EXPLOITING AMBIENT NOISE FOR COHERENT PROCESSING OF MOBILE VECTOR SENSOR ARRAYS

A Dissertation
Presented to
The Academic Faculty

by

Brendan Nichols

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in
George W. Woodruff School of Mechanical Engineering

Georgia Institute of Technology

May 2018

EXPLOITING AMBIENT NOISE FOR COHERENT PROCESSING OF MOBILE VECTOR SENSOR ARRAYS

Approved by:

Dr. Karim Sabra, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Justin Romberg
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Julien Meaud
School of Mechanical Engineering
Georgia Institute of Technology

David Trivett
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Costas Arvanitis
School of Mechanical Engineering
Georgia Institute of Technology

Date Approved: March 15, 2018

ACKNOWLEDGEMENTS

I would like to firstly acknowledge my thesis advisor, Dr. Karim Sabra for his continued guidance and support. His knowledge and insight into the fundamentals of the research aided in its progression, even when it appeared it had reached a dead end. In addition, I would like to thank him for the financial support through the many years of research including travel funding for data collection trips and conferences, which greatly improved my learning experience.

In the same vein, I would like to thank Dr. Mike Traweek from the Office of Naval Research, who graciously provided funding through Code 321-MS. Without his continued support, this work could not have been realized in the first place.

Furthermore, I would like to thank Dave Trivett and Jim Martin for their instrumental efforts in developing a data collection platform. The design, funding, purchasing, manufacturing, bench testing and field experiments were all made possible by them. Jim Martin was also very helpful in data collection, analysis and troubleshooting efforts required for developing such a data collection system.

Data collection efforts were also aided by Lt. Chris Verlinden of the Coast Guard Academy, who provided the research vessel free of charge. Furthermore, the Coast Guard Academy cadets and staff were invaluable during the operation of the vessel and deployment of the array.

I would like to thank my thesis committee for their continued time: Dr. Julien Meaud, Dr. Costas Arvanitis, Dr. Justin Romberg and Dave Trivett. Their questions and input during the proposal and defense were very helpful and insightful.

Lastly, I would like to acknowledge my family and friends for their support. My wife has been supportive and helpful in keeping me motivated, focused and happy, without which I could not have completed this work. My family also deserves recognition for their support, and for the many privileges they have provided me. Finally, I would like to thank my friends for being a distraction from work when I needed one, and helpful listeners when the occasional problem just needed to be discussed to be worked out.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

AEL	Array Element Localization
AESL	Array Element Self-Localization (see AEL)
AESS	Array Element Self-Synchronization
AIS	Automatic Identification System
ARAP	Augmented Reliable Acoustic Path
DC	Direct Current
DIFAR	Directional Frequency Analysis and Recording
DOA	Direction of Arrival
FFT	Fast Fourier Transform
GA	Genetic Algorithm
GB	Gigabyte
GPS	Global Positioning System
GPSDO	GPS-Disciplined Oscillator
Hz	Hertz
IFFT	Inverse Fast Fourier Transform
IMU	Inertial Measurement Unit
LBL	Long Baseline
LDV	Laser Doppler Vibrometer
LTI	Linear Time-Invariant
NAS	Non-Acoustic Sensor
NED	North-East-Down (coordinate system)
OCXO	Oven-Controlled Crystal Oscillator
PSD	Power Spectral Density
PSNR	Peak Signal-to-Noise Ratio
RANSAC	Random Sample Consensus
RMS	Root Mean Square
RMSE	Root Mean Square Error
R/V	Research Vessel

S/PDIF	Sony/Philips Digital Interface Format
SBL	Short Baseline
SCS	Sensor Coordinate System
SD	Secure Digital (card)
SNR	Signal-to-Noise Ratio
TBWP	Time Bandwidth Product
TDMA	Time-Division Multiple Access
TDOA	Time-Difference of Arrival
VHF	Very High Frequency (radio waves)

SUMMARY

A network of mobile sensors, such as vector sensors mounted to drifting floats, can be used as an array for locating acoustic sources in an ocean environment. Accurate localization using coherent processing on such an array dictates the locations of sensor elements must be well-known. In many cases, a mobile, submerged array cannot meet this requirement, however the presence of ambient acoustic noise provides an opportunity to correct sensor location errors. It has been previously shown that ambient noise correlations across separated, fixed hydrophones can provide the separation distance between them (K. G. Sabra et al., 2005, IEEE J. Ocean Engineering, Vol. 30). A time-domain framework for this method is presented for the case of vector sensors in isotropic ambient noise to quantify their gain relative to traditional hydrophone correlations. Furthermore, a novel method is presented for identifying hidden ambient noise correlation peaks when the separation distance is changing, and its accuracy is found to match that of GPS. Lastly, a novel weighted coherent processing algorithm is presented and its performance compared to traditional methods, finding increased localization precision even in the presence of severe noise. This method is applied to locating a source, and succeeds using both GPS and ambient-noise-corrected sensor locations. All experimental data used in these studies were collected from a novel vector sensor array, and details of its design and deployment are presented as well.

CHAPTER 1
INTRODUCTION

Locating objects in the ocean is a difficult task due to the limited depth at which light penetrates and propagates. Even in clear water, objects beyond a hundred meters or so cannot be "seen" in the traditional sense, and in deep or turbid water, visibility may be quasi-null. Fortunately, sound propagates with much less attenuation underwater, and in ideal conditions, it can propagate for hundreds, if not thousands, of kilometers (Worcester et al., 1999). Locating and tracking objects thus relies on detecting the sounds emitted by an object. It has applications ranging from naval security (J. C. A. Shipps, B.M., 2004) to wildlife monitoring (Thode et al., 2010), among others.

To obtain the direction towards a source, multiple sensors can be placed in the environment at different locations. The difference in arrival time of a sound at each sensor gives information on the location of its source through a process called beamforming (or array signal processing) (Van Trees, 2004). As part of this process, the locations of each sensor are required to compare the time-difference-of-arrival (TDOA) of the recorded sound with the theoretical TDOA. If the sensor locations are not accurately known, the localization performance suffers (Culver & Hodgkiss, 1988; S. E. Dosso, Collison, Heard, & Verrall, 2004).

However, if passive and covert operations are desired, obtaining accurate sensor locations in an ocean environment can prove difficult. Global positioning system (GPS) signals do not propagate underwater; acoustic positioning systems are non-covert and limit sensor mobility; and dead-reckoning accumulates errors much too large for source

localization (Paull, Saeedi, Seto, & Li, 2014). However, there exists an additional source of information which can be exploited to correct the errors inherent in dead-reckoning.

Ambient noise exists in every ocean environment to some degree, from low frequency seismic noise (Woolfe, Lani, Sabra, & Kuperman, 2015) to high frequency wind-generated surface noise (Brooks & Gerstoft, 2009). It has been shown that correlations of ambient noise across separated sensors results in the Green's function (or channel impulse response) between them (Roux & Kuperman, 2004). Knowing the propagation speed, the direct-arrival peak of the Green's function results in the separation distance between sensors. This peak forms as a result of the ambient noise sources which reside in a narrow endfire beam formed by the baseline of the two sensors (Roux & Kuperman, 2004). The peak can thus be enhanced by beamforming the ambient noise recordings to isolate only contributing sources (Leroy et al., 2012).

An acoustic vector sensor, which measures both pressure and acoustic particle velocity, functions as a single sensor with multiple co-located components. As a result, a single sensor is able to be steered, i.e. it has inherent directionality (Nehorai & Paldi, 1994). Its directionality could be exploited to improve the emergence of an ambient noise correlation peak. This becomes important when the allowable observation length is limited by sensors which are drifting, thus changing the environment as a measurement is being made (Woolfe, Sabra, & Kuperman, 2015). In extreme cases, changes in the environment occur so rapidly that the Green's function peak has yet to emerge from correlation noise. A stochastic search algorithm can attempt to locate hidden peaks in many short observations of the ambient noise to obtain the environmental changes on a short time scale (Woolfe, Sabra, et al., 2015). Using this information, the distance between any pair of

sensors is obtained, and knowledge of the distance between all pairs of sensors can correct the sensor locations (S. E. Dosso et al., 2004). Correcting the sensor locations then results in better localization performance.

The main contributions of this work are to: analyze the performance of vector sensors for ambient noise correlations in relation to traditional hydrophone correlations; present a stochastic search methodology for identifying ambient noise correlation peaks hidden in noise; analyze the effects of noise and sensor positional errors in vector sensor source localization; and demonstrate the improvement of source localization using ambient-noise-corrected sensor locations. Lastly, a new free-floating vector sensor array was developed to obtain experimental data used in the above-mentioned analyses.

Chapter 2 details the design and deployment of the free-floating vector sensor array used to collect experimental data for later chapters. The data preprocessing steps are also presented here, since many of them are used identically in later chapters.

Chapter 3 presents the theory behind the emergence of the Green's function from ambient noise correlations. It introduces a novel time-domain interpretation of vector sensor ambient noise correlations. Furthermore, it analyzes the performance of vector sensor correlations relative to standard hydrophone correlations using theoretical and experimental results.

Chapter 4 presents the theory of identifying Green's function peaks from ambient noise correlations when the medium is rapidly varying. A novel stochastic search method is tested on simulated and experimental data to demonstrate its performance in measuring sensor separation distances. In addition, Chapter 4 presents the theory of correcting sensor

locations using distances measured in this way, commenting on each method's ability to obtain valid results.

Chapter 5 defines a standard vector sensor beamforming methodology, adding a novel weighting method to improve localization performance. The performance of this method and traditional methods is compared in the presence of sensor noise and location errors, for both simulated and experimental data. Lastly, the improvement of source localization is quantified when using sensor locations corrected using the stochastic search method described in Chapter 4.

CHAPTER 2
DATA COLLECTION

2.1 Introduction

Obtaining experimental data is a critical component in effectively testing proposed methods of vector sensor ambient noise array element localization and source localization. The experimental data will serve to reinforce simulated findings as well as provide a foundational proof-of-concept for the methods studied in later chapters. Without experimental validation, it is difficult to draw conclusions on the functionality or efficacy of results drawn from simulated methods.

The desired dataset for testing ambient noise array element localization (AEL) contains synchronized recordings of ocean ambient noise on multiple vector sensors whose separation distances are changing at different rates. The separation distances must also be small (on the order of tens to hundreds of meters). Ideally, the exact locations of the sensors are also measured to have a ground-truth to which the ambient noise localization can be compared. Furthermore, there should be few interfering sources present, and the locations of any sources in the area should at least be well-known. For testing source localization, the separation distances should be larger than the ambient noise case (on the order of hundreds to thousands of meters) if absolute localization is required (range and bearing). However, if sharing data with the ambient noise array shape, only bearing can be effectively obtained (Nichols & Sabra, 2015). Obviously, the source to be localized should also have a known position as a ground-truth. In addition, it would be beneficial if the source traverses a large area of possible locations around the array to test localization in a variety of scenarios.

To date, the only experimental data which verifies ambient noise correlation Green's function extraction uses hydrophones or hydrophone arrays. Roux et al. demonstrated experimental success in extracting Green's function information from fixed hydrophones in a 100 – 300 Hz band (Roux & Kuperman, 2004). In 2005, experimental work by Sabra et al. demonstrated the technique multiple times and over a wider band, and showed it could be used for array element self-synchronization (AESS) and array element self-localization (AESL) (Sabra, Roux, & Kuperman, 2005a, 2005b; K. G. Sabra et al., 2005). A year later, Siderius et al. demonstrated the technique for passive fathometry and sub-bottom profiling, which also included drifting hydrophone arrays (Martin Siderius, Harrison, & Porter, 2006). Further experimental results were given by Harrison and Siderius in 2008 (Harrison & Siderius). An important finding in these works by Siderius et al. is that beam-steered correlations can aid in the emergence of a coherent peak, thus reducing the required averaging time. In addition, the direction of sensor drift was perpendicular to the ambient noise propagation direction (i.e. the changes in environment were solely due to changing bathymetry beneath the sensors). As a result, they did not encounter the scenario where the averaging duration was limited by the drift rate.

Two other papers applied ambient noise correlations for Green's function extraction in 2008 (Brooks & Gerstoft, 2009; Fried, Kuperman, Sabra, & Roux, 2008), both corroborating the technique for bottom-mounted hydrophones in different bands and using different noise sources. Brooks and Gerstoft used a tropical storm as the ambient noise source, whereas Fried et al. used distributed biologic noises from the croaker fish (*Sciaenidae*) family. Later in 2010, Siderius et al. improved emergence of the Green's function for fathometry using adaptive processing, further reinforcing the notion that

selective use of ambient noise sources leads to improved extraction (M. Siderius et al.). Around the same time, Godin et al. demonstrated passive ocean tomography using bottom-mounted hydrophone arrays, estimating the sound speed profile through pair-wise correlations across the arrays (Godin, Zabotin, & Goncharov, 2010).

Beamformed hydrophone ambient noise correlations were again shown to improve Green's function extraction in 2012 and 2013 by Leroy et al. and Lani et al., both using data from bottom-mounted vertical line arrays in the presence of distributed shipping noise (Lani, Sabra, Hodgkiss, Kuperman, & Roux, 2013; Leroy et al., 2012). In addition to using beamformed ambient noise correlations, Woolfe et al. demonstrated the approach for measuring changing sound speed between hydrophone triads separated by over 100 km (Woolfe, Lani, et al., 2015a). In this work, the rate at which sound speed varies constrains the allowable averaging duration, but a sufficient averaging duration was achieved to obtain a coherent correlation peak. This limitation was again introduced in Woolfe and Sabra 2015 (Woolfe & Sabra, 2015b), and later a proof-of-concept formulation of the stochastic search method was proposed (Woolfe, Sabra, et al., 2015c). The experimental data used in these studies came from moored hydrophone triads recording in the 1 – 40 Hz band in the presence of ice noise, shipping noise, and seismic noise. The experimental data used for validating the stochastic search method still maintained a distinct correlation peak, and simulated additive noise was introduced to artificially decrease the signal-to-noise ratio (SNR). However, it proved the concept of using a stochastic search on ambient noise correlations to enable shorter averaging times, thus capturing faster environmental changes.

The final experimental test which most closely aligns with the goals of this work was presented in 2016 by Naughton et al. (Naughton et al., 2016). Ambient noise correlations were obtained for a freely-drifting hydrophone array, where the distance between sensors varied over the course of the experiment. It was demonstrated that occasional coherent peaks could be obtained even for the short (~10 second) averaging time allowed by the drifting sensors. However, the most likely reason for the emergence of these peaks was the presence of endfire sources (i.e. shipping noise), which was confirmed by coherently beamforming the array to determine the direction of the dominant energy. In addition, the work used a peak-detection algorithm to identify the sensor separation distances, which works well when the ambient correlation yields a strong peak. However, over half of the peak locations were inaccurate by more than 10 ms (~15 meters assuming 1500 m/s sound speed). This indicates a large number of correlations did not produce a strong correlation peak for the rather short averaging time dictated by the sensor drift rate.

From the past experimental tests, it is clear that novel experimental data would enable ambient noise correlations on drifting sensors, where the drift rate limits the allowable averaging time for the emergence of coherent peaks (e.g. such a constraint was experienced in the experimental data of Naughton et al. (Naughton et al., 2016)). In effect, the emergence of coherent peaks should *not* occur, and a technique such as that employed in Woolfe et al. (Woolfe, Sabra, et al., 2015c) could be applied to estimate the distances between sensors. A benefit of estimating inter-sensor distance is that it enables a measurable reference (i.e. GPS locations) to corroborate the stochastic search findings, whereas the experimental data in Woolfe et al. required the varying temperature to be

measured with the noise correlation method itself, or alternatively rely on imprecise external data sources (e.g. Argo (2000)).

In addition to drifting sensor data, an area not yet studied experimentally involves the use of vector sensors. It was shown that beamformed ambient noise correlations improve Green's function emergence from ambient noise correlations (Harrison & Siderius, 2008; Lani et al., 2013; Leroy et al., 2012; Martin Siderius et al., 2006; M. Siderius et al., 2010). Similarly, a vector sensor provides directionality through the weighting of its components (see Figure 2), and could improve upon the existing body of hydrophone research. In addition to improving ambient noise correlations, a mobile/drifting vector sensor array could also yield some insights into source localization not yet explored experimentally.

Previous experiments utilizing vector sensors began as far back as the 1980's where a neutrally-buoyant, freely drifting array of infrasonic (1 – 10 Hz) floats were designed and deployed (G. L. D'Spain et al., 1992; Hodgkiss & Anderson, 1983). The first localization efforts demonstrated intensity processing and conventional beamforming on single sensors (G. L. D'Spain et al., 1992), and later more comprehensive theory was developed (Nehorai & Paldi, 1994), and even improved (Hawkes & Nehorai, 1998, 1999, 2000, 2001, 2003). However, experimental data was generally limited to incoherent processing of single sensors, e.g. directional frequency and recording (DIFAR) sensors (Greene et al., 2004; Swartz, 2003). Smith and van Leijen later applied traditional cardioid and novel non-linear hippioid processing to DIFAR buoy data (Smith & van Leijen, 2007). The first true vector sensor array processing experimental data was presented in 2009 by Poulsen, where a 19-element towed vector sensor array was deployed in sea trials during 2006 and 2007

(Poulsen, 2009). Further experimental data was obtained in 2010 by Felisberto et al, Santos et al, and Thode et al. using a towed vector sensor array for tracking whales (Felisberto, Santos, & Jesus, 2010; Santos, Felisberto, & Jesus, 2010; Thode et al., 2010). To date, no distributed vector sensor array has been deployed and coherently processed, as the positional accuracies and temporal synchrony required poses a formidable challenge in an ocean environment.

As such, novel experimental data would involve a distributed, mobile vector sensor array where the sensor elements are synchronized and their locations known. Not discussed in this thesis is a large body of work undertaken by Dave Trivett, James Martin, Dr. Karim Sabra, Dr. Kevin Smith, and many others to mount vector sensors to autonomous littoral gliders (Alaska Native Technologies, LLC) in order to obtain such a dataset. Issues with clock synchrony, glider self-noise, navigation, and dead-reckoning accuracy prevented such efforts from yielding an ideal dataset. However, data from two surfaced gliders (effectively floats) was obtained in 2012, such that GPS could be used to locate the sensors and the self-noise and navigation were not an issue. Clock synchrony was also not guaranteed, but proved to be sufficient for coherent processing (Nichols & Sabra, 2015). Many of the lessons learned from partaking in these experiments have guided the design and development of the free-floating vector sensor array which is described in this chapter. Section 2.2 describes the array design, such as the hardware elements and their functionality. Section 2.3 describes the data collection and processing steps required to obtain usable data from the array. Furthermore, it details the data collection efforts over five deployments between 2015 and 2016, summarizing the lessons learned and usable data collected from each.

2.2 Array Design

2.2.1 Vector Sensors

The vector sensor available for this array is the Wilcoxon VS-301 (see Figure 1), a compact sensor which measures acoustic pressure and particle velocity using a hydrophone and three-axis accelerometer co-located within the sensor body (Meggitt). Since it measures sensor acceleration to obtain fluid particle velocity, it must be mounted with a compliant suspension to allow it to freely move in response to the surrounding fluid motion (see Figure 11). In addition, the acceleration channels will require integration to obtain velocity. Both will limit the lower usable frequency of the sensor, the suspension limit is due to its resonance frequency and the integration limit is due to $1/j\omega$ noise amplification at low frequencies. The sensor is also sensitive to flow noise (Lauchle, Wang, & Howe, 2002), which can be especially prevalent at low frequencies. In practice, the lower usable frequency will be on the order of tens to hundreds of Hertz. The upper frequency is limited by the accelerometer's internal resonance frequency, which varies from unit to unit but is typically around 3.5 kHz.

Figure 1: Wilcoxon VS-301 vector sensor

Within the band of approximately 100 Hz to 2 kHz, the sensor frequency response is roughly flat. If the acceleration data is properly integrated and multiplied by the fluid's

specific acoustic impedance $\rho_0 c$ (typ. $1.5 \cdot 10^6$ Rayls), the magnitude of the resulting velocity will equal that of pressure. In this scenario, the components of the sensor can be weighted and summed, effectively resulting in a beamformed single sensor with cardioid directionality pattern (see Figure 2).

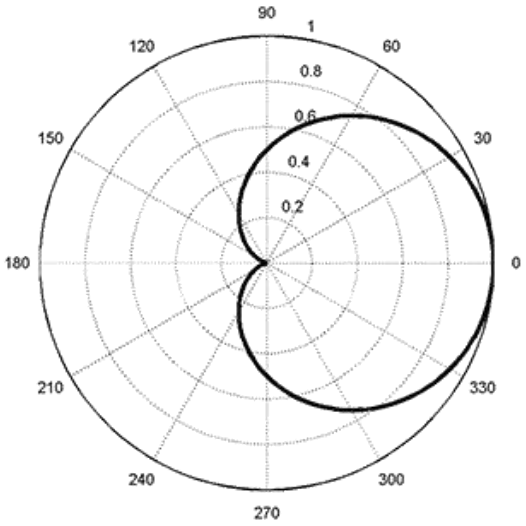


Figure 2: Directionality of single vector sensor steered toward 0 degrees

Figure 2 shows the sensor directionality pattern when steered towards the right. This pattern represents the sensitivity of the steered sensor to sound arriving from a given direction. It shows that the sounds coming from the right are recorded normally, and sounds coming from the left are not recorded at all. The sensor can be steered in any direction desired simply by changing the weights used (see Chapter 5 for further treatment). In order for the steering direction to be specified in a meaningful coordinate system (e.g. aligned with a local North, East, Down system), the sensor's orientation in the coordinate system needs to be known. The recorded velocity components in the sensor coordinate system can then be appropriately rotated into the global coordinate system.

This work makes use of a North-East-Down (NED) coordinate system, which means the positive x-, y-, and z-axes are aligned with North, East and Down respectively. This global coordinate system assumes the ocean surface is a flat plane defined by $z = 0$, and is the default convention used in aerial and nautical navigation. To rotate the sensor coordinate system (SCS) into the global coordinate system (NED), the orientation of the SCS needs to be known in the NED frame. To achieve this, the vector sensor measures the gravity and magnetic field vectors with an internal inertial measurement unit (IMU). The gravity vector is measured with a static accelerometer (separate from the acoustic accelerometer), and determines which way is Down in the SCS. The magnetic field is measured with a magnetometer, and this determines the direction of magnetic north. In the vicinity of the experimental site, the magnetic declination (variation of magnetic north from true north) is known and remains relatively constant. Knowing all three pieces of information, the orientation of the sensor is known in a NED coordinate system. The vector sensor IMU data is packaged with other operational data (e.g. temperature, voltage, error codes, etc.) on the non-acoustic sensor (NAS) data line as a digital bitstream. This bitstream is recorded as an analog signal by the recorder and later digitized to form bits and bytes (see Section 2.3.2).

To augment and validate the vector sensor IMU data, additional IMU's were created, such as the one in Figure 3. These use an accelerometer (ADXL345) and magnetometer (HMC5883L) which are polled by an Arduino Pro Mini microprocessor which then saves the data to a microSD card on a SparkFun OpenLog (SparkFun).

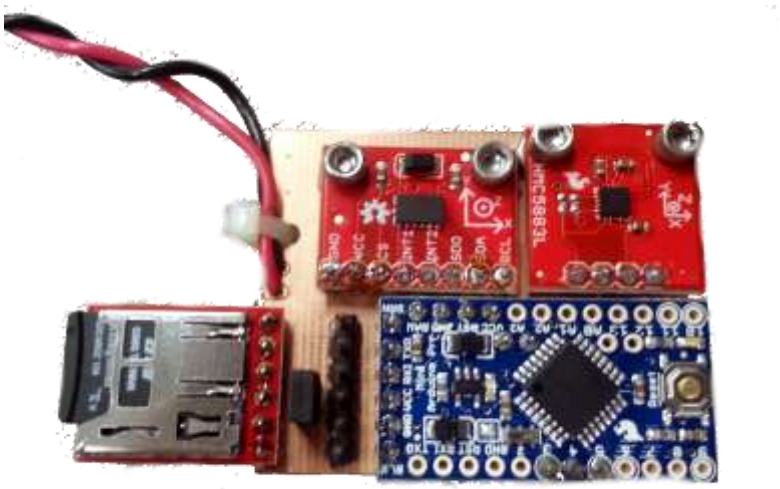


Figure 3: Arduino-based inertial measurement unit (IMU)

These units were placed inside the pressure vessel, and as such, did not measure the sensor's absolute orientation. Rather, they measured the orientation of the pressure vessel, which was coupled to the sensor with known relative orientation. Thus, the orientation of the sensor could be inferred from the orientation of the pressure vessel.

2.2.2 Data Recording and Synchrony Hardware

To record the data from the vector sensors in a synchronized way, multiple pieces of commercially-available hardware were combined. The recording unit is a Tascam DR-680 multi-track recorder (see Figure 4), which is capable of simultaneous sampling of up to six channels of data (TASCAM). It stores the data as wave files on an embedded SD card, and supports a maximum of 32 GB cards. At its default sample rate of 44100 Hz, this allows up to 100 hours of total recording time. Recording five tracks simultaneously (pressure, three velocity components, and NAS data channel) limits the usable recording time to around 20 hours.

Figure 4: DR-680 multi-track recorder

The multi-track recorder specifies a flat frequency response from 20 Hz – 20 kHz (+0.5/-1.5 dB) and 0.007% distortion at 1 kHz. It employs anti-aliasing filters, but does not specify their parameters. Its maximum input level is +24 dBu (~12.3 V), which dictates the scaling of the wave file value (± 1) to volts.

The default clock on the recorder is not stable enough to allow multiple recorders to remain synchronized over many hours of recording. However, the recorders do allow for an external clock source via a digital input called S/PDIF (Sony/Philips Digital Interface Format). The original purpose of this input is to carry digital audio data into the recorder, but it additionally can carry clock information in the digital data frame. Thus, the recorder's clock can be overridden by inputting S/PDIF data with a custom clock source.

To achieve this, a rubidium atomic clock from Symmetricom (SA.33m) is used to generate a very stable 10 MHz clock signal (see Figure 5). It drifts in frequency much less than standard crystal oscillators used in typical commercial equipment, and is even orders of magnitude more stable than an oven-controlled crystal oscillator (OCXO). These were

obtained as surplus from old array hardware, and as such are about 10 years old. Their performance at this age unfortunately does not meet that of new hardware, but they perform suitably for acoustic synchrony.

Figure 5: Symmetricom SA33.m rubidium clock

Since the clock sources only a 10 MHz square wave, and the recorder requires a standardized S/PDIF frame input, a commercial analog to S/PDIF converter was purchased which employs the Cirrus CS8406 digital audio interface transmitter chip. However, this commercial converter uses its own crystal oscillator to supply the CS8406, so it needed to be removed and replaced with the clock signal from the rubidium oscillator. This was achieved using a custom-built clock source board (see Figure 6).

Figure 6: Clock circuit board. From left to right: S/PDIF converter, 4 MHz NAS clock, DC/DC converters, rubidium clock.

The clock board contains the S/PDIF converter on the left, followed by a 4 MHz crystal oscillator (for the vector sensor data channel), two DC-DC converters, and the rubidium clock on the right. A yellow LED indicates whether the rubidium clock has locked onto a steady frequency by turning off when the clock is ready. Thus, the rubidium clock drives the CS8406 on the S/PDIF converter, which passes the clock signal via data frames to the multi-track recorder. Since the recorder's default clock frequency is about 11.2 MHz, the change to 10 MHz means the actual sample rate will be 39062.5 Hz.

With all recorders running on separate stable clocks, the remaining problem is synchronizing all the recorders with each other. To accomplish this, a "sync pulse" is simultaneously recorded on every recorder on a single channel of data at the beginning and end of the experiment. A reference recorder is used to play back the sync pulse on auxiliary cables branched to each recorder, using equal-length BNC cables split from the reference recorder. The sync pulse is comprised of two 10-second linear sweeps from DC to 18 kHz (see Figure 7) with a 10-second pause between them.

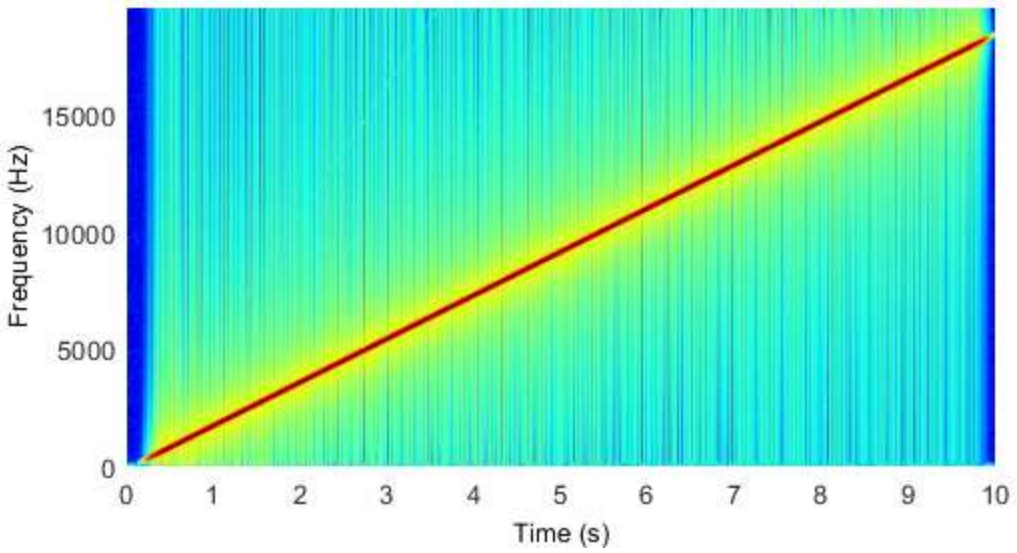


Figure 7: Spectrogram of a single sync chirp. A sync pulse consists of a sync chirp, 10 seconds of silence, and another sync chirp.

One sync pulse (i.e. two sets of chirps) is injected into the acoustic record at the beginning of the experiment, after all clocks have stabilized/locked. Another is inserted 20 – 60 minutes later, and a final sync pulse is applied at the end of the record. The use of three pulses allows the first two to correct for the average difference in clock frequencies, and the third confirms the records remained synchronized throughout the experiment. Additionally, any combination of sync pulses can be used to re-synchronize the acoustic data (see Section 2.3.3).

To demonstrate the clock stability, two different clocks were connected to a frequency counter in relative mode. This records the relative frequency of the clocks by counting clock pulses for each over a short period of time (200 s). Thus, the ratio of number of pulses in this time window is given by

$$R = \frac{N_a}{N_b} \quad (1)$$

where N_a and N_b are the number of pulses in the counting duration for clock A and B, respectively. Alternatively, the ratio can be described by the ratio of clock frequencies, since the number of clock periods is given by

$$N_x = f_x T \quad (2)$$

where f_x is the frequency of the x-th clock and T the counting duration. Substituting Equation 2 into Equation 1, the counting ratio is

$$R = \frac{f_a}{f_b} \quad (3)$$

Thus, a ratio of unity denotes equal clock frequencies. Plotting $(R - 1)f_b$, assuming clock B is the reference clock having f_b = 10 MHz frequency, shows the difference in clock frequencies over the course of a week (see Figure 8).

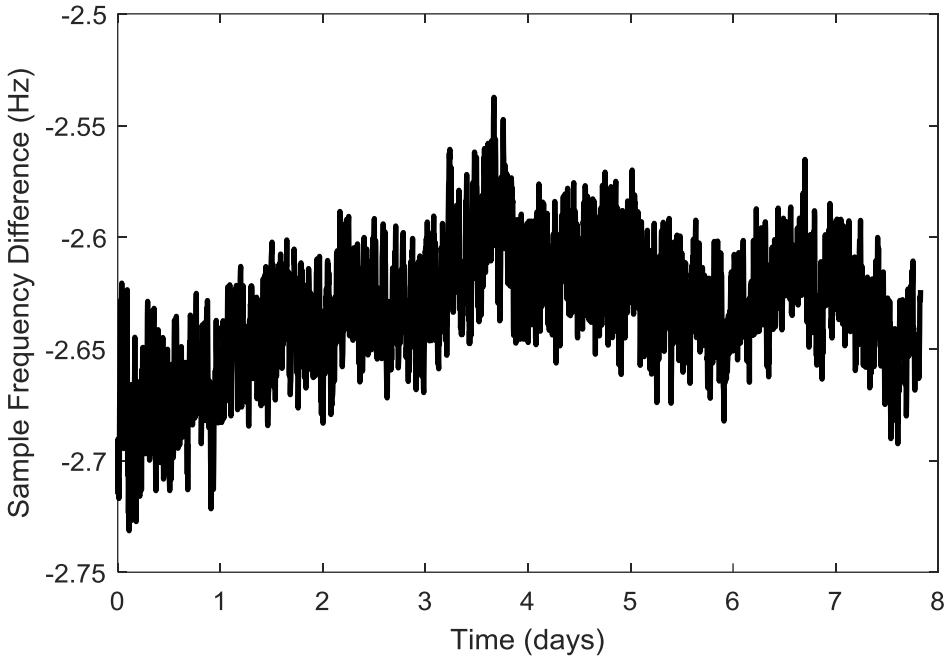


Figure 8: Difference in clock frequencies measured in 200-second intervals over the course of a week. The reference clock is assumed to have 10 MHz frequency.

The relative ratio plotted in Figure 8 shows that clock A was consistently slower than clock B. To determine the effective sample rate at a given time, the total number of samples up to that point is divided by the time elapsed. Assuming again that clock B is the reference, operating at a steady 10 MHz, the total number of samples counted by clock A is

$$\sum N_a = N_b \cdot \sum R = f_b T \cdot \sum R \qquad 4$$

and thus, the effective sample rate at time *t* is

$$\tilde{f_a}(t) = \frac{\sum N_a}{t} = \frac{f_b T}{t} \cdot \sum R \qquad 5$$

and the difference in sample rates is then

$$f_{diff}(t) = \tilde{f_a} - f_b = f_b\left(\frac{T}{t}\sum R - 1\right) \quad 6$$

The sample rate difference using all recordings up to time *t* is plotted in Figure 9 over the course of a week.

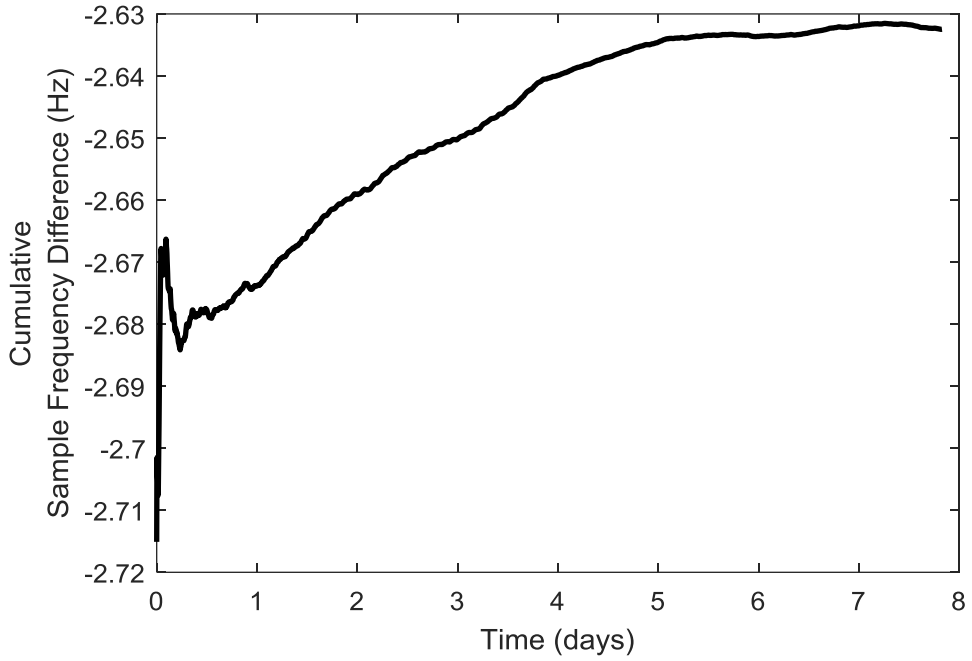


Figure 9: Cumulative clock frequency difference, summing results from 200-second intervals

Figure 9 shows that over the course of a week, clock A had a sample frequency about 2.64 Hz slower than clock B. However, the absolute size of the difference in frequency is not the concern. Applying two sync pulses to the acoustic record, one at the start and another sometime later allows for correction of the average sample rate difference between clocks. For example, adding a sync pulse one hour into the record gives a measurement of the number of samples taken in that hour, and dividing by 3600 seconds (1 hour) gives the cumulative sample frequency over the hour. In this dataset, performing a sync pulse one hour in would have resulted in an effective sample rate of clock A of

9,999,997.3 Hz, assuming clock B was the reference at 10 MHz. Then, only the deviation of clock A from its measured value would result in errors when compared to clock B. This error, expressed in parts-per-million (ppm), is plotted in Figure 10.

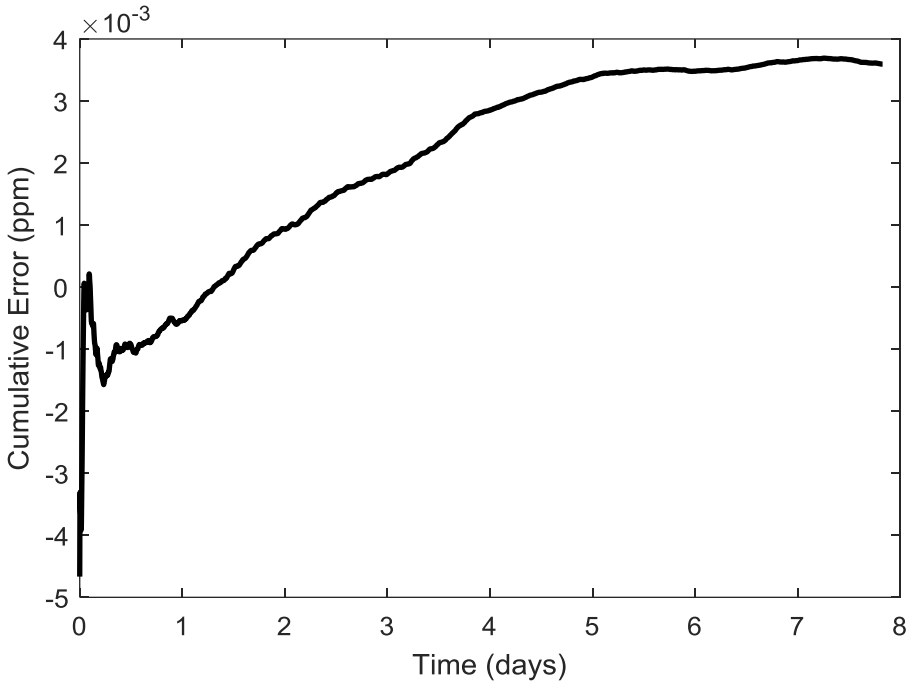


Figure 10: Cumulative clock frequency difference variation. Mean drift was corrected with synthetic sync pulse 1 hour into recording.

The result of Figure 10 shows that the relative clock error (after correcting for the mean difference in clock rates) remains within 4e-3 ppm over an entire week. Over the course of the longest possible recording (20 hours), the worst-case relative clock error is less than 2e-3 ppm, which equates to 144 μs, equal to a propagation distance of 21.6 cm (1500 m/s sound speed). Over a more realistic deployment (8 hours due to battery life), the clock error would be within 58 μs (8.6 cm at 1500 m/s sound speed). This proves the clocks are stable enough to ensure clock drift errors are within reason for localizing sensors and sources. These results are summarized in Table 1.

