals of interest, the threshold design cannot be used to determine what caused the sound event. Unfortunately, this is rarely the case. The second method considered, fingerprinting, can be used to differentiate between signals of interest. However, depending on the method

employed, the math used to search through raw data for a particular fingerprint can become quite complex and require an untenable number of processor clock cycles. As an example, consider the use of relative minima and maxima (peaks) for fingerprinting events of interest. The algorithm could employ a combination of amplitudes and the number of peaks to differentiate between events. A technique this simple could be implemented in hardware without a problem in the current system. However, a more complex fingerprinting technique operating in the frequency domain or performing real-time correlation on all raw data would not be a viable option. Since, the simple fingerprinting algorithm would only be effective if such features were the defining difference between events of interest, an algorithm whose complexity fell between these two extremes might need to be designed if a fingerprinting method was deemed necessary. The defining features that separate events of interest can change with each application/environment. Thus, specific applications for the acoustic goniometer must use the option that provides acceptable accuracy while minimizing the strain on the processor. Many options exist for creating such algorithms and should be explored for the sake of improving system performance and flexibility.

The final decision for testing the goniometer hardware was made with expediency and the capabilities of the hardware as the primary factors of concern. The event detection state is the only part of the goniometer state machine that processes all incoming data. As such, performing complex math operations as part of the event detection state was deemed inadvisable. Changing the processor to a more capable piece of hardware or adjusting the event detection state to subsample incoming data could allow for more complex algorithms to be employed. The current algorithm employs a

simple comparison to a threshold value stored on the SD card. Every measurement from the sensors is checked against the threshold set by the researcher. Once one of the sensors output values falls below the threshold, a flag is set which moves the system to the next state. Although simple in execution and lacking in event source selectivity, this method was proven effective in the Chapter 4 simulations.

5.1.3.2 Event Windowing

An event window in the current algorithms is split into three distinct parts as shown in Figure 5.5: Pre Window, Event, and Post Window. The event shown in the figure is registered when one of the sensor signals crosses the threshold (shown as the Mic2 signal crossing the dotted line). While crossing the threshold defines the event, this does not indicate the start of the event. In order for the correlation algorithm (see the discussion of the DOA calculation in Section 5.1.3.3) to be accurate, the start of the event must be included in the window. Thus, a section of the data prior to the event (designated as Pre Window) makes up the first part of the event. The event itself (crossing the threshold) is the second part. Finally, notable features of the event must also be included in order for the correlation algorithm to work. So, a “Post Window” of sufficient length to include at least the first minimum is included in the event window as well.

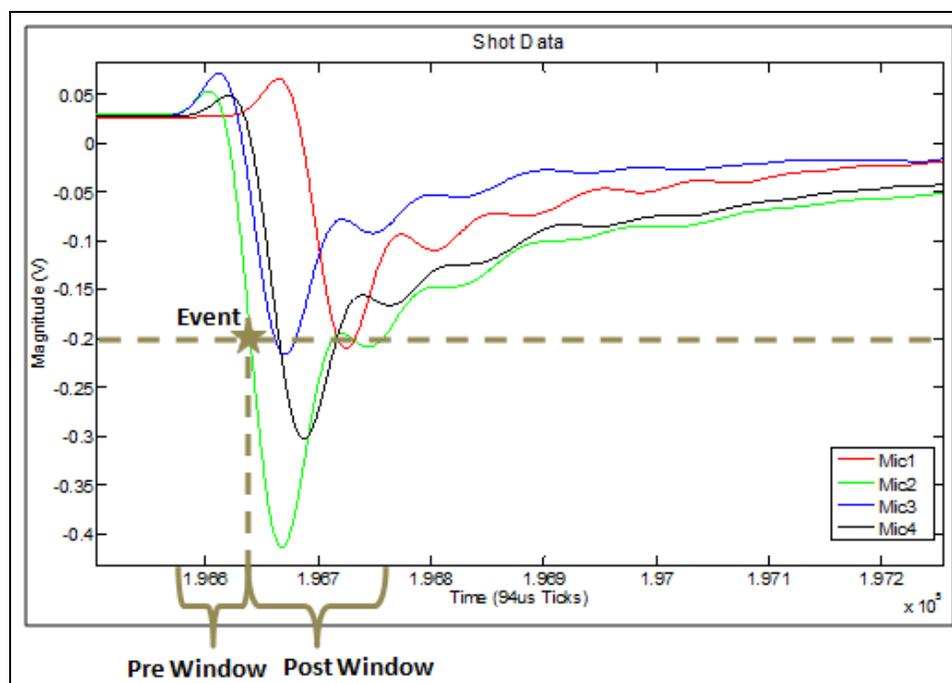


Figure 5.5 Event Window Parts

The event windowing state is the most lengthy in the goniometer state machine. Unlike the simulations, the firmware implementation of this algorithm is handicapped by the real-time data collection process. The simulations were made simpler by the fact that all of the data had been collected in advance. Further, all of the simulation data was held in a single multi-dimensional array. For the actual hardware, data had to be processed as it was collected in multiple single dimension arrays shared by all four sensors. Splitting the data across multiple arrays created several windowing scenarios that had to be handled by the embedded system. The best case scenario is one in which the sound event occurs somewhere near the middle of one buffer allowing for a sufficient number of samples before and after the event occurred to provide data for the full length of the event window. In this situation, all of the data needed for analysis can be analyzed with minimal indexing complications. However, given the size of the event windows needed for the test scenarios and the limited length of the sensor buffers, this case is somewhat

rare. A far more likely occurrence is the situation where data must be taken from two sample buffers in order to obtain a full window of the event. If the sound event occurs near the beginning of a sample buffer, data is needed from the end of the previous buffer to fill the event Pre Window. However, if the sound event occurs near the end of a sample buffer, data must be collected from the next sample buffer in order to fill the Post Window. Getting data from the next buffer has an added complication that the next buffer may not be ready. In this situation, the state machine must save its place and release the processor until data is available.

Taking so many possibilities into account requires careful handling of indexes and a significant amount of processor time. Data selected for a window must be copied from the sample buffers in order to free the buffers for storing the next samples. Additionally, all digital filtering must be completed during the windowing stage to keep the strain on system resources to a minimum. One could argue that, since each sample must be compared against the event threshold, filtering during the event detection stage makes the most sense. However, doing so would dramatically increase the number of computations as this would apply the filter to all of the data collected by the goniometer. By performing the filtering in the windowing state, filtering is only applied to data of interest, and the vast majority of the unused collected data is left unfiltered (reducing processor strain).

5.1.3.3 DOA Calculation

The DOA calculation portion of the state machine handles all calculations specific to the acoustic goniometer. While event detection and windowing could be applied to most event sensing applications, the correlation and angle calculations performed in this state are specific to goniometry. This part of the state machine is split into three sections.

The first of which simply calculates the averages (DC offset) of the sensor data in order to subtract it from each of the signals. Once the averages are determined, the second part of this state performs a correlation between the sensor pairs (as described in Chapter 4) to determine the difference in arrival times at each sensor. Finally, the third portion of the DOA state uses the equations described in Chapter 3 to calculate the azimuth and elevation for the sound event.

5.2 Hardware

The hardware used to prove the feasibility of the acoustic goniometer can be split into three basic parts: the platform, the sensor, and the antenna. The platform consists of the main processing unit and circuit board. A sensor includes the microphone and associated circuitry used to sense acoustic events. Finally, the antenna is the system as a whole including geometric layout and support structure. The next three sections discuss each portion of the hardware as well as any potential improvements.

5.2.1 Platform

The selection of a processing unit is one of the fundamental steps in any embedded system design. Given the strict/demanding processing requirements of an acoustic goniometer, an inexpensive general purpose microprocessor would not be adequate even if it included the requisite FPU. Specialized DSPs or more capable microprocessors could potentially provide the necessary resources and adequately meet the processing deadlines. However, they still impose timing stricter than ideal constraints and limit the possibilities of upgrading the system as technology improves (i.e., faster ADCs are developed). An FPGA provides a better solution for interfacing to faster ADCs since the interface would remain a parallel connection, and the only limitation on speed is

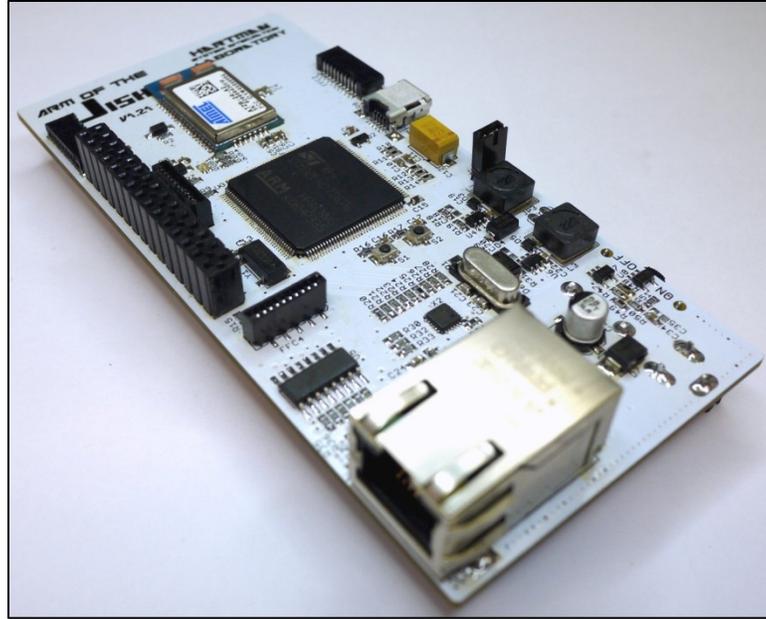
given by the ADC's maximum sampling rate. Unfortunately, FPGAs are not well suited to the high precision floating point calculations and trigonometry required for acoustic goniometry. Floating point and trigonometry IP cores are available but suffer from multiple problems depending on their cost. Free cores are inefficient, use up a large number of LEs, have poor accuracy, or suffer from a combination of these issues. More expensive cores solve these problems but increase the final cost of the design significantly. Thus, a better solution (as compared to an FPGA) would combine the sampling/processing capabilities of an FPGA with the precision floating point calculations of a DSP or processor with an FPU.

Several recently developed options exist that combine an FPGA and a capable processor (with FPU), but the best of these (as of this writing) for the acoustic goniometer is Altera's Cyclone V SX System on a Chip (SoC). This chip boasts a highly capable FPGA with a dual-core ARM Cortex A9 MP Core processor built into its silicon. Several wide parallel interface options exist to transfer data between the FPGA and the ARM processor. Furthermore, the ARM includes floating point hardware and is clocked at 800 MHz. Terasic provides a development board for the Cyclone V, which includes a number of peripherals to ease development including 1 GB of SDRAM for both the FPGA and ARM, a USB virtual communication port, and a gigabit Ethernet port. Such a system should provide more than adequate resources for developing a proof of concept acoustic goniometer design. The key benefit to using this FPGA in the acoustic goniometer design is the ability to offload large scale, time intensive tasks from the processor to a piece of hardware. In the current state of the research, the processor (or computer) is still responsible for the entire goniometry process from event detection to the final DOA

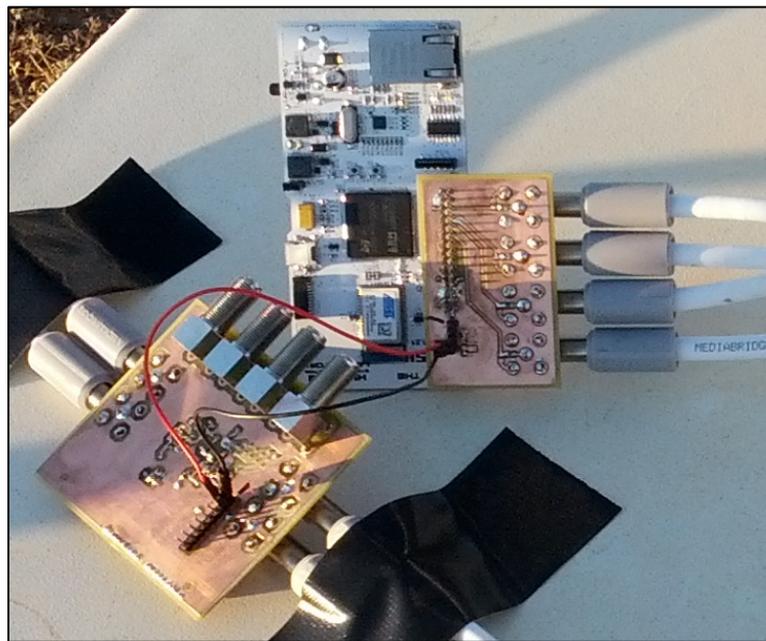
calculation. A design including an FPGA could at least make use of the FPGA for reading data from the high speed ADC and detecting events. These two tasks are simple enough for the FPGA to handle well, and handling them in hardware would keep the processor free for the more difficult calculations requiring floating point math (e.g., filtering, cross correlation, and DOA calculation) as well as data recording. Alternatively, given the simplicity of the cross correlation method currently selected and assuming a more complex algorithm is not required, correlation could be handled by the FPGA as well. Doing so would free up more processor resources and could allow the system to respond faster to events. While all of the benefits mentioned for this platform choice may seem attractive, FPGAs/SoCs have one very important negative aspect. The use of an SoC requires knowledge of a hardware description language (HDL) in addition to the ability to write firmware for a microprocessor. This added requirement creates a significant barrier to research insofar as it would limit further development of the acoustic goniometer's algorithms to researchers experienced in writing HDL (e.g., computer and electrical engineers). Thus, although an SoC or FPGA may provide many useful design features, the use of either platform would negatively impact the primary goal of lowering barriers to research in the field of acoustic goniometry.

The circuit board shown in Figure 5.6 (a) was the first hardware platform used by the acoustic goniometer. Figure 5.6 (b) and Figure 5.6 (c) show the motherboard connected to a sensor interface board and the block diagram for the motherboard, respectively. The board was designed by the Hartman Systems Integration Lab (HSIL) at Boise State University as part of their environmental sensor system research. Although the design was intended for other uses, the designers made it capable of general purpose

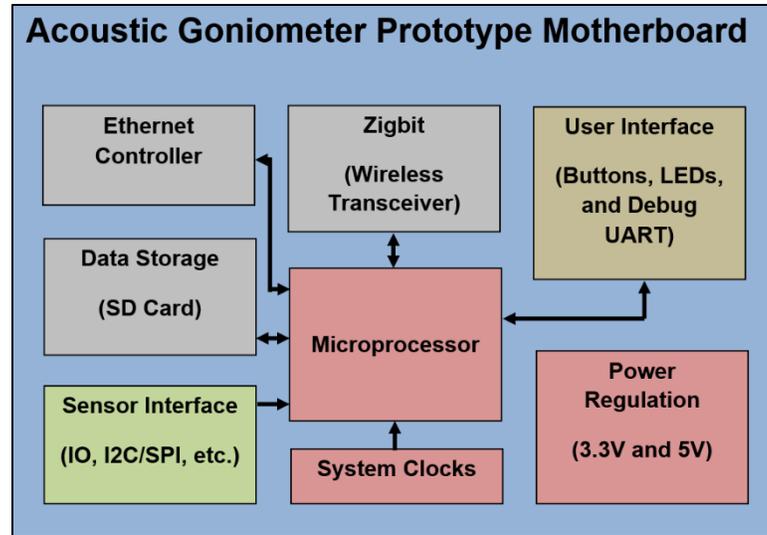
hardware prototyping. The system has a plethora of features to aid in development, but the most useful for the current research were the high speed processor (up to 168 MHz clock speed) with an FPU and high speed internal ADC (12 bit resolution at up to 2.4 MHz sample rates), the multipurpose pins available on the breakout header, the power regulation, and the secure digital (SD) card. The board was used to record the data collected by the infrasound sensors (see Section 5.2.2) for many of the simulations described in Chapter 4. The firmware used for this research included some custom work but depended largely on the development done by the HSIL and the real-time operating system used by the board (ChibiOS). The custom firmware took samples from the 4 ADC channels with 12-bit resolution at a rate of approximately 10.6 kHz and recorded them to the on board SD card. Although other rates were technically possible, slower sampling would have been insufficient to the task, and faster sampling rates caused instabilities in system operation resulting in data corruption. If this platform were to be used more extensively to test the effect of higher sample rates on goniometer accuracy, a faster system clock and more custom firmware would be required in order to achieve the faster sample rates.



(a) Prototype Motherboard



(b) Interface Boards Connecting Motherboards to Goniometer Sensors

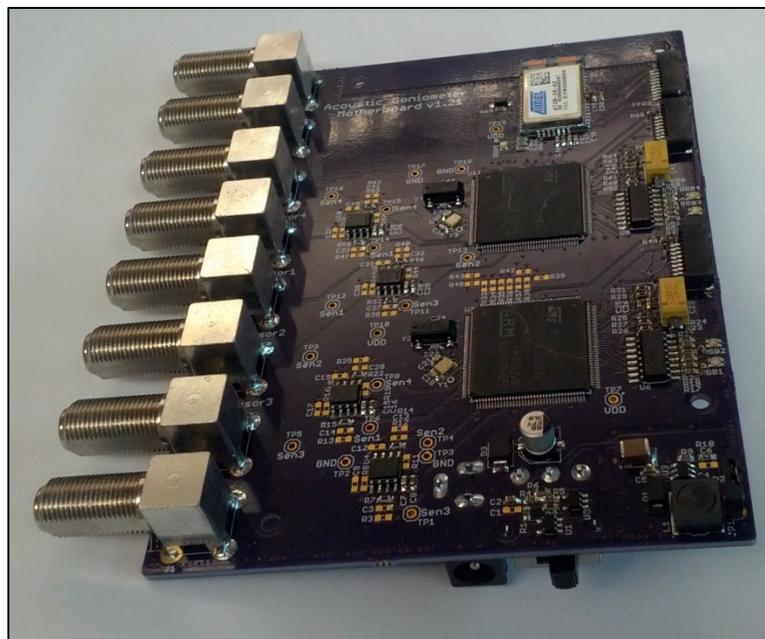


(c) Motherboard Block Diagram

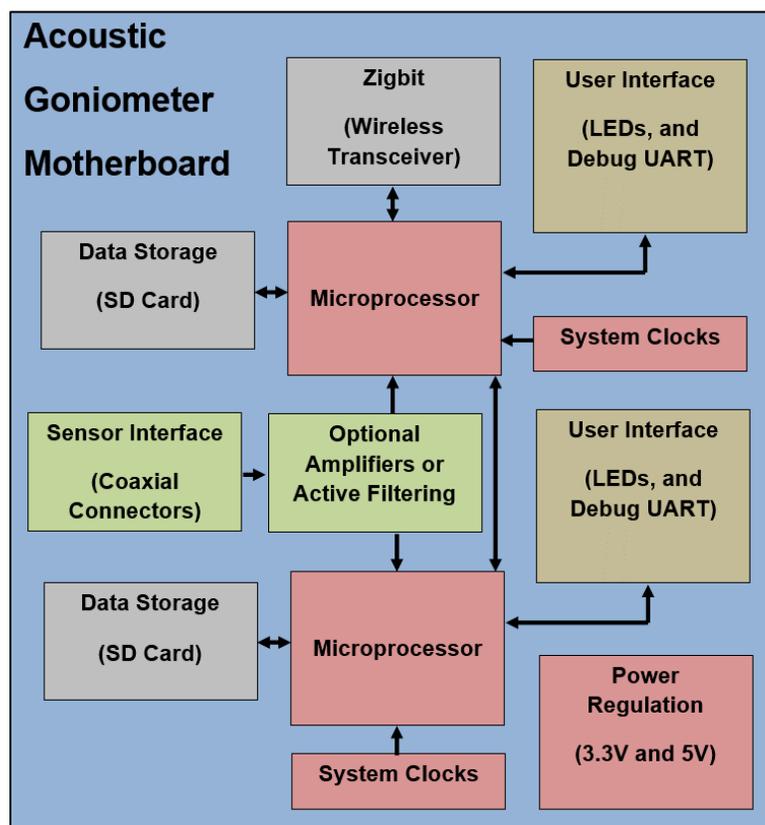
Figure 5.6 Acoustic Goniometer Prototype Hardware Platform

A complete implementation of the acoustic goniometer has been realized using the processor from the first prototype (see Figure 5.7). The ARM M4 CPU is more than capable of handling the goniometer calculations for most low frequency applications. However, due to timing constraints and the desire to record the raw data, the final design includes 2 processors: one to handle goniometry calculations and one to handle data storage. Writing data to a storage medium is a lengthy process (by microprocessor standards) even for small amounts of data and is even more cumbersome for the vast quantity of data recorded by the goniometer (see Section 5.1.2). While the processor is busy communicating with the storage medium, little else can be accomplished. Consequently, since the goniometer calculations require a significant amount of processor resources, one microprocessor cannot handle both tasks efficiently unless a more capable/expensive processor is selected. Mitigating the impact of data storage in order to implement the system on a single processor is possible but not advisable at this

stage of the research due to the sacrifices which must be made to bring such an implementation to fruition. Instead of recording all raw data, the goniometer could subsample the raw data thus storing only a small portion. In theory, a goniometer interested in the infrasound range could easily sample the data at rates as low as 200 Hz and still be able to reproduce the signals of interest. This would reduce the number of stored samples by a factor of at least 50. Such significant savings could allow the system to operate with a single processor. However, such an algorithm would significantly increase the amount of error present in the DOA calculations. While the signals could be reproduced (in theory), the key requirement of operation for the DOA calculations is the determination of the difference in TOA at each sensor. This ability is predicated on the ability of the system to accurately determine the TOA for an event. As the sample rate decreases, the acoustic goniometer's TOA determination becomes less accurate and the overall performance of the system suffers. Additionally, in order to validate the simulations and verify the functionality of the hardware, the raw data from the sensors must be collected and analyzed. Once the goniometer calculations have been proven satisfactorily accurate, this algorithm change (or similar ones) could potentially be used to reduce the system to one microprocessor as long as the lowered system accuracy was acceptable for the given application. As another option, the recording algorithm could be made to store data only for events further reducing the number of required writing transactions with the storage medium. This could mitigate the cost of storing data sufficiently to allow a single processor to handle the full goniometer implementation without negatively impacting the performance of the system. The sacrifices made to realize this system should, however, be noted.



(a) Final Acoustic Goniometer Motherboard



(b) Motherboard Block Diagram

Figure 5.7 Acoustic Goniometer Motherboard

While the dual processors do increase the cost of the system, the added utility and expandability they provide should not be discounted. Although the previous discussion suggests ways in which the system could operate on a single processor, the intent was not to preclude researchers from exploring the idea that a dual processor system enables the possibility of cooperative processing. The hardware has been designed such that 3 communication protocol signals connect the two processors (UART, I²C, and SPI). This feature was added to allow for data sharing between the two processors with the idea that cooperative parallel computing techniques might be a beneficial avenue to explore.

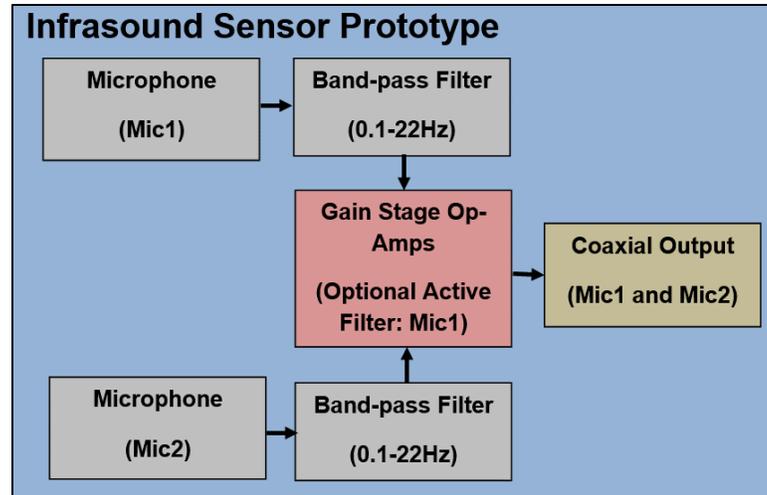
In addition to the twin processors, the current Acoustic Goniometer Motherboard design includes two other features of interest. The first is the reconfigurable voltage followers that connect the sensor signals to the processor ADCs. These were designed to allow for active filtering or provide additional gain as demanded by a given application. In the default configuration, they merely provide a high impedance buffer to prevent the sampling process from affecting the signals. The second feature of interest is the inclusion of a Zigbit radio. The radio uses the Zigbee communication protocol allowing it to form wireless mesh networks. This feature makes the system capable of being used as part of a wireless sensor network for more ambitious, large-scale projects. Wireless communication could provide a way to confirm events, pinpoint locations of events, and apply data fusion techniques with other sensors to gain more complete environmental data.

5.2.2 Sensor

Due to the wide array of applications requiring microphones and the availability of many publications on sound sensor design (e.g., [13], [18], etc.), creating an acoustic

sensor for a goniometer can be a fairly simple task. The primary consideration is the target frequency range as determined by the acoustic signature of the desired event. Once the range of interest is known, a microphone may be selected with an appropriate frequency response, and any filtering and amplification circuits can be designed. The only potential complications that arise in the course of this process are caused by limitations imposed by the system specifications (e.g., cost, sensor adaptability/flexibility). Since the current research goals required the goniometer to adapt to a wider range of applications while maintaining as low of a cost as possible, the sensor design process included a few extra challenges.

