

AN EMBEDDED REAL-TIME PASSIVE UNDERWATER ACOUSTIC
LOCALIZATION SYSTEM USING A COMPACT SENSOR ARRAY

by

Jordin McEachern

Submitted in partial fulfillment of the requirements
for the degree of Master of Applied Science

at

Dalhousie University
Halifax, Nova Scotia
April 2023

© Copyright by Jordin McEachern, 2023

Table of Contents

| | |
|--|-------------|
| List of Tables | iv |
| List of Figures | v |
| Abstract | viii |
| List of Abbreviations and Symbols Used | ix |
| Acknowledgements | xiii |
| Chapter 1 Introduction | 1 |
| 1.1 Background | 1 |
| 1.2 Research Objectives and Contributions | 5 |
| 1.3 Organization of Thesis | 7 |
| Chapter 2 Localization using Compact Arrays | 8 |
| 2.1 Time Difference of Arrival | 8 |
| 2.2 Definition of Cross-correlation Algorithms | 9 |
| 2.2.1 3D Position Estimation | 11 |
| 2.3 Sensor Geometry Evaluation | 14 |
| 2.3.1 Estimation Limitations | 14 |
| 2.3.2 Simulation | 15 |
| 2.3.3 Improvement | 16 |
| 2.4 Delay Estimation Accuracy in the Presence of Noise | 21 |
| 2.4.1 Accuracy After Interpolation | 23 |
| 2.5 Low-complexity Matched Field Processing Algorithm | 24 |
| Chapter 3 Implementation on Embedded Processor | 29 |
| 3.1 Proposed System on Chip Architecture | 29 |
| 3.2 Interface to the Sensor Array | 31 |
| 3.2.1 TracFind | 31 |
| 3.2.2 TracIO | 33 |
| 3.2.3 TracStream | 35 |

| | | |
|---------------------|---|-----------|
| 3.3 | Petalinux | 36 |
| 3.3.1 | The File System | 36 |
| 3.3.2 | Petalinux Design Flow | 36 |
| 3.3.3 | Serial Communication | 40 |
| Chapter 4 | Experimentation at Sea | 42 |
| 4.1 | Preliminary Tests in the Aquatron | 42 |
| 4.2 | Parrsboro Berm Trial | 47 |
| 4.3 | Herring Cove Sea Trial | 50 |
| 4.4 | Long Term Deployment in the Bay of Fundy | 54 |
| 4.5 | New Zealand Sea Trial | 57 |
| Chapter 5 | Conclusion | 64 |
| 5.1 | Summary of Contributions | 64 |
| 5.2 | Future Work | 67 |
| Bibliography | | 69 |
| Appendix A | Conferences and Publications | 72 |
| A.1 | Annual Electrical and Computer Engineering Graduate Conference | 72 |
| A.2 | 20th IEEE Interregional NEWCAS Conference | 72 |
| A.2.1 | CMC Microsystems TEXPO 2022 Graduate Student Research Competition | 72 |
| A.3 | 16th International Conference on Underwater Networks & Systems | 73 |
| Appendix B | Software and Tools | 75 |

List of Tables

| | | |
|-----|---|----|
| 2.1 | Sensor positions on the remote platform in the Bay of Fundy | 18 |
| 2.2 | Sensor positions of the deployed planar array in Herring Cove | 20 |
| 4.1 | Sensor positions on the platform in the Aquatron trial | 44 |
| 4.2 | Summary of trials performed during the Berm deployment, including the transmitters used, start time, and end time for each trial. | 49 |
| 4.3 | Sensor positions on the starboard trawl door in New Zealand | 58 |
| 4.4 | Sensor positions on the port trawl door in New Zealand | 59 |

List of Figures

| | | |
|------|---|----|
| 1.1 | Sample of a harbour porpoise clicks received on 5 channels. | 3 |
| 1.2 | Capture of the harbour porpoise map in Grand Passage. | 6 |
| 2.1 | Far field approximation of TDoA AoA. | 9 |
| 2.2 | Intersection of multiple hyperboloids. | 12 |
| 2.3 | Azimuth and inclination error as a function of height using a tetrahedral structure using the TDoA algorithm. | 16 |
| 2.4 | Range and depth error as a function of height using a tetrahedron structure. | 17 |
| 2.5 | Sensor positions on the remote platform in the Bay of Fundy. | 18 |
| 2.6 | Mean inclination and azimuth error as a function of height for the structure deployed in the Bay of Fundy. The grey line shows the physical height of the sensor. | 19 |
| 2.7 | Sensor positions of the deployed planar array in Herring Cove. | 20 |
| 2.8 | Representative sample of the spectrum during the Herring Cove Trial deployment. | 21 |
| 2.9 | Azimuth estimation of the vessel in the Herring Cove deployment compared with the true azimuth obtained using the GPS. | 22 |
| 2.10 | Sensor geometry constrained to the trawl door, adapted from the Herring Cove geometry and improved to minimize the localization error. | 23 |
| 2.11 | Azimuth estimation of the vessel in the berm test compared with the true azimuth obtained using the GPS. | 24 |
| 2.12 | Mean delay estimation error of different pre-processing techniques as a function of SNR. | 25 |
| 2.13 | Mean delay estimation error of different pre-processing techniques using 64x linear interpolation as a function of SNR. | 26 |
| 2.14 | Comparison of the position estimation obtained using the proposed and TDoA localization algorithms on data from the spiral simulation. | 27 |

| | | |
|------|--|----|
| 2.15 | Comparison of the range and azimuth estimation error distributions between the proposed and TDoA localization algorithms on data from the spiral simulation. | 28 |
| 3.1 | Firmware architecture for the real-time remote sensor. | 30 |
| 3.2 | Network diagram of the network containing the ZYBO Z7 and the pressure case. | 31 |
| 3.3 | Activity diagram for the program running on the Zynq Development Board. | 32 |
| 3.4 | Program sequence diagram for the TracFind protocol. | 33 |
| 3.5 | Program sequence diagram for the Trac protocol. | 34 |
| 3.6 | Program sequence diagram for the TracStream protocol. | 35 |
| 3.7 | Two-partition file system used by Petalinux. | 37 |
| 3.8 | Exporting the Hardware Descriptor File using Vivado. | 38 |
| 3.9 | Using Petalinux Config to create the Device Tree Source (DTS). | 38 |
| 3.10 | Using Petalinux Build to create the DTB, bootloaders, kernel, file system, and boot images. | 39 |
| 3.11 | Using Petalinux Package to create the filesystem image, bootloader, and kernel image. | 40 |
| 3.12 | Complete workflow of configuring and building Petalinux using custom designed hardware. | 41 |
| 4.1 | Measurements of the Aquatron trial (top view). | 43 |
| 4.2 | Measurements of the Aquatron trial (side view). | 43 |
| 4.3 | Sensor positions on the platform in the Aquatron trial. | 44 |
| 4.4 | Representative sample of the 27.5 kHz clicks received during the Aquatron trial in the frequency domain (top) and time domain (bottom). | 45 |
| 4.5 | The initial deployment of the platform deployed in Parrsboro on the berm. | 48 |
| 4.6 | The recovery of the platform deployed in Parrsboro on the berm. | 48 |
| 4.7 | Depth of the water column measured using the ADCP during the Parrsboro berm deployment. | 50 |

| | | |
|------|--|----|
| 4.8 | Sound speed in the water column measured using the ADCP during the Parrsboro berm deployment. | 51 |
| 4.9 | Measured delays between sensors using the icTalk at different depths and waypoints in the Herring Cove deployment. | 52 |
| 4.10 | Azimuth estimation of the icTalk at different depths and waypoints in the Herring Cove deployment compared with the true azimuth obtained using the GPS. | 53 |
| 4.11 | Inclination estimation of the icTalk at different depths and waypoints in the Herring Cove deployment compared with the true inclination obtained using the GPS. | 53 |
| 4.12 | Representative sample of the spectrum during the Parrsboro deployment. | 55 |
| 4.13 | Sample of clicks received on 5 channels in the Parrsboro deployment. | 55 |
| 4.14 | Measured delays between sensors in the Parrsboro deployment. | 56 |
| 4.15 | Capture of the harbour porpoise map in Parrsboro. | 57 |
| 4.16 | Sensor positions on the starboard trawl door in New Zealand. | 58 |
| 4.17 | Sensor positions on the port trawl door in New Zealand. | 59 |
| 4.18 | Measured delays between sensors in the New Zealand deployment. | 60 |
| 4.19 | Measured delays between sensors in the New Zealand deployment, filtered by removing outliers using a threshold on the moving median. | 60 |
| 4.20 | Representative sample of the spectrum during the New Zealand deployment. | 61 |
| 4.21 | Sample of the multiple sources observed in the New Zealand deployment. | 61 |
| 4.22 | Azimuth estimation of the clicks identified in the New Zealand deployment. | 62 |
| 4.23 | Inclination estimation of the clicks identified in the New Zealand deployment. | 62 |
| 4.24 | Azimuth and range estimation of the clicks identified in the New Zealand deployment, shown on a polar plot. | 63 |
| A.1 | Poster presented during the TEXPO 2022 Graduate Student Research Competition. | 74 |

Abstract

In this thesis, a passive underwater acoustic localization system using a compact sensor array is developed which receives underwater acoustic sensor data and outputs the position estimations to a map display in real-time. Through simulation, it is evaluated using 130 kHz pulses, which is representative of the harbour porpoise echolocation clicks. The localization system is tested in several environments including the Aquatron, Bay of Fundy, Herring Cove, and New Zealand. The Time Difference of Arrival localization algorithm is used to estimate the position of sound sources using the difference of propagation time between multiple sensors. The implementation also improves upon a traditional grid search by using a lookup table stored in a hyperoctree to reduce the execution time of a position estimation. Additionally, a method to analyze and reduce the estimation error for different sensor geometries is developed. Finally, the impact of noise is mitigated by using various pre-processing techniques.

List of Abbreviations and Symbols Used

ADCP Acoustic Doppler Current Profiler. vi, vii, 49, 50, 51

AoA Angle of Arrival. v, 3, 8, 9

ARM Advanced RISC Machine. 14, 29, 31, 32, 33, 35, 64

AWGN Additive White Gaussian Noise. 21, 23, 76

AXI Advanced eXtensible Interface. 38

CCC Classic Cross-Correlation. 9, 10, 11, 22, 23

CMC Canadian Microelectronics Corporation. 6

DAQ Data Acquisition. 33, 35

DSP Digital Signal Processing. 5, 29

DTB Device Tree Binary. vi, 38, 39

DTS Device Tree Source. vi, 37, 38, 39

ext4 Fourth EXtended Filesystem. 36, 39

FAT32 File Allocation Table (32-bit). 36

FFT Fast Fourier Transform. 9, 14, 24

FORCE Fundy Ocean Research Centre for Energy. xiii, 7, 47, 54, 56, 66

FPGA Field Programmable Gate Array. 3, 29, 36, 37, 64, 65

FSBL First-Stage BootLoader. 36, 39

GCC Generalized Cross-Correlation. 10, 22, 23

GNU GNU's Not Unix!. 36, 75

GPS Global Positioning System. v, vii, 17, 20, 21, 22, 24, 51, 52, 53

GUI Graphical User Interface. 5, 75

HDF Hardware Descriptor File. vi, 37, 38, 39

HDL Hardware Description Language. 37

HWH Hardware Handoff. 37

IEEE Institute of Electrical and Electronics Engineers. 5

IFFT Inverse Fast Fourier Transform. 9, 14, 26

IO Input / Output. xii, 38

IoUT Internet of Underwater Things. 1

IP Internet Protocol. 32, 33, 46

IPv4 Internet Protocol Version 4. 32

IPv6 Internet Protocol Version 6. 32

LCMFP Low-Complexity Matched Field Processing. xiii, 7, 8, 20, 21, 24, 64, 67, 72, 75

LOS Line-of-Sight. 3, 4, 25, 27, 76

LUT LookUp Table. 13, 14, 76

MAC Media Access Control. 32

MCCC Multi-channel Cross-Correlation Coefficient. 24, 25, 28

MFP Matched Field Processing. 2, 4

MTU Maximum Transmission Unit. 31

PAM Passive Acoustic Monitoring. 2

PCM Pulse-Code Modulation. 35

PHAT Phase Transform. 10, 11, 22, 23

PL Programmable Logic. 29, 32, 33, 35

PS Processing System. 29, 40

RISC Reduced Instruction Set Computer. ix, 14

RTC Real-Time Clock. 31

SCOT Smoothed Coherence Transform. 10, 22

SD Secure Digital. 29, 30, 33, 36

SDK Software Development Kit. 37

SNR Signal-to-Noise Ratio. v, 3, 7, 11, 21, 22, 23, 25, 26, 66, 67

SoC System-on-Chip. 3, 4, 5, 7, 24, 29, 64, 72

SPL Sound Pressure Level. 35

SSD Solid-State Drive. 46

TCL Tool Command Language. 37

TCP Transmission Control Protocol. 31, 33

TDoA Time Difference of Arrival. v, vi, 2, 3, 4, 5, 6, 7, 8, 9, 15, 16, 20, 21, 25, 27, 28, 51, 52, 59, 64, 65, 66, 67, 72

TDoMA Time Difference of Multipath Arrival. 4

Tk Toolkit. 75

TR Turbulent Research. 30, 31, 45, 46, 47

UART Universal Asynchronous Receiver / Transmitter. 30, 40

UDP User Datagram Protocol. 31, 32, 35

UIO User-Space IO. 38

USB Universal Serial Bus. 40

UTC Coordinated Universal Time. 6, 49, 54

UW-STREAM UnderWater Sensing and Transmission using Electro-Acoustic Modems.
xiii, 24

VC Visitor Centre. 7, 47, 54, 56, 66

VHDL Very High-Speed Integrated Circuit Hardware Description Language. 29, 30, 37

WAVE Waveform Audio File Format. 27, 75

ZYBO ZYnq BOard. vi, 29, 30, 31, 46

Acknowledgements

A special thanks to Fundy Ocean Research Centre for Energy (FORCE) for funding this research through the Mitacs Accelerate program.

With deep appreciation, I would like to thank my supervisor Dr. Jean-François Bousquet for his invaluable guidance, expertise, and insight throughout the completion of this thesis. Dr. Bousquet's mentorship has been instrumental in shaping my research and helping me grow as a researcher.

I would also like to extend my sincere thanks to my colleagues in the UnderWater Sensing and Transmission using Electro-Acoustic Modems (UW-STREAM) lab for creating a welcoming and engaging environment that made my research experience truly enjoyable. In particular, I would like to express my gratitude to Ehsan Malekshahi for his invaluable knowledge and deep understanding in underwater localization, and for developing the Low-Complexity Matched Field Processing algorithm used in parts of this thesis.

Lastly, I want to express my deepest appreciation to my mother for being my constant source of support and inspiration throughout my academic journey. Her encouragement and belief in me have been instrumental in my success. I am forever grateful for her consistent dedication, sacrifice, and endless support that have helped me achieve my academic goals.

Chapter 1

Introduction

This chapter provides the background and motivation for the work done on a passive harbour porpoise localization system. It describes the need for a real-time implementation of the processing and deployment constraints for different applications. First, in Section 1.1, the background of the project is provided; then, in Section 1.2, the objectives of the project and contributions are described; finally, in Section 1.3, the organization of the thesis is described.

1.1 Background

There is significant interest in increasing our access to the Earth's oceans and discovering and exploiting its resources sustainably. To monitor the environment, fleets of underwater vehicles sent on missions as well as instruments that are moored at the bottom of the ocean are mounted with heterogeneous sensors to enable the Internet of Underwater Things (IoUT) [1]. The choice of sensor array size depends on the available space, the power consumption, and the data processing capability. In this work, a remote acoustic sensor array is proposed to detect and localize marine mammals.

Detection and localization of marine mammal species using passive acoustics are of great interest for monitoring animals using low-power nodes [2]. For example, fishermen must avoid catching hector dolphins in their nets and can only detect and locate this specific species by installing hydrophone sensors near the net entrance on the trawl door. Additionally, as reported in [3], in the Bay of Fundy, it is critical to detect the presence of harbour porpoises, to prevent entanglement with the tidal turbine equipment being deployed. Also, harbour porpoises are also accidentally caught in fisherman equipment, so the system used for hector dolphins can also be adapted to work with harbour porpoises.

An alternative solution to minimize interaction between the harbour porpoise and man-made infrastructure involves installing acoustic alarms with the goal of running the equipment safely. These alarms do not estimate the location of marine life, but instead attempt to scare them away. However, these alarms may interfere with the echolocation harbour

porpoises use and may lead to them being confused. Alternatively, harbour porpoises may become accustomed to these sounds, making them ineffective.

Another alternative is to use an active sonar system to locate harbour porpoises. An active system has some advantages over a passive system, namely the known transmit time, power, and exact signature. This allows for an easier and more accurate position estimation of nearby objects or marine life. However, identifying the specific item or species can be challenging. Finally, an active system requires a dedicated, high-frequency sound source, which increases complexity and power consumption.

Passive acoustic sensor arrays work by listening for the noises/vocalizations that marine life naturally create. Harbour porpoises emit highly-directional, high-frequency clicks centred around 130 kHz [4] and each click lasts approximately 50 to 200 microseconds [5]. Harbour porpoises emit approximately 100 to a maximum of 500 of these clicks per second during hunting and prey capture, and approximately 20 per second during navigation. These click trains have a high source level, typically around 114 to 148 dB (re 1 μ Pa @ 1 m) with a maximum observed level of 180 dB (re 1 μ Pa @ 1 m) [6]. These clicks typically have a beam width of approximately 16.5 degrees for the horizontal plane (azimuth) and 16.5 degrees for the vertical plane (elevation) [7]. The narrow beam enables the harbour porpoise to have a reliable method of localizing prey. An example of a received harbour porpoise click can be found in Figure 1.1. This recording was made using a 5-sensor array in a horizontal “plus” formation on 2019-08-14.

There are a few notable passive acoustic localization algorithms. Passive Acoustic Monitoring (PAM) utilizes two or more hydrophones to localize these clicks in different dimensions by measuring Time Difference of Arrival (TDoA) between the different hydrophones [2]. In a localization system containing only two hydrophone sensors, the sensitivity of these high-frequency narrowband clicks to the cross-correlation leads to ambiguities due to spatial aliasing of the signal [8]. However, using a larger number of hydrophones in a compact array can mitigate this sensitivity and enhance the results [9].

In comparison, using Matched Field Processing (MFP), discussed for example in [10, 11], the source position is estimated in the frequency domain and uses the propagation characteristics to improve the estimation accuracy. It should also be noted that the accuracy of the localization algorithm depends on the geometry of the array and the number of sensors. As explained in [12], a minimum of 4-elements positioned in 3-orthogonal axes are required

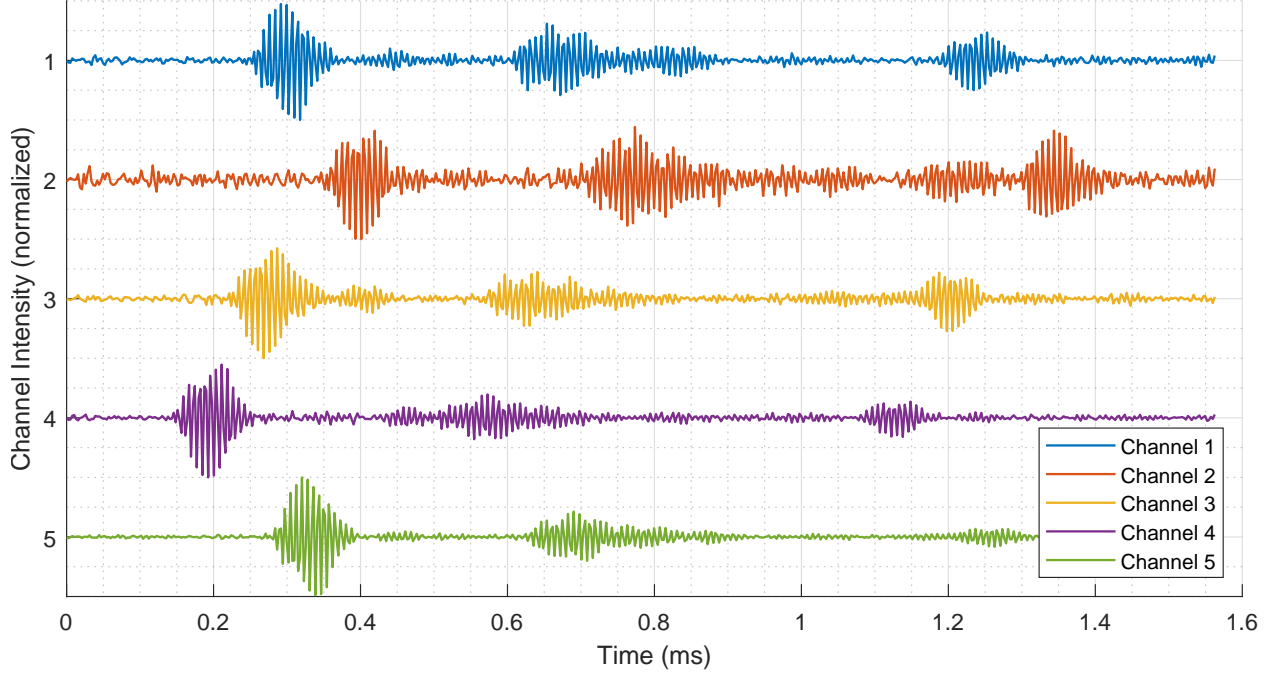


Figure 1.1: Sample of a harbour porpoise clicks received on 5 channels.

to localize in three dimensions. The tetrahedron is commonly used for this purpose [13]. As previously observed in [14, 15], the range estimation is deteriorated using compact arrays.

Acoustic localization algorithms have previously been implemented for real-time operation. In [8], a TDoA algorithm is implemented using a System-on-Chip (SoC) to obtain the horizontal Angle of Arrival (AoA) or azimuth. The system uses two hydrophones, which limits the array Signal-to-Noise Ratio (SNR) gain, as well as the ability to localize in 3D when only considering the Line-of-Sight (LOS) path. When using a linear array, it is impossible to resolve two signals that arrive at either side of the array, commonly referred to as the left-right ambiguity. Also, in [16], a TDoA algorithm is implemented for in-air applications using a dedicated microprocessor and the phase delay is calculated in the frequency domain. A similar two-microphone system has been implemented in [17] on a Field Programmable Gate Array (FPGA).

There have been some attempts to reduce the number of sensors required to make an accurate 3D position estimation. For example, in [18], a single hydrophone is used to estimate the position of sperm whales in 3D using a technique that exploits multipath in combination with a ray-tracing propagation model of the environment. However, as mentioned in [18], this approach requires sufficient knowledge of the bathymetry around the array and could be

ineffective in simple (such as flat or constant slope) bathymetries due to the reliance on the variability in azimuthally dependent bathymetry. Additionally, the technique to differentiate different path arrivals rely on the impulsive nature of the whale clicks, making it ineffective for broadband signals.

The work done in [19] removes the requirement for an impulsive signal and can estimate the location of a source with a broadband signal. This technique is referred to as Time Difference of Multipath Arrival (TDoMA) and operates by using pulse compression to measure the TDoMA quantities [19]. The improvement of using multipath in [19] was demonstrated by using TDoA to estimate two potential source azimuths, then, the range was estimated by considering multipath using a flat bottom, finally, the azimuth ambiguity is solved by using the bathymetry. While the system in [19] improves upon the traditional TDoA and removes some limitations in [18], it still requires sufficient knowledge of the bathymetry and may be ineffective in bathymetries without enough variation.

An alternative approach to improve upon a traditional TDoA system is described in [20], which uses multiple compact sensor arrays spread over a large distance. Using this approach, the knowledge of the bathymetry around the sensors is not required, since the large distance between arrays allows for an accurate 3D position estimation using just the LOS path to each of the nested sensor arrays. In [20], two compact arrays were separated with a distance of 1 km and were used to accurately estimate the position of Cuvier's beaked whales using the echolocation clicks that they emit. However, this approach has two main downsides that must be overcome before an accurate estimation can be made. First, there is a click association problem where time-synchronized data must be shared between the arrays, which is a problem that was solved in [20]. The other issue is that low-energy or highly directional sound sources may not be able to be received by both arrays, which is necessary for an accurate position estimation.

The proposed algorithm is implemented on a Xilinx Zynq-7000 SoC. Its performance is assessed in presence of noise, and the system is further deployed in the ocean to localize in real-time different signatures: pulses representative of harbour porpoises as well as shipping noise. It should be noted that in [21], a low-complexity implementation of the MFP algorithm is described to localize harbour porpoises. In comparison, in this work, the system is augmented with a TDoA algorithm, and the system geometry is optimized to estimate the range and azimuth of the source.

1.2 Research Objectives and Contributions

The overall goal of this research is to implement a real-time, passive acoustic localization system to detect and localize harbour porpoises. This is completed by developing a SoC that can perform the detection and localization in real-time. This localization is conducted by analyzing the sound received on several hydrophones (underwater microphones) to listen for harbour porpoise clicks.

This research can be broken down into four key objectives. First, a localization algorithm must be chosen by considering the benefits and drawbacks of different algorithms. Second, the sensor geometry must be analyzed and optimized to reduce the estimation error. Third, the Digital Signal Processing (DSP) for the localization algorithm must be implemented in real-time. Fourth, the system must be deployed and tested in the ocean.

The first long-term deployment of this project is completed in Parrsboro, Nova Scotia, and is described in detail in Section 4.4. The objective of this deployment is to localize harbour porpoises to determine their behaviour around underwater equipment. Analyzing the behaviour of harbour porpoises around this equipment plays a critical role in ensuring the safe operation of the said equipment.

The second long-term deployment of this project is completed in New Zealand and is described in detail in Section 4.5. In this trial, a fishing vessel with two trawl doors and a trawl net is used. In this configuration, the trawl net is attached to the two trawl doors, which are pulled along the sea floor behind the fishing vessel. Four hydrophones were mounted on each of the trawl doors and were used to detect and localize nearby marine life. The purpose of this trial is to ensure that harbour porpoises are not accidentally caught in the trawl net.

The first contribution for this work is the development of a real-time passive localization system on an embedded platform that can be deployed in the ocean to locate marine life. This involved implementing different localization algorithms, such as TDoA on an SoC running the Petalinux operating system. This SoC is also programmed to communicate directly to obtain sound measurements from an underwater platform containing a hydrophone array. These sound measurements were used in the localization system, and location estimations were used in a real-time map Graphical User Interface (GUI) that was developed to show the approximate location of nearby marine life. This work is described in Chapter 3 and was tested in Chapter 4. This work was published in the 20th Institute of Electrical and Electronics Engineers (IEEE) Interregional NEWCAS Conference in Québec City, Canada [21].

The work published in [21] was also demonstrated at the Canadian Microelectronics Corporation (CMC) Microsystems TEXPO 2022 Graduate Student Research Competition in Québec City, Canada. Before this demo occurred, harbour porpoise vocalizations were recorded in Grand Passage, Nova Scotia, on 2019-08-14 starting at 12:03:28 Coordinated Universal Time (UTC). Then in the demo, this recording was used as the input to the real-time localization system and the estimated position of harbour porpoise was shown as the recording was played. The estimated position was shown on a map. A capture of the map in the demonstration is shown in Figure 1.2 and the location of the hydrophone array is represented as a H in a red circle. Additionally, the exact range and azimuth estimation is shown on the top left, as well as the polar coordinates converted to latitude and longitude coordinates.