Table 1: Clock Synchrony Bench-test Summary

Clock Drift Duration (hr)	**Max. Relative Error (μs)**	**Effective Distance Error (cm)** c = 1500 m/s
8	58	8.6
20	144	21.6

2.2.3 Float Hardware

As mentioned in Section 2.2.1, vector sensors require a compliant suspension to allow free movement with the surrounding fluid. This is accomplished using standard 3”-3.5” x 1/8” rubber bands wrapped around a slotted cylinder and pulled around the sensor body (see Figure 11). Varying types of rubber bands were tested, however no conclusive data was obtained on the optimal choice. It seemed a tradeoff exists in choosing the suspension stiffness. On the one hand, a loose suspension decreases the resonant frequency, but on the other hand, it also allows larger sensor motion. If the surface float pulls the sensor cage through the water, the looser suspension might cause the sensor to bump into the cage.

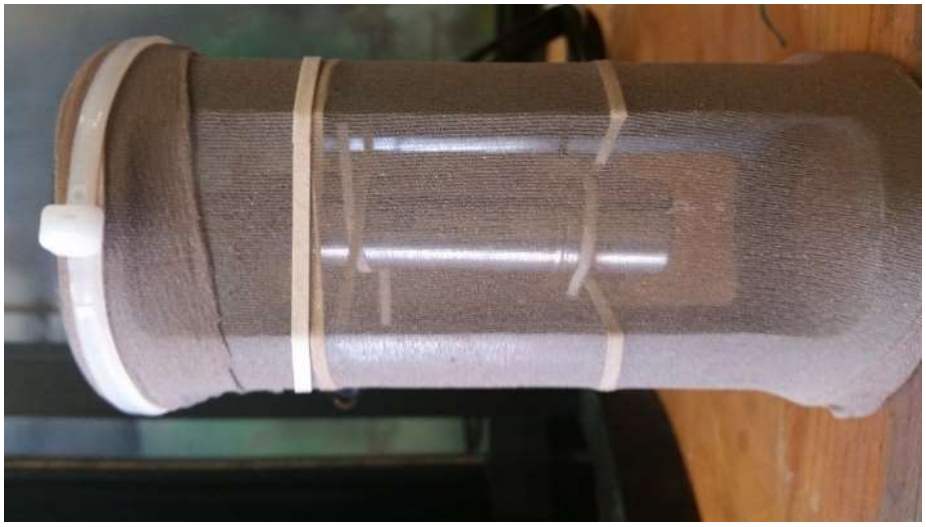


Figure 11: Vector sensor mounting inside sensor cage

In addition to a rubber band suspension, the sensor cage was covered in nylon stockings to help reduce flow around the sensor. Care must be taken when installing the stockings to avoid areas which can trap bubbles. A hot soldering iron can be used to poke a small hole somewhere near the top of the cage while simultaneously sealing the nylon fibers. The sensor cages (Figure 11) were mounted on 1/4”-20 threaded rod near the top of the frame (see Figure 12).

Figure 12: Instrument can, top to bottom: flow shield and sensor cage, instrument can, and drogue

The frames are repurposed hardware from the ARAP array (McEachern, McConnell, Jamieson, & Trivett, 2006), where the sensor cage sits near the top, protected by perforated plastic sheets as additional flow shielding (see Figure 12). The bottom of the frame houses the instrument can, inside which the batteries and recording electronics reside. The battery packs consisted of 24 D-cell alkaline batteries in an 8S3P configuration (3 parallel sets of 8 series batteries). Conservatively, their capacity is about 16000-20000 mAh.

In later deployments of the array, the instrument frame was also fitted with a 4’x4’ plastic sheet to act as a drogue. The entire frame is suspended from the surface float by tethers of varying lengths (see Figure 13). These tethers were originally static lines in parallel with a rubber bungee material, but were later changed out for 8.5 mm diameter

dynamic climbing rope. The suspension requires some compliance to allow the surface float to move vertically with the surface gravity waves while leaving the sensor frame relatively stationary at depth.

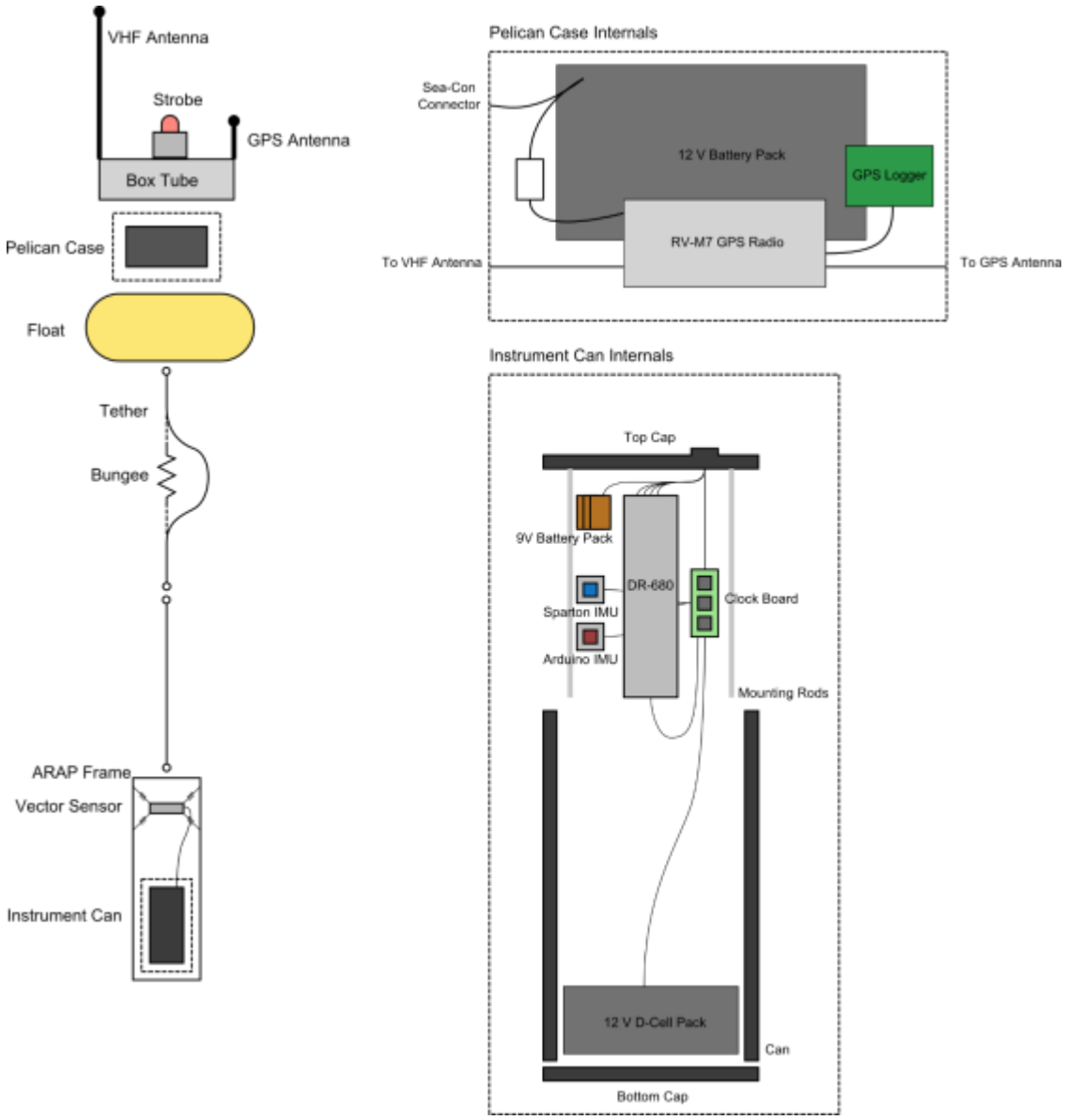


Figure 13: System schematic

The surface floats serve two purposes, the first being to support and suspend the sensor frame. The second is to house the antennae, GPS receiver, strobe and batteries above

water. This allows the position of the float to be measured and used as a reference when locating the sensors using ambient noise correlations. The first revision of the floats repurposed the ARAP hardware (see Figure 14).

Figure 14: Float design, Revision 1

These tended to float somewhat low in the water, and there was concern the instruments might sink were the instrument can to leak and fill with water. The second revision added further floatation elements to add enough reserve buoyancy to prevent such a problem (see Figure 15).

Figure 15: Float design, Revision 2

However, after deploying the second revision of float, it was found the acceleration channels on the vector sensor were overwhelmed with impulsive noise (see Appendix B). It was determined the source of this noise was due to the sudden vertical motions caused by the surface float pulling the instrument frame upward. The effect was more pronounced on days with higher sea-state (see Appendix B), and the accelerometer present in the backup IMU (see Section 2.2.1) measured large accelerations in the vertical direction which were correlated with the impulsive noise events.

To fix this issue, the floats were redesigned to have minimal differential buoyancy. In other words, the cross-sectional area of a horizontal plane sectioning the float was minimized. This means that when a gravity wave passes the float and increases the water height by a small amount, the increase in buoyancy force on the float is minimized. Revision 2 of the floats had very large cross-sectional area, and thus the differential

buoyancy when a wave passed meant the float "rode" the wave rather than let it pass by. This in turn meant the sensor frame was "riding" the waves as well. In addition, the instrument frame drogue serves as a large vibrational damper, restricting motion of the frame upward when the surface float rises. Thus, the force from the float is mostly converted into stretching of the tether rather than displacement of the frame. The third revision of the float design used 4, 4" PVC pipes, each about 10 feet in length (see Figure 16). A 20-pound weight was suspended from a threaded rod below the floats to ensure it remained vertical in the water (see Figure 17). Lastly, a red buoy was attached to the float to aid in its retrieval on the winch, as well as locating the floats visually. The third revision of float was observed to have much smaller noise present on the acceleration channels. Furthermore, they retained their vertical position in the water when a wave passed by, appearing to "sink" below the oncoming wave.

Figure 16: Float design, Revision 3. GPS antenna and waterproof box are mounted between flotation members.

Figure 17: Float design, Revision 3 surfaced. The flotation members are connected at the bottom with two parallel plates, which mount the rod holding ballast weights below.

2.2.4 GPS Positioning and AIS

To measure GPS positions, and also aid in retrieving the floats, a GPS modem was placed inside a waterproof enclosure on the surface floats (see previous section). The GPS modem is a Raveon RV-M7 (see Figure 18) (Raveon). It requires a GPS antenna to obtain GPS locations, and a VHF antenna for sending messages to the base station on the research vessel (also a RV-M7 radio).

Figure 18: Raveon GPS modem

GPS locations are received by the modem, and sent over the VHF link using a time-division multiple access (TDMA) scheme. They can be received by a base station radio on the research vessel at a range up to 5 km depending on conditions. The GPS data is logged on a base station computer and also used to locate the floats for retrieval. In addition, the GPS data is logged on the float itself by connecting a SparkFun OpenLog to the serial port on the Raveon, with some custom signal-level conversion circuitry in between to protect the OpenLog. The electronics within the enclosure on the surface float was powered by a rechargeable 12V sealed lead-acid battery.

The float GPS locations were broadcast by the radios at an interval of 10 seconds, mainly to save battery power. The research vessel location was tracked using the base station radio GPS antenna as well as the vessel's onboard GPS. Other vessels in the area could be tracked using historical automatic identification system (AIS) data. Unfortunately, this only covers larger vessels which are required to use the system, as well as some larger personal craft which opt to use the system. Many of the smaller vessels in the area are not tracked using AIS, and could not be reliably tracked with the research vessel radar since it was far from the array.

2.2.5 Conclusion

This section described the design and development of the array hardware which will be used to collect vector sensor ambient noise data on drifting sensors for testing AEL. The array will also be able to record vessel noise for testing source localization performance.

2.3 Data Collection

2.3.1 Introduction

As mentioned in Section 2.2.1, the vector sensor data does not immediately result in acoustic pressure and particle velocity in a global coordinate system. Many preprocessing steps need to be performed before usable data is obtained for analysis in the following chapters. The steps are:

1. Digitizing the NAS data stream to obtain sensor orientation
2. Rotating acceleration components into NED coordinate system
3. Scaling acceleration and pressure measurements into physical units
4. Integration of acceleration and conversion into pressure units
5. Synchronization of data across different sensors
6. End-use specific preprocessing (e.g. filtering)

Steps 1-4 are described in Section 2.3.2, Step 5 is described in Section 2.3.3, and Step 6 is described in Section 2.3.4.

2.3.2 Vector Sensor Processing

The VS-301 non-acoustic (NAS) data is a digital signal recorded by one analog channel on the DR-680 multi-track recorder. The bitrate is approximately 4 kilobits/sec

and the recorder sample rate is about 40 kHz, thus each symbol takes up 10 samples in the analog recording. A digital ‘1’ is represented by a high/positive voltage, and digital ‘0’ represented by a low/negative voltage. Successive symbols simply take up multiple symbol lengths in the record (see Figure 19).

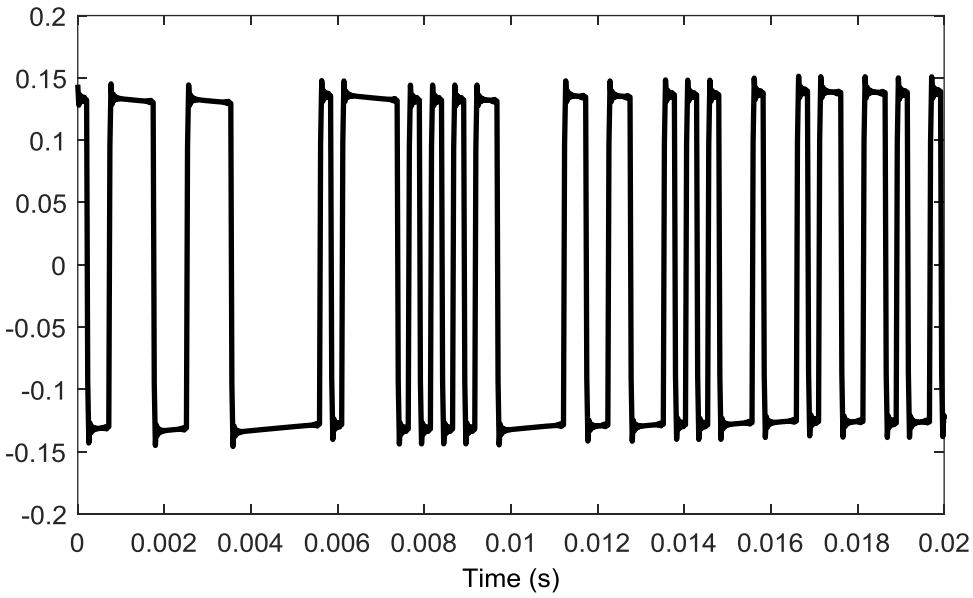


Figure 19: Sample NAS bitstream recorded as analog signal on DR-680

The NAS data is first digitized using a simple scaled signum function

$$d(t) = 0.5(\operatorname{sgn} n(t) + 1) \qquad 7$$

where $n(t)$ is the NAS raw data stream, *sgn* is the signum (sign) function, and $d(t)$ is the digital signal. The time-difference between zero-crossings of the signal $d(t)$ determine how many symbols reside in a given state, and the magnitude of $d(t)$ determines which symbol resides in that state. These bits are written to a file, loaded with varying bit offsets, and descrambled using a proprietary method. The resulting descrambled bytes form

packets of 40 bytes each. The packets contain information such as the sensor gravity vector, magnetic heading vector, temperature, voltage, error code and checksum.

To properly apply the sensor rotation into a global coordinate system, the components of the data packets needed to be scaled or rearranged in some cases. To ascertain the scaling and order of the gravity and heading vector components, each axis of the sensors was pointed downward and the sensor rotated about that axis by 360 degrees. Plotting the resulting magnetic and gravity vectors shows their relationships (see Figure 20).

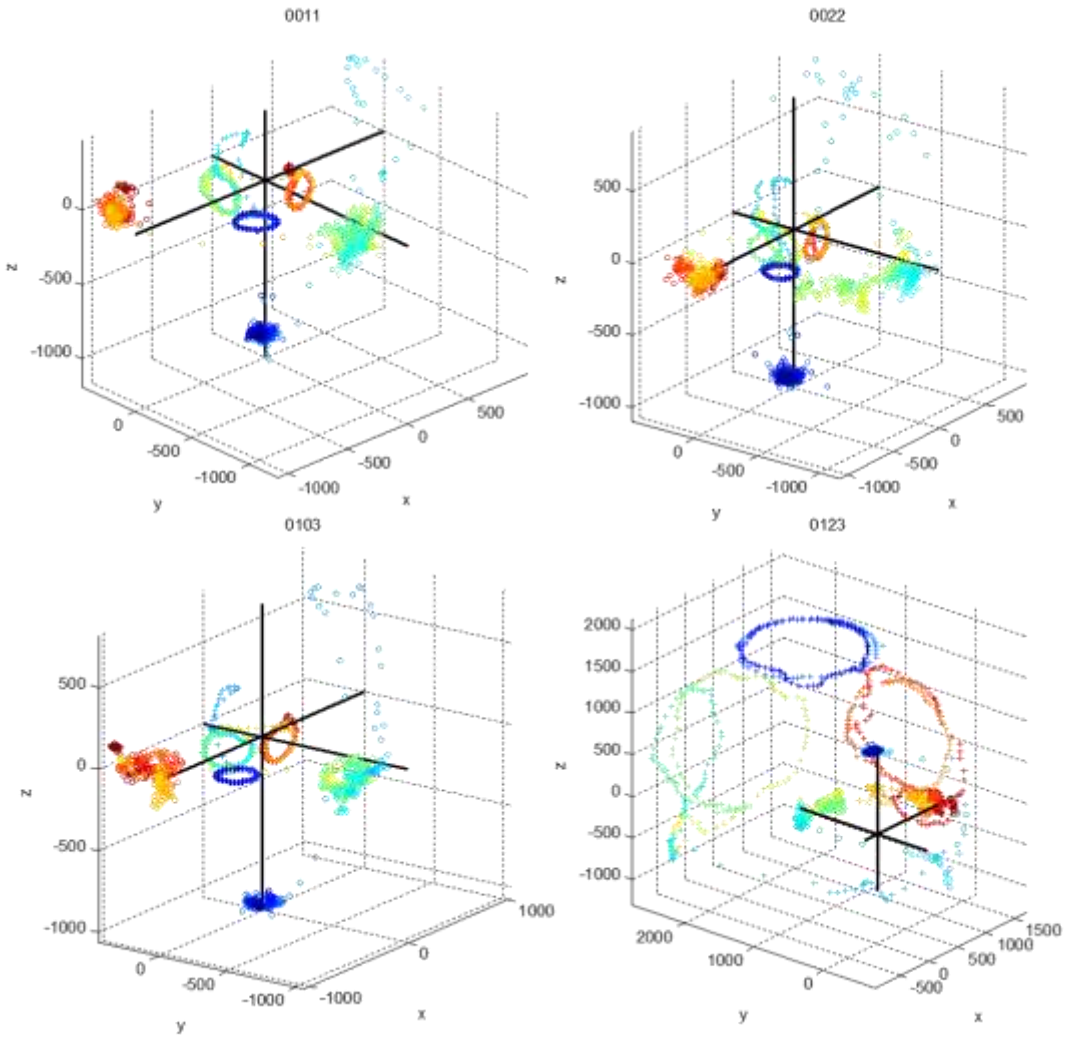


Figure 20: Orientation calibration for VS-301 serial numbers 0011, 0022, 0103, and 0123. Gravity vectors are displayed as open circles, and magnetic field vectors as closed dots. The colors of both denote the elapsed time, starting at blue and going to red. The sensors were oriented with the +Z, +Y, +X axes pointing down, in that order, while rotated around each axis.

Figure 20 shows the gravity vector as open circles, and the magnetic field vector as closed dots. The time at which the vector was recorded is denoted by color, beginning with blue and transitioning to red. Older sensors having serial numbers 0011, 0022, 0103 show the gravity vector is positive in the “up” direction, as opposed to the newer sensor, 0123, which is positive in the “down” direction (i.e. pointing the positive axis to the ground results in a positive measurement). The magnetic vector is inclined downward at the

location of the test, meaning it should have a large positive component along the axis which was oriented downward. This was not the case for the older sensors, but was the case for the newer sensor. Furthermore, the color of the gravity vector and magnetic vector circle surrounding its axis should match. In other words, the magnetic vector should circle the axis which was pointed down during the rotation. In the older sensors, there appears to be a transposition in the x- and y-axes. For example, when the y-axis is pointed down (red gravity circles on negative y-axis), the red magnetic vector dots circle the apparent x-axis. Any use of these sensors in the experiment must take care to record which sensor is used, and each sensor needs to be calibrated in this way to ensure orientation information is valid.

In addition to identifying the scaling and order of the magnetometer components, an in-situ calibration was performed to remove "hard-iron" and "soft-iron" distortions (Konvalin, 2008). These distortions shift the origin of magnetometer data and skew the sensitivity of the magnetometer in different axes, respectively. The result is that magnetometer data will take on a shifted ellipse (see Figure 21), which needs to be converted into a centered circle for accurate heading measurements. To achieve this, the sensor frames are rotated slowly in a full circle during the experiment to obtain a full ellipse of data points. The simplest method involved driving the research vessel in a large circle, taking about 20 seconds to complete, while all sensors were on deck in their standard orientation. Plotting the horizontal magnetic vector data collected during this time shows the ellipsoidal shape caused by distortions (see Figure 21).

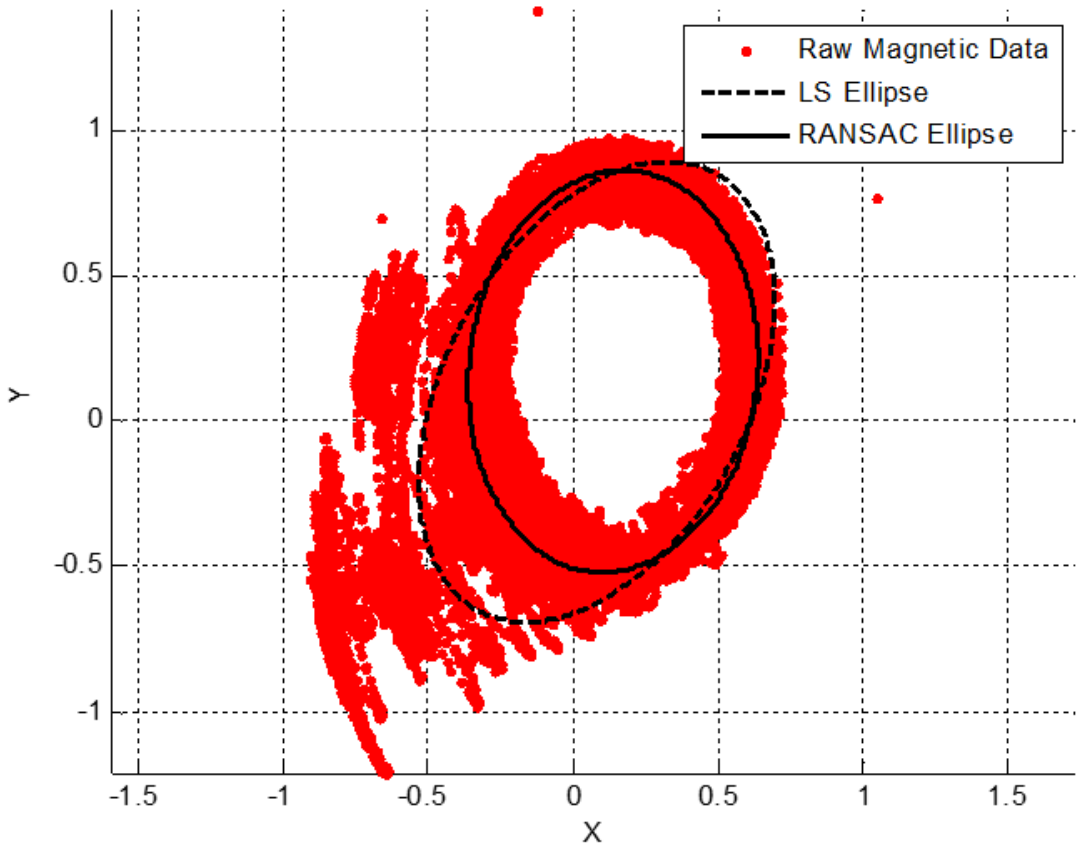


Figure 21: Example magnetometer data in the horizontal plane (red dots), with a least-squares ellipse fit (black dashed line) and a RANSAC-filtered ellipse fit (black solid line).

However, simply fitting a least-squares ellipse to the data results in a poor fit (dashed black line) due to outliers in the data. Employing an outlier-robust curve-fitting technique called RANSAC (Fischler & Bolles, 1981) results in a much better fit to just the inliers (solid black line), which is used to transform the magnetometer data into a circle (see Figure 22).

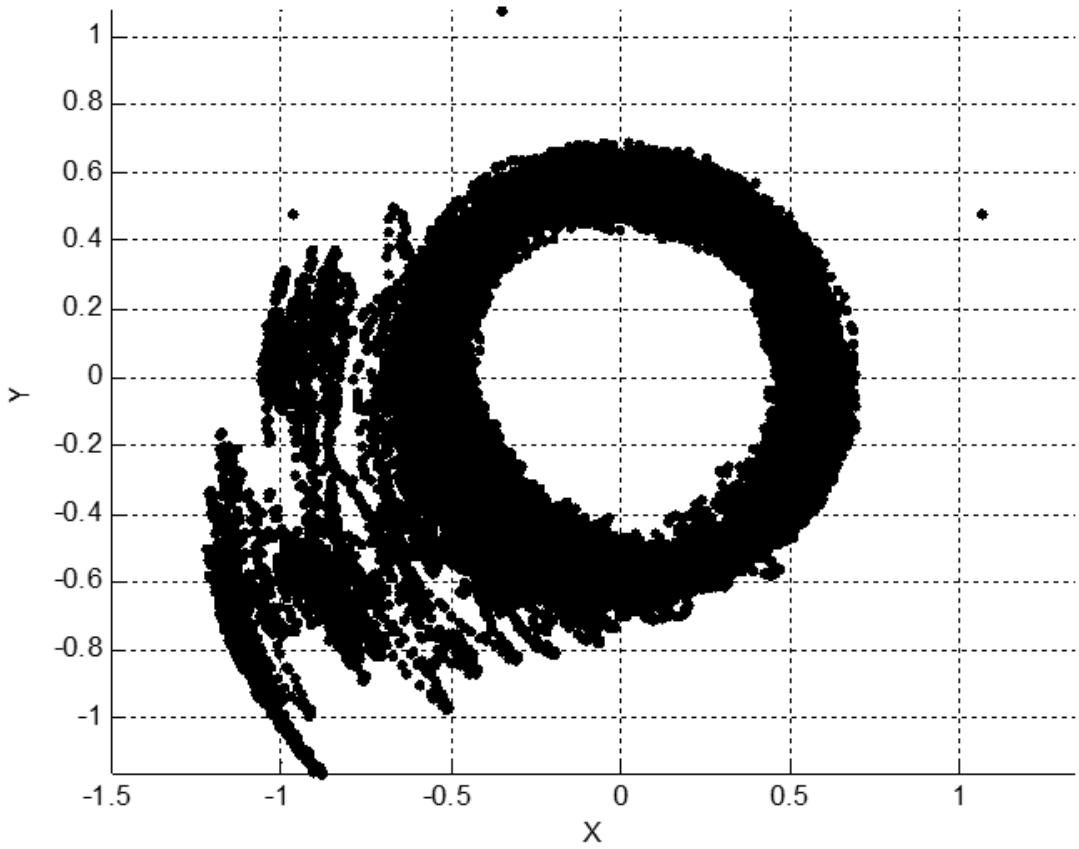


Figure 22: Magnetometer data in the horizontal plane, corrected by the RANSAC-filtered ellipse fit (black solid line in Figure 21).

With calibrated magnetometer and accelerometer data, the axes of the global coordinate system (e.g. NED) can be found in the sensor coordinate system (SCS). The "down" vector is given by

$$\vec{w} = \frac{\vec{g}}{|\vec{g}|} \qquad 8$$

where $\vec{g}$ is the gravity vector, whose positive orientation is "down". The "magnetic east" vector is then given by

$$\vec{v} = \vec{w} \times \frac{\vec{m}}{|\vec{m}|} \tag{9}$$

where $\vec{m}$ is the magnetic field vector (corrected) and the × operator denotes cross-product. Finally, the "magnetic north" vector is given by

$$\vec{u} = \vec{v} \times \vec{w} \tag{10}$$

The rotation matrix from the sensor coordinate system to a magnetic north NED coordinate system is then

$$R_{SCS} = [\vec{u}\ \vec{v}\ \vec{w}]^T \tag{11}$$

and a rotation from a magnetic north coordinate system to a true north coordinate system by a simple z-axis rotation by the magnetic declination angle δ (assumed constant) is given by

$$R_z = \begin{bmatrix} \cos\delta & \sin\delta & 0 \\ -\sin\delta & \cos\delta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{12}$$

All acceleration data can then be transformed from its sensor coordinate system into a common global coordinate system by the transformation

$$\vec{a}_{NED} = R_z R_{SCS}\, \vec{a}_{SCS} \tag{13}$$

where $\vec{a}_{SCS}$ is the acceleration in the sensor coordinate system and $\vec{a}_{NED}$ is the acceleration in the NED coordinate system.

Step 3 simply requires scaling the pressure and acceleration wave file measurements into their respective physical units. To obtain voltages, the values are multiplied by 12.3 Volts/count (+24 dBu), which was obtained from the DR-680 gain

specifications (see Section 2.2.2). Pressure data is scaled using the hydrophone sensitivity of -162 dB re 1V/μPa. Accelerometer data is scaled using varying sensitivities (see Appendix A), generally between 3 and 10 V/g, and using 9.81 m/s^2 per g. The resulting units of pressure are in Pascals, and acceleration in m/s^2.

Step 4 requires integration of acceleration and scaling into pressure units. The scaling is a simple multiplication by the fluid's specific acoustic impedance $\rho_0 c = 1.5 \cdot 10^6$ for water. Integration is implemented in the frequency domain by point-wise multiplication of the spectrum by $1/j\omega$, except at zero-frequency which is multiplied by zero. After steps 1 – 4 are performed on each sensor, the data on each is ready to be analyzed, however if data from multiple sensors is to be analyzed simultaneously, it must be loaded synchronously.

2.3.3 Synchronizing Data

To ensure data across multiple sensors is synchronized, multiple "sync pulses" were applied to each sensor's record (see Section 2.2.2). The first sync pulse effectively synchronizes the start time of each record, and the second is used to synchronize the end time since the clocks each operate at slightly different, unknown frequencies. To obtain the start times and effective sample rates of each data record, one clock is chosen as the reference clock to which all others are synchronized. Then, the delays of each sync pulse are found using a frequency-domain correlation

$$D_i(\omega) = R(\omega) \cdot F_i(\omega)^* \qquad 14$$

where $R(\omega)$ is the reference sync pulse spectrum, $F_i(\omega)^*$ is the conjugate spectrum of the i-th sensor's delayed sync pulse, and $D_i(\omega)$ is the delay correlation spectrum. The time

delay is then obtained by a linear fit of the unwrapped phase of $D_i(\omega)$ within the sync pulse range of frequencies (200 Hz – 18000 Hz in this case). The delay for the *i*-th sensor's sync pulse is then given by

$$\delta_i = \frac{s_i}{2\pi} \tag{15}$$

where s_i is the linear-fitted slope of the phase of $D_i(\omega)$. Finally, each channel of data can obtain its offset from the reference channel start time Δ_i and its sample frequency f_{si} by linear-fitting the points

$$\vec{p} = [\boldsymbol{p} \;\; \boldsymbol{p} + \boldsymbol{\delta_i}] \tag{16}$$

where $\boldsymbol{p}$ is a vector of all pulse times, and $\boldsymbol{\delta_i}$ a vector of all the *i*-th sensor pulse delays. The slope and y-intercept of this linear fit will be m and b, respectively. The offset and sample frequency are obtained by

$$\Delta_i = -\frac{b}{m} \tag{17}$$

and

$$f_{si} = m \cdot f_s \tag{18}$$

where f_s is the assumed sample frequency of the reference clock. The reference clock has by definition $\boldsymbol{\delta_{ref}} = 0$, thus $m_{ref} = 1$ and $b_{ref} = 1$. Then, $\Delta_{ref} \equiv 0$, meaning all other clock's offsets are relative to the reference clock. A negative offset indicates the record needs to start earlier to be in sync, and a positive offset means the record needs to start later to be synchronized.

In most cases, the offsets and sample frequencies of the individual clocks means they won't be aligned sample-for-sample. Thus, the reference clock sample times are used to linearly interpolate all other data points onto a common time axis. This procedure is also performed on other data sources such as the GPS locations, which due to the TDMA protocol are not simultaneously-sampled.

2.3.4 Preprocessing Data

The final and optional step in preparing vector sensor data for analysis is referred to as preprocessing. The steps which can be taken include downsampling, filtering, whitening and clipping. Downsampling was performed using MATLAB's built-in "resample" function with a decimation factor of 8. This aids in the storage and manipulation of data since the default sample rate is 39.0625 kHz, much higher than the highest usable frequency of about 2 kHz. Thus, a downsampled sample rate of ~4883 Hz is just high enough to satisfy the Nyquist-Shannon sampling theorem.

Filtering is performed by first windowing the data in time with a Tukey window (r=0.01), and then multiplying the data spectrum by a Tukey window with r=0.1. Alternatively, a 4th-order Butterworth filter is applied using MATLAB's "filtfilt" zero-phase filtering function. Whitening is performed similarly in the frequency domain, except in addition to the Tukey window, the spectrum is divided by

$$E(\omega) = \sqrt{p(\omega)p^*(\omega) + v_x(\omega)v_x^*(\omega) + v_y(\omega)v_y^*(\omega) + v_z(\omega)v_z^*(\omega)} + \epsilon \qquad 19$$

which is the energy spectrum computed from the vector sensor pressure and velocity spectra $p(\omega)$ and $v_{x,y,z}(\omega)$. A small "machine precision" value ϵ is added to avoid division by zero. Whitening is a step which helps equalize the contribution of ambient noise with

different amplitudes at different frequencies, and makes the noise more “white”, hence the term whitening (Karim G. Sabra et al., 2005b).

Lastly, clipping involves replacing any data values beyond a given amplitude threshold with the threshold. Typically, this threshold is three times the standard deviation of the data, such that only a small number of large impulsive events are clipped. This step helps reduce the impact of these impulses on further processing, and helps to make the data more noise-like.

2.3.5 Experimental Data Collected

2.3.5.1 April 2015

The first at-sea trial of the float hardware took place on the R/V Savannah, approximately 100 miles SE of Savannah, GA near the continental shelf (see Figure 23). The floats were deployed for 17 hours overnight, and retrieved in the morning. During much of this time, the research vessel was operating silently at a distance of 12 km from the array.

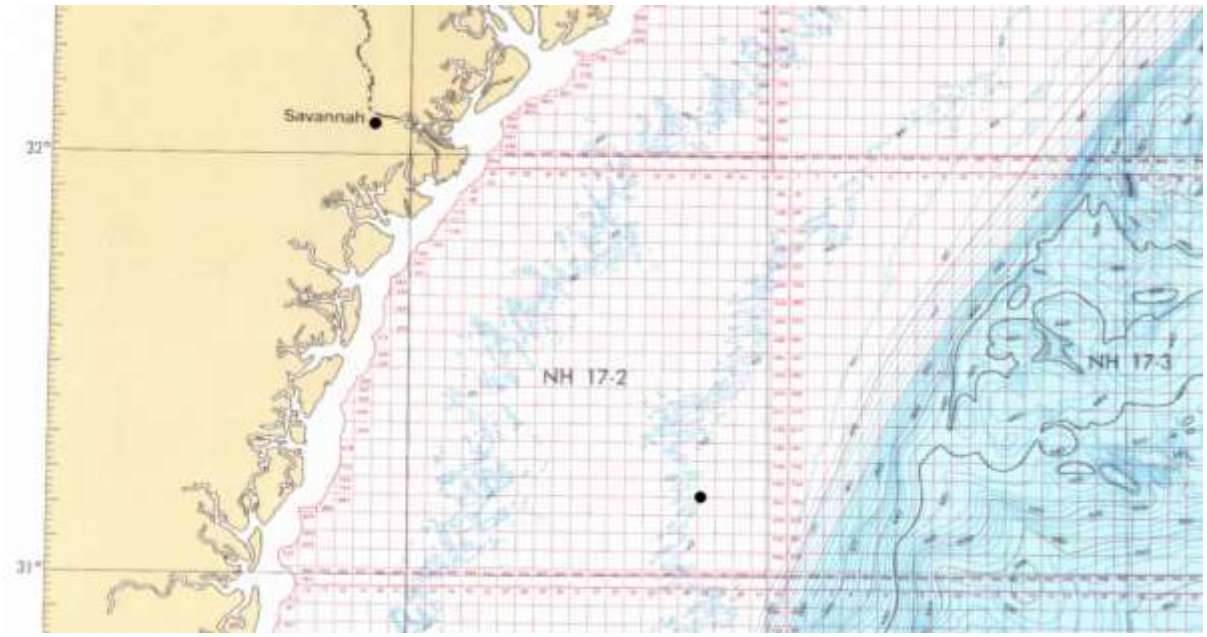


Figure 23: April 2015 deployment area (black dot), 31.15 N, 80.15 W

One of the four floats experienced GPS radio errors and was not deployed. Another float was capsized during deployment, flooding its GPS antenna which caused it to fail. Furthermore, the batteries were shorting to the metal pressure vessel, causing them to drain rapidly. The clocks did not stay synchronized as a result. There is not likely much usable data collected from this experiment.

2.3.5.2 August 2015

The second experiment testing the float hardware was performed in the Long Island Sound, departing port in New London, CT (see Figure 24). The research vessel used was the R/V Michael J. Greeley, graciously operated by the United States Coast Guard Academy. Two days of deployments were used during this experiment, the first a shorter hardware testing day and the second which aimed to collect the required data.

Figure 24: August 2015 deployment site (black dot), 41.2 N, 72.1 W

This deployment was more successful, with no battery or GPS problems. However, some of the clocks in use were an older GPSDO (GPS-disciplined oscillators) type, which were unstable. In addition, the power connectors to the clock were not physically latching, and were accidentally unplugged during removal of the recording hardware, causing loss of whatever synchrony was present up to that point. It is likely that three of the floats remained reasonably synchronized until that point, but the final sync pulse was unable to verify this except for one float pair on August 23rd (see Table 2). In total, only one pair is known to be synchronized, and only hydrophone data is likely to be useful from this experiment due to impulsive noise on the accelerometers.

2.3.5.3 April 2016

This experiment was a repeated deployment to the location tested in August 2015 (see Figure 24). All clocks did not remain synchronized, as the batteries were drained before retrieval. This was likely the combined result of the newer Symmetricom clocks using more power and the attempt to get the clocks to lock by running them overnight

before the experiment. Cold weather may also have played a role in decreasing the batteries' available voltage. Not much usable data was obtained from this experiment.