The most difficult and important part of designing a flexible acoustic sensor was the selection of the microphone. Given that the frequency range of interest for the bulk of this work was in the infrasound range, the option of picking a specialized infrasound microphone was tempting. However, taking this shortcut would have significantly increased cost since these microphones are typically sold for \$1000-\$2500 each (as of the writing of this document). Aside from this issue, picking specialized microphones could easily have led to an implementation dependent on specialized microphones for each application. Thus, the decision was made to use a simple electret condenser microphone with a wide frequency range. Although inexpensive electret microphones do not typically have a flat frequency response in the infrasound range, other researchers indicated that such devices could be used with reasonably accurate results in infrasonic sensing applications [13-17]. Since a specific frequency range of interest had been defined for the current research, the only other drawback to choosing the inexpensive microphone option was its response to a wide range of frequencies. However, this problem is easily solved



(b) Infrasound Sensor Block Diagram



(c) Infrasound Sensor Board

Figure 5.8 Initial Prototype Infrasound Sensor

Consider the signal produced by the microphone designated as “MIC JP1” on the left side of Figure 5.8 (a). In the current configuration depicted by Figure 5.8 (a), the signal from the microphone passes through a simple RC band-pass filter before being amplified. The upper and lower cutoff frequencies selected for the filter are approximately 22 Hz and 0.1 Hz, respectively. The frequency cutoffs were determined using previous research which indicated this to be within the desired range for certain

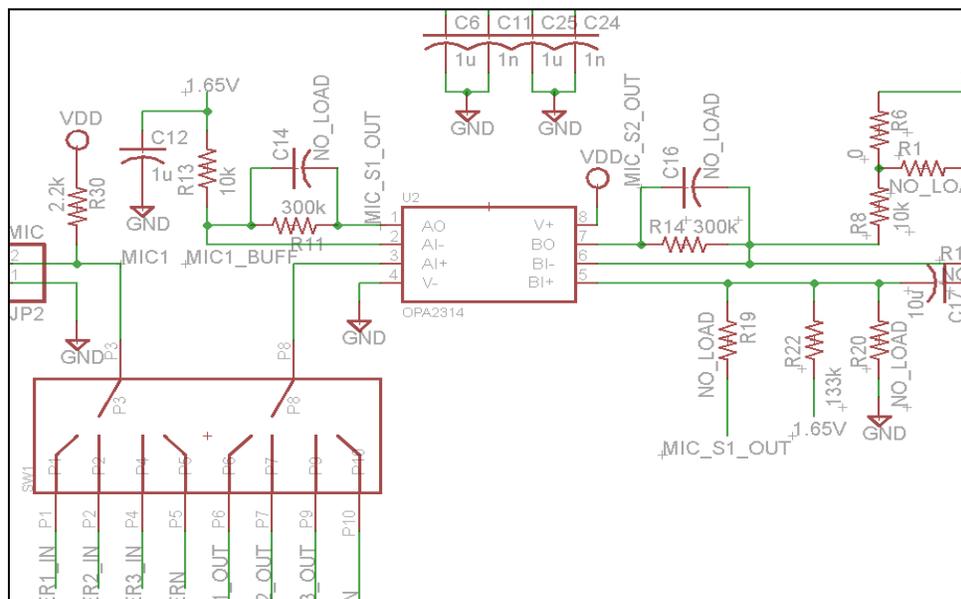
types of infrasound monitoring [13-17]. However, one should note that the frequency response of the microphone extends up to 22 kHz, and the sensor could be used to monitor other acoustic events above the infrasound range by simply adjusting the values of the RC filter to move the cutoff frequencies to the desired range. Another point of interest in this part of the circuit is the constant 1.65V DC offset applied to the signal via the resistor of the high pass filter (see R24 Figure 5.8 a). In an effort to make the design as simple as possible, this offset was added to keep the signal from clipping at the negative rail without the need for both positive and negative supplies on the operational amplifier. After passing through the filtering and being given the offset, the signal is amplified and finally sent to the ADC on the processor. The gain was set to 30 but can be adjusted easily by changing the resistor values. A simplified explanation of the circuit's function can be seen in the block diagram (Figure 5.8 b).

Although complete in detail of the infrasonic sensor circuit used by the initial prototype, the description in the previous paragraph only discussed half of the schematic shown in Figure 5.8 (a). The other half is mostly just a mirror image of the schematic described for "MIC JP1." Aside from the desire not to have the inputs of the second operational amplifier in a dual amplifier package floating, this design serves two other purposes. The first is to provide the ability for minor sensor data fusion. The inclusion of a second microphone in the design in close proximity would allow the system to better guard against erroneous event detections or aid in detection and time of arrival determination [16]. If one sensor fails or does not respond clearly to a particular event of interest, the other may act as a backup or provide more useful data if it has a better response. The second purpose of this second circuit is to allow for more advanced

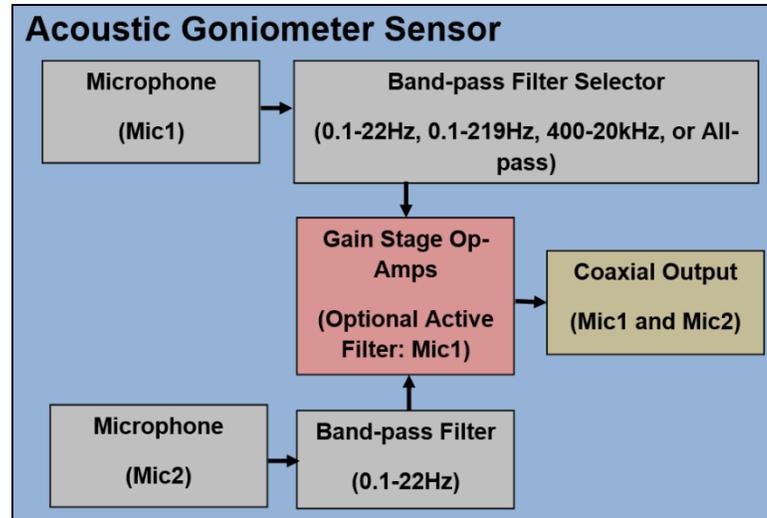
filtering on the “MIC JP1” circuit. By changing a few configuration resistors, the second microphone can be bypassed, and the second operation amplifier can be used to create an active filter for the first microphone signal. The sensor was designed similarly to the acoustic goniometer as a whole with the idea in mind to allow for adaptability and deployment for a wide range of applications.

While the initial prototype served its purpose admirably and helped prove the feasibility of the design, the sensor needed several improvements before the implementation could be considered finalized (see Figure 5.9). One such point of improvement to increase the sensor’s flexibility was the filtering system. As mentioned previously, the initial prototype acoustic goniometer used fixed hardware filters to remove undesirable noise and select the frequency range of the microphone. These filters could be adjusted by changing resistor values and configurations given a little effort and a certain amount of soldering ability. However, such adjustments would be difficult for a researcher not versed in electrical engineering concepts and could be made easier by making adjustments to the initial design. The possibility of removing all analog filters in favor of completely digital implementations was considered. This option appears attractive since it allows for the most flexibility. However, doing so would mean moving the filtering firmware from its current location in the goniometer state machine (after event detection) to the raw data processing stage in order to provide the frequency selectivity currently handled by the analog filters. As such, the processor would have to perform the filtering operation on all raw data dramatically increasing the number of required floating point operations and negatively impacting the responsiveness and performance of the goniometer. Thus, the microphone hardware was modified to include

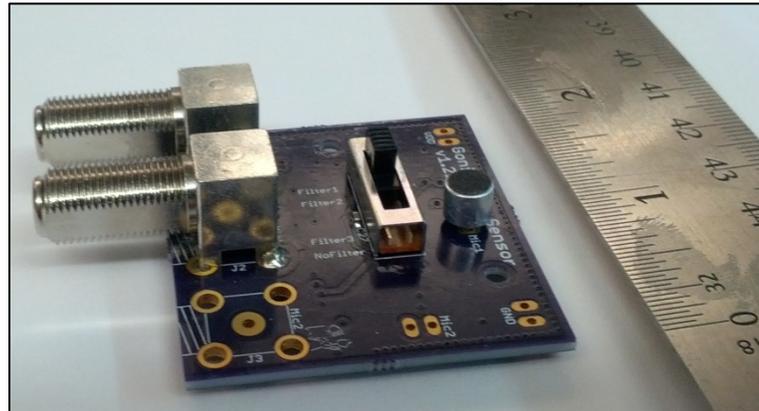
an analog filtering system with selectable frequency ranges. In order to make the system more customizable for varying research needs, the method of filter selection chosen was a simple slide switch. The final design shown in Figure 5.9 (a) is quite similar to the schematic for the initial prototype shown in Figure 5.8 (a). The only difference is the inclusion of a double pole four throw switch, which replaced the band-pass filter for the “MIC JP1” signal. The two poles of the switch connected to the microphone and the input of the operation amplifier can be connected to the inputs and outputs of 4 analog filters with pre-determined ranges. The purpose of the analog filter is not to provide a perfectly clean signal but only to limit the frequency range of the microphone. Thus, pre-determined ranges can be safely used without damaging the flexibility of the system. The default configuration of the sensor includes 3 band-pass filters and 1 all-pass filter. The ranges on the band-pass filters are set for low frequency infrasound (0.1 Hz to 22 Hz), wide range infrasound (0.1 Hz to 219 Hz), and audible frequency (400 Hz to 20 kHz). A simplified explanation of the final sensor design can be seen in Figure 5.9 (b).



(a) Final Sensor Schematic



(b) Final Sensor Block Diagram



(c) Final Sensor Board (~2x2 inches)

Figure 5.9 Final Prototype Acoustic Goniometer Sensor

5.2.3 Mechanical Systems

While the bulk of this research is focused on the hardware and firmware development, the design of an acoustic goniometer requires a significant amount of mechanical design as well. In order to test the hardware and firmware, a sensor configuration including the number of sensors, the geometry of the antenna, and inter-sensor spacing had to be selected. Additionally, a practical structure had to be created for sensor deployment. Enclosures also had to be created to protect against environmental

hazards damaging the goniometer as well as to prevent interference with data acquisition and analysis.

5.2.3.1 Antenna Design

Designing an antenna for an acoustic goniometer is a challenging problem dependent largely upon the application of interest. The key aspects of an antenna design which must be addressed are the number of sensors, the geometry of the antenna, and the spacing between each sensor. Other points of concern are minor and are usually seen with any deployed sensor system (e.g., case design, material used for structure, method of mounting, etc.).

Determining the number of sensors required for an acoustic goniometer is a fairly straight forward task. If the device is to be used in a study only concerned with two dimensional space (no elevation), the minimum number of sensors is 3. However, if elevation is of interest (as with the current research), at least 4 sensors are required. These rules supply a minimum number, but more sensors may be added to improve the reliability of the data or the accuracy of the calculations. Additional sensors may be used to provide alternative sensor pairs if correlation between some sensors does not provide trustworthy results (see example in Section 4.2, the analysis of Figure 4.9). According to Van Lancker, depending on the processing method, an increased number of sensors can be used to improve the signal to noise ratio of the cross correlation calculation [13]. However, an increase in the number of sensors makes antenna deployment more challenging, increases the cost of the goniometer, and comes at a cost in processing time due to the increased number of floating point calculations required to include more sensor

pairs. Thus, the goal of any antenna design should be to use the minimum number of sensors required to maintain the desired system performance.

In his master's thesis, Van Lancker researched antenna geometry for acoustic goniometers extensively [13]. According to his work, antenna geometry for a given goniometry application is determined by the characteristics of the source to sensor configuration (i.e., the expected angles at which the sound events will be received by the antenna). For example, the most important factor concerning the design of the acoustic sensors that affects the design of the antenna is microphone directionality. If the microphones are directional, the antenna geometry must be selected with this in mind. When an expected direction of arrival is known for an application, directional microphones can be used to increase the performance of an antenna geometry that favors the angles of interest. However, if such is not the case, the antenna geometry must make up for the directionality of the microphones in order to avoid favoring particular DOA calculations with greater accuracy than others.

As with the other two aspects of antenna design, the research performed by Van Lancker provided useful insight into the selection of inter-sensor spacing. Assuming the limited resolution of an actual hardware implementation is not imposed (i.e., infinite precision sampling and calculations are used), the initial assessment may be made that larger inter-sensor spacing should always produce more accurate results since larger distances between sensors produces greater differences in arrival times at sensors improving the effectiveness of correlation calculations. However, this is only the case to a certain extent. In order to satisfy Nyquist and avoid anomalous data, the sensor spacing must be less than half the wavelength of the acoustic event of interest [13].

5.2.3.2 Structural Design

Since the purpose of the prototype was to prove the feasibility of the system in general, the antenna geometry was selected with ease of implementation in mind. The antenna design used by the current research was modeled after the geometry explained by Figure 4.2. Four sensors were attached to a PVC frame in a regular tetrahedral pattern with an approximate spacing of 2 m. The fully assembled antenna can be seen in Figure 5.10. The PVC frame was designed to make assembly and disassembly as simple as possible in order to allow for rapid deployment and testing of the prototype. In designing for ease of deployment, certain sacrifices were made in the realm of durability and system performance. In the current setup, the sensors are held in place by tape at measured intervals along the PVC frame. The measurements must be done at each deployment and are not exact. Further, give in the tape allows for small movements in the sensors after they are placed. Another problem with the spacing can be seen clearly in Figure 5.10. Notice that sensor 4 at the apex of the tetrahedron is taped to the PVC on the side closest to the camera. This places the sensor closer to sensor 2 (see sensor on the right side of the figure) than sensor 1 (rear sensor in the figure). Although each of the mentioned problems with sensor spacing is minor, they still introduce a source of error that could be removed with a better antenna design. As for the durability of the antenna, PVC is fairly flexible and the joints were not sealed. Consequently, the structure was able to sway and bend in the wind going so far as to move in significant gusts. While this was acceptable for the testing phase, a more solid material than PVC with more exact sensor spacing should be created for the final design. A more rigid material should be selected,

and the antenna should be anchored to the ground with either stakes (for a less permanent deployment) or concrete (for a more permanent solution).

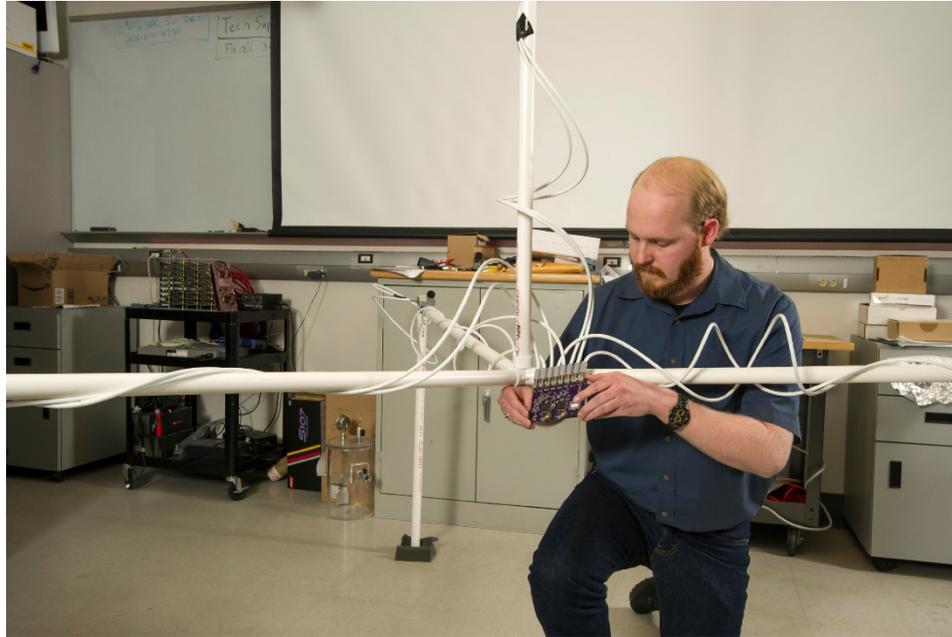


Figure 5.10 Acoustic Goniometer Antenna Prototype

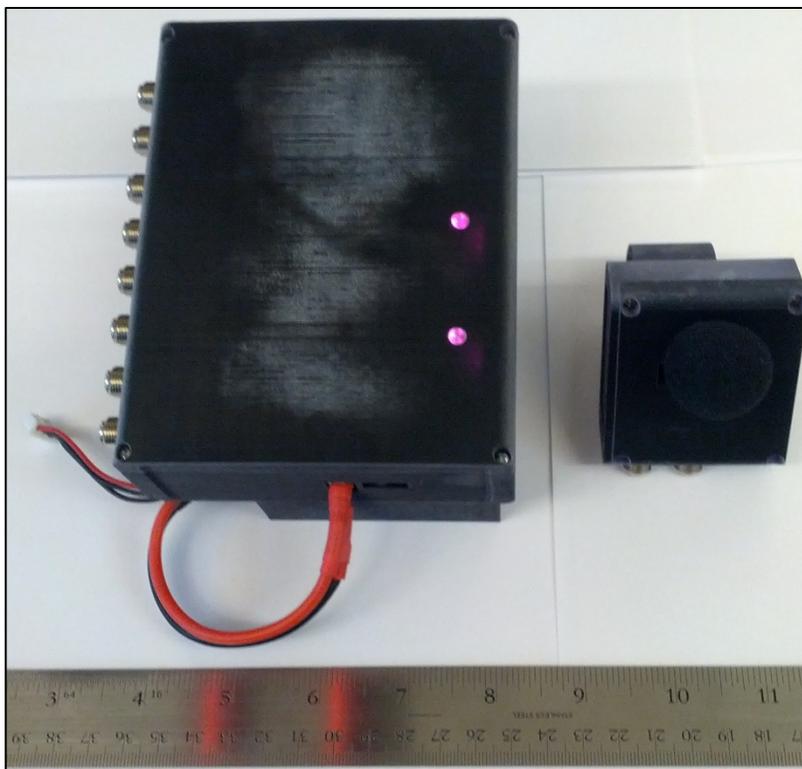
5.2.3.3 Enclosure Design

In addition to needing a structure for supporting the sensors, the acoustic goniometer design further required enclosures to protect the sensors and motherboard from the elements. While not the ideal solution for outdoor environments, stereolithography (SLA) was used to print prototype cases for the acoustic goniometer. Figure 5.11 (a) shows the first prototype enclosures, and Figure 5.11 (b) and Figure 5.11 (c) show the second generation enclosures. The sensor board enclosure (Figure 5.11 a) is slightly more complex than the motherboard enclosure (Figure 5.11 c) as it has more of an impact on the system's performance. Both enclosures include outlets for the coaxial connectors as well as cutouts for user interface accessibility (e.g., frequency selection slide switch on the sensor board and SD card cutouts on the motherboard). Each of the

enclosures further includes mounting brackets designed to fit the PVC structure designed for the current prototype. However, the sensor board enclosure also includes a small wind screen to help combat noise generated by low speed winds (less than 10 mph). A small tube (not shown in the figure) extends both into the case to enclose the microphone and outside the case to provide a mounting point for the windscreen. Another key difference other than size (shown in Figure 5.11 c) is the battery pocket on the back side of the motherboard enclosure. The only difference between the two generations of enclosures is the material selected. The second generation enclosures are still 3-D printed but are made from a more flexible material in order to prevent breakage caused by rough test procedures. The enclosures have been tested and proven reliable for a prototype but are not without drawbacks, which would have to be addressed in a more permanent solution.



(a) Sensor Enclosure



(b) Motherboard Enclosure (left) with Sensor Enclosure (right)



(c) Motherboard Enclosure (side view)

Figure 5.11 Acoustic Goniometer Enclosures

The SLA material suffers from several important drawbacks. First, cases printed in this fashion are expensive to obtain. Second, the material properties are not ideal and

change over time and exposure to various environmental factors. The material when first printed is semi-rigid and breaks easily. Over time, the material continues to become more rigid and brittle. This process is accelerated when the material is exposed to any form of UV light. Consequently, this material is unsuited to outdoor applications in the presence of sunlight. What little flexibility the material has presents a problem to the goniometer in the form of increased wind sensitivity. The enclosures shown in the figure were tested on a fairly windy day (15 mph wind with gusts above 20 mph) with wind screens in place. Figure 5.12 shows the raw data collected by the processor programmed with the ADC Reader firmware. Even with the wind screens, the enclosure's material was just flexible enough to allow it to act like a speaker diaphragm in the presence of high speed wind (amplifying the infrasonic noise due to wind). The gun shots from this data cannot be separated from the wind noise since the frequency and amplitude of the signal generated by the wind is similar to those of the gun shot signals. The only positive feature of this type of enclosure is the quick turnaround of custom designs. This allowed multiple prototypes to be quickly developed for basic system tests. However, a more suitable material (e.g., more sturdy, weatherproof, windproof, etc.) should be selected for any long term deployments.

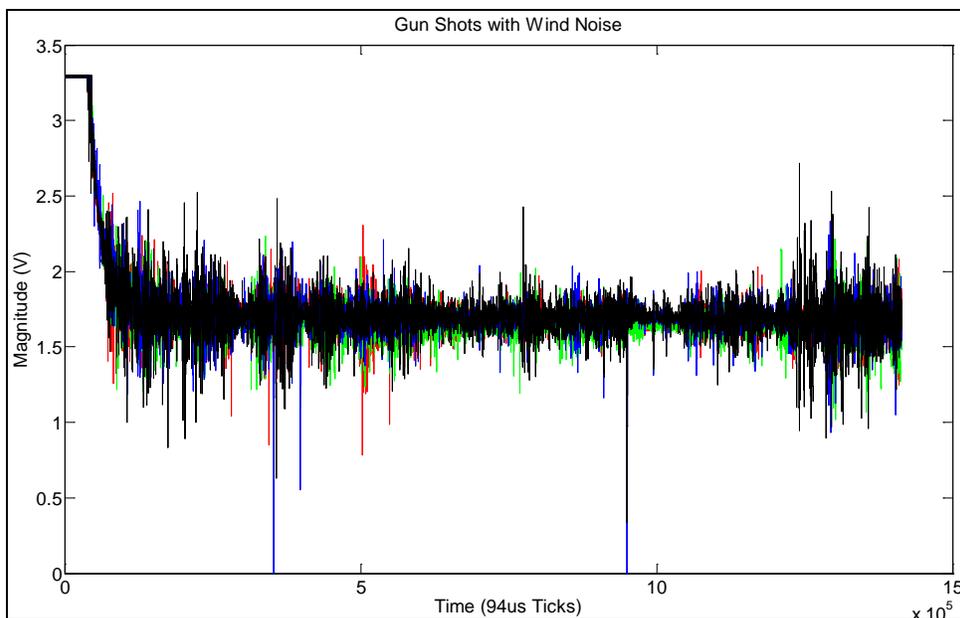


Figure 5.12 Gun Shot Data with Noise in Windy (Gusts at 20 mph) Conditions

As mentioned previously, wind noise is a significant point of concern for the acoustic goniometer. Since wind can generate noise in any of the frequency ranges of interest, any outdoor application could potentially suffer from this problem. Removing wind noise via filtering (digital or analog) is a difficult task. The signal generated by the wind is often the same frequency as the signal generated by the event of interest. So, differentiating between an actual event caused by the phenomena under study and one caused by a gust of wind is impossible without additional information provided by a more complex suite of sensors. Many of the authors of the previous research discussed in Chapter 2 either discussed the issue of wind noise as part of their research or made combating wind noise the focus of their studies (e.g., [23] and [27]). Most of those who detailed their attempts to solve this problem resorted to various enclosure modifications ranging from complex wind screen designs to simple commercial products or even burying their sensors. Given that development of complex wind negating systems lay

outside the scope of the current research, mitigating this issue was explored as part of the enclosure design process using simple commercially available solutions and custom acoustic foam structures instead of attempting more complex solutions. Figure 5.13 shows some of the wind screen designs developed for the acoustic goniometer.