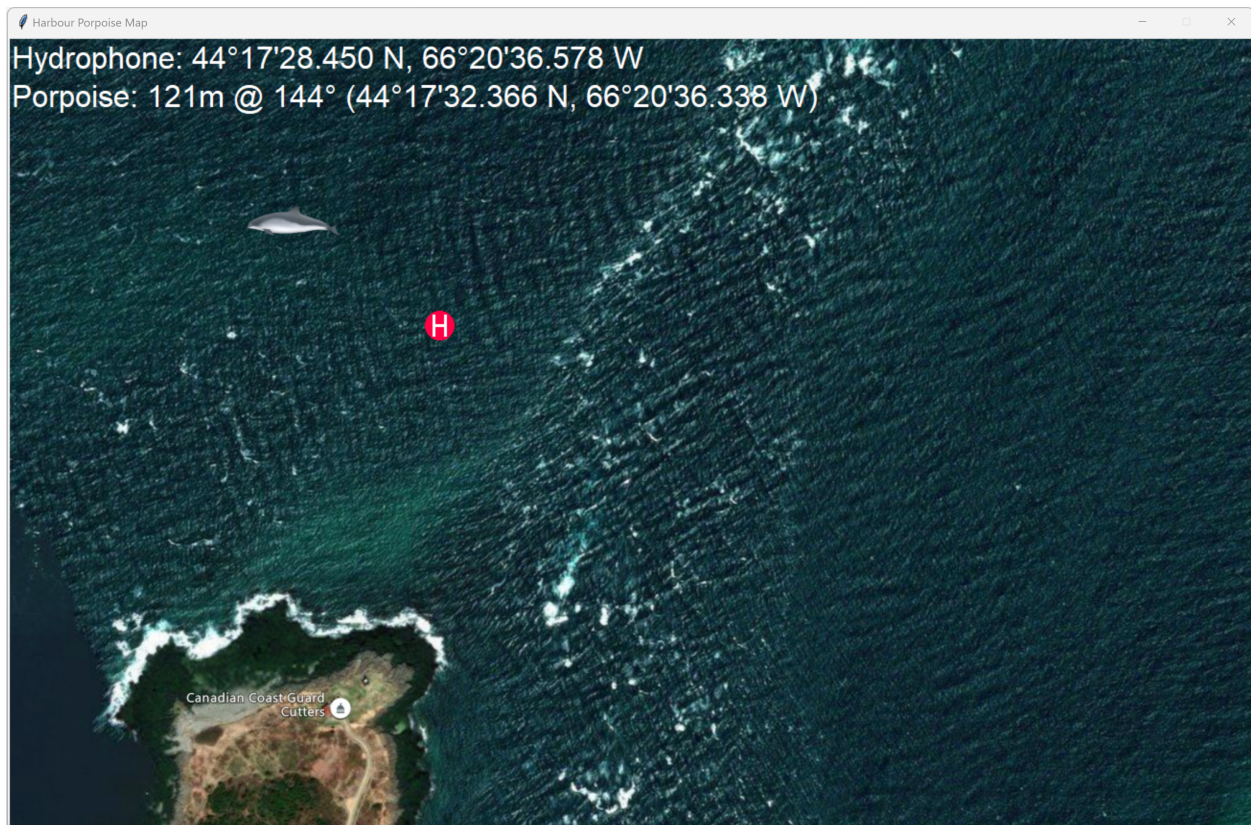


Figure 1.2: Capture of the harbour porpoise map in Grand Passage.

As a second contribution, figures of merit were defined to quantify the accuracy of different sensor geometries and the deployed sensor geometry was optimized to minimize the localization error while using the TDoA algorithm. Additional improvements were studied

to the standard TDoA algorithm to improve the performance and accuracy. First, many optimizations were created to enable this algorithm to work with a high resolution grid in real time. Additionally, different pre-processing and interpolation techniques were evaluated to mitigate the effect of noise at different SNR levels. This work is described in Chapter 2 and was tested in Chapter 4. This work was published in the 16th International Conference on Underwater Networks & Systems in Boston, United States of America [22]. Appendix A contains information on each publication completed during the development of this thesis.

As a third contribution, the performance of the localization algorithm was evaluated in three different ocean environments, as described in Chapter 4. First, the system was tested on the berm near the Fundy Ocean Research Centre for Energy (FORCE) Visitor Centre (VC), just outside Parrsboro. Second, the system was tested in Herring Cove using two different transmitters. Third, the system was tested on a trawl door off the coast of New Zealand.

1.3 Organization of Thesis

This thesis is organized as follows. In Chapter 2, localization using compact arrays is covered, including the TDoA algorithm, cross-correlation algorithms, sensor geometry evaluation, and a Low-Complexity Matched Field Processing (LCMFP) algorithm. Then, in Chapter 3, the implementation of an embedded processor is covered, including the SoC architecture, the interface to the sensor array, and the firmware architecture. Then, in Chapter 4, the experimentation at sea is covered including preliminary tests in the Dalhousie Aquatron Laboratory and sea trials in Parrsboro, Herring Cove, the Bay of Fundy, and New Zealand. Finally, Chapter 5 concludes the work and provides insight for potential future work.

Chapter 2

Localization using Compact Arrays

In this chapter, the TDoA algorithm and the procedure for mapping a set of delays between sensors to the location estimate is described. Specifically, in Section 2.1, the far field Angle of Arrival (AoA) calculation is described; in Section 2.2, the pre-processing to get accurate delays on a digital processor is described. Then, in Section 2.3, the limitations and evaluation of sensor geometries is described; then, in Section 2.4, the delay estimation of different pre-processing techniques are evaluated. Finally, in Section 2.5, a Low-Complexity Matched Field Processing localization algorithm is described and compared to the TDoA algorithm.

2.1 Time Difference of Arrival

In this section, the TDoA algorithm is described, including the calculation of the AoA.

Generally, it is not realistic to synchronize clocks under the sea surface, and TDoA is adopted. Using TDoA, the distance between the transmitter and the receiver cannot be calculated directly. The TDoA algorithm overcomes this by using signal measurements between sensor i and sensor j and measures the delay $\Delta\tau_{ij} = t_{rx,i} - t_{rx,j}$ to determine the difference in propagation distance for two sensors $\Delta d_{ij} = d_i - d_j = c\Delta\tau_{ij}$, which depends on the AoA.

The AoA α can be calculated by assuming that the sound source is sufficiently far away, leading to a plane wave wavefront on the scale of the sensor separation. The geometry of this approximation can be found in Figure 2.1. The AoA can be calculated using

$$\alpha = \cos^{-1} \left(\frac{\Delta d_{ij}}{|p_i p_j|} \right), \quad (2.1)$$

and requires the distance between the two sensors $|p_i p_j|$ and the time difference $\Delta\tau_{ij}$ between sensors i and j . Note that this angle is with respect to the vector $\overrightarrow{p_i p_j}$ connecting sensors i and j , resulting in two potential bearings $\pm\alpha$ in 2 dimensions and the surface of a cone in 3 dimensions.

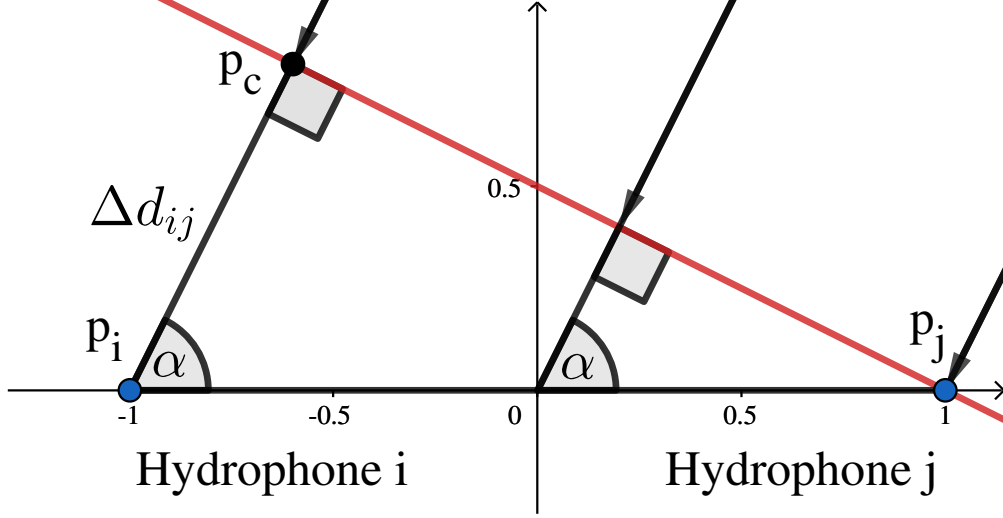


Figure 2.1: Far field approximation of TDoA AoA.

Additional sensors forming a linear array can be used to improve the reliability and accuracy of the estimation by using the average AoA estimations. Furthermore, sensors that are separated horizontally and vertically can be used to estimate the azimuth and elevation. This is achieved by determining the source position that corresponds to the measured time delays, as discussed in Section 2.2.1.

2.2 Definition of Cross-correlation Algorithms

In this section, different pre-processing techniques are described to optimize the TDoA algorithm and improve the delay estimation between sensors.

The Classic Cross-Correlation (CCC) as described in [23] is calculated in the time domain using

$$\Phi_{xy}[\tau] = \sum_{l=0}^{L-1} \overline{\mathbf{x}[l]} \mathbf{y}[l + \tau], \quad (2.2)$$

or using $\Phi_{xy}[\tau] = \mathcal{F}^{-1} \left(\overline{\mathcal{F}(\mathbf{x}[k])} \cdot \mathcal{F}(\mathbf{y}[k]) \right)$ in the frequency-domain, where $\overline{\{\cdot\}}$ denotes the complex-conjugate of its argument, $\mathcal{F}\{\cdot\}$ denotes Fast Fourier Transform of its argument, and $\mathcal{F}^{-1}\{\cdot\}$ denotes the Inverse Fast Fourier Transform of its argument.

The CCC can be used to find the $\Delta\tau_{ij}$ between sensor $i = 0$, and all other sensors. The calculated delays are compared to a modelled set of delays obtained from a grid search over

an expected set of azimuth and range values. In addition to the CCC, a band-pass frequency filter can be applied as a pre-processing step on the measured signal. This filter mitigates the impact of the noise contained in the frequencies that are not relevant to the expected sound source frequencies.

Additionally, the Hilbert transform can be used as a pre-processing step to extract the envelope of the measured periodic signal. Using the envelope at baseband improves the reliability in comparison to utilizing the passband data. Due to the nature of a periodic signal, a poor cross-correlation estimation of the passband data could have an error of a whole number of wavelengths. This issue is mitigated by using the Hilbert transform because it replaces the periodic signal with the envelope of the signal.

Furthermore, the Generalized Cross-Correlation (GCC) as described in [23] can be used to improve the accuracy of obtaining the delays between sensors over the CCC. The GCC calculation is performed in the frequency domain similar to the CCC with an additional frequency-dependent weighting function $\psi_p(f)$ applied. The GCC is calculated in the frequency domain using:

$$\Phi_{gcc,xy}[\tau] = \mathcal{F}^{-1} \left(\overline{\mathcal{F}(\mathbf{x}[k])} \cdot \mathcal{F}(\mathbf{y}[k]) \cdot \psi_p(f) \right), \quad (2.3)$$

Various candidate weighting functions $\psi_p(f)$ are described in [23] such as the CCC, Roth, Smoothed Coherence Transform (SCOT), Phase Transform (PHAT), Eckart, and the ML/HT. The simplest weighting function is used for the CCC, which weights all frequencies equally

$$\psi_p(f) = 1.$$

Next, the Roth weighting function divides by the auto-correlation of the first signal

$$\psi_p(f) = \frac{1}{\mathcal{F}(\Phi_{xx}[\tau])},$$

which is useful in suppressing frequencies where $\mathcal{F}(\Phi_{xx}[\tau])$ is the largest [23]. The SCOT weighting function applies this approach to both signals by normalizing the cross-correlation result by dividing by the geometric mean of the two auto-correlations of the two signals

$$\psi_p(f) = \frac{1}{\sqrt{\mathcal{F}(\Phi_{xx}[\tau])\mathcal{F}(\Phi_{yy}[\tau])}}.$$

which [23] describes as applying a prewhitening filter before the cross-correlation, mitigating frequency bands with a high level of noise. Next, the PHAT weighting function normalizes the cross-correlation result by dividing by the magnitude of the cross-correlation result

$$\psi_p(f) = \frac{1}{|\mathcal{F}(\Phi_{xy}[\tau])|},$$

which leads to PHAT ignoring the amplitude of the cross-correlation in the frequency domain and only considers the phase [23]. As noted in [23], the PHAT suffers in low signal power scenarios because normalizing the amplitude in the frequency domain amplifies the effect of the noise. As such, this weighting function should only be used in high SNR scenarios, and other techniques should be used in lower SNR scenarios, such as using the Hilbert transform and/or use the weighting function for the CCC. Alternatively, the Eckart weighting function could be used which is similar to the PHAT with an added multiplication with the cross power spectrum between x and y , which suppresses frequencies where $\mathcal{F}(\Phi_{xy}[\tau]) = 0$ [23].

2.2.1 3D Position Estimation

In this section, the algorithm is described to search the optimal position for a given set of delays between a reference sensor and each additional sensor.

In Section 2.1, the 3D solution of possible source locations for a given delay was approximated as a cone. If the source and receiver are relatively close to each other, the solution can not be approximated in this way. Assuming spherical spreading with an isovelocity sound speed profile with no refraction, the true solution set is the surface of a one of the sheets of a hyperboloid of two sheets. Using multiple delay measurements, the source position can be calculated by determining the intersection of multiple hyperboloids. An example of the intersection of multiple hyperboloids can be seen in Figure 2.2. Using some array geometries, such as a linear array, the intersection of these hyperboloids can be easily calculated. However, for an arbitrary, potentially dynamic geometry, this intersection may not be easily calculated. To overcome this, a numerical approach can be used in the form of a grid search.

The searching algorithm makes use of a grid search in spherical coordinates over the expected azimuth, elevation, and range to estimate the optimal source position. This search operates by picking a hypothetical source position p in the area of interest and calculates the expected delays between the reference sensor and each other sensor n : $\tau_{expected,n}(p)$ given the current sound speed and sensor geometry. These expected delays are compared with the

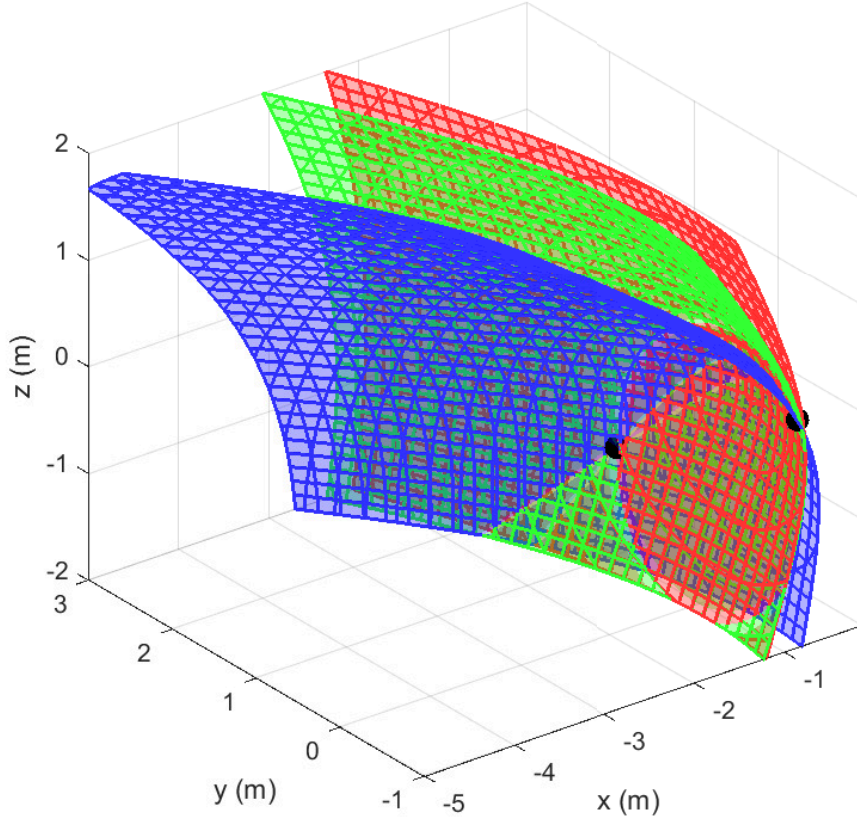


Figure 2.2: Intersection of multiple hyperboloids.

measured delays $\tau_{measured,n}(p)$ to evaluate the error $\mathbf{E}(p) = \sum_{n=1}^N |\tau_{measured,n} - \tau_{expected,n}(p)|$. The optimal source position estimation is the position with the minimum error between the measured and expected delays:

$$\hat{p}_0 = \arg \min_p \{ \mathbf{E}(p) \}. \quad (2.4)$$

One disadvantage of this grid search is that every search point must be evaluated to find the optimal source position estimation. This grid search resolution is a function of the number of azimuth ($n_{azimuth}$), elevation ($n_{elevation}$), and range (n_{range}) in the given area of interest. This leads to a time complexity for the grid search of $\mathcal{O}(n_{azimuth} * n_{elevation} * n_{range})$. This indicates that increasing the grid resolution in all three dimensions by a factor of k will lead to an increase of computation time by a factor of k^3 . Depending on the hardware and

power consumption available, this could limit real-time grid searches to lower resolutions.

Another disadvantage is caused by the quantization error involved with sampling the sensor data. Multiple nearby search points at a given resolution may lead to the same set of expected delays and in turn having the same minimum error $\mathbf{E}(p)$ at each of these points. This leads to each of these points being equally likely as being the original source position, leading to an ambiguity. In this scenario, the mean/centre of these points is used as the position estimation to minimize the average estimation error. This increases the complexity of the algorithm, as all search points must be evaluated even if a point with $\mathbf{E}(p) = 0$ is found because the location of each grid search point with the minimum error is required to minimize the average estimation error.

To optimize the time complexity of the implemented algorithm, a LookUp Table (LUT) for a given sensor geometry, sampling frequency F_s , and sound speed c can be used. This table contains an entry for each unique sets of possible delays and maps these delays to the pre-computed centre location. This greatly reduces the computation cost involved with mapping a set of delays to the position estimation. The number of entries in the table depends on which delays are physically possible, which is a function of the number of sensors N , sensor separation d_i , sound speed, and sampling frequency. The maximum delay occurs when the direction of propagation is parallel to the direction of the sensor separation, leading to a measured delay of $\tau_n = \text{round}(d_i/c)$ in number of samples. This leads to a delay range of $2 * \text{round}(d_i/c)$ and worst-case complexity $\mathcal{O}(\prod_{i=1}^N \text{round}(d_i/c))$, which assumes each sensor delay measurement is independent. In reality, this is not the case and a significant portion of these theoretical delay sets are not possible, greatly reducing the number of required entries. Instead of searching for through every point in the search grid, the LUT only needs to consider each entry in the table.

Additionally, this time complexity can be reduced further by using techniques such as spacial hashing or by storing the table in a data structure such as a k-d tree or a D-dimensional hyperoctree as described in [24]. In this work, a hyperoctree was implemented due to the simplicity of the design and the predicted improvement in the computation time. This reduces the time complexity compared to the linear LUT from $\mathcal{O}(N_{entries})$ to $\mathcal{O}(\log(N_{entries}))$.

These time complexity optimization techniques were tested using the improved geometry as discussed in Section 2.3.3 with a 3,001 by 3,001 by 210 point grid with 0.1 m horizontal spacing and 0.2 m vertical spacing (leading to a grid size of 300 m by 300 m by 42 m). With

this grid resolution, 950449711 points must be evaluated to find the best location estimation. Using a LUT, this number is reduced by a factor of approximately 3,500 to 265,200 entries in the LUT. Finally, using a hyperoctree requires considering a minimum of 24, mean of 896, and maximum of 1,759 entries in the LUT. On average, this is a 296 times reduction compared to the LUT and a 1,060,770 times reduction compared to the base grid search.

The final optimization was done to cross-correlation used to measure the time delay between sensors. In the implemented system, this calculation is performed in the frequency domain, which requires calculating the FFT and then the IFFT of the signals. During the calculation of the FFT each sample is multiplied by $e^{-i2\pi m/N}$, which is implemented by computing the sine and cosine of $2\pi n/N$ where m is the time domain sample index and N is the window size. After choosing the maximum window size allowed, these values of sine and cosine can be computed once and stored in a LUT to allow for faster execution of the FFT. These tables are also used in the IFFT where a similar computation occurs. With the combination of these optimizations, the time taken to compute the cross-correlation of two signals with 1024 samples on the Advanced RISC Machine (ARM) processor is reduced from 12.630 ms to 1.828 ms, which allows for more frequent localization estimations to occur in real-time.

2.3 Sensor Geometry Evaluation

In this section, the impact of the sensor geometry is described. Specifically, in Section 2.3.1, the limitations of different sensor geometries are described; in Section 2.3.2, an approach is described to evaluate a sensor configuration based on the expected range and azimuth error. Then, in Section 2.3.3, this evaluation is used to improve an existing sensor geometry with several constraints.

2.3.1 Estimation Limitations

The geometry of the sensor array greatly impacts the precision of the localization estimation. In general, a greater spacing between sensors improves the precision of the final estimation. This is due to the fact that increasing the separation raises the signal delay between sensors, allowing for a higher resolution in delay between sensors measured in number of samples at a given sampling frequency. The relationship between sensor separation and delay range is linear.

Additionally, the number of spatial dimensions an array spans greatly impacts which dimensions can be resolved and which dimensions lead to ambiguities. For example, even in an ideal scenario, a single point sensor spanning 0 dimensions cannot resolve the location of the source in any spacial dimension. Whereas a 1D linear array aligned on the X axis can resolve the X coordinate of the source, but is unable to estimate the Y or Z coordinate. This is because a source anywhere along a circle on the YZ plane at a given X coordinate leads to the same delays measured between each sensor. In comparison, a 2D planar array in the XY plane provides an estimation of the true location on one side of the plane at $Z = z$, as well as the mirror image at $Z = -z$, leading to an ambiguity. Finally, an array spanning all three dimensions in an ideal scenario can estimate the position of the source in the three dimensions.

For TDoA, simulations are run to assess the impact of the geometry on the estimation accuracy. Specifically, the estimation algorithm is run for each position on a grid, and the mean error is evaluated. In a first instance, a 4-element tetrahedral structure is modelled, and the impact of the tetrahedral edge length on the accuracy is evaluated. In a second instance, the accuracy is assessed using an array geometry constrained on the mounting structure that is provided for the deployment in the Bay of Fundy.

2.3.2 Simulation

In Figure 2.3, the mean azimuth and inclination error is shown for a tetrahedron structure as a function of vertical separation for the top element. As can be observed, both the azimuth and inclination errors are accurate, on average, to within a degree for a vertical separation below 5 cm.

Next, in Figure 2.4, the mean horizontal range and depth error is shown for a tetrahedral structure as a function of the tetrahedral edge length. Note that the relative distance is reported. As can be observed, the range error is above 20% for a vertical separation lower than one-meter. This vertical distance is often unreasonable on small platforms deployed in the ocean, particularly since the current can put a mechanical strain on the array.

For localization of harbour porpoises in the Bay of Fundy, the array is mounted on a platform that is installed at the bottom of the Minas Basin. Nominally, the depth is approximately 30 meters, and the position of each element is shown in Figure 2.5 and is described in Table 2.1.

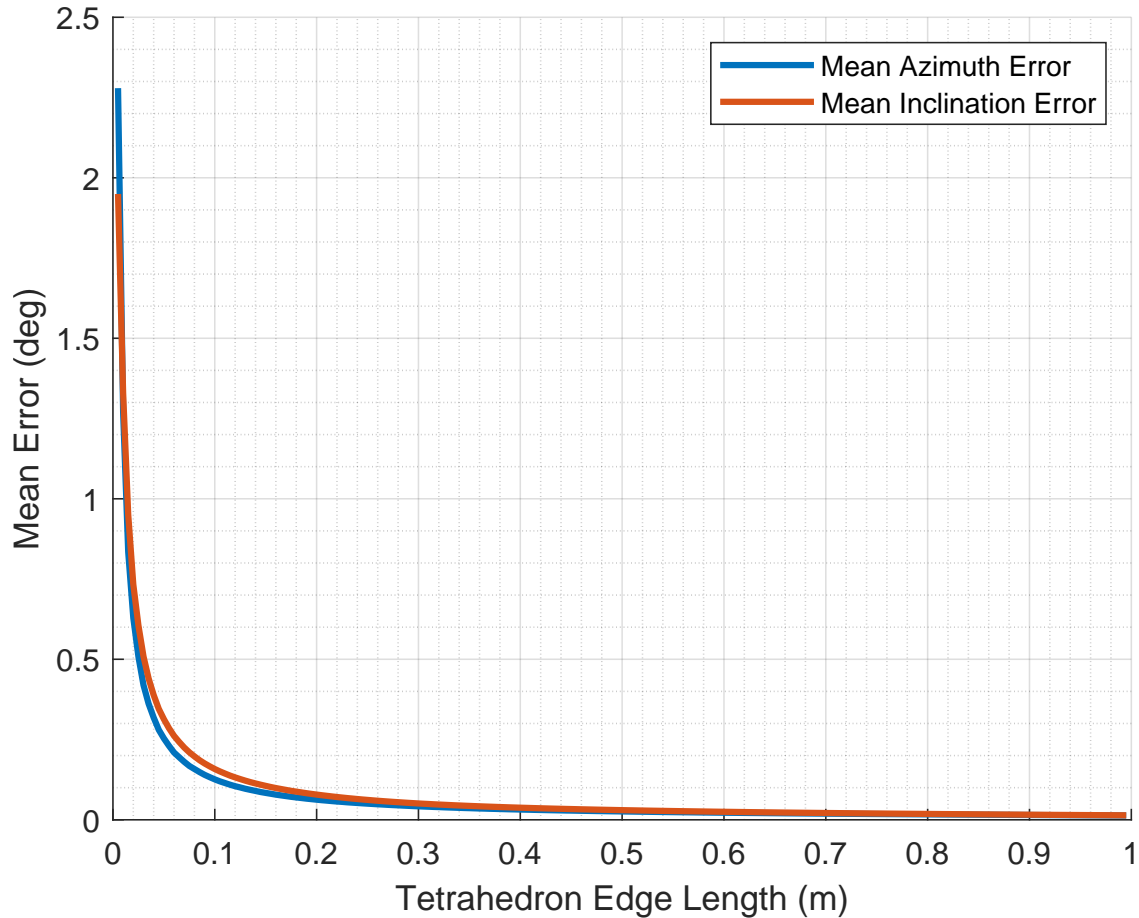


Figure 2.3: Azimuth and inclination error as a function of height using a tetrahedral structure using the TDoA algorithm.

The simulation results for this structure are shown in Figure 2.6, as a function of the vertical separation of Sensor #2. As can be shown, the inclination and azimuth mean error are both below one degree for a vertical separation above 0.1 meter.

2.3.3 Improvement

On 2021-11-02, a trial was performed in Herring Cove, Nova Scotia, Canada. In this trial, data was captured using a 4-sensor, 2D vertical planar array. In this array, the average horizontal and vertical separation between sensors were 0.2804 m and 0.1290 m, respectively. The position of each element is shown in Figure 2.7 and is described in Table 2.2. A spectrogram detailing the combination of the observed artificial sound sources can be seen in Figure 2.8. The boat sonar is the sound source analyzed in this section to obtain an azimuth estimation. The analysis of the icTalk clicks can be found in Section 4.3. Note that the

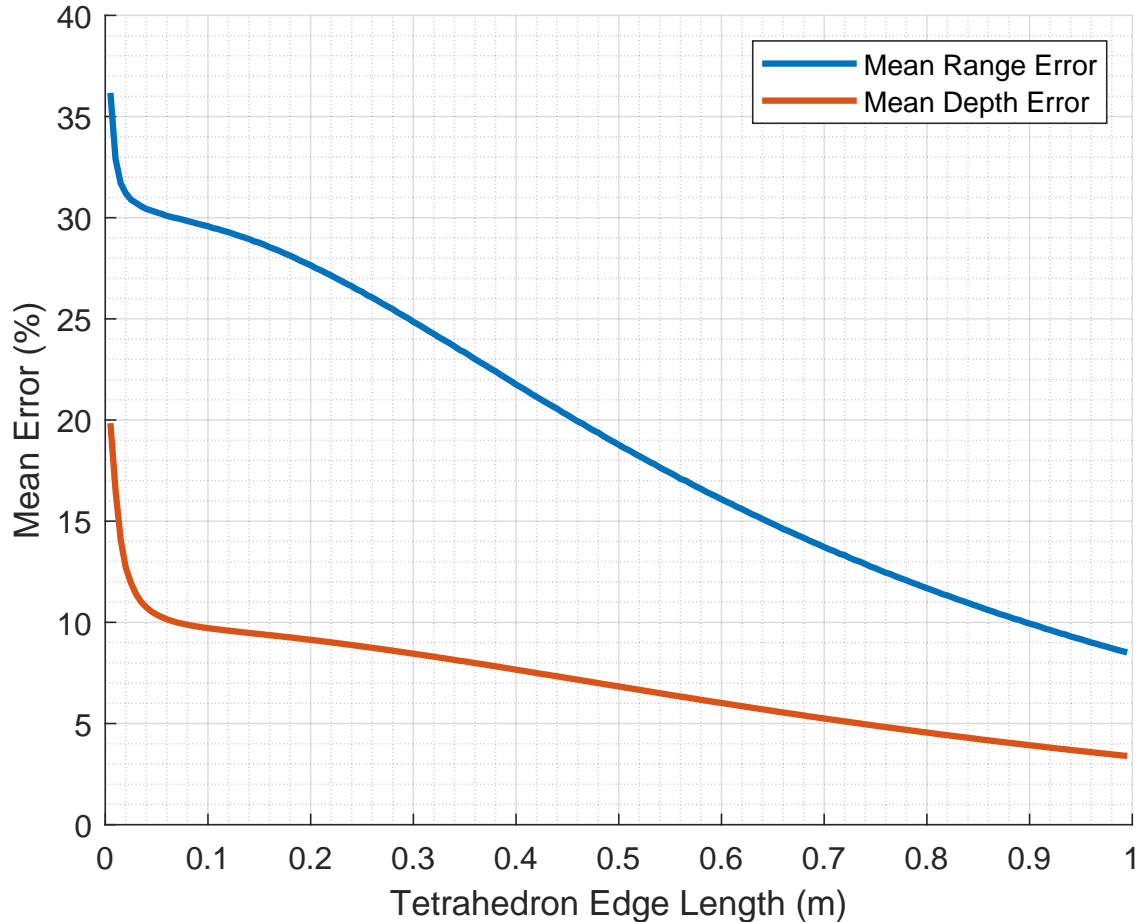


Figure 2.4: Range and depth error as a function of height using a tetrahedron structure.

power of the icTalk is much lower than the power of the boat sonar, making it difficult to automatically detect.