2.3.5.4 June 2016

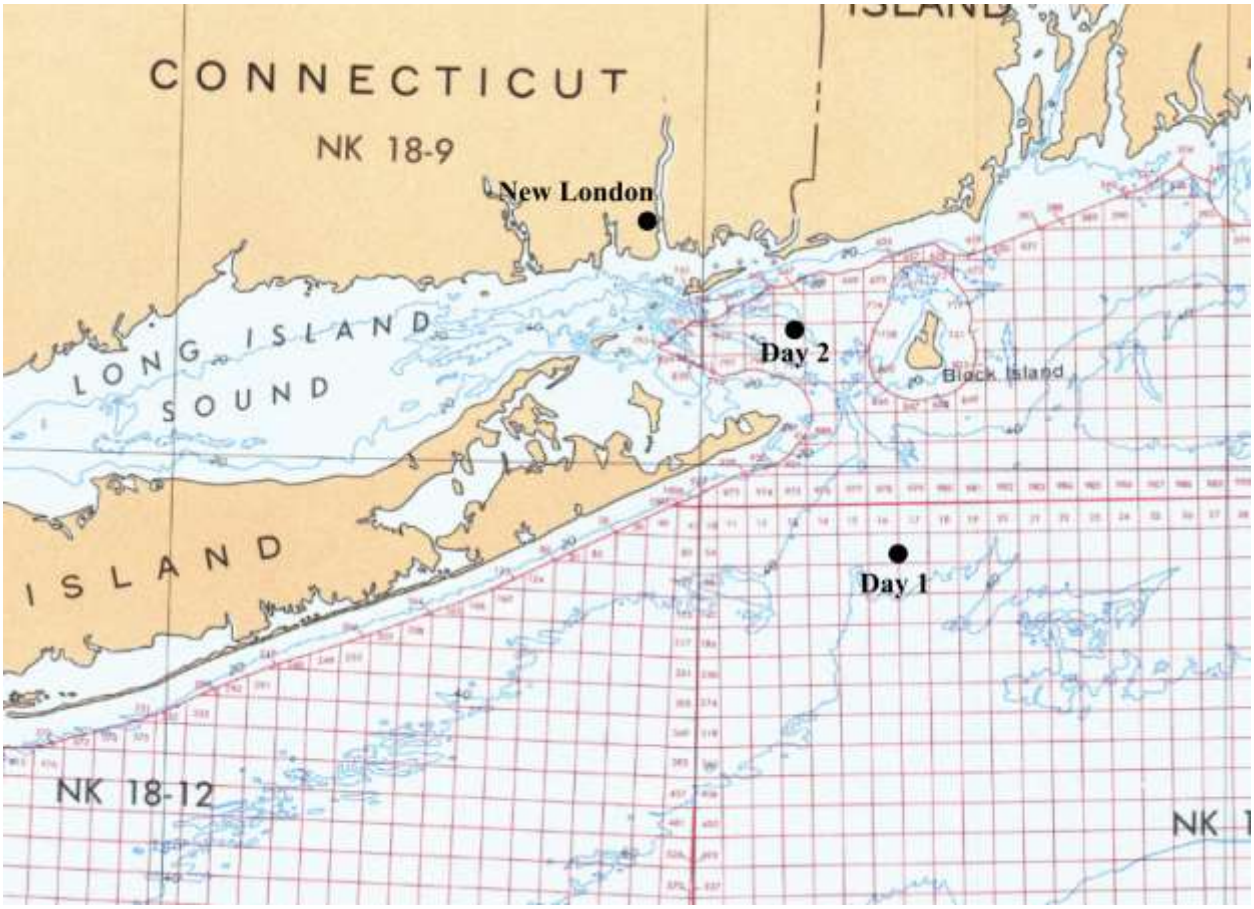


Figure 25: June 2016 deployment sites (black dots) for Day 1 (40.9 N, 71.7 W) and Day 2 (41.2 N, 71.9 W)

This experiment attempted to collect better quiet ambient noise data by traveling further out to sea to deploy the sensors (see Figure 25). The recorder gains were increased to better capture the ambient noise, however the high sea state meant the surface floats induced so much noise into the acceleration channels they were almost entirely clipped (see Appendix B). Fortunately, all recorders remained synchronized and no battery problems were encountered. The second day deployed much closer and with lower sea-

state and gains, however the impulse noise was still prevalent on the acceleration channels. Only one pair of floats remained synchronized on Day 2 due to a few Symmetricom clocks which seemed to unlock when mechanically disturbed (likely old, failing clocks). Hydrophone data from either day may still be useful by-products of this deployment.

2.3.5.5 October 2016

Figure 26: October 2016 deployment site (black dot), 41.2 N, 71.9 W. River deployment occurred in the river near the New London marker.

The final deployment of the floats occurred over two days in October 2016, in the same location as Day 2 in June 2016 (see Figure 26). The first day was designated to test the hardware, and in addition to the free drift deployment, the sensor frames were dropped to the bottom of the river outside the Coast Guard Academy to obtain static ambient noise recordings. The second day of data collection obtained the most quiet ambient noise data and vessel maneuvering. Three of four sensors remained synchronized, however one of these had a malfunctioning GPS antenna and reported erratic GPS measurements. The redesigned third revision of the floats nearly eliminated impulsive noise on the acceleration channels (see Appendix B). In total, over 8 hours of usable vector sensor ambient noise

data was collected, with at least an hour of research vessel maneuvering as well. In addition, there were many nearby vessels, some of which were tracked using AIS. Although not in an ideal, quiet location, this dataset proved to be useful for the analyses in the remainder of this work.

2.3.5.6 Summary

In total, eight days of deployment data were collected, however only two contained all of the data necessary for vector sensor ambient noise correlations and source tracking. A few others may have hydrophone-only data which is usable for other purposes. Also, if precise synchrony across sensors is not required, many more datasets could hold useful information. A summary of the synchronization status and major problems encountered on each deployment is given in Table 2.

Table 2: Experiment Summary

Experiment Date	Synchronized Floats	Possible Data Problems
April 7, 2015	None	N/A
August 22, 2015	Maybe 0002-0004	Impulses[1]
August 23, 2015	0003 & 0004, maybe 0005	Impulses[1]
April 2, 2016	None	N/A
June 20, 2016	All	Extreme Clipped Data
June 22, 2016	0004 & 0005	Impulses[1]
October 29, 2016	0002, 0004, 0005	Sensor Noise[2]
October 30, 2016	0002, 0004, 0005	

[1]: Impulses present on acceleration data from float design (see Section 2.2.3)
[2]: One sensor channel became static part-way through experiment

2.4 Conclusion

The requirements of a new freely-floating vector sensor array for collecting ambient vector sensor noise on drifting sensor platform were introduced. The design and hardware components of the array were presented, and the steps required for obtaining useful data from the array were outlined. A summary of the array's deployments was given in relation to the usable data yielded from each. In short, a newly-designed vector sensor array collected ambient noise and vessel noise data which can be analyzed in the following chapters.

CHAPTER 3
VECTOR SENSOR AMBIENT NOISE CORRELATIONS FOR GREEN'S FUNCTION EXTRACTION

3.1 Introduction

Ambient noise exists in any acoustic environment to some extent, and can be exploited as a source of information about the environment or the sensors placed within. It has been shown theoretically and experimentally that correlations of ambient noise across separated sensors contains the Green's function (also referred to as the Channel Impulse Response) between them (Lani et al., 2013; Leroy et al., 2012; Naughton et al., 2016; Roux & Kuperman, 2004; Karim G. Sabra et al., 2005b; K. G. Sabra et al., 2005). Encapsulating both environmental and sensor placement information, the Green's function defines the propagation of sound between two separated sensors. Exact knowledge of the Green's function is not generally guaranteed in every acoustic environment, and its estimation has many uses ranging from seismology, structural monitoring, ocean tomography, fathometry, sub-bottom profiling, or simply inter-sensor distance estimation (Godin et al., 2010; Harrison & Siderius, 2008; K. G. Sabra et al., 2005; Martin Siderius et al., 2006).

The presence of ambient noise in the ocean can be exploited to estimate the Green's function (Roux & Kuperman, 2004). The ambient noise source may be wind-driven surface noise (high frequency), ambient shipping noise (mid-frequency), or geologic/seismic activity (low frequency), to name a few. Many distributed noise sources exist in any case, and some are well-positioned for their sound to propagate directly between sensors (see Figure 27). In an ocean waveguide, such sources within a narrow region in the endfire of the sensors contribute to the emergence of the Green's function (Roux & Kuperman, 2004).

Thus, to develop the Green's function from the interfering broadside noise, long averaging times and/or bandwidth must be used. To lessen these requirements, arrays of sensors can be spatially filtered to focus on only contributing sources (Leroy et al., 2012).

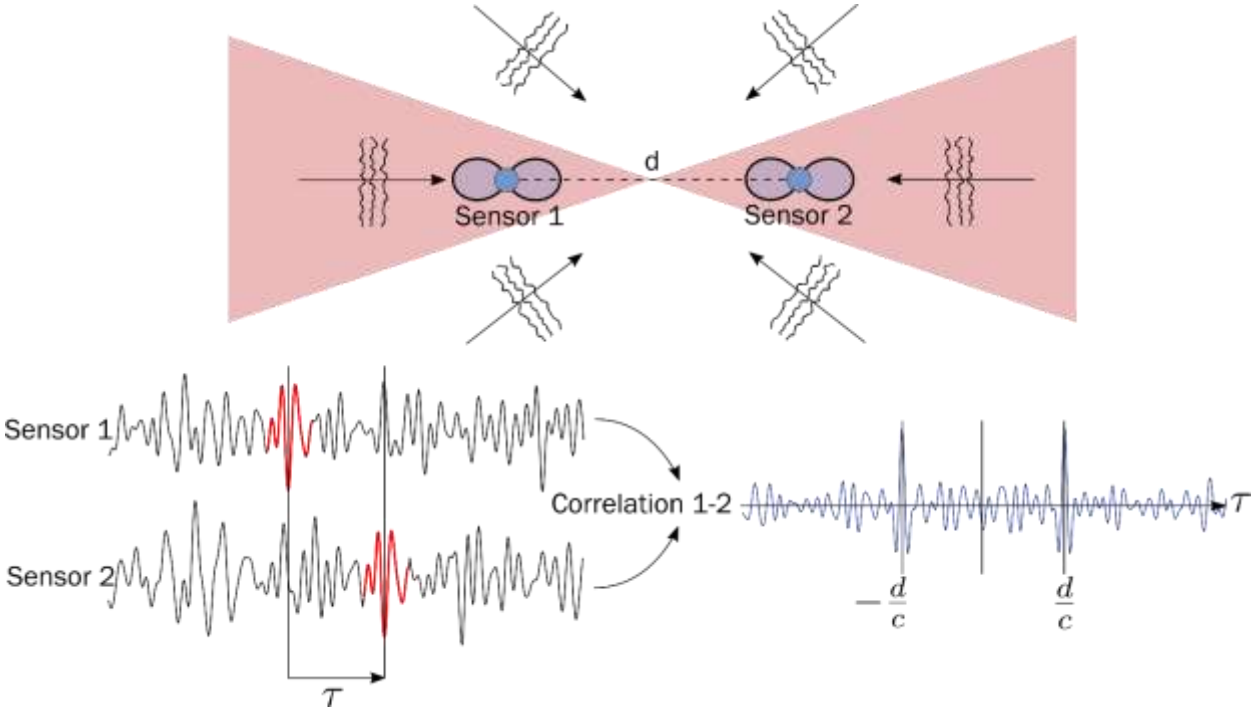


Figure 27: Ambient noise correlation theory for sensors separated by a distance *d*, where sources within the red cones support the emergence of correlation peaks. The correlation of data from sensor 1 and 2 will highlight source contributions such as that highlighted in red, producing two peaks at ± *d/c* given enough averaging time.

Alternatively, the extraction of the Green's function can be improved by leveraging the directionality of vector sensors to reject interfering ambient noise sources. Owing to the directionality of the velocity channels, a vector sensor can be steered to record the component of velocity along the baseline of two sensors (Gerald L. D'Spain, Luby, Wilson, & Gramann, 2006). The resulting dipole beampattern means sound sources broadside to the array (i.e. perpendicular to the baseline) no longer contribute to the ambient noise correlation. In a traditional hydrophone correlation, these sources serve only to increase

the noise power recorded on each sensor, and do not contribute to the emergence of the Green's function. As shown by Bendat and Piersol (Bendat & Piersol, 2010) and also by Sabra et al. (Karim G. Sabra et al., 2005b), the variance of the correlation of finite-length uncorrelated white noise sequences is proportional to the product of noise powers. Thus, increasing the recorded ambient noise power without any improvement in emergence of the Green's function only increases the confounding noise in the correlations. In Section 3.2, expressions for the ambient noise correlation and its derivative are derived for vector sensors in an isotropic ambient noise field in a free space environment to compare the traditional hydrophone correlation and a vector sensor velocity correlation. In contrast to existing works which present the results in the frequency domain (Cox, Lai, & Bell, 2009; Hawkes & Nehorai, 2001), these results will present the correlations in the time domain, building on the results presented by Rafaely (Rafaely, 2000). The simple convolutional expression derived in Section 3.2 provides a unique interpretation of the ambient noise correlation process.

In Section 3.3, the benefit of using vector sensors over hydrophones is quantified and examined for varying degrees of self-noise on the pressure and velocity channels, and experimental correlations are examined to corroborate the theoretical findings.

3.2 Vector Sensor Ambient Noise Correlations

3.2.1 Setup

Consider a pair of separated directional sensors whose separation distance is d (see Figure 28), in the presence of isotropic ambient noise. Without loss of generality, assume a coordinate system whose Z-axis lies along the baseline of the sensors, and that both sensor's acoustic x-axes are aligned with the global frame Z-axis. Any point $\vec{\boldsymbol{r}}$ in the global

coordinate system is defined by a radius (r), azimuth angle (ϕ) and polar angle (θ) (see Figure 28).

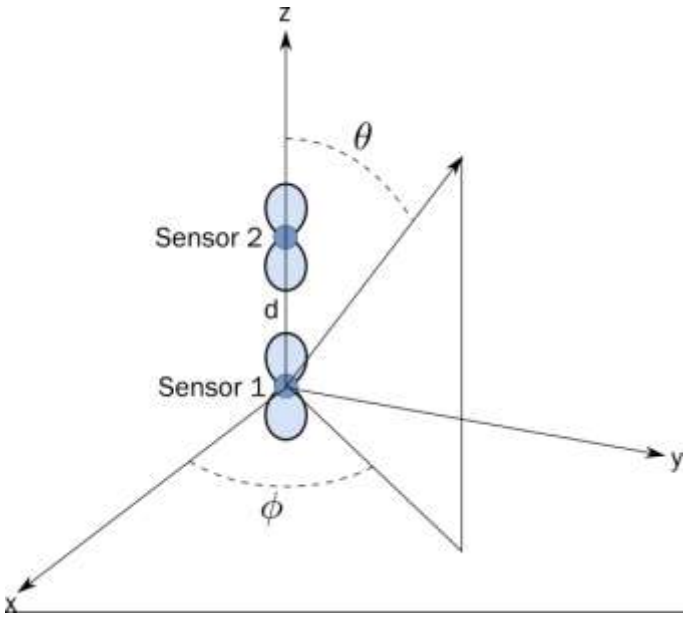


Figure 28: Diagram of vector sensor separated by distance d along the z-axis. Ambient noise propagates with wave vectors described by the azimuth angle ϕ and polar angle θ.

Now, decompose the isotropic ambient acoustic field into plane waves propagating in all directions and express the acoustic pressure as

$$p(\vec{\boldsymbol{r}}, t) = \frac{1}{\sqrt{4\pi}} \int_S s_i(t - \Delta t_{ir})\, \mathrm{d}S \qquad 20$$

where S denotes integration over the surface of the unit sphere, $s_i(t)$ is the ambient noise plane wave propagating in a given direction, and Δt_{ir} the propagation delay of the i-th direction plane wave to position $\vec{\boldsymbol{r}}$. The field's magnitude is divided by an arbitrary factor of $\sqrt{4\pi}$ for later normalization purposes. Furthermore, the velocity of the field, after conversion to pressure units by multiplying by the specific acoustic impedance ($\rho_0 c$) is

$$\vec{\boldsymbol{v}}(\boldsymbol{r},\ t) = \frac{1}{\sqrt{4\pi}} \int_S s_i(t - \Delta t_{ir}) \vec{\boldsymbol{u}}_{\boldsymbol{i}}\, \mathrm{d}S \tag{21}$$

where $\vec{\boldsymbol{u}}_{\boldsymbol{i}} = [\cos\phi_i \sin\theta_i \quad \sin\phi_i \sin\theta_i \quad \cos\theta_i]^T$ is the sensor's directional response vector in the coordinate system in Figure 28. Lastly, the recorded data vector of the two-sensor array will be given by

$$\vec{\boldsymbol{D}}(t) = \left[p(\vec{\mathbf{0}}, t) \quad \vec{\boldsymbol{v}}^T(\vec{\mathbf{0}}, t) \quad p(\vec{\boldsymbol{d}}, t) \quad \vec{\boldsymbol{v}}^T(\vec{\boldsymbol{d}}, t)\right]^T \tag{22}$$

where $\vec{\mathbf{0}}$ is the origin and location of the first sensor, and $\vec{\boldsymbol{d}} = [0 \quad 0 \quad d]^T$ is the location of the second sensor.

For any two components of the data vector, whether on the same sensor or different sensors, the correlation between them is given by

$$C_{uv}(\tau) = E[D_u(t) D_v(t + \tau)] \tag{23}$$

assuming the noise processes are stationary and thus the correlation depends only on the lag τ. Substituting the components D_u and D_v yields

$$C_{uv}(\tau) = \frac{1}{4\pi} E\left[\int_S s_i(t - \Delta t_{iu}) l_u\, \mathrm{d}S \cdot \int_S s_j\big(t - \Delta t_{jv} + \tau\big) l_v\, \mathrm{d}S\right] \tag{24}$$

where l_u, l_v denote the necessary factors for pressure (unity) or velocity (correct component of $\vec{\boldsymbol{u}}$), depending upon which is being correlated. Rearranging the above result gives

$$C_{uv}(\tau) = \frac{1}{4\pi} \int_S \int_S \mathrm{E}\big[s_i(t - \Delta t_{iu}) s_j\big(t - \Delta t_{jv} + \tau\big) l_u l_v\big]\, \mathrm{d}S\, \mathrm{d}S \tag{25}$$

Assuming all individual planewaves from different directions are independent (i.e. uncorrelated), i.e.

$$E\left[s_i(t)s_j(t+\tau)\right] = 0 \quad \forall t, \forall \tau, i \neq j \tag{26}$$

the correlation simplifies to

$$C_{uv}(\tau) = \frac{1}{4\pi}\int_S \mathrm{E}[s_i(t-\Delta t_{iu})s_i(t-\Delta t_{iv}+\tau)]l_u l_v \,\mathrm{d}S \tag{27}$$

To further simplify, define the relative propagation delay between sensors $\tau' = \Delta t_{iv} - \Delta t_{iu}$, and the definition $C_{ss}(\tau) = E[s_i(t)s_i(t+\tau)]$ and substitute to obtain

$$C_{uv}(\tau) = \frac{1}{4\pi}\int_S C_{ss}(\tau-\tau')l_u l_v \,\mathrm{d}S \tag{28}$$

For components on the same sensor, the propagation delay $\tau' = 0$, and for components on different sensors, $\tau' = d\cos\theta\,/c$. The ambient noise autocorrelation is defined by $C_{ss}(\tau)$. The terms l_u, l_v will be either unity for pressure channels or a component of $\vec{\boldsymbol{u}}$ for velocity channels.

3.2.2 Correlations on Same Sensor Components

For correlations between components of the same sensor, the correlation in Equation 28 simplifies to

$$C_{uv}(\tau) = \frac{C_{ss}(\tau)}{4\pi}\int_S l_u l_v \,\mathrm{d}S \tag{29}$$

where l_u, l_v are either unity or a component of the vector $[\cos\phi \sin\theta \quad \sin\phi \sin\theta \quad \cos\theta]$, depending on whether pressure or velocity components are correlated.

3.2.3 Correlations Across Sensors

The correlation in Equation 28 for components on different sensors is given by

$$C_{uv}(\tau) = \frac{1}{4\pi}\int_0^{2\pi}\int_0^{\pi} C_{ss}(\tau - d\cos\theta\,/c) l_u l_v \sin\theta \,\mathrm{d}\theta\,\mathrm{d}\phi \tag{30}$$

where the unit sphere surface integral has been specified for the coordinate system in Figure 28. Since l_u, l_v will be either unity or a component of the vector $[\cos\phi \sin\theta \quad \sin\phi \sin\theta \quad \cos\theta]$, the double integral over azimuth and elevation can be separated into parts

$$\begin{aligned} C_{uv}(\tau) = \frac{1}{4\pi}\int_0^{2\pi} z_u(\phi)\, z_v(\phi)\,\mathrm{d}\phi \\ \cdot \int_0^{\pi} \mathrm{C_{ss}}(\tau - d\cos\theta\,/c)\, e_u(\theta) e_v(\theta) \sin\theta\,\mathrm{d}\theta \end{aligned} \tag{31}$$

where $l_x = z_x(\phi) e_x(\theta)$ is broken into azimuthal (z) and elevational (e) parts. Substitution of the variable $\tau' = d\cos\theta\,/c$ on the elevation integral and simplifying cancelling terms yields

$$C_{uv}(\tau) = \frac{c}{4\pi d}\int_{0}^{2\pi} z_u(\phi)\, z_v(\phi)\, \mathrm{d}\phi \cdot \int_{-d/c}^{d/c} \mathrm{C_{ss}}(\tau - \tau')\, e_u(\cos^{-1}(c\tau'/d)) e_v(\cos^{-1}(c\tau'/d))\mathrm{d}\tau' \quad (32)$$

Now, define a rectangular windowing function

$$\Pi(x) = \begin{cases} 1, & |x| \leq 1/2 \\ 0, & |x| > 1/2 \end{cases} \quad (33)$$

so that the correlation can be written generally as

$$C_{uv}(\tau) = S_{uv} \int_{-\infty}^{\infty} W_{uv}(\tau') C_{ss}(\tau - \tau')\, \mathrm{d}\tau' \quad (34)$$

where the scaling factor S_{uv} and windowing function $W_{uv}(\tau)$ are given by

$$S_{uv} = \frac{c}{4\pi d}\int_{0}^{2\pi} z_u(\phi)\, z_v(\phi)\, \mathrm{d}\phi \quad (35)$$

and

$$W_{uv}(\tau') = \Pi\left(\frac{c\tau'}{2d}\right) e_u(\cos^{-1}(c\tau'/d))\, e_v(\cos^{-1}(c\tau'/d)) \quad (36)$$

Note that as a result of the formulation in Equation 34, the correlation of ambient noise between separated sensors is simply the convolution of a windowing function with the ambient noise autocorrelation, scaled by a constant factor. Alternatively, the correlation of ambient noise across sensors can be viewed as linear time-invariant (LTI) system whose input is the noise process autocorrelation, and whose output is the correlation of separated

sensor components in the ambient noise field (see Figure 29). The transfer function of such an LTI system is simply the product of the scaling factor and windowing function.

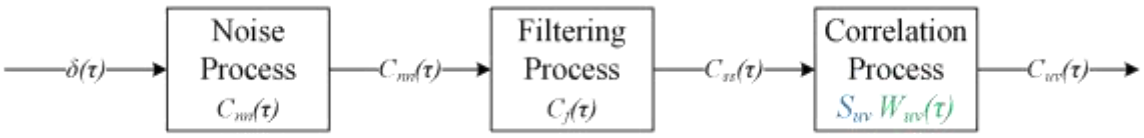


Figure 29: LTI system interpretation of the ambient noise correlation process. Each system/process is represented as a box whose transfer function is specified within.

3.2.4 Simplification

Simplification of the scaling factor (Equation 35) and windowing function (Equation 36) is possible for each pair of correlations (i.e. pressure to pressure, pressure to velocity, velocity to velocity). When correlating the data vector, a covariance matrix is formed as

$$R(\tau) = E\left[\vec{\boldsymbol{D}}^H \vec{\boldsymbol{D}}\right]$$

$$= \begin{bmatrix} C_{p1,p1} & C_{p1,vx1} & C_{p1,vy1} & C_{p1,vz1} & C_{p1,p2} & C_{p1,vx2} & C_{p1,vy2} & C_{p1,vz2} \\ & C_{vx1,vx1} & C_{vx1,vy1} & C_{vx1,vz1} & C_{vx1,p2} & C_{vx1,vx2} & C_{vx1,vy2} & C_{vx1,vz2} \\ & & C_{vy1,vy1} & C_{vy1,vz1} & C_{vy1,p2} & C_{vy1,vx2} & C_{vy1,vy2} & C_{vy1,vz2} \\ & & & C_{vz1,vz1} & C_{vz1,p2} & C_{vz1,vx2} & C_{vz1,vy2} & C_{vz1,vz2} \\ & & & & C_{p2,p2} & C_{p2,vx2} & C_{p2,vy2} & C_{p2,vz2} \\ & & & & & C_{vx2,vx2} & C_{vx2,vy2} & C_{vx2,vz2} \\ & & & & & & C_{vy2,vy2} & C_{vy2,vz2} \\ & & & & & & & C_{vz2,vz2} \end{bmatrix} \quad 37$$

where the lower diagonal is omitted due to correlation symmetry (i.e. $C_{uv}(\tau) = C_{vu}(-\tau)$), and the explicit dependence on τ is omitted for clarity. Dividing the 8 x 8 matrix into submatrices for each sensor yields

$$R(\tau) = \begin{bmatrix} R_{11}(\tau) & R_{12}(\tau) \\ R_{21}(\tau) & R_{22}(\tau) \end{bmatrix} \tag{38}$$

where a submatrix $R_{ij}(\tau)$ denotes the 4 x 4 covariance matrix for components of sensor i and j. Mirroring the structure of the covariance submatrices, the scaling factor and windowing functions for cross-sensor ($i \neq j$) correlations are

$$S = \frac{c}{d}\begin{bmatrix} 1/2 & 1/2 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 1/4 & 0 \\ 0 & 0 & 0 & 1/4 \end{bmatrix} \tag{39}$$

$$W(\tau) = \Pi\left(\frac{c\tau}{2d}\right) \cdot \begin{bmatrix} 1 & \frac{c\tau}{d} & 0 & 0 \\ -\frac{c\tau}{d} & \left(\frac{c\tau}{d}\right)^2 & 0 & 0 \\ 0 & 0 & 1-\left(\frac{c\tau}{d}\right)^2 & 0 \\ 0 & 0 & 0 & 1-\left(\frac{c\tau}{d}\right)^2 \end{bmatrix} \tag{40}$$

and the correlations for same-sensor ($i = j$) components are

$$R_{ii}(\tau) = C_{ss}(\tau) \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/3 & 0 & 0 \\ 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 1/3 \end{bmatrix} \tag{41}$$

Since the Green's function emerges from the derivative of the ambient noise correlations, a more useful representation might be the derivative of Equation 34:

$$\dot{C}_{uv}(\tau) = S_{uv}\, W_{uv}(\tau) * \dot{C}_{ss}(\tau) = S_{uv}\, \dot{W}_{uv}(\tau) * C_{ss}(\tau) \tag{42}$$

where the dot superscript denotes a derivative with respect to τ, and the $*$ operator denotes convolution. Either of the two expressions are valid, and each provide a different perspective on the formation of the ambient noise correlation derivative. The first

expression convolves the windowing function with the derivative of the noise autocorrelation. The second convolves the derivative of the windowing function with the noise autocorrelation. Both cases yield identical results, but offer different interpretations which will be examined in the following section.

3.2.5 Discussion

The windowing functions for cross-sensor correlations are plotted in Figure 30, normalized to unit amplitude. The window function describes the LTI impulse response, or in other words, is the resulting correlation for an ambient noise process whose autocorrelation is a dirac delta function (e.g. an ideal white noise sequence of infinite bandwidth).

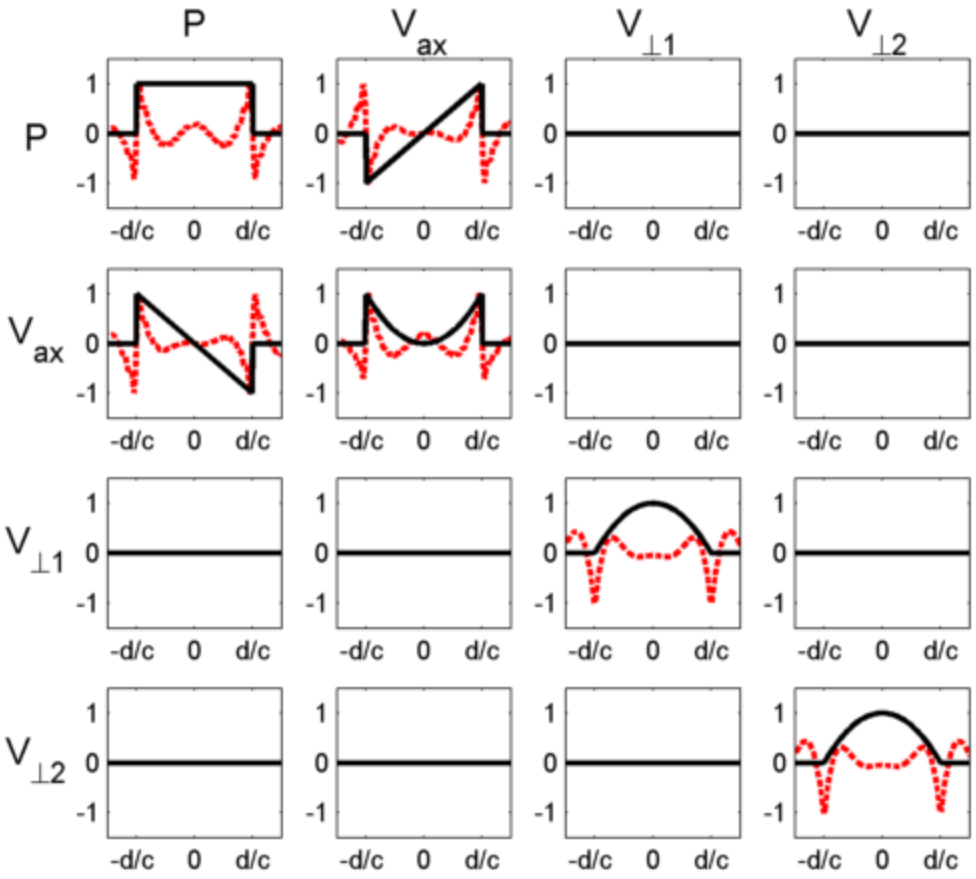


Figure 30: Windowing functions (see Equation 40) for various pairs of vector sensor channel correlations (black solid lines). Windows are normalized to unit amplitude for comparison, and also normalized on the horizontal axis between ±*d/c*. Shown in red dashed lines are normalized ambient noise correlations (see Equation 34) for an example bandwidth of 150-1500 Hz.

Upon initial inspection, the pressure correlation window corroborates existing analytical results from both the time (Rafaely, 2000) and frequency domain (Cox et al., 2009; Hawkes & Nehorai, 2001). The other correlation windows are not as readily verified, but also correspond to the inverse Fourier transform of the frequency domain results in Cox et al. and Hawkes & Nehorai (see Appendix C for comparisons). From the figure, it is clear that in the infinite-band ideal noise case, the axial velocity correlation is "sharper" than the corresponding pressure correlation. If the correlations were to be used to identify the

separation distance, the velocity correlation would slightly outperform pressure due to having a more pronounced peak associated with the direct arrival.

However, in many real applications there exists a lower frequency limit which can be measured, especially with vector sensors. Due to digital recording limitations such as storage, sample frequency, and aliasing, there also exists an upper limit to the frequencies recorded. As such, the ambient noise autocorrelation will deviate from an ideal Dirac delta function. For future examples, assume the ambient noise is band-limited Gaussian white noise with a center frequency f_c and bandwidth B, and whose one-sided power spectral density is a constant σ_s^2/B between frequencies $f_c - \frac{B}{2}$ and $f_c + \frac{B}{2}$. Given that the noise autocorrelation is the inverse fourier transform of the power spectral density, it can be written analytically as (see Appendix D)

$$C_{ss}(\tau) = \sigma_s^2 \cdot \cos(2\pi f_c \tau) \cdot \mathrm{sinc}(\pi B \tau) \qquad 43$$

The result is a sinc-windowed cosine at the center frequency, whose envelope width decreases with increasing bandwidth, approaching a Dirac delta function as bandwidth goes to infinity. For example, the noise autocorrelation for a few example bands are plotted in Figure 31.

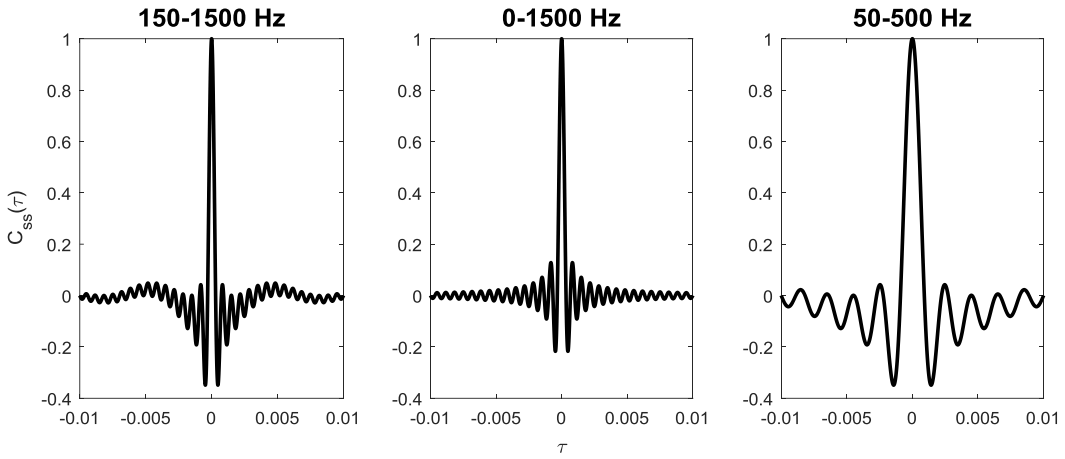


Figure 31: Sample noise autocorrelations (see Equation 43) for a) 150-1500 Hz b) 0-1500 Hz and c) 50-500 Hz filtered white noise.

When convolved with the windowing functions, the realistic noise autocorrelation functions tend to highlight areas with sharp transitions, i.e. the "steps" at $\tau = \pm d/c$. For the same bands demonstrated in Figure 31, the pressure ambient noise correlation computed from the convolution in Equation 34 is compared to the theoretical window (black dashed line) in Figure 32.

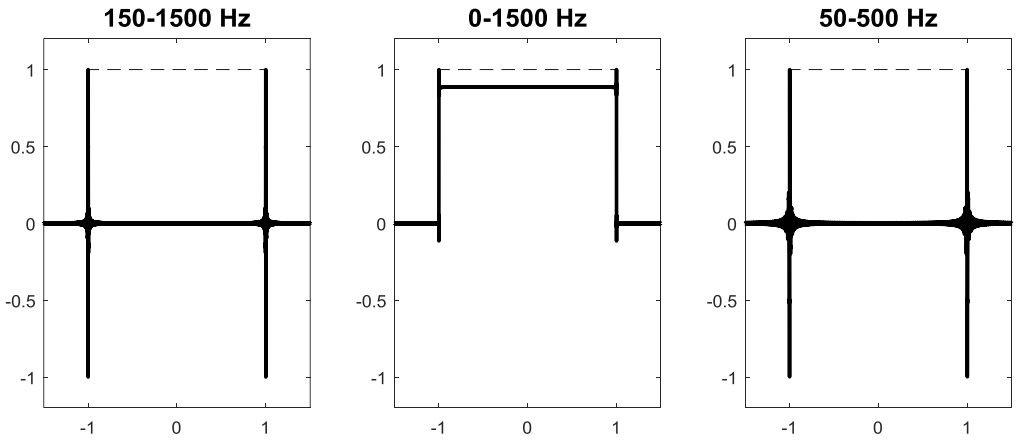


Figure 32: Sample ambient noise correlations (see Equation 34) for pressure channels, normalized to unit amplitude and unit separation distance for a) 150-1500 Hz band b) 0-1500 Hz band and c) 50-500 Hz band. The windowing function is plotted in dashed lines.

As seen in Figure 32, the ambient noise correlation for pressure does not approach the shape of the windowing function unless the lower frequency limit is very near zero. For this reason, it is possible to estimate the separation distance directly from bandlimited ambient noise correlations, since the location of the sharp peaks seem to correspond directly with the $\tau = \pm d/c$ (unity in the normalized plots).

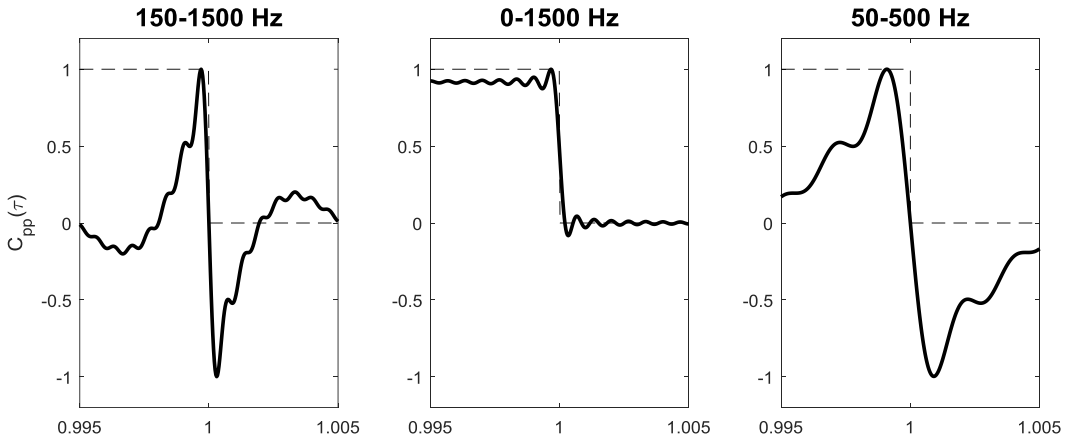


Figure 33: Inset of positive peaks in Figure 32, with the windowing function plotted in dashed lines.

However, upon closer inspection (see Figure 33), it becomes clear that the peaks of the ambient noise correlation do not exactly align, and the error is exacerbated for lower bandwidth. For this reason, the Green's function should strictly be estimated from the *derivative* of the ambient noise correlation. This approach is also the recommended approach in the literature (Karim G. Sabra et al., 2005b), and is discussed in further detail there.

3.2.6 Derivative of Ambient Noise Correlations

The derivative of the ambient noise correlation is expressed in two forms in Equation 42. The first interpretation uses the same windowing functions plotted graphically

in Figure 30 and convolves them with the noise autocorrelation *derivative*. For the example used previously (Equation 43), the derivative of the noise autocorrelation is given by

$$\dot{C}_{ss}(\tau) = \sigma_s^2 \cdot \left[\frac{\cos(2\pi f_c \tau) \cdot (\cos(\pi B \tau) - \operatorname{sinc}(\pi B \tau))}{\tau} - 2\pi f_c \sin(2\pi f_c \tau)\ \operatorname{sinc}(\pi B \tau)\right] \quad 44$$

where the ambient noise power is σ_s^2, the signal bandwidth B and center frequency f_c. For the same bands presented previously, the ambient noise autocorrelation derivatives are plotted in Figure 34.

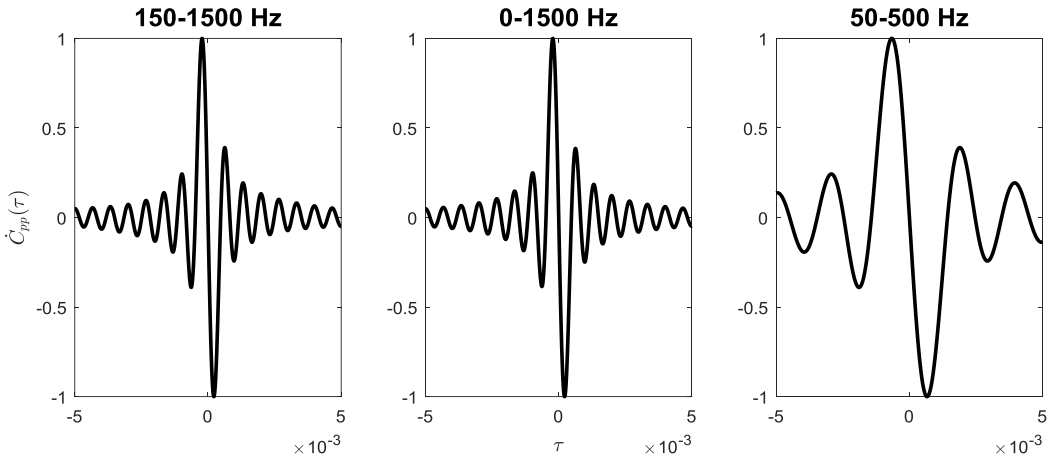


Figure 34: Sample noise autocorrelation derivatives (see Equation 44) for a) 150-1500 Hz b) 0-1500 Hz and c) 50-500 Hz band.