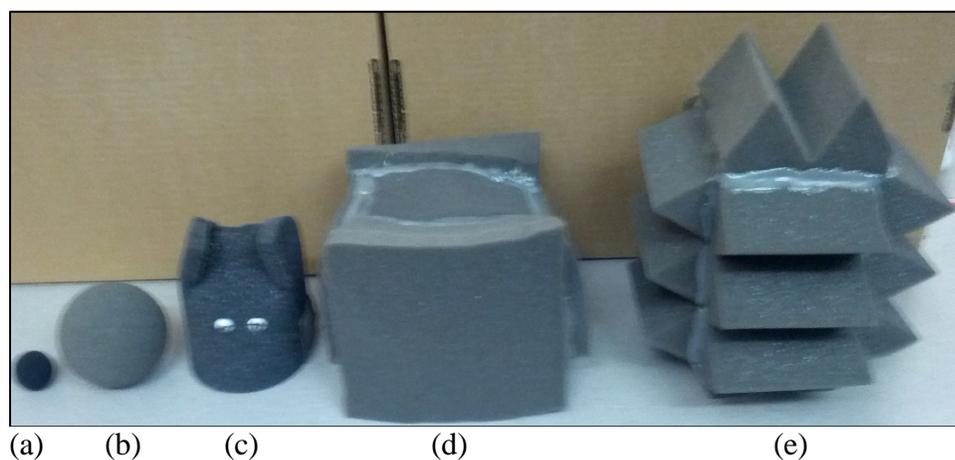
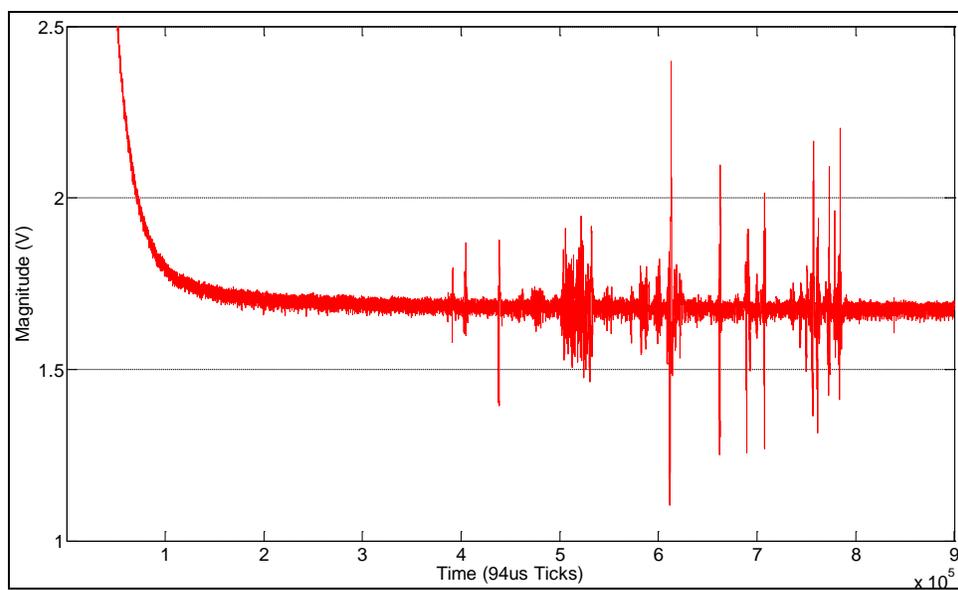


Figure 5.13 Commercial and Custom Wind Screens

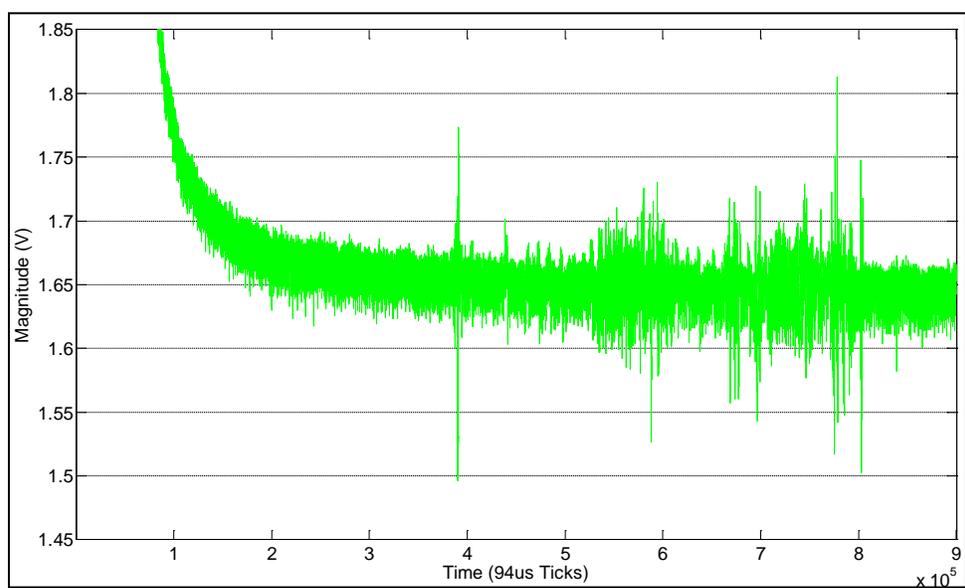
Two of the windscreens in Figure 5.13 are commercial products used for microphones in music applications (a and b), and the other three are custom built foam enclosures made from two types of acoustic foam. The two on the far left of the figure (a and b) are 2.5 cm and 6.5 cm foam ball windscreens designed to fit small microphones. The SLA tube on the case (see discussion of Figure 5.11 a) provides a mounting point for these wind screens and is used to isolate the microphone from the rest of the case. Since the case is not airtight and can act as a speaker in high speed winds, the isolation of the microphone is an important feature in achieving the best results with any windscreen. One weakness of the foam balls is that they cover only the opening to the microphone. Although covering the microphone opening does keep wind from directly influencing the microphone, this does not protect the microphone from high speed winds using the case

as a speaker. Thus, the other three wind screens were designed to fit over the entire goniometer sensor enclosure.

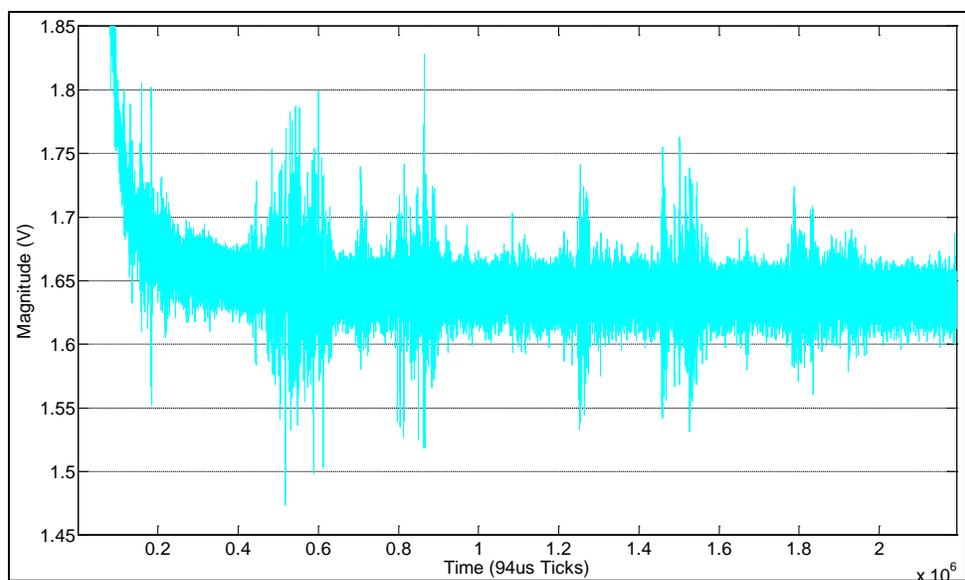
The results of testing the windscreens in a laboratory environment can be seen in Figure 5.14. Both of the foam balls perform well in low speed winds, but the larger ball (Figure 5.13: option b) performed marginally better in higher speed winds (greater than 20 mph). The next three windscreen options to the right of the balls in Figure 5.13 were constructed to combat high speed winds. The one in the middle with the eyes (c; 7 cm cube) is made of a thinner, more porous foam than the other two (d and e). Although it did perform marginally better than the foam balls (a and b) in high speed winds, this windscreen did not reduce wind noise enough to allow the goniometer to work effectively in high speed winds. The two windscreens on the right side of the figure (d and e) are made from the same type of acoustic foam as one another. Option d (14 cm cube) has the ridges facing in toward the microphone, and option e (14 x 14 x 18 cm) has the ridges facing outward. A comparison of Figure 5.14 (d) and (e) shows that the design with the ridges facing in (d) is the superior windscreen. The windscreen with the ridges facing outward (e) appeared to perform worse than even the large foam ball (b) and could not be used to effectively reduce wind noise. However, the windscreen with the ridges facing in toward the microphone showed promise with only one spike being capable of causing erroneous events to be detected.



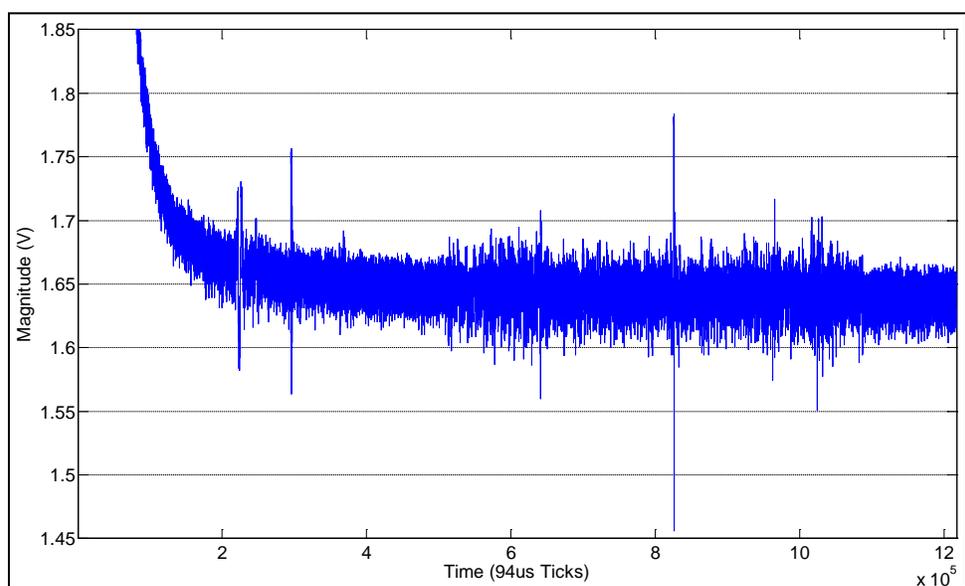
(a) Small Foam Ball (option a)



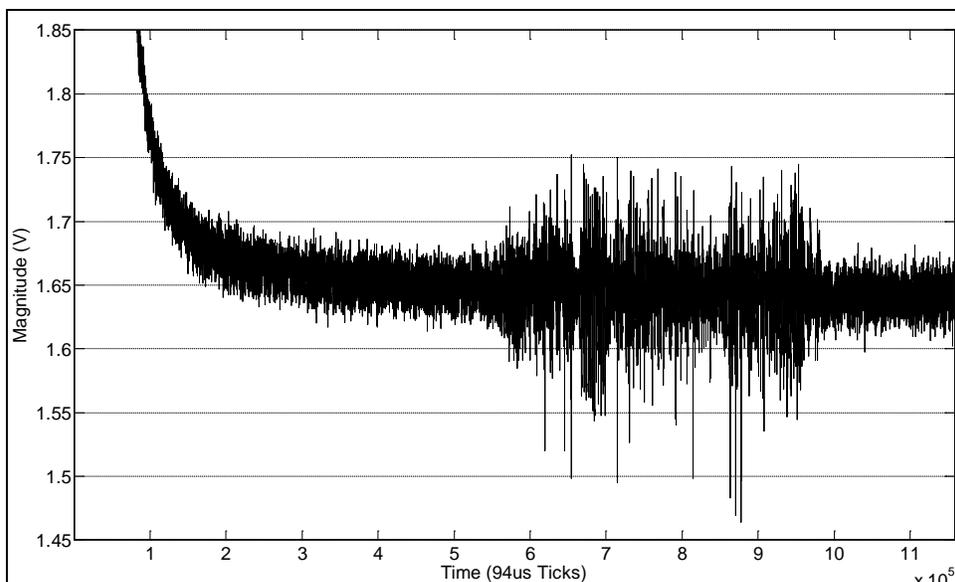
(b) Large Foam Ball (option b)



(c) Custom with Eyes (option c)



(d) Custom with Ridges Facing In (option d)

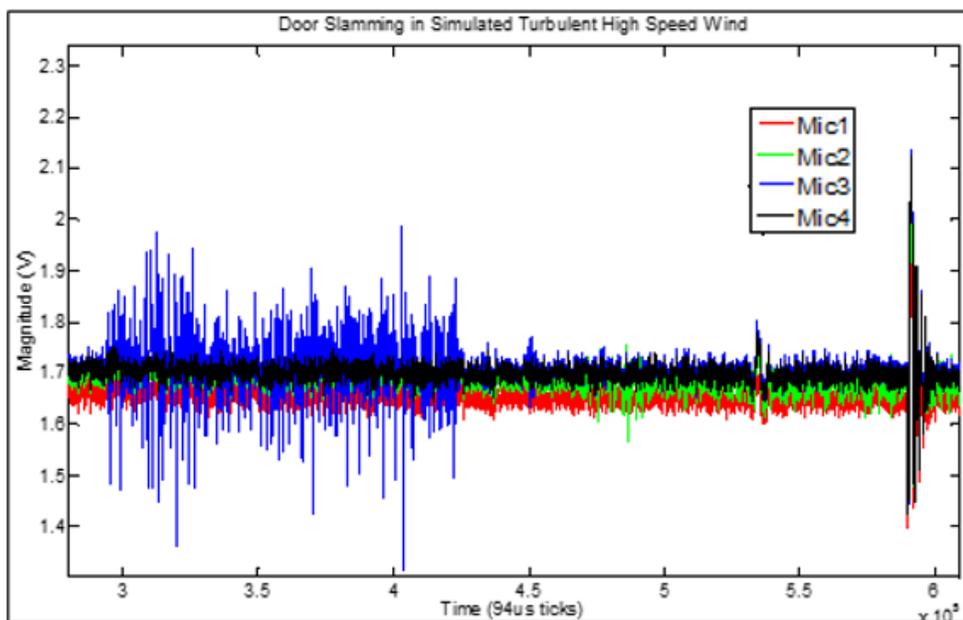


(e) Custom with Ridges Facing Out (option e)

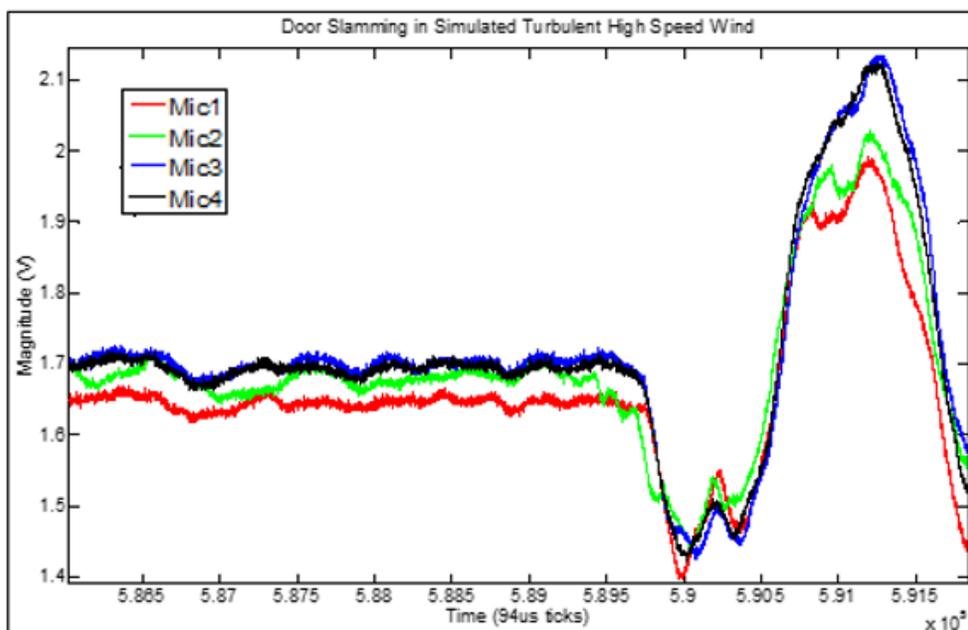
Figure 5.14 Wind Screens Tested in 20mph Gusts

Due to the promise shown by the custom windscreen option with the ridges facing in (Figure 5.13: option d), further research was conducted into defeating high speed winds as part of the mechanical design. The theory was developed that a multi-stage windscreen approach could potentially improve wind noise reduction. So, a test was performed where simulated wind was applied to one of the acoustic goniometer sensors equipped with not only the custom windscreen (ridges facing in: option d) but also a large foam ball. The large foam ball was placed on the microphone tube (see Figure 5.11). Then, the custom wind screen was used to enclose both the sensor case and the large foam ball. Once turbulent, high speed wind was simulated on this microphone, a door slamming test was conducted in order to determine both the ability of this solution to combat wind and to verify that any attenuation of the desired signal would be acceptable. The results from one of the door slams of this test can be seen in Figure 5.15 (other tests produced similar results). The sensor with the windscreen option under test is represented

by the data set with the green line. A brief glance at Figure 5.15 (a) shows a rather noisy signal for the data set with the blue line. This was included to provide an idea as to the effects of the wind on a sensor protected only by the small foam ball. The same method was used to simulate wind for a brief period on the sensor represented by the blue line as was employed to test the windscreen setup on the sensor represented by the green line. The figure clearly shows that a significant amount of wind noise was generated for this test. However, even with the simulated wind, the green line shows little response above the other sensors, which were not subjected to any simulated wind. Slightly prior to the 5×10^5 tick mark (see the Time axis of Figure 5.15 a), a small response causing the green line to dip below 1.6V can be seen. The magnitude of this response, however, is not nearly large enough to trigger an erroneous event. Furthermore, as evidenced by Figure 5.15 (b), the data attenuation is minimal enough to be of little concern and should be considered more than acceptable in light of the solution's ability to combat high speed wind noise. Careful inspection of the green line in Figure 5.15 (b) does show a difference as compared to the other sensor data. Around the 5.895×10^5 tick mark, an oscillation in the green signal can be seen, which is not present in the other three signals. This oscillation is the effect of the turbulent wind not prevented by the windscreen. While this might cause a problem for accurate correlation for the test shown, the fact that all four sensors would be affected similarly by the wind in an actual field test should be noted. Thus, the data clearly shows that the multi-stage windscreen option shows definite promise for combatting turbulent winds up to 20 mph.



(a) Single Door Slam Test with Wind Noise on Green Data



(b) Door Slam Event with Wind Noise on Green Data

Figure 5.15 Best Windscreen Solution Test Results

Based on the results from testing the different wind screen configurations, the success of the windscreen option proves that mechanical solutions to the high speed wind

problem are possible. However, the limitations of the tested options should be considered. Unless the windscreens shown above are properly (possibly permanently) secured, the potential exists for higher speed winds to remove the windscreens entirely, thus negating their ability to combat wind noise. Furthermore, while tests performed on wet windscreens as compared to dry windscreens showed no difference in performance, more testing would be required to determine if a windscreen is a weatherproof solution. Some of the authors of previous research discussed in Chapter 2 experimented with mechanical methods for combatting wind noise, which showed promise and would be worth exploring. While the subject was not always discussed in great detail by some (the solution only being visible from images of the sensor array), many relied on large acoustic foam windscreens or custom weatherproof enclosures that incorporated some form of acoustic foam. Others experimented with sampling tubes extending beyond a weatherproof enclosure, which worked to isolate the system from turbulent winds. While testing some of these approaches were beyond the constraints of this research, they show that the problem of wind noise is approachable, solvable, and worthy of further research with the current acoustic goniometer design.

CHAPTER SIX: RESULTS

The acoustic goniometer as described in Chapter 5 has been built and extensively tested. The results detailed and analyzed in this chapter prove the capabilities of the system and indicate good potential for further research in the field of acoustic goniometry using the current antenna design. Data from both the acoustic goniometer and ADC reader are presented in this chapter. While the acoustic goniometer data is of most interest to the work presented in this dissertation, the ADC reader's raw data provides a useful tool for graphical representation and analysis of the data.

In order to prove feasibility of the acoustic goniometer, several tests of the full system in real-world environments were conducted. Laboratory tests were conducted first to validate the system before field testing was performed. As with the simulations discussed in Chapter 4, door closing events were used to create an infrasound source for the acoustic goniometer to track. Once this verification of the system was completed, similarly to the experiments described in Chapter 4.3, several shooting tests were designed and completed to prove the real-time data processing capabilities of the acoustic goniometer and to determine the accuracy of the system under real-world conditions. This chapter details the setup of the tests and reports the results.

6.1 Data Collection

All data presented in this chapter was collected using the acoustic goniometer without the aid of any computer system for any part of the data processing. The ADC reader (Chapter 5.1.2) was used to collect raw data in order to check the performance of

the acoustic goniometer and provide further insight during analysis. All graphs showing raw data were generated using Matlab, and all raw data analysis was performed using the simulation algorithms described in Chapter 4.

As mentioned in Chapter 5.1.2, the ADC Reader has to store data as quickly as possible due to the large quantity of data collected at a significant sampling rate. One unfortunate consequence of this form of storage is a completely human unreadable data file (see Figure 6.1). The figure shows the stored data in the columns labeled 0-b. The “Address” and “Dump” columns show the address of the data within the file on the card and the computer’s attempt to display the data in ASCII characters, respectively. Figure 6.1 makes clear the need for the Hex File Processor program discussed in Chapter 5.1.2 for converting the file into a known/readable format for further processing in an analysis program. Another drawback to storing such a large amount of data so quickly is the lack of time stamps for correlating data in the raw file to detected events in the acoustic goniometer data file. Recording time stamps for each piece of raw data would be impossible for the embedded platform while maintaining the necessary sample rate. Thus, all graphs of the raw data from the ADC Reader have an x-axis whose ticks measure 94 us (sampling period) starting from time zero rather than a specific date and time. While this is not ideal, the results are still accurately recorded and the data can still be used to serve its intended purpose.

Address	0	1	2	3	4	5	6	7	8	9	a	b	Dump
00000000	ff	0f	fb	0f	fb	0f	fe	0f	ff	0f	f9	0f	ÿ.û.û.þ.ÿ.û.
0000000c	fd	0f	fd	0f	fc	0f	fb	0f	ff	0f	fe	0f	ÿ.ÿ.û.û.ÿ.þ.
00000018	fc	0f	fb	0f	fe	0f	fc	0f	fd	0f	fe	0f	û.û.þ.û.ÿ.þ.
00000024	fc	0f	fc	0f	ff	0f	f7	0f	fc	0f	fc	0f	û.û.ÿ.÷.û.û.
00000030	fe	0f	fb	0f	ff	0f	fb	0f	fe	0f	f7	0f	þ.û.ÿ.û.þ.÷.
0000003c	fe	0f	fc	0f	fc	0f	fd	0f	fd	0f	fd	0f	þ.û.û.ÿ.ÿ.ÿ.
00000048	fc	0f	fb	0f	fe	0f	fa	0f	fd	0f	fb	0f	û.û.þ.ú.ÿ.û.
00000054	fd	0f	fb	0f	ff	0f	fa	0f	fd	0f	fc	0f	ÿ.û.ÿ.ú.ÿ.û.
00000060	ff	0f	fe	0f	fd	0f	fb	0f	fe	0f	fb	0f	ÿ.þ.ÿ.û.þ.û.
0000006c	fb	0f	fb	0f	ff	0f	f7	0f	ff	0f	fe	0f	û.û.ÿ.÷.ÿ.þ.

Figure 6.1 ADC Reader Raw Data File

Previously, all data presented in this dissertation has been in the form of raw data collected from the acoustic goniometer's ADC reader accompanied by post processing calculations and analysis. This chapter presents data from the actual acoustic goniometer calculated in real-time on the embedded hardware. Figure 6.2 shows an example of data recorded by the acoustic goniometer along with keys for interpreting the data. Since showing the recorded text files would do little other than clutter this document and poorly convey the performance of the goniometer, another method for presenting the data was selected. For the analysis of system performance presented in this chapter, the DOA calculations recorded by the acoustic goniometer are compared to the expected DOA in each test determined based off the known location of each event source. These comparisons provide a metric for determining the percent error for each test, which is then averaged with similar (repeated) tests and reported. The data file example shown in the figure is presented merely to illustrate how the acoustic goniometer provides a DOA measurement without the need for any further processing. Two types of information packets are recorded in the data file shown in Figure 6.2: identifiers and measurements.

and elevation (G= Goniometer_EL). For every detected event, each of these sensors reports a value to provide a complete DOA measurement.

Similarly to the identifier packets, the measurement packets can be broken down into several distinct components as shown in Figure 6.2 (c). The “M” at the front of the packet identifies it as a measurement. This is followed by the unit and sensor IDs and a time stamp indicating when the measurement was taken. The “Raw Value” referred to in the key usually represents raw/unprocessed data for most other sensor types. In the case of the acoustic goniometer sensors, this value is not merely unprocessed data but a debug value used to analyze other aspects of the system’s performance depending on the test being run. For example, a raw value of 0 for both azimuth and elevation is indicative of a calculation error (thus providing a means of checking the validity of the data for a given DOA measurement). Finally, the last field gives the final calculated value for the sensor. In the case of the acoustic goniometer, this is an angle measurement in degrees with respect to the positive x-axis and positive y-axis for the azimuth and elevation, respectively.

6.2 Laboratory Tests

6.2.1 Test Setup

While the laboratory was not an ideal environment due to the numerous sources for multipath, it provided a convenient location to verify functionality and test the acoustic goniometer under near-worst case conditions. For these tests, the system was not expected to accurately calculate the source of the infrasound event. The primary goal of these tests was to gauge the capabilities of the system in the presence of extreme multipath and to verify the functionality of the system prior to field testing.

The laboratory tests of the acoustic goniometer used a setup similar to the one found in Chapter 4.2. A slamming door in the laboratory was used to generate infrasound events for the goniometer to detect and track. Figure 6.3 shows a top-down view of the laboratory layout for the tests. The acoustic goniometer was deployed in the regular tetrahedral geometry used for other tests and is represented by the four red dots in each of the figures (labeled S1-S4 for sensor1-sensor-4 or mic1-mic4). The blue dot marks the location of the source (door). The most important feature of the setup that should be noted is the proximity of the sensors to large objects (walls, desks, etc.). Although not shown in the figure, S4 was located just inches in front of a light fixture on one side and had ceiling mounted electrical/networking racks within 0.6 m on two other sides. Given the number of sources for multipath, the acoustic goniometer could not be expected to perform well in this environment, but important features of the system could still be tested. The goniometer was placed approximately 8.43 meters from the door and was rotated in order to change the azimuth for the source's DOA in each test. Figure 6.3 (a) shows the configuration of the goniometer antenna for the tests performed at 90° . The orientation of the acoustic goniometer for the tests performed at 45° and 135° are shown in Figure 6.3 (b) and Figure 6.3 (c), respectively.