The estimated and true azimuth obtained using Global Positioning System (GPS) measurements throughout the trial can be seen in Figure 2.9. Due to the planar nature of the array, an ambiguity exists in the azimuth estimation. Eliminating the effect of this ambiguity by normalizing all sources positions to one side of the sensor plane lead to an average azimuth estimation error of 15.98° . The estimated and true azimuth obtained using GPS measurements throughout the trial can be seen in Figure 2.9. However, due to the relatively low vertical separation between sensors, the elevation angle was unable to be estimated correctly. This in turn impacted the range estimation, resulting in a poor average range error of 39.93%.

The geometry used in the Herring Cove trial was evaluated using a grid search with a 101 by 101 by 10 grid with 5 m spacing (leading to a grid size of 500 m by 500 m by 50 m).

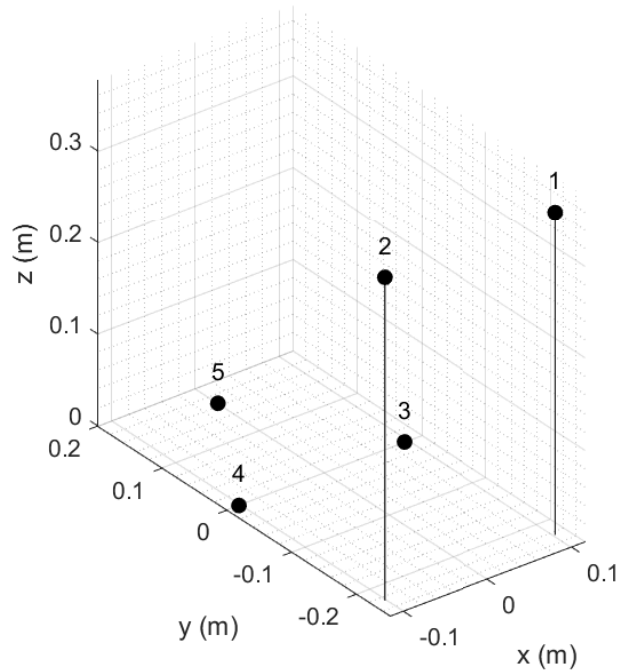


Figure 2.5: Sensor positions on the remote platform in the Bay of Fundy.

Table 2.1: Sensor positions on the remote platform in the Bay of Fundy

| N | X (m) | Y (m) | Z (m) |
|---|---------|---------|--------|
| 1 | 0.1016 | -0.2261 | 0.3520 |
| 2 | -0.1016 | -0.2261 | 0.3520 |
| 3 | 0.0953 | 0.0000 | 0.0000 |
| 4 | -0.1016 | 0.0000 | 0.0000 |
| 5 | 0.00 | 0.1651 | 0.0000 |

The existing geometry had a unique locations percentage of 19.37% (19757/102010) and a maximum repeats of 129. This indicates that for a given set of delays, there is an average of approximately 5.1 input grid search points that correspond to this set of given delays. This specific grid search was used in all simulated modifications to the sensor geometry.

The geometry from the Herring Cove trial was modified to have a better estimation accuracy and reliability based on the two metrics of unique locations and maximum repeats. Different modifications to the geometry were evaluated using these metrics, and the best possible geometry will be used in future trials. The modification to the geometry increased the unique location percentage to 42.88% (43737/102010) and reduced the maximum number

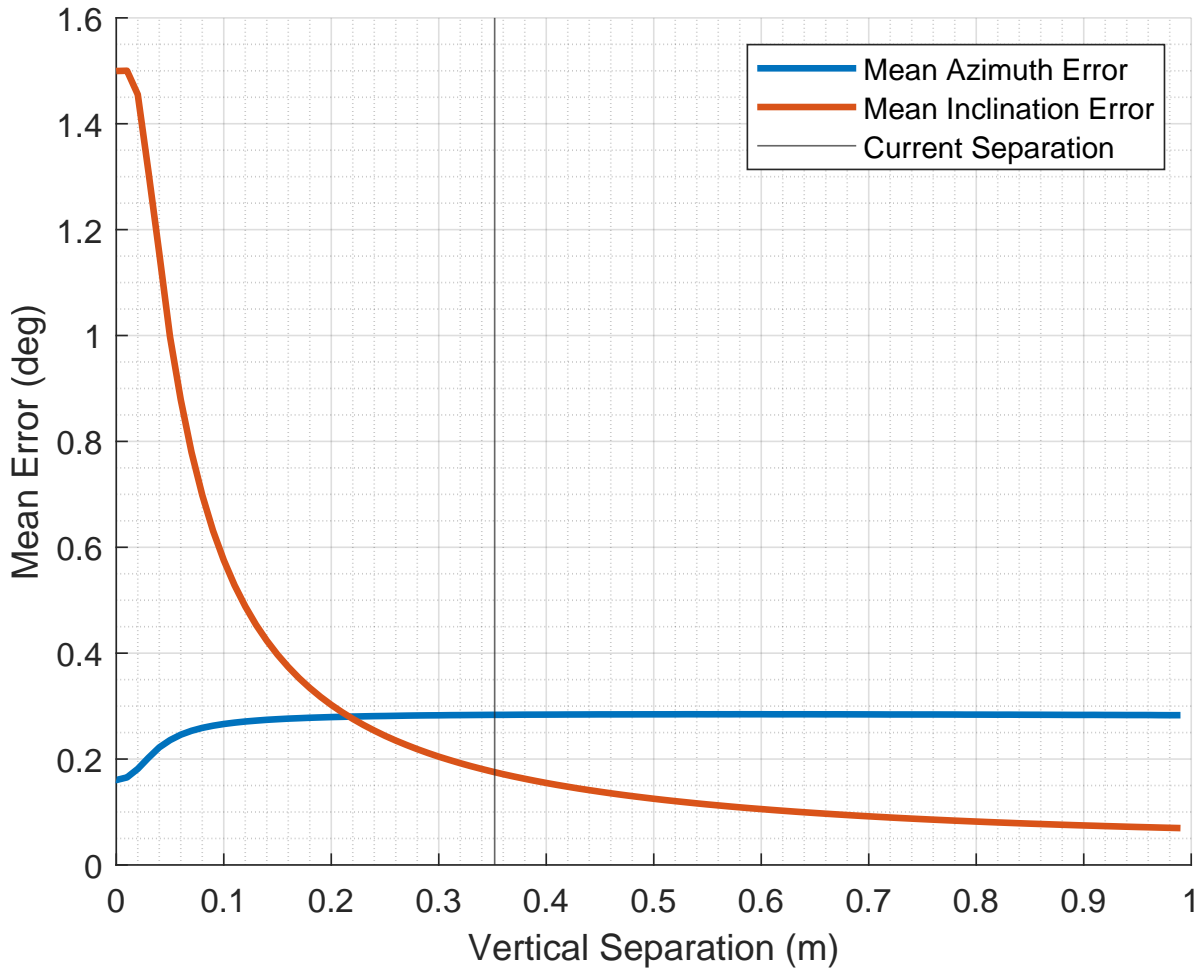


Figure 2.6: Mean inclination and azimuth error as a function of height for the structure deployed in the Bay of Fundy. The grey line shows the physical height of the sensor.

of repeats to 73. This indicates that for a given set of delays, there is an average of approximately 2.3 input grid search points that correspond to this set of given delays. This geometry is constrained to the trawl door shown in Figure 2.10. In this figure, the hydrophone sensors are positioned along the top, top right, and the right edge of the trawl door. This trawl door is one of the two doors towed along the seabed, holding a fishing net along the seabed.

In the Summer of 2021, the platform shown in Figure 4.5 was deployed near the shore in the Bay of Fundy. A test was run during which different pulses were transmitted from a research vessel: a 147 kHz signal generated by the vessel, as well as a 27-kHz signal generated using an acoustic transmitter described in [25]. The sonar pulse can be representative of a series of harbour porpoise clicks, while the 27-kHz signal is a signature that can resemble that of the dolphin's. The pulses were generated while the vessel was drifting. The acoustic

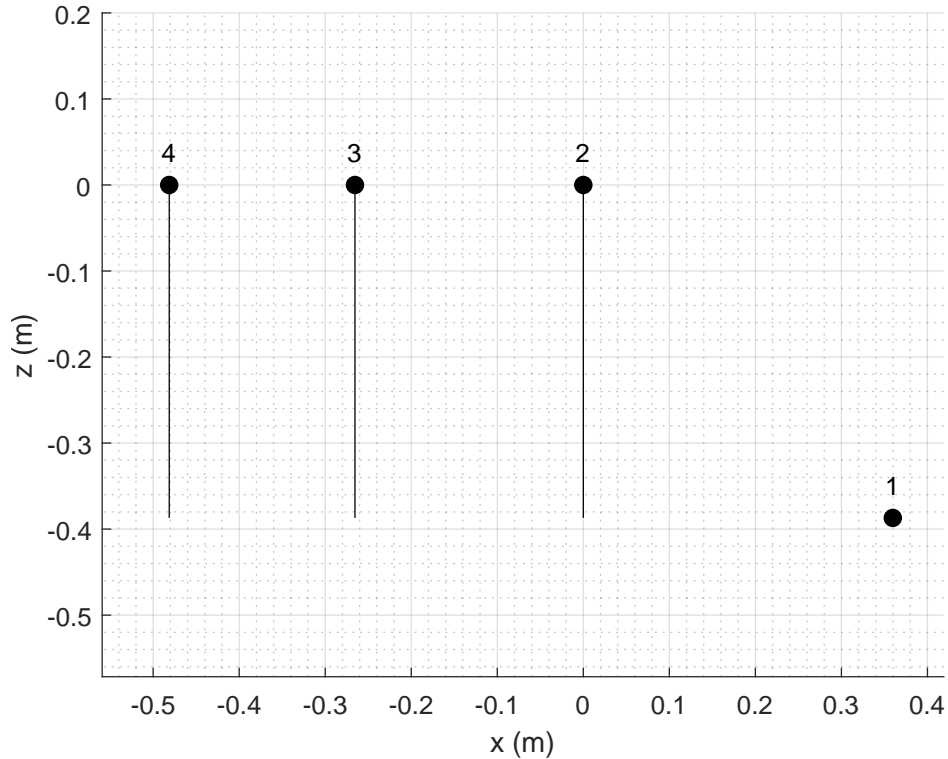


Figure 2.7: Sensor positions of the deployed planar array in Herring Cove.

Table 2.2: Sensor positions of the deployed planar array in Herring Cove

| N | X (m) | Y (m) | Z (m) |
|---|---------|--------|---------|
| 1 | 0.3597 | 0.0000 | -0.3870 |
| 2 | 0.0000 | 0.0000 | 0.0000 |
| 3 | -0.2654 | 0.0000 | 0.0000 |
| 4 | -0.4813 | 0.0000 | 0.0000 |

data using four elements was recorded, and the GPS coordinates of the vessel were recorded during the experiments. It should be noted that these tests were run in very shallow water conditions, below 15 meters. Also, the area these trials were performed in is subject to extremely high currents.

The TDoA and LCMFP localization algorithms were tested on these signatures while the boat was drifting. Unfortunately, it was not possible to obtain a reliable localization using the 147-kHz sonar pulse. This is attributed to the fact that the signal attenuates quickly at these frequencies, and was buried in ambient noise. Moreover, the signal is subject to reverberation from the metallic walls surrounding the sensors. In the future, the plates will be supported using rods, rather than full metallic walls.

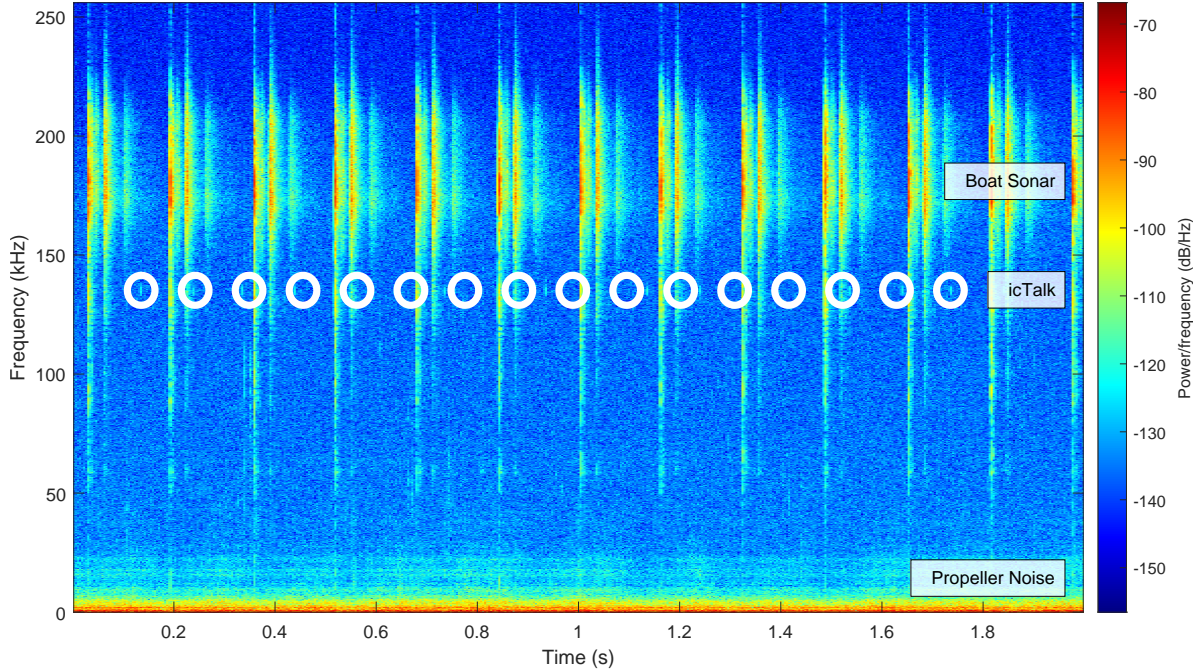


Figure 2.8: Representative sample of the spectrum during the Herring Cove Trial deployment.

For the 27-kHz clicks, the azimuth estimation using the TDoA algorithm is shown in Figure 2.11. These estimation points were automatically filtered. Nonetheless, a few estimation points are valid and follow the GPS coordinates. While the LCMFP algorithm was also run to estimate the azimuth, it was completely unreliable. This is attributed to the fact that the LCMFP algorithm can be sensitive to the environmental conditions, particularly in shallow environments.

2.4 Delay Estimation Accuracy in the Presence of Noise

In this section, the effect of Additive White Gaussian Noise (AWGN) on the delay estimation accuracy is evaluated using different pre-processing techniques as a function of SNR; then in Section 2.4.1, linear interpolation is used as a pre-processing technique to mitigate the effect of noise.

To evaluate the effect of noise on the delay estimation error, AWGN was added to simulated porpoise click signals with varying SNR. The porpoise click signal was generated using a 130 kHz signal with an envelope of half a period of a 5000 Hz signal. This envelope was chosen to have click duration of 0.1 ms and to prevent having sharp transitions before and after the click.

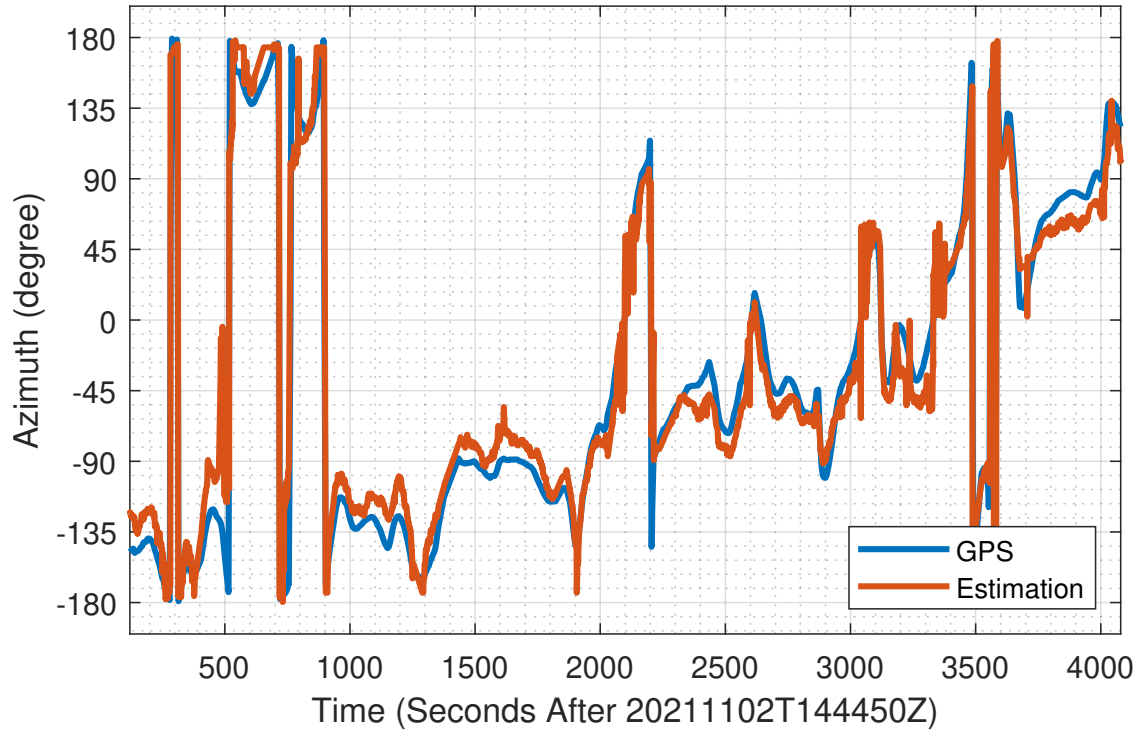


Figure 2.9: Azimuth estimation of the vessel in the Herring Cove deployment compared with the true azimuth obtained using the GPS.

The presence of noise has a significant impact on obtaining the correct delay between sensors. The level of impact depends on the SNR and which pre-processing technique is used. For example, below an SNR of -15 dB, all techniques lead to a similar mean delay error of 10 pulse widths, effectively being unable to estimate the delay between sensors. Between an SNR of -15 dB and 40 dB, the CCC + Hilbert and phase transform both outperform the CCC. The CCC reaches a minimum error of 5.44% of the pulse width at an SNR of 5+ dB, with the Hilbert transform it is greatly reduced to 0.5% at 20+ dB SNR. The GCC PHAT has a minimum error of 0.53% at 40+ dB SNR, and the GCC PHAT + Hilbert 0.52% at 40+ dB SNR. This indicates that the CCC + Hilbert transform is the best-performing pre-processing technique except in extremely high (30+ dB) SNR scenarios where CCC + Hilbert, GCC PHAT, and GCC PHAT + Hilbert are effectively tied. This can be seen in Figure 2.12. Note that GCC SCOT was evaluated and had identical performance to GCC PHAT in this simulation.

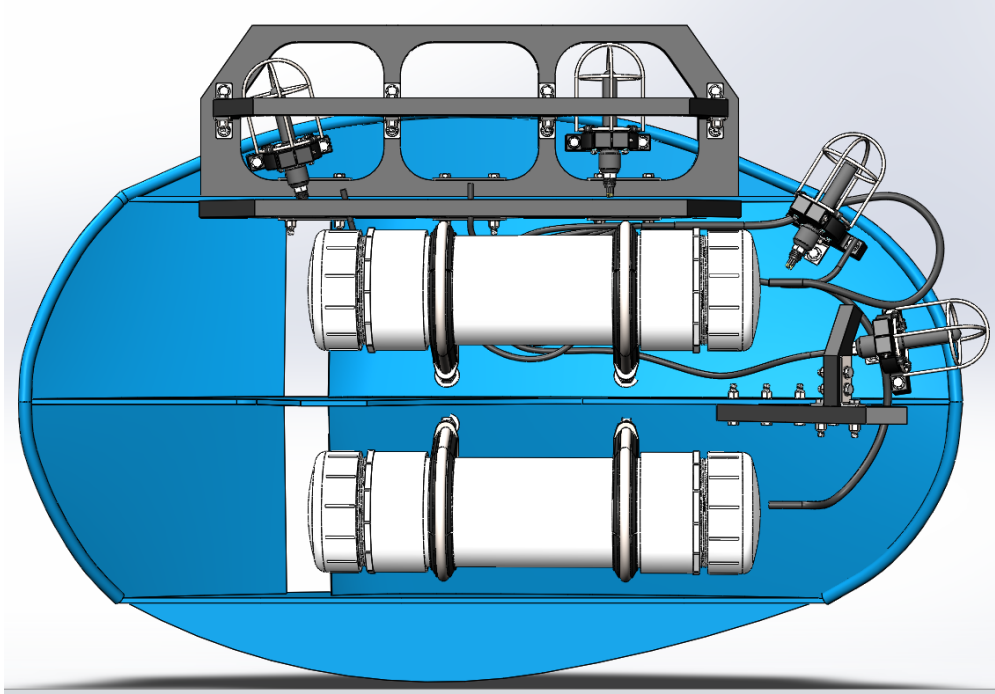


Figure 2.10: Sensor geometry constrained to the trawl door, adapted from the Herring Cove geometry and improved to minimize the localization error.

2.4.1 Accuracy After Interpolation

In this section, the use of interpolation to mitigate the effect of AWGN noise on the delay estimation is described.

To further reduce the delay estimation error in high SNR situations, the measured signal can be interpolated to increase the precision of the delay estimation. For SNR greater than approximately 0 dB, interpolation can change which pre-processing technique has the lowest error. The mean delay error vs SNR results with an interpolation ratio of 64x can be seen in Figure 2.13. Note that at this interpolation ratio, the CCC has the lowest error, reaching a minimum of 0.01% at 30 dB SNR. Additionally, the CCC + Hilbert technique has a lower error with interpolation vs without interpolation at SNR greater than 24 dB and quickly approaches the CCC mean delay error at high (30+ dB) SNR. Finally, the GCC PHAT is unable to estimate the delay with any meaningful accuracy (mean error consistently at 20 pulse widths), while the GCC PHAT + Hilbert technique yield much better results. This indicates that the Hilbert transform is required to have an accurate delay estimation while using the GCC PHAT technique and an interpolation ratio of 64x.

Although, interpolating the measured signals also comes with some downsides. Notably,

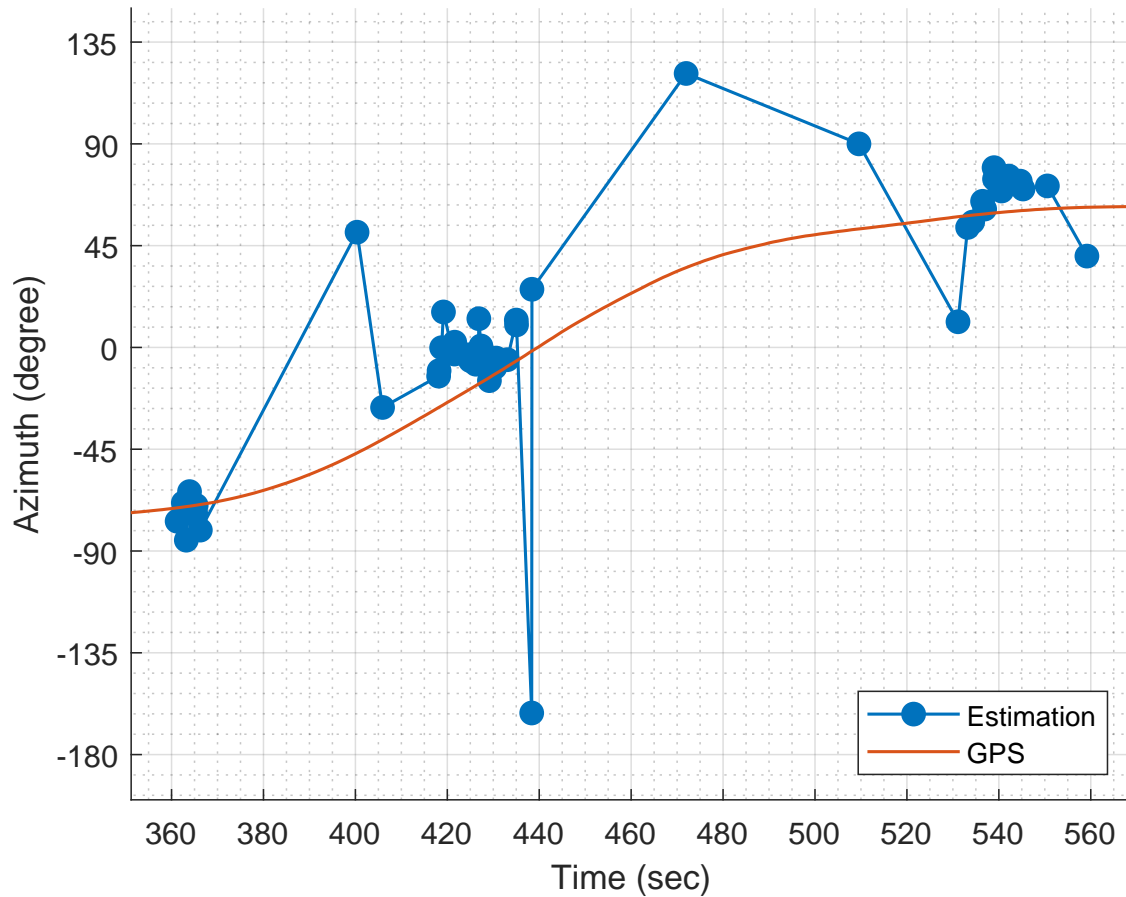


Figure 2.11: Azimuth estimation of the vessel in the berm test compared with the true azimuth obtained using the GPS.

interpolation increases the required computational resources and time required to estimate the time delay. This increase is caused by requiring a larger buffer to store the interpolated data, as well as requiring a larger FFT window size. Determining an interpolation ratio requires making a compromise between the estimation accuracy and computation time.

2.5 Low-complexity Matched Field Processing Algorithm

In this section, a LCMFP algorithm implemented using Multi-channel Cross-Correlation Coefficient (MCCC) is described. This algorithm was developed in the UW-STREAM lab and is described in-depth in [26]. In this chapter, the algorithm is summarized using a simulation.

The real-time algorithm that is implemented in the SoC is based on a MCCC localization algorithm [27] previously developed to locate targets using shipping of opportunity in [26].