As was the case with the ambient noise autocorrelation, the main lobe width is inversely proportional to the bandwidth. When convolved with the windowing functions, narrower main lobes will produce sharper (i.e. more precise) peaks for estimating the separation distance. A difference between the derivative of the autocorrelation and the autocorrelation itself is that the inclusion of near-zero frequencies does not have much an effect, due to the scaling by $j\omega$ in the frequency domain when taking a derivative. This is

evident in the ambient noise correlation derivative examples plotted in Figure 35, obtained from the convolution in Equation 42.

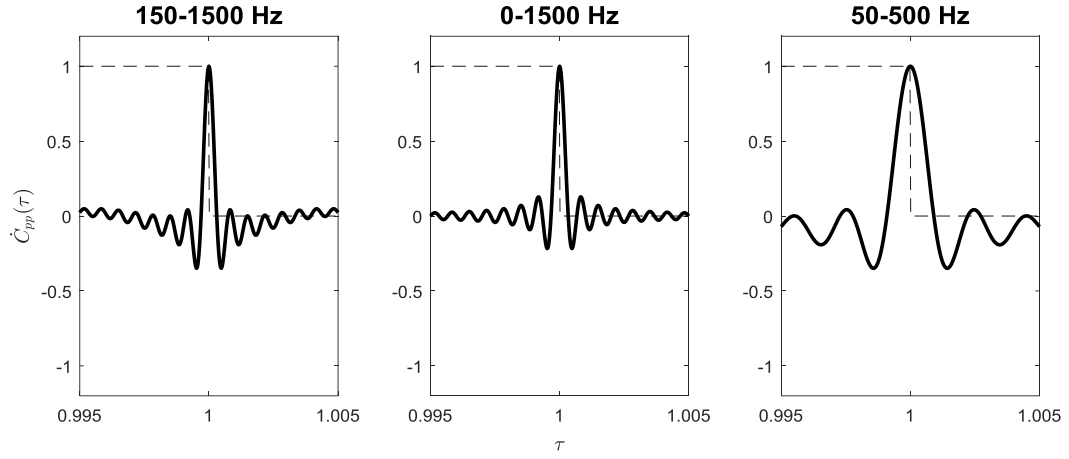


Figure 35: Sample ambient noise correlation derivatives (see Equation 42) for pressure channels, normalized to unit amplitude and unit separation distance for a) 150-1500 Hz band b) 0-1500 Hz band and c) 50-500 Hz band. The windowing function is plotted in dashed lines. Note the similarity of these peaks to those in Figure 31.

The ambient noise correlation derivative provides peaks at the propagation delay which can be identified and used to estimate the sensor separation distance. These peaks are nearly identical to the noise autocorrelation function, a fact which is more easily verified using the second interpretation of Equation 42. Rather than taking the derivative of the noise autocorrelation, this interpretation convolves the derivative of the windowing function with the noise autocorrelation. The windowing function derivatives are plotted in Figure 36, normalized to unit amplitude for comparison purposes.

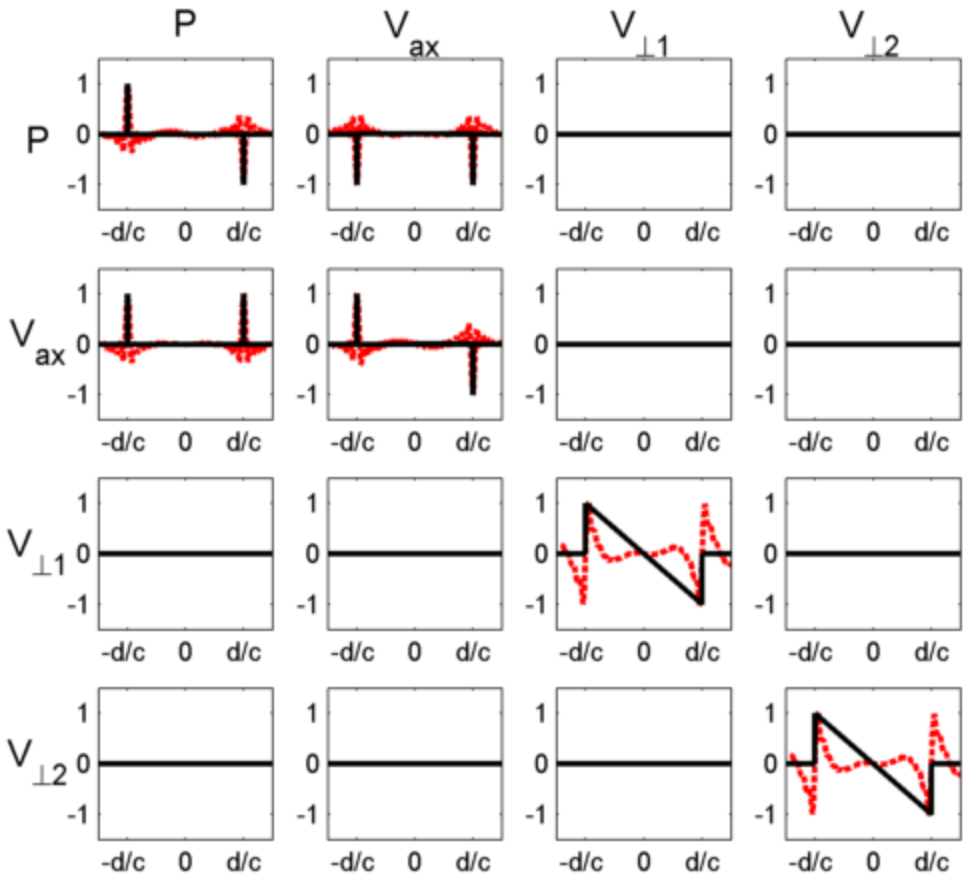


Figure 36: Derivatives of windowing functions (see Equation 40) for various pairs of vector sensor channel correlations (black solid lines). Windows are normalized to unit amplitude for comparison, and also normalized on the horizontal axis between ±*d/c*. Shown in red dashed lines are normalized ambient noise correlation derivatives (see Equation 42) for an example bandwidth of 150-1500 Hz.

It now becomes clearer why peaks of the ambient noise correlation are nearly identical to the noise autocorrelation shape, due to the windowing functions being comprised of two widely-separated impulsive peaks, which convolve almost separately with the noise autocorrelation, unchanging it. The result of the convolution is two peaks at $\tau = \pm d/c$ having all the characteristics of the ambient noise autocorrelation. It is worth noting that the sign of one peak may be negative, a consequence of taking the derivative of the windowing function. This sign change is unimportant if only the location of the peak is desired, and is further discussed in Sabra et al. (Karim G. Sabra et al., 2005b).

3.2.7 Conclusion

Correlations between vector sensor data collected in an ambient noise field can be used to obtain the Green's function direct arrival time in a free space environment. Expressions for an isotropic ambient noise field pressure and particle velocity were correlated to obtain a simple convolution expression for the correlation result directly in the time-domain. Examples in various frequency bands were presented and compared, and they indicated the need to use the derivative of ambient noise correlations for Green's function extraction. The same frequency band examples were presented for the ambient noise correlation derivative, using both interpretations of the derivative expression. Both interpretations resulted in the ambient noise correlation derivative being comprised of ambient noise autocorrelation peaks centered on the propagation delay times. Identifying these peaks and assuming a constant propagation speed results in an estimate of the sensor separation distance.

3.3 Performance of Vector Sensor Green's Function Extraction

3.3.1 Theoretical Performance Measures

As seen in Section 3.2, correlations of data from vector sensors in an ambient noise field provide information on the Green's function between sensors. However, it is important to note the theory only provides the *expected value* of the ambient noise correlation. If only a limited number of samples are recorded for a finite duration, there will be a non-zero variance in the correlation tails given by (Karim G. Sabra et al., 2005b)

$$\mathrm{Var}[C_{uv}(|\tau| \gg 0)] \approx \frac{C_{uu}(0)C_{vv}(0)}{2B_w T_r} \qquad 45$$

Thus, the variance in the correlation tails is proportional to the product of the powers, $C_{uu}(0)C_{vv}(0)$, in each correlated signal, and decreases with increasing observation length T_r or bandwidth B_w. One interpretation of this result is that there exists some correlation process "noise" as a result of a finite-length correlation which corrupts the expected value Green's function peaks. If the variance of this noise is too large, the Green's function peaks may not be visible. Thus, the variance of the correlation dictates the required number of samples (i.e. averaging time) required for the Green's function to emerge (Karim G. Sabra et al., 2005b).

If in addition to the acoustic noise defined in Equations 20 and 21, there also exists uncorrelated noise on each sensor, the variance will increase, potentially masking the acoustic noise correlation peak which results in the Green's function estimate. The model for measured acoustic pressure in the presence of additive noise is given by

$$p(\vec{\boldsymbol{r}}, t) = \frac{1}{\sqrt{4\pi}} \int_S s_i(t - \Delta t_{ir}) + n_{pi}(\vec{\boldsymbol{r}}, t)\, \mathrm{d}S \qquad 46$$

where $n_{pi}(\vec{\boldsymbol{r}}, t)$ is the *i*-th component of the time-domain pressure noise signal at a point $\vec{\boldsymbol{r}}$. It is assumed to be uncorrelated for different indices *i* and at different locations, and has standard deviation σ_{np}. The noise signal is defined inside the integral so that the pressure signal-to-noise ratio (SNR) can be simply defined as

$$SNR_p = \frac{\sigma_s}{\sigma_{np}} \qquad 47$$

where σ_s is the standard deviation of the acoustic/ambient noise signal $s_i(t)$. Likewise, the model for acoustic velocity in the presence of noise is given by

$$\vec{\boldsymbol{v}}(\boldsymbol{r},\ t) = \frac{1}{\sqrt{4\pi}} \int_S s_i(t - \Delta t_{ir})\vec{\boldsymbol{u}}_{\boldsymbol{i}} + \vec{\boldsymbol{n}}_{\boldsymbol{vi}}(\vec{\boldsymbol{r}}, t)\, \mathrm{d}S \qquad 48$$

where $\vec{\boldsymbol{n}}_{\boldsymbol{vi}}(\vec{\boldsymbol{r}}, t)$ is an indentically-distributed, independent noise vector which is uncorrelated for different locations and indices *i*. Depending on the source of noise, the magnitude of $\vec{\boldsymbol{n}}_{\boldsymbol{vi}}$ may depend on the sensor's directional response.

We define **extrinsic noise** as noise which is affected by the sensor's directionality. Extrinsic noise is thus acoustic in some sense, but *not* a result of the ambient acoustic noise which contributes to the emergence of the Green's function. Rather, extrinsic noise encompasses effects such as local flow noise which is uncorrelated across sensors. In this case, the magnitude of $\vec{\boldsymbol{n}}_{\boldsymbol{vi}}$ is multiplied by the sensor's directional response $\vec{\boldsymbol{u}}_{\boldsymbol{i}}$.

In contrast, **intrinsic noise** is defined as noise which is non-acoustic in nature, i.e. intrinsic to the sensor and its electronics. This noise encompasses effects such as amplifier electronic noise and quantization noise. As a result, it is not affected by the sensor's directionality, and thus the standard deviation of any component of $\vec{\boldsymbol{n}}_{\boldsymbol{vi}}$ is σ_{nv}. For extrinsic noise, the covariance matrix is diagonal with the diagonal elements being $\vec{\boldsymbol{u}}_{\boldsymbol{i}}\sigma_{nv}$. Like the pressure SNR, the velocity SNR is defined as

$$SNR_v = \frac{\sigma_s}{\sigma_{nv}} \qquad 49$$

To quantify the emergence of a Green's function peak from the surrounding noise, a peak signal-to-noise ratio (PSNR) is defined as the peak correlation value to three times the correlation tail standard deviation. Using this definition and substituting Equation 45 gives

$$SNR_{uv} = \frac{\max C_{uv}(\tau)}{3 \cdot \mathrm{Std}[C_{uv}(|\tau| \gg 0)]} = \frac{\max C_{uv}(\tau)}{3} \sqrt{\frac{2B_w T_r}{C_{uu}(0) C_{vv}(0)}} \qquad 50$$

An expression for the signal power, $C_{ii}(0)$, is obtained in a very similar way to Equation 29, and will be equal to

$$C_{ii}(0) = D_i \sigma_s^2 + N_v \sigma_{ni}^2 \qquad 51$$

where $D_i = \frac{1}{4\pi} \int_S l_i^{\,2} \, \mathrm{d}S$, is the integral of the sensor's directionality over the unit sphere (see Equation 29); N_v is either equal to unity for intrinsic noise, or equal to D_i for extrinsic noise; and σ_{ni}^2 is the noise power of either pressure or velocity noise, depending on the index *i*.

Finally, to represent the gain of a velocity correlation to a pressure correlation, define the ratio of their respective SNR's as

$$SNR_{v/p} = \frac{SNR_{vv}}{SNR_{pp}} \approx \frac{C_{pp}(0)}{C_{vv}(0)} = \frac{1 + SNR_p^{-2}}{D_v + N_v SNR_v^{-2}} \qquad 52$$

The approximation in Equation 52 stems from the pressure and velocity correlation peaks being nearly identical (see discussion in 3.2.5). Note that $D_p = 1$ and $D_v = 1/3$ in the case of ideal vector sensors in isotropic ambient noise. Substitution of definitions in Equations 47 and 49 result in the right-hand expression in Equation 52. To confirm the approximation made in Equation 52, the pressure and velocity SNR's as defined in Equation 50 were computed across a range of possible bandwidths and separation distances and the ratio is plotted in Figure 37.

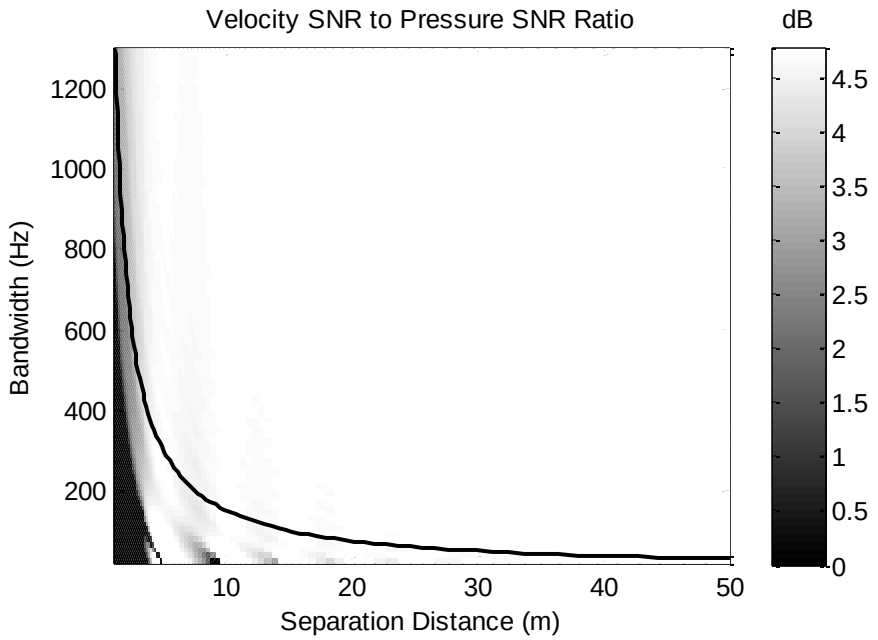


Figure 37: Ratio of velocity SNR to pressure SNR vs. separation distance and bandwidth. $SNR_p = SNR_v = 1$, band = [150, 150+Bw] Hz, sound speed = 1500 m/s. Solid black line denotes area below which the bandwidth and separation distance are such that no correlation peak is discernable.

As evidenced by the constant ratio of velocity SNR to pressure SNR plotted in Figure 37, there is no considerable difference in the correlation peak (i.e. numerator of Equation 50) for pressure or velocity. The velocity outperforms pressure by 4.77 dB (3x) only due to the decreased signal power in the velocity autocorrelation. As such, it can be assumed the performance of velocity correlations will be improved across all reasonable ranges and bandwidths (such that a peak is resolvable in the first place). Thus, plotting $SNR_{v/p}$ as a function of the pressure and velocity SNR's will show a regime when velocity correlations outperform pressure correlations (see Figure 38 and Figure 39).

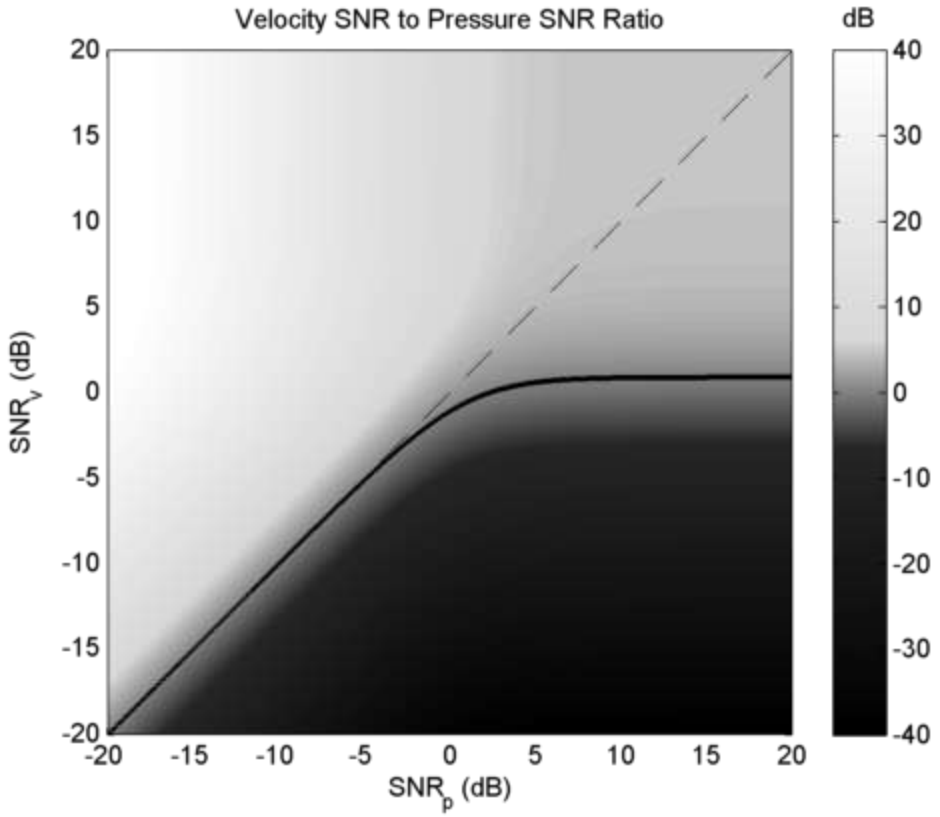


Figure 38: Ratio of velocity correlation PSNR to pressure correlation PSNR with varying amounts of intrinsic noise on pressure and velocity channels, quantified by SNR_p and SNR_v.

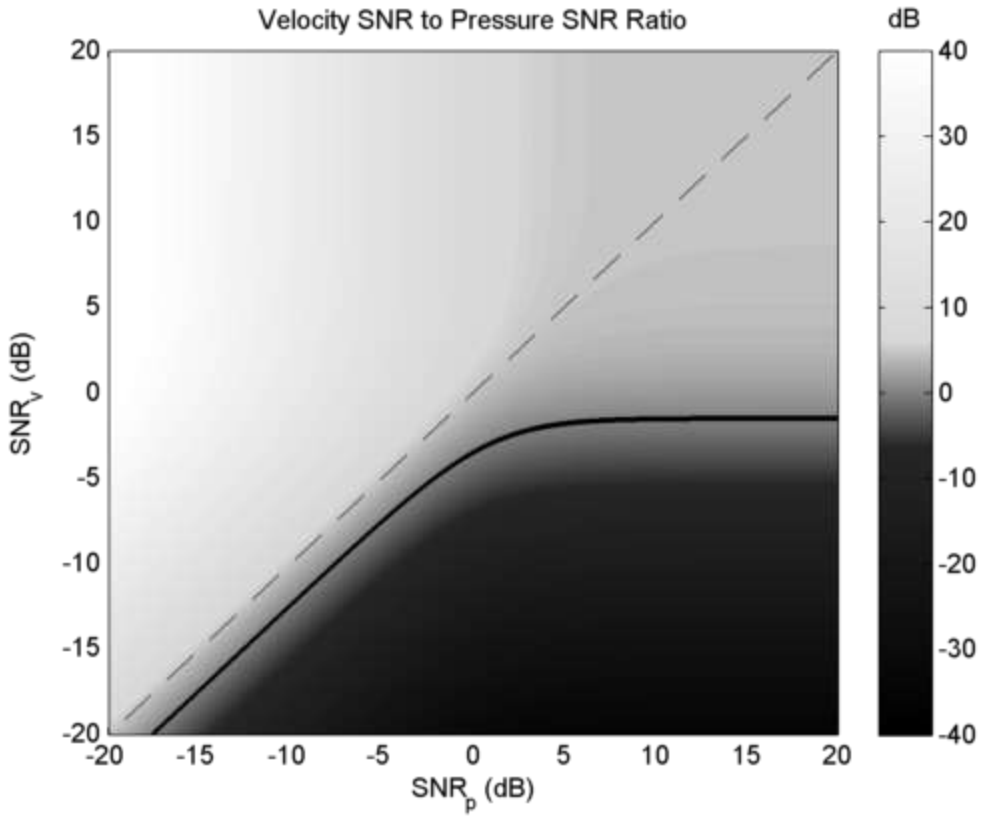


Figure 39: Ratio of velocity correlation PSNR to pressure correlation PSNR with varying amounts of extrinsic noise on pressure and velocity channels, quantified by SNR_p and SNR_v.

As seen in the intrinsic noise case (see Figure 38), velocity correlations outperform pressure correlations for equal noise levels (dashed line). If the velocity SNR exceeds ~1 dB, it will outperform pressure even if the pressure is noiseless. Only in extreme velocity noise relative to pressure (bottom-left quadrant) will a pressure correlation outperform velocity. The results are even more beneficial in the extrinsic noise case (see Figure 39). Velocity correlations outperform pressure by a factor of 3 for equal SNR (dashed line), since both ambient acoustic and local-acoustic noise interference is reduced by the sensor directionality. Above a velocity SNR of approximately -2 dB, it will always outperform pressure. In low velocity SNR cases, the pressure SNR must be at least 5 dB greater for pressure correlations to significantly outperform velocity correlations.

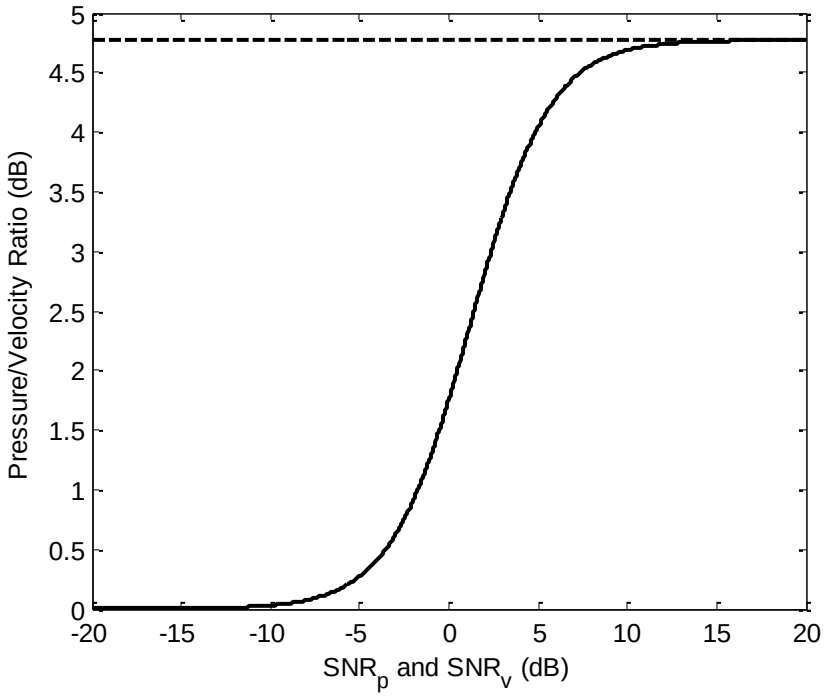


Figure 40: Ratio of velocity correlation PSNR to pressure correlation PSNR with equal amounts of noise on pressure and velocity channels, denoted SNR_p and SNR_v. Intrinsic noise model is shown as a solid line and extrinsic noise model is shown as dashed line.

For the case where the pressure and velocity noise levels are the same, the ratio of the resulting $SNR_{p/v}$ is plotted in Figure 40. The dashed line is the extrinsic noise reference of 3x reduction in velocity noise (4.77 dB). The solid shows the variation for the intrinsic noise case. For very low SNR, the intrinsic noise dominates any acoustic noise, thus both pressure and velocity perform equally. As the intrinsic noise level is reduced, and the SNR increases, the gain of the velocity correlations approaches the maximum, where the intrinsic noise is negligible in relation to the acoustic noise. Thus, the velocity sensor achieves a 3x gain due to its decreased sensitivity to broadside ambient noise sources.

In addition to the relative SNR as a method of comparing velocity and pressure performance, the required time-bandwidth product (TBWP) can be compared. Given the peak SNR described in Equation 50 should be greater than 0 dB (i.e. the peak and noise amplitudes are equal), the TBWP can be solved as

$$(B_w T_r)_{uv} \geq \frac{9C_{uu}(0) \cdot C_{vv}(0)}{2(\max C_{uv}(\tau))^2} \tag{53}$$

where the values in Equation 51 can be substituted in the numerator to yield

$$(B_w T_r)_{uv} \geq \frac{9(D_u\sigma_s^2 + N_v\sigma_{nu}^2)(D_v\sigma_s^2 + N_v\sigma_{nv}^2)}{2(\max C_{uv}(\tau))^2} \tag{54}$$

Noting that the correlation $C_{uv}(\tau)$ is proportional to the noise autocorrelation amplitude (see Equation 43), a normalized correlation can be defined as

$$\bar{C}_{uv}(\tau) = \frac{C_{uv}(\tau)}{\sigma_s^2} \tag{55}$$

which is calculated with a normalized noise autocorrelation

$$\bar{C}_{ss}(\tau) = \frac{C_{ss}(\tau)}{\sigma_s^2} \tag{56}$$

which has a maximum value of one. Thus, the expression of Equation 54 can be rewritten in terms of the channel signal-to-noise ratios as

$$(B_w T_r)_{uv} \geq \frac{9(D_u + N_v SNR_u^{-2})(D_v + N_v SNR_v^{-2})}{2(\max \bar{C}_{uv}(\tau))^2} \tag{57}$$

It is worth noting that the maximum of the correlation is proportional to c/d (see Equation 39). This means that the required TBWP is proportional to $(d/c)^2$. Thus, increasing the distance by a factor of two means the required averaging time increases four-fold.

Simplifying Equation 57 for the pressure and axial velocity components and dividing through by the (assumed fixed) bandwidth, the required averaging times are given by

$$(T_r)_{pp} \geq \frac{9\left(1 + SNR_p^{-2}\right)^2}{2B_w\left(\max \bar{C}_{pp}(\tau)\right)^2} \tag{58}$$

and

$$(T_r)_{vv} \geq \frac{(1 + 3N_v SNR_v^{-2})^2}{2B_w(\max \bar{C}_{vv}(\tau))^2} \tag{59}$$

These results indicate that for equal SNR on pressure and velocity channels, and if the noise is extrinsic (i.e. $N_v = 1/3$), the required averaging time is 9 times higher for pressure correlations than velocity correlations. This is because the pressure correlations have 3 times the standard deviation in their tails, and the standard deviation is proportional to the inverse of the square root of TBWP. The relationship is not as straightforward for differing amounts of SNR on pressure and velocity channels, or if the noise source is intrinsic. For this reason, simulations are carried out to plot the required averaging time as a function of the channel SNR.

Simulations employ Equation 34 to obtain the maximum value of $C_{uv}(\tau)$ seen in the denominator of Equations 58 and 59. The expressions for the noise autocorrelation are obtained from Equation 43, normalized by dividing by σ_s^2. The resulting averaging times

are plotted for a reference distance of 10 meters and reference bandwidth of 1000 Hz (200-1200 Hz) (see Figure 41). The reference distance was chosen such that distinct peaks are present in the noise correlation for the chosen bandwidth.

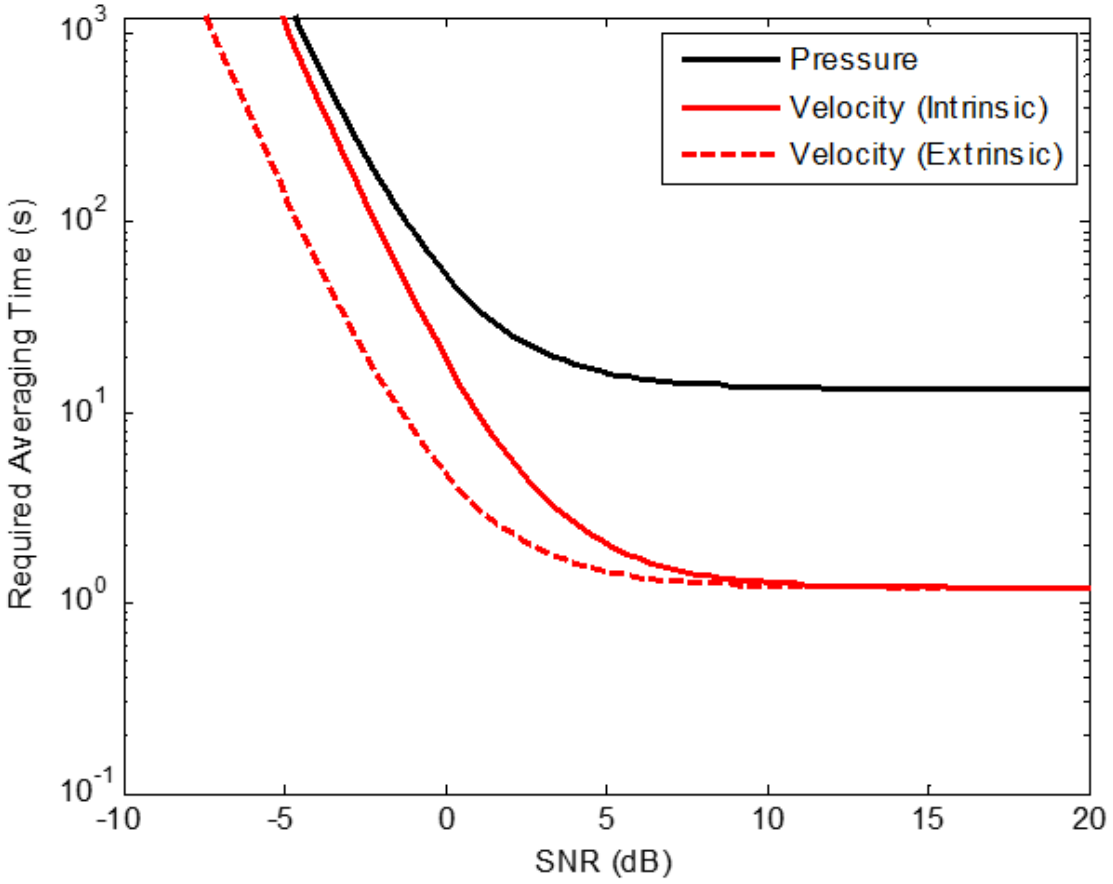


Figure 41: Required averaging time for reference separation of 10 m, reference band of 1000 Hz. Times are shown for pressure correlations (black solid), velocity correlations with intrinsic noise (red solid) and extrinsic noise (red dashed).

The lines shown in Figure 41 give the averaging time required for 10-meter separation distance and 1000 Hz bandwidth, for the pressure correlation (black line), and velocity correlation (red lines). Both intrinsic (solid) and extrinsic (dashed) noise cases are shown for velocity. Pressure results are unaffected by the source of noise, as both intrinsic

and extrinsic noise is received at equal power for a hydrophone. The results show that the averaging time required is less for velocity correlations than pressure correlations for equal SNR. Equivalently, horizontal lines on the graph indicate the difference in SNR required for velocity and pressure to require equal averaging times. For low noise cases, pressure correlations require about 10x the averaging time than velocity. This is due to the factor of 9 previously mentioned, plus the slightly increased correlation peak values obtained by velocity correlations.

As was previously mentioned, the required TBWP is proportional to $(d/c)^2$, so to compute what the required averaging times are from Figure 41 for a different separation distance than its reference (10 m), the time simply needs to be scaled by a factor

$$F_d = \left(\frac{d}{d_{ref}}\right)^2 \tag{60}$$

where d is the actual separation distance. Likewise, the required averaging time is linearly proportional to the inverse of the bandwidth, so results from the reference plot in Figure 41 can be obtained by scaling by the factor

$$F_B = \frac{B_{ref}}{B_w} \tag{61}$$

where B_w is the actual bandwidth. For example, the required averaging time for a velocity correlation with intrinsic noise at 0 dB is about 10 seconds for the reference case. At a distance of 20 meters and a bandwidth of 500 Hz, the required averaging time would be 80 seconds (i.e. $10 \cdot \left(\frac{20}{10}\right)^2 \cdot \frac{1000}{500}$).

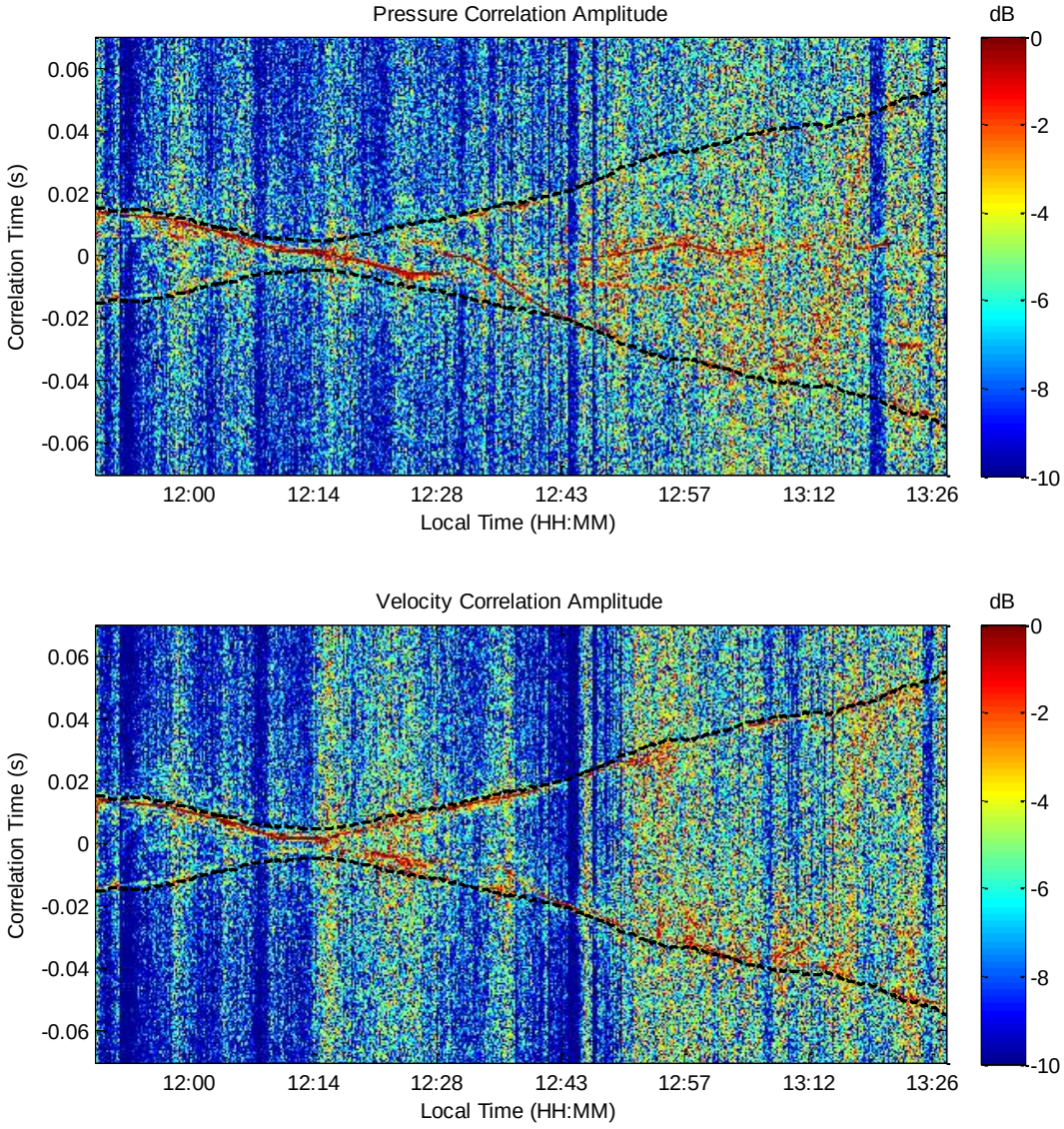
3.3.2 Experimental Performance

As predicted by the theory presented in Section 3.2, both the pressure and velocity ambient correlation peak should be nearly equal in magnitude. The theory presented in Section 3.3.1 dictates that the standard deviation of correlation tails is lower for velocity correlations relative to pressure. To examine these traits experimentally, ambient noise correlations were obtained from the October 30 experimental data collected in the Long Island Sound (see Figure 42). The sensors were at fixed depths of 20 and 26 meters, and their separation distance varied between approximately 12 and 80 meters. Data from each sensor was preprocessed with the filter, clip, and then whiten steps described in Section 2.3.4.

Figure 42: Deployment location of sensors during October 30th 2016 experiment

Correlations of pressure and velocity data were obtained from successive 1 second windows over a period of approximately 2 hours during which the research vessel was

silent and more than 4 km removed from the array. Their amplitudes, obtained from the magnitude of the Hilbert transform, were smoothed over the experiment run time using a 10 second moving average window (see Figure 43a-b). The ratio of the pressure correlation amplitude to velocity correlation amplitude is plotted as a function of the experiment run time in Figure 43c.



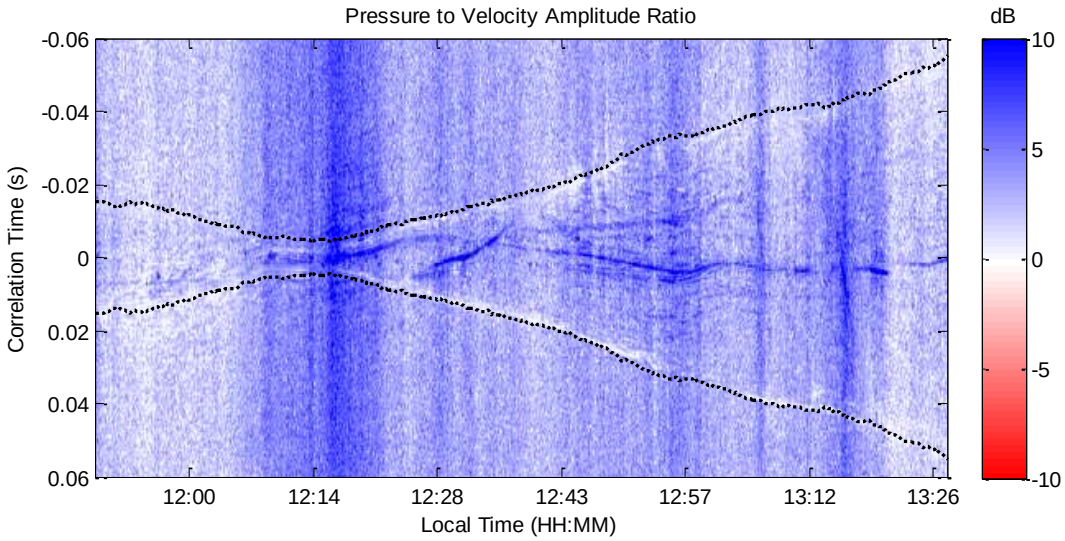


Figure 43: a) Pressure correlation amplitudes, normalized per-correlation. b) Axial velocity correlation amplitudes, normalized per-correlation c) Experimental pressure to velocity correlation amplitude ratio. Amplitudes are obtained from magnitude of Hilbert transform, dB values obtained with 10 $\log_{10}$.