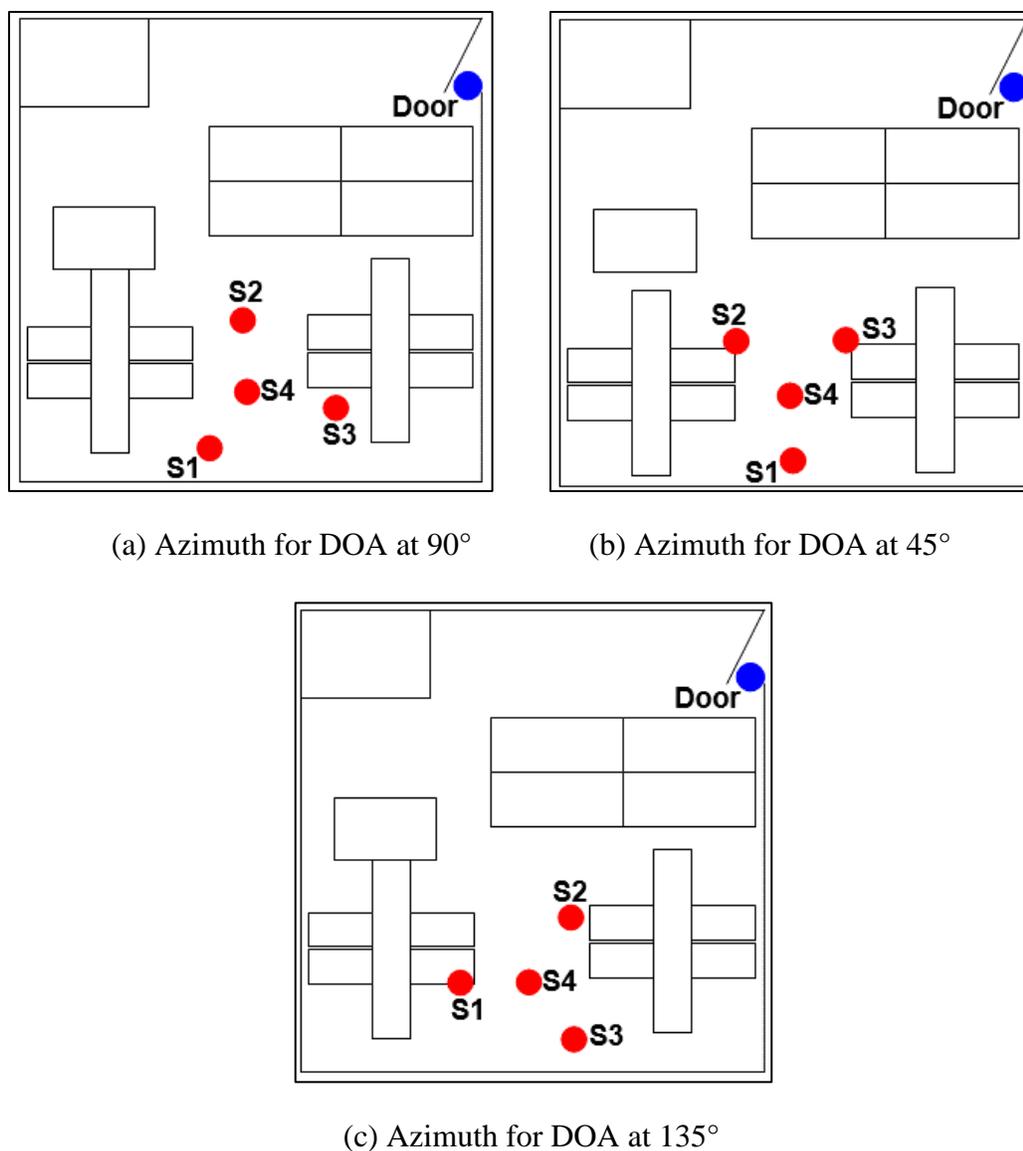
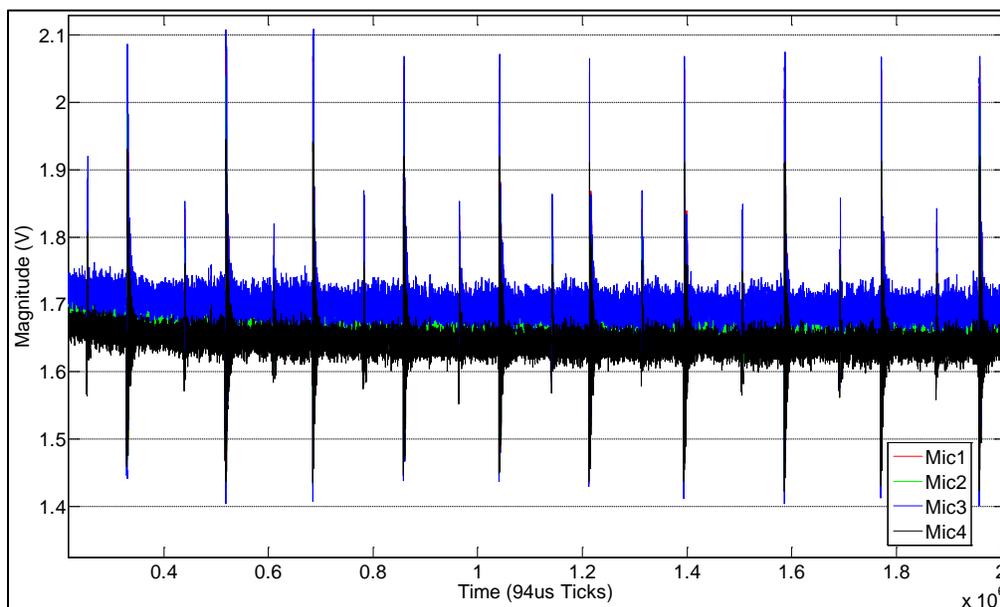


Figure 6.3 Laboratory (top-down view) Setup with Indicated Source (Blue) and Sensor Placement (Red)

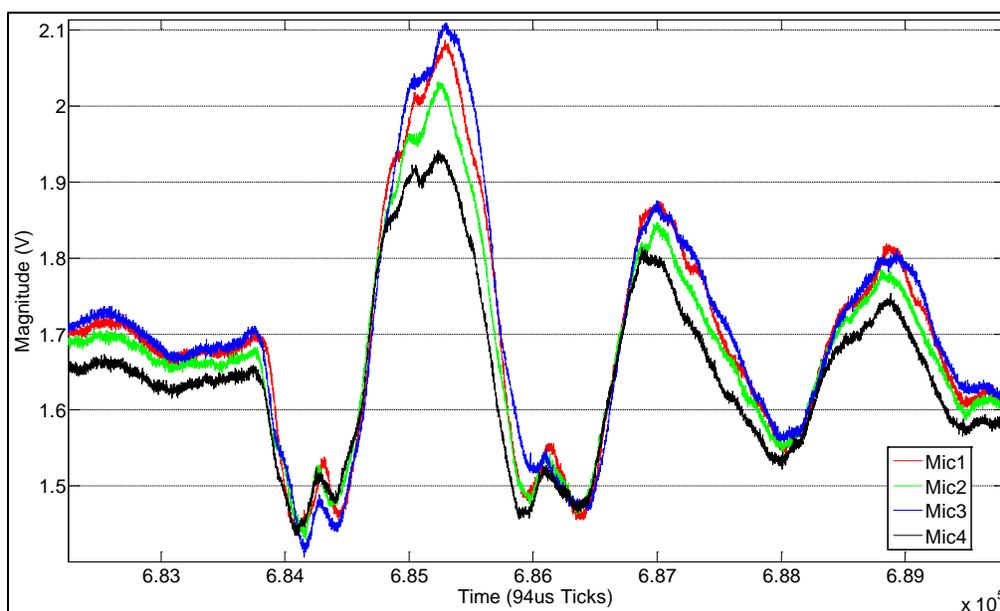
6.2.2 Data Analysis

The raw data collected by the ADC Reader firmware of the acoustic goniometer for the laboratory tests shows that the data collection system is working as expected. Figure 6.4 shows the raw data from some of the tests performed with the source (door) at 90° . The amplitudes of the test are all reasonably uniform (see Figure 6.4 a). The smaller spikes in amplitude, which can be seen between the events, were caused by the door

opening prior to the closing event. The events appear to have arrived at the sensors in the expected order for a source at 90° (see Figure 6.4 b: arrival order started with Mic2 and Mic3 followed by Mic4 and then Mic1).



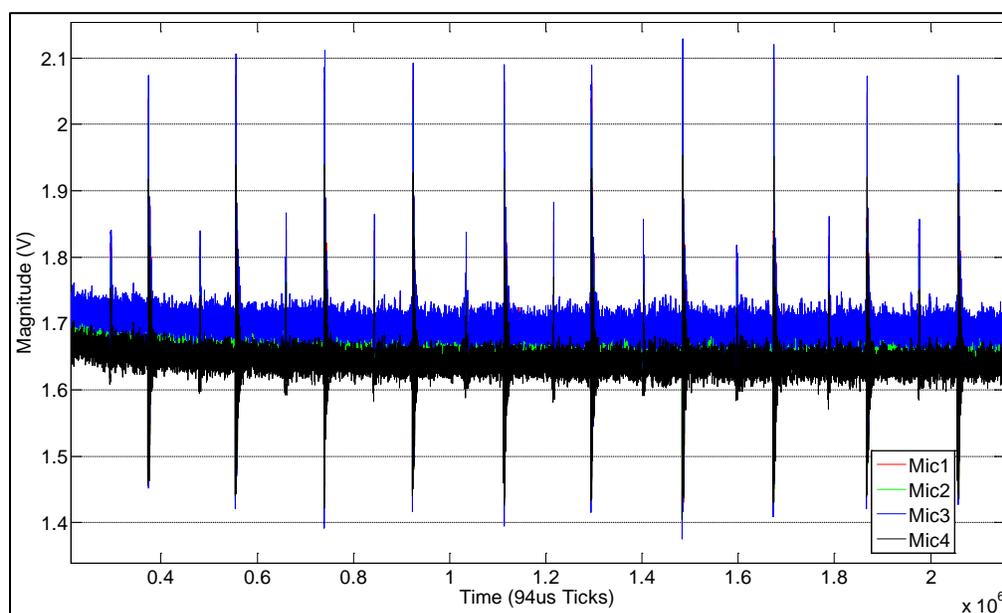
(a) Raw Data from 10 Events at 90° Azimuth



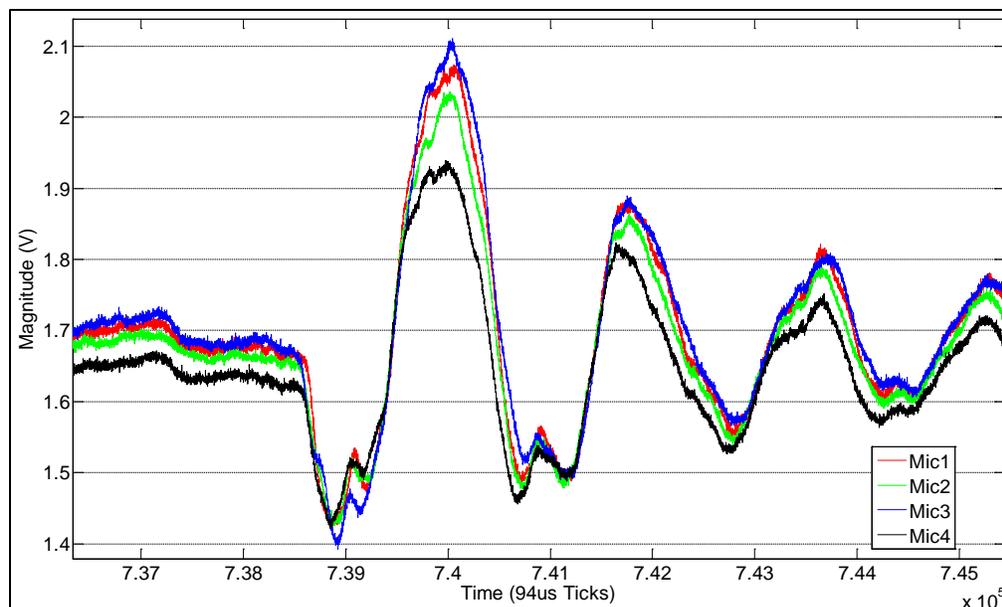
(b) Magnified View of Second Event

Figure 6.4 Raw Data from Laboratory Test with 90° Azimuth

The data graphed in Figure 6.5 depicts the raw results from some of the tests performed at 45° . As with the results at 90° , these results show uniform amplitudes. A comparison with the results in Figure 6.4 shows that the door slamming event produces a predictable event fingerprint, and the acoustic goniometer's sampling algorithms produce repeatable results. Careful examination of Figure 6.5 (b) shows the order of arrival at the sensors changed to match the expected order for a source at 45° (Mic3 followed by Mic2 and Mic4 simultaneously and finally Mic1).



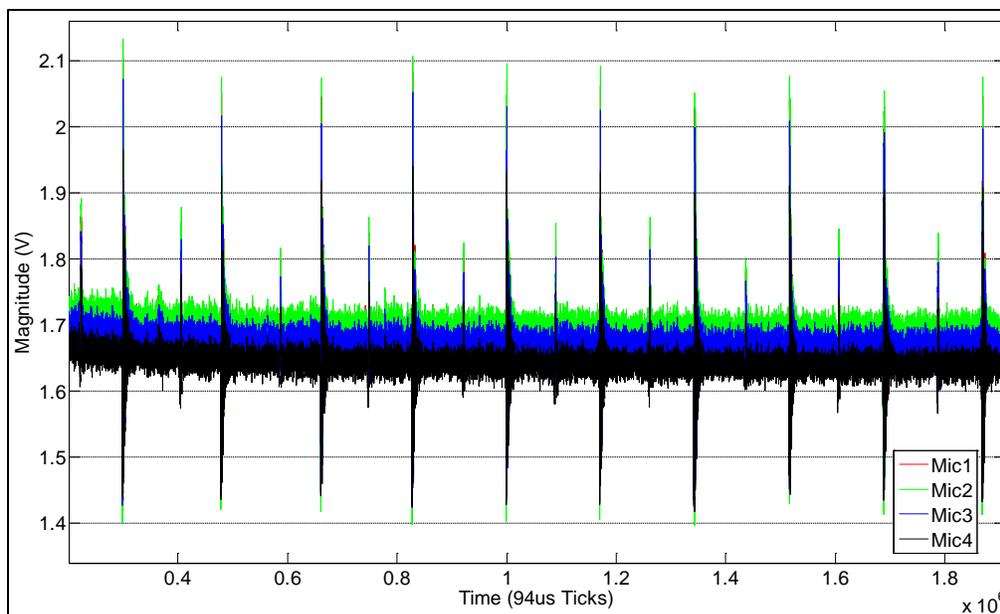
(a) Raw Data from 10 Events at 45° Azimuth



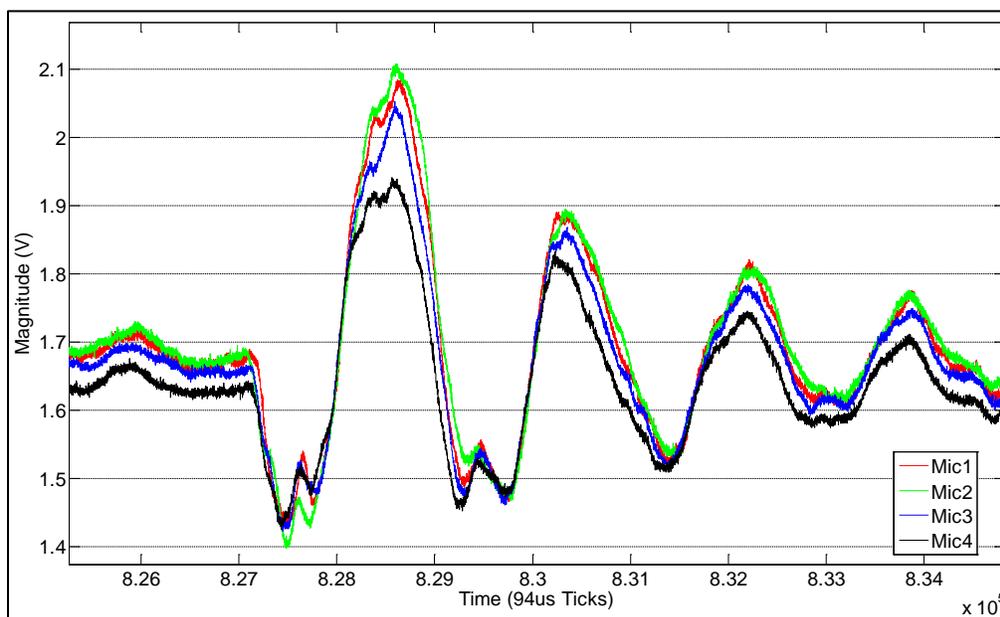
(b) Magnified View of Third Event

Figure 6.5 Raw Data from Laboratory Test with 45° Azimuth

A sample from the final series of laboratory tests performed with the source at 135° can be seen in Figure 6.6. Once again, the amplitudes of the events are reasonably uniform and closely match the events recorded with the DOA at 90° and 45°. This further proves the repeatability of the system's data collection capabilities, which is a crucial component of producing reliable DOA results. Finally, the order of arrival for the events occurred as expected for a source at 135° (see Figure 6.6 b: arrival order started with Mic2 followed by Mic3 and Mic4 simultaneously and finally Mic1).



(a) Raw Data from 10 Events at 135° Azimuth



(b) Magnified View of 1 Event

Figure 6.6 Raw Data from Laboratory Test with 135° Azimuth

One feature worthy of note in each of the figures in this section (see Figure 6.4, Figure 6.5, and Figure 6.6) is the obvious presence of multipath reflections in the recorded signal. Compare the raw data of these figures to the raw data from a shooting

event in Figure 6.7. Each of the figures above show an initial event arrival with a downward spike in the data. Due to the close proximity of the surrounding objects (sources for reflections), this initial spike was most likely affected by reflections, but such error is not readily apparent in the figure. However, the rest of the signal shows multiple obvious reflections as indicated by the repeated large peaks and valleys. As a point of comparison, the gun shot data (see Figure 6.7) collected in an outdoor environment without objects in close proximity to the goniometer shows a single major event and only a few small reflections (peaks and valleys).

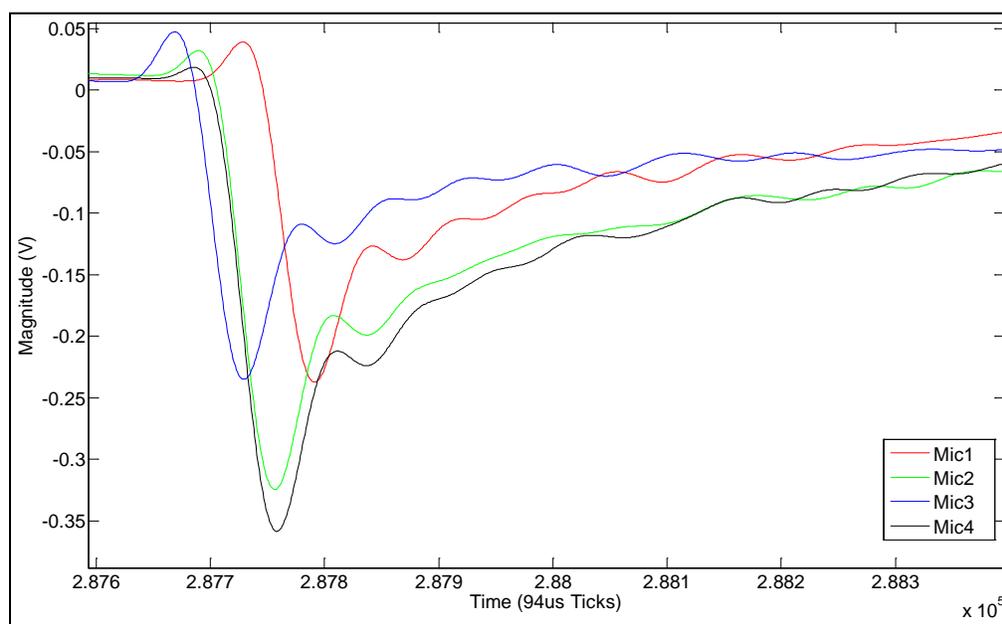


Figure 6.7 Shooting Test Data with Minimal Evidence of Multipath Reflections

The differences between the data collected in environments with and without multipath error provides a good example of the extent to which multipath can affect the raw data of the acoustic goniometer. As mentioned previously, aside from the obvious reflections shown in the signals, even the data indicating the original event must be questioned in such an environment. Since each sensor had a different number of objects

around them at varying distances, each microphone would have experienced a unique effect from multipath reflections. As an example, some of the sensors may not have seen a reflected signal until well after the initial event (e.g., S2 in Figure 6.3 a). Others, due to the proximity of objects around them (e.g., S4 as a result of the light fixture), could have seen a reflection close enough to the start of event to make the initial spike appear to arrive later. Thus, the calculation for the DOA could be greatly changed. However, verifying such a problem requires an analysis of the acoustic goniometer's performance under these conditions.

6.2.3 Acoustic Goniometer Performance Analysis

As expected for an environment with severe multipath issues, the accuracy of the DOA calculations was terrible. However, the acoustic goniometer performed reasonably well considering the limitations imposed by the environment. For all tests, not a single event was missed nor a false event recorded. The simple event detection algorithm worked perfectly in the controlled laboratory environment. The azimuth calculated for the source at 90° had an average error of 33.22%, and all of the azimuth calculations for this angle were within 2.1° of one another. For the source with the azimuth at 45° , the average error was 8.33%, and the azimuth calculations were all in perfect agreement. With the source azimuth at 135° , the acoustic goniometer was only able to determine the DOA for 40% of the events. When the goniometer was able to calculate the DOA for events with the azimuth at 135° , the average error was 8.33%, and the calculations were all in perfect agreement.

Determining the accuracy of the elevation results for the laboratory tests was not feasible given the proximity of the door to the goniometer. The door is approximately as

tall as the goniometer antenna and was located within 9 m of the sensors. Thus, selecting a single elevation would be an inaccurate representation. Calculations for the elevation ranged from 19.5° to 56.8° . Given that the center of the goniometer was located at approximately half the height of the door, most of the values in this range would be believable. However, considering the results of the azimuth calculations and the multipath problems introduced by the environment, poor accuracy can be safely assumed for these calculations as well.

While the laboratory tests were not able to confirm the accuracy of the system, useful information was gleaned from the results. The uniformity of the event fingerprints coupled with the good repeatability of the DOA calculations proved the system to be reliable enough for a field test. Furthermore, the analysis of the raw data in Matlab agreed with the DOA angles recorded by the goniometer to within a margin of 2% for all calculations. Thus, the algorithms in the embedded hardware were proven to be functioning properly (as compared to their intended design in Matlab). The laboratory tests verified the functionality of the acoustic goniometer and proved that the system was capable of producing repeatable results and recording raw data reliably.

6.3 Lecture Hall Tests

6.3.1 Test Setup

Another test of the system in an indoor environment was conducted to further assess the response of the system in poor environmental conditions. This second indoor test was conducted inside a large, empty lecture hall. The goniometer was placed on top of the desks in an area with plenty of clearance between the top of the antenna and the ceiling (see Figure 6.8). As with the laboratory tests, the red dots in the figure indicate the

placement of the sensors, and the blue dots indicate the location of the sources (doors at approximately 90° and 210° azimuth). The source at 90° azimuth was approximately 4 m from the goniometer, and the source at 210° azimuth was around 7.6 m away. The placement of the sensors and the size of the room were selected such that the multipath effects would be less severe than those experienced during the laboratory tests.