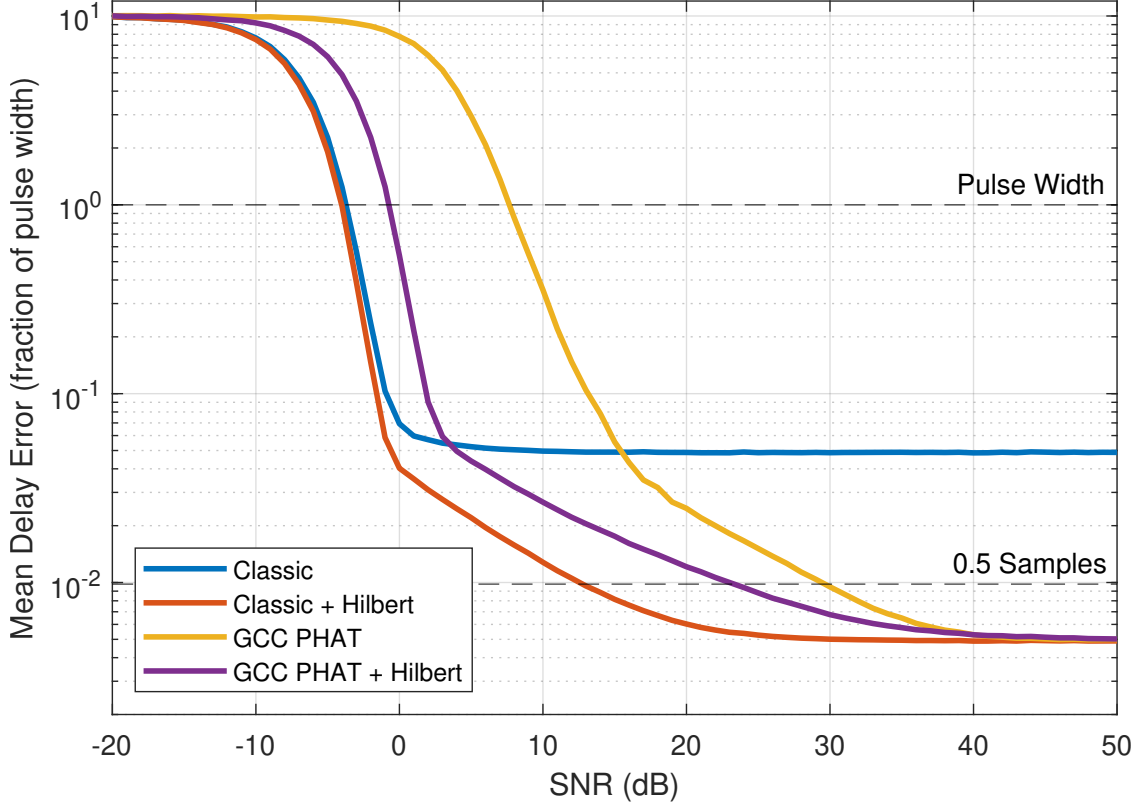


Figure 2.12: Mean delay estimation error of different pre-processing techniques as a function of SNR.

By taking into account the LOS path, as well as a second path that reflects from the surface, as will be shown the range estimation is improved compared to the standard TDoA algorithm. MCCC is a time domain approach to estimate the position $p_0 = (x_0, y_0, z_0)$ of a source for a signal $s(t)$ given a received signal $r_n(t)$ at the n^{th} hydrophone located at $p_n = (x_n, y_n, z_n)$, $n \in \mathbb{N}$. For a hydrophone array with N elements in a noiseless media, the received signal at the n^{th} element at time t is calculated as

The measured signals are modelled by $r_n(t_k)$, where $n = 1, 2, \dots, N$ at discrete time $t_k = t_{\min} + k \Delta t$, where Δt is the sampling time and $k = 0, 1, \dots, L - 1$, and the transmitted signal is modelled by $s(t_k, p_0)$, where p_0 is the source position.

To reduce the signal processing complexity, the processing is realized in the frequency domain: the transmit signal is defined as $S(\omega) = R_n(\omega)/P(p_0, p_n, \omega)$, where $P(p_0, p_n, \omega)$ is the channel gain as a function of frequency between the source position p_0 , and the sensor at position p_n . Note that $R_n(\omega)$ is the frequency response of the measured time signal $r_n(t_k)$. The subset of discrete frequencies in the band of interest can be pre-set to mitigate

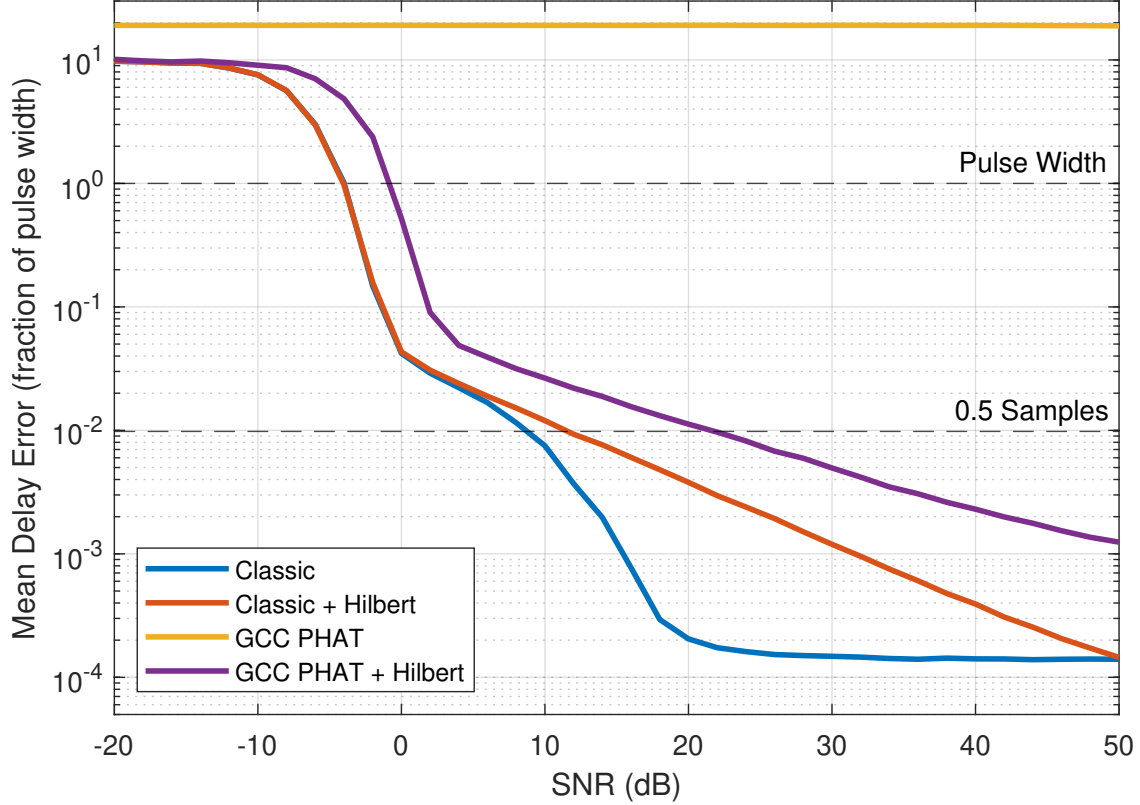


Figure 2.13: Mean delay estimation error of different pre-processing techniques using 64x linear interpolation as a function of SNR.

out of band noise, and reduce complexity. After the signal is returned to the time domain, the cross-correlation matrix corresponding to a hypothetical source at the search point p is assessed on a window of $L = 2048$ samples using

$$\mathbf{C}_c(p) = \frac{1}{L} \sum_{k=0}^{L-1} (\bar{\mathbf{s}}^\top(t_k, p) \bar{\mathbf{s}}(t_k, p)), \quad (2.5)$$

where $\bar{\mathbf{s}}(t_k, p)$ is a row vector which elements are the IFFT of $S(\omega)$ at time t_k correspond to each sensor. Finally, the source location is estimated in (2.6) using a search as described in [28].

$$\hat{p}_0 = \arg \min_p \left\{ \det [\mathbf{C}_c(p)] \right\}. \quad (2.6)$$

$$r_n(t) = \alpha_0(p_0, p_n) \cdot s(t - \tau_{0,n}(p_0)) + \alpha_1(p_0, p_n) \cdot s(t - \tau_{1,n}(p_0)), \quad (2.7)$$

where $\alpha_0(p_0, p_n)$ is the gain of the LOS path, while $\alpha_1(p_0, p_n)$ is the gain of the path reflected from the surface. Similarly, $\tau_{0,n}$ and $\tau_{1,n}$ are the delay for the LOS path and for the path from the surface reflection, respectively. The delays can be obtained by calculating the total path length of each path and by dividing by the speed of sound c .

To quantify the difference between the proposed algorithm and the TDoA algorithm, a simulation was performed on a 4-element planar sensor array. This array is similar to a 5-element plus/star + formation, except one of the exterior sensors was removed. This simulation involved generating a Waveform Audio File Format (WAVE) file with 100 clicks by sweeping the azimuth from -180 degrees to $+180$ degrees and the range from 120 m to 60.5 m. Sweeping the azimuth and range creates a spiral pattern, which can be seen in Figure 2.14.

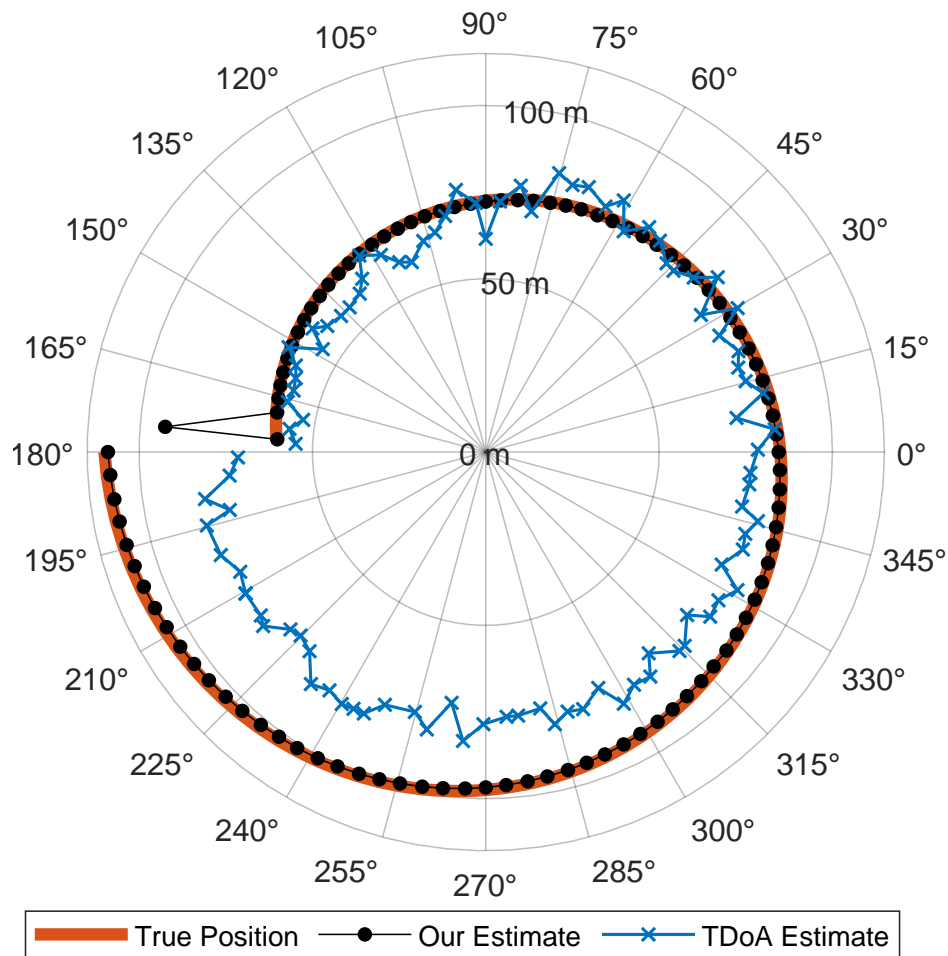


Figure 2.14: Comparison of the position estimation obtained using the proposed and TDoA localization algorithms on data from the spiral simulation.

In this simulation, using either the proposed algorithm MCCC or the TDoA results in an accurate azimuth estimation with a mean error <1 degree. However, the TDoA algorithm was unable to accurately determine the range of the simulated source, while the proposed algorithm was able to find the range. This is likely due to the lack of vertical sensor separation and limited resolution available to the TDoA algorithm. Due to the nature of obtaining the signal delay by using the cross-correlation, the resolution of the delay is limited to a whole number of samples. In comparison, the proposed algorithm is able to consider the phase of the signal and is not limited in the same way.

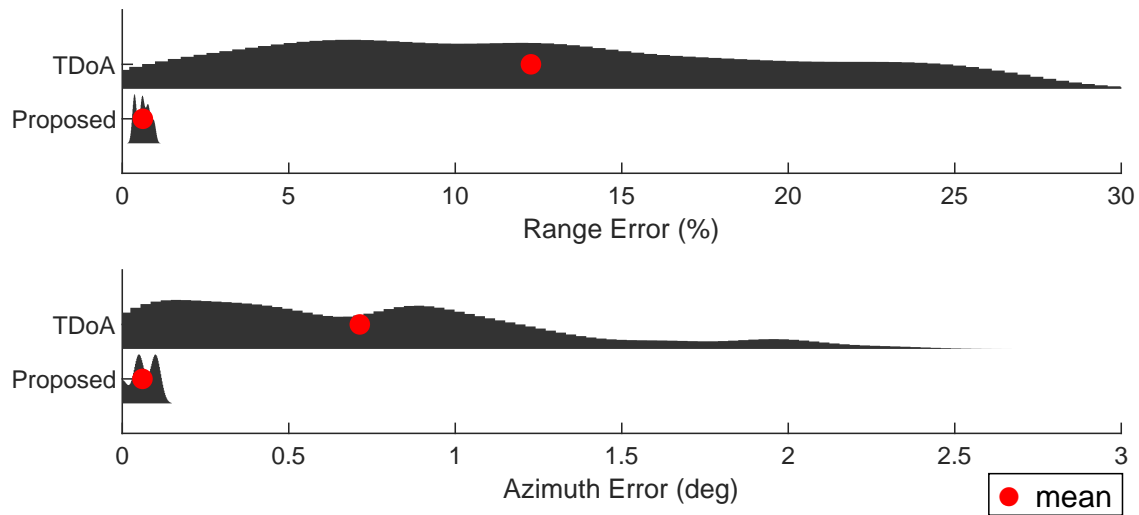


Figure 2.15: Comparison of the range and azimuth estimation error distributions between the proposed and TDoA localization algorithms on data from the spiral simulation.

The error distribution for the azimuth and range estimation is summarized in Figure 2.15, with the mean noted in red. As can be observed, the range and azimuth error variance of the proposed algorithm is much smaller than for the standard TDoA algorithm.

Chapter 3

Implementation on Embedded Processor

The objective of this chapter is to describe the real-time system architecture for the localization algorithm. The firmware architecture including the software controller will be discussed in Section 3.1; then, in Section 3.2, the real-time interface to the sensor will be described; finally, in Section 3.3, the use of an operating system on the embedded processor is justified.

3.1 Proposed System on Chip Architecture

The remote sensor is programmed on the ZYNq BOard (ZYBO) Z-7010 Development Board, which is a specialized DSP platform. At the core, the Zynq-7000 SoC contains a dual-core ARM Cortex-A9 processor as the Processing System (PS) and Programmable Logic (PL) equivalent to the Artix-7 FPGA. The PL serves to compute real-time array processing in an ultrasonic band up to 150 kHz. The combination of the PS and PL allows for a system capable of operating in real-time that can also be reconfigured through software.

The SoC is configured such that the ARM Cortex-A9 processor runs Petalinux as the operating system, which is a modified version of the Linux kernel that Xilinx maintains for embedded devices. Petalinux is used to run user code on the processor, and serves as the interface between the processor, the peripherals programmed in Very High-Speed Integrated Circuit Hardware Description Language (VHDL), and a host control computer. Using Petalinux allows for easier development by providing access to the various tools and libraries included in the Linux kernel. This means it is possible to compile and run an executable that was originally written for a Linux desktop computer on the ZYBO. The included standard libraries allow for easy access to files systems (such as on a Secure Digital (SD) card), networking, and creating and managing multiple processes. Additionally, Linux tools for compiling code and creating and running scripts are also available.

The different peripherals that are controlled by the Zynq processor to enable the remote sensor are shown in Figure 3.1. The Ethernet driver is used to share high throughput information between the processor running the real-time localization algorithm with the

acoustic sensor array; the microSD Card driver is used to record windows containing recent harbour porpoise signatures; the Zynq processor interfaces with a peripheral programmed in VHDL that runs a marine mammal detection and localization algorithm; finally, a Universal Asynchronous Receiver / Transmitter (UART) interface between the localization system and a host computer serves to send information to be displayed.

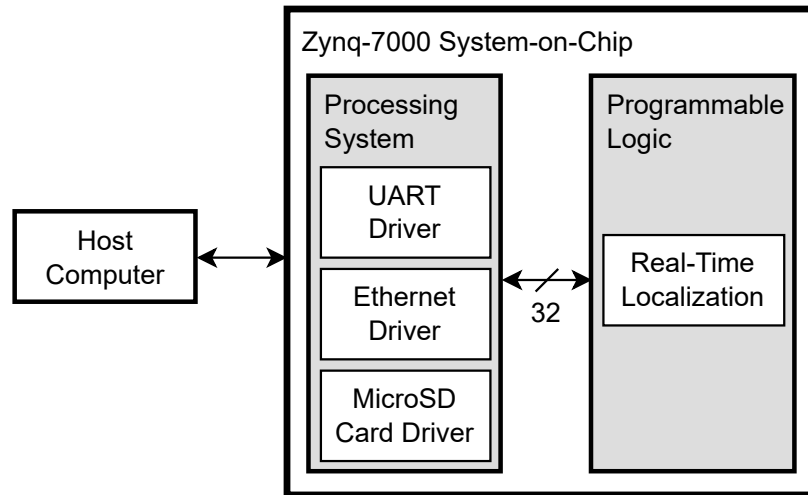


Figure 3.1: Firmware architecture for the real-time remote sensor.

For this remote sensor, the real-time localization processor communicates with a 5-element acoustic recorder manufactured by Turbulent Research (TR) that is configured to create sample 16-bit samples at 384 kHz. Through the Ethernet interface, a network is configured to maintain a link between the Zynq Development Board and the pressure case. This network configuration is shown in Figure 3.2. As shown, the ZYBO is connected directly to the pressure case using the Ethernet interface. The communication interface defined by TR allows data transfer from the recorder, as well as reconfiguration of the recorder. It does this with three different protocols. The first, TracFind, is used to locate devices on the network. The second, TracIO, is used to control the device and receive important device information. The third protocol, TracStream is used exclusively to receive the raw sound data streamed from the pressure case. These protocols are described in more detail in Section 3.2.

The main program running on the Zynq controls the connection to the pressure case and initiates the detection and localization algorithm. The state transition diagram for the localization program can be seen in Figure 3.3. One of the main constraints for this design is the limitations imposed by the maximum transfer rate while data streaming. At a sample

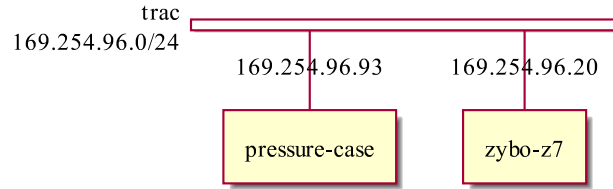


Figure 3.2: Network diagram of the network containing the ZYBO Z7 and the pressure case.

rate of 384 kHz, it is only possible to stream on 2 channels from the pressure case. This is not enough data to determine the location of the porpoise, but it is enough to determine if a porpoise is making a sound nearby. After a porpoise sound event is detected, the data streaming stops, and the file recording begins. This file can contain many more channels (up to five), allowing for localization of the porpoise. This cycle continues forever.

3.2 Interface to the Sensor Array

The interface to the TR sensor array is facilitated using the Trac suite of protocols: TracFind, TracIO, and TracStream. These protocols are used to find and configure other Trac devices on the network, stream, and record channel measurements using an array of hydrophones. The protocol operates by using the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) transport layers on top of the Internet Protocol internet layer. TracFind and TracStream both embed application messages inside UDP, while TracIO uses TCP. Multiple application messages can be included in a single transport layer packet. Additionally, a single message can be fragmented and sent in multiple transport packets, although this is typically only used for large packets which are longer than the Maximum Transmission Unit (MTU).

The program running on the ARM Cortex-A9 Processor is responsible for creating all the required connections to the sensor array. These connections make use of the many networking libraries provided in the Petalinux operating system.

3.2.1 TracFind

TracFind is a simple UDP-based protocol for finding and performing the initial setup of Trac devices on a given network. This protocol includes six message types used to enquire info, provide info, factory reset, factory set, read the Real-Time Clock (RTC), and set the RTC. To find all Trac devices on a network, a user can send out a UDP broadcast containing

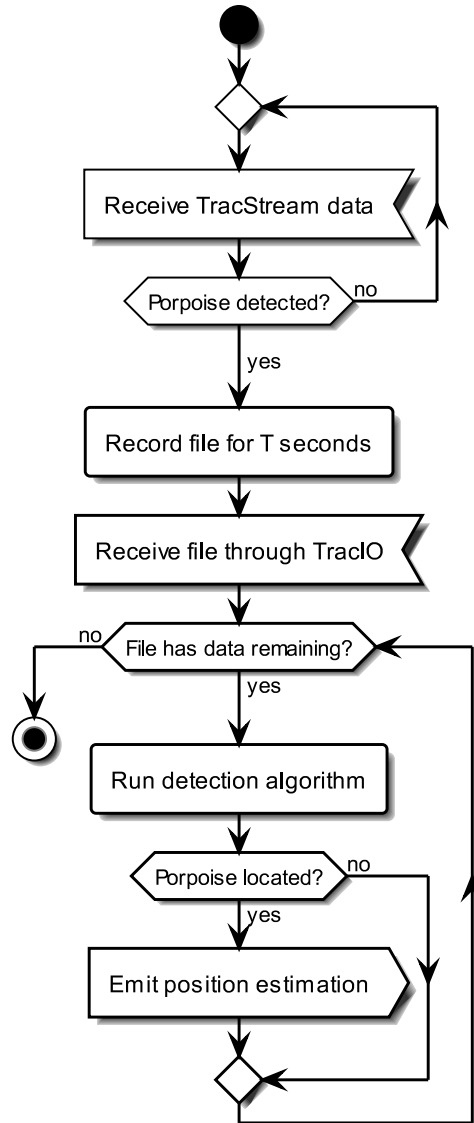


Figure 3.3: Activity diagram for the program running on the Zynq Development Board.

the enquire message. Then, all connected Trac devices will reply using UDP by sending out a TracFind info message to the Internet Protocol (IP) address that sent the enquire message. This info contains the product ID, revision number, unit number, Media Access Control (MAC) address, Internet Protocol Version 4 (IPv4)/Internet Protocol Version 6 (IPv6) address, the status of the file system lock, and the device name.

In the implemented design, the ARM Cortex-A9 Processor sends out the TracFind Enquire UDP broadcast message requesting information on all Trac devices connected to the network, while the Zynq-7000 PL remains idle. The Trac device in the pressure case responds back with the TracFind Info UDP message. This sequence can be seen in Figure 3.4.

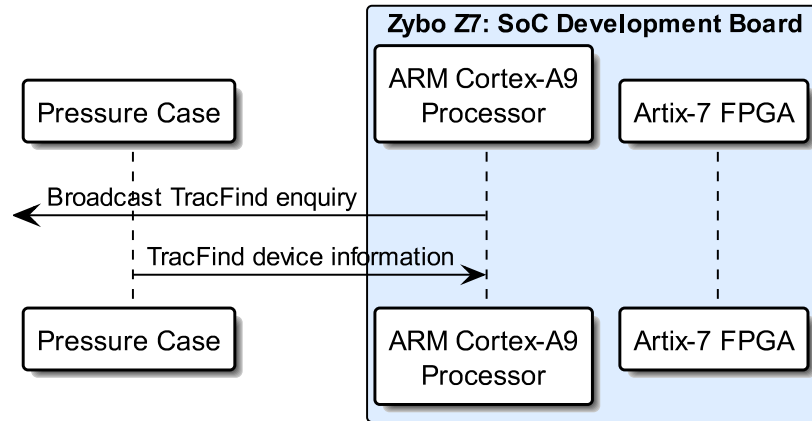


Figure 3.4: Program sequence diagram for the TracFind protocol.

3.2.2 TracIO

TracIO is a TCP-based protocol for configuring, controlling, and obtaining the status of Trac devices on the network. In particular, this protocol is used to control the streaming and recording status of the sensor array. Additionally, it is used to configure which hard drive is in use and to ensure that it is properly formatted. Finally, it is used to configure the bit depth and sampling rate of the Data Acquisition (DAQ) system used in both streaming and recording.

Utilizing the IP address found using TracFind, the ARM processor initiates a bidirectional TCP connection to the Trac device in the pressure case. Then, the ARM processor queries the pressure case for important device information such as the protocol version, firmware version, max packet size, etc. Next, the pressure case is formatted (all existing files deleted) and set to stream on two channels. This data stream is transferred to the ARM processor, which forwards it to the Zynq-7000 PL to run the detection algorithm. This data stream continues until a detection is observed and the ARM processor requests a switch from two-channel data streaming to five-channel data recording. Once this recording has been completed, the ARM processor requests for the most recent file to be transferred over. This file is stored onto the microSD card in the Zynq Development Board. Once the file transfer is complete, the ARM processor sends over short windows of channel data to the Zynq-7000 PL, which replies with the estimated position. The ARM processor then sends this position estimation to the map display. This sequence can be seen in Figure 3.5.

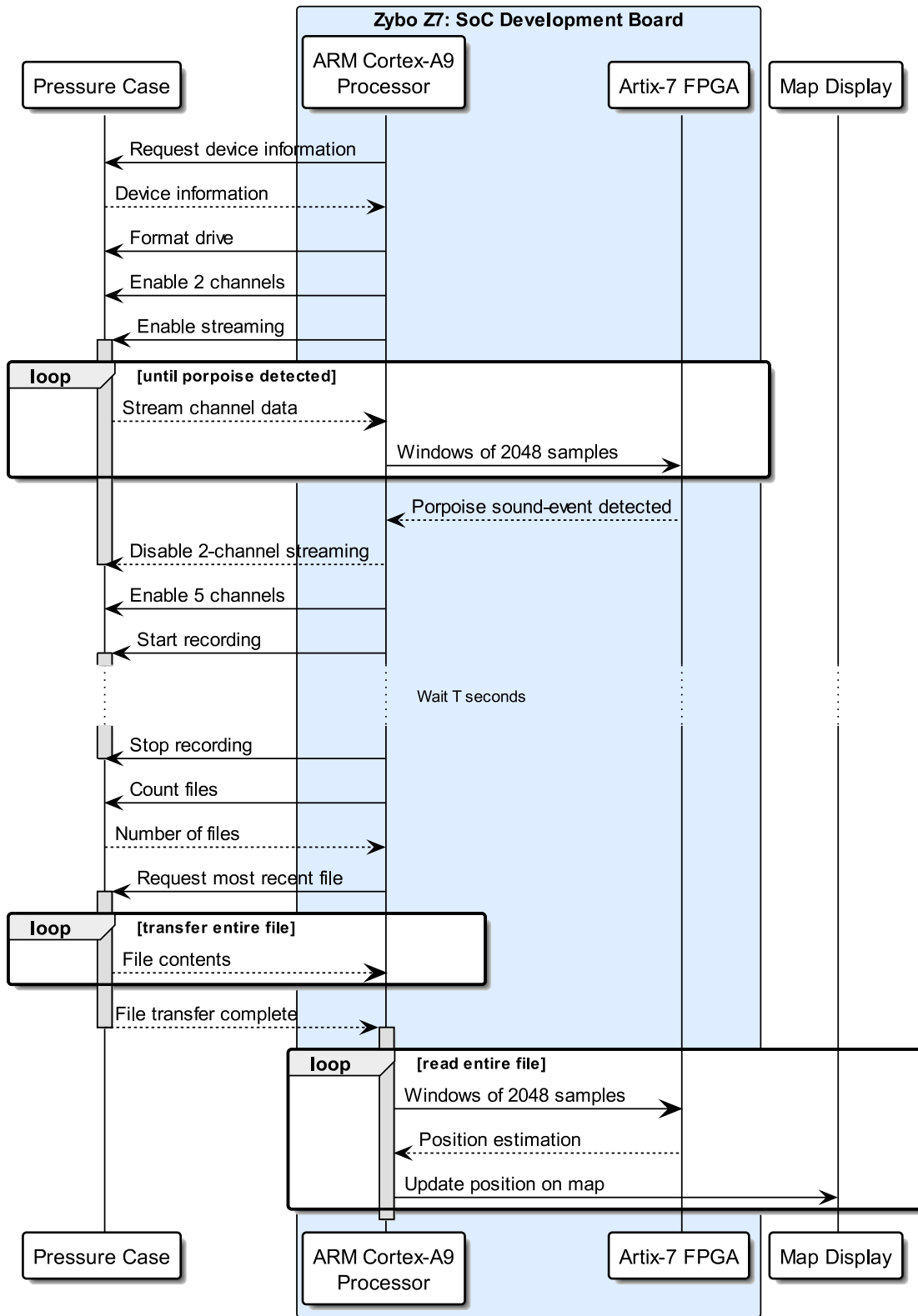


Figure 3.5: Program sequence diagram for the Trac protocol.