The black dotted lines in Figure 43 were obtained by estimating the sensors' separation distance from GPS coordinates and dividing by an assumed sound speed of 1500 m/s to obtain the propagation delay. Note that along this line, the correlation amplitude ratio is approximately 0 dB, corroborating the finding that ambient noise correlation peaks are nearly identical as predicted in Section 3.2.5. Another important feature of Figure 43 is the predominantly blue color seen in the correlation amplitude ratio, meaning the velocity correlation noise is generally lower than that of the pressure correlation. To quantify these findings, the ratio of pressure and velocity standard deviations in the correlation tails (defined by the region $|\tau| > 0.1$ seconds) is plotted in Figure 44a. The ratio for the standard deviation in the correlation interior is plotted in Figure 44b. The interior region is defined by $|\tau| < 0.8 \cdot d/c$, where d is the GPS-estimated separation distance at the correlation center time.

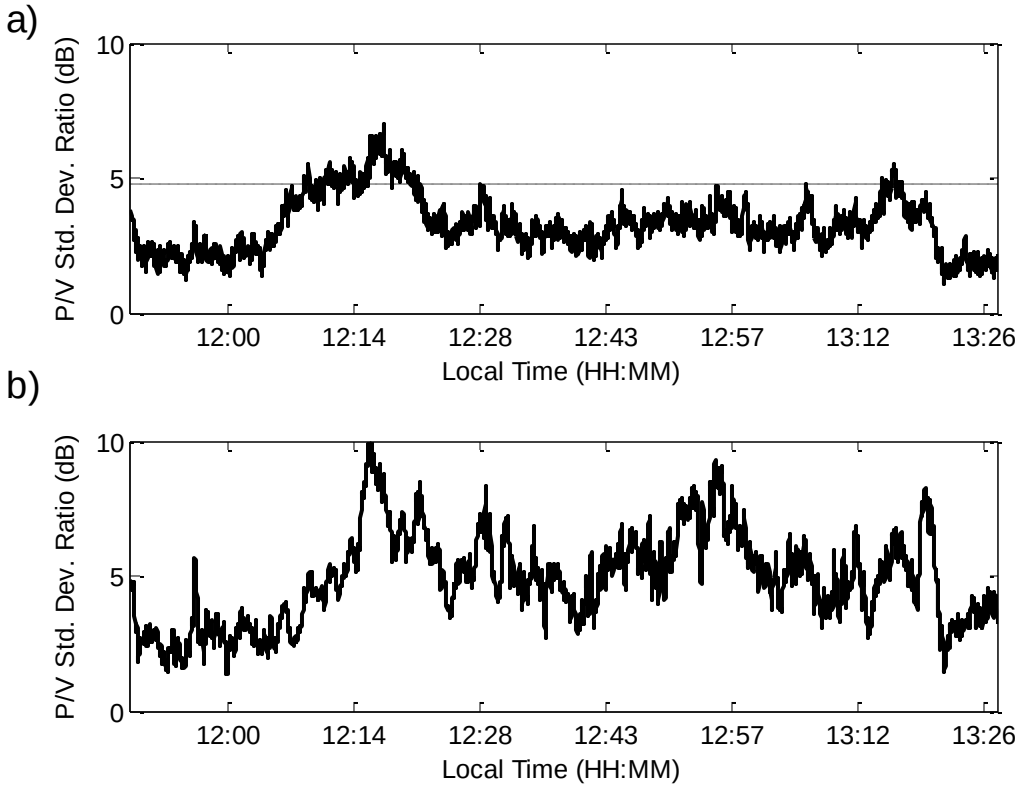
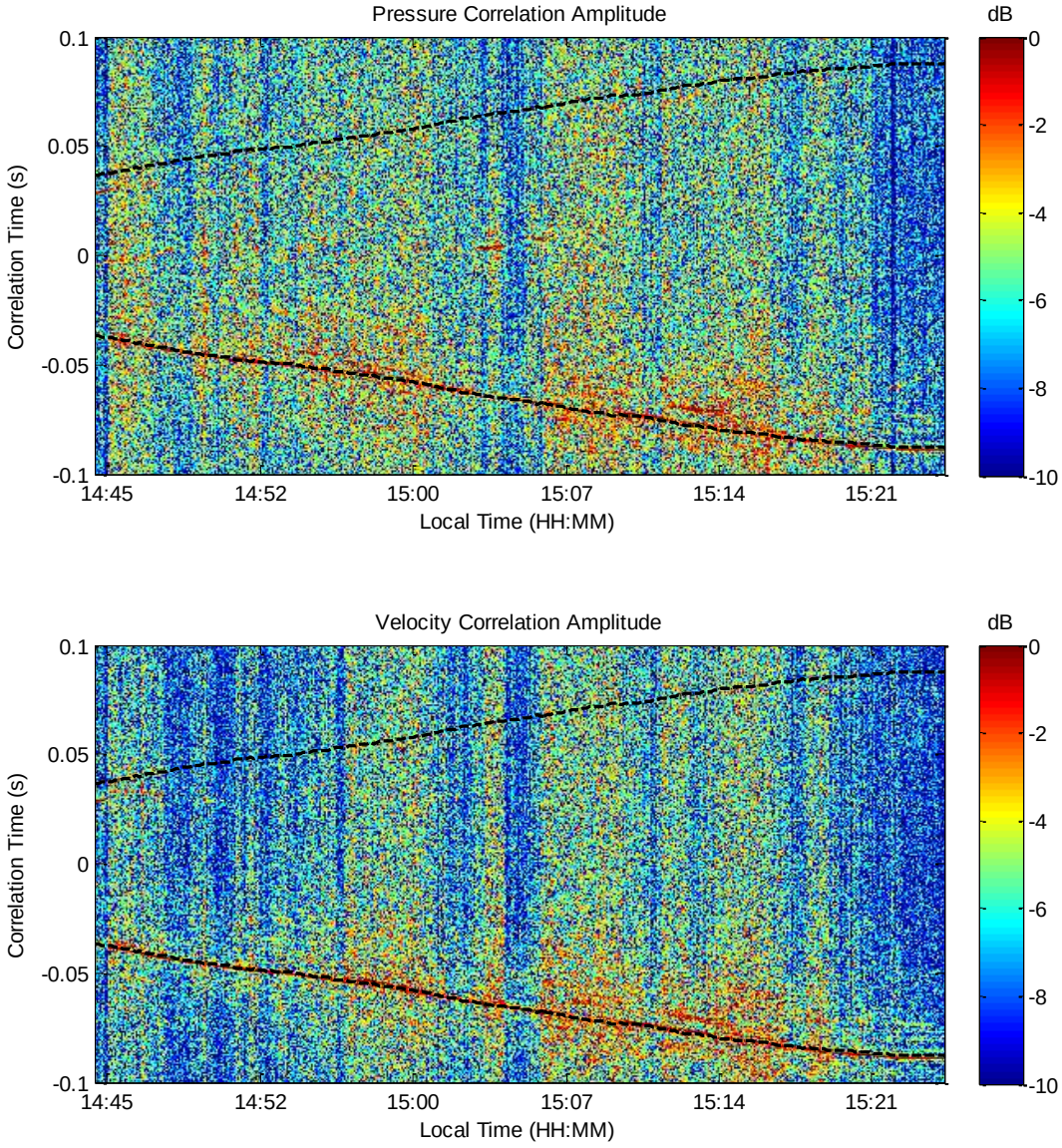


Figure 44: a) ratio of P/V standard deviation in the correlation tails ($|\tau| > 0.1$) with the equal-noise theoretical value (4.77 dB) in dashed line. b) P/V standard deviation ratio of the interior correlation ($|\tau| < 0.8 \cdot d/c$).

Typical values of the pressure to velocity noise amplitude ratio were between 3 and 5 dB, well-predicted by the theory of Section 3.3.1, assuming the intrinsic or extrinsic pressure and velocity noise SNR's were roughly equal (see Figure 44a). Also of note is the effect of interfering sources (mainly tugboats and ferries operating in the area) as streaking dark lines in the correlations of Figure 43c, and the increased ratio of Figure 44b. The increase in the amplitude ratio becomes more pronounced as the sources passed the array broadside (correlation lag of approximately zero), and when the sensors were at their closest separation distance. Thus, the velocity channel dipole directionality limits the effect

of broadside interferers, whereas a pressure channel remains sensitive to broadside interferers, hence the increased correlation amplitude ratios seen in Figure 44b.

Repeating the analysis for the second quiet time observed from 14:45 – 15:25 local time, the correlation amplitudes are plotted in Figure 45a/b. The ratio of pressure to velocity amplitude is plotted in Figure 45c.



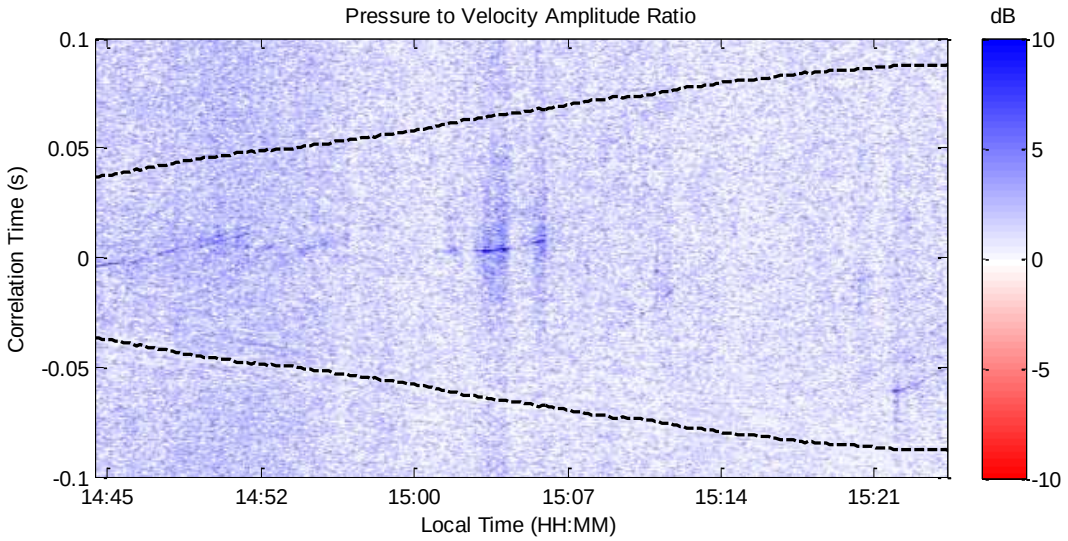


Figure 45: a) Pressure correlation amplitudes, normalized per-correlation. b) Axial velocity correlation amplitudes, normalized per-correlation c) Experimental pressure to velocity correlation amplitude ratio. Amplitudes are obtained from magnitude of Hilbert transform, dB values obtained with 10 $\log_{10}$.

The ratio is approximately 0 dB for the strong peak in the negative time delays, but is no longer equal to 0 dB for the positive side. This could be a result of insufficient averaging time to develop the ambient noise correlation peak. The negative time delay peak is not a result of ambient noise, but of a far-field interfering source identified in the AIS record. For such a source, the pressure and particle velocity signal received by the sensors should have been equal in amplitude after the velocity data was transformed into pressure units via multiplication by the specific acoustic impedance. Despite this shortcoming, the ratio of standard deviation in the correlation tails and interior can be compared to quantify the decrease in velocity correlation noise (see Figure 46). Again, the region defining the tails is $|\tau| > 0.1$ and the interior is defined by $|\tau| < 0.8 \cdot d/c$.

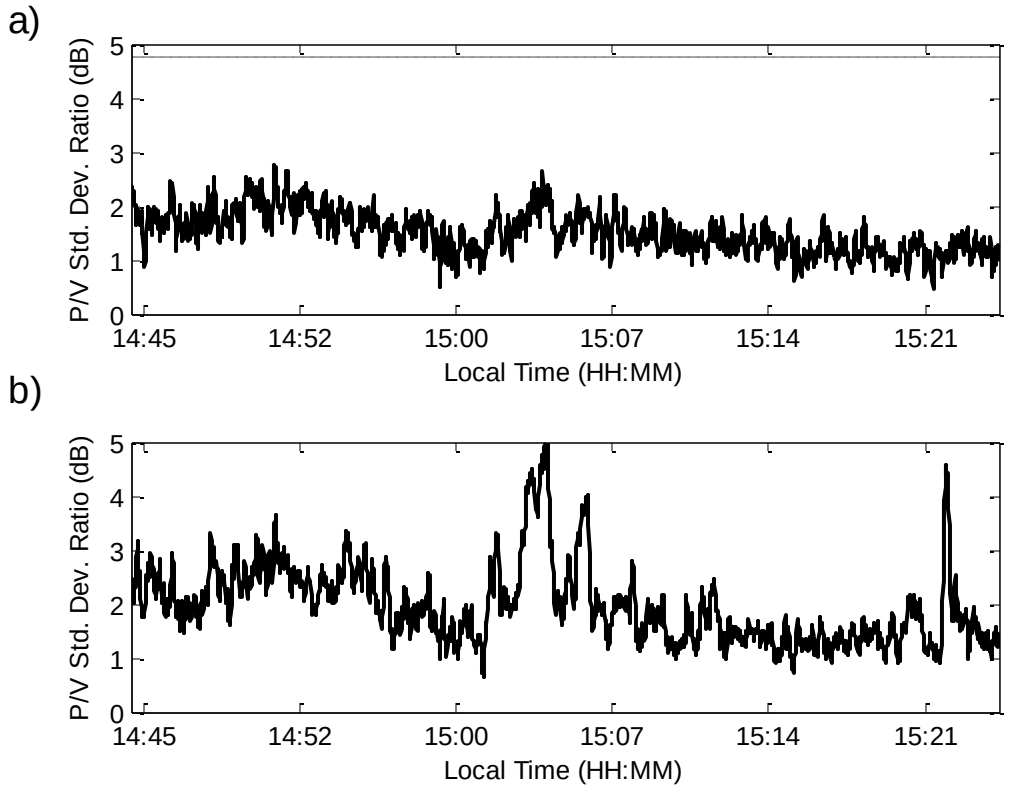


Figure 46: a) ratio of P/V standard deviation in the correlation tails ($|\tau| > 0.1$) with the equal-noise theoretical value (4.77 dB) in dashed line. b) P/V standard deviation ratio of the interior correlation ($|\tau| < 0.8 \cdot d/c$).

The ratio of pressure to velocity correlation standard deviation in the tails is about 2 dB, indicating that there is not equal noise in the pressure and velocity recordings. A gain of 4.77 dB is expected if the pressure and velocity noise are equal, however a gain of 2 dB is achieved if the velocity extrinsic noise is slightly higher than that of pressure (see Figure 39). The interior correlation ratio is slightly higher due to the occasional broadside interfering source, but also averages about 2 dB. This is likely because the interior correlations are not as corrupted with interfering sources.

3.3.3 Conclusion

Noise variance estimates were added to the expected value expressions developed in Section 3.2 to obtain expressions for the SNR of a pressure or velocity ambient noise correlation. The ratio of the velocity SNR to the pressure SNR was examined under varying separation distance and bandwidths, and found to be constant across both. This ratio was then examined for varying amounts of pressure and velocity channel noise, for both the intrinsic noise and extrinsic noise cases. Velocity ambient noise correlations were found to outperform pressure correlations in all cases except where a significant amount of noise is present on both channels (< 0 dB SNR) and the velocity noise is significantly higher than pressure (> 5 dB difference).

3.4 Conclusion

Correlations of vector sensor data in an isotropic ambient noise field were derived and expressed in a simple convolutional framework. These expressions indicate the pressure and velocity correlation expectations are nearly identical for the typical frequency bands used experimentally. Furthermore, the development showed the importance of using the correlation derivative when attempting to locate the Green's function peak. In addition to examining the expectation of the correlations, the variance was analyzed to quantify the SNR of the correlation process in the presence of noise. Pressure and velocity correlations were compared using their respective PSNR's as well as their required averaging times for Green's function extraction. In most cases, except where extreme velocity noise is present, a velocity correlation outperforms a pressure correlation.

CHAPTER 4
ARRAY ELEMENT LOCALIZATION OF MOBILE VECTOR SENSOR ARRAYS USING AMBIENT NOISE CORRELATIONS

4.1 Introduction

Ambient noise exists in any acoustic environment to some extent, and can be exploited as a source of information about the environment or the sensors placed within. It has been shown theoretically and experimentally that correlations of ambient noise across separated sensors contains the Green's function (also referred to as the Channel Impulse Response) between them (Lani et al., 2013; Leroy et al., 2012; Naughton et al., 2016; Roux & Kuperman, 2004; Karim G. Sabra et al., 2005b; K. G. Sabra et al., 2005). Encapsulating both environmental and sensor placement information, the Green's function defines the propagation of sound between two separated sensors. Exact knowledge of the Green's function is not generally guaranteed in every acoustic environment, and its estimation has many uses ranging from seismology, structural monitoring, ocean tomography, fathometry, sub-bottom profiling, or simply inter-sensor distance estimation (Godin et al., 2010; Harrison & Siderius, 2008; K. G. Sabra et al., 2005; Martin Siderius et al., 2006).

In the case of inter-sensor distance estimation, the Green's function specifies the travel time of sound propagating along the path between sensors. Assuming the propagation speed is known and the path is direct (i.e. straight), the distance between sensors can be inferred from the measured travel time. Knowledge of the distance between sensors becomes critical for applications where the sensors are mobile and their locations carry a great deal of uncertainty. It is also highly important to have accurate knowledge of sensor locations for coherent processing of sensor array data (Culver & Hodgkiss, 1988; S.

E. Dosso et al., 2004). For example, localizing a source accurately requires the inter-sensor distances to be known within fractions of a wavelength of the source signal (0.1 – 1 m for 1500 – 150 Hz in the ocean, using a fraction of 10%). Common methods of measuring sensor locations include using fixed arrays, global positioning system (GPS), or acoustic transponders ("pingers") (Paull et al., 2014).

In a fixed array, the sensor locations are measured before deployment, and assumed to remain fixed throughout their use. In some instances, even an array which is "fixed" can suffer from undesirable sensor perturbations (S. E. Dosso et al., 2004; S. E. E. Dosso, Gordon R., 2006; Morley, Dosso, & Chapman, 2009). Fixed arrays also suffer from not being adaptable to changing mission intent or environments. A mobile array offers the ability to adaptively reposition, add or remove elements as dictated by changing requirements. If the mobile array has access to GPS signals, their positions can be obtained with uncertainties between 0.7 and 1.5 meters (Grimes, 2008), which resides on the threshold of usable accuracy for underwater applications. However, GPS signals do not propagate to any significant depth underwater, and thus sensors which remain submerged for extended periods of time must rely on dead reckoning or acoustic transponders to obtain their positions (Paull et al., 2014). Dead reckoning iteratively updates the estimate of sensor location through knowledge of the sensor orientation and speed, however it suffers greatly from measurement error accumulation and perturbations, where errors of tens to hundreds of meters can be common (Paull et al., 2014). Acoustic transponders, or "pingers", can remedy this problem by emitting signals from beacons whose locations are known. Within this area of sensor localization, there exist many different modes of operation. Long baseline (LBL) modes use fixed beacons, typically arranged along the boundary of the

operation area and moored to the seafloor. Their accuracy, typically 1 m or better, comes at the cost of a restricted operational area, high setup cost, and reduced secrecy (Paull et al., 2014). GPS-tracked surface beacons may also be used with comparable performance to LBL systems, but still suffer from limited operational area and significantly reduced secrecy. Short baseline (SBL) systems use beacons or active transponders in a much smaller array, typically small enough to be mounted to a vessel or towed behind it (Paull et al., 2014). They can ascertain the distance and bearing to the sensors, which can be localized if the position of the SBL system is known. Performance of SBL systems can approach that of LBL systems while offering the benefit of a mobile tracking area (Paull et al., 2014). However, the cost and loss of secrecy of such a system is far worse than the more passive LBL systems. If a sensor's location underwater is to be accurately determined underwater in a flexible, cheap and clandestine way, acoustic transponders may not be a suitable solution. This leaves ambient noise correlations as one of the few viable solutions for localizing a sensor underwater in a totally covert manner.

The presence of ambient noise in the ocean can be exploited almost as a LBL transponder system, where the "beacons" are distributed throughout the entire environment (Roux, Sabra, Kuperman, & Roux, 2005). The ambient noise source may be wind-driven surface noise (high frequency), ambient shipping noise (mid-frequency), or geologic/seismic activity (low frequency), to name a few. In any case, the sound from these sources will eventually propagate through all possible paths between two separated sensors. Thus, correlations of data collected from separated sensors ought to carry some information about the propagation delays experienced by each path. When averaged over long periods of time, an estimate of the Green's function between sensors emerges from

the ambient noise correlations, or more specifically from its derivative (Roux & Kuperman, 2004; Karim G. Sabra et al., 2005b). For fixed hydrophones, this has been experimentally verified (Fried et al., 2008; Lani et al., 2013; Leroy et al., 2012; Roux & Kuperman, 2004; Karim G. Sabra et al., 2005b), and requires averaging times on the order of minutes. In the case of drifting sensors, the same procedure should work in theory, assuming the sensors are quasi-stationary over the averaging duration (i.e. assumed stationary for the averaging duration, then moving to new locations for the next averaging duration). However, this either limits the allowable sensor drift speed or averaging duration. Conservatively, the direct arrival of the Green's function should not change by more than 10% of the center frequency's wavelength over an averaging period (0.1 – 1 m for 1500 – 150 Hz). If averaging lengths of one minute or more are to be used, drift speeds must be incredibly slow (6 – 60 m/hr for frequencies between 150 – 1500 Hz). Alternatively, the averaging time can be shortened to accommodate higher drift speed (2 – 20 s for 180 m/hr over 1500 – 150 Hz bands). Unfortunately, at such short averaging times, a reliable estimate of the Green's function may not have fully emerged from the ambient noise correlations.

A proposed method for improving Green's function extraction in the presence of a rapidly fluctuating medium employs a stochastic search algorithm to search for peaks hidden in noise (Woolfe, Sabra, et al., 2015). As previously stated, drifting sensors constrain the allowable averaging time for the sensors to remain quasi-stationary. However, using a shorter averaging time than required means the Green's function peaks will be buried in correlation noise. Accepting this limitation, and assuming the time delays of each peak is known *a priori*, the correlation peaks can be shifted to a common time. Upon summing multiple short-time correlations with aligned peaks, the peaks will coherently

sum, while the noise will incoherently sum. The result would be identical to a long-time correlation of stationary sensors, displaying a prominent correlation peak centered at the origin. For any set of shifts other than the true time delays, the summed result will be smaller (see Figure 47). This method has been experimentally verified as a proof-of-concept for the case of stationary hydrophones with changing speed of sound in a low frequency regime (1 – 40 Hz) (Woolfe, Sabra, et al., 2015). It has yet been demonstrated on a higher frequency regime for drifting sensors. Furthermore, the stochastic search was not constrained or optimized to produce more accurate solutions.

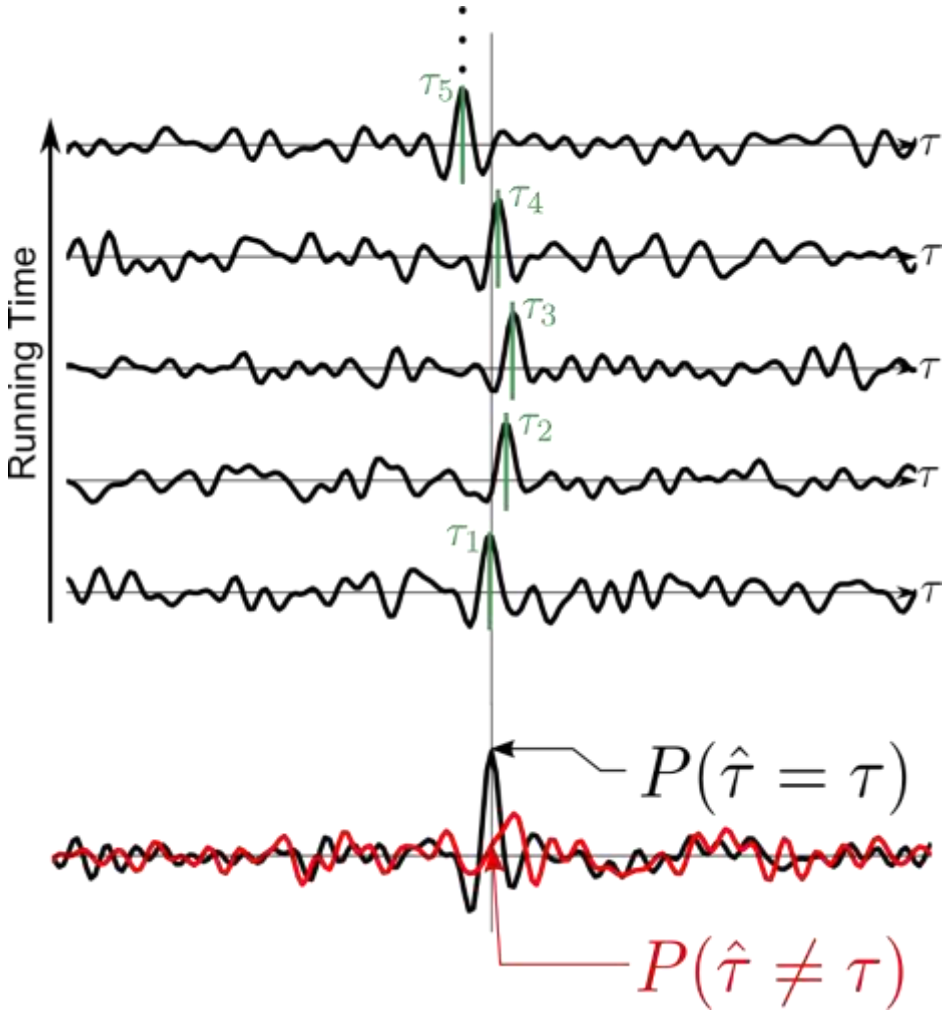


Figure 47: Correlation ensemble with hidden peaks (exaggerated for demonstration) having time delays τ_i. The shifted-and-summed correlation when the estimated time delays $\hat{\tau}$ match the true time delays τ is shown in black on the bottom. An example sum is shown when the time delays are not correct (red).

Thus, the goal of the stochastic search is to estimate the locations of hidden peaks in many short-time correlations. If the stochastic search chooses the correct time delays of each peak, the shifted and summed result will have a sharp peak and reduced noise. Otherwise, a poor estimate results in the correlation peaks not adding coherently, and the resulting sum being smaller (see Figure 47). In this way, the stochastic search repeatedly guesses the time delays of hidden Green's function peaks within noise, and evaluates the fitness of any guess using the shift-and-sum metric (Woolfe, Sabra, et al., 2015). Many

stochastic search methods exist, such as simulated annealing, particle filters, and genetic algorithms, to name a few (Mitchell, 1999). They excel at computationally complex problems such as this, where doing an exhaustive search of the entire space is computationally prohibitive. However, they also rarely find a global optimum, usually settling on a local optimum solution. For example, consider the correct estimation of Green's function peak locations, except a single estimate is incorrect. The shifted and summed correlation would still appear to be a sharp peak with relatively large amplitude. If the stochastic search never guessed the fully-correct solution, it would never find the slightly more optimal sum and simply return the "local optimum" solution. Having one misplaced estimate of a large group of correct estimates might not be a problem, since physical limitations dictate a sudden change in the distance between sensors is not possible. Thus, filtering of the stochastic search results could be performed to mitigate these problems.

An alternative approach to avoiding erroneous estimates is to explicitly inform the stochastic search of physical constraints. A simple method of doing this would be to constrain the solution space such that the difference between successive time delays is physically admissible. Another possible method could use parameterized motion models and allow the stochastic search to vary the model parameters. Both methods are valid, and Section 4.2.2 will examine the benefits and weaknesses of each in identifying the correlation peaks, leading to robust Green's function estimates.

As discussed before, knowledge of the Green's function propagation delay and sound speed yields a measurement of the distance between sensors. The extraction of Green's function delays from short-time correlations can be augmented using both vector

sensor correlations over using hydrophones only (see results from Chapter 3) as well as using a stochastic search algorithm. Assuming this process is performed for all pairs of sensors, the locations of each can be determined, to within an arbitrary translation and rotation (proper rigid translation). Such an unknown translation needs to be estimated using other means, or simply ignored if subsequent source localization does not require absolute accuracy. Leveraging ambient noise in this way could also augment the performance of any of the previously discussed methods (LBL, SBL, GPS, dead reckoning, etc.), thus eliminating the problem of the array's unknown proper rigid translation.

There exist many ways in which the locations of the array elements can be determined, a process hereafter referred to as Array Element Localization (AEL). Each method attempts to invert the non-linear function which produces inter-sensor distances from the sensor locations. As the function is non-linear, no closed-form inverse solution exists. The first and simplest method of inverting such a function is to linearize it about an operating point using a Taylor series expansion. The linearization is used to solve for a solution, and this process is repeated until the error in the solution is acceptably small (S. E. Dosso, Fallat, Sotirin, & Newton, 1998). However, this method does not take measurement error or *a priori* location error into consideration. Instead of a simple least squares optimization, a regularized cost function can be employed which considers either inter-sensor distance errors, prior sensor location errors, or both (S. E. Dosso et al., 2004). This method can achieve decent performance while also being rather simple to implement. Another method of AEL which can outperform the regularized least-squares method is to employ a Kalman filter which adds additional information/constraints given the past state of the sensor array. For example, the array can be assumed to freely drift with random

accelerations, and thus the next location of a sensor is likely to be in the direction it is drifting (Culver & Hodgkiss, 1988). This method has the drawback of applying a motion model to the array, which requires some knowledge of how the array moves and what accelerations it may experience. Furthermore, the method requires the random variables to be Gaussian, a constraint which is usually acceptable. If unacceptable however, a more advanced state estimation algorithm such as a particle filter could be used, but at the cost of increased complexity. For most situations, a regularized least-squares minimization will yield acceptable AEL results with minimum complexity.

This chapter will first introduce the stochastic search problem, then show simulation and experimental methodologies and results for Green's function extraction from ambient noise correlations in Section 4.2. Section 4.3 will introduce the AEL problem formulation and solution methods and compare their solutions for finding the sensor locations from measurements of the inter-sensor distances.

4.2 Sensor Separation Distance Estimation Using Stochastic Search

4.2.1 Stochastic Search Problem Statement

Consider two vector sensors separated by a time-varying distance *d*, in the presence of ambient noise (see Figure 48). The finite-time correlation estimate is given by

$$\hat{C}_{12}(\tau) = \frac{1}{T}\int_0^T s_1(t)s_2(t+\tau)\,dt \qquad 62$$

where $s_i(t)$ is the data recorded from sensor *i*, which may be either a pressure or velocity signal. Even an arbitrarily-steered beamformed signal could be used as the signal $s_i(t)$, for example from a vertical line array (Harrison & Siderius, 2008; Lani et al., 2013; Leroy et

al., 2012). Also, the time-derivative of a recorded signal may be used if the correlation derivative

$$\frac{d}{d\tau}\hat{C}_{12}(\tau) = \frac{1}{T}\int_0^T s_1(t)\frac{d}{d\tau}s_2(t+\tau)\,dt \qquad 63$$

is desired (Karim G. Sabra et al., 2005b). It is important to note the difference between the theoretical correlation derived in Section 3.2 (see Equation 23) and that of Equation 63, as the latter is only a finite-length approximation (see Section 3.3). As described in Section 3.3.1, the Green's function peaks may be buried in noise if the averaging time is insufficient. However, because the distance between sensors is now time-varying, the theoretical emergence of the Green's function from long time-averaged correlations is not guaranteed. Shorter time windows where the array can be considered *quasi-stationary* must be used, even if the resulting correlation shows no sign of Green's function peaks.

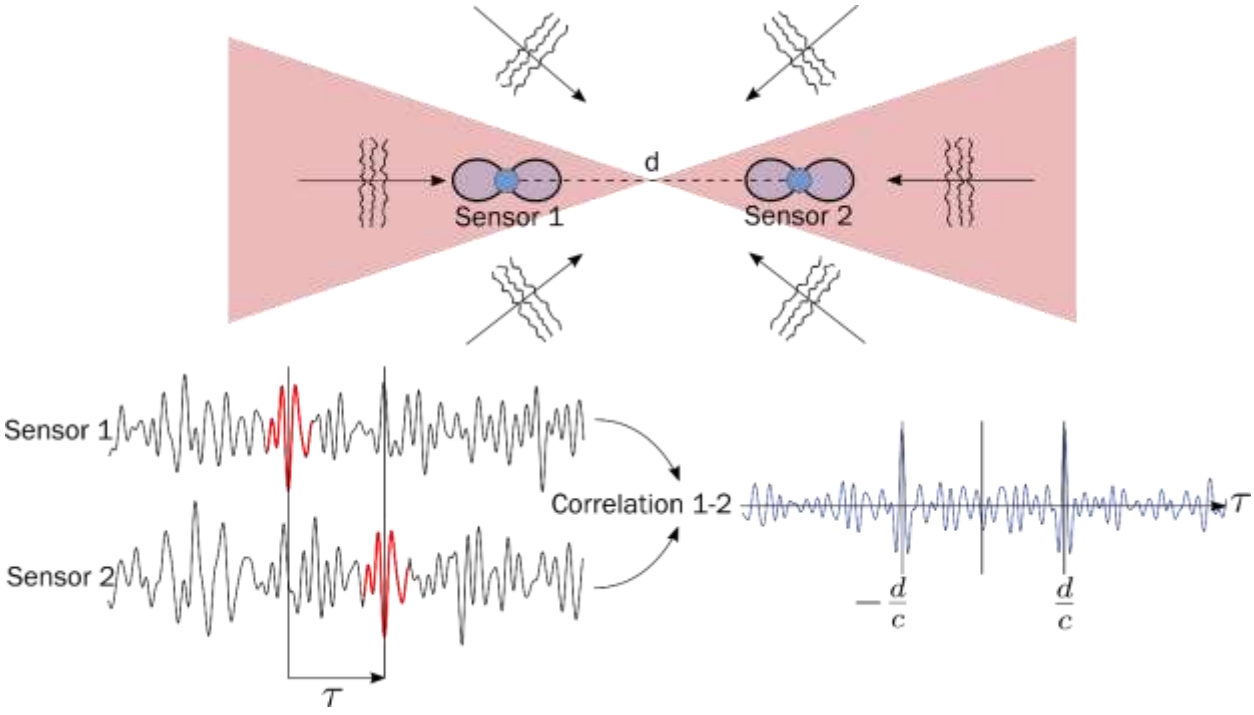


Figure 48: Ambient noise correlation theory for sensors separated by a distance *d*, where sources within the red cones support the emergence of correlation peaks. The correlation of data from sensor 1 and 2 will highlight source contributions such as that highlighted in red, producing two peaks at ± *d/c* given enough averaging time.

Thus, the primary challenge is to locate the Green's function peaks in the presence of noise. It is assumed that regardless of the quality of the correlated signals $s_i(t)$, the sensor motion is such that the averaging time is insufficient to develop clear correlation peaks. Define a peak signal-to-noise ratio (PSNR) as the ratio of Green's function peak to three times the correlation noise standard deviation

$$PSNR = \frac{\left|\hat{C}_{12}(\tau = d/c)\right|}{3 \cdot \mathrm{std}\left[\hat{C}_{12}(|\tau| \gg 0)\right]} \qquad 64$$

When the peak is equal in size to the 99% amplitude threshold of the correlation noise, PSNR will be 1 (0 dB). Thus, the regime suited to a stochastic search requires a PSNR less than 1.

To estimate the peak time delays, a multitude of short-time correlations will be needed to synthetically extend the averaging time. Consider an ensemble of short-time correlations

$$\hat{C}_{12}(\tau, t_i) = \frac{1}{T}\int_{t_i}^{t_i+T} s_1(t')s_2(t'+\tau)\,dt' \tag{65}$$

where the running time, t_i, can be a set of discrete times at which a correlation is obtained. A simple choice of running times could be $t_i = iT$ for integers $i = 0, 1, \ldots N_c - 1$, corresponding to a total of N_c correlations, each T seconds long with no overlap between successive correlation windows.

Now, if the correlation length T is chosen such that the sensor separation distance is approximately constant for $t_i \leq t \leq t_i + T$, each short-time correlation will behave as the theory in Section 3.2 predicts. Peaks should exist for the *i*-th correlation at the positive and negative time delay

$$\tau_i = \frac{d\left(t_i + \frac{T}{2}\right)}{c} \tag{66}$$

where the sensor separation distance is assumed to be a constant value equal to the separation distance at the center of the correlation window. In other words, the *i*-th separation distance is given by

$$d_i = d\left(t_i + \frac{T}{2}\right) \tag{67}$$

Applying the theory described in Section 3.2, the expected correlation ensemble will be described by

$$C_{12}(\tau, t_i) \approx \pm Y(\tau - \tau_i) \pm Y(\tau + \tau_i) \quad (68)$$

where $Y(\tau)$ is the function describing the peaks present in the correlation. When $C_{12}(\tau)$ is computed from the signal derivatives, $Y(\tau) \approx C_{ss}(\tau)$. This was the result presented in Section 3.2.6 for the ambient noise correlation derivatives, where the peaks are almost identical to the ambient noise process autocorrelation. For plain correlations (i.e. no time derivative), $Y(\tau)$ will be the integral of the ambient noise autocorrelation (e.g. see Section 3.2.5). More complicated functions for $Y(\tau)$ may also be used, if the appropriate theory is developed for more complicated correlation setups. In any case, the shape of the peak function $Y(\tau)$ should be known, the only unknown being its location within the noise.

To quantify the quality of an estimate of the peak locations, a shifted and summed correlation function

$$P(\hat{\boldsymbol{\tau}}, \tau) = \sum_{i=0}^{N_c-1} \hat{C}_{12}(\tau + \hat{\tau}_i, t_i) \quad (69)$$

is defined, where $\hat{\boldsymbol{\tau}} = [\hat{\tau}_0, \hat{\tau}_1, \ldots, \hat{\tau}_{N_c-1}]$ is a vector of the peak estimate locations. In other words, the function $P(\hat{\boldsymbol{\tau}}, \tau)$ shifts each short-time correlation by the estimated peak location for that correlation. It then sums all the shifted short-time correlations over the longer ensemble. If the hidden peaks in each short-time correlation are aligned after shifting, they sum coherently whereas the correlation noise sums incoherently. For example, the function $P(\hat{\boldsymbol{\tau}}, \tau)$ when all estimates are equal to the true delays ($\hat{\tau}_i = \tau_i \; \forall \; i$) is approximately

$$P(\hat{\boldsymbol{\tau}} = \boldsymbol{\tau}, \tau) \approx \pm N_c \cdot Y(\tau) + e(\tau) \quad (70)$$

where $e(\tau)$ accounts for the sum of correlation noise and negative correlation peaks which were summed incoherently. For any selection of time delays not equal to the correct time delays, the central sum component $N_c \cdot Y(\tau)$ will not be as large, since the individual peaks won't have summed coherently (see Figure 49). An objective function

$$\Phi(\hat{\boldsymbol{\tau}}) = \left| \int_{-T}^{T} P(\hat{\boldsymbol{\tau}}, \tau) \cdot Y(\tau)\, d\tau \right| \tag{71}$$

is defined as the zero-lag correlation between the shifted and summed correlation function and the expected peak shape. Large values of the objective function indicate the estimated time delays better match the true time delays of underlying correlation peaks hidden within noise.