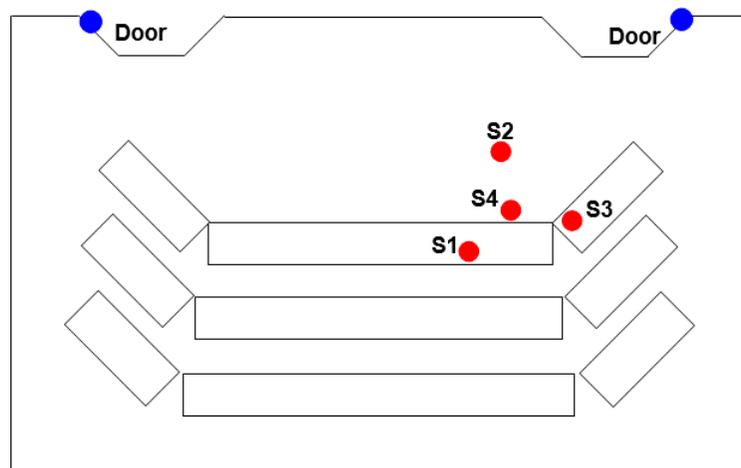
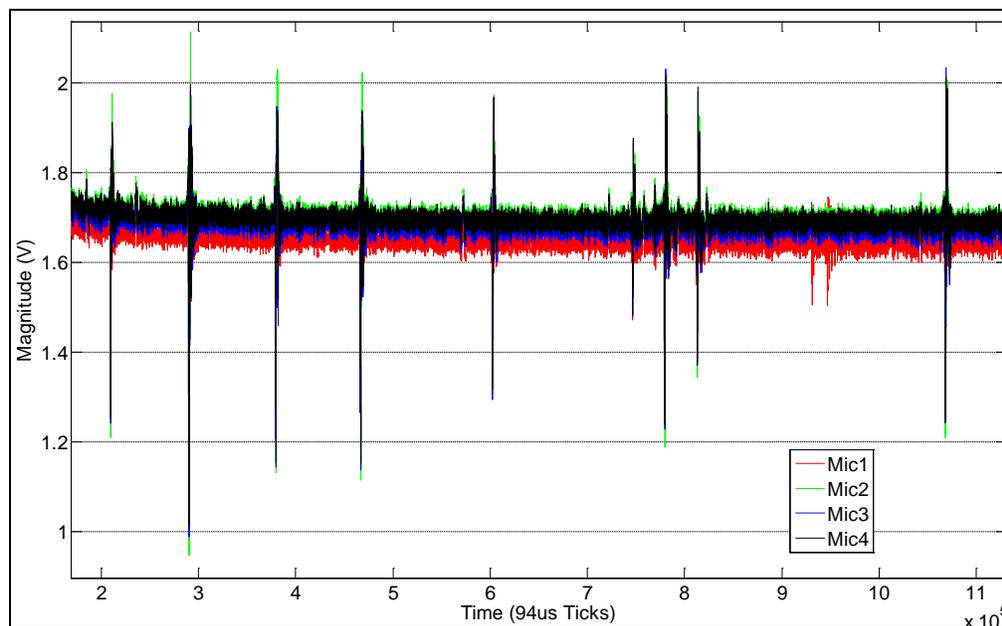


Figure 6.8 Lecture Hall (top-down view) Setup with Indicated Sources (Blue) and Sensor Placement (Red)

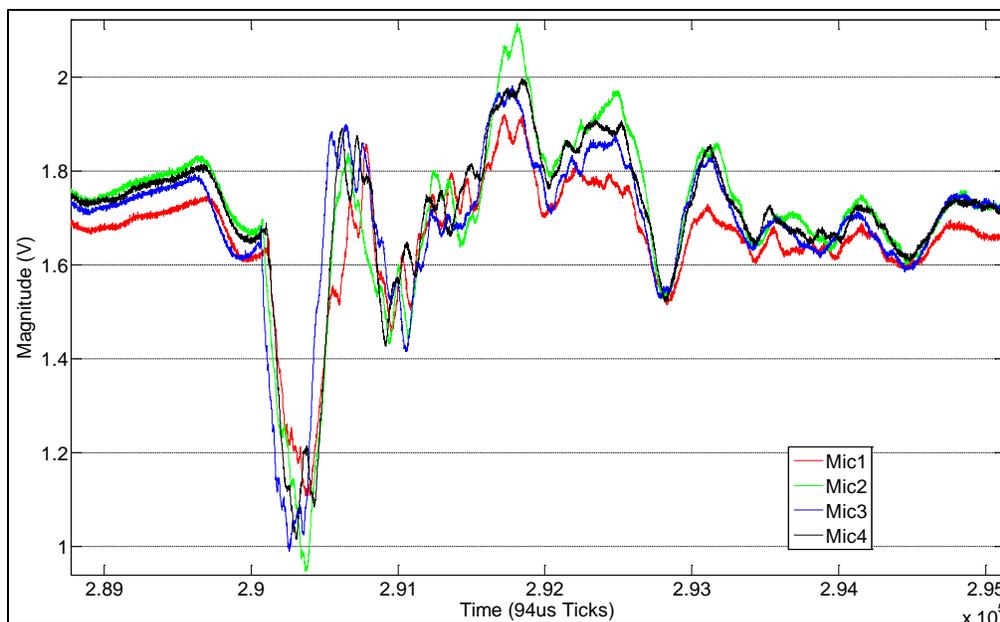
6.3.2 Data Analysis

Figure 6.9 shows the raw data collected during the lecture hall test. The first interesting feature of the recorded signals is the variability of the amplitudes (compare the first event with the ninth). The cause for the difference in signal shape and amplitude can be attributed to the repeatability of the noise source. While the laboratory had a door that could be allowed to close by itself in order to create repeatable fingerprints, the lecture hall doors were configured to close softly (minimizing their noise production). Consequently, doors had to be forcibly slammed in order to create signals that would trigger the goniometer state machine. As such, the force used to close the door varied and produced varying sound waves. Another interesting aspect of the data can be seen in

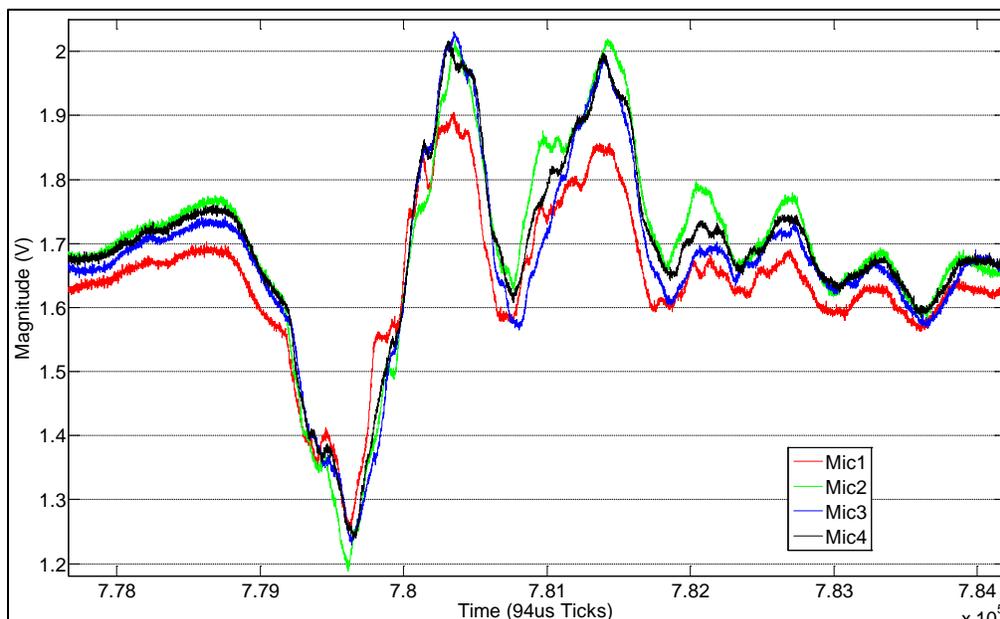
Figure 6.9 (b) and (c). Compared to door closing events seen in the laboratory (see Section 6.2.2), the events in Figure 6.9 appear to have very different fingerprints. The change in environmental conditions (e.g., difference in door shutting mechanism, change in multipath producing elements, etc.) caused the fingerprints to appear drastically different from those seen in the laboratory.



(a) Lecture Hall Door Events at 90° azimuth (first five events) and 210° azimuth (last six events)



(b) Second Event from Figure a (event at 90° azimuth)



(c) Seventh Event from Figure a (event at 210° azimuth)

Figure 6.9 Raw Data from Lecture Hall Test

The raw data shows that signals appeared to arrive mostly as expected (see Figure 6.9 b and c). For the source at 90° azimuth, the sound wave arrived at Mic2 and Mic3

first. However, a significant amount of time appears to have elapsed with the sound arriving at Mic3 before Mic2. Since a sound traveling at 90° azimuth to the acoustic goniometer should be received by both Mic2 and Mic3 simultaneously, this shows that the environmental conditions in the lecture hall were still not free enough from multipath to allow for proper goniometer function. Additionally, the sound traveling at 210° azimuth to the goniometer should have arrived at Mic2 before being seen at Mic1 (arriving later at Mic4 and Mic3). While this appears to occur correctly at the first major minima, the initial negative slope of the data shows the sound arriving first at Mic1 followed by the other three microphones nearly simultaneously. The events recorded in the lecture hall tests further demonstrate the effects of multipath and the inability of the acoustic goniometer to operate in confined spaces.

6.3.3 Acoustic Goniometer Performance

The acoustic goniometer performed well for being operated in an indoor environment. The system identified and recorded all 11 events. However, due to the effects of multipath reflections, the DOA calculations were not accurate. The system was only able to determine a DOA for 54.5% of the events recorded. In the cases where a DOA was not calculated, the presence of reflections caused the effective time of arrival to be the same for more than two sensors. Since this occurrence should be impossible for the assumptions used to model the way in which sound travels, this resulted in a division by zero preventing the system from completing the DOA calculation. When the system was able to perform a calculation for the 90° azimuth source, the error was approximately 33.3%. The average error for the successful 210° azimuth calculations was around 68%. While the DOA results still showed the system to be incapable of operating in enclosed

spaces, the results still prove that the acoustic goniometer works as expected (compared to the analysis of the raw data and simulations).

6.4 Field Tests

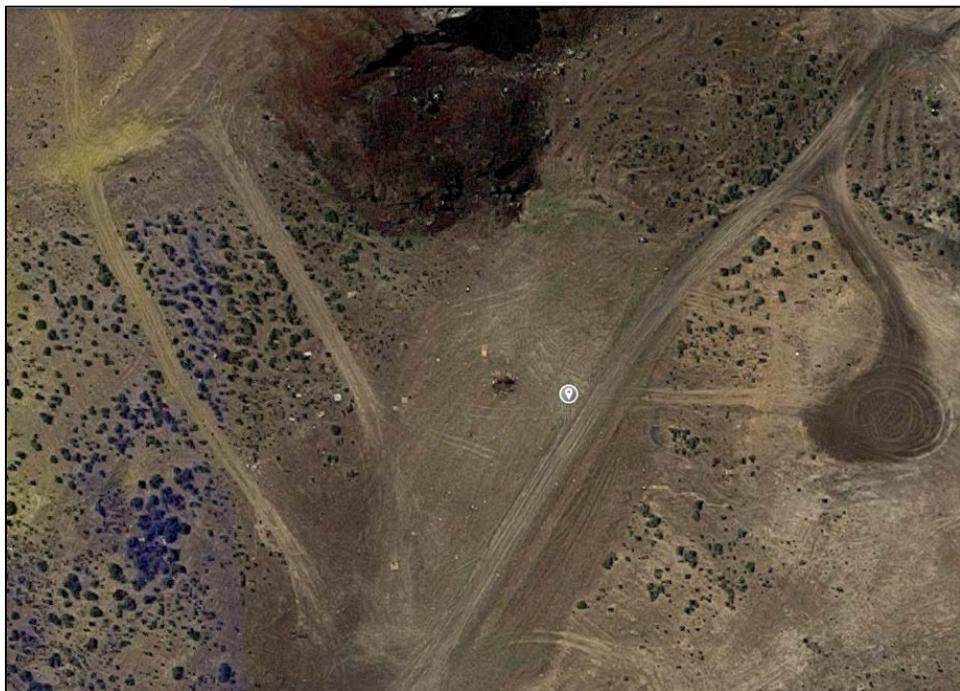
6.4.1 Test Setup

The field tests for the acoustic goniometer used a setup and process nearly identical to the shooting tests described in Chapter 4. The acoustic goniometer antenna was deployed in a desert location surrounded by hilly terrain. Figure 6.10 shows the area surrounding the goniometer. The region is a small, elevated valley surrounded on every side except the left by hills (see Figure 6.10 b). The hills provide a good backdrop for firing the AR-15 (hills supply a stopping point for bullets) and create a mild source of multipath error. While minimizing sources of error during testing is desirable, no natural environment is ideal. Thus, the multipath properties of this region were deemed acceptable for the goniometer tests. In addition to the multipath, another less than ideal feature of the site was the uneven ground upon which the tests took place. The images in Figure 6.10 (b), (c), and (d) make the valley inside the hills look fairly flat. However, close inspection of Figures 6.10 (a) shows that the valley actually not only slopes down (toward the left side of the goniometer), but also that the area where the shots were fired has many small peaks and valleys of its own. While this did not affect the azimuth calculations in any appreciable manner, the uneven ground made determination of the elevation of the sound sources with the available equipment impossible. However, since the equations for azimuth and elevation are so closely related, verified functionality in the azimuth calculation guarantees similar performance in the elevation calculations. Further, based on the small size of the peaks and valleys, an estimation of the elevation at each

location to be within 2° of 0° was reasonable. Thus, this feature of the environment was also deemed acceptable.



(a) Acoustic Goniometer in Field



(b) Top-Down Satellite View of Terrain (gray dot marks approximate goniometer location)



(c) Slightly Angled Top-Down Satellite View of Terrain (gray dot marks approximate goniometer location)



(d) Angled Satellite View of Terrain (gray dot marks approximate goniometer location)

Figure 6.10 Goniometer Field Test Terrain

Once the goniometer was deployed, gun shots were fired by multiple researchers from various locations ranging from approximately 18 m to 26 m away from the center of the goniometer at angles from approximately 45° to 135° in azimuth at approximately 0° in elevation ($\pm 2^\circ$ due to hilly terrain). The AR-15 was fired from a standing position by

each researcher without the aid of a tripod. Figure 6.11 shows an approximation of the layout of the goniometer and event locations in the region shown in Figure 6.10. Since the use of survey equipment was not an option, in order to measure the angles, a simple grid was used to select locations from which to fire the AR-15. A point ~18.3 m from the center of the goniometer along a line running through the center and the leg of the antenna holding sensor 1 (S1: mic1) was selected as being the location for the 90° (azimuth) event source. Then, the remaining locations were selected at 9.14 m intervals along a line parallel to the one which can be drawn between sensor 2 (S2: mic2) and sensor 3 (S3: mic3). This method of measuring angles is not perfect but provided reasonable accuracy and proved adequate for the purpose of the test. However, in order to get more exact error measurements, these tests would have to be conducted with more precise measurements of the angles (both azimuth and elevation).

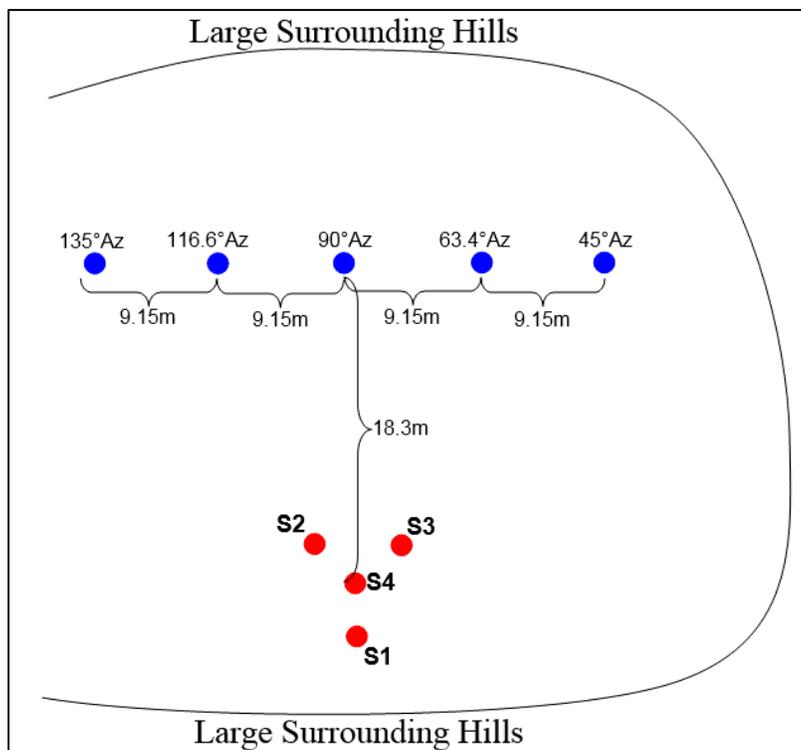
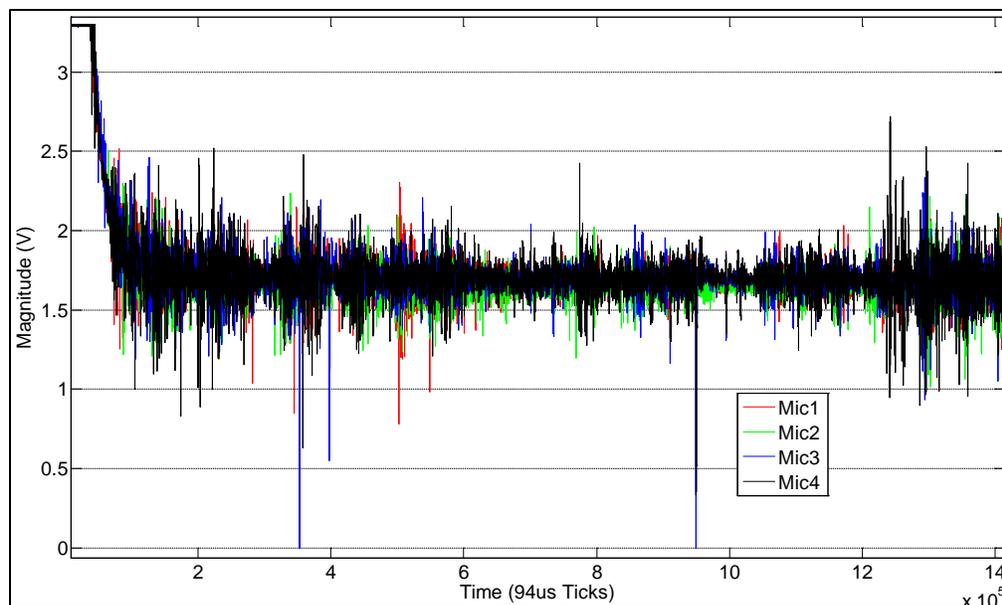


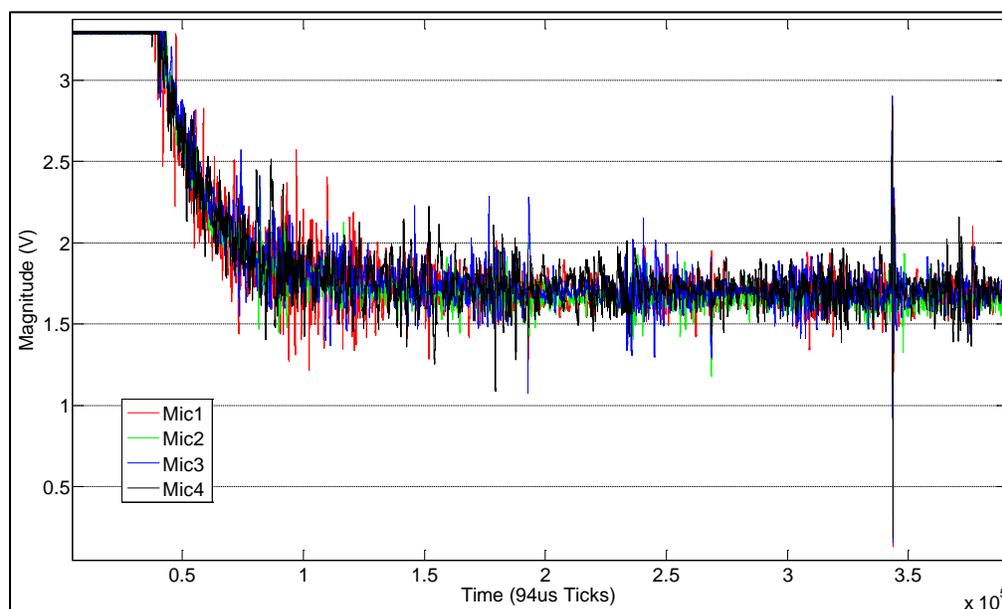
Figure 6.11 Field Test Event Source Layout

6.4.2 Data Analysis

After the shot event sources were marked off, multiple tests were run with several shots being fired from the predetermined locations (see Figure 6.11). Shots were fired from the locations in order from right to left (45° first; 135° last). The goniometer firmware performed all calculations in real-time with timestamps using millisecond resolution. The first attempt at testing the system in the field showed very poor performance on the part of the acoustic goniometer. The goniometer reported more than twice the number of expected events for multiple tests. Upon analysis of the raw data, the reason for this poor performance was determined to be wind noise. Twenty mile per hour (and greater) winds were present during the tests, and the wind screens in use by the system were not sufficient to mitigate the effects. Data from 2 of the tests performed in high speed winds can be seen in Figure 6.12. Figure 6.12 (a) contains shot data from all 5 shooting locations depicted in Figure 6.11. However, picking the shots out of the noise is not possible using the simple thresholding algorithm employed by the acoustic goniometer. The results in Figure 6.12 (b) are similar except only a few shots were fired for this test. Still, the shots are indistinguishable from the spikes produced by the wind. Thus, the cause for the poor performance of the goniometer was obvious, and wind noise had to be factored into the design of the goniometer.



(a) Test 1 Data



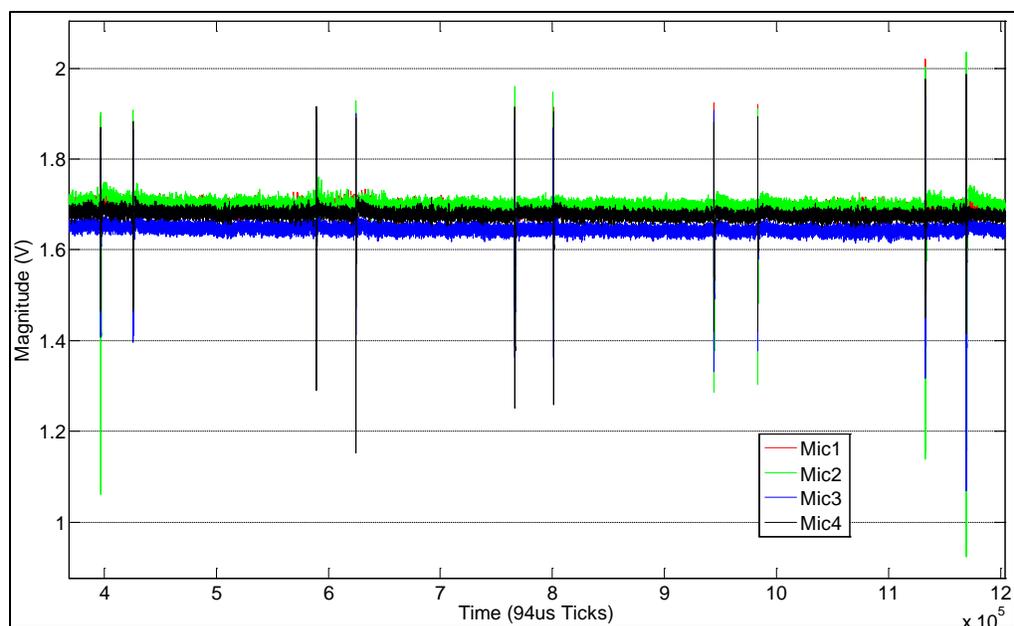
(b) Test 2 Data

Figure 6.12 Data from High Speed Wind Tests without Sufficient Wind Screens

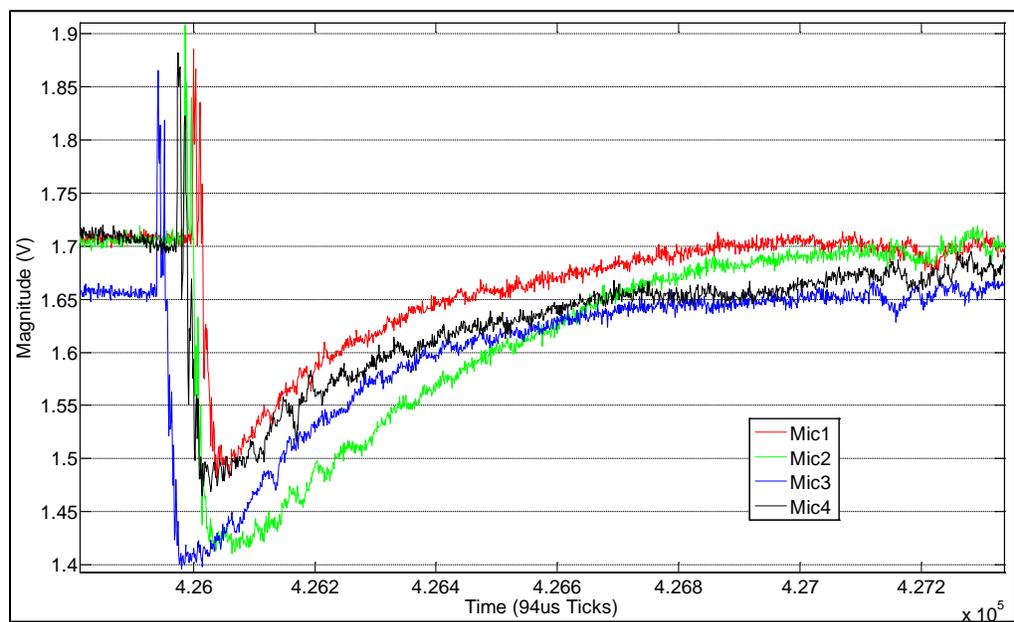
Another interesting feature of the acoustic goniometer data that can be seen in Figure 6.12 is the sensor warm-up period. Most of the other figures presented in this dissertation focused on the analysis of data and were more useful for this purpose when a

magnified view of the data of interest. Since Figure 6.12 was included to aid in the discussion of high magnitude wind noise, all data from the moment the goniometer was turned on to the moment it was turned off can be seen in the figure. As such, the brief period where the sensors are at the upper rail of the system (3.3V) and then decay to their resting voltage ($\sim 1.65\text{V}$) is visible. This period, which can be seen to last until approximately the 10,000 tick mark in Figure 6.12 (b), lasts around 9.4 s and is consistent for every test. The cause for this warm-up period is the charge time for the many capacitors used in the creation of the sensors (mics). During this time, event detection is inadvisable as the system performance would likely be low. Rather, this time must be considered as part of the goniometer's initialization process. This does not affect the system in any appreciable fashion but is worthy of note as a feature of the acoustic goniometer.

Once the wind problem from the first series of tests was addressed (see Chapter 5.2.3.3), several more tests were conducted with far superior results to the first attempt. Figure 6.13 shows the raw data recorded by the ADC Reader firmware (see Section 5.1.2) for a few of these tests. While slightly noisier than laboratory tests, the data shows reasonably clean signals (see Figure 6.13 b) despite 8 mph winds present during these tests. The relative smoothness of the signals (as compared to the data in Figure 6.12) allowed the thresholding algorithm employed by the goniometer to function properly. False positives were entirely eliminated, and not a single event was missed. Thus, the wind screens and filters (both analog and digital) performed well under realistic conditions.