3.2.3 TracStream

TracStream is a simple UDP-based protocol for acquiring channel data in real-time. This protocol contains two main message types used to for channel data acquisition and for channel noise estimation. Both of these messages The DAQ operates using Pulse-Code Modulation (PCM) and can send channel data in PCM16 or PCM24 for 16-bit or 24-bit samples, respectively. Both messages contain a sequence number to allow the receiver to correctly interpret channel data in the event of dropped or reordered packets. The DAQ message specifies the bit depth of the samples, the sampling frequency (Hz), a map of which channel data is contained in the message, the number of samples in the message, and the raw sample data. The noise estimation message contains the frequency bin size (Hz), Sound Pressure Level (SPL) for each bin, time span analyzed (ms), number of overlapping samples, SPL max, SPL min, and SPL average.

In the implemented design, the ARM Cortex-A9 Processor uses the TracIO protocol to initiate streaming using the TracStream UDP protocol, while the Zynq-7000 PL remains idle. The Trac device in the pressure case responds streams PCM16 channel data using TracStream until the streaming is disabled using the TracIO protocol. As the TracStream data is received by the ARM Cortex-A9, the data is forwarded to the detection algorithm running on the Zynq-7000 PL. This sequence can be seen in Figure 3.6.

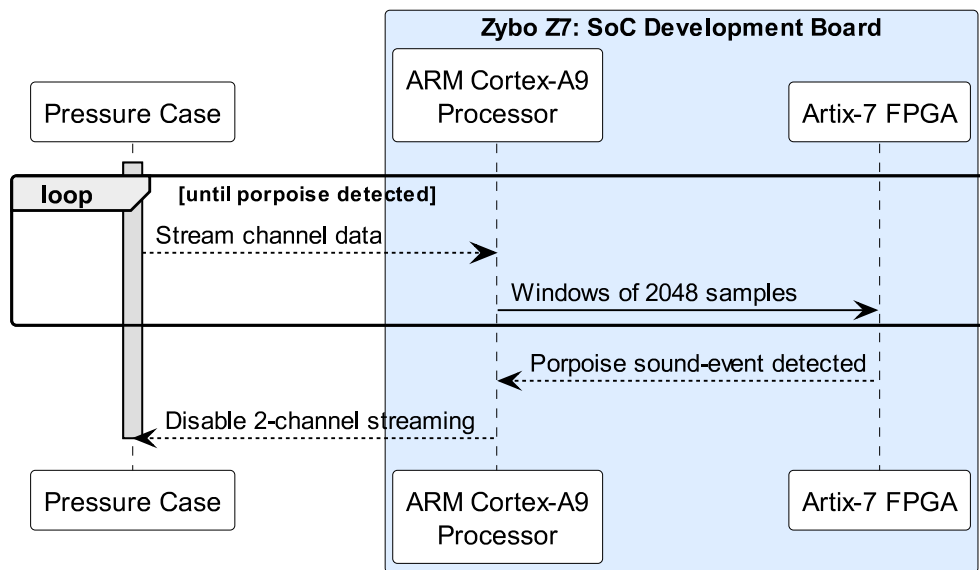


Figure 3.6: Program sequence diagram for the TracStream protocol.

3.3 Petalinux

As previously mentioned, the Petalinux Linux distribution is used as the operating system on the Zynq Development Board. In Section 3.3.1, the file system and important files used by Petalinux are described, then, in Section 3.3.2, the process of designing custom hardware for use in Petalinux is described. Finally, in Section 3.3.3, the communication between a host computer and the Zynq Development Board is described.

3.3.1 The File System

Petalinux can be stored on a microSD card using two partitions, one containing the required data to boot the operating system and then another for the user file system. The first partition was labelled BOOT and uses the File Allocation Table (32-bit) (FAT32) file system with a size of at least 500 MB and contains two key files: BOOT.bin and image.ub. The BOOT.bin file contains the First-Stage BootLoader (FSBL), the FPGA Bitstream, and the U-Boot Bootloader. All of these files are used to boot the Petalinux operating system and configure the FPGA hardware. The image.ub file contains information about the device tree, contains the kernel image, and the Early Userspace Image (initramfs) which are all used during the boot sequence.

The second partition was labelled FILES using the Fourth EXtended Filesystem (ext4) which uses the remaining space of the microSD card. This partition contains the user file system used in Petalinux and follows a standard Linux file structure. The first important directory is bin which contains a list of symbolic links to command line utilities such as grep (global regular expression print), ls (list files), mv (move files) etc. Additionally, there is a bin directory in the usr directory which has programs such as gcc (GNU's Not Unix! (GNU) Compiler Collection), g++ (GNU C++ Compiler), and sudo. Another important directory is the home folder, which contains the home folders for each user. It is where user generated files such as program source code and user compiled executables. The structure for the partitions can be seen in Figure 3.7.

3.3.2 Petalinux Design Flow

The Petalinux design flow starts by using the Xilinx Vivado Design suite to design and properly test your hardware design description. The hardware design is created using a

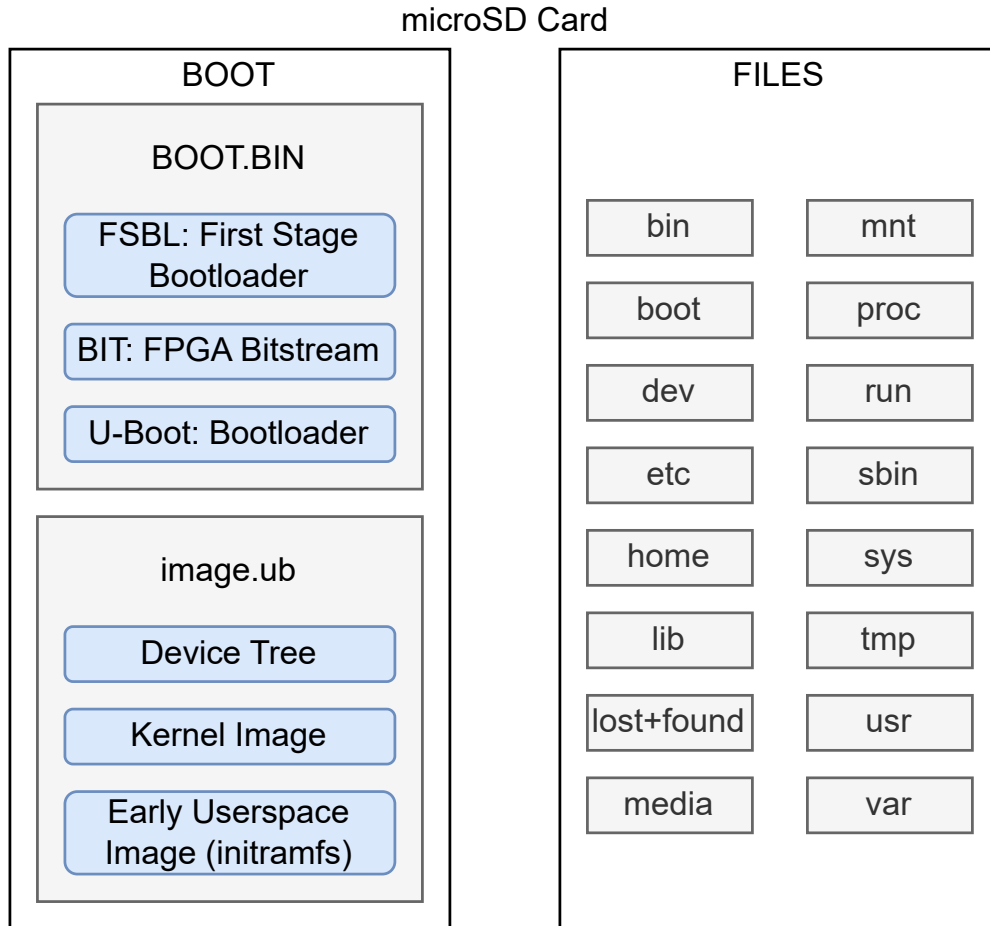


Figure 3.7: Two-partition file system used by Petalinux.

Hardware Description Language (HDL) such as Verilog or VHDL. Once the design is created, the design can be exported to a Hardware Descriptor File (HDF) by performing the synthesis and implementation of the design as shown in Figure 3.8. Furthermore, after running the implementation, the Bitstream (BIT) can be generated. The HDF contains three main portions, the Hardware Handoff (HWH), the Tool Command Language (TCL), and the FPGA Bitstream. The HWH allows for collaboration between the hardware design from Vivado and the software design used by the Xilinx Software Development Kit (SDK). The BIT is a file that is used to program the FPGA and is loaded during the boot sequence.

Next, Petalinux Config is used to create a configured Device Tree Source (DTS) file. To create this file, the HDF from Vivado is combined with the configurations for the root file system and the Linux kernel, as shown in Figure 3.9. The root file system configuration is used to specify which tools and libraries should be accessible from Petalinux. The DTS is

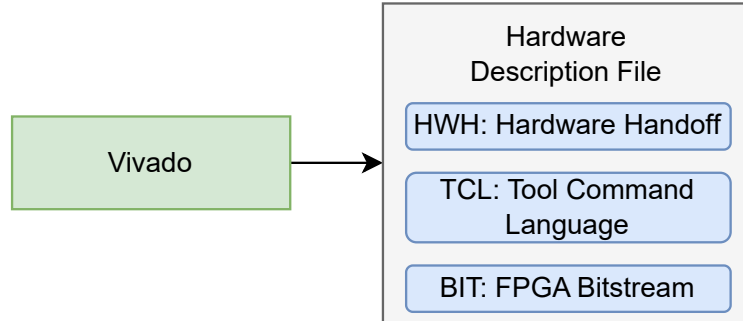


Figure 3.8: Exporting the Hardware Descriptor File using Vivado.

a file that describes all the hardware components from the processor and memory to other devices connected in the system. Additionally, it contains the software configuration used while running Petalinux and is included in the boot image of the system. After running Petalinux Config, it is crucial that the DTS is configured to include any custom hardware blocks to be available in the Petalinux environment. This is typically accomplished by adding all Advanced eXtensible Interface (AXI) devices as generic User-Space IO (UIO) devices in the DTS. This exposes the device in Petalinux and allows for sending data between the Petalinux operating system and the custom hardware.

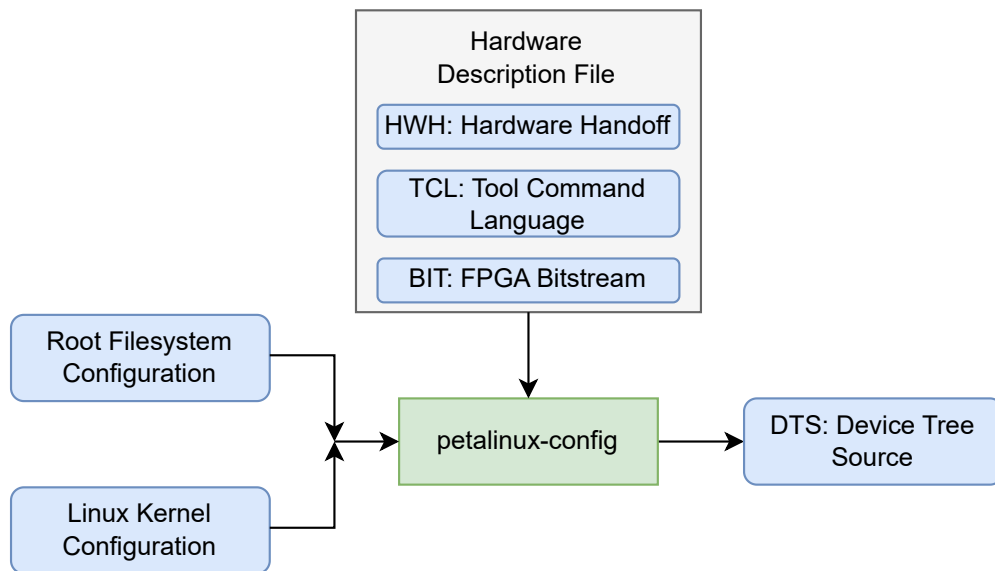


Figure 3.9: Using Petalinux Config to create the Device Tree Source (DTS).

After running Petalinux Config, the DTS can be built into several components using Petalinux Build, as shown in Figure 3.10. The first component is the Device Tree Binary

(DTB) which is simply a representation of the DTS in a binary format which can be embedded into the boot image of the system. The next two files are the FSBL and the U-Boot Bootloader, which are both used during the final boot sequence of Petalinux. Additionally, the Linux kernel is built and a root file system is created. Finally, the final boot images are created.

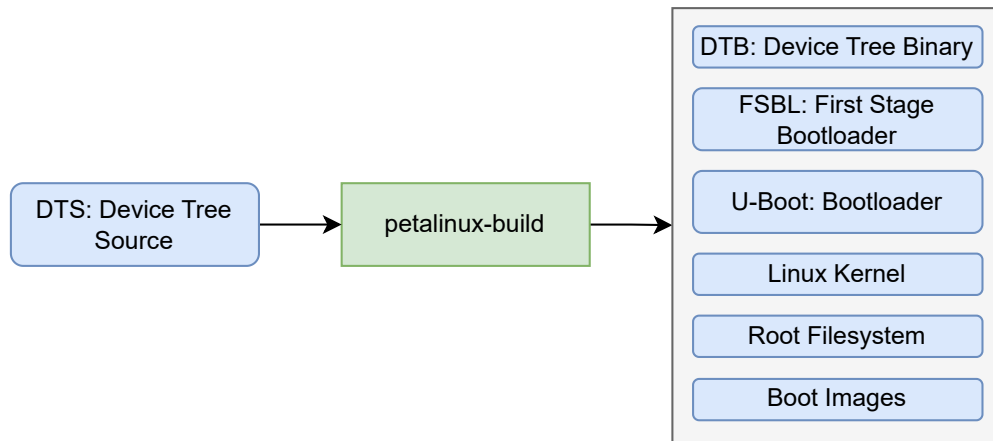


Figure 3.10: Using Petalinux Build to create the DTB, bootloaders, kernel, file system, and boot images.

The results of the Petalinux Build process can then be packaged up using Petalinux Package as shown in Figure 3.11. This process combines the results of the Petalinux Build Process with the BIT file produced by Vivado. The files are packaged into three files: `rootfs.ext4`, `BOOT.BIN`, and `image.ub`. The first file is `rootfs.ext4`, which contains the root ext4 file system used by Petalinux as described in Section 3.3.1. This is the main file system used throughout the runtime of Petalinux containing all the specified tools and libraries, but it is not bootable by itself. The `BOOT.BIN` file contains the bootloader required to boot Petalinux, and `image.ub` contains the kernel image that is loaded by the bootloader.

By combining all of these steps, a bootable Petalinux using custom hardware can be created as shown in Figure 3.12. As previously mentioned, the design flow starts with Vivado, where the HDF is created. Then, Petalinux Config is used to create the DTS. Petalinux Build uses this DTS to build the required files to boot. Finally, Petalinux Package combines these files into a root file system, a kernel image, and the bootloader used to boot the Petalinux system.

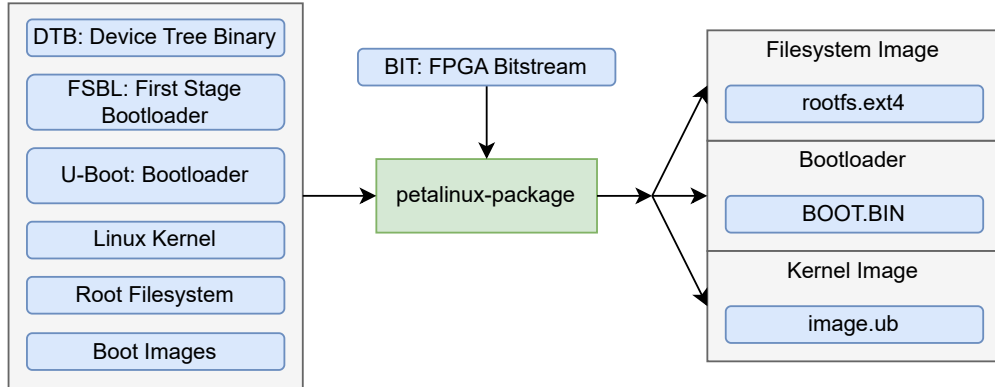


Figure 3.11: Using Petalinux Package to create the filesystem image, bootloader, and kernel image.

3.3.3 Serial Communication

Communication with the Petalinux operating system running on the Zynq-7000 PS is facilitated using a UART serial connection. This communication is accomplished by connecting the Zynq Development Board using a micro Universal Serial Bus (USB) cable to a host computer. Once connected, a serial console such as PuTTY can be used to send and receive data over the serial connection. This is used by the host computer to send control commands to be executed on the development board. The output of these commands are then sent back to the host computer over the serial connection.

This serial connection can be used to create, compile, and run programs directly on the Zynq Development Board. For instance, the command “vi main.c” can be sent from the host computer to execute the vi text editor on the development board. Then, the user could use vi through the serial connection to write a simple c program. After exiting vi, the gcc program could be used to compile the code by executing the control command “gcc -o main main.c”. Finally, the control command “./main” could be sent to run the compiled program and any output would be sent to the host computer via the serial connection.

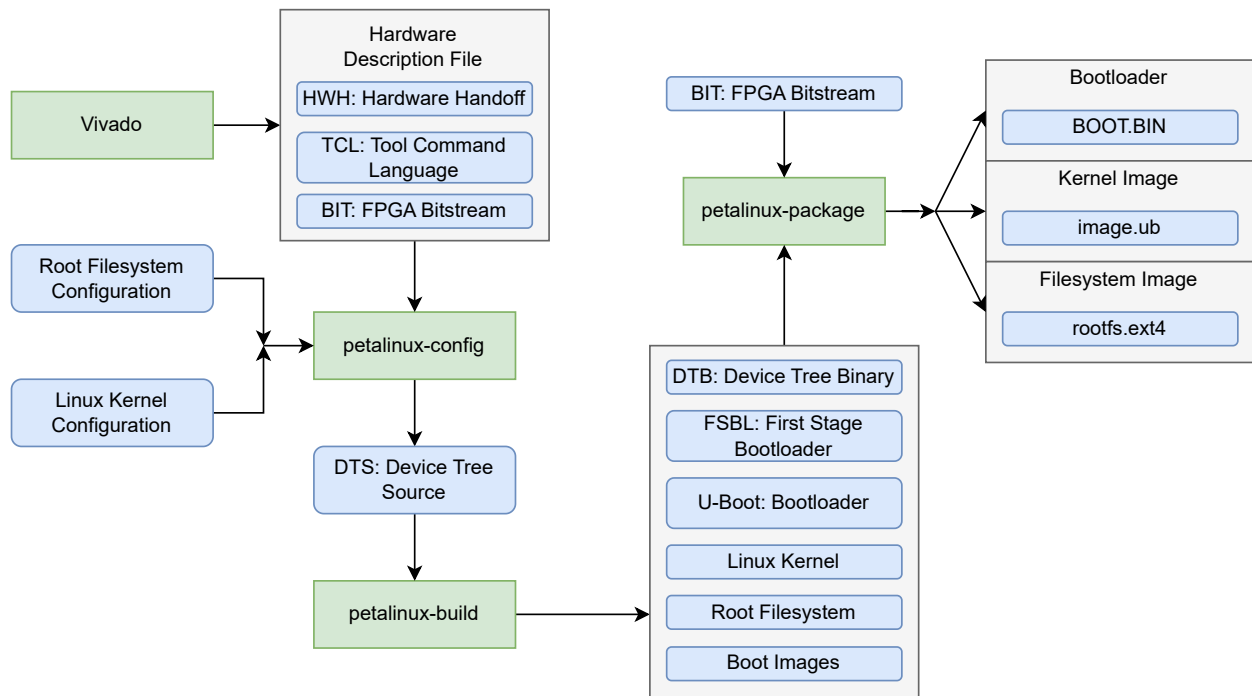


Figure 3.12: Complete workflow of configuring and building Petalinux using custom designed hardware.

Chapter 4

Experimentation at Sea

The objective of this chapter is to describe several of the experiments conducted throughout the development of the localization system. Preliminary control tests that were conducted in the Dalhousie University Aquatron will be discussed in Section 4.1. Then, the sea trial conducted in Herring Cove will be discussed in Section 4.3. Next, the berm trial and subsequent deployment conducted in Parrsboro will be discussed in Section 4.2 and Section 4.4, respectively. Finally, a sea trial conducted off the coast of New Zealand will be discussed in Section 4.5.

4.1 Preliminary Tests in the Aquatron

The first experiments were conducted on 2021-05-12 and 2021-05-13 in the Aquatron Laboratory. The Aquatron is a laboratory for controlled aquatic research which contains tanks of different shapes and sizes full of sea-water. These tanks are used to run many types of experiments in a controlled and reproducible environment. There are both rectangular and circular tanks that allow for testing different multi-path scenarios.

These experiments were conducted to verify the proper operation of the localization system. These tests were performed in a rectangular tank at the Aquatron in Dalhousie University. The tank, filled with seawater, measured 7.3 m by 9.1 m with a maximum depth of 4.25 m. During the tests, the water surface was 0.50 m below the top of the tank, leading to an effective depth of 3.75 m. The single transmitter was positioned near the middle of one of the 7.3 m edges of the tank at a depth of 1.47 m, while the five-receiver array was positioned near the middle of the opposite edge anchored to the bottom of the tank. A top view of the deployment is shown in Figure 4.1 and a side view is shown in Figure 4.2.

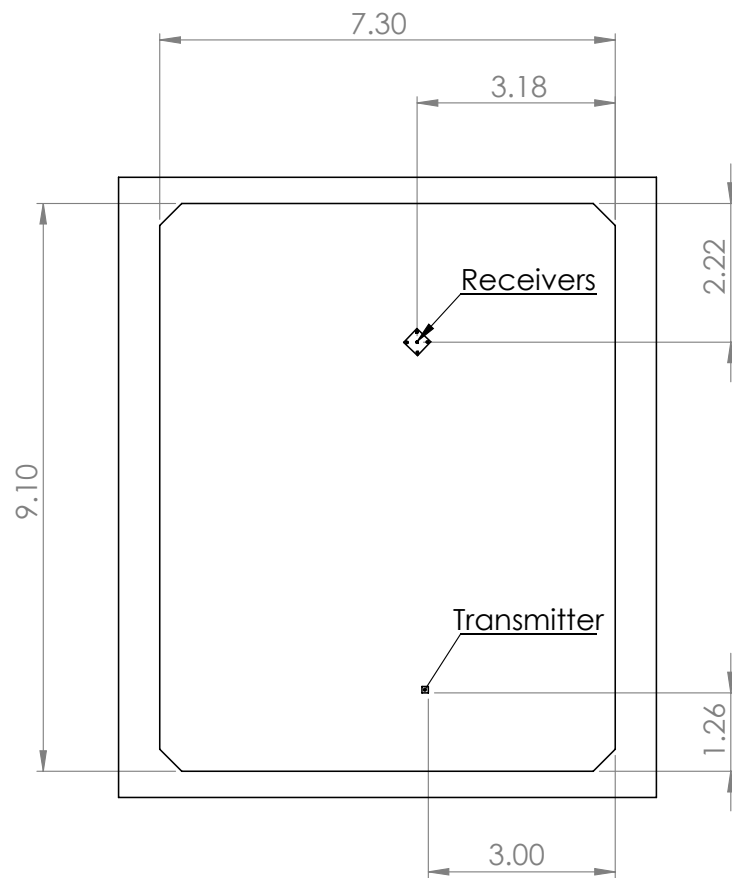


Figure 4.1: Measurements of the Aquatron trial (top view).

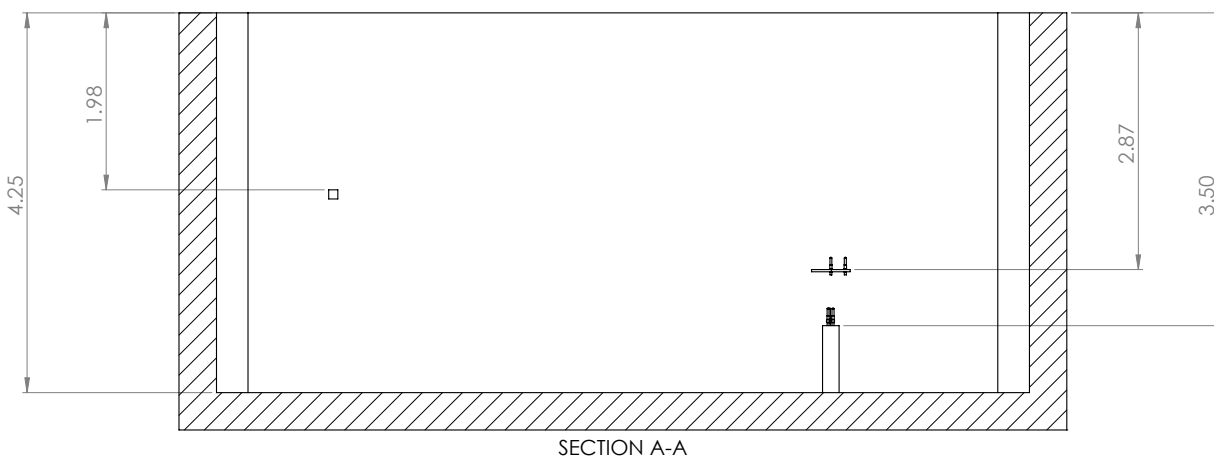


Figure 4.2: Measurements of the Aquatron trial (side view).

Note: measurements are in metres

The five-receiver array was composed of two horizontal planes of receivers. The bottom plane had three receivers in a “T” formation, where each receiver was 4.4 cm away from the centre of the array. The top plane was 24.7 cm above the bottom plane and had two receivers, one in the centre of the array and one 16.1 cm away towards the closest wall of the tank. This geometry was chosen as it closely resembles the proposed geometry to be deployed and should allow for estimation of range, bearing, and depth. This geometry is described in Table 4.1 and can be seen in Figure 4.3.

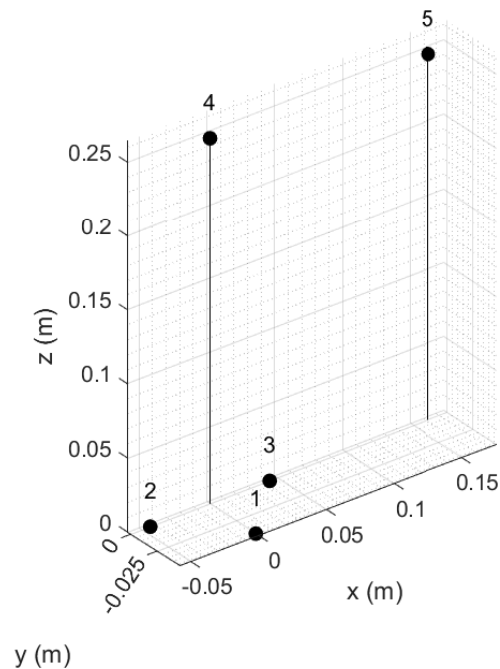


Figure 4.3: Sensor positions on the platform in the Aquatron trial.

Table 4.1: Sensor positions on the platform in the Aquatron trial

| N | X (m) | Y (m) | Z (m) |
|---|--------|--------|-------|
| 1 | 0.000 | -0.044 | 0.000 |
| 2 | -0.044 | 0.000 | 0.000 |
| 3 | 0.044 | 0.000 | 0.000 |
| 4 | 0.000 | 0.000 | 0.247 |
| 5 | 0.161 | 0.000 | 0.247 |

The first tests performed were performed in air. This test involved turning on the controller in the pressure case and connecting the case directly to a host laptop. The laptop

was used to issue several commands to ensure proper functionality of all subsystems including data streaming, file recording, file saving, and file transferring. All of these tests were performed using the Trac software provided by TR.

After the initial set of air tests were executed successfully, the Trac software was used to start a file recording. Once recording commenced, the pressure case was submerged into the pool and manoeuvred into the position previously indicated in Figure 4.1. The recording lasted the entire duration of the transmission and localization tests, only stopping after the pressure case was brought back to the surface. This recording captured on 5 channels and included recordings of the transmission and localization tests.