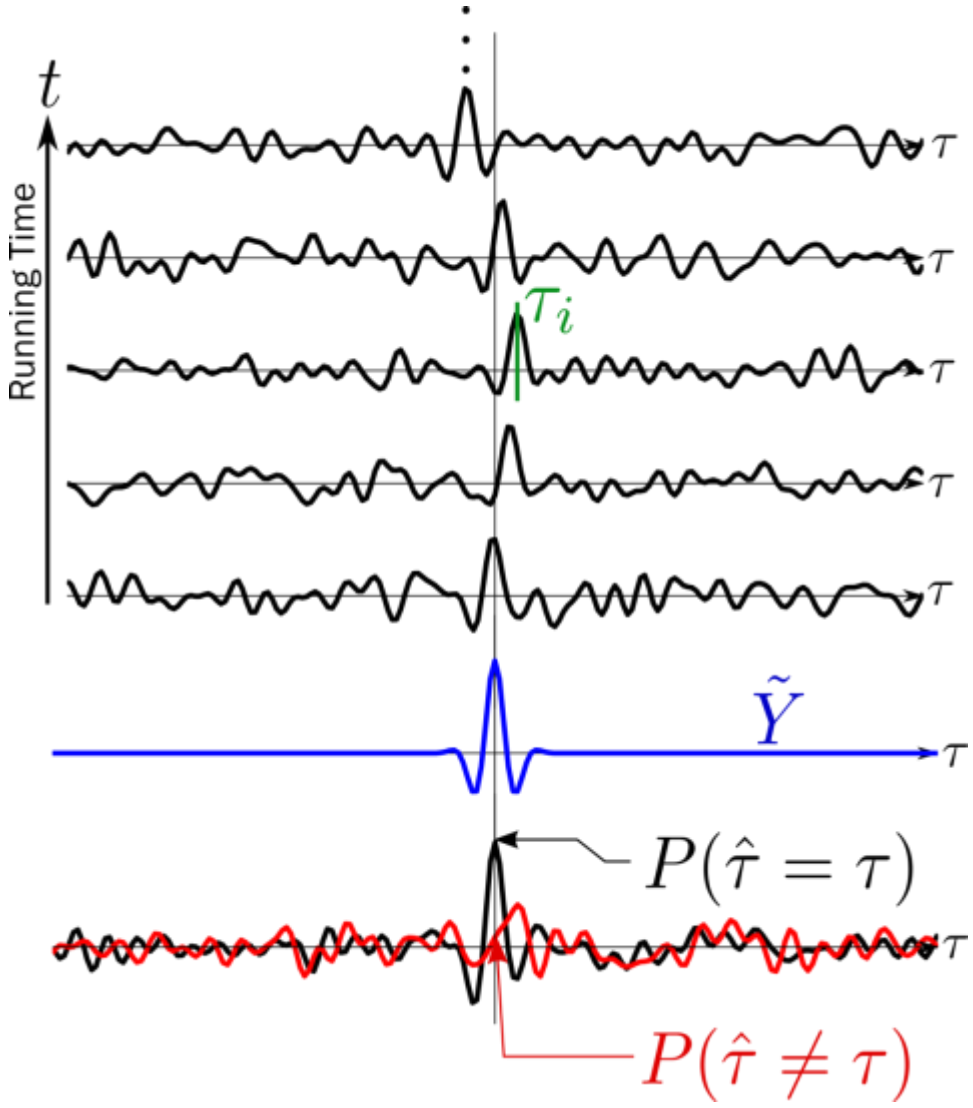


Figure 49: Correlation ensemble with hidden peaks (exaggerated for demonstration) having time delays τ_i. The shifted-and-summed correlation when the estimated time delays $\hat{\tau}$ match the true time delays τ is shown in black on the bottom. An example sum is shown when the time delays are not correct (red). These summed correlations are correlated with the reference peak shape $\tilde{Y}$ (blue).

The objective function in Equation 71 could be used directly to brute-force search for the correct time delays. A simple example demonstrating the computational complexity of such a problem highlights the need for a stochastic search method. Imagine ambient noise correlations were obtained for a duration of only two minutes, and that the sensor drift rate constrained the allowable correlation duration to be 10 seconds. With no overlap

in correlation windows, this requires estimation of only 12 time delays. Assuming the prior estimate of sensor separation was rather precise (e.g. GPS-measured), the search area might be $\pm$1.5 m, or $\pm$1 ms. If the resolution of the distance estimation is desired to be 10 cm (i.e. a time-delay ~67 μs assuming a water sound speed of ~1500 m/s), this equates to 30 possible delays to evaluate per correlation window. Since each of the 30 possible delays must be tested with all other permutations of the other time delays, the total number of guesses to make is $30^{12} = 5.3 \cdot 10^{17}$. Running such a simple and restricted search at 100 billion evaluations per second would still take over two months to return the optimum solution. A stochastic search can find the optimum or near-optimum solution in real time or faster even on modest hardware.

4.2.2 Simulation Methodology

To evaluate the performance of various stochastic search methods and the numerous parameters they possess, a simulation environment was implemented. A simulated environment allows the peak locations to be known exactly, as well as the PSNR of the peak to its surrounding noise. The PSNR is defined in a similar way to Equation 50, and is given by

$$PSNR = \frac{Y(0)}{3 * \text{std}[N(\tau)]} \tag{72}$$

where $Y(\tau)$ is again the known peak shape of a noise correlation process (see Equation 43), and $N(\tau)$ is simulated additive band-limited noise (see Figure 50). An ensemble of correlations is thus simulated as

$$C_{12}(\tau, t_i) = Y(\tau - \tau_i) + N_i(\tau) \tag{73}$$

where the i-th correlation center time is t_i, the peak delay is τ_i, and the additive noise realization $N_i(\tau)$. The noise is simulated by taking random white noise and filtering it to within the frequency band of interest. This is performed in the frequency domain by windowing the noise in the time domain with a Tukey window (r = 0.03) equal in length to the sequence, then windowing in the frequency domain with a Tukey window (r = 0.1) encompassing the operating band. Lastly, the noise is divided by its standard deviation and multiplied by the desired standard deviation σ_N. This ensures the noise sequence has standard deviation σ_N, so the PSNR is known. The magnitude of $Y(\tau)$ is kept constant, and the noise variance adjusted accordingly. Also, it is worth noting that Equation 73 is missing the negative time delay, but since the separation distance is typically much larger than the search window employed by the stochastic search, it is irrelevant whether the negative peak exists.

To generate realistic values for the peak time delays τ_i, real separation data was obtained from float GPS data collected off the coast of New London, CT in October 2016. A time period was chosen which exhibited somewhat fast drift speeds, with a maximum velocity of 200 m/hr inferred from GPS measurements. Since GPS data was sampled at 10 second intervals, a linear interpolation was used to obtain the separation distance between samples, at a rate of 1 second updates for the simulated correlation ensemble.

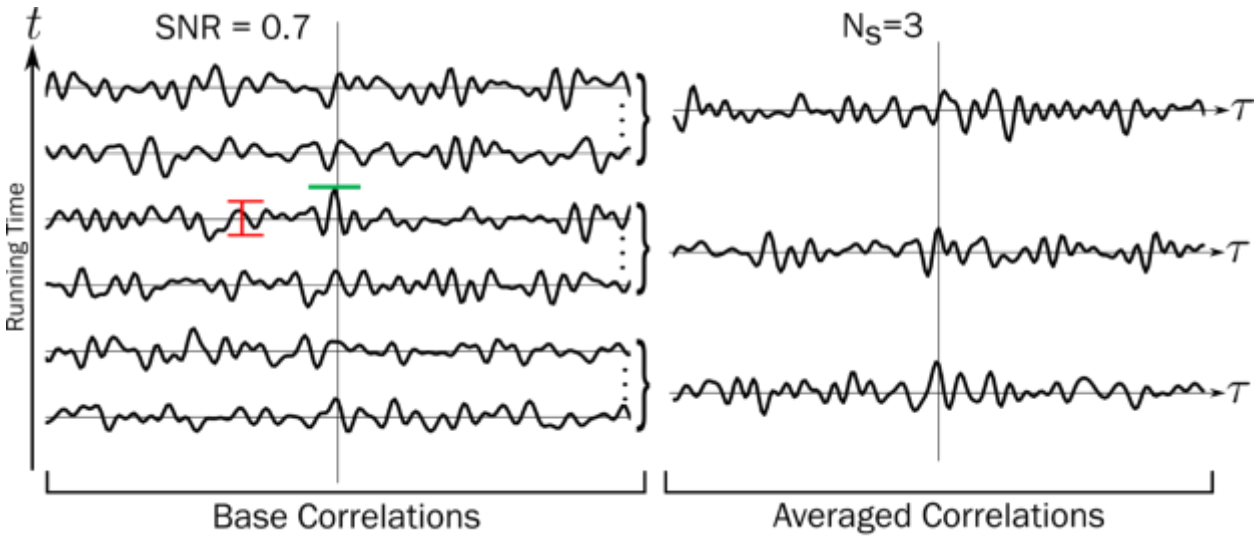


Figure 50: Simulation of ambient noise correlations, where base correlations (left) are simulated with known PSNR (0.7). A number of the base correlations are averaged to form a total of N_s averaged correlations (right).

For each simulation, an ensemble correlation was created with a given PSNR (see Figure 50), and an assumed base correlation length (e.g. 1 second). The resulting base correlation contained a total of 900 correlations, covering a fifteen-minute period. To then simulate the effect of increasing the correlation averaging times, multiple base correlations were averaged (see Figure 51). Performing averaging of the base correlations in this way also accounts for the possibility that successive peak functions will incoherently sum if the averaging length is too long.

Once all base correlations have been averaged, the resulting N_s correlations are given to a stochastic search method to attempt to identify the peak locations. However, the number of averaged correlations (N_s) will often be much larger than a stochastic search can feasibly identify simultaneously (e.g. 900 correlations total). Thus, the averaged correlations are segmented into pieces of length N, and each is identified separately by the stochastic search (see Figure 51).

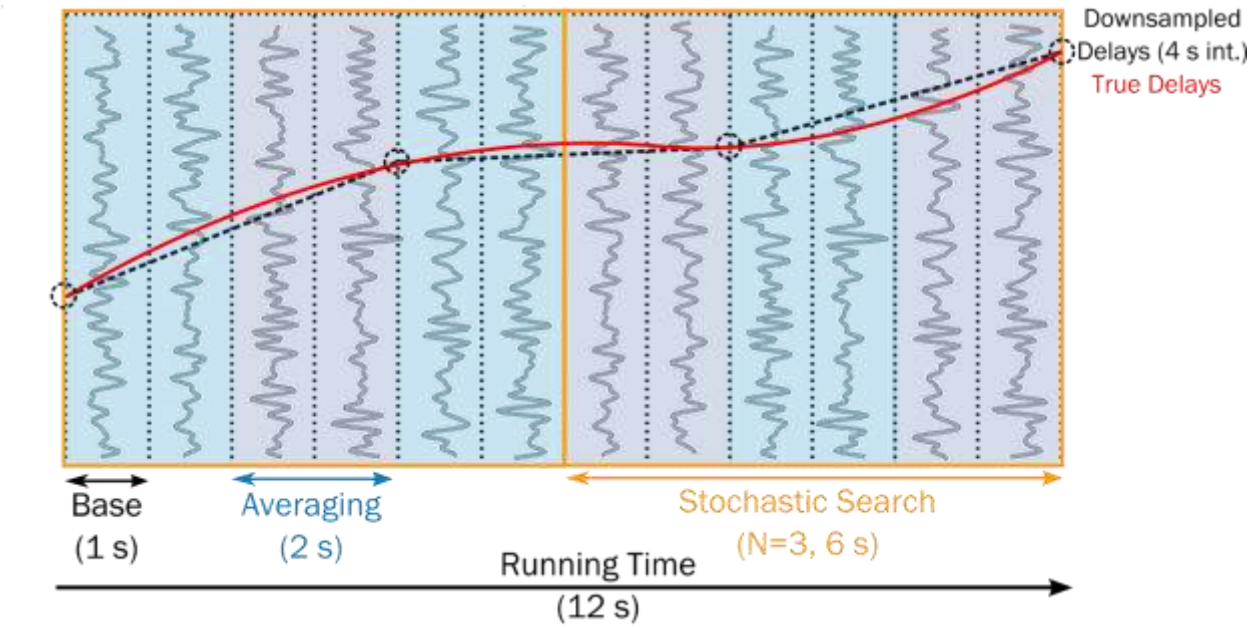


Figure 51: Stochastic search methodology. Base correlations (12 total) are shown in black, and the true time delays of their hidden peaks in red. The downsampled true delays are plotted with dashed circles, with a linear interpolation between them. Correlations are averaged in groups (blue regions), and the stochastic search operates on segments of averaged correlations (yellow boxes).

The example demonstrated in Figure 4 shows a total number of correlations $N_c = 12$. The base correlation averaging duration is assumed to be 1 second, for a total of 12 seconds of correlations. The averaging length is $T = 2$ seconds, which means every 2 correlations are averaged (blue regions) to obtain a total of 6, 2-second correlations. Then, the stochastic search is segmented using $N = 3$ to identify a total of 6 time delays in 2 separate searches. The initial conditions of the search can either be obtained from the true delays (red curve) or the downsampled delays (black dashed curve). The example in Figure 51 uses downsampled delays obtained from linear interpolation of the true delays downsampled to a 4 second period.

Table 3 shows the salient parameters which are varied in the following simulations. The correlation averaging length T is varied by averaging different numbers of the

simulated base correlations. If too small, the correlation peaks will remain buried in noise and require a larger number of shifted and summed correlations in the stochastic search to obtain a coherent peak. If made too large, the drifting correlation peaks will begin to sum incoherently (i.e. "smear"). This limit is defined by the drift rate and is described in Section 4.1. The parameter N is the number of parameters to estimate in each stochastic search. Decreasing this number too low could result in the shifted and summed correlations to still have an SNR of less than 0 dB, and thus the objective function (Equation 71) will not recognize a good solution. In addition, decreasing N will speed up each individual search, but will require more searches overall. Increasing N too much will result in the stochastic search having too many parameters to estimate, and increase the chances of a suboptimal global maximum is obtained (see Appendix E). Lastly, the base SNR is varied to simulate any number of effects which serve to decrease the correlation peak size in relation to the noise. These effects were discussed in Section 3.3 and include increased sensor separation distance or increased sensor noise.

Table 3: Salient Parameter Table

Parameter Description	**Name**	**Typical Values**	**Considerations**	
			Make Smaller	**Make Larger**
Correlation Averaging Length	T	1 – 30 seconds	Coherent peaks buried in noise → increase N	Increased SNR, but possible smearing of coherent peaks (drift)
Stochastic Search Segment Length	N	5 – 30 time delays	Too few correlations to sum → no peak in objective function	Too many time delays for stochastic search to estimate
Base SNR	$PSNR$	-20 – 0 dB	Need more averaging to develop peaks → increase N, T	Less noise, above 0 dB → peak detection

Simulations varying each of the parameters in Table 3 were carried out for different stochastic search methods. Presented here are the results for the genetic algorithm (see Appendix E for others). The genetic algorithm has its own set of many different parameters, and most were left at the default MATLAB settings. However, changes were made to limit the maximum number of generations to 1000, set the population size to 100, set the number of population elites to 2, and set the initial population to the prior guess (e.g. GPS estimated separation distances).

The first method is hereafter referred to as the "unconstrained" genetic algorithm. The only constraint is a rectangular window within which the solution must lie. The bounds of this window are obtained by expanding the limits of the true separation distances. The expansion coefficient K_B was taken to be 4, and the bounds calculated using the equation

$$\vec{\boldsymbol{\tau}}_{\boldsymbol{B}} = [\tau_{lb}, \tau_{ub}] = \frac{1}{2c}(R_{max} + R_{min}) + \frac{K_B}{2c}[-(R_{max} - R_{min}), (R_{max} - R_{min})] \quad 74$$

where the distances R_{max} and R_{min} are the maximum and minimum measured separation distances. Lastly, the starting population estimates were determined from a downsampled measurement of the separation distance. This mimics the scenario where GPS or other localization methods are only obtained infrequently. In between, the separation distance is linearly interpolated. For these simulations, the period of downsampled separation distance measurements was taken to be 5 minutes. Thus, the starting estimate of the genetic algorithm is linearly-interpolated from actual separation distance measurements taken only every 5 minutes.

The second method compared was the "constrained" genetic algorithm. The rectangular bounds used in the unconstrained method were identical, however the addition of a difference constraint between successive estimates was taken to be less than δ, which was expanded from actual separation distances using the equation

$$\delta = \frac{K_D}{c} \cdot \max|R_i - R_{i-1}| \quad 75$$

where R_i denotes the separation distance measured at t_i, and the maximum is taken over the entire duration of the correlation ensemble. The difference expansion coefficient K_D was taken to be 5, thus allowing the largest difference in successive time delays to be five times that seen in the true separation distance. Note that the difference constraint is only applied to the identification subsection containing N time delays. No constraints were carried across the boundaries of successive segments. Such a constraint is possible if the previous segment's solution is already known. The constrained method used identical

genetic algorithm parameters to the unconstrained method, and used the same starting estimate obtained from linearly-interpolated 5-minute true separation distance updates.

To implement the difference constraint, a set of linear constraints was specified to constrain the difference in successive peak locations. The MATLAB algorithm requires the linear constraint to be of the form

$$A\vec{\hat{\boldsymbol{\tau}}} \leq \vec{\boldsymbol{b}} \tag{76}$$

where $\vec{\hat{\boldsymbol{\tau}}}$ is the solution estimate vector. To constrain the differences between successive estimates, the matrix was given by

$$A = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & \ddots \end{bmatrix} \tag{77}$$

and the vector

$$\vec{\boldsymbol{b}} = [\delta_1 \quad \delta_1 \quad \delta_2 \quad \delta_2 \quad \ldots]^T \tag{78}$$

where the values δ_i were generally all equal to a constant δ. The value of δ could be either fixed (e.g. 1 ms or 1.5 m maximum change in successive estimates), or determined by the maximum difference observed in the prior (e.g. GPS) data (see Equation 75). The latter generally resulted in smaller constraints, so it was often multiplied by a factor (typ. 5) to allow for larger changes to occur during the solution search.

The resulting solutions returned by the genetic algorithm were evaluated with respect to the true peak locations using the root-mean-square error (RMSE)

$$RMSE = c \cdot \sqrt{\frac{1}{N_s}\sum_{i=1}^{N_s}(\hat{\tau}_i - \tau_i)^2} \quad (79)$$

which takes the square root of the average squared error of the estimated time delays $\hat{\tau}_i$, and true time delays τ_i, then converts the value to units of meters by multiplication with the baseline sound speed $c = 1500\, m/s$. The RMSE expressed in meters is useful for comparing the performance of the distance estimation to other measurements of distance (e.g. GPS).

4.2.3 Simulation Results

The simulation results are computed for the values $PSNR$ = [1, 0.85, 0.7, 0.5, 0.3, 0.2, 0.1, 0.05, 0.01] or equivalently, in decibels $PSNR$ = [0, -0.7, -1.5, -3, -5.2, -7, -10, -13, -20] dB. The averaging duration values simulated were $T = [1, 2, 5, 10, 20, 30]$ seconds, and the stochastic search segment lengths N = [5, 10, 20, 30]. Following the parameter order in Table 3, the results for each parameter variation are shown below.

4.2.3.1 Changing Averaging Length

Referring to Table 3 to guide the results found here, the first effect which will be examined is changing the correlation averaging length T. The first two plots in Figure 52 and Figure 53 show the variation in RMSE for constant N = 10.

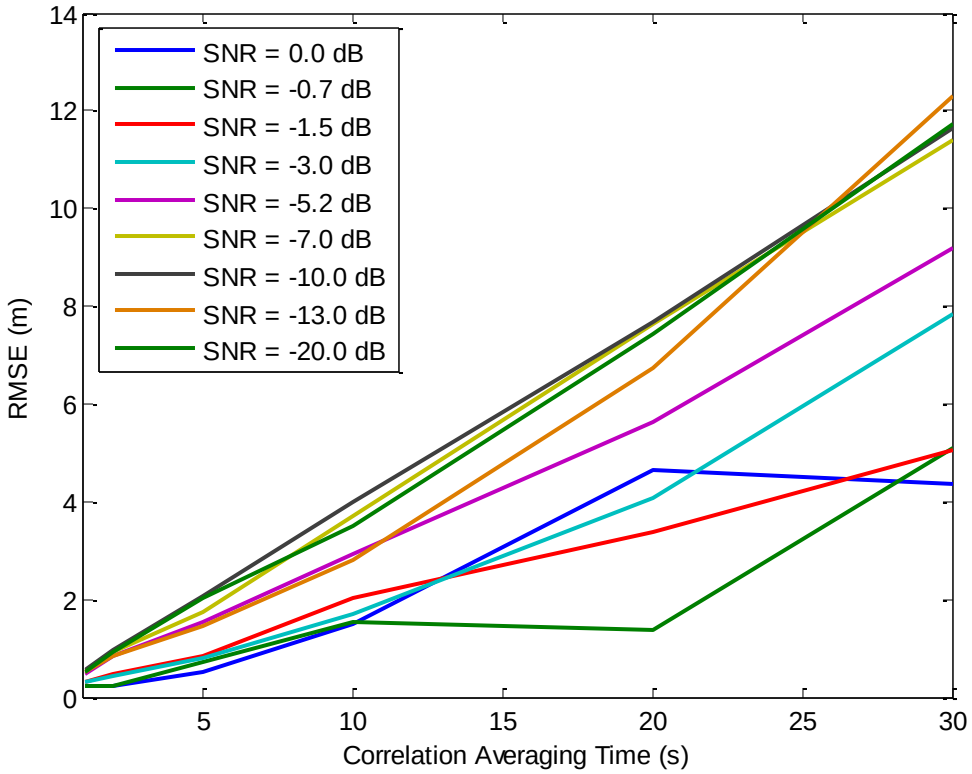


Figure 52: RMSE of unconstrained genetic algorithm vs. averaging time and SNR, N = 10

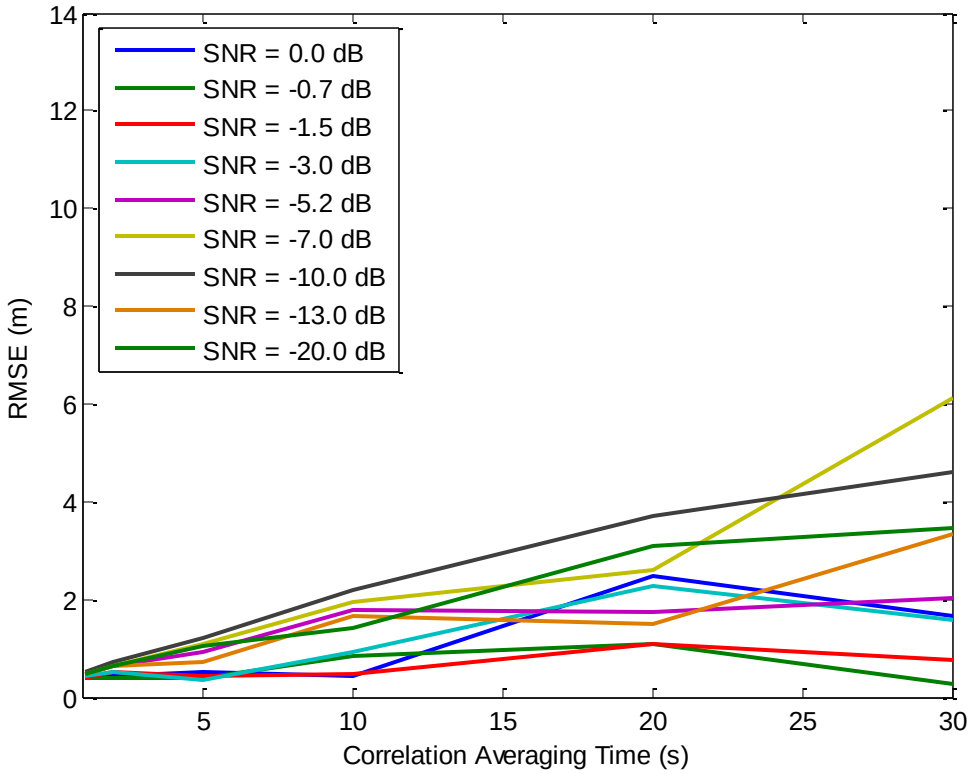


Figure 53: RMSE of constrained genetic algorithm vs. averaging time and SNR, N = 10

From these figures, there are two key takeaways:

- Increasing the averaging time increases the error, more so for the unconstrained method
- Decreasing the SNR increases the error, but the results are somewhat random

The variation in RMSE vs. averaging length for constant SNR = -3 dB is plotted in Figure 54 and Figure 55.

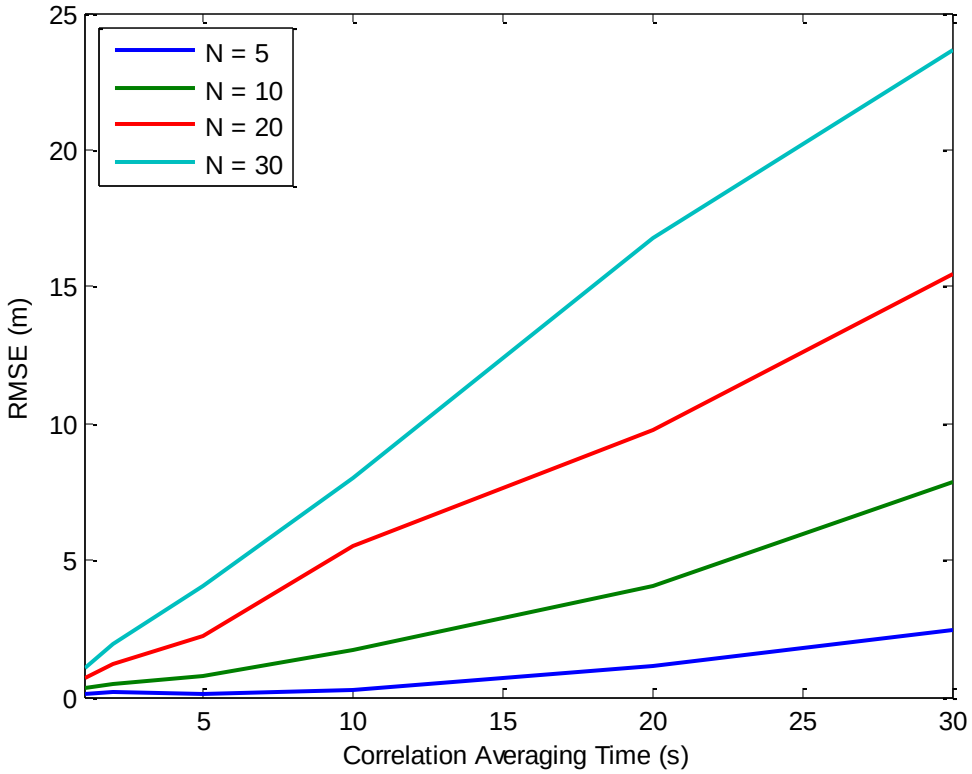


Figure 54: RMSE of unconstrained genetic algorithm vs. averaging time and N, SNR = -3 dB

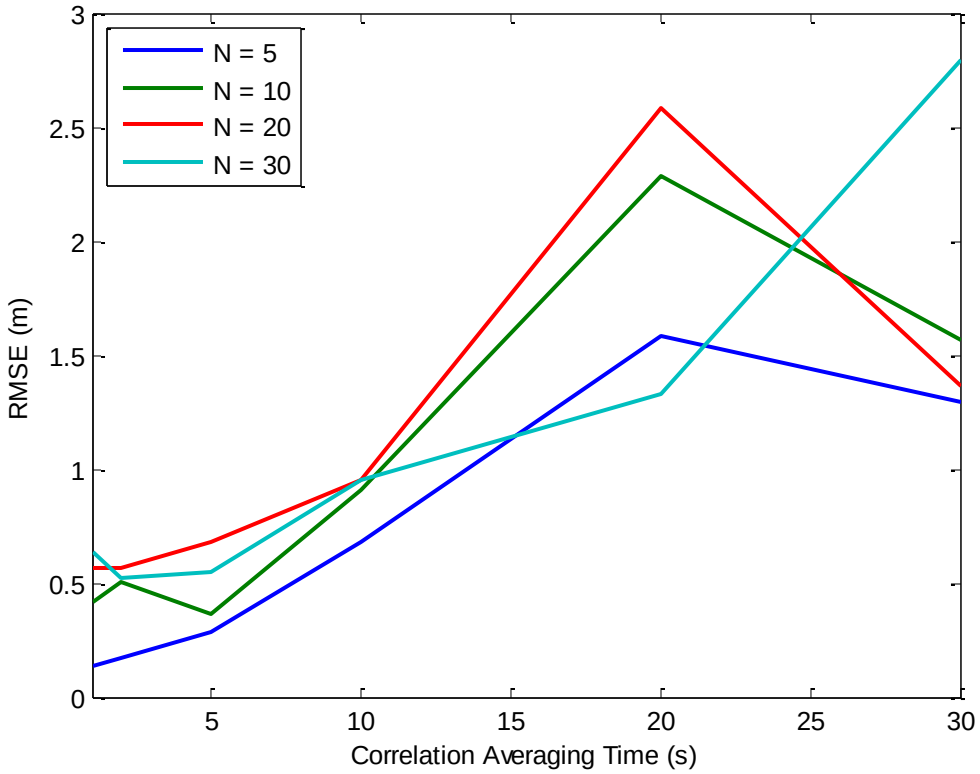


Figure 55: RMSE of constrained genetic algorithm vs. averaging time and N, SNR = -3 dB

From these results, the first takeaway remains the same:

- Increasing the averaging time increases the error
- When unconstrained, the increase in error is worse for longer stochastic search lengths

Finally, the variation in averaging time is shown across all values of SNR and N in stacked surface plots in Figure 56 and Figure 57.

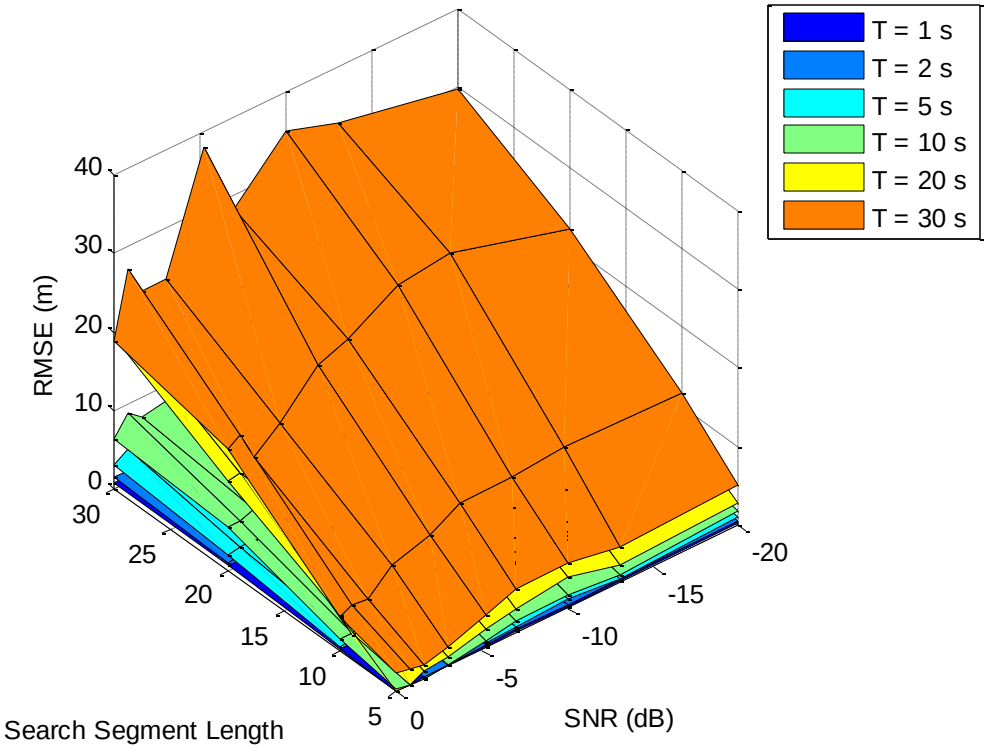


Figure 56: RMSE of unconstrained genetic algorithm vs. averaging time, SNR, and T

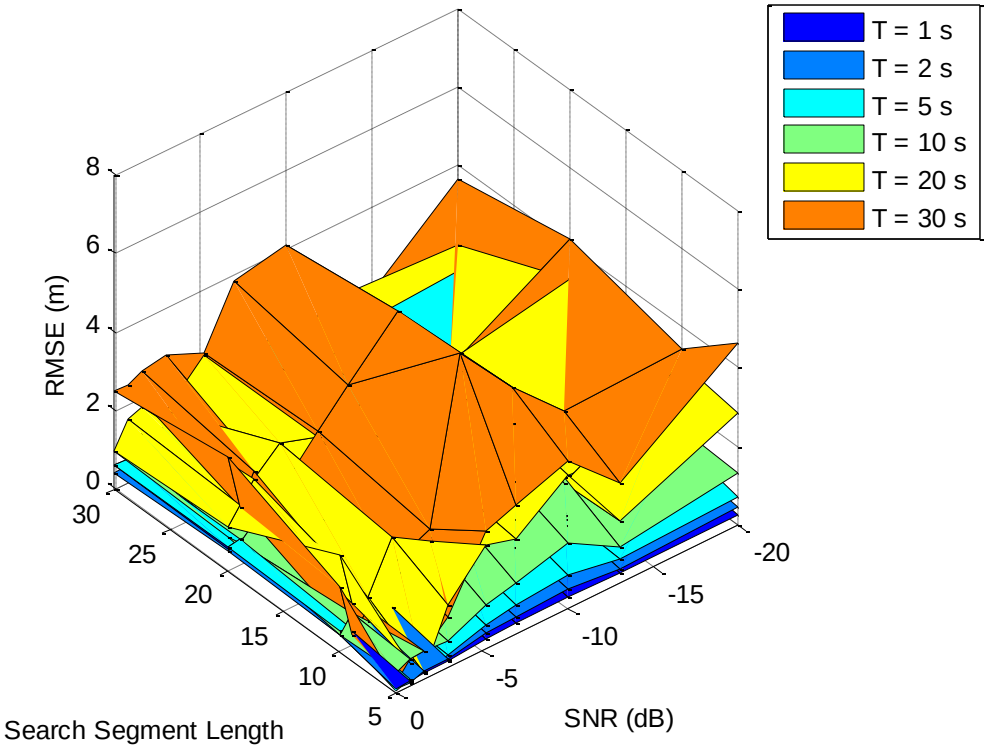


Figure 57: RMSE of constrained genetic algorithm vs. averaging time, SNR, and T

The results here corroborate the findings from the plots of N = 10 and SNR = -3 dB in that the **increased averaging time leads to increased RMSE**. The increase is less severe for the constrained method.

4.2.3.2 Changing Segment Length

The second parameter in Table 3 is the stochastic search segment length N. Figure 66 and Figure 67 show the variation in RMSE with varying segment length for T = 2 seconds.

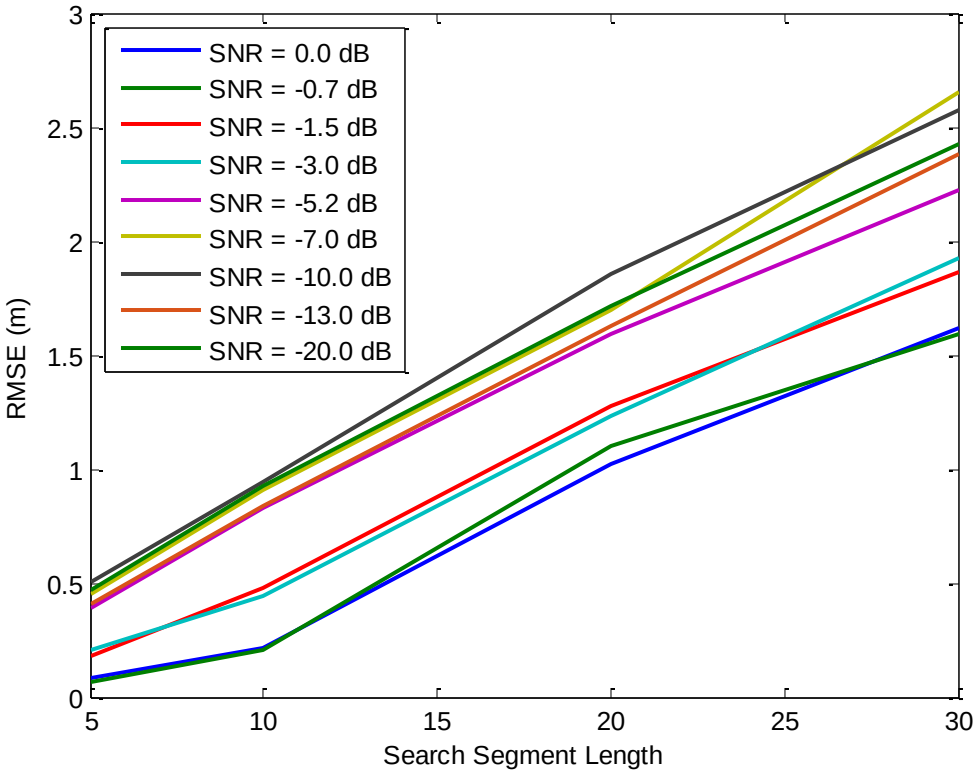


Figure 58: RMSE of unconstrained genetic algorithm vs. N and SNR, T = 2 s

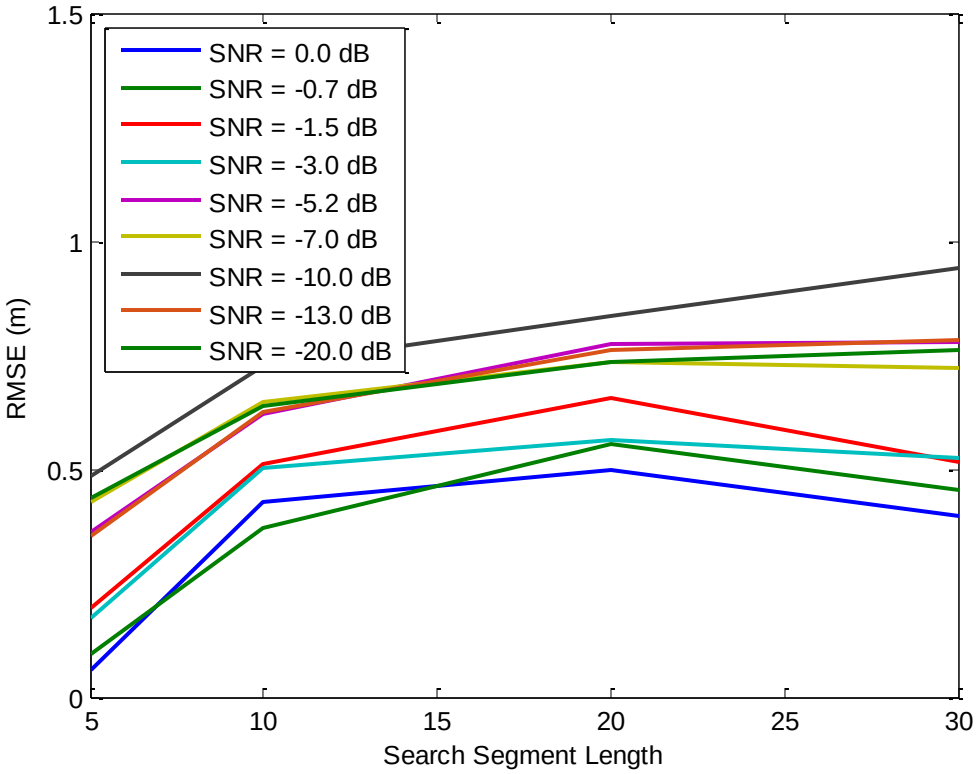


Figure 59: RMSE of constrained genetic algorithm vs. N and SNR, T = 2 s

From these results, the takeaways are

- Increasing the segment length increases the error
- Decreasing SNR increases the error somewhat, more so for the unconstrained method

The variation in RMSE with changing segment length is plotted in Figure 60 and Figure 61 for constant SNR = -3 dB.