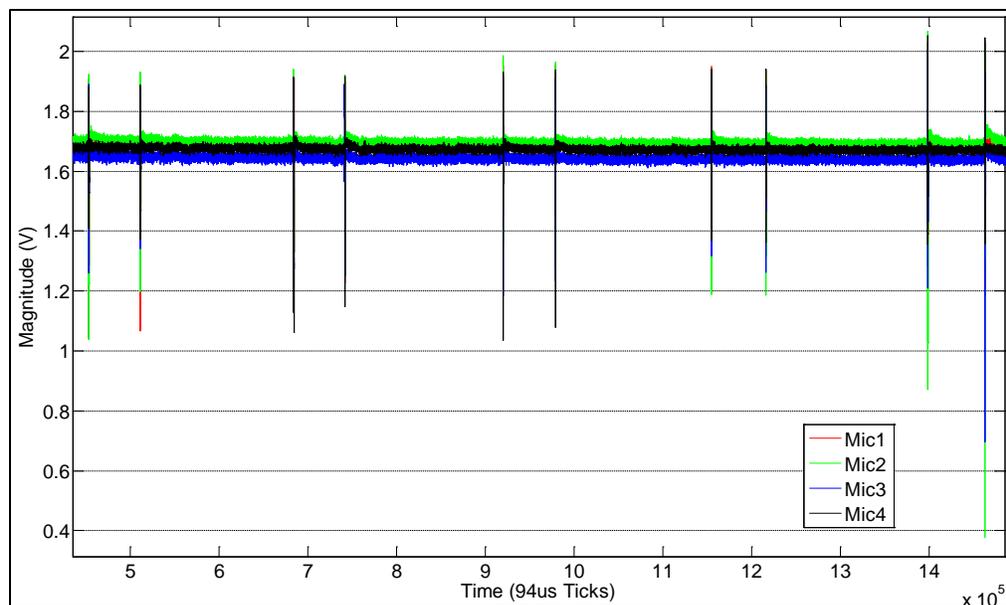
(a) Event Data Collected from all Five DOA Azimuth Angles



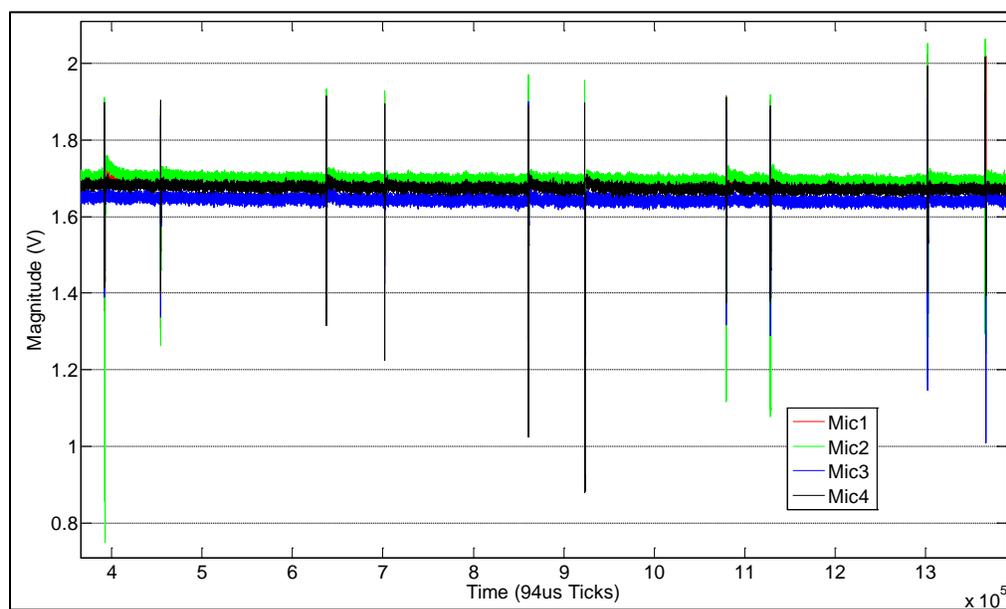
(b) Magnified View of Data Collected from a Shot Fired at 45° Azimuth (second event in figure a)

Figure 6.13 Raw Test Data Collected from 10 Shooting Events

An interesting aspect of the data worthy of notice is the magnitude of some signals. The last event recorded in the test series shown in Figure 6.13 (a) shows a minimum value for one sensor which drops below 1V. This occurs again for the same angle (135° azimuth: 9th and 10th events) in Figure 6.14 (a). Then, in Figure 6.14 (b), the same phenomenon occurs for two other event angles (45° and 90° azimuth: 1st and 6th events, respectively). Since the rails of the system are 0-3.3V and railing any/all of the signals would most likely prevent accurate correlation from taking place, signals which drop so low are good indicators that the gain of the system should be adjusted for this particular phenomenon. The same would be true in the event that data was seen to spike close to 3.3V. The gain needed for the amplifiers on an acoustic goniometer's sensors varies depending on several factors including: sensitivity of the microphones to the frequency of interest, the energy of the sound produced by the given phenomenon, the distance from the source to the antenna, and the configuration of the surrounding environment as it pertains to additive noise caused by reflections of the original sound. One specific gain cannot be selected to fit all possible applications. Thus, characterization of an event via the collection and analysis of raw data as shown in Figure 6.13 is necessary for any project using an acoustic goniometer. This is by no means a failing of the system but rather an example of one of the features that must be tuned to fit a particular application.



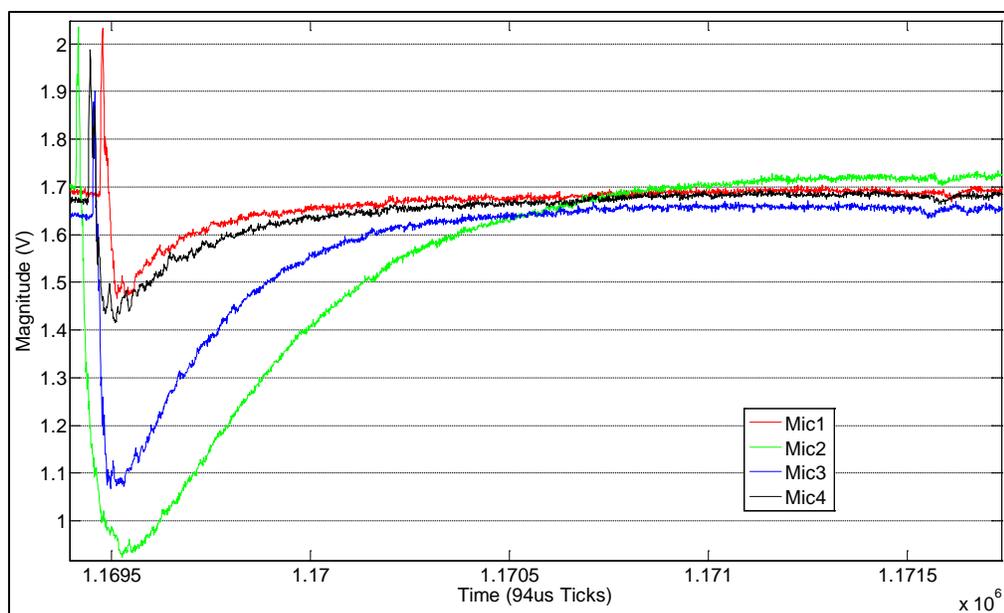
(a) Second Series of Event Data Collected from all Five DOA Azimuth Angles



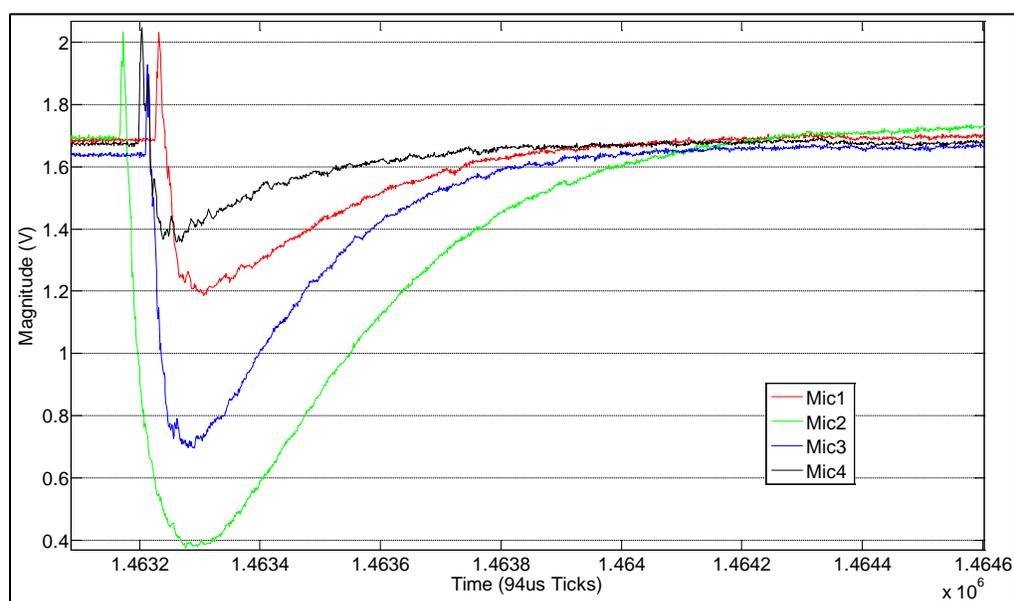
(b) Third Series of Event Data Collected from all Five DOA Azimuth Angles

Figure 6.14 Raw Test Data Showing Variations in Magnitude Irrespective of DOA

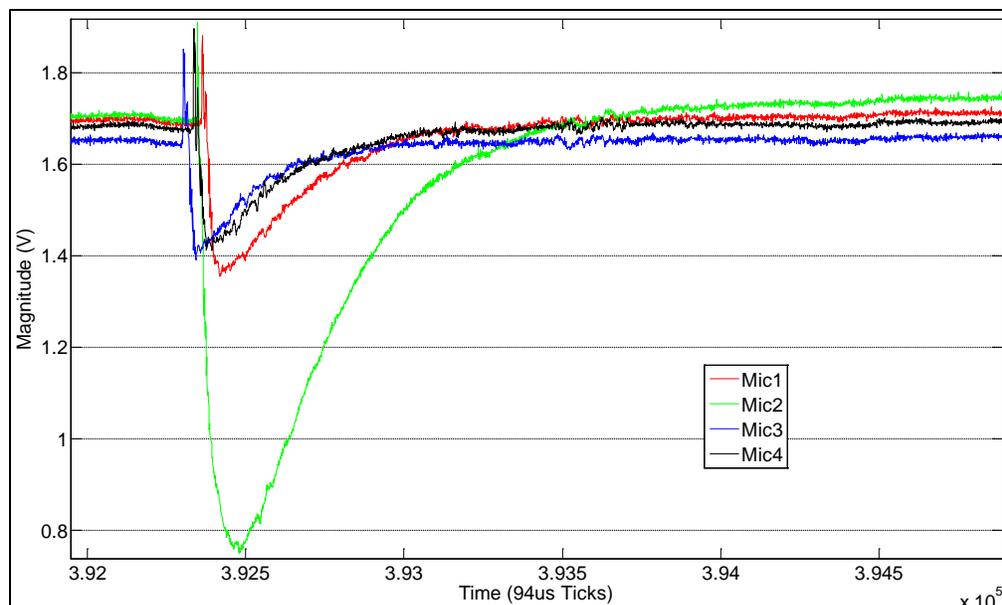
A comparison of the raw data shown from the three test series in Figure 6.13 (a) and Figure 6.14 makes the variability of event amplitudes for the gunshot tests readily apparent. Figure 6.15 shows magnified views of the events previously discussed as dipping below 1V in magnitude. The variability of the amplitudes can be seen for all event angles, including those whose minimum value was still above 1V, but the instances where the magnitude dipped below 1V provide an example sufficient to the analysis in this discussion. Notice in Figure 6.15 (a) and Figure 6.15 (b) that the increased amplitude has not affected the order of arrival for the signals. Thus, since the cross correlation algorithm is not dependent upon the amplitude of the signal, the DOA calculation is not negatively impacted by this phenomenon. Based on the examples presented in these first two figures, a relationship between the 135° azimuth and the occurrence of a drastic drop in the magnitudes of Mic2 and Mic3 could be suggested. This can be immediately dismissed by the examples presented in Figure 6.15 (c) and Figure 6.15 (d) where all of the sensors behave differently than the first two figures and the angles are 45 and 90, respectively. So, the behavior is not specific to any given angle. One might be tempted to postulate that this behavior is sensor dependent based on the similar behavior of Mic2 and Mic3 in Figure 6.15 (a) and Figure 6.15 (b). However, the same behavior can be seen from Mic4 in Figure 6.15 (d). Furthermore, the amplitudes of Mic1 and Mic4 also vary there amplitudes between Figure 6.15 (a) and Figure 6.15 (b). Thus, this type of behavior is most assuredly not a quirk associated with a particular sensor or sensors. Consequently, the hardware must not be the cause for the variability of the amplitudes. Understanding this behavior requires more careful analysis of the event type and environment.



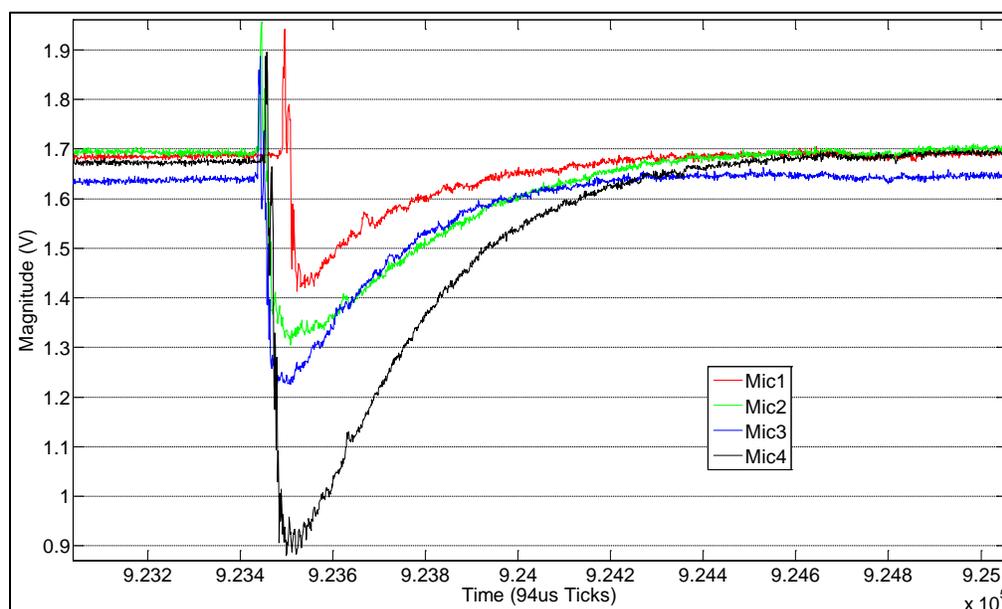
(a) Figure 6.13 (a) Event 10 (135° Azimuth Source)



(b) Figure 6.14 (a) Event 10 (135° Azimuth Source)



(c) Figure 6.14 (b) Event 1 (45° Azimuth Source)



(d) Figure 6.14 (b) Event 5 (90° Azimuth Source)

Figure 6.15 Magnified View of Amplitude Variance Occurring Irrespective of DOA

A previous discussion listed factors that typically affect the amplitude of an event as detected by the goniometer sensors (microphone sensitivity, sound energy, distance

from source to sensors, and environmental factors). The same type of ammunition was used for all shots fired: caliber, powder charge, casing material, bullet material, and manufacturer. High quality ammunition (such as was used for these tests) is very precisely manufactured in order to ensure every bullet performs in the same fashion. While every process has acceptable tolerance levels, this effectively means that each bullet fired should produce a sound event with nearly the same energy level. Since the microphone sensitivity and the distances from the event sources to the sensors did not change, something pertaining to the environment is the most likely cause of this phenomenon. Given that the physical structures surrounding the goniometer did not change (e.g., rocks, hills, etc.), the most plausible explanation is that wind gusts in the valley were responsible for affecting the perceived energy of the sound events. Further testing of the system using the same event source and environment would be required to confirm this theory. Since the variability in amplitude did not negatively impact the acoustic goniometer's performance in any noticeable way, this behavior was deemed acceptable.

6.4.3 Acoustic Goniometer Performance Analysis

Raw data aside, the acoustic goniometer firmware performance (real-time DOA calculations) during the field tests was good for all source angles tested. All of the calculations showed less than 5% error as compared to the expected approximate angles, and every event was properly identified and recorded with no false events. Results from the test data analysis is shown in Table 6.1. Over 100 shots were fired during the course of testing, and the error (as compared to the expected angle) was calculated for each event. Since only azimuth was purposefully varied, elevation error is not shown in the

table. The average error for the elevation of approximately 0° was 2.39%, and the worst error was 4.31%. The standard deviation for the elevation angles was 1.34. The sources of error for this test are the same as those mentioned in Chapter 4.3 (e.g., lack of survey equipment for measuring precise angles, approximated sensor locations on the prototype structure, finite numerical resolution, etc.). Most of these issues are fixable, but require either more skilled mechanical designs than are within the abilities of the author or a familiarity with survey equipment. Given that antenna geometries have been shown to affect goniometer performance [13], the apparent favor shown by the system toward certain DOA angles is most likely due to the characteristics of the gunshot phenomena and the design of the goniometer's antenna geometry. Based on the work done by Van Lanker [13], this explanation seems like the most valid theory, however further exploration into this behavior is worth considering. Overall, the acoustic goniometer performed well. The results show that the system is capable of being used for localization of events to within a reasonable tolerance. Further fine tuning of the algorithm thresholds and windowing parameters could produce more accurate results (see Chapter 3.3.2.2 for a discussion on how windowing affects accuracy). However, reasonable accuracy has been attained with the current values, and the scope of this research was never intended to be limited to a single acoustic goniometer application. Thus, fine tuning and improving the acoustic goniometer's performance for a specific application must be left for future researchers whose goal is to use the current work for more specific research.

Table 6.1 Analysis of acoustic goniometer test results shown in terms of accuracy for various DOA angles

Angle (azimuth)	45°	63.4°	90°	116.6°	135°
Average Error (%)	1.64	1.24	1.48	2.25	1.9
Average Azimuth (°)	42.05	61.17	92.66	112.55	138.42
Worst Error (%)	3.6	1.5	1.78	4.44	2.44
Worst Azimuth (°)	38.52	60.7	93.20	108.61	139.39
Standard Deviation	3.60	1.17	1.33	3.03	2.96

In addition to proving the primary goniometer algorithms, several other algorithm options were explored during the field tests. One feature added to the SD card capabilities of the system was the ability to change portions of the filtering algorithm. As mentioned previously, every goniometer application is unique and has different requirements. In an effort to further explore adjusting the system to adapt to various needs, two changes to the working algorithm were tried: the addition of a moving average filter after the event detection stage and the use of a more complex DC offset adjustment. The moving average filter was meant to smooth the data in an attempt to improve the accuracy of the correlation algorithm and (by extension) the DOA calculation. For this particular field test, the moving average filter did not seem to make any noticeable difference. However, the algorithms continued to run without causing any problems for the system (no missed events, system failures, etc.). Thus, although the conditions under which this field test were conducted did not require the use of a smoothing filter, certain applications that may require smoothing in order to function properly could make use of such a filter without fear of causing the goniometer to fail. The change to the DC offset was an attempt at adjusting all sensors by the same amount rather than subtracting each sensor's DC offset (average) from itself. The method used to calculate the shared DC offset was a simple weighted average where the weights were calculated based on the number of times each

sensor was used for the DOA calculation as part of a sensor pair. This algorithm did not produce desirable results insofar as it greatly reduced the accuracy of the acoustic goniometer.

CHAPTER SEVEN: FUTURE WORK

At the current stage, the general purpose acoustic goniometer research can be safely deemed successful. The design tests and simulations have proven the feasibility of the endeavor. The algorithms have been shown to produce an accurate DOA calculation to within an acceptable tolerance. Further, the sensor design has been successfully field tested in real-world conditions and proven on an easily deployable antenna structure with an embedded system recording and processing data stably at the desired frequency. However, this does not mean improvements to the design could not be made, nor does it indicate a lack of important facets to explore with the continued development of the general purpose acoustic goniometer. Several areas would greatly benefit from further research. The following sections detail possible areas for improvement and more exploration.

7.1 Algorithm Research

The firmware which is responsible for the acoustic goniometer functionality performs adequately and produces reasonably accurate results. However, a few areas of the firmware could benefit from further research. In particular, the scheduling (operating system) and event handling algorithms would be the best candidates for improvement.

7.1.1 Scheduling

In the current implementation, the scheduling tasks are handled almost entirely by an embedded real-time operating system (ChibiOS). Embedded real-time operating systems can be very beneficial in embedded system designs. However, the requirements

of the acoustic goniometer do not merit the use of such an advanced scheduling system. The acoustic goniometer in the current hardware consumes so much of the processor's resources that it would be impossible to do any other meaningful tasks (unless sampling is significantly reduced). Thus, complex scheduling for providing different tasks with adequate processor resources is neither done nor needed. ChibiOS was used as a means to expedite development of the acoustic goniometer for the purpose of proving the feasibility of the system but was never intended to be part of the final solution. The negatives to keeping ChibiOS in the goniometer implementation at this stage outweigh the benefits. Namely, the added complexity for further developing the goniometer in a multithreaded environment and the use of the processor's resources are negatively impacting improvements to the system. Developing code for a multithreaded operating system requires more from the firmware designer and adds new challenges to system development/modification, which could create a potential barrier to research for some scientists. Since the primary goal of this research was to lower barriers, removing this particular feature would be a step in the right direction. Additionally, ChibiOS consumes a fair number of processor resources and is responsible (in part) for the current limit on sampling speed. Switching to a simple, cooperative scheduling algorithm would free up system resources potentially allowing for faster sampling rates and increasing the utility/accuracy of the acoustic goniometer. This solution could also be done in such a way as to simplify the firmware, making it easier to understand and modify. Further research and development should be done in this area to determine the extent of the benefits gained by removing the operating system, and the new acoustic goniometer

firmware should be tested alongside another goniometer running the original firmware to make the most of the comparison.

7.1.2 Event Handling

The algorithms used by the current implementation of the acoustic goniometer have proven adequate to the task of event detections and DOA calculations. However, more research would be needed to determine whether these algorithms represent the best solution or simply a functional implementation. While determining the absolute best algorithms in general to use for acoustic goniometry may not be possible due to dependencies upon application (varying noise sources, cross-sensitivities between event signatures, etc.), some algorithms may prove more adept in a wider range of circumstances than others. The current acoustic goniometer design should be used to research and implement event detection and time delay estimation algorithms more thoroughly to determine the best implementation for a general purpose system (i.e., highest accuracy for the widest range of applications). This research would take a considerable investment of time and a deployment of acoustic goniometers to measure a significant number of phenomena. A project as staggering as this may require the cooperation of multiple agencies, but the determination of the best set of algorithms (or even more information on performance of the various combinations) would be invaluable to the further reduction of barriers to research in this field.

Event detection algorithms have been explored the least in the current research and would be the area of algorithm research, which would benefit the most from further experimentation. The current method for event detection employed by the goniometer prototype uses a simple threshold technique coupled with a predetermined data window

to detect an event. This simple method was easy to implement and requires few system resources, but it suffers from a few significant drawbacks. A purely threshold based algorithm can not differentiate between an event of interest and other phenomena (e.g., acoustic/electrical noise). Further, determination of the threshold can be problematic if the sound produced by the event of interest experiences drastic attenuation over distances or objects along the path of the sound create reflected versions of the signal leading to multiple detected events for a single phenomenon. Such problems have been mitigated in the current research by limiting signals of interest to the infrasound domain (little attenuation over large distances) and employing a windowing technique around a detected event in order to ignore reflections of the original sound. However, these solutions still do not address the issue of event differentiation (e.g., distinguish between an avalanche and a gun shot). As an alternative, fingerprinting and correlation could be explored instead of thresholding in order to make the process selective to events of interest. By first characterizing the acoustic fingerprint of an event and using the fingerprint as part of a correlation filter, events occurring in the same frequency range could be differentiated without putting a significant strain on system resources. The drawback to this method would be the difficulty in event characterization and the challenge of keeping the goniometer simple to use. The threshold method can be implemented such that a researcher could adjust the thresholds and window length to meet the needs of their research. A fingerprinting algorithm, however, would require them to somehow store the fingerprint of their event for use in the correlation filter. Another option worth further exploration is the use of the Fisher Statistic approach (see Chapter 3.5.3). Due to the computational complexity of this method, either the system

would need to be modified to use a more capable processor (e.g., the newest generation of the upgraded part from ST, SOC, DSP, etc.), or the firmware would need to be adjusted to use a lower sampling rate (limiting the viable frequency range of the goniometer) and to move from ChibiOS to a simpler (less resource hungry) cooperative scheduling algorithm. Still, other options for detecting events may exist which could be simple for any given researcher to use and solve the problem of event differentiation. Further research is needed to flesh these out and determine the best solution.