The localization test was performed by repeatedly sending a 20 ms snippet of a 27.5 kHz sine wave every 1 second. This signal is a low-frequency approximation of the clicks emitted by harbour porpoises, which each last 50 to 600 microseconds and are centred around 130-140 kHz. The arrivals of these bursts measured at each hydrophone allow for the determination of the position of the sound source. This train of bursts for one element of the array can be seen in Figure 4.4.

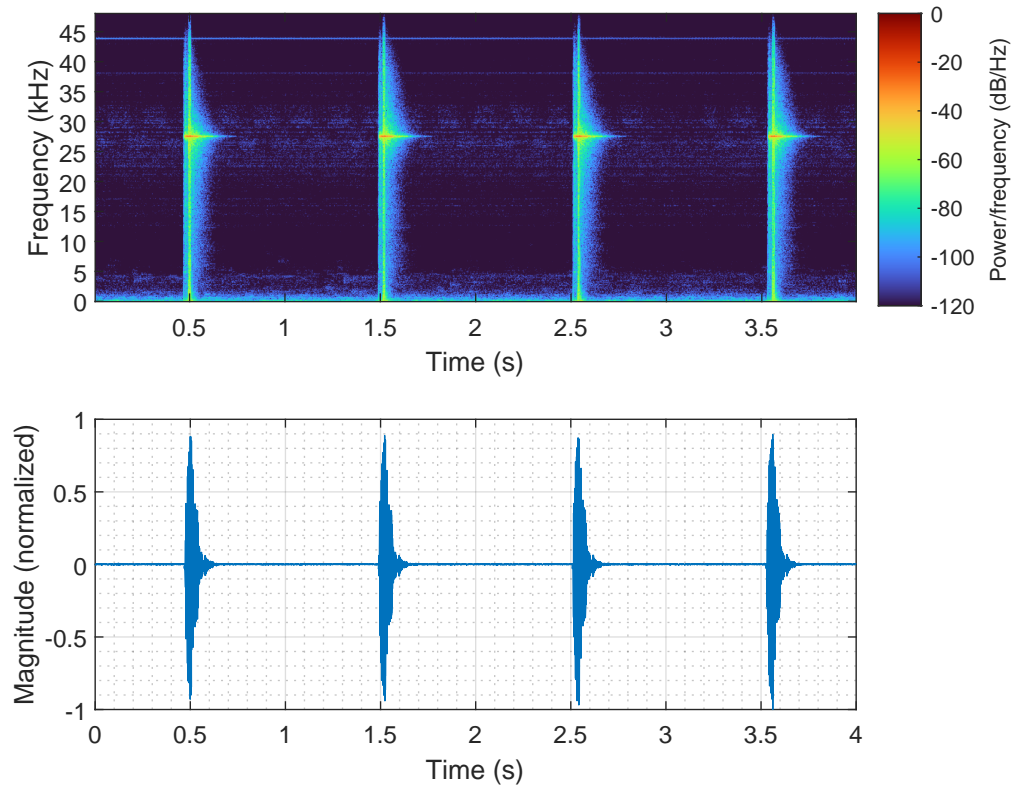


Figure 4.4: Representative sample of the 27.5 kHz clicks received during the Aquatron trial in the frequency domain (top) and time domain (bottom).

This test revealed that connecting the ZYBO directly to the power amplifier for the transmitter leads to a very poor transient response at the start of each burst. This transient response was improved upon by inserting an op-amp to both buffer the signal output by the ZYBO and amplify it by a factor of 2. This leads to a much cleaner transient response, since the power amplifier is not directly loading the output of the ZYBO.

The protocol tests were run by first querying the device for important device information including the IP address, firmware version, available features using TracFind and TracIO. The result of the TracFind test shows that the **TR Acoustic Device** is accessible at an IP address of **169.254.96.94**. This IP address was used to successfully create a TracIO connection with the TR Acoustic Device. Using this connection, the device announced it was capable of all required features such as acquiring data on 5 channels, recording, and streaming. However, it does not support optional features such as the dual sample rate mode, advanced file manipulation, and the sampling high pass filter.

Then, the second test was run to enable data streaming on two channels sampled at 96 kHz with a gain of 30.0 dB. This data stream was saved for a duration of 100 ms and stored in a file on a host computer containing the raw channel data for both channels. The protocol was then used to disable this data stream. This test was repeated for a total of three occurrences, successfully demonstrating that the protocol can be used to acquire raw channel data in real time.

During the next test, a file was recorded containing four channels sampled at 96 kHz, each with a gain of 30.0 dB. This data was recorded and saved directly to the Solid-State Drive (SSD) in the pressure case. After recording for 30 seconds, the protocol was used to disable the file recording. The protocol was then used to transfer over the 30 seconds of data recorded using four hydrophones to a host computer. Additionally, this recorded data was also transferred using the Trac software provided by TR. The files obtained using the protocol and the Trac software were then compared and found to be identical. This test demonstrated that the protocol can be used to record and obtain raw channel on four channels.

Finally, all of these tests were integrated together and performed sequentially. First, data streaming on two channels was sampled at a frequency of 96 kHz. Similar to the second test, this data was saved for 100 ms on a host computer. Then, the data from four hydrophones was recorded onto the internal SSD for a duration of 30 seconds. This file was

then transferred over to a host computer using the protocol. This test was repeated for a total of five occurrences. This test demonstrated that the software can successfully utilize both modes of operation (streaming and recording) and can switch between them at any time.

Overall, these tests demonstrated that the Trac protocol can be used in an underwater environment similar to that of the planned final deployment. This started by successfully finding and creating a connection to the TR Acoustic Device. Using this connection, the two mode of operation were successfully demonstrated, both individually and integrated together. These two modes of operation will allow for porpoise detection using the data from two hydrophones and porpoise localization using the data from four hydrophones.

4.2 Parrsboro Berm Trial

The second experiment was conducted on 2021-10-01 at the FORCE VC outside Parrsboro, Nova Scotia. This test was conducted on the berm of the Minas Passage in the Bay of Fundy. This area is known for being a high flow environment and has the highest tides in the world. This enabled the test platform to be deployed above water on the berm during low tide, while the platform would be 10-15 m below the water surface during high tide. This allowed for relatively easy deployment and recovery of the platform. The geometry of the sensor array on the platform is shown in Figure 2.5 and is described in Table 2.1.

Before the trial was conducted, the platform was deployed on the Berm during low tide in the afternoon of 2021-10-17. The deployed platform can be seen in Figure 4.5. Then, the main trial was conducted in the late morning of 2021-10-01 during high tide. Finally, the platform was recovered during low tide in the early morning of 2021-10-06. The recovered platform can be seen in Figure 4.6. The trial was scheduled to ensure the largest tidal range between deployment and the trials performed. This was done by considering both the lunar semidiurnal cycle and the lunisolar cycle. The lunar semidiurnal cycle is the typical tide cycle, where there are approximately two high and two low tides per day. The lunisolar cycle is a longer cycle involving the interaction of both the sun and the moon, which has a period of approximately 2 weeks. During new and full moons, spring tide occurs where the tidal range is the largest. In contrast, during first quarter and third quarter moons, neap tide occurs where the tidal range is the smallest. The dates of the deployment and trial were chosen to be close to spring tide to allow for the highest tidal range.



Figure 4.5: The initial deployment of the platform deployed in Parrsboro on the berm.



Figure 4.6: The recovery of the platform deployed in Parrsboro on the berm.

Due to the challenges involved with a shallow environment, a large tidal range is favoured to allow the trial to occur in water that is as deep as possible. In a shallow environment, the effect of multipath can severely impact and degrade the performance of underwater acoustic communication and localization systems. Additionally, the Bay of Fundy is a high water flow environment (several m/s is typical), so the trial was started during high tide which is one of the times the flow is the slowest. To analyze the effect of flow on the localization system, multiple trials were conducted.

Throughout the trials, the under platform remained stationary while a sea vessel was used to move several transmitters. In total, twelve trials were performed where the sea vessel was stopped upstream and drifted downstream. This was done to eliminate the noise produced by the propeller. The first transmitter used was the vessel echosounder. This transmitter periodically emits a 5 ms chirp around 150 kHz several times per second (typically 20 to 30). The second transmitter used was the icTalk, which emitted a short click at 130 kHz ten times per second. Finally, a custom transmitter was used to emit a 1 ms 27.5 kHz click at 20 followed by 200 times per second. The summary of the trials performed during the Berm deployment can be found in Table 4.2.

Table 4.2: Summary of trials performed during the Berm deployment, including the transmitters used, start time, and end time for each trial.

| Trial # | Transmitter | Start Time (UTC) | End Time (UTC) |
|---------|--------------------------------|------------------|----------------|
| 1 | Vessel Echosounder | 13:48 | 13:51 |
| 2 | Vessel Echosounder | 14:04 | 14:11 |
| 3 | icTalk | 14:15 | 14:18 |
| 4 | icTalk | 14:18 | 14:24 |
| 5 | icTalk + Custom @ 20 click/sec | 14:28 | 14:35 |
| 6 | Custom @ 20 click/sec | 14:40 | 14:45 |
| 7 | Custom @ 200 click/sec | 14:47 | 14:48 |
| 8 | Custom @ 200 click/sec | 14:50 | 14:54 |
| 9 | Vessel Echosounder | 14:56 | 15:01 |
| 10 | Custom @ 20 click/sec | 15:06 | 15:11 |
| 11 | icTalk + Vessel Echosounder | 15:17 | 15:20 |
| 12 | Vessel Echosounder | 15:22 | 15:26 |

Throughout the trials, an Acoustic Doppler Current Profiler (ADCP) was used to measure the sea depth and the sound speed. The depth measured using the ADCP on the day of the trials can be seen in Figure 4.7, where the start of each trial is labelled. Note that

an additional 0.7 m was added to the depths since the ADCP is 0.7 m above the seabed. Additionally, the sound speed measured throughout the trials can be seen in Figure 4.8. Note that the sound speed profile is assumed to be constant due to the shallow nature of the trials.

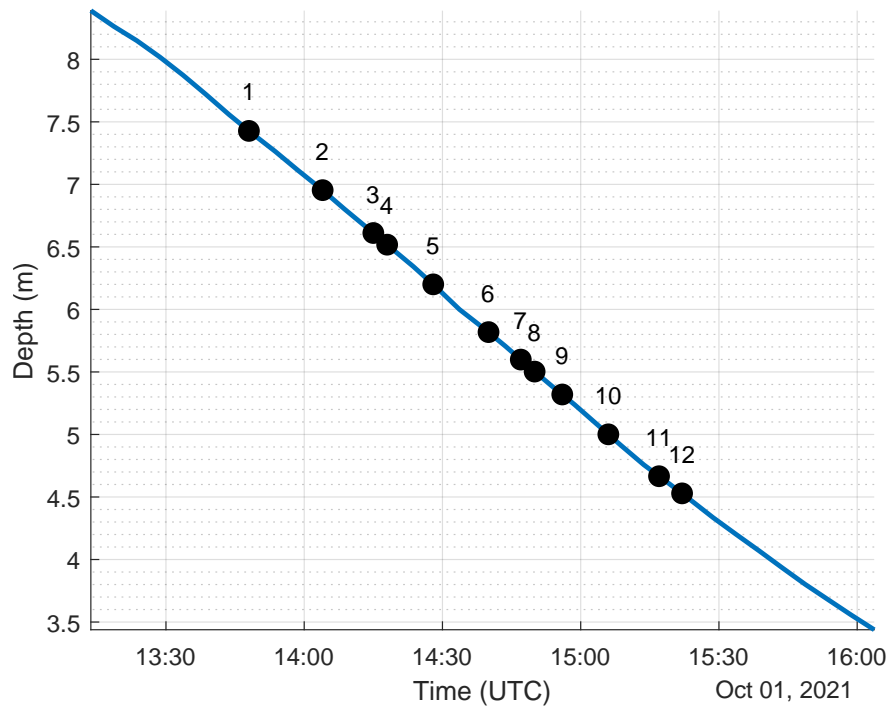


Figure 4.7: Depth of the water column measured using the ADCP during the Parrsboro berm deployment.

4.3 Herring Cove Sea Trial

The third experiment was conducted on 2021-11-02 off the coast of Herring Cove, which is just outside the Halifax harbour in Nova Scotia. In this trial, the Titan Reliance sea vessel was used to deploy a 4 element vertical planar array which was anchored to the sea bed. Throughout the entire trial, the vessel sonar was active, periodically emitting a 170 kHz pulse approximately 6 times per second. The localization results for the sonar is shown in Section 2.3.3. Due to the planar nature of the array, a localization ambiguity exists where there are two position estimations, one on each side of the plane. In the case of a vertical array, this causes an ambiguity in the azimuth estimation, where the estimation results in two different azimuth angles and the true azimuth cannot be determined. The azimuth can

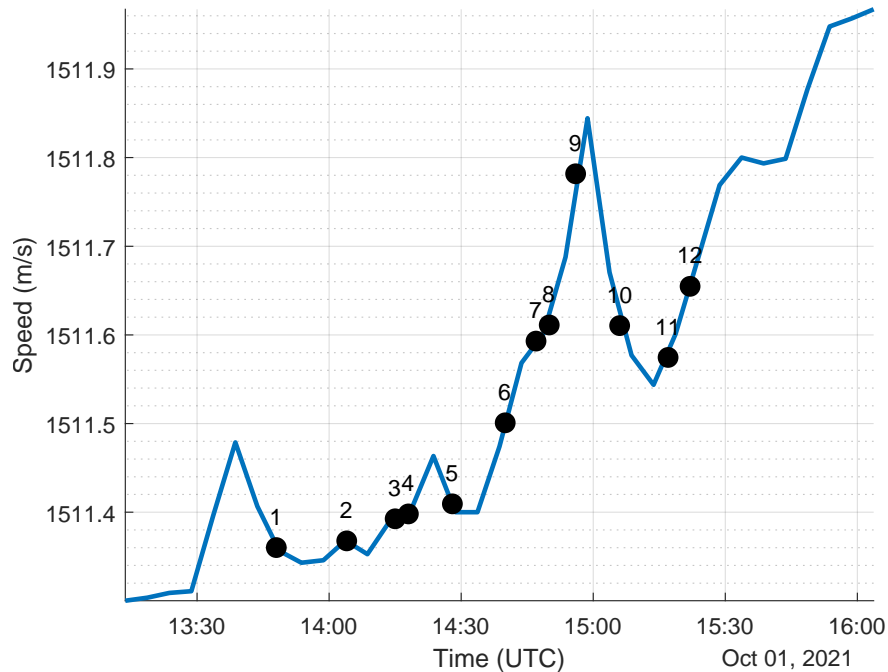


Figure 4.8: Sound speed in the water column measured using the ADCP during the Parrsboro berm deployment.

be estimated by determining the position of the sound source relative to one side of the plane, where the correct side is given by the GPS. The geometry of the planar sensor array is shown in Figure 2.7 and is described in Table 2.2.

After the array was deployed, the sea vessel navigated to eight waypoints spread around the sensor array. At these waypoints, an additional transmitter was used at varying heights of 6 m, 12 m, and 18 m. This transmitter was the icTalk as described in Section 4.2, which was still configured to emit a short click at 130 kHz ten times per second. The TDoA localization algorithm was used on the recordings of the icTalk. However, a manual process of identifying 15 arrival windows per waypoint (5 at each depth) was required due to the very low signal power of the icTalk. The delay estimation between each sensor in each of 15 windows at each of the 8 waypoints is shown in Figure 4.9. In this figure, the delays are shown in blue, red, and orange dots for the delays between channels 1 and 2, 1 and 3, and 1 and 4 respectively. Also, large vertical lines separate the time at each waypoint and thin lines separate the time at each depth. The depths are written along the top and the waypoints are listed along the bottom.

These delays were used as the input for the TDoA localization algorithm. The results of the azimuth estimation are shown in Figure 4.10. This figure is formatted in the same way

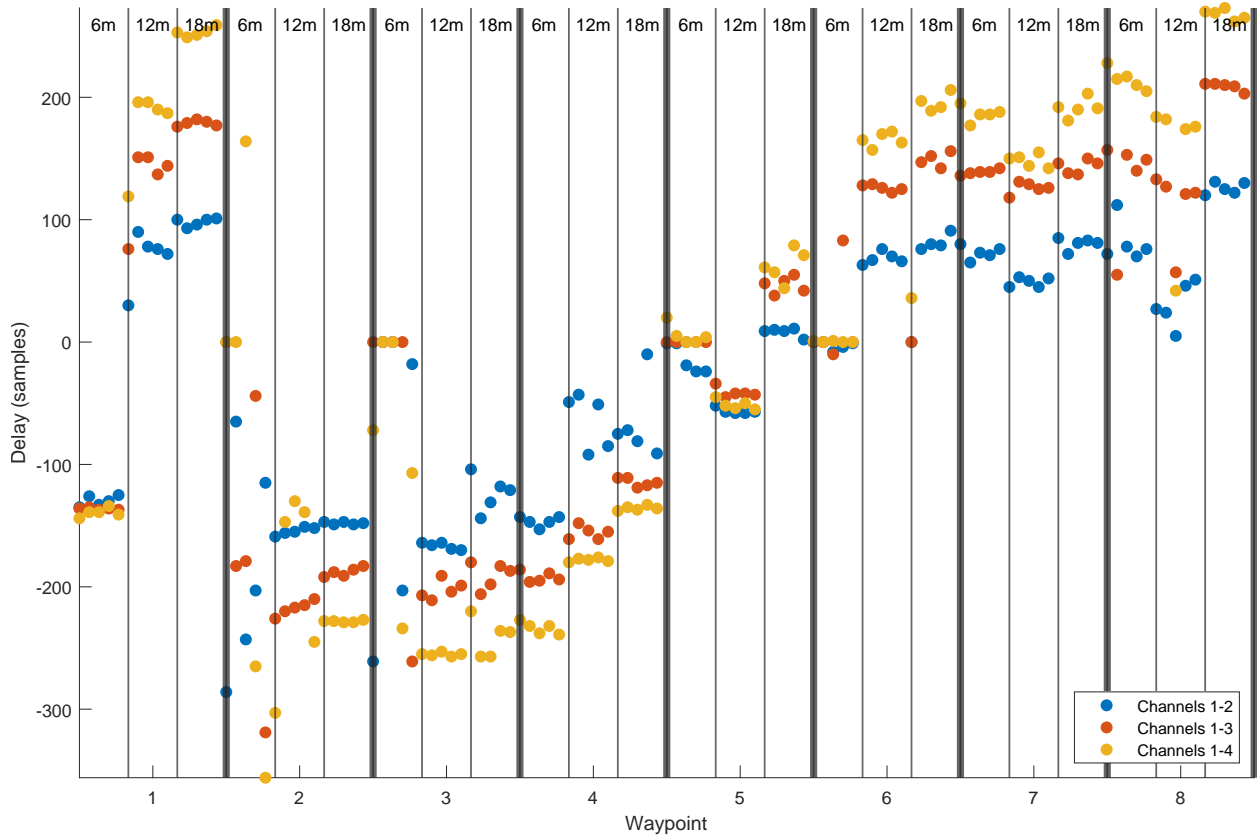


Figure 4.9: Measured delays between sensors using the icTalk at different depths and waypoints in the Herring Cove deployment.

as Figure 4.9, where large vertical lines separate each waypoint and thin lines separate each depth. To analyze the accuracy of the localization algorithm, the azimuth obtained using the GPS is shown. Generally, the azimuth estimation follows the expected azimuth from the GPS. However, the estimate fails at certain azimuth and depth combinations.

The TDoA localization algorithm was also used to obtain the inclination estimation in Figure 4.11. This figure is formatted in the same way as Figure 4.9 and Figure 4.10, where large vertical each waypoint and thin lines separate each depth. To analyze the accuracy of the localization algorithm, an estimate of the inclination is shown, which is based on the planned depth of the transmitter and the horizontal distance between the transmitter and receiver. Note that this distance sometimes changed between different depth trials. No consistent trend between the expected inclination and the estimated inclination can be found. This indicates that while this geometry allows for an azimuth estimation, it is very poor at estimating inclination.

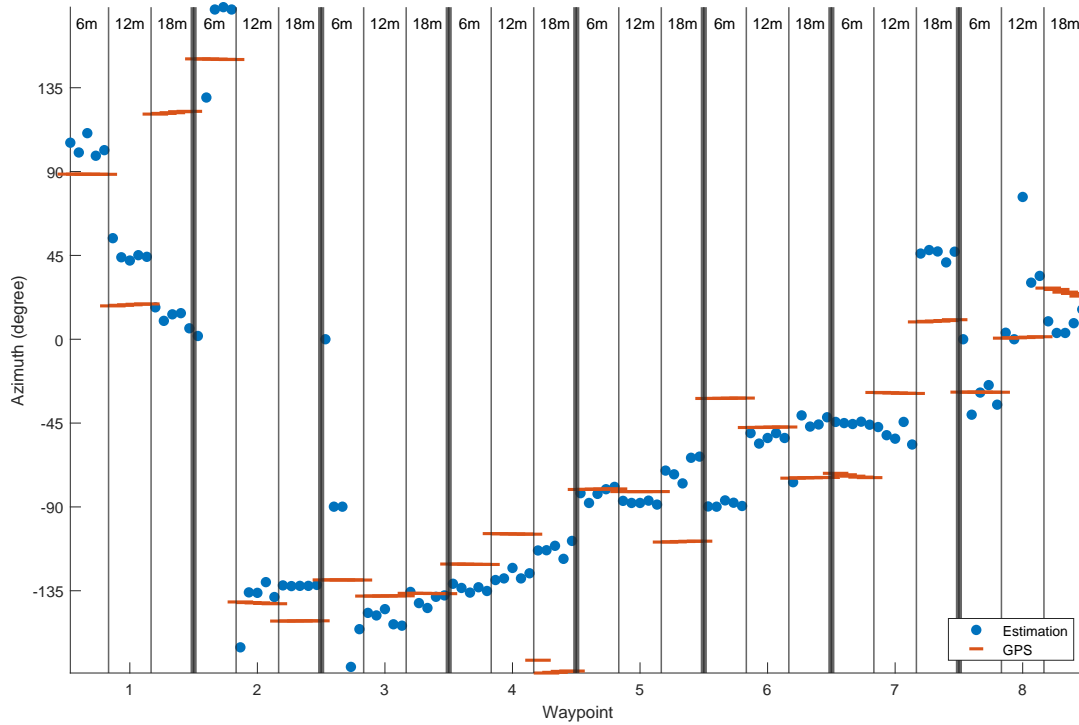


Figure 4.10: Azimuth estimation of the icTalk at different depths and waypoints in the Herring Cove deployment compared with the true azimuth obtained using the GPS.

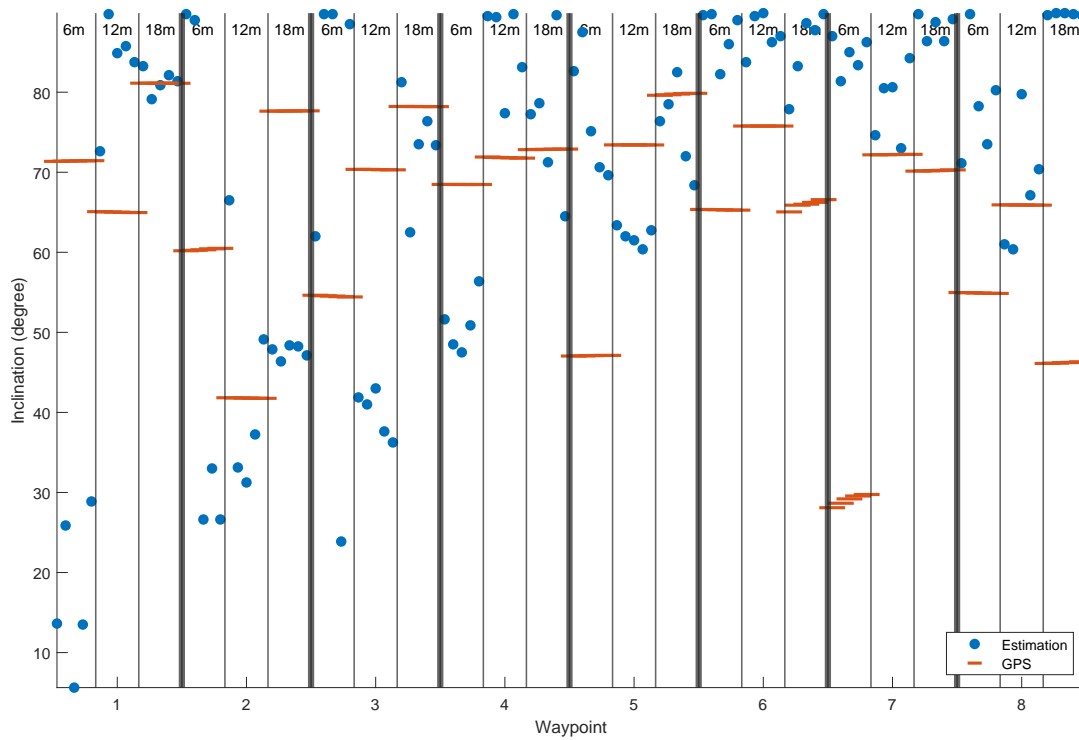


Figure 4.11: Inclination estimation of the icTalk at different depths and waypoints in the Herring Cove deployment compared with the true inclination obtained using the GPS.

4.4 Long Term Deployment in the Bay of Fundy

The fourth experiment began on 2022-05-25 when the platform tested in Section 4.2 was deployed 1 km off the coast of the FORCE VC. Once deployed, various tests were conducted to ensure the platform was fully operational. The first test performed was to verify proper functionality of the light and camera system to ensure they operated as expected. The next test conducted ensured that the system in the remote pressure case was powered on and was receiving power externally from the 1 km cable. After these tests were performed, several recordings were created and transferred from the sensor platform to the VC. The sensor case was configured with a schedule to create a two-minute recording every two hours. In total, 38 of these two minute recordings were captured during the trial between 2022-06-27 and 2022-06-30. This schedule was chosen to enable observing the environment in both the daytime and nighttime. Additionally, recordings were made during very low tide and very high tide. The geometry of the sensor array on the platform is shown in Figure 2.5 and is described in Table 2.1.

During these recordings, several sound sources were identified, including artificial (man-made) and natural (underwater vocalizations). A spectrogram of the recording that started on 2022-06-29 at 21:43:01 UTC can be seen in Figure 4.12. In this spectrogram, many artificial sound sources can be seen, including some constant tones at lower frequencies of 24 kHz and 45 kHz and some higher frequency tones at 106 kHz and 158 kHz. Additionally, there are consistent clicks at 114 kHz throughout the recording at 13 clicks per second.

Another sound source can be seen in a recording captured on 2022-06-28 at 21:08:48 UTC in Figure 4.13. This sound source was best captured on channels 2, 4, and 5 which are the sensors on the port side and front of the platform. The sound source resembles a click train with a duration of approximately 100 ms and contains a total of approximately 116 clicks, leading to a click rate of 1160 times per second. Due to the extremely high click rate and the very controlled click period, this sound source is likely to be artificial.