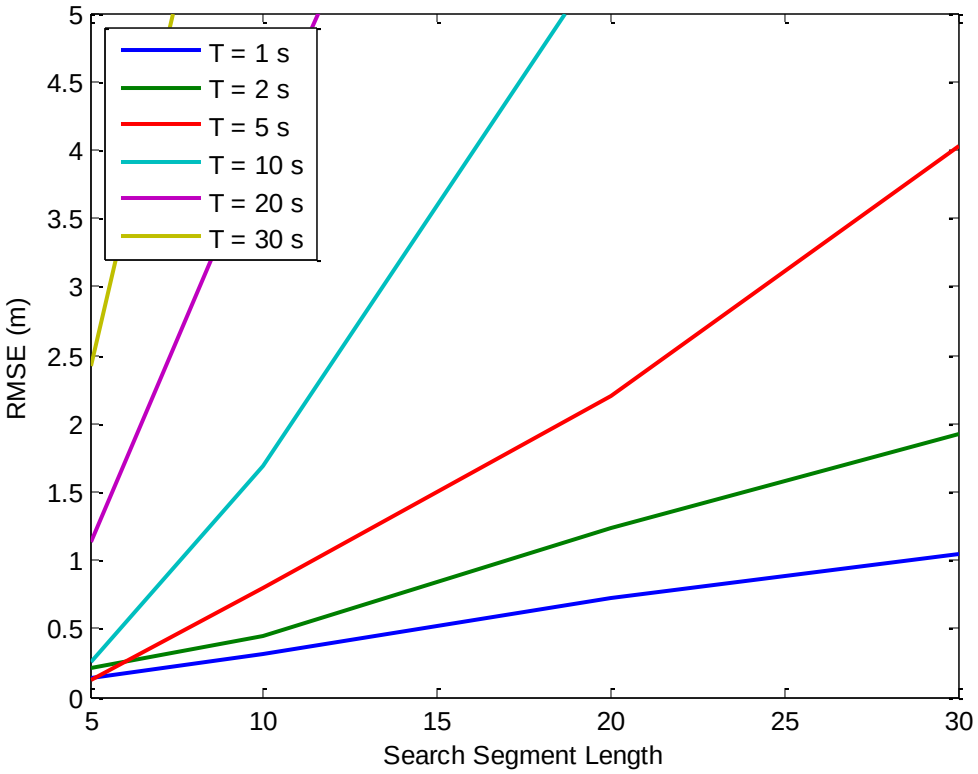


Figure 60: RMSE of unconstrained genetic algorithm vs. N and T, SNR = -3 dB

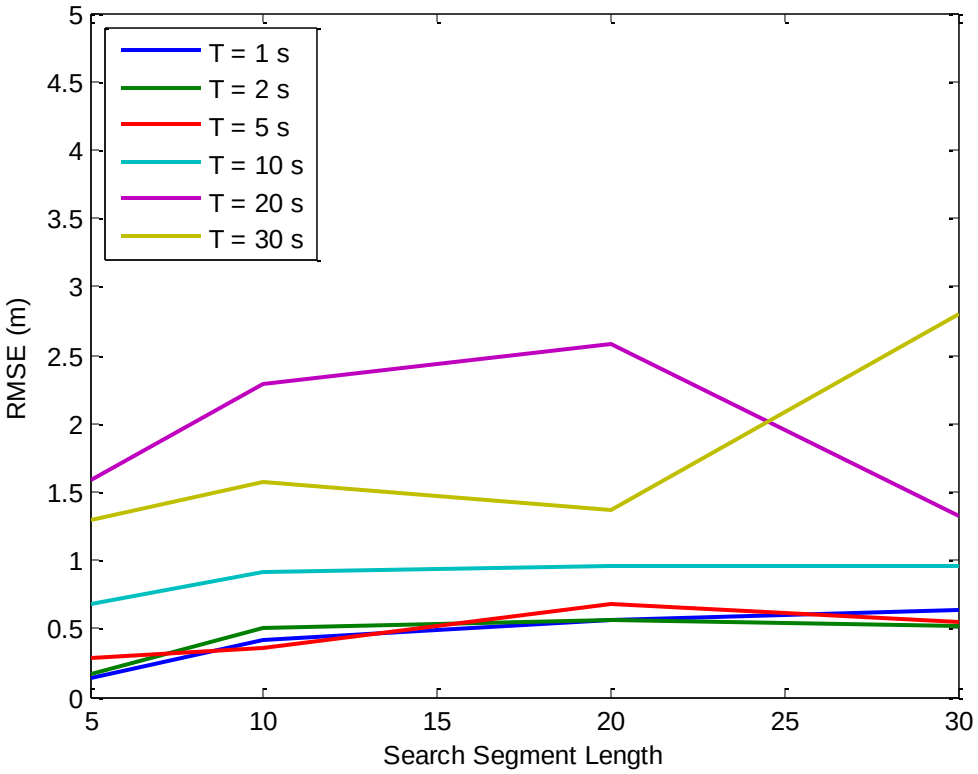


Figure 61: RMSE of constrained genetic algorithm vs. N and T, SNR = -3 dB

The takeaways from these results are

- Increasing N increases error, but much less so for the constrained method
- Increasing averaging time increases error, especially for unconstrained method

Finally, the variations in segment averaging length are plotted as stacked surfaces in Figure 62 and Figure 63.

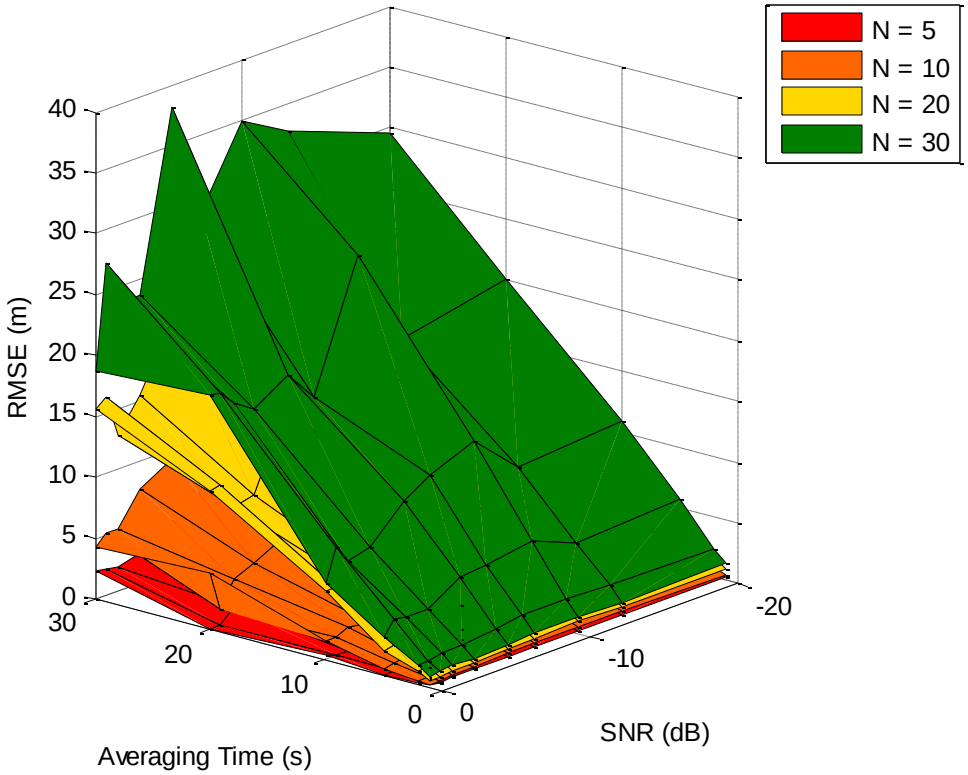


Figure 62: RMSE of unconstrained genetic algorithm vs. T, SNR, and N

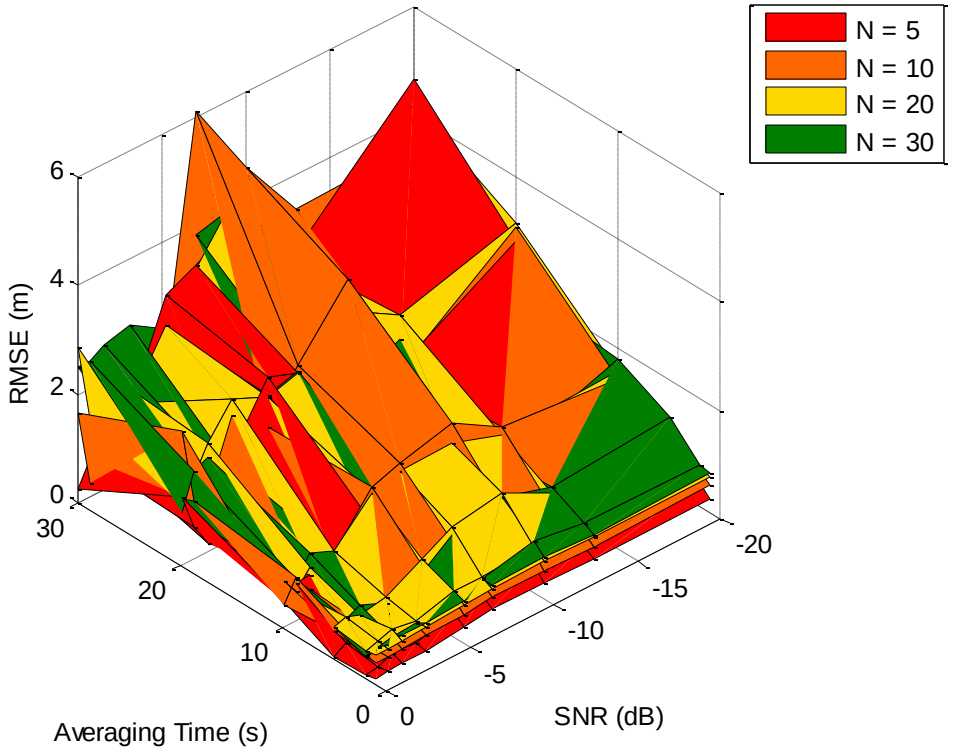


Figure 63: RMSE of constrained genetic algorithm vs. T, SNR, and N

The takeaways from these figures are that

- Increasing N for the unconstrained method greatly increases error
- Increasing N for constrained method only slightly increases error
- Increasing T for unconstrained method greatly increases error
- Increasing T for constrained, only slightly increases error

In summary, the effect of increasing N appears to be very similar to increasing the averaging length. Furthermore, the increase in error when unconstrained may indicate that the error is resulting from the increased estimation duration. In other words, the further the sensors drift during an observation window, the more error will result when attempting to

identify time delays. Constraining the difference in successive values can somewhat abate these effects.

4.2.3.3 Changing SNR

Finally, the last parameter of Table 3 to vary is the base correlation SNR (equivalently called PSNR). This parameter generally is not controlled in an experiment, as the ambient noise sources and interfering noise sources will dictate the correlation process SNR. The variation in RMSE is plotted versus the SNR for an averaging length T = 2 seconds in Figure 64 and Figure 65.

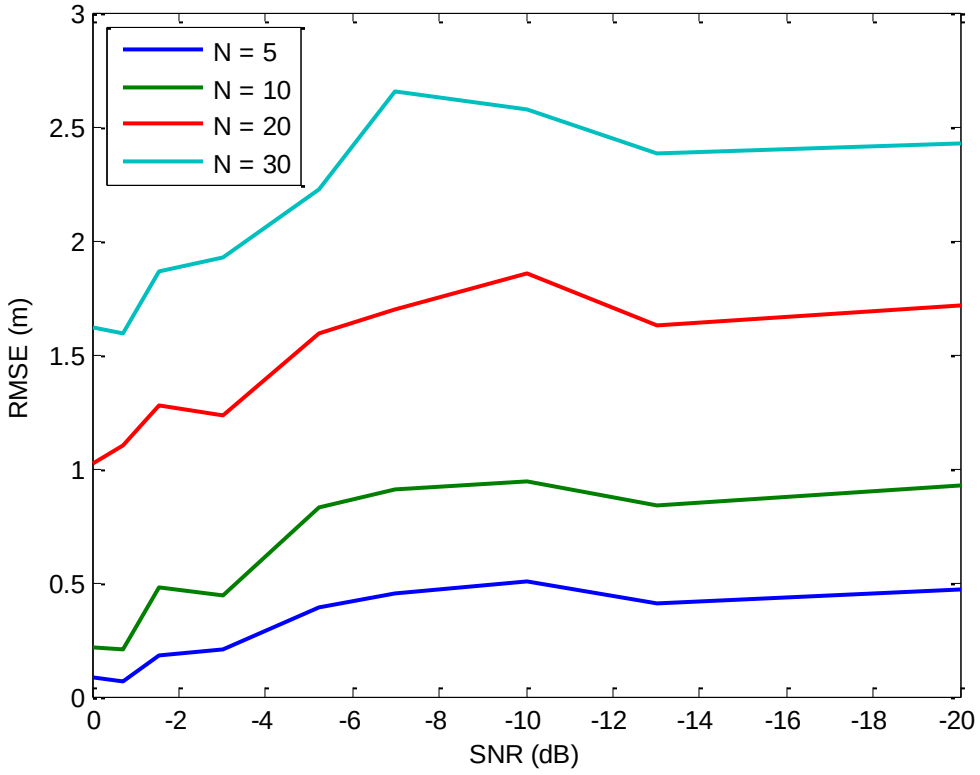


Figure 64: RMSE of unconstrained genetic algorithm vs. SNR and N, T = 2 s

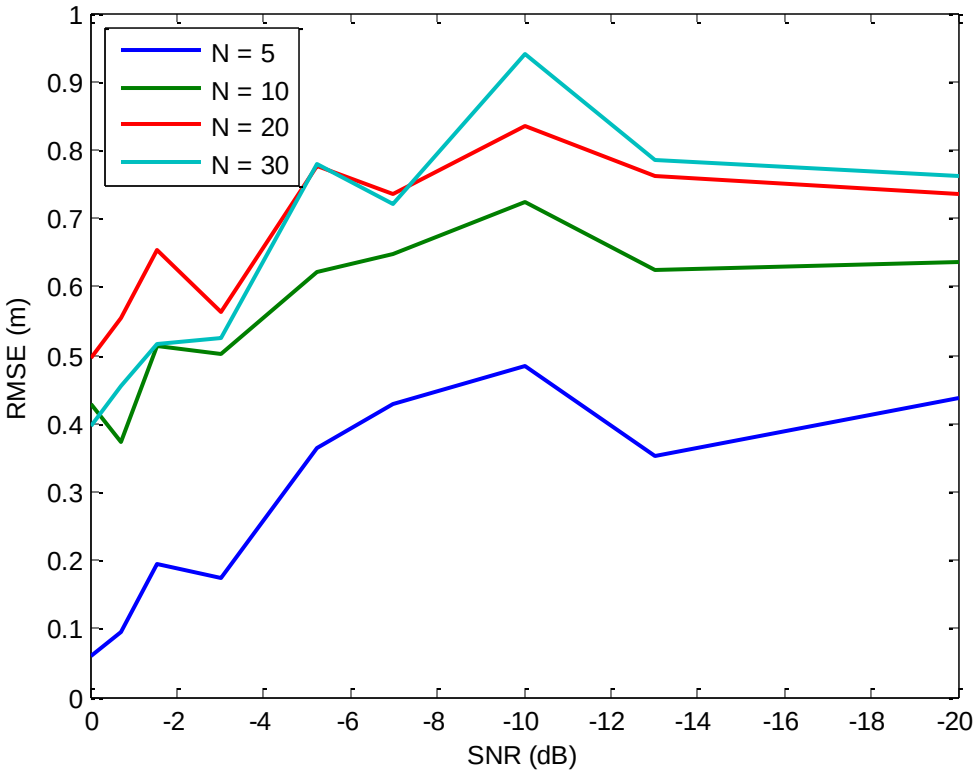


Figure 65: RMSE of constrained genetic algorithm vs. SNR and N, T = 2 s

The takeaways from these results are

- Increasing the noise (i.e. decreasing SNR) increases the error
- Increasing the segment length increases the error, most strongly for unconstrained method

The error is plotted for varying SNR and averaging time for a segment length of 10 in Figure 66 and Figure 67.

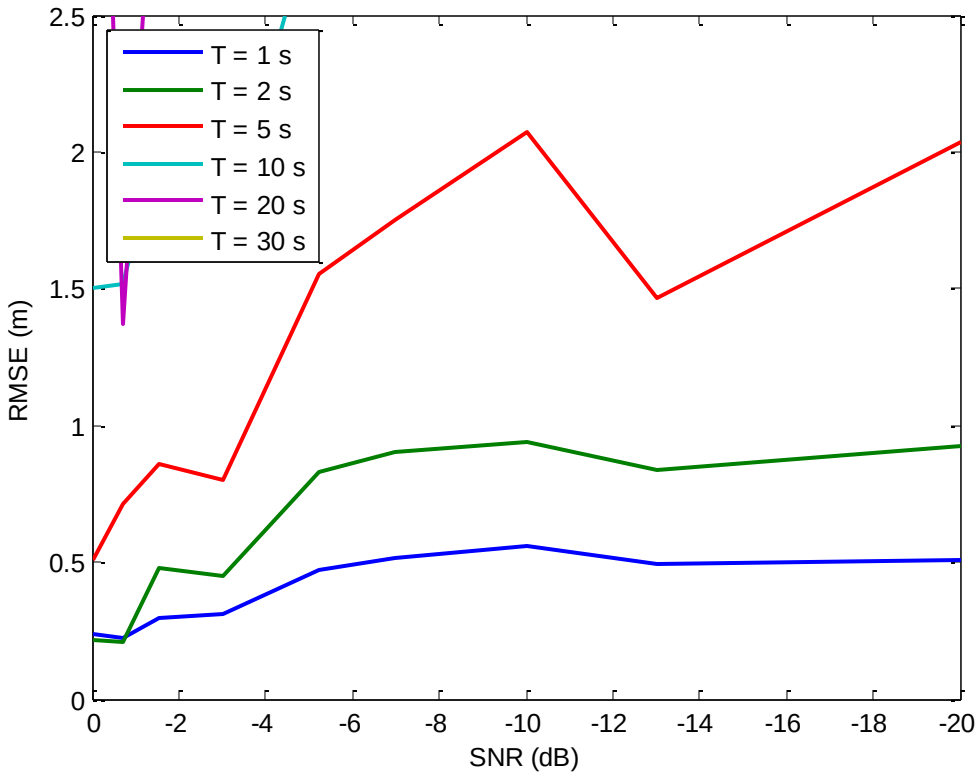


Figure 66: RMSE of unconstrained genetic algorithm vs. SNR and T, N = 10. High values of T are not displayed due to their large error.

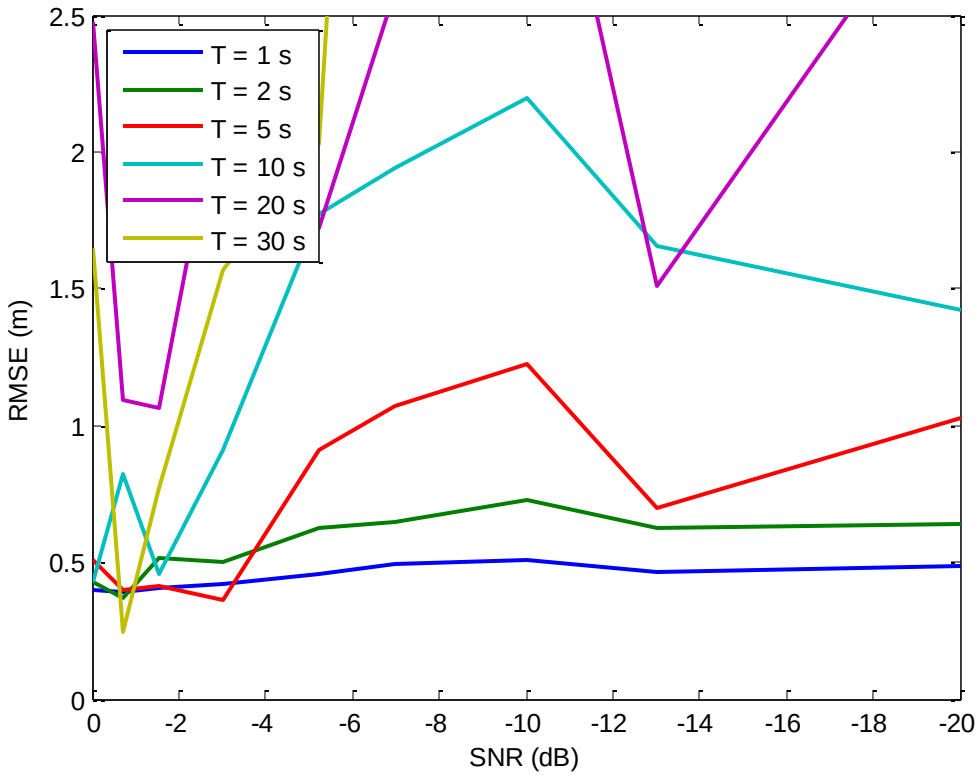


Figure 67: RMSE of constrained genetic algorithm vs. SNR and T, N = 10. High values of T are not displayed due to their large error.

The takeaways from these results are

- Increased noise (decreased SNR) increases the error
- Constrained GA with short averaging duration mitigates the effect

Finally, the effect of changing SNR is demonstrated in stacked surface plots in Figure 68 and Figure 69.

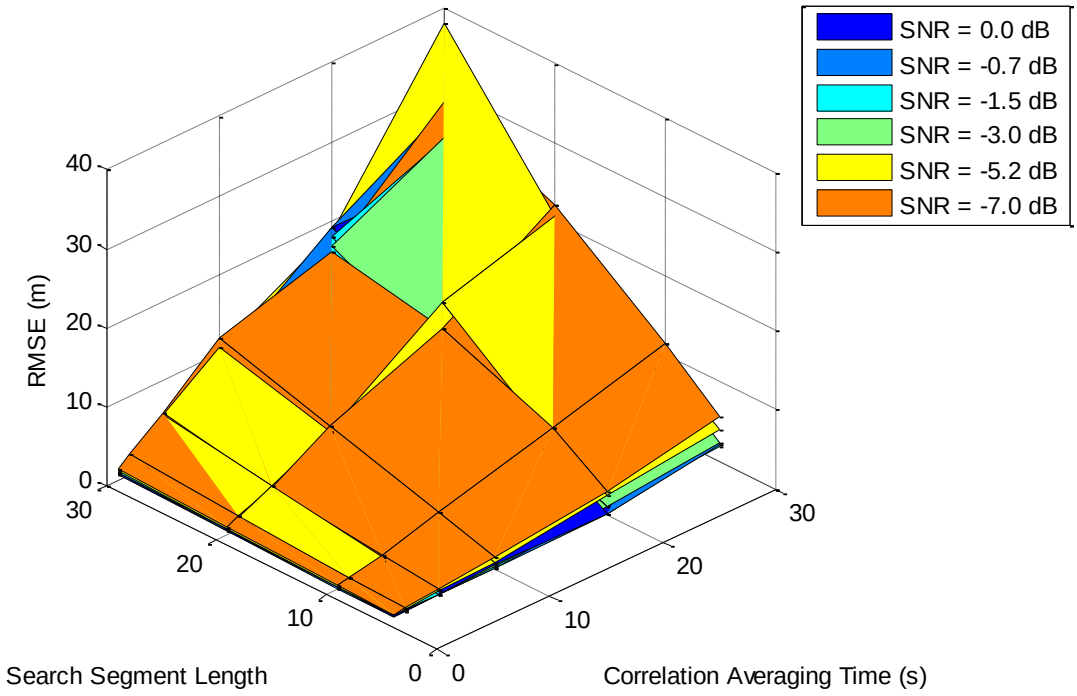


Figure 68: RMSE of unconstrained genetic algorithm vs. T, N, and SNR.

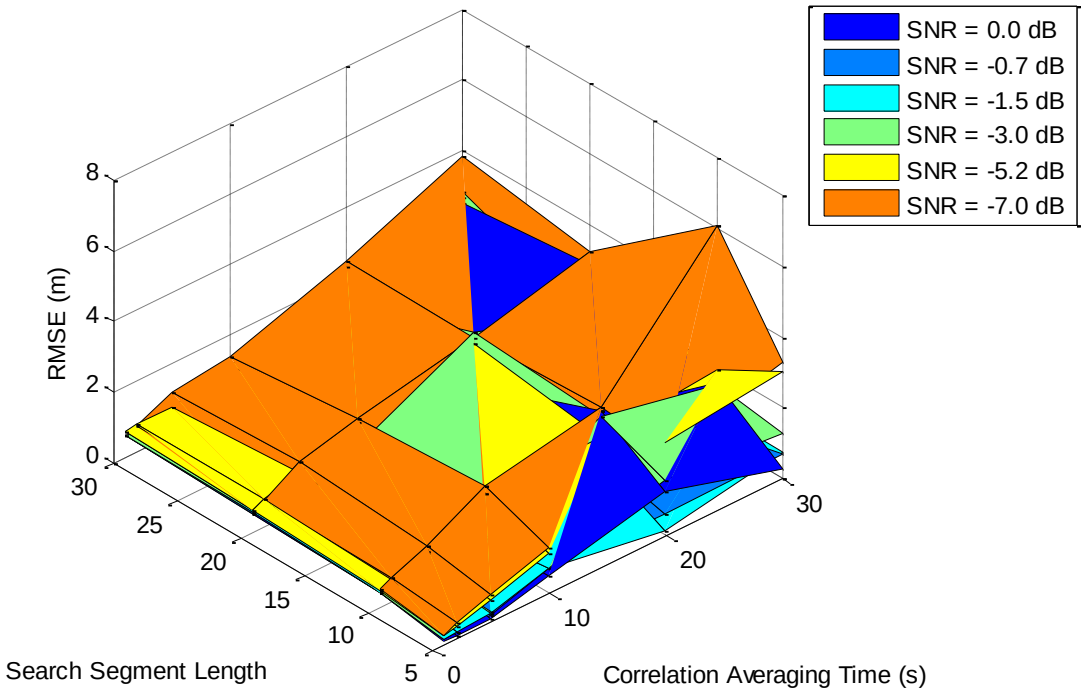


Figure 69: RMSE of constrained genetic algorithm vs. T, N, and SNR.

The results are more difficult to interpret when plotted as stacked surfaces, since the SNR does not have as severe an effect on the RMSE and the surfaces all intersect. It is likely that improved results would be obtained by running multiple statistical trials at each SNR to obtain the average performance, however this would be very time-consuming to evaluate for every permutation of parameter. In any case, the takeaways for this section are

- Increased SNR decreases error, or increased noise increases error
- Constrained method is more robust to low SNR, however this may be caused simply by the constraint and not the actual identification of peaks

4.2.4 Experimental Methodology

As explained in Section 2.3.5, the usable experimental data which contains synchronized, clean pressure and velocity channel data from floats with valid GPS positions comes from the October 30, 2016 dataset. The two floats with valid GPS positions were 0004 and 0005, with fixed depths of 20 and 26 meters. During the deployment, there were two times during which the R/V Greeley was at a distance and running silently, the first between approximately 11:49 and 13:28 local time, and the second between 14:45 and 15:22. The first quiet time was corrupted with loud interfering vessels in the area and was not suitable for ambient noise correlation identification (see Appendix F).

During the second quiet time, lasting approximately 40 minutes, the sensors drifted from 55 meters apart to 130 meters apart, diverging the entire time (see Figure 70). The maximum drift speed seen during this period was 0.055 m/s, or 200 m/hr.

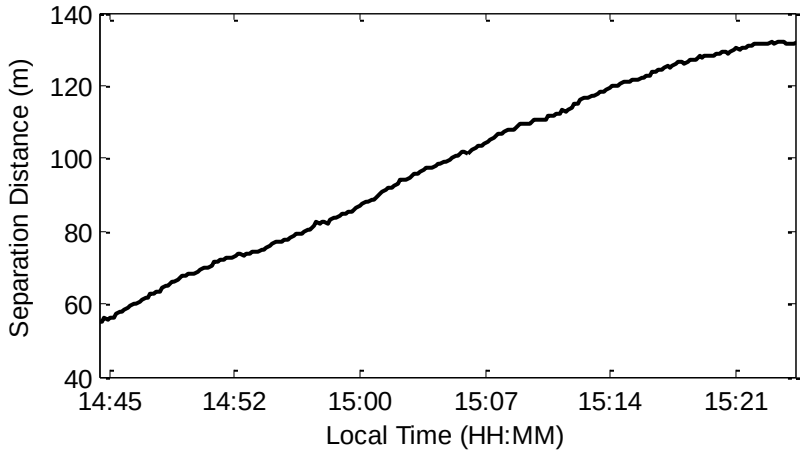


Figure 70: Separation distance between floats 0004 and 0005 during quiet time 2.

An ensemble of correlations of 1 second length are obtained and plotted in Figure 71 and Figure 72. They are normalized with respect to the total maximum observed throughout the quiet time. Per-correlation normalized plots were shown in Figure 43 and

Figure 45 (see Section 3.3.2). The preprocessing steps used in preparing the data for correlation was to first filter the data, then integrate the acceleration and convert to pressure units, then filter, clip, and whiten the resulting data (see Section 2.3.4 for more information on these processes). The bandwidth chosen for the filtering was obtained by time-windowing the full-bandwidth correlations around the supposed ambient noise correlation peak and plotting the peak's spectrum. The resulting spectrum is nearly flat across 200 – 1500 Hz, with a small decrease in the amplitude of the 800-900 Hz band. Thus, the bandwidth was chosen to be maximally wide, and encompassed 200 – 1500 Hz. The bandwidth chosen also avoids the low-frequency velocity noise (< 100 Hz) and the velocity resonant frequency (1.8 – 3 kHz). As seen in Section 3.3.1, the variance of the ambient noise correlation decreases with the time-bandwidth product, thus the widest allowable band was chosen.

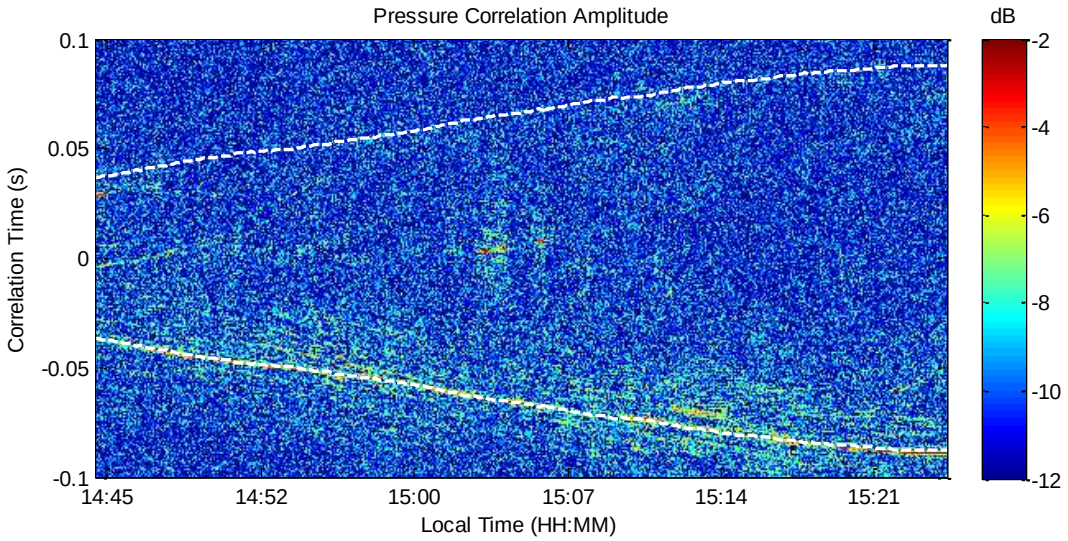


Figure 71: Pressure correlation ensemble amplitudes for quiet time 2. The amplitude is relative to the overall maximum amplitude, and is 10 $\log_{10}$ of the Hilbert transform magnitude.

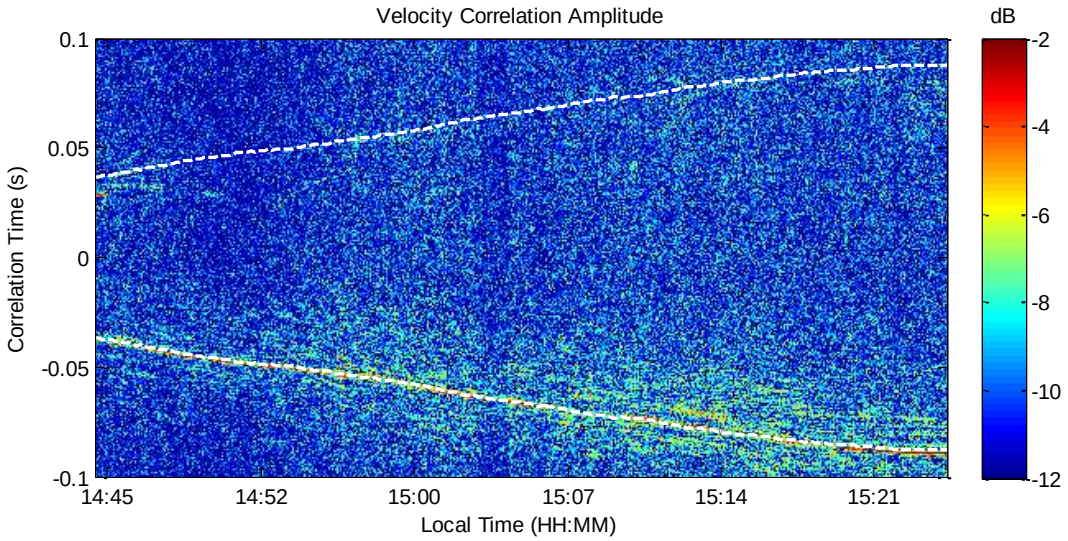


Figure 72: Velocity correlation ensemble amplitudes for quiet time 2. The amplitude is relative to the overall maximum amplitude, and is 10 $\log_{10}$ of the Hilbert transform magnitude.

Note that the correlations are very homogenous, with few interfering ridges. They also are more similar to the per-correlation normalized plots in Figure 43 and Figure 45 (see Section 3.3.2), since the per-correlation maximum is nearly the same as the overall maximum. The only interfering source seen during this quiet time is on the negative time delays, and was likely the result of the tug "Pocomoke", which was tracked (using AIS data) traveling along the array's endfire direction. It traveled from 8 km to 15 km from the array over this period. Despite the presence of this interferer, the positive time delays were relatively clean and presented the opportunity to identify ambient noise correlation peaks.

To identify the hidden ambient noise peaks in the positive time delays of the correlation ensembles plotted in Figure 71 and Figure 72, the same genetic algorithms presented in Section 4.2.3 for simulation results are employed. Where possible, all parameters used were identical. However, the main difference in the experimental results is the use of the GPS-measured separation distances as the source of the starting guess and

window constraints (see Equation 75). A fundamental difference between the simulated and experimental cases is that the GPS-measured distances themselves have errors associated, and cannot be assumed to be the "ground-truth" solution as was the case in simulation.

For this reason, it is important to verify the results by plotting the SNR gain over time, or relative to the assumed solution (e.g. GPS-derived). The SNR gain is defined as the PSNR (see Equation 72) of the shifted and summed correlations. If stated as a single number, it is assumed to be the PSNR of the entire shifted and summed correlation ensemble. If plotted as a function of time, it is assumed to be the PSNR of the sum only until that point in time. Defining the PSNR as a function, as defined in Equation 72, the SNR gain at a particular time t_G is given by

$$SNR_G(\hat{\boldsymbol{\tau}}, t_G) = PSNR\left[\sum_{i=0}^{G-1} \hat{C}_{12}(\tau + \hat{\tau}_i, t_i)\right] \tag{80}$$

where $\hat{\boldsymbol{\tau}} = [\hat{\tau}_0, \hat{\tau}_1, \dots, \hat{\tau}_{N_c-1}]$. If SNR_G is given with no explicit dependence on time, it is assumed the sum is taken over all correlations in the ensemble (i.e. $G = N_c$). Note that the estimated time delays $\hat{\tau}_i$ could be either a stochastic search solution, or any assumed prior solution (e.g. GPS-derived). To compare two different solutions, their SNR gains can be compared using a relative ratio

$$SNR_{rel}(\hat{\boldsymbol{\tau}}_\mathbf{1}, \hat{\boldsymbol{\tau}}_\mathbf{2}, t_G) = \frac{SNR_G(\hat{\boldsymbol{\tau}}_\mathbf{1}, t_G)}{SNR_G(\hat{\boldsymbol{\tau}}_\mathbf{2}, t_G)} \tag{81}$$

which gives the ratio of the SNR gains. As before, the omission of an explicit time t_G indicates the SNR gain is taken for the entire correlation ensemble. In this way, the

stochastic search solution can be compared to a prior estimate such as the GPS-estimated solution.

To compare the performance of different methods, both RMSE (see Equation 79) and SNR_{rel} will be examined. Three different methods were compared, with different values of $N = [10, 20, 30]$. Both pressure and axial velocity correlations are compared, and the correlation derivative is used to simplify the peak detection (see discussion in Section 3.2.6). Lastly, the averaging length of the correlation ensemble was varied using the values [1, 2, 5, 10, 20, 30] seconds.

The first method of stochastic search was the "unconstrained" genetic algorithm. It is identical to the method used in simulation results, where the only constraint is a rectangular window within which the solution must lie. The bounds of this window were obtained by expanding the limits of the GPS-measured separation distances with an expansion coefficient $K_B = 4$ (see Equation 74). Lastly, the starting population estimates were determined from downsampled separation distance measurements taken every 10 minutes. Thus, the starting estimate of the genetic algorithm is linearly-interpolated from GPS measurements assumed to be taken only every 10 minutes, despite them being sampled at 10 second intervals.

The second method compared was the "constrained" genetic algorithm, which is again identical to the method used in simulation. The difference expansion coefficient K_D was taken to be 5, thus allowing the largest difference in successive time delays to be five times that seen in the GPS-measured separation distance. Lastly, the starting estimates were obtained from linearly-interpolated 10-minute GPS updates.

The third method is almost identical to the "constrained" method, using the same constraints and parameters for the genetic algorithm. However, the starting estimate was instead linearly interpolated from the measured GPS separation distances, sampled with the original period of 10 seconds. This method is included in the event both methods 1 and 2 fail for a certain set of parameters. It indicates whether the estimation failed due to the problem being impossible to solve even with the "correct" answer as a starting point.

4.2.5 Experimental Results

The first results presented are for the positive time delays of the 14:45 – 15:25 period using the pressure and axial velocity correlations between sensors 0004 and 0005. For demonstrative purposes, a sample set of solutions are plotted between 15:12 and 15:17 in Figure 73, Figure 74, and Figure 75. The sample is taken from results run using the velocity correlation derivative, with an averaging length of 2 seconds, and N = 10 parameters estimated per segment.