The other area of algorithm research that requires further exploration is the time delay calculation algorithm (or correlation algorithm). The current implementation of the acoustic goniometer uses a threshold to detect an event in the signal from each sensor followed by a standard correlation operation between each signal within an event window to determine the time delay for the event arriving at each sensor. Regardless of whether this method continues to be employed or the fingerprinting algorithm discussed in the preceding paragraph is employed instead, finding the best correlation method is an important aspect to furthering this research. Part of this question has been addressed in the current research (see the comparison of various algorithms in Chapter 4.2). However, the standard correlation approach selected as the best was determined under limited testing conditions. The raw data showed little evidence of unwanted noise especially in the frequency range of interest. All unwanted frequencies were easily removed with a simple digital low pass filter and a wind screen placed over the microphone. Under these conditions, the standard correlation method significantly outperformed the method of applying correlation in the frequency domain. However, according to [13], correlation in the frequency domain provides better performance in the presence of noise. Further

testing in noisier conditions could help to validate this claim and could make frequency domain correlation a more appealing option. Finally, the technique implemented for testing feature selection and comparison to determine time delays was extremely simple. If a more complex set of features was used (e.g., slope, second derivative of the signal, etc.), the results of such an approach might be improved significantly enough to make feature comparison a contender. However, using overly complex features could make the algorithm become so complex as to make it the worst choice for an embedded system with limited resources. Further testing is required to determine the viability of this method as well as its ranking among the other correlation algorithms. The acoustic goniometer developed as part of this dissertation would serve as the perfect platform for continuing this research and answering these questions.

7.2 Hardware Research

The hardware developed for the acoustic goniometer research presented in this dissertation has been proven adequate to the task of determining DOA and has been shown to be adaptable to fit multiple research needs. The design is flexible and can monitor any number of phenomena if the system is properly deployed with its gain and SD card parameter adjusted to meet the needs of the given project. Furthermore, the antenna can be easily reconfigured into any desired geometry by simply building a new antenna structure and modifying the dimension matrix in the SD card. However, as with any research project, the current implementation of the acoustic goniometer leaves room for improvement in multiple areas. The following paragraphs detail some ideas that would enhance the performance of the system and/or further lower barriers to research in the field of acoustic goniometry.

7.2.1 Motherboard

The acoustic goniometer motherboard performed well for the initial research and would serve as an excellent platform for further research in a myriad of projects. The processor has been proven to be capable of performing the requisite calculations and will support many types of algorithm modifications. However, as mentioned multiple times throughout this dissertation, the processor is a limitation for the exploration of several avenues of research. More complex algorithms (e.g., Fisher Statistic, frequency domain correlation, advanced noise filtering, etc.) and faster sampling/data processing could be done in an embedded platform but would require a faster system clock and/or more RAM. One solution that has been suggested is upgrading the processor to another part from ST Microelectronics' family of ARM processors. As of this writing, their current offering includes the STM32F7 part, which has nearly double the RAM and a clock rate that is 20% faster than that of the processor currently used by the acoustic goniometer (STM32F4). This processor was not selected for the current research for two reasons: it was not available at the inception of this project, and it was only released in a ball grid array (BGA) form factor during the course of the current research. The difficulties associated with a BGA part are seen in its placement and soldering. Special equipment would have been needed or the part would have needed to be soldered by a board house (thus increasing the cost). However, the STM32F7 is currently available in an easy-to-solder 144-pin LQFP with similar pin configurations to the STM32F4 currently being used. While a few modifications to the board would be necessary to use this new part, the STM32F7 would provide more system resources for testing more advanced algorithms or faster sampling speeds while allowing researchers to make use of the majority of the

current code base. All tradeoffs considered, this option would be the most prudent choice for the continued research of a general purpose acoustic goniometer.

Another viable option for a solution to the problem presented in the preceding discussion would be the use of an entirely different processing platform. Many potential avenues of exploration exist if one were to abandon the ST products or microprocessors in general. However, the unfortunate drawback to these options is the time required to redesign the system upon switching parts. An FPGA could be used to handle all of the acoustic goniometer processes and would provide potentially the fastest platform. Unfortunately, such an option would create a barrier to research for anyone desiring to modify the system but not versed in hardware description languages (HDLs). Another possibility which has been mentioned previously is the use of an SOC. If the FPGA were used only for simple unchanging functions (e.g., sampling the sensors, data pre-processing/filtering, etc.), developers could reap the benefits of using an FPGA for faster data collection and continue to make changes to the higher level algorithms without needing to know an HDL. As another alternative to the FPGA option, a high speed DSP part could be used in place of a standard microprocessor. These parts are designed for processing large quantities of data using floating point math in a timely manner. Switching to a DSP could provide more resources than the STM32F4 and allow for even more exploration of complex algorithms. All of these options would increase the system's capabilities and allow for more advanced research. However, the increased development time and drastically increased cost of these options should be noted. Even if time is less of a concern, the cost of these components could potentially make the acoustic goniometer too expensive for many research projects. Additionally, for the

majority of research projects which would use an acoustic goniometer, many of these options may supply far more resources/speed than is actually needed (thus increasing the cost for no tangible benefit). Thus, while these may be avenues worthy of further research, the tradeoffs should be carefully weighed before proceeding.

Another design aspect that could benefit from improvement would be the data storage capabilities. The current system uses SD cards to store raw data and DOA calculation results. This works well for short term tests (less than 2 weeks) at the current sample rate. However, if longer term testing was desired or the processor was upgraded to perform faster sampling, the use of SD cards would severely limit the goniometer's capabilities. Adding a solid state drive (SSD) would be the best option for improving the goniometer's data storage abilities. An SSD would provide sufficient storage space for long term tests and faster write speeds while still keeping power consumption reasonably low.

Finally, the motherboard includes a wireless transceiver for which code was developed and fully tested but which has never been used for any research purpose. The transceiver uses a short range radio but could be used for creating a mesh network of acoustic goniometers to monitor an area of interest. This feature opens up an entire area of cooperative data processing and event source pinpointing which should be explored. Small, inexpensive acoustic goniometers deployed to cooperatively monitor an area could be useful for a potential danger warning system or natural disaster monitoring system. Additionally, longer range radios could be added to expand the range of the network without having to increase the number of goniometers (or network repeaters) needed to maintain the network.

7.2.2 Sensors

The acoustic sensors created for the current acoustic goniometer are flexible in their use/modification abilities and have performed well. The reconfigurable nature of the amplifier circuit allows the sensor to incorporate active filters or even multiple microphones. While these features were not explored as part of the research for this dissertation, they should be used for future research into the benefits of redundant sensors and potential noise reduction. Additionally, a modification should be made to the gain stage of the acoustic sensor to incorporate a potentiometer instead of a fixed resistor. This minor change would allow for easy gain adjustment in the field and could save researchers valuable time.

Another area that would benefit from further research is the incorporation of other sensor technologies into the acoustic goniometer design. The current prototype simply samples ADC channels that could be connected to any analog signal. Other sensors (e.g., accelerometer, photo, pressure, etc) could be used in conjunction with the acoustic sensors to test sensor data fusion algorithms on an embedded goniometer platform. Some research has already been conducted in this area (see Chapter 2), but the use of a low-power embedded system for data collection could prove useful in certain environments/circumstances.

7.2.3 Mechanical Systems

One of the more difficult aspects of acoustic goniometry research is the deployment of the sensors. While the current acoustic goniometer developed for this research has addressed a major part of this problem (i.e., significant reduction of inter-sensor spacing), further development of the antenna structure is needed. As shown in

Chapter 5, the current prototype uses a PVC structure with tape holding the sensors in place. For easier short term deployment and to allow for long term deployment, this structure should be redesigned to include a better and more accurate sensor mounting system. Furthermore, more sturdy materials that would flex less in wind should be selected for the final implementation. These improvements could ease the deployment process and improve the performance/accuracy of the acoustic goniometer.

A significant amount of research was conducted in the area of wind noise reduction. However, a method for combatting severe wind (e.g., greater than 20 MPH) was not determined. While some research into this problem has been conducted by researchers in various fields of study, more work should be done to determine a more reliable means of mitigating the effect of this noise source.

CHAPTER EIGHT: CONCLUSIONS

The goal set forth for this research has been successful insofar as an inexpensive, small acoustic goniometer, which is easy to deploy and capable of being adapted to meet the needs of a wide range of research applications while maintaining reasonable accuracy, has been created. The current design has lowered barriers to research and provides several key advantages over existing goniometers including:

- **Real-time analysis capabilities** as compared with systems that collect data for post-event analysis
- **Low cost** as compared with systems requiring the use of data loggers, computers, and/or expensive sensors
- **High adaptability** as compared with systems having fixed geometries, fixed frequency ranges, and/or large inter-sensor spacing requirements
- **Small size** as compared with the average inter-sensor spacing range for acoustic goniometers (20-30 m)
- **Minimal power usage** as compared with systems requiring computers for real-time data analysis

This work has effectively lowered several barriers to research by providing not only a working prototype but also documentation of the design process, tradeoffs, and guidelines for future expansion of the research and use of the acoustic goniometer. Lessons learned from the research discussed in Chapter 2 have enabled the creation of an acoustic goniometer, which uses the best features from previous designs, improves upon several aspects, and provides a sensor array that will meet the needs of a plethora of research projects. The important characteristics of the theory behind acoustic goniometer

design and use has been detailed in Chapter 3. The explanation of the theory further includes a discussion of the most common algorithms, filtering techniques, and pertinent sound propagation theory making modification of the current design simpler and thus promoting more advanced research. Simulations that verify the theory and confirm the viability of the implementation of a small acoustic goniometer were created, detailed, and analyzed in Chapter 4. The results of the research conducted and the culmination of the information gleaned from all previous chapters are explained in Chapter 5 where the full implementation is described. Finally, the facets of the research left for future work were outlined in Chapter 7.

Acoustic monitoring of environments and phenomena have provided mankind with a better understanding of the world for centuries. Sounds can warn dangers both immediate and impending, provide a means of tracking movement, and supply information that helps increase the understanding of natural phenomena to the end that people can better prepare for natural disasters. Thus, the field of acoustic goniometry is a well explored area of study that cannot by any means be considered as a new area of study. Many sensor arrays have been created to support research projects and knowledge in the field continues to grow (see Chapter 2). However, no other project has endeavored to create a general purpose solution that can support a wide array of research projects and further work in the field of acoustic goniometry. The current design has achieved this goal with its design tailored to be adaptable to the range of research fields described in Chapter 2 as well as many others.

The simulations completed during the course of this research were successful in verifying the theory and proving the feasibility of the small scale acoustic goniometer.

They provided a good estimate of the highest achievable accuracy of the various algorithms (both separately and working in conjunction as a system) for the embedded implementation and highlighted potential sources for error. The significance of multipath was clearly established for an enclosed environment, and the corresponding simulation served to help determine some of the limitations of the acoustic goniometer design. Finally, the simulations played an important role in the development of the embedded implementation both by serving as a test platform to determine the best algorithms for the given test cases (gunfire and door closings) and by providing an easy means of determining the D matrix and its pseudo-inverse for the DOA calculations that were necessary when reconfiguring the goniometer antenna geometry.

The current firmware is flexible and provides efficient, accurate results. Furthermore, modification of the firmware is made simple by the current system in that the only requirement is the ability of the researcher to write/modify C code to fit their purpose. The algorithms selected to test the hardware are not the most advanced algorithms in the field, but they were adequate to the task of proving the system and achieved reasonable accuracy. Further, more advanced algorithms generally necessitate the research and characterization of a particular phenomenon in order to fully implement and evaluate their performance. Since this was outside the scope of the design of a general purpose goniometer, which would allow more specialized research to be completed, such research was left for those interested in studying specific phenomena. Some algorithms suited to general purpose systems were not evaluated due to their complexities and the limitations of the current hardware (e.g., Fisher Statistic approach). Given the promise shown by these algorithms, further exploration is warranted but would

require either different (more complex/advanced) hardware or significant modifications to the firmware. Removing the embedded operating system in favor of a less complex cooperative scheduling scheme would free up a significant amount of processor resources in the processor and could allow the current design to use more complex algorithms. The use of an embedded real-time operating system for an acoustic goniometer appears to have a cost that outweighs the benefits, and the system would at the very least be able to sample faster even if a more complex algorithm was not desired. Further study into algorithms and the use of a less complex scheduling scheme would still benefit from an analysis of the current design (the choices made and the analysis of the system's performance) and could easily make use of the research presented in this dissertation. The current firmware on the goniometer was successful for both DOA calculations and testing the capabilities of the hardware, and it allows room for and supports further exploration into more complex algorithms.

Design of a full implementation of the acoustic goniometer in firmware and embedded hardware has been completed, brought to fruition, and tested. The hardware chosen for this research was carefully selected after weighing all of the tradeoffs involved. While one could argue for the use of faster, more specialized processors, FPGA only implementations, or a combination of the two (SOC) for an improved design, the ease of continued development and modification had to be considered. More advanced hardware would provide resources to enable the use of more accurate algorithms and to increase sampling rates. However, these solutions would either greatly increase the cost of the system or create a more complex design requiring specialized skill sets to allow researchers to adapt the hardware for more specialized purposes. The current design

allows for rapid reconfiguration to adapt the antenna (both spacing and geometry), frequency range, filters (both analog and digital), and sampling speed to meet the needs of a particular research project. Additionally, changing sensor models or technologies requires only that a coaxial interface is made available for the motherboard to sample the analog signal of the new device. Although more robust solutions may be required for some research projects, enclosures have been designed and created to provide basic protection for the equipment, and reduction of environmental noise (e.g., wind noise) has been taken into consideration and addressed within reason as allowed for by the scope of this research and time constraints.

The research in its current state has been proven successful and shows promise for further exploration. Future simulation and experimentation could be done in a myriad of areas as detailed in Chapter 7. As many of the areas discussed in the preceding chapter as possible should be addressed. However, the primary focus of this research was the development of a general purpose acoustic goniometer that would lower barriers to research and encourage further development and research. All of the goals have been met, and the field of acoustic research has and will continue to benefit from the contributions of this work and the research that it will inspire.

REFERENCES

- [1] Bedard, Alfred J.; Nishiyama, Randall T., "Infrasound generation by large fires: experimental results and a review of an analytical model predicting dominant frequencies," Geoscience and Remote Sensing Symposium, 2002. IGARSS '02. 2002 IEEE International , vol.2, no., pp.876,878 vol.2, 2002
- [2] Nishiyama, R.T.; Bedard, A.J., Jr.; Kirschner, A. L., "Strong winds over mountains and infrasound: possible applications for detecting regions related to aircraft turbulence reports," Geoscience and Remote Sensing Symposium, 2002. IGARSS '02. 2002 IEEE International , vol.2, no., pp.879,881 vol.2, 2002
- [3] Zhu, X., Xu, Q., Zhou, J., and Tang, M. (2013). "Experimental study of infrasonic signal generation during rock fracture under uniaxial compression." International Journal of Rock Mechanics and Mining Sciences, Volume 60, June 2013, Pages 37-46, ISSN 1365-1609
- [4] Werner-Allen, G.; Johnson, J.; Ruiz, M.; Lees, J.; Welsh, M., "Monitoring volcanic eruptions with a wireless sensor network," Wireless Sensor Networks, 2005. Proceedings of the Second European Workshop on , vol., no., pp.108,120, 31 Jan.-2 Feb. 2005
- [5] Rud, S.W.; St.Jacque, N.; Vant, A.D.; Jiann-Shiou Yang, "Non-Invasive Infrasound Heart Murmur Detection with a Support Vector Machine (SVM) Classification Approach," Systems, Man and Cybernetics, 2006. SMC '06. IEEE International Conference on , vol.4, no., pp.3503,3508, 8-11 Oct. 2006
- [6] Simon, G., Maroti, M., Ledeczi, A., Balogh, G., Kusy, B., Nadas, A., Pap, G., Sallai, J., and Frampton, K (2004). "Sensor Network-based Countersniper System." SenSys '04 Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (2004).

- [7] Barger, J. E. (2007). U.S. Patent No. 7,292,501B2. "Compact Shooter Localization System and Method." Washington, DC: U.S. Patent and Trademark Office.
- [8] Barger, J. E., Milligan, S., Brinn, M., and Mullen, R. (2008). U.S. Patent No. 7,408,840B2. "System and Method for Disambiguating Shooter Locations." Washington, DC: U.S. Patent and Trademark Office.
- [9] Barger, J. E., Coleman, R., and Stanley, J. (2013). U.S. Patent No. 8,555,726B2. "Acoustic Sensors for Detecting Shooter Locations from an Aircraft." Washington, DC: U.S. Patent and Trademark Office.
- [10] Barger, J. E. and Stanley, J. (2010). U.S. Patent No. 7,787,331B2. "Sensor for Airborne Shooter Localization System." Washington, DC: U.S. Patent and Trademark Office.
- [11] Barger, J. E., Mullen, R., Cruthirds, D., and Coleman, R. (2012). U.S. Patent No. 8,320,217B1. "Systems and Methods for Disambiguating Shooter Locations with Shockwave-Only Location." Washington, DC: U.S. Patent and Trademark Office.
- [12] Barger, J. E., Milligan, S., Brinn, M., and Mullen, R. (2007). U.S. Patent Application Publication: Pub No. 2007/0237030A1. "Systems and Methods for Determining Shooter Locations with Weak Muzzle Detection." Washington, DC: U.S. Patent and Trademark Office.
- [13] Van Lancker, E. (2001). "Acoustic Goniometry: A Spatiotemporal Approach." Thèse École Polytechnique Fédérale de Lausanne (EPFL) Doctoral Dissertation (2001)
- [14] Vadam, V., Rossi, M., and Van Lancker, E. (1998). "Infrasonic Monitoring of Snow-Avalanche Activity: What do We Know and Where do We Go From Here?" *Annals of Glaciology*, v26, (1998).
- [15] Chritin, V., Rossi, M., and Bolognesi, R. (1996). "Snow Avalanches: Automatic Acoustic Detection for Operational Forecasting." *ACUSTICA* 82 (1996): S-173.
- [16] Scott, E., Hayward, C., Kubichek, R., Hamann, J., Pierre, J., Comey, B., Mendenhall, T. (2007). "Single and Multiple Sensor Identification of Avalanche-

- generated Infrasound.” *Cold Regions Science and Technology*, v47, Issues 1–2, p159-170, 1 January 2007.
- [17] Comey, R., and Mendenhall, T. (2004). "Recent Studies Using Infrasound Sensors to Remotely Monitor Avalanche Activity." *International Snow Science Workshop Proceedings*; Jackson, WY. 2004.
- [18] Kinnerup, R. (2011). "Ultra Low Frequency Infrasonic Measurement System." *Technical University of Denmark Master's Thesis* (2011).
- [19] Sugimoto, T.; Koyama, K.; Kurihara, Y.; Watanabe, K., "Measurement of infrasound generated by wind turbine generator," *SICE Annual Conference*, 2008, vol., no., pp.5,8, 20-22 Aug. 2008.
- [20] Chilo, J.; Lindblad, T., "A Low Cost Digital Data Acquisition System for Infrasonic Records," *Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, 2007. *IDAACS 2007. 4th IEEE Workshop on* , vol., no., pp.35,37, 6-8 Sept. 2007
- [21] Paros, J.; Migliacio, P.; Schaad, T., "Nano-resolution sensors for disaster warning systems," *OCEANS, 2012 - Yeosu* , vol., no., pp.1,5, 21-24 May 2012
- [22] Dickey, J.T.; Mikhael, W.B., "An adaptive technique for isolating the seismic response of an infrasound sensor," *Circuits and Systems (MWSCAS), 2012 IEEE 55th International Midwest Symposium on* , vol., no., pp.1028,1031, 5-8 Aug. 2012
- [23] Bedard, Alfred J., "Detection of infrasound from natural and civilization sources: measurement of complex signal/noise environments," *Geoscience and Remote Sensing Symposium*, 2000. *Proceedings. IGARSS 2000. IEEE 2000 International* , vol.3, no., pp.1195,1197 vol.3, 2000
- [24] Chilo, J., and Lindblad, T. (2007). "Real-Time Signal Processing of Infrasound Data Using 1D Wavelet Transform on FPGA Device," *Real-Time Conference, 2007 15th IEEE-NPSS* , vol., no., pp.1,5, April 29 2007-May 4 2007
- [25] Wei Wang; Shimin Wei; Qizheng Liao; Yaqin Xia; Danlin Li; Junzi Li, "Fuzzy K-means clustering on infrasound sample," *Fuzzy Systems*, 2008. *FUZZ-IEEE*

2008. (IEEE World Congress on Computational Intelligence). IEEE International Conference on , vol., no., pp.756,760, 1-6 June 2008
- [26] Ham, F.M.; Park, S., "A robust neural network classifier for infrasound events using multiple array data," *Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on* , vol.3, no., pp.2615,2619, 2002
- [27] Chilo, J.; Jabor, A.; Liszka, L.; Eide, A.J.; Lindblad, T.; Bergkvist, L.P.N.-O.; Stahlsten, T.; Andersson, B.L.; Karasalo, I.; Cederholm, A., "Filtering and extracting features from infrasound data," *Real Time Conference, 2005. 14th IEEE-NPSS* , vol., no., pp.5 pp., 10-10 June 2005
- [28] Ham, F.M.; Acharyya, R.; Lee, Y.-C.; Garces, M.; Fee, D.; Whitten, C.; Rivera, E., "Classification of Infrasound Surf Events Using Parallel Neural Network Banks," *Neural Networks, 2007. IJCNN 2007. International Joint Conference on*, vol., no., pp.720,725, 12-17 Aug. 2007
- [29] Renshi Li; Reddy, V.V.; Khong, A.W.H., "Quadratic phase coupling analysis for infrasound vehicle detection," *Circuits and Systems (APCCAS), 2010 IEEE Asia Pacific Conference on* , vol., no., pp.891,894, 6-9 Dec. 2010
- [30] Wu Yanjie; Fan Changyuan; Li Yiding; Wang Baoqiang, "Design of Lightning Location System Based on Photon and Infrasound Detection," *Electronic Measurement and Instruments, 2007. ICEMI '07. 8th International Conference on* , vol., no., pp.1-603,1-606, Aug. 16 2007-July 18 2007
- [31] Jacobsen, F., and Juhl, P. M. (2013). "Fundamentals of general linear acoustics." 6th Edition.
- [32] Guenneau, S., Movchan, A., Pjjetursson, G., and Ramakrishna, S (2007). "Acoustic Metamaterials for sound focusing and confinement." *New Journal of Physics*, vol. 9, (2007).
- [33] Havens, S., H.-P. Marshall, J. B. Johnson, and B. Nicholson (2014), Calculating the velocity of a fast-moving snow avalanche using an infrasound array, *Geophys. Res. Lett.*, 41, 6191–6198, doi:10.1002/2014GL061254.

- [34] Marcillo, O., J. B. Johnson, and D. Hart (2012), Implementation, characterization, and evaluation of an inexpensive low-power low-noise infrasound sensor based on a micromachined differential pressure transducer and a mechanical filter, *J. Atmos. Oceanic Technol.*, 29(9), 1275–1284, doi:10.1175/JTECH-D-11-00101.1.
- [35] Ulivieri, G., E. Marchetti, M. Ripepe, I. Chiambretti, G. De Rosa, and V. Segor (2011), Monitoring snow avalanches in Northwestern Italian Alps using an infrasound array, *Cold Reg. Sci. Technol.*, 69(2–3), 177–183, doi:10.1016/j.coldregions.2011.09.006.
- [36] Kogelnig, A., E. Surinach, I. Vilajosana, J. Huebl, B. Sovilla, M. Hiller, and F. Dufour (2011), On the complementariness of infrasound and seismic sensors for monitoring snow avalanches, *Nat. Hazards Earth Syst. Sci.*, 11(8), 2355–2370, doi:10.5194/nhess-11-2355-2011.
- [37] Smart, E., and E.A. Flinn (1971), Fast frequency-wavenumber analysis and fisher signal detection in real-time infrasonic array data processing, *Geophys. J. Int.*, 26(1–4), 279–284, doi:10.1111/j.1365-246X.1971.tb03401.x.
- [38] Blandford, R.R. (1974), An automatic event detector at the Tonto forest seismic observatory, *Geophysics*, 39(5), 633–643.
- [39] Portelo, J.; Bugalho, M.; Trancoso, I.; Neto, J.; Abad, A.; Serralheiro, A., "Non-speech audio event detection," *Acoustics, Speech and Signal Processing*, 2009. ICASSP 2009. IEEE International Conference on , vol., no., pp.1973,1976, 19-24 April 2009.
- [40] Atrey, P.K.; Maddage, M.C.; Kankanhalli, M.S., "Audio Based Event Detection for Multimedia Surveillance," *Acoustics, Speech and Signal Processing*, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on , vol.5, no., pp.V,V, 14-19 May 2006.
- [41] Li Lu; Fengpei Ge; Qingwei Zhao; Yonghong Yan, "A SVM-Based Audio Event Detection System," *Electrical and Control Engineering (ICECE)*, 2010 International Conference on , vol., no., pp.292,295, 25-27 June 2010.

- [42] Bardeli, Rolf; Stein, Daniel, "Uninformed Abnormal Event Detection on Audio," Speech Communication; 10. ITG Symposium; Proceedings of , vol., no., pp.1,4, 26-28 Sept. 2012.
- [43] Chin, M.L.; Burred, J.J., "Audio event detection based on layered symbolic sequence representations," Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on , vol., no., pp.1953,1956, 25-30 March 2012.
- [44] Sirio, G. (2015). ChibiOS Real Time Embedded Operating System. Retrieved from: <http://www.chibios.org/dokuwiki/doku.php>

APPENDIX A

Building the Acoustic Goniometer

The table below is the Bill of Materials (BOM) for the acoustic goniometer motherboard and sensors. The board and schematic files can be obtained by contacting Dr. Sin Ming Loo at smloo@boisestate.edu.