For each of the clicks shown in Figure 4.13, the time delay between arrivals were measured using a band-pass filter between 110kHz and 160kHz and calculated using the classic cross-correlation. Channel 5 was taken to be the reference channel, and the measured delays with respect to this reference channel are shown in Figure 4.14. These delays show there is a consistent 76 sample (0.198 ms) delay between channels 1 and 5 and either a 34 sample (0.089 ms) or -2 sample (-0.005 ms) delay between channels 3 and 5. The consistency of

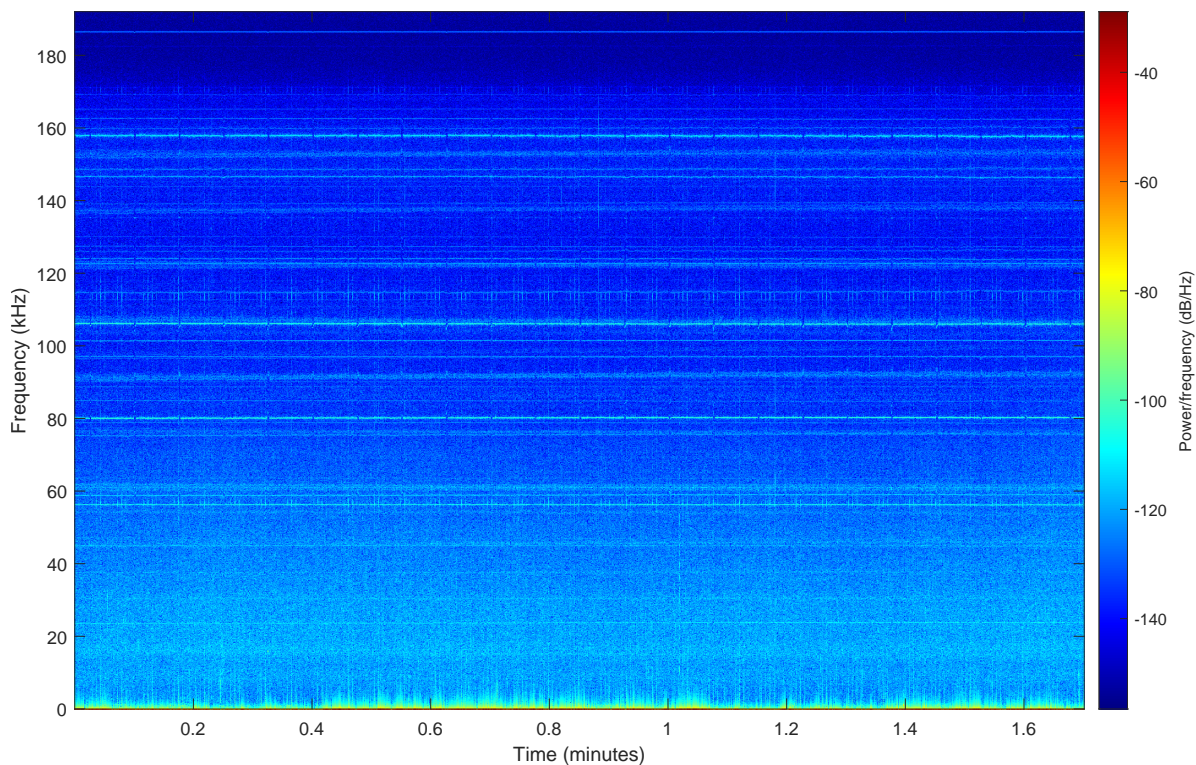


Figure 4.12: Representative sample of the spectrum during the Parrsboro deployment.

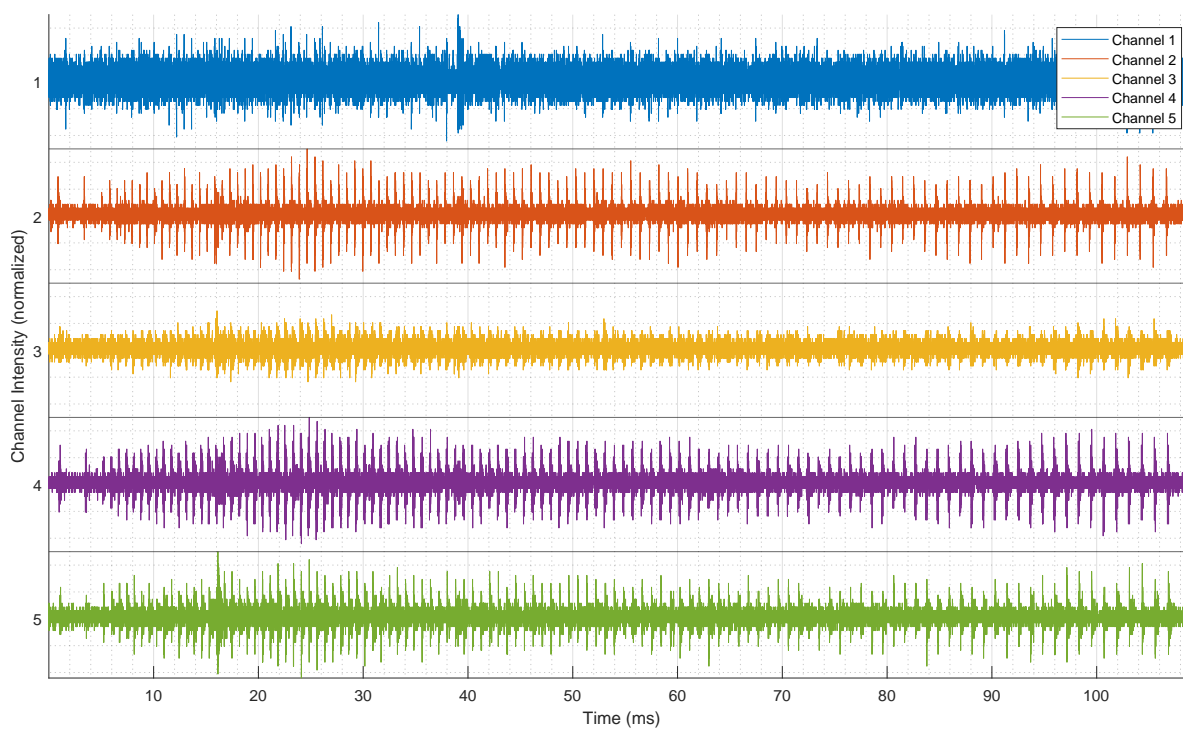


Figure 4.13: Sample of clicks received on 5 channels in the Parrsboro deployment.

these delays indicate that the source is stationary relative to the sensor array. However, channels 2 and 4 do not have a consistent delay with respect to channel 5, which indicates that the source is not stationary. This contradiction indicates that the signal observed on the sensors is likely an artifact caused by something on the platform. This could be caused by other instruments on the platform creating sound or creating a pulsating load on the power supply for the sensors and recorder.

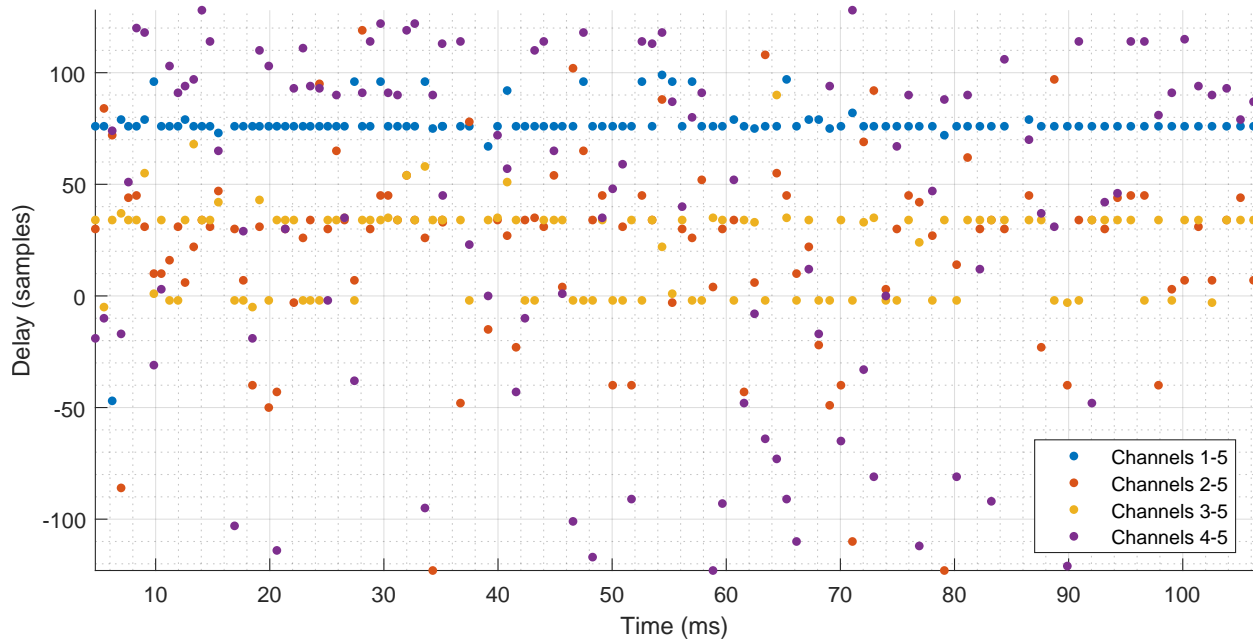


Figure 4.14: Measured delays between sensors in the Parrsboro deployment.

In a demonstration of the potential real-time aspect of the localization system, a porpoise-like sound source was simulated travelling in a spiral around a sensor array. This simulated data was used as an input for the localization system, and the spiral motion of the harbour porpoise was shown on the developed harbour porpoise map program. A capture of the map in the demonstration is shown in Figure 4.15. Note that in this map, the H in the red circle represents the location of the hydrophone sensor array. Additionally, the exact range and azimuth estimation is shown on the top left, as well as the polar coordinates converted to latitude and longitude coordinates. Also of note is the FORCE VC, which is visible in the top right of the map.



Figure 4.15: Capture of the harbour porpoise map in Parrsboro.

4.5 New Zealand Sea Trial

The fifth sea experiment took place between 2022-09-17 and 2022-10-31 off of the coast of New Zealand. In this trial, two trawl doors as described in Section 2.3.3 were towed along the seabed holding a fishing net. In this configuration, there were a total of eight hydrophones sensors, four on each of the doors. The geometry of the sensors are as described in Section 2.3.3. Unfortunately, the data from the second hydrophone on the starboard trawl door failed to be captured, so there are only three sensors that were operational on that side. Whereas, all four sensors were operational on the port trawl door. The geometry of the starboard sensor array is shown in Figure 4.16 and is described in Table 4.3. Furthermore, the geometry of the port sensor array is shown in Figure 4.17 and is described in Table 4.4

Throughout the sea trial, several species were observed, including the Hector's dolphin and the dusky dolphin. During the observations on 2022-09-22, a group of at least six adult Hector's dolphins were spotted swimming around the vessel, with a concentration in the stern region. The delays obtained using the third and fourth sensor (with respect to the

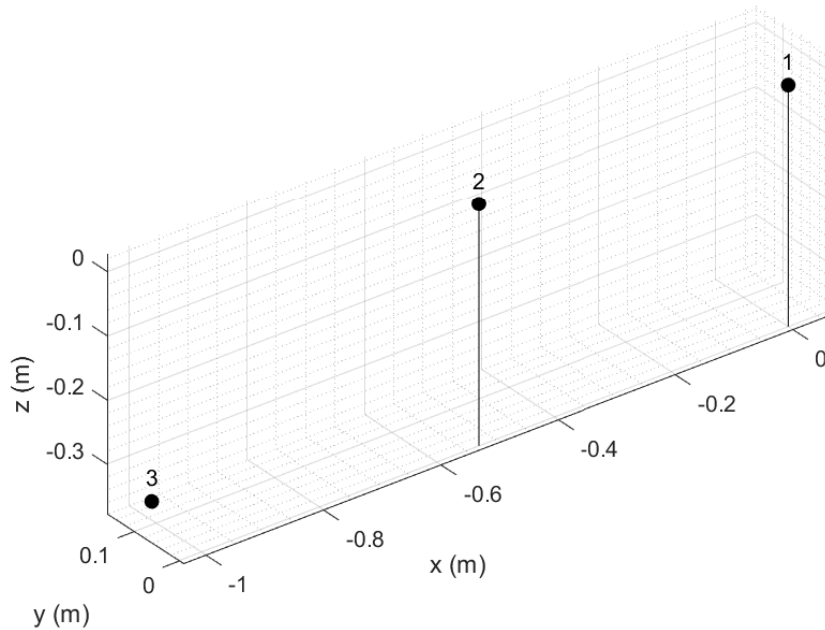


Figure 4.16: Sensor positions on the starboard trawl door in New Zealand.

Table 4.3: Sensor positions on the starboard trawl door in New Zealand

| N | X (m) | Y (m) | Z (m) |
|---|--------|-------|--------|
| 1 | 0.000 | 0.000 | 0.000 |
| 2 | -0.528 | 0.000 | 0.000 |
| 3 | -0.970 | 0.150 | -0.377 |

first sensor) are shown in Figure 4.18. Note that there are multiple artificial sound sources present, which represent the majority of the delays obtained. Namely, the sonar present on the vessel dominates other the naturally occurring, directional sounds of the different types of dolphins. However, at around 650 seconds into the observations, there is a clear region with smoothly changing delays. The delays from this time were filtered by removing outliers using a threshold on the moving median and are shown in Figure 4.19.

The delays shown in Figure 4.19 feature several regions of smoothly changing delays, which indicates continuous motion of a single sound source. However, there are regions where the delays of other sound sources are shown instead. This represents one of the challenges present in this environment, where there is a significant amount of activity in the frequency range of these Hector's dolphins. As mentioned before, the sonar on the vessel is rapidly sending out pulses in this same frequency range. A spectrogram of the combination of

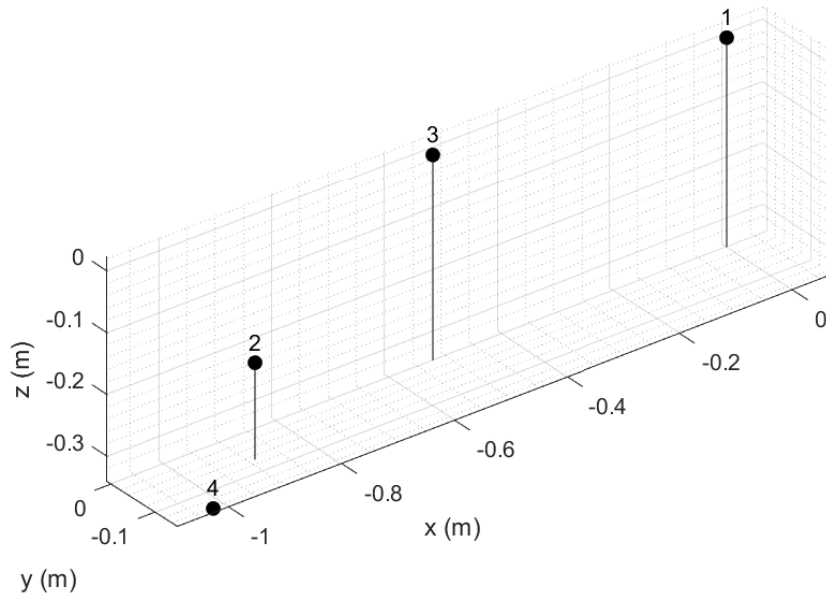


Figure 4.17: Sensor positions on the port trawl door in New Zealand.

Table 4.4: Sensor positions on the port trawl door in New Zealand

| N | X (m) | Y (m) | Z (m) |
|---|--------|--------|--------|
| 1 | 0.000 | 0.000 | 0.000 |
| 2 | -0.890 | -0.069 | -0.183 |
| 3 | 0.522 | 0.000 | -0.006 |
| 4 | -1.020 | -0.142 | -0.340 |

artificial and natural sound sources can be seen in Figure 4.20. Additionally, this combination can be seen in the time domain with a high-pass filter applied in Figure 4.21.

The positions of the dolphins were approximated over time using these delays by utilizing the TDoA algorithm. The azimuth and inclination estimations over time can be seen in Figure 4.22 and Figure 4.23, respectively. In the azimuth plot, the majority of the observations occur between -160 degrees and -90 degrees, which is in the stern of the vessel. This agrees with the observations notes that indicate the Hector's dolphins were spotted in the stern region.

In addition to the previously shown azimuth and inclination estimations, the range of Hector's dolphins was approximated. This is a very crude estimation of the range, as there were only three sensors capturing data at this time. The estimated azimuth and range are

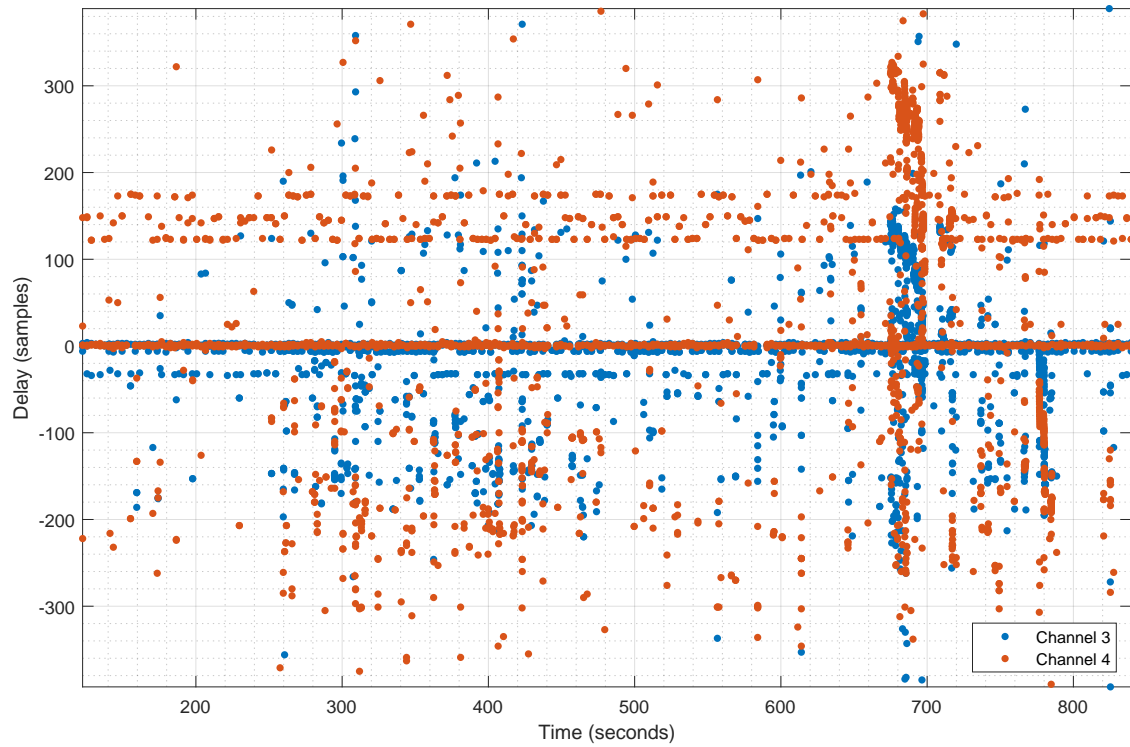


Figure 4.18: Measured delays between sensors in the New Zealand deployment.

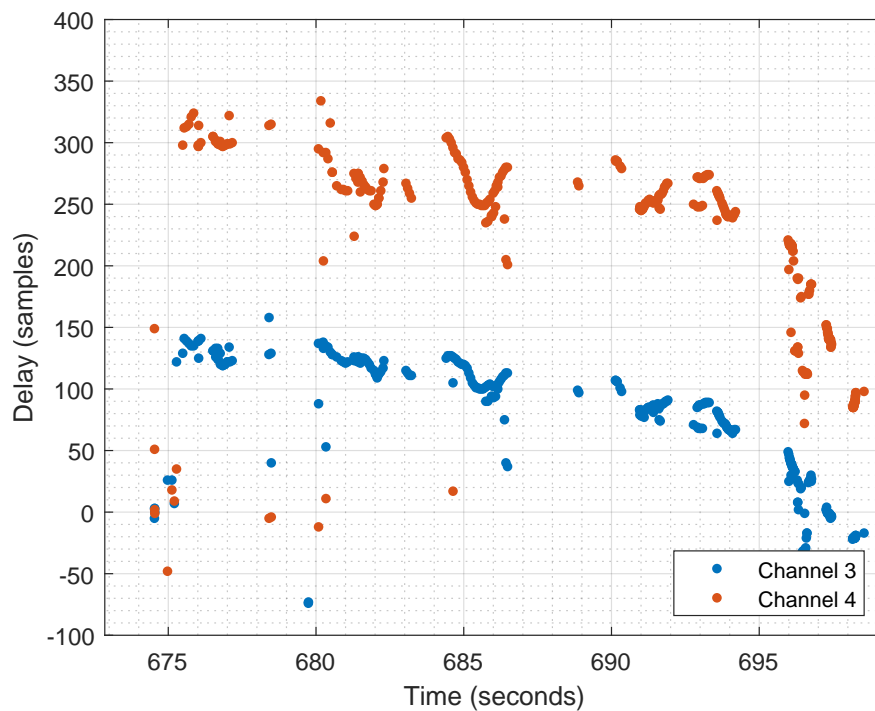


Figure 4.19: Measured delays between sensors in the New Zealand deployment, filtered by removing outliers using a threshold on the moving median.

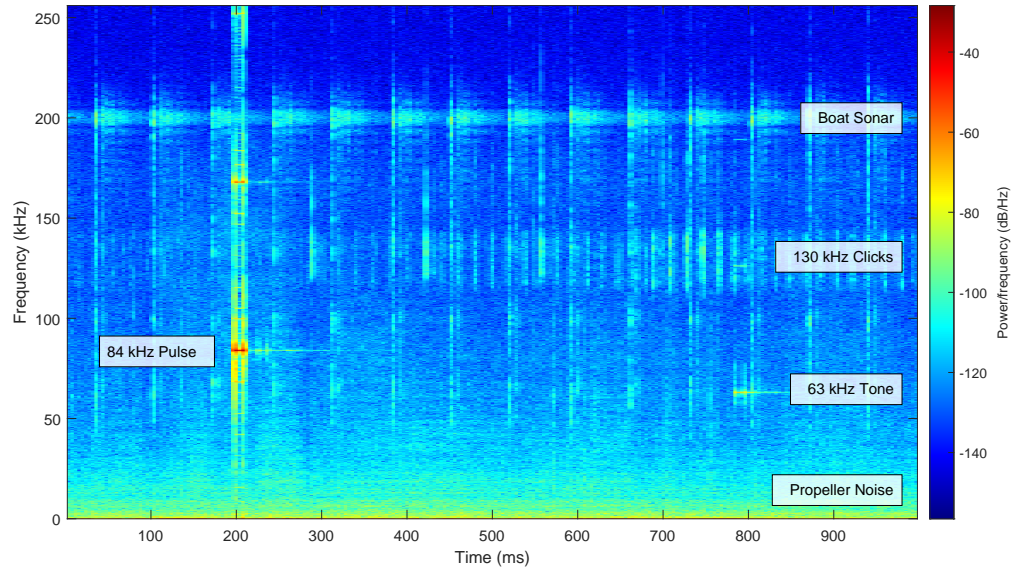


Figure 4.20: Representative sample of the spectrum during the New Zealand deployment.

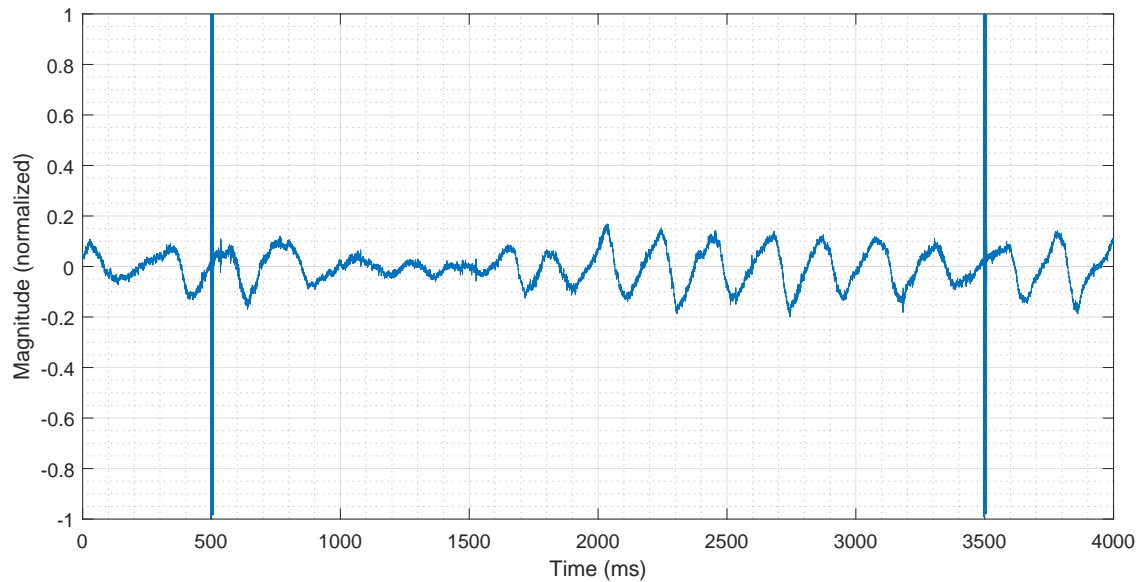


Figure 4.21: Sample of the multiple sources observed in the New Zealand deployment.

shown in the polar plot shown in Figure 4.24. Note that in this plot, the starboard sensor array is positioned along the horizontal axis, where an azimuth of 0 degrees is towards the boat and an azimuth of 180 degrees is towards the net. In this plot, the passage of time is indicated using the hue of the dots. The initial observations are indicated in blue, then further observations are indicated in green, then orange, then yellow, and the last observations indicated in red.

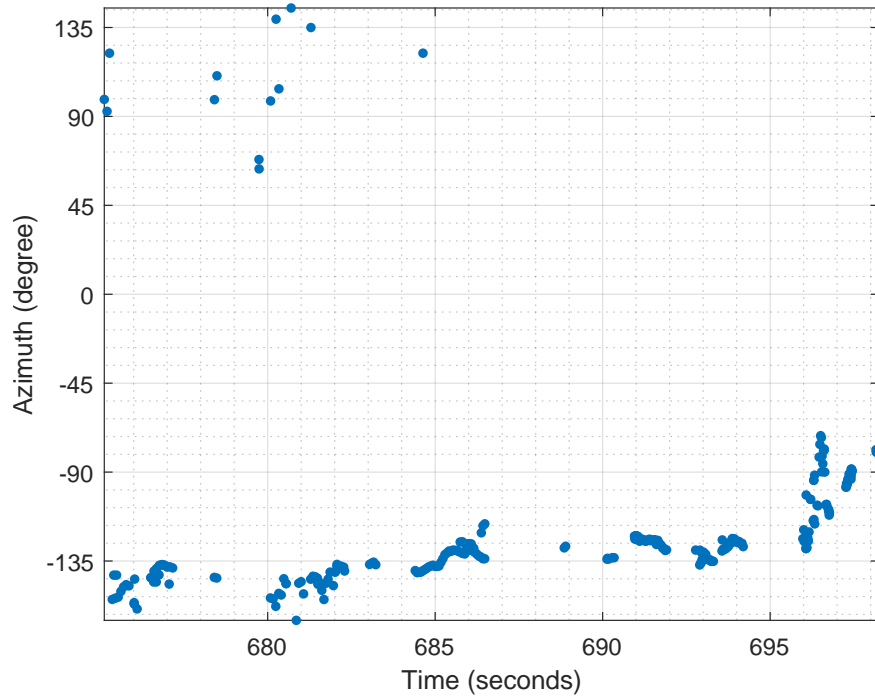


Figure 4.22: Azimuth estimation of the clicks identified in the New Zealand deployment.

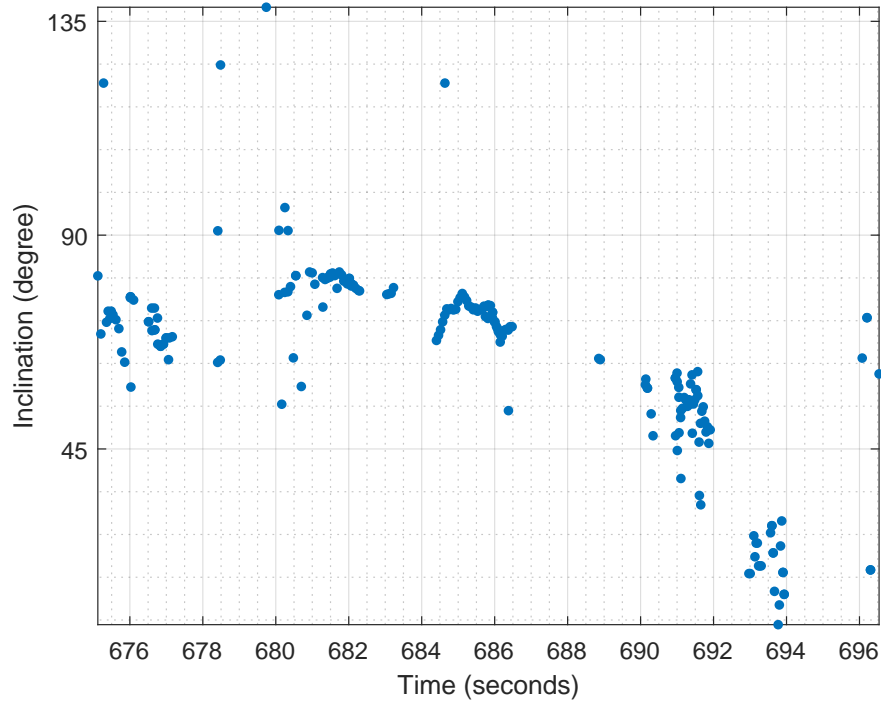


Figure 4.23: Inclination estimation of the clicks identified in the New Zealand deployment.

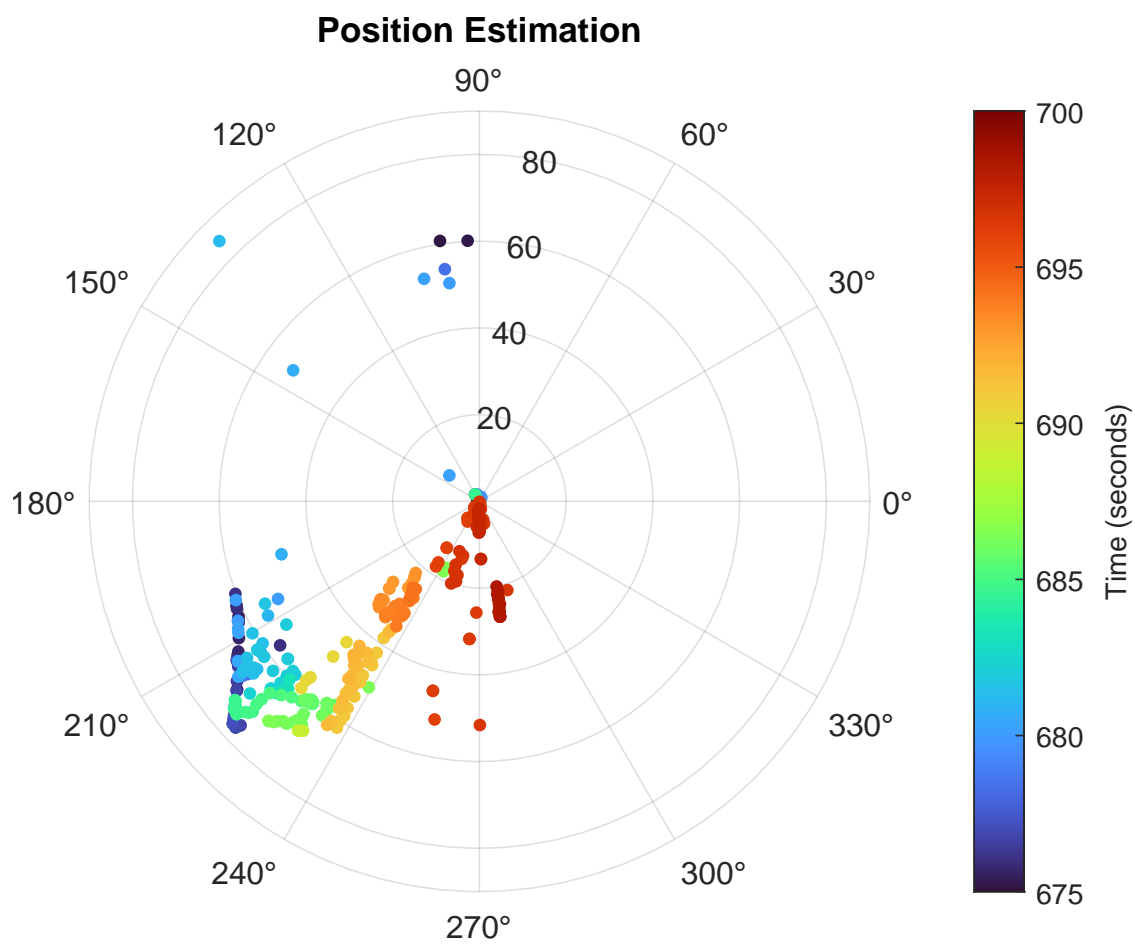


Figure 4.24: Azimuth and range estimation of the clicks identified in the New Zealand deployment, shown on a polar plot.

Chapter 5

Conclusion

In this chapter, the thesis contributions are summarized. Specifically, Section 5.1 outlines the major contributions of this work. Section 5.2 provides suggestions for potential future work.

5.1 Summary of Contributions

This thesis presents the development of a passive real-time underwater localization system utilizing the TDoA algorithm. This system was implemented on a Xilinx Zynq-7000 SoC, which contains both a Zynq dual-core ARM Cortex-A9 processor and an Artix-7 FPGA. The chosen development platform facilitated testing of various localization algorithms on the FPGA and allowed for easy controller implementation using Petalinux. A low-complexity Matched Field Processing algorithm was implemented on this platform in [21] and, in simulation, it outperformed a traditional TDoA algorithm in terms of mean and maximum estimation error. This resulted in an average reduction in the range estimation error by a factor of 20 and reduced the average azimuth estimation error from 0.7 degrees to 0.06 degrees. Additionally, it was demonstrated that the variance of these two errors is much lower, leading to a more predictable error performance. However, the algorithm was not ultimately implemented as the final algorithm due to its sensitivity to environmental conditions. These conditions include the spacing of the sensors, the depth of the sensor array, the depth of the water column, and the sound speed profile. Ultimately, TDoA was implemented as it was much less sensitive to these parameters, allowing for an accurate estimation in challenging or changing conditions.

The TDoA algorithm, similar to the implemented LCMFP algorithm, was also implemented on the Xilinx Zynq-7000 SoC. However, the algorithm was executed on the ARM Cortex-A9 processor instead of the Artix-7 FPGA. This decision was made to enable the potential for running multiple localization algorithms simultaneously, with LCMFP on the FPGA and TDoA on the ARM processor. Moreover, the TDoA algorithm could easily be

transferred to other platforms without requiring a FPGA. In order to support arbitrary sensor geometry, the implementation of the TDoA algorithm utilized a grid search to find the grid-points that correspond to the measured delay set. This enabled the determination of both the mean position and the standard deviation of the estimate. However, the grid search presented a significant challenge since increasing the maximum range or resolution of the grid size required an increase in the search grid size, which imposes a limit on real-time operation.

This limit for real-time operation was increased by utilizing a lookup table. This table contains entries for each possible localization result and is dependent on the sensor geometry, sampling frequency, and the sound speed. After using the table, minor changes to the sound speed can be accounted for by scaling the range estimation. Hence, the table for an array of sensors can be pre-computed by considering the maximum sampling frequency and used in all scenarios. For instance, in the case described in Section 2.2.1, a 3,001 by 3,001 by 210 point grid was used, which results in 950,449,711 grid points that must be evaluated for each localization estimation. Using the lookup table, this number was reduced by a factor of 3,500 down to 265,200 entries in the lookup table, significantly reducing the time required to perform a localization estimation. This allows for a larger grid size to be used in real-time scenarios.

Additionally, the limit was increased further by utilizing a hyperoctree, which is a space-partitioning data structure. This structure allows for efficient querying for the measured delay set, which removes the requirement to consider every entry in the table. In the case described in Section 2.2.1, the lookup table required considering all 265,200 entries, while the hyperoctree reduced this by an average factor of 296 down to 896 entries. Compared to the full grid search, using the hyperoctree represents a 1,060,770 times reduction in the number of entries each localization estimation must consider.

These improvements lead to even more drastic results for larger grids. For instance, in Section 4.5, the localization system used an 8,001 by 8,001 by 611 grid that includes 39,113,776,611 points. By applying the lookup table, the number of entries reduced by a factor of 153,612 down to 254,209 entries. Finally, using the hyperoctree structure further reduces the number of entries considered to a minimum of 48, a mean of 255, and a maximum of 656. Overall, this represents an average improvement by a factor of approximately 153 million compared to the original grid search, which enables the use of this large grid in

real-time.

In addition to these optimizations created for the implementation of the TDoA algorithm, further improvements were made by improving the sensor geometry to reduce the average estimation error, as described in Section 2.3. This improvement was accomplished by evaluating the different possible sensor configurations by computing the estimation error for each grid point surrounding the array. The sensor geometry was constrained to having four sensors along the boundaries of the trawl door. An exhaustive search was performed by simulating sensors in positions along a 0.01 m grid around the trawl door. The sensor geometry with the lowest error was mounted on a small trawl door, and deployed in the New Zealand trial as described in Section 4.5.

Finally, the performance in the presence of noise was improved by analyzing different delay estimation techniques. In Section 2.4, the accuracy of the delay estimation using different techniques was evaluated at different SNRs to determine which technique should be used to minimize the estimation error. Furthermore, in Section 2.4.1, the input signal was linearly interpolated to further reduce the estimation error. This analysis indicated that combining the classic cross-correlation with interpolation leads to the minimum delay estimation error. However, this increases the complexity of the calculation and increases the memory required, which reduces the number of estimations that can be performed in real-time. To balance computation time with delay estimation error, the classic cross-correlation combined with the Hilbert transform was implemented, which was the best technique that did not require interpolation.

The implemented system was tested in various different environments to verify the capabilities of the system and to analyze the estimation accuracy. The system was first evaluated in the Dalhousie Aquatron to ensure proper operation of the main systems in an underwater environment, mainly the controller program that communicated with the sensor array using the Trac protocol suite as described in Section 4.1. After this initial trial, another trial was performed using a 5-element sensor platform on the berm near the Fundy Ocean Research Centre for Energy (FORCE) VC just outside Parrsboro as described in Section 4.2. Next, the system was tested in a sea trial in Herring Cove as described in Section 4.3 to evaluate the mean estimation error and provided insight into the limitations of a planar array. Then, the system was tested in a long-term underwater deployment off the coast of Parrsboro as described in Section 4.4.

After performing the controlled testing in the Dalhousie Aquatron and using an artificial sound source in Herring Cove, the system was deployed for a series of trials in New Zealand. During the one-month deployment, several trials occurred as described in Section 4.5 using sensors attached to both the starboard and port trawl doors. During the analysis of these trials, the time delays between sensors were measured using the classic cross-correlation using the Hilbert transform. Then, the positions corresponding to these delay sets were determined by querying the pre-computed hyperoctree for the given sensor geometry. Finally, the positions were visualized using a scatter plot showing the horizontal position estimate of the Hector’s dolphins, as well as a scatter plot showing the estimated inclination of the dolphin. This showed that while the dolphin approached the array horizontally, it also approached the water surface, avoiding the sensor array on the seabed.

Throughout the completion of this thesis, a real-time acoustic underwater localization system was developed using a compact sensor array. This sensor geometry was improved upon by developing and using a set of figures of merit that allowed to assess the accuracy of the localization algorithm. Additionally, the base TDoA algorithm was optimized to allow for real-time operation of the localization system. This system was tested in a variety of different environments before ultimately being used in a long-term deployment in New Zealand. Throughout the New Zealand deployment, the behaviour of Hector’s dolphins was observed using the localization system to determine how they were impacted during trawling.

5.2 Future Work

This thesis covered a system for performing underwater localization in real-time using the TDoA and the LCMFP algorithms. Potential future work could be done to investigate if any other localization algorithm could improve upon TDoA while not being as sensitive as LCMFP to changing or uncertain environmental conditions. Additionally, different pre-processing techniques could be done to better estimate the signal delays between sensors, especially in low SNR or extreme multipath environments. Furthermore, the localization system could be more dynamic by using different pre-processing techniques depending on the circumstances and available time budget. In the case of a rapid click train, the system may opt to forego interpolation in order to maintain real-time estimation. Conversely, higher levels of interpolation could be used in circumstances with infrequent clicks to enable a more accurate position estimation.

Additionally, further work could be done to investigate more complicated static or even changing sensor geometry. For instance, the sensor geometry could adapt based on an original rudimentary estimation to provide for a more accurate final estimation of the location of the sound source. Alternatively, multi-path could be exploited by placing well-placed reflective surfaces near the sensor array to gain additional information about the position of the sound source.

Finally, the localization system could be directly integrated with the underwater sensor array. This would greatly reduce the amount of data transferred between an underwater remote platform and the control computer on land. Additionally, it would simplify the handshaking between different systems and remove the communication overhead required by the Trac suite of protocols.

Bibliography

- [1] M. Jahanbakht, W. Xiang, L. Hanzo, and M. R. Azghadi, “Internet of Underwater Things and Big Marine Data Analytics—A Comprehensive Survey,” *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 904–956, 2021.
- [2] J. Macaulay, J. Gordon, D. Gillespie, C. Malinka, and S. Northridge, “Passive acoustic methods for fine-scale tracking of harbour porpoises in tidal rapids,” *The Journal of the Acoustical Society of America*, vol. 141, no. 2, pp. 1120–1132, 2017.
- [3] D. Gillespie *et al.*, “Harbour Porpoises Exhibit Localized Evasion of a Tidal Turbine,” *Aquatic Conservation: Marine and Freshwater Ecosystems*, vol. 31, no. 9, pp. 2459–2468, 2021. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/aqc.3660>
- [4] L. Kyhn, F. Jensen, K. Beedholm, J. Tougaard, M. Hansen, and P. Madsen, “Echolocation in sympatric Peale’s dolphins (*Lagenorhynchus australis*) and Commerson’s dolphins (*Cephalorhynchus commersonii*) producing narrow-band high-frequency clicks,” *Journal of Experimental Biology*, vol. 213, no. 11, pp. 1940–1949, 2010.
- [5] P. Sørensen, D. M. Wisniewska, F. H. Jensen, M. Johnson, J. Teilmann, and P. T. Madsen, “Click communication in wild harbour porpoises (*phocoena phocoena*),” *Scientific Reports (Nature Publisher Group)*, vol. 8, pp. 1–11, Jun. 2018.
- [6] K. T. Clausen, M. Wahlberg, K. Beedholm, S. Deruiter, and P. T. Madsen, “Click communication in harbour porpoises *phocoena phocoena*,” *Bioacoustics*, vol. 20, no. 1, pp. 1–28, 2011. [Online]. Available: <https://doi.org/10.1080/09524622.2011.9753630>
- [7] W. W. L. Au, R. A. Kastelein, T. Rippe, and N. M. Schooneman, “Transmission beam pattern and echolocation signals of a harbor porpoise (*phocoena phocoena*),” *The Journal of the Acoustical Society of America*, vol. 106, no. 6, pp. 3699–3705, 1999. [Online]. Available: <https://doi.org/10.1121/1.428221>
- [8] J. F. Valente and J. C. Alves, “Real-time TDOA measurements of an underwater acoustic source,” in *OCEANS 2016 MTS/IEEE Monterey*, 2016, pp. 1–7.
- [9] I. Urazghildiiev, B. Martin, and D. Hannay, “The accuracy of bearing estimates of wideband signals produced by marine animals,” *IEEE Journal of Oceanic Engineering*, vol. 46, no. 3, pp. 1057–1067, 2021.
- [10] Y. Zhou, W. Xu, H. Zhao, and N. R. Chapman, “Improving Statistical Robustness of Matched-Field Source Localization via General-Rank Covariance Matrix Matching,” *IEEE Journal of Oceanic Engineering*, vol. 41, no. 2, pp. 395–407, Apr. 2016. [Online]. Available: <http://ieeexplore.ieee.org/document/7128421/>

- [11] X. Zhang, E. Song, J. Huang, H. Liu, Y. Wang, B. Li, and X. Yuan, "Acoustic Source Localization via Subspace Based Method Using Small Aperture MEMS Arrays," *Journal of Sensors*, vol. 2014, pp. 1–14, 2014. [Online]. Available: <https://www.hindawi.com/journals/js/2014/675726/>
- [12] X. Li, Z. D. Deng, L. T. Rauchenstein, and T. J. Carlson, "Contributed Review: Source-localization algorithms and applications using time of arrival and time difference of arrival measurements," *Review of Scientific Instruments*, vol. 87, no. 4, p. 041502, Apr. 2016. [Online]. Available: <http://aip.scitation.org/doi/10.1063/1.4947001>
- [13] J.-b. Zhao, Y.-x. Liu, C.-j. Liu, and Y. Ling, "A three-dimensional matching localization algorithm based on helix triangular pyramid array," *Journal of Central South University*, vol. 28, no. 3, pp. 816–833, Mar. 2021. [Online]. Available: <https://link.springer.com/10.1007/s11771-021-4647-4>
- [14] D. Zhao, W. Seong, K. Lee, and Z. Huang, "Shallow water source localization using a mobile short horizontal array," *Journal of Systems Engineering and Electronics*, vol. 24, no. 5, pp. 749–760, 2013.
- [15] Z.-H. Michalopoulou and M. Porter, "Matched-field processing for broad-band source localization," *IEEE Journal of Oceanic Engineering*, vol. 21, no. 4, pp. 384–392, 1996.
- [16] G. Fabregat *et al.*, "Design and Implementation of Acoustic Source Localization on a Low-cost IoT Edge Platform," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 67, no. 12, pp. 3547–3551, 2020.
- [17] D. Halupka *et al.*, "Low-Power Dual-microphone Speech Enhancement using Field Programmable Gate Arrays," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3526–3535, 2007.
- [18] C. O. Tiemann, A. M. Thode, J. Straley, V. O'Connell, and K. Folkert, "Three-dimensional localization of sperm whales using a single hydrophone," *The Journal of the Acoustical Society of America*, vol. 120, no. 4, pp. 2355–2365, 2006. [Online]. Available: <https://doi.org/10.1121/1.2335577>
- [19] J. Gebbie, M. Siderius, R. McCargar, J. S. Allen, and G. Pusey, "Localization of a noisy broadband surface target using time differences of multipath arrivals," *The Journal of the Acoustical Society of America*, vol. 134, no. 1, pp. EL77–EL83, 2013. [Online]. Available: <https://doi.org/10.1121/1.4809771>
- [20] M. Gassmann, S. M. Wiggins, and J. A. Hildebrand, "Three-dimensional tracking of cuvier's beaked whales' echolocation sounds using nested hydrophone arrays," *The Journal of the Acoustical Society of America*, vol. 138, no. 4, pp. 2483–2494, 2015. [Online]. Available: <https://doi.org/10.1121/1.4927417>
- [21] J. McEachern, E. Malekshahi, and J.-F. Bousquet, "Passive Localization Algorithm using a Highly Integrated Acoustic Sensor Array," in *2022 20th IEEE Interregional NEWCAS Conference (NEWCAS)*, Jun. 2022, pp. 397–401.

- [22] J. McEachern, J.-F. Bousquet, S. Kiehbardroudinezhad, and B. Martin, “Harbour porpoise localization system using compact acoustic sensor arrays,” in *Proceedings of the 16th International Conference on Underwater Networks & Systems*, ser. WUWNet '22. New York, NY, USA: Association for Computing Machinery, 2022. [Online]. Available: <https://doi.org/10.1145/3567600.3568151>
- [23] C. Knapp and G. Carter, “The generalized correlation method for estimation of time delay,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 4, pp. 320–327, 1976.
- [24] D. Meagher, “Octree encoding: A new technique for the representation, manipulation and display of arbitrary 3-d objects by computer,” Rensselaer Polytechnic Institute, Troy, NY, Tech. Rep., 1980.
- [25] S. Singh, M. Crispo, J.-F. Bousquet, and S. Aljendi, “A Janus Compatible Software-defined Underwater Acoustic Multiple-input Multiple-output Modem,” *Intl. J. Distributed Sensor Networks*, vol. 17, no. 4, 2021. [Online]. Available: <https://doi.org/10.1177/15501477211010663>
- [26] M. Mirzaei Hotkani *et al.*, “Underwater Target Localization Using Opportunistic Ship Noise Recorded on a Compact Hydrophone Array,” *Acoustics*, vol. 3, no. 4, pp. 611–629, 2021. [Online]. Available: <https://www.mdpi.com/2624-599X/3/4/39>
- [27] J. Benesty, J. Chen, and Y. Huang, *Microphone Array Signal Processing*. Berlin: Springer, 2008.
- [28] F. Jensen *et al.*, “Computational ocean acoustics,” *The Journal of the Acoustical Society of America*, vol. 97, no. 5, pp. 3213–3213, 1995. [Online]. Available: <https://doi.org/10.1121/1.411832>

Appendix A

Conferences and Publications

A.1 Annual Electrical and Computer Engineering Graduate Conference

A portion of the work done in this thesis was presented at the 2022 Annual Electrical and Computer Engineering Graduate Conference. This presentation was titled “A real-time Acoustic Localization System on an Embedded Processor” and contained information on the background and problem statement of the thesis, details of the TDoA algorithm, details of the implementation, plans for deployment, and challenges faced converting the MATLAB model to custom hardware. This presentation was awarded the 2nd Place Prize, Masters Award.

A.2 20th IEEE Interregional NEWCAS Conference

A portion of the work done in this thesis was presented at the 20th IEEE Interregional NEWCAS Conference in Québec City, Canada. This work was published as:

J. McEachern, E. Malekshahi, and J.-F. Bousquet, “Passive Localization Algorithm using a Highly Integrated Acoustic Sensor Array,” in *2022 20th IEEE Interregional NEWCAS Conference (NEWCAS)*, Jun. 2022, pp. 397–401.


A.2.1 CMC Microsystems TEXPO 2022 Graduate Student Research Competition

The work presented in Section A.2 was also demonstrated at the CMC Microsystems TEXPO 2022 Graduate Student Research Competition. In this competition, the poster shown in Figure A.1 was presented that details the motivation, the objectives, and the calculations used for the LCMFP algorithm. Additionally, it explains the SoC architecture, SoC implementation, and SoC results. Finally, it shows the results of a simulation and ends with concluding remarks.

A.3 16th International Conference on Underwater Networks & Systems

A portion of the work done in this thesis was presented at the 16th International Conference on Underwater Networks & Systems in Boston, United States of America. This work was published as:



J. McEachern, J.-F. Bousquet, S. Kiehbardroulinezhad, and B. Martin, “Harbour porpoise localization system using compact acoustic sensor arrays,” in *Proceedings of the 16th International Conference on Underwater Networks & Systems*, ser. WUWNet '22. New York, NY, USA: Association for Computing Machinery, 2022. [Online]. Available: <https://dl.acm.org/doi/10.1145/3567600.3568151>



DALHOUSIE UNIVERSITY
Inspiring Minds

Passive Localization Algorithm using a Highly Integrated Acoustic Sensor Array

Jordin McEachern, Ehsan Malekshahi, Jean-François Bousquet
Electrical and Computer Engineering Department, Dalhousie University

UW-STREAM
Underwater Sensing and Transmission using Electro-Acoustic Modems

Motivation

There is significant interest to increase our access to the Earth's oceans, to discover and exploit its resources in a sustainable fashion. To monitor the environment, instruments are moored at the bottom of the ocean and are mounted with heterogeneous sensors. The choice of sensor array size depends on the available space, the power consumption, and the data processing capability. In this work, a remote acoustic sensor array is proposed to detect and localize marine mammals.

Research Objectives

- Develop a small form-factor, low-power acoustic sensor for harbour porpoise localization.
- Compute in real-time the localization algorithm on a System-on-Chip (SoC).
- Implement the peripherals in VHDL and develop the controller in the PetaLinux operating system.
- Compare the real-time algorithm with a MATLAB simulation to demonstrate the potential for the remote platform to reliably detect marine mammals.
- Evaluate and optimize the resource and power requirements of the implemented SoC design.

System-on-Chip Architecture

- On the Zybo Z7-20 combining ARM Cortex-A9 Processor with a Xilinx Artix-7 FPGA.
- Three main layers to create a real-time localization system:
 - **Communications:** algorithm parameters through UART interface and channel measurements through ethernet interface.
 - **Zynq Processor:** software used to drive the FPGA and provide updates to the map display.
 - **VHDL Peripherals:** hardware peripherals used to perform localization, including the propagation model of the environment.




Fig. 5: Deployed Five-element Platform

Localization Algorithm Calculations

The localization algorithm is implemented using the following three equations:

1. Propagation model which is the superposition of the two arrivals of the signal s with different gains α and arrival times τ .

$$r_n(t) = \alpha_0(p_0, p_n) \cdot s(t - \tau_{0,n}(p_0)) + \alpha_1(p_0, p_n) \cdot s(t - \tau_{1,n}(p_0))$$
2. The cross-correlation matrix between channel estimated transmit signal and a hypothetical source at test position p .

$$C_c(p) = \frac{1}{L} \sum_{k=0}^{L-1} (\bar{s}^T(t_k, p) \bar{s}(t_k, p))$$
3. Estimate source position \hat{p}_0 by determining the test source position p which leads to the lowest determinant of the cross-correlation matrix C_c .

$$\hat{p}_0 = \arg \min_p \{ \det\{C_c(p)\} \}$$

System-on-Chip Implementation

- Uses fixed-point and floating-point representation depending on the expected scale of operations.
- Uses the Coordinate Rotation Digital Computer (CORDIC) and Fast Fourier Transform (FFT) IP-Cores provided by Xilinx to obtain the phase relationship (arctangent) between signals.

System-on-Chip Results

- Accuracy of the implemented SoC was evaluated by comparing the ambiguity function obtained using the FPGA and a MATLAB simulation.




Fig. 1: Five-element Acoustic Sensor Array Recorder




Fig. 2: Harbour Porpoise

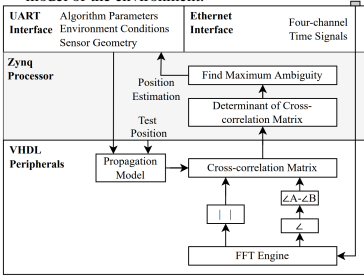


Fig. 3: System-on-Chip Architecture

Localization Algorithm Results

- Proposed algorithm compared to standard Time Difference of Arrival (TDoA) using a simulation.

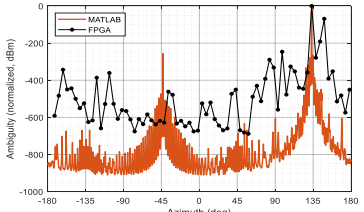


Fig. 4: Ambiguity Function of the Real-Time System

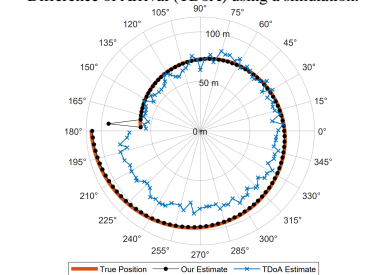


Fig. 6: TDoA vs. Proposed Algorithm Estimation Results

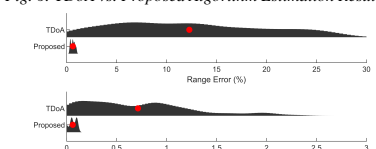


Fig. 7: TDoA vs. Proposed Algorithm Error Distribution

Conclusion

- Accuracy of the implemented SoC was evaluated by comparing the ambiguity function obtained using the FPGA and a MATLAB simulation.
- System runs in real-time using four sensors, with an additional sensor for added redundancy.

Special thanks to Fundy Ocean Research Centre of Energy

1 TEXPO 2022 | June 20, 2022 | Jordin McEachern




Figure A.1: Poster presented during the TEXPO 2022 Graduate Student Research Competition.

Appendix B

Software and Tools

The following software and tools were utilized during the completion of this thesis:

- **latlon23** Python library used to convert between polar coordinates and geographic coordinates (latitude and longitude) used in the porpoise map program.
- **MATLAB** Throughout the design phase of this thesis, MATLAB played a crucial role that allowed for rapid prototyping. MATLAB continued to be used for simulations and data analysis. It was also used to verify the implemented design matched with the expected results obtained through MATLAB. Additionally, MATLAB was used for most of the data visualization, including graphs and spectrograms.
- **pyserial** Python library used to communicate between the host computer running the porpoise map program and the Zybo development board.
- **SOLIDWORKS** 3D modelling software which was used to model the Dalhousie Aquatron trial structure.
- **Tkinter** Python library that provides bindings to the Toolkit (Tk) GUI toolkit which was used to create the porpoise map program.
- **Xilinx Vitis** Unified Software Platform used to develop the driver for the custom hardware before transitioning to run under Petalinux.
- **Xilinx Vivado** Design suite used to design and test custom hardware design description for the LCMFP algorithm.

Additionally, these programs were created and compiled using gcc (GNU's Not Unix! (GNU) Compiler Collection) and run on Petalinux to perform the real-time localization:

- **tdoa-delay-finder** Program used to read a WAVE file and run a peak-detection algorithm in combination with finding the delays between channels.

- **tdoa-delays-to-position** Program used to convert a delay set into a final position estimation by using the hyperoctree structure.
- **trac-receiver** Program used to communicate with the sensor platform using the Trac protocol suite.

Finally, the following programs were created and compiled using gcc and ran on a host computer:

- **geometry-analysis** Program used to analyze the specified sensor geometry to determine the mean estimation error of the geometry.
- **tdoa-precompute** Program used to analyze a search grid using the specified sensor geometry and saves the LUT used to convert between a delay set and a position estimation.
- **wave-gen** Program used to generate simulation data of ideal porpoise-like clicks on multiple channels. Configurable to use either just the direct LOS path or contain additional surface and sea bed bounces. Can also be configured to add AWGN to the signal.