Table A.1 Acoustic Goniometer Motherboard Bill of Materials

Motherboard BOM				
Qty (per build)	Value	Device	Parts	Supplier P/N
5	1uF	C-USC0603	C5, C8, C16, C29, C36	
32	0.1uF	C-USC0603	C2, C25, C44, C46, C47, C50, C53, C54, C55, C57, C58, C59, C61, C62, C63, C65, C66, C67, C68, C70, C71, C72, C73, C74, C75, C76, C78, C79, C81, C82, C83, C84	
1	0.01uF	C-USC0603	C4	
4	2.2uF	C-USC0603	C56, C64, C69, C80	
6	1nF	C-USC0603	C7, C15, C28, C35, C60, C77	
4	5pF	C-USC0603	C18, C21, C34, C38	
4	6pF	C-USC0603	C22, C24, C39, C43	
1	22uF	C-USC1812	C10	
4	33nF	C-USC0603	C26, C45, C48, C49	
2	47uF	C-USC1206	C51, C52	
1	68uF	CPOL-USD	C11	PCE5006CT-ND
2	470uF	CPOL-USCT7343	C23, C42	495-1537-1-ND
18	NO_LOAD	C-USC0603	C1, C3, C6, C9, C12, C13, C14, C17, C19, C20, C27,	na

			C30, C31, C32, C33, C37, C40, C41	
8	1k	R-US_R0603	R7, R12, R15, R22, R32, R46, R50, R59	
1	5k	R-US_R0603	R6	
2	10	R-US_R0603	R45, R68	
8	10k	R-US_R0603	R1, R9, R70, R71, R77, R78, R79, R80	
10	20	R-US_R0603	R17, R18, R21, R23, R49, R51, R53, R55, R65, R66	
1	22k	R-US_R0603	R4	
10	68k	R-US_R0603	R26, R27, R28, R29, R31, R58, R60, R61, R63, R64	
2	100k	R-US_R0603	R5, R69	
1	200k	R-US_R0603	R2	
5	150k	R-US_R0603	R20, R24, R52, R56, R67	
1	30.9k	R-US_R0603	R10	
31	NO_LOAD	R-US_R0603	R3, R8, R11, R13, R14, R16, R19, R25, R30, R33, R34, R35, R36, R37, R38, R39, R40, R41, R42, R43, R44, R47, R48, R54, R57, R62, R72, R73, R74, R75, R76	na
5	APHFT1612PBA SURKVGAC	RGBLED	RGB1, RGB2, RGB3, RGB4, RGB5	350-2096-1-ND
2	CRYSTAL_ABS2 5-32.768KHZ	Crystal	Y2, Y4	535-10241-1-ND
2	16MHz OSC	7M- 12.000MAAJ- T	Y1, Y3	644-1100-1-ND
8	COAX_RA_MO LEX_WM5387	COAX	J1, J2, J3, J4, J5, J6, J7, J8	WM5387-ND
3	FFC10_10FMN- SMT-A-TF	FFC10_10FM N-SMT-A-TF	FFC1, FFC2, FFC3	455-1934-1-ND
4	OPA2314	LMC6482	U4, U5, U8, U10	296-30336-5-ND
2	MICROSD	MICROSD	J10, J11	HR1964CT-ND
2	ULQ2003	ULQ2003	U6, U9	ULQ2003-nd
2	STM32F103ZE	STM32F103Z E	U7, U11	497-11766-ND
1			U12	ATZB-24-

	ZIGBIT_DC_EXT ANT	ZIGBIT_DC_E XTANT		A2RCT-ND
1	LM2734	LM2734	U3	LM2734XMK/ NOPBCT-ND
1	TL331	TL331	U2	296-10168-1- ND
1	22u	SRR6038SRR 6038	L1	SRR6038- 220YCT-ND
1	1103M2S3AQE 2	1103M2S3A QE2	S1	EG2478-ND
1	B230A	B230A	D1	B230A-FDICT- ND
1	BAT30	1N4148	D2	B0540WSDICT -ND
1	BHX1-1025	BHX1-1025	BT1	BHX1-1025- ND
1	CP-102AH-ND	CP-102AH- ND	J9	CP-102AH-ND
1	FERRITEBEAD	FERRITEBEAD	FB1	445-5218-1- ND
1	LMV431	LMV431SOT2 3-5	U1	LMV431AIM5 /NOPBCT-ND
1	MBR130TIG	MBR130TIG	D4	MBR130T1GO SCT-ND
1	MBRS130LT3	MBRS130LT3	D3	MBRS130LT3 GOSCT-ND

Table A.2 Acoustic Goniometer Sensor Board Bill of Materials

Sensor BOM				
Qty (per build)	Value	Device	Parts	Supplier P/N
1	0.33uF	C-USC0603	C21	
5	1nF	C-USC0603	C9, C10, C11, C13, C24	
7	1uF	C-USC0603	C5, C6, C7, C8, C12, C25, C27	
2	3.3uF	C-USC0603	C19, R18	
1	3.6nF	C-USC0603	C20	
5	4.7uF	C-USC0805	C1, C2, C4, C15, C26	
4	10uF	C-USC0805	C17, C18, C22, C23	
1	10uF	CPOL-USSMCB	C3	
2	NO_LOAD	C-USC0603	C14, C16	na

2	0	R-US_R0603	R2, R6	
1	1k	R-US_R0603	R3	
6	2.2k	R-US_R0603	R7, R12, R23, R28, R29, R30	
3	10k	R-US_R0603	R8, R13 ,R16	
1	27k	R-US_R0603	R10	
1	39	R-US_R0603	R26	
3	133k	R-US_R0603	R17, R22, R27	
2	300k	R-US_R0603	R11, R14	
10	NO_LOAD	R-US_R0603	R1, R4, R5, R9, R15, R19, R20, R21, R24, R25	na
3	COAX_RA_MOL EX_WM538	COAX	J1, J2, J3	WM5387- ND
1	DP4T_SWITCH- 450-1619	DP4T_SWITCH -450-1619	SW1	450-1619- nd
1	LM4051_REGUL ATOR	LM4051_REGU LATOR	U1	LM4051- ND
2	MIC	MIC	JP1, JP2	668-1153- nd
1	MPZ2012S331A	L-USL2012C	FB1	445- 5218-1- ND
1	OPA2314	LMC6482	U2	296- 30336-5- ND

APPENDIX B

Deploying the Acoustic Goniometer

This appendix describes setup, deployment, and data analysis for the acoustic goniometer. The firmware source code, Matlab code, and the data analysis program can be obtained by contacting Dr. Sin Ming Loo (smluo@boisestate.edu). The default settings for the acoustic goniometer as designed for the dissertation assume a regular tetrahedral antenna with an inter-sensor spacing of approximately 2.032 m, and the application is assumed to be a door closing event (event detection window/threshold values). If this is the desired configuration, the reader may skip the “Preparation” section and move straight to deployment.

Preparation

Preparing the acoustic goniometer for deployment is a fairly straightforward process. In order to perform DOA calculations, the goniometer motherboard must know the geometry of its antenna and the orientation of the coordinate system in which it is to operate. To change inter-sensor spacing, geometry, or antenna orientation (or all three), the researcher has to provide the goniometer motherboard with an updated Dinv matrix (inverse position matrix) on the system SD card. Additionally, the system’s event detection algorithms can be modified by adjusting SD card parameters to fine tune the system for a particular application.

The Matlab code should allow a researcher to easily perform the calculations for either a tetrahedral or cubical antenna geometry. If one of these geometries is desired,

only a few steps are needed to calculate D_{inv} . First, open either *Generalized_Tetrahedron_Sim.m* or *Generalized_Cube_Sim.m* in Matlab. Next, find the instantiation of the variable “d” (see line 6 in Figure B.1 a). Variable “d” defines the inter-sensor spacing and can be adjusted to fit the needs of the researcher. If a change in orientation is desired, the research can change the position of the sensors by editing the values of x_1 - x_4 (see lines 15-18 in Figure B.1 a). At this point, the sensor pairs that are used for the DOA calculation should be noted. Figure B.1 (b) shows the calculations for the D and D_{inv} matrices. On line 48 (Figure B.1 b), the D matrix is defined by the sensor pairs x_2 - x_1 , x_3 - x_1 , x_4 - x_1 , and x_4 - x_3 . These are the pairs used by the acoustic goniometer firmware to perform the DOA calculation. **NOTE:** Changing the actively used sensor pairs would require a change to the firmware. Changing the sensor pairs in the Matlab code **WILL CAUSE THE DOA CALCULATIONS TO FAIL**. Finally, once all edits are complete, run the simulation. Type “format long; D_{inv} ” in the Matlab terminal, and hit Enter to view the calculated D_{inv} matrix.


```

Config.ini - Notepad
File Edit Format View Help
SENSOR.SOUND.CAL_TABLE_SIZE = 14
SENSOR.SOUND.DB_LIST = "48.0,50.0,55.0,60.0,65.0,70.0,75.0,80.0,85.0,90.0,95.0,100.0,105.0,110.0"
SENSOR.SOUND.CAL_LIST = "0.61,0.622,0.63,0.6468,0.665,0.7227,0.8090,0.9853,1.2540,1.6470,1.996,2.462,3.154,3.285"
SENSOR.SOUND.DBA_OFFSET = 0
SENSOR.TEMP.INTERVAL = 100 # number of SCHEDULER_SEC intervals, usually 100 = 1 s
SENSOR.TEMP.SCALE = 1
SENSOR.TEMP.OFFSET = 0
SENSOR.VBATT.INTERVAL = 100 # number of SCHEDULER_SEC intervals, usually 100 = 1 s
SENSOR.VBATT.SCALE = 4
SENSOR.VBATT.OFFSET = 0
SENSOR.ADCREADER.SPEED = 7
# Goniometer settings: DINV assumes a sensor spacing of 80 inches or 2.032meters
SENSOR.GONIOMETER.DINV = "-0.492125984251969,0.328083989501312,0.164041994750656,-0.164041994750656,0.284129069483084,0.189419375"
SENSOR.GONIOMETER.NUM.SENSPAIR = 4
SENSOR.GONIOMETER.THRESHOLD = 1.5 # Door event: 1.55; ARI5: 1.5
SENSOR.GONIOMETER.EVENT.PWINDOW = 50 # Door event: 200; ARI5: 50
SENSOR.GONIOMETER.EVENT.WINDOW = 300 # Door event: 300; ARI5: 300
SENSOR.GONIOMETER.EVENT.LENGTH = 10 # Door event: 10; ARI5: ?
SENSOR.GONIOMETER.FILTER = 0

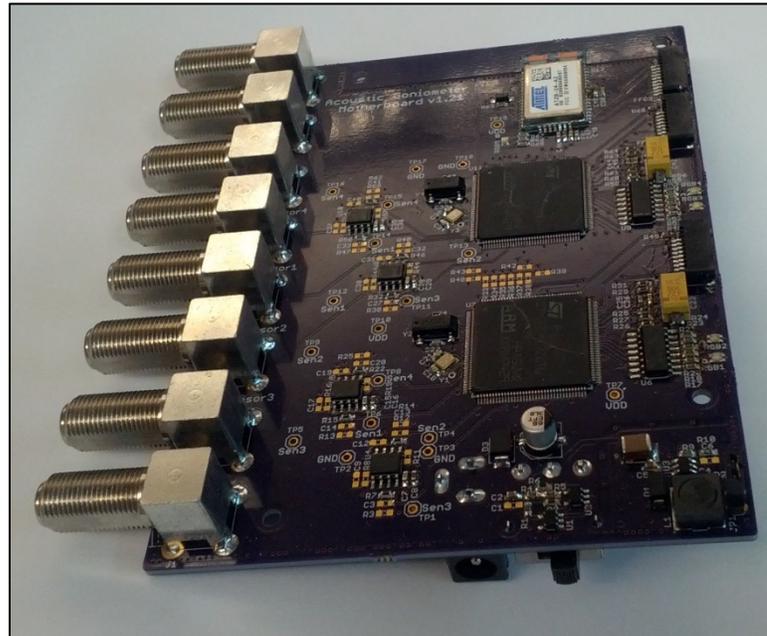
```

Figure B.2 Goniometer Configuration File

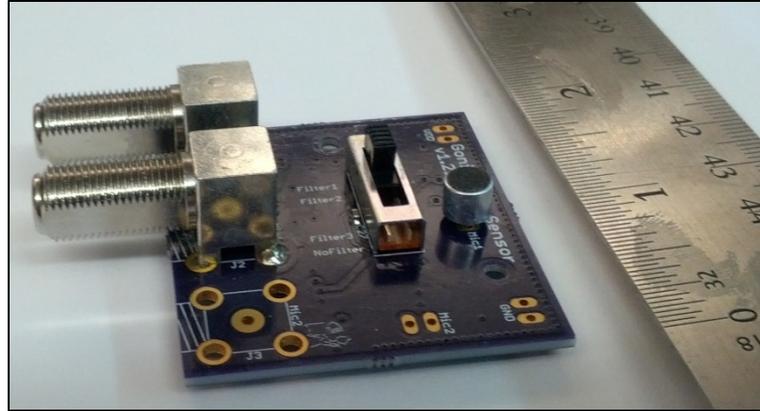
Besides the Dinv matrix, several other values which control the goniometer's operation are shown in Figure B.2: *threshold*, *window*, *pwindow*, *length*, and *filter*. These values should be edited and used to fit particular research applications. *Threshold* sets the voltage at which an event is registered. A drop by any sensor below this value will cause the goniometer to register an event and calculate a DOA (azimuth and elevation). *Window* defines the length of the total event window (in samples) and affects the accuracy of the correlation algorithm. *Pwindow* sets the number of samples prior to the event which is used as part of the event window (also affects system accuracy). *Length* is used to set the number of filled buffers between detected events. This value is used to specify the time expected for the sensor signals to stabilize after an event occurs. A value of 10 roughly equates to an expected time of 1 second between events ($1024 \text{ samples_per_buffer} * 10 \text{ buffers} / 10600 \text{ samples_per_second}$). Finally, *filter* is a value ranging from 0 to 7 which selects various digital filtering options and debug values to be stored on the SD card. This feature did not produce positive results under test conditions but is still documented in the firmware (see Appendix C).

Deployment

Once the SD card values are set, deploying the acoustic goniometer requires very little effort. After assembling the structure for the antenna and attaching the sensors at the desired locations, the cables must be connected. On the motherboard (see Figure B.3 a), the power terminals are the coaxial connectors designated J1, J2, J7, and J8. The sensor signal terminals are the coaxial connectors designated J3, J4, J5, and J6. On the sensor boards (see Figure B.3 b), the power terminal is the one marked J1, and the signal terminal is the one marked J2. After connecting the cables, insert the goniometer SD card (the one with the Dinv matrix) into the slot closest to the power switch (J10). Insert the ADC Reader SD card into the slot closest to the Zigbit (J11). Finally, plug in the battery and slide the power switch to the “on” position.



(a) Motherboard



(b) Sensor

Figure B.3 Acoustic Goniometer**Data Analysis**

Analysis of the data post collection can be done using the Hex File Processor program (see Figure B.4). The program values can be adjusted (allowing for modification to the hardware in future revisions). However, the default values are appropriate for the current setup of the acoustic goniometer. Start by clicking the “Create CSV” button to open a file dialogue. Select the raw data file created by the ADC Reader (**ONLY 1 at a TIME**) and click “Open.” The program will parse the data and create a “.csv” file with the same name as the original (in the same directory). This file can then be used to create graphs using Excel, Matlab, or any other program capable of recognizing the .csv format.

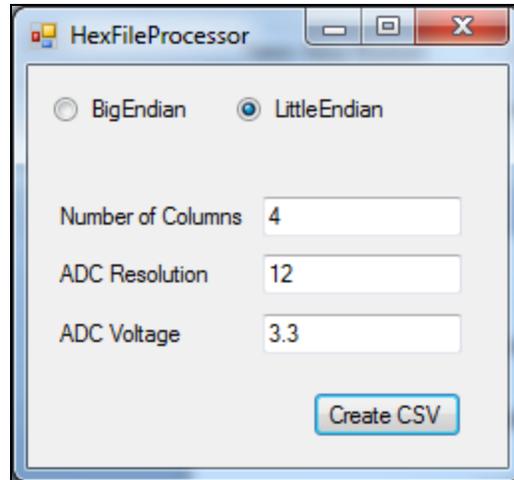


Figure B.4 Hex File Processor: Raw ADC Reader Data to CSV Converter

If desired, the *Goniometer.m* file can be used in conjunction with Matlab to analyze the data and find better values (window, pwindow, threshold, and length) for the goniometer configuration file. The file is a parameterized function which can be called in a Matlab terminal using the following syntax: “[Az, El, t_all] = Goniometer(data, rmin, rmax, d, Ts);” where Az is azimuth, El is elevation, t_all is a matrix holding differences in TOA between sensor pairs, data is the raw data, rmin is the start of the event window, rmax is the end of the event window, d is the inter-sensor spacing, and Ts is the sample period.

APPENDIX C

Understanding the Acoustic Goniometer Firmware

The acoustic goniometer firmware is available to anyone wishing to use/continue this research and can be obtained by contacting Dr. Sin Ming Loo (smloo@boisestate.edu). The purpose of this appendix is not to detail every line of code in the firmware but should provide a researcher with adequate background to ease the process of modification. The acoustic goniometer uses a real-time embedded operating system (ChibiOS) to handle low level hardware communications and scheduling tasks. No further information is provided here, but full detail of the operating system can be found at www.chibios.org. Fully understanding the operation of ChibiOS is not necessary for modifying the goniometer or raw data collection algorithms.

The acoustic goniometer firmware is controlled/configured by a file named “GONIOMETER_DEV.h” found in the source’s “boards\Goniometer\Platform” directory. This file determines the mode in which the system operates: Goniometer or ADC_Reader (raw data collection). Figure C.1 shows the constants used to determine the mode of the system: *SENSOR_EN_GONIOMETER* and *SENSOR_EN_ADC_READER*. The mode is selected by setting 1 of these constants (ONLY 1) to “ENABLES” while the other is set to “DISABLES.” For the current hardware, one processor is programmed as a goniometer, and the other is programmed for raw data collection. The function of the processors is interchangeable. However, if use of the wireless radio is desired, the

processor closest to the radio should be programmed as the transmitting processor (only this processor has access to the wireless radio).

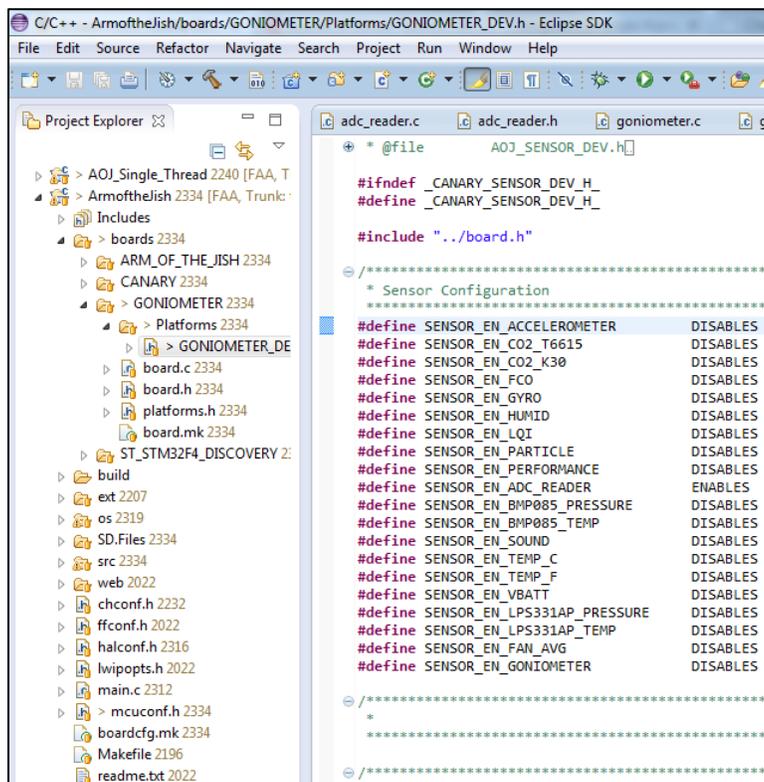


Figure C.1 GONIOMETER_DEV.h- Selecting the Operating Mode

While the two modes perform very different functions, they both share a common need to store large quantities of data before either processing (goniometer) or writing to an SD card (ADC reader). Due to the size of the buffers used in the current implementation, keeping the buffers in normal program memory is not an option. Instead, they are stored in CCM RAM. This is done by tagging the instantiation with an “__attribute__” tag recognized by the linker as shown in Figure C.2.

```

#if SENSOR_EN_GONIOMETER == ENABLES
/*****
 * Global Variables
 *****/
//static unsigned int Scheduler_Flag;           // Scheduler callback flag for this sensor.
static unsigned int ADC_Flag;                 // ADC callback flag for this sensor.
static SensorData_t Sensor_Data_AZ, Sensor_Data_EL; // Structure for managing the sensor data.
static bool eldataReady;

volatile GoniometerStorageBuffer_t adclog_buffer __attribute__((section(".ccmbuf"))) __attribute__((aligned(4)));
static adcsample_t goniometer_sample_buffer[GONIOMETER_ALL_CH_BUFFER_SIZE];

```

(a) Goniometer Storage Buffer Instantiation

```

static char* triggerStr;

#if SENSOR_EN_ADC_READER == ENABLES
static int buffer_counter= 0;
static FILE hADCReaderLog;
volatile ReaderStorageBuffer_t adclog_buffer __attribute__((section(".ccmbuf"))) __attribute__((aligned(4)));
static char ccm_bypass_buffer[READER_ALL_CH_BUFFER_SIZE * READER_SAMPLE_SIZE];
#endif

```

(b) ADC Reader Storage Buffer Instantiation

Figure C.2 CCM RAM Buffer Instantiation

The remainder of the code is application specific depending upon which mode is chosen. The goniometer code is contained entirely within two files: *goniometer.c* and *goniometer.h*. The ADC Reader code uses 3 files: *adc_reader.c*, *adc_reader.h*, and *fs.c*. Other files are included in the source which may be of interest to more advanced users (e.g., *network.c*, *zigbit.c*, *data_manager.c*, *sensor.c*, etc.) as they allow for expanded capabilities related to wireless sensor networks, creating SD card controlled variables, and data management. These are not covered in detail in this appendix. However, the source includes ample commenting which should provide adequate guidance to the advanced user.

Goniometer

As with all other code modules in the source, the goniometer is split into 2 files. The header file contains all of the constants which define the operation of the acoustic goniometer (e.g., ADC channels used, storage buffer size, speed of sound, etc.). The c file

houses the actual algorithms. The goniometer algorithm uses 4 functions:

goniometer_adc_callback, *Sensor_Goniometer_Init*, *Sensor_Goniometer_AZ_Task*, and *Sensor_Goniometer_EL_Task*.

The ADC on the microprocessor is sampled via direct memory access (DMA) and can not store data directly to CCM RAM. Thus, the responsibility of the callback function is to pull data out of the program memory storing the ADC values and store them in the CCM RAM. This function is automatically called by the system every time the program memory storage buffer is half full.

The initialization (*Init*) function is called once at startup and sets the initial values (including grabbing the relevant settings from the SD card) for the goniometer.

Additionally, the initialization function starts the ADC sampling process using the *goniometer_convgrp* variable shown in Figure C.3 (full documentation for controlling the ADC can be found on the ChibiOS website).

```

*****
*   ADCConversionGroup
*   Discussion:   Objects created to specify the specific ADC conversion
*               operation, one for continuous, one for single conversion
*               enables circular buffer mode
*   bool_t      adc_channels_num_t  Number of analog channels belonging to the group
*   adccallback_t  callback function associated with the group
*   adccallback_t  error callback function
*   uint32_t     ADC CR1 register initialization data
*   uint32_t     ADC CR2 register initialization data
*
*   uint32_t     ADC SMPR1 register initialization data
*               In this field must be specified the sample times
*               for channels 10...18
*
*   uint32_t     ADC SMPR2 register initialization data
*               In this field must be specified the sample times
*               for channels 0...9
*
*   uint32_t     ADC SQR1 register initialization data
*               Conversion group sequence 13...16 + sequence length
*
*   uint32_t     ADC SQR2 register initialization data
*               Conversion group sequence 7...12 + sequence length
*
*   uint32_t     ADC SQR3 register initialization data
*               Conversion group sequence 1... 6 + sequence length
*****
static ADCConversionGroup goniometer_convgrp =

```

Figure C.3 ADC Initialization Variable for Setting Channels, Sample Rate, and Resolution

Finally, the task functions are where the goniometer calculations take place. The *EL_Task* performs no function other than to provide the operating system with an indication when elevation data is ready. The *AZ_Task* actually contains the entire goniometer state machine (as described in Chapter 5). The state machine is well documented with comments and intuitive naming conventions. The states are clearly separated and have comment blocks defining the “Event Detection,” “Event Windowing,” and “DOA Calculation” portions.

ADC Reader

The ADC reader code is essentially a more simplified version of the goniometer code. The *adc_reader.h* file controls the same features for the ADC reader that *goniometer.h* controls for the goniometer (e.g., ADC channels used, storage buffer size, speed of sound, etc.). The c file uses the same callback function and initialization function setup (including the ADC initialization variable). The key difference (besides the lack of DOA calculation) is found in the operation of the callback function. Instead of directly accessing the CCM RAM buffer, the callback function for the ADC reader passes a pointer to the program memory buffer to a function in *fs.c* called *start_adcreader_write*. This function copies the data into the CCM RAM storage buffer and sets a flag which indicates that data is waiting to be stored to the SD card. When the SD card is allowed processor time, its task checks for available data, pulls it from the CCM RAM storage buffer, and writes the data to the SD card.