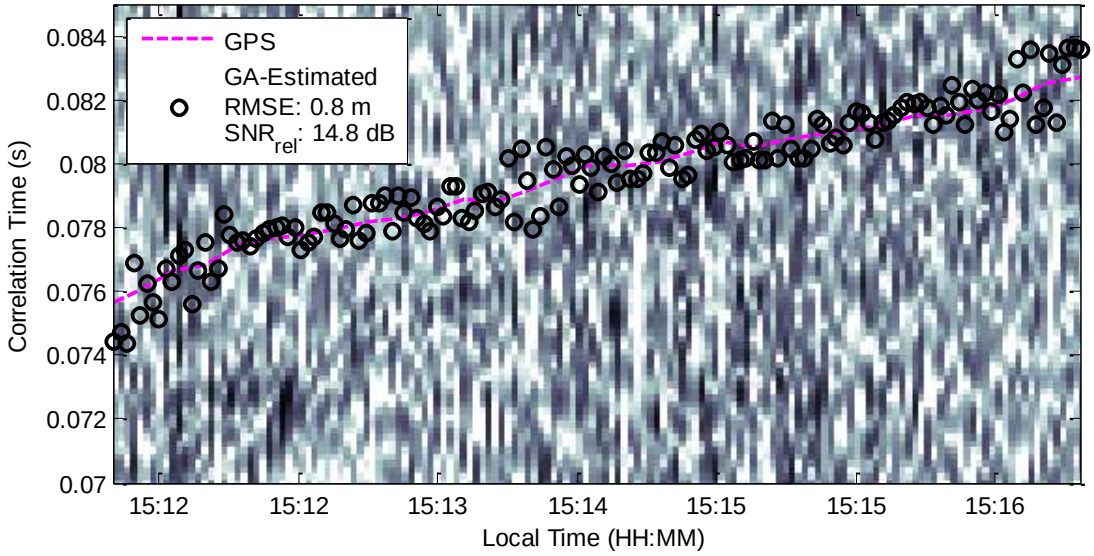


Figure 73: Sample solution for unconstrained genetic algorithm (black circles). The GPS-estimated correlation lag is shown in magenta, and the underlying velocity correlation derivative amplitude (10 $\log_{10}$ magnitude of Hilbert) is plotted with black = 0 dB and white = -10 dB.

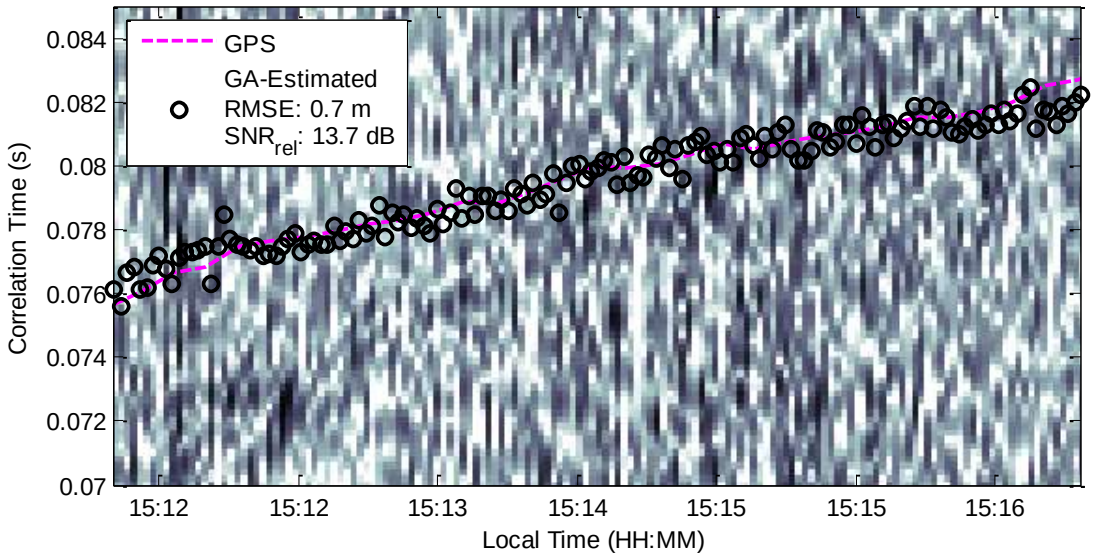


Figure 74: Sample solution for constrained genetic algorithm, started with downsampled GPS estimates (black circles). The GPS-estimated correlation lag is shown in magenta, and the underlying velocity correlation derivative amplitude (10 $\log_{10}$ magnitude of Hilbert) is plotted with black = 0 dB and white = -10 dB.

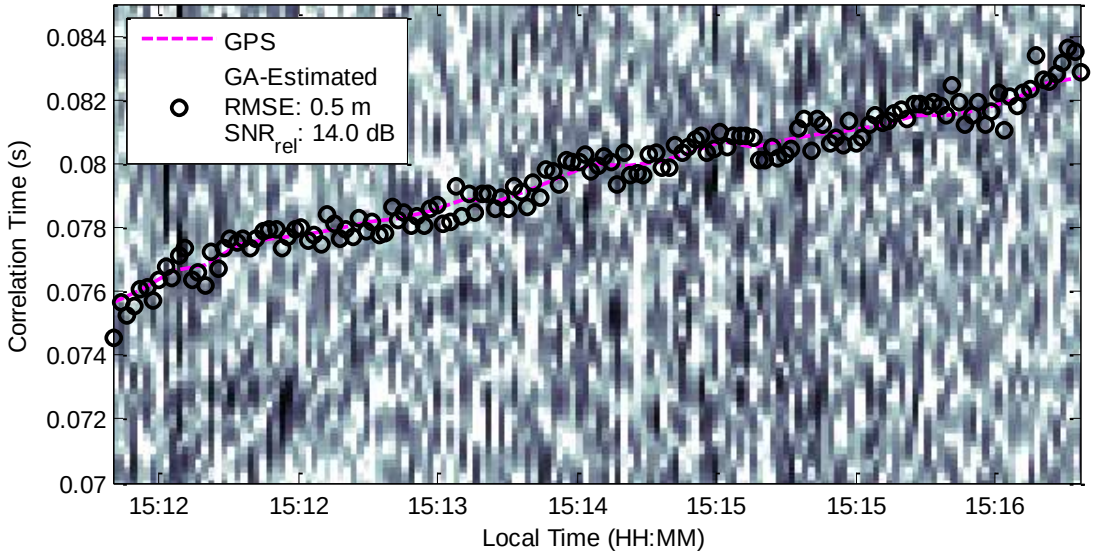


Figure 75: Sample solution for constrained genetic algorithm (black circles), started with the GPS-estimated correlation lag (magenta). The underlying velocity correlation derivative amplitude (10 $\log_{10}$ magnitude of Hilbert) is plotted with black = 0 dB and white = -10 dB.

From the three sample solutions for each method, the RMSE is visible as the "spread" of the solution from the GPS-estimated separation distance (magenta line). Furthermore, the "realism" of the solution is apparent in the frequency and magnitude of large differences between estimates (e.g. the unconstrained estimate is highly unrealistic around 15:14). Lastly, the SNR gains of each method are not readily compared, since the averaging length is too short for the correlation peaks to be visible to the naked eye. As this is simply a small selection of a great number of solutions, they will not be presented for every set of parameters. Their RMSE (see Equation 79) and SNR_{rel} (see Equation 81) will be compared instead in the following figures.

Plotted in Figure 76 is the RMSE (with respect to GPS) of the solutions obtained for the three different genetic algorithms, for each of the three different numbers of

parameters estimated. Figure 77 shows the same results, but omitting the unconstrained results so the differences between constrained methods become visible.

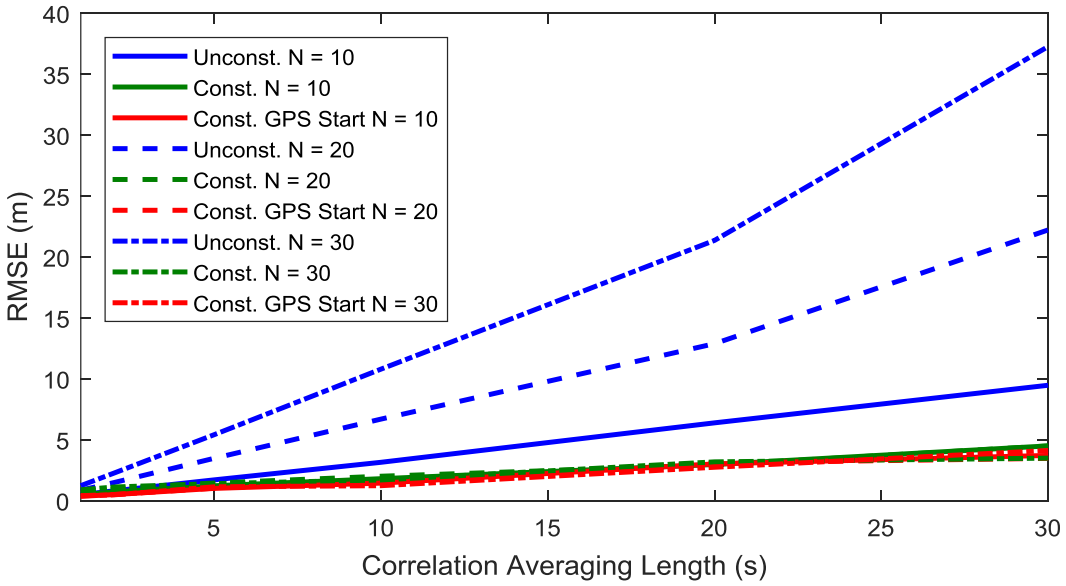


Figure 76: Pressure correlation derivative RMSE of genetic algorithm methods vs. averaging length. Unconstrained is shown in blue, constrained with downsampled GPS starting in green, and constrained with GPS starting estimates in red. The segment length N = 10 is shown as solid lines, N = 20 as dashed lines, and N = 30 as variable-dashed line.

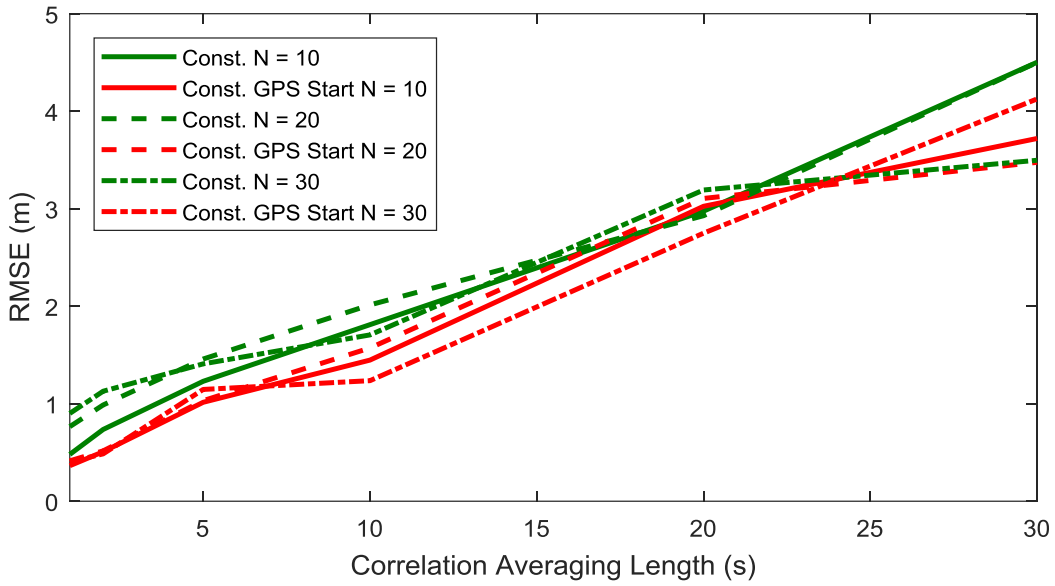


Figure 77: Pressure correlation derivative RMSE of constrained genetic algorithm methods vs. averaging length. Constrained with downsampled GPS starting is in green, and constrained with GPS starting estimates in red. The segment length N = 10 is shown as solid lines, N = 20 as dashed lines, and N = 30 as variable-dashed line.

The results of Figure 76 show that an increase in the averaging length of the correlations increases the RMSE of the results, as was seen in the simulation results of Section 4.2.3. For the unconstrained genetic algorithm, the increase in stochastic segment length N also increases the RMSE, which was an effect also seen in simulation results. This result does not hold for the constrained methods, as only the averaging length seems to influence the RMSE. The number of parameters estimated does not seem to detriment the estimate, since the difference constraint helps keep the entire solution feasible even for larger numbers of parameters. Put another way, the estimation with only 10 parameters might be nearly identical to identifying 30 parameters if the extra 20 parameters are simply "following" the trend of the other 10 due to the difference constraint. A final takeaway from Figure 77 is that using the GPS measurements as the starting estimates, rather than the downsampled estimates, provides a minor improvement in the resulting error. This is

to be expected, since the RMSE is assuming the GPS is the correct solution, and thus starting the genetic algorithm with this solution ought to result in a maximally-similar solution.

Similar results for the velocity correlations were obtained, the RMSE of which are plotted in Figure 78 and Figure 79.

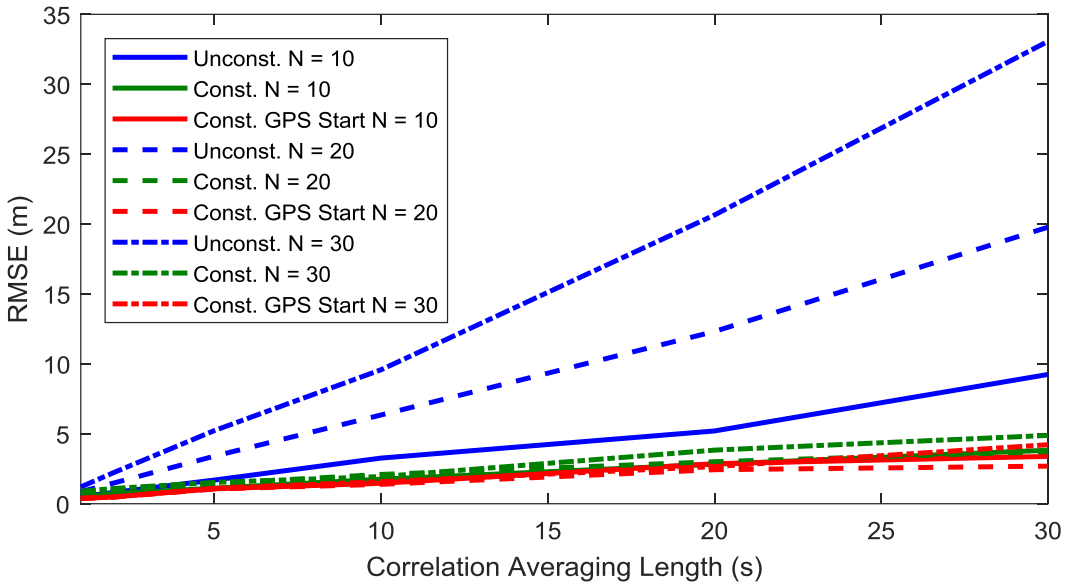


Figure 78: Velocity correlation derivative RMSE of genetic algorithm methods vs. averaging length. Unconstrained is shown in blue, constrained with downsampled GPS starting in green, and constrained with GPS starting estimates in red. The segment length N = 10 is shown as solid lines, N = 20 as dashed lines, and N = 30 as variable-dashed line.

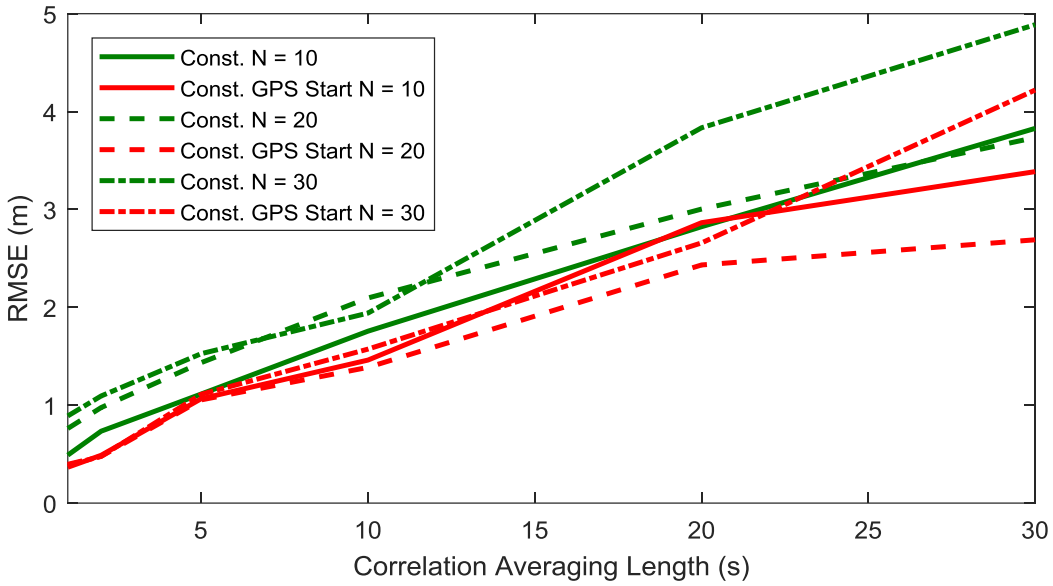


Figure 79: Velocity correlation derivative RMSE of constrained genetic algorithm methods vs. averaging length. Constrained with downsampled GPS starting is in green, and constrained with GPS starting estimates in red. The segment length N = 10 is shown as solid lines, N = 20 as dashed lines, and N = 30 as variable-dashed line.

The results are so similar in fact, they are not plotted together, as the resulting plot is quite difficult to interpret. However, a different error metric is employed to further highlight their differences. The SNR of the shifted and summed correlations using the genetic algorithm delays is compared to the SNR of shifted and summed correlations using GPS delays. This indicates how much better the genetic algorithm is at obtaining coherent peak locations, and the metric is defined as (see Equation 81)

$$SNR_{rel} = \frac{SNR_G(\hat{\boldsymbol{\tau}}_{\boldsymbol{GA}})}{SNR_G(\hat{\boldsymbol{\tau}}_{\boldsymbol{GPS}})} \qquad 82$$

Figure 80 is comparable to Figure 76, except that the metric plotted is SNR_{rel} (see Equation 82). Figure 81 shows the same SNR_{rel} metric, but plotted for the velocity correlations.

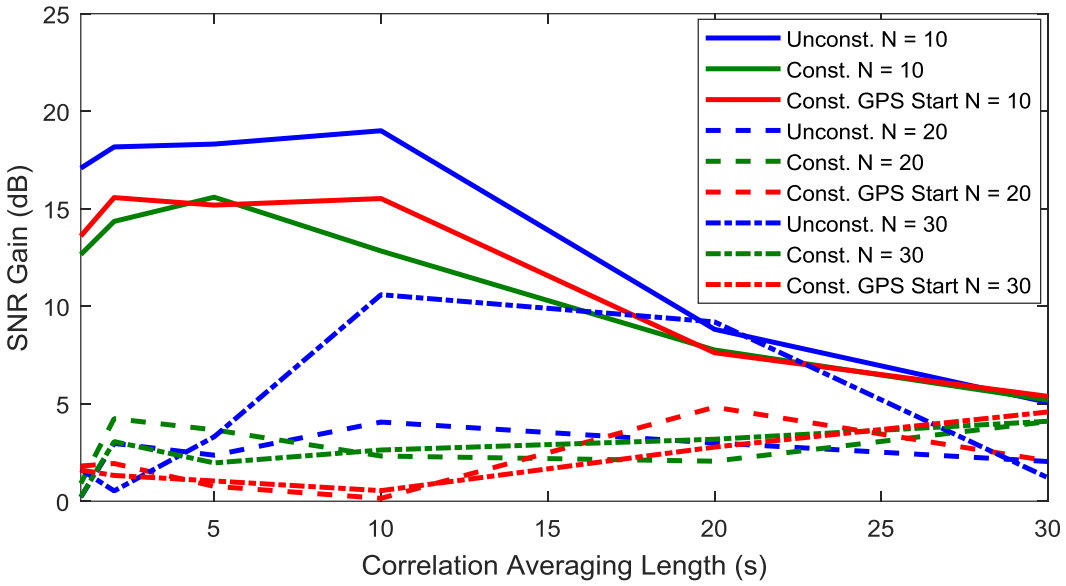


Figure 80: Pressure correlation derivative SNR_{rel} of genetic algorithm methods vs. averaging length. Unconstrained is shown in blue, constrained with downsampled GPS starting in green, and constrained with GPS starting estimates in red. The segment length N = 10 is shown as solid lines, N = 20 as dashed lines, and N = 30 as variable-dashed line.

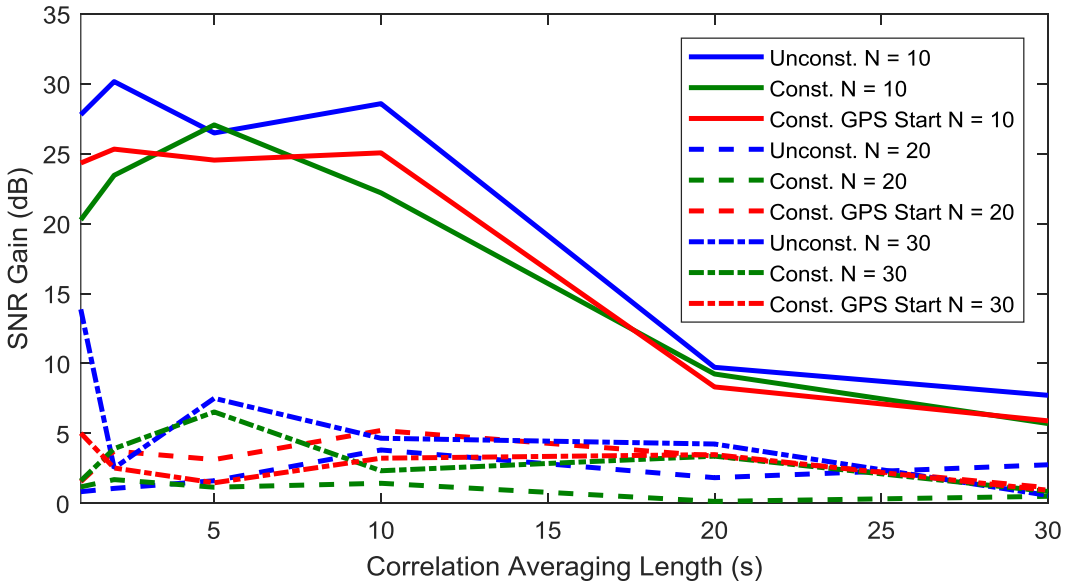


Figure 81: Velocity correlation derivative SNR_{rel} of genetic algorithm methods vs. averaging length. Unconstrained is shown in blue, constrained with downsampled GPS starting in green, and constrained with GPS starting estimates in red. The segment length

N = 10 is shown as solid lines, N = 20 as dashed lines, and N = 30 as variable-dashed line.

From Figure 80 and Figure 81, it becomes clear that the number of parameters is important for all three methods. For N = 20 and 30, the SNR gain relative to the SNR gain of the GPS-estimated separation distance is not consistently different from unity. However, when N = 10, the SNR_{rel} follows a more consistent trend. For short averaging lengths, the SNR gain relative to GPS is high, and tapers off for longer averaging time to about 7 dB. Also, there appears to be a maximum SNR gain between 2 and 10 second correlation averaging lengths. This could be due to the drift rate of ~200 m/hr which designates the averaging time be between 2 and 10 seconds to limit the "smearing" of correlation peaks. The unconstrained method obtains the highest gain relative to GPS, which indicates that despite its larger RMSE, it is finding higher SNR solutions. This indicates that both RMSE and SNR_{rel} are important metrics in identifying the validity and quality of a solution. The RMSE assumes the GPS solution is the exact, true solution. This most certainly is not the case, since GPS errors and sensor/antenna drift could cause the true solution to be inaccurate by as much as 1-2 meters. This is demonstrated by the stochastic search SNR gains being higher than those of the GPS-estimated separations, indicating the GPS separations are actually sub-par. Thus, without a precisely-known true solution, the RMSE does not fully encapsulate the performance of a certain method.

Plotted below in Figure 82 and Figure 83, the pressure and velocity RMSE and SNR_{rel} are compared for N = 10, the case in which the methods were able to identify peaks which increased SNR.

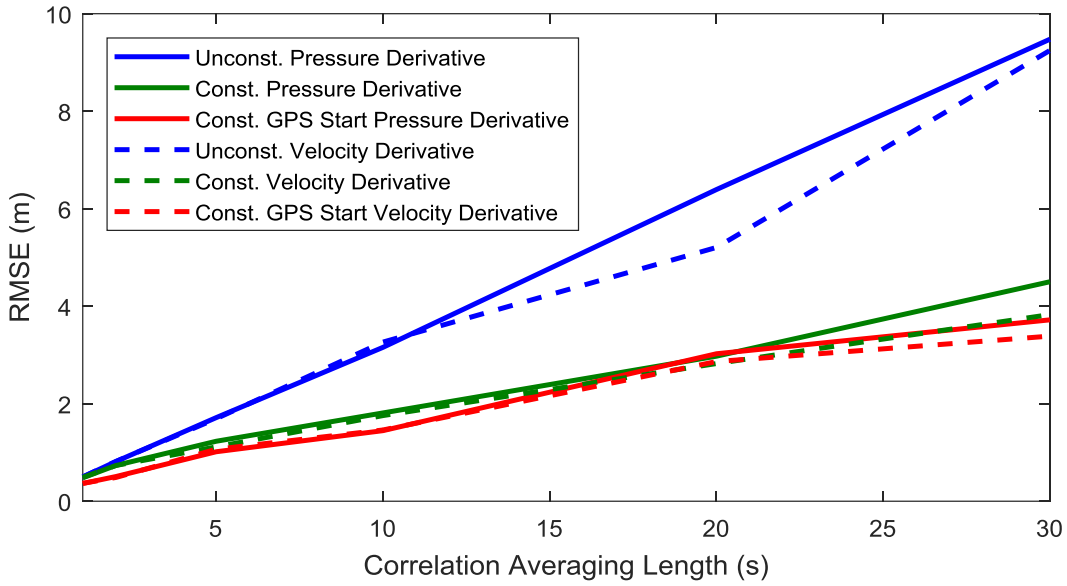


Figure 82: RMSE for pressure (solid) and velocity (dashed) correlation derivative genetic algorithm methods vs. averaging length. Unconstrained is shown in blue, constrained with downsampled GPS starting in green, and constrained with GPS starting estimates in red. Only results for N = 10 are shown.

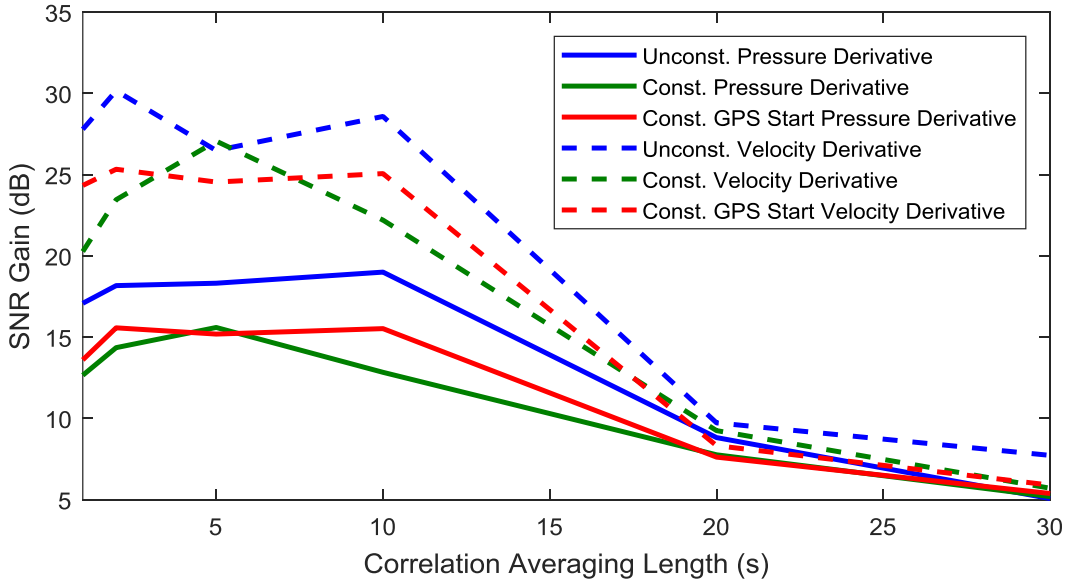


Figure 83: SNR_{rel} for pressure (solid) and velocity (dashed) correlation derivative genetic algorithm methods vs. averaging length. Unconstrained is shown in blue, constrained with downsampled GPS starting in green, and constrained with GPS starting estimates in red. Only results for N = 10 are shown.

The results indicate that the estimates made from velocity correlations (dashed lines) had higher SNR gain relative to the GPS than the pressure correlation estimates (solid lines) by a factor of about 1.5. However, the difference in performance using RMSE as the metric is not as conclusive. Comparing the three methods, both pressure and velocity correlations perform best for shorter averaging times. Again, starting the constrained genetic algorithm with the true GPS separation was slightly better than using the downsampled GPS separation (red vs. green curves).

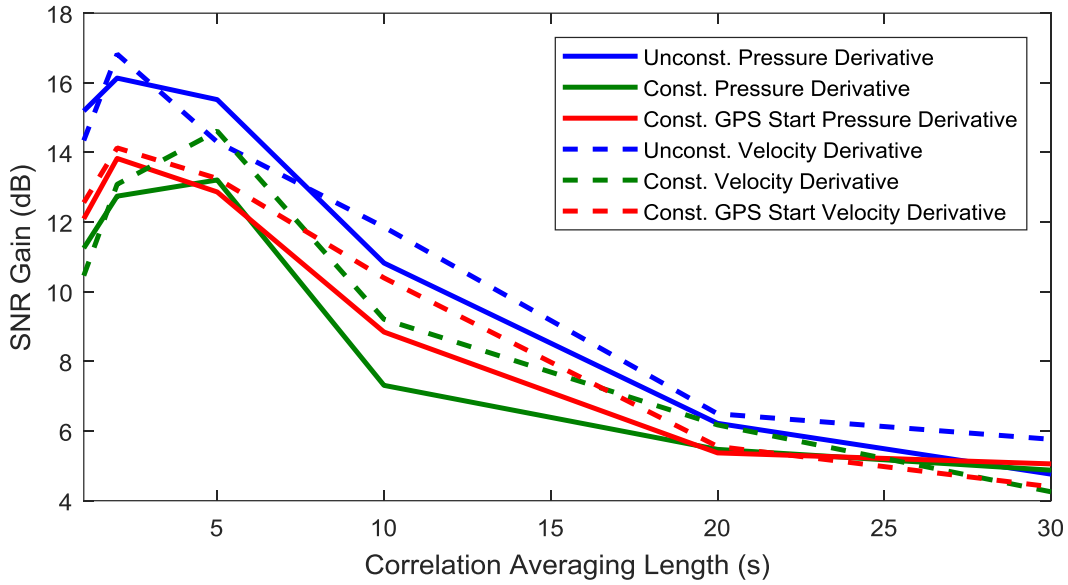


Figure 84: SNR_{avg} for pressure (solid) and velocity (dashed) correlation derivative genetic algorithm methods vs. averaging length. Unconstrained is shown in blue, constrained with downsampled GPS starting in green, and constrained with GPS starting estimates in red. Only results for N = 10 are shown.

Shown in Figure 84 is the plot of just the SNR gains of each method (i.e. not relative to the GPS gain). The similarity of the SNR gains for pressure and velocity correlations indicates it is the GPS gain which is lower for the velocity correlations, leading to a higher SNR_{rel}. Since the GPS separation is identical for pressure and velocity, this must mean the velocity correlations have worse coherent averaging along the GPS separation delays than

pressure. This could be due to the numerical integration required to go from acceleration to velocity, adding noise to the resulting correlations that is not present in the pressure correlations. There may also be more extrinsic noise present on the acceleration channels due to their mounting situation (see Section 2.2.3). This effect is best shown by plotting the SNR gain of each method over time (see Figure 85).

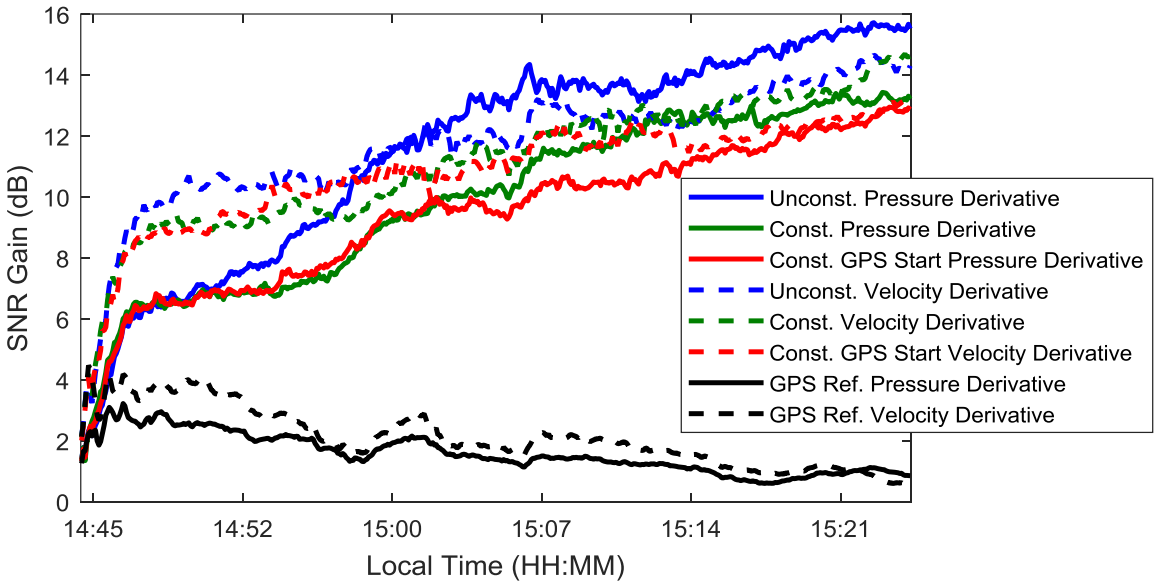


Figure 85: SNR_{avg} over time for pressure (solid) and velocity (dashed) correlation derivative genetic algorithm methods. Unconstrained is shown in blue, constrained with downsampled GPS starting in green, and constrained with GPS starting estimates in red. Only results for N = 10, T = 5 seconds are shown. The SNR_{avg} of the GPS-measured time delays is shown in black.

Figure 85 indicates the genetic algorithms can continue increasing the SNR of the shifted and summed correlations as time goes on. The GPS-estimated separation distances, however, do not increase the SNR of the shifted and summed correlations. In other words, the GPS-estimated separation distances do not accurately describe the ambient peak locations. Another result of Figure 85 shows that the SNR gain of the velocity correlation is higher than that of pressure until the very end of the time window. It is possible that an

event around 15:21 caused the velocity SNR gain to decrease when using GPS. The result is that the SNR_{rel} of velocity seems to be higher than pressure when compared at 15:25, even though that is not the case when comparing SNR gains at different times (e.g. 15:14).

4.2.6 Conclusion

Simulations of Green's function peak extraction were performed using unconstrained and constrained genetic algorithms. Unconstrained searches with large numbers of parameters is unsuited for precise detection of the Green's function peak locations, however, constraining the solution to be physically admissible allows the genetic algorithm to find the peaks to within GPS error.

Experimental results were difficult to come by due to the limitations of the experimental data; however, the preliminary results corroborate the simulation results. Namely, **velocity correlations outperform pressure correlations** due to the reduced noise present in a velocity correlation. Also, **shorter correlation averaging provides better results** since the correlation peaks will add coherently, rather than being "smeared" during the averaging process. Lastly, the **stochastic search length should be as short as possible**, yet long enough that coherent peaks can form if the correct solution is chosen. Otherwise, the stochastic search may settle on a sub-par local optimum.

As for the stochastic search method employed, it does not appear to depend much on the exact search methodology used. However, some form of constraint is necessary to maintain physically admissible solutions. Unconstrained genetic algorithm results were able to maximize SNR gain, but at the cost of realistic solutions. This may indicate a more optimal choice of objective function exists which considers physical constraints, rather than applying constraints explicitly. Constraining the difference between successive

estimates provided a simple and functional method for estimating peak locations. The performance of such methods in estimating sensor separation distance was comparable, or arguably better than the GPS-estimated distances.

With a method available for identifying the sensor separation distance between any pair of sensors in an array, the remaining problem (see Section 4.3) is to correct the locations of each sensor in the array, given their estimated separation distances.

4.3 Array Element Localization / Correction

4.3.1 Simple Theory

Assume an array is composed of N sensors, with estimated prior locations in need of correction. In relation to this work, the prior estimates come from GPS or dead-reckoning, and the corrections are made using the distances between each sensor measured by an ambient noise correlation process described in Section 4.2.1. Consider a vector of distances between each pair of elements

$$\vec{\boldsymbol{d}} = [d_{12} \quad d_{13} \quad \ldots \quad d_{23} \quad d_{24} \quad \ldots \quad d_{(N-1)N}]^T \tag{83}$$

where d_{ij} is the distance between sensor i and j. Note that $d_{ij} = d_{ji}$ and thus the distance vector contains only $N(N-1)/2$ elements to avoid redundant distances being included. There exists some non-linear function D such that

$$\vec{\boldsymbol{d}} = D(\vec{\boldsymbol{m}}) \tag{84}$$

where the parameter vector

$$\vec{\boldsymbol{m}} = [\vec{\boldsymbol{r}}_1^T \quad \vec{\boldsymbol{r}}_2^T \quad \ldots \quad \vec{\boldsymbol{r}}_N^T]^T \tag{85}$$

contains the 3-dimensional locations of each sensor $\vec{\boldsymbol{r}}_{\boldsymbol{i}}$. The non-linear function D cannot be simply inverted to solve for the parameter vector given measurements of the distances between sensors. Instead, it is linearized using a Taylor series expansion

$$\vec{\boldsymbol{d}} = D(\vec{\boldsymbol{m}}_{\boldsymbol{0}}) + J\,\overrightarrow{\boldsymbol{\delta m}} \tag{86}$$

around and operating point $\vec{\boldsymbol{m}}_{\boldsymbol{0}}$, with a Jacobian matrix J (see Appendix G) and differential parameter vector $\overrightarrow{\boldsymbol{\delta m}}$ (S. E. Dosso et al., 1998). Rearranging to put known terms together the resulting equation is given by

$$J\,\overrightarrow{\boldsymbol{\delta m}} = \vec{\boldsymbol{d}} - D(\vec{\boldsymbol{m}}_{\boldsymbol{0}}) \equiv \overrightarrow{\boldsymbol{\delta d}} \tag{87}$$

where $\overrightarrow{\boldsymbol{\delta d}}$ is defined as the difference between the measured inter-sensor distances and the inter-sensor distances resulting from the parameter vector guess $\vec{\boldsymbol{m}}_{\boldsymbol{0}}$. Following the gradient to arrive at a solution which minimizes the error $\overrightarrow{\boldsymbol{\delta d}}$ requires solving for the increment in parameter vector $\overrightarrow{\boldsymbol{\delta m}}$. The above equation is generally over-defined for large N, since the number of inter-sensor distances $N(N-1)/2$ exceeds the degrees of freedom $3N$ (or $2N$ for planar recalibration). Thus, a minimum square error solution is obtained using the pseudoinverse of J to find the parameter vector increment

$$\overrightarrow{\boldsymbol{\delta m}} = (J^T J)^{-1} J^T\,\overrightarrow{\boldsymbol{\delta d}} \tag{88}$$

Finding the solution for the parameter vector involves starting at an initial guess, such as the prior estimate obtained from GPS. Equation 88 is used to solve for an increment in parameter vector, and the increment is added to the prior. From this new point, the process repeats until the error vector $\overrightarrow{\boldsymbol{\delta d}}$ is suitably small. Figure 86 demonstrates a nominal test array of three elements, constrained to two degrees of freedom within the plane. The

measured distances are each 1 meter, and the resulting estimates of array locations are shown in Figure 87 after three iterations of the algorithm. Note that the array has settled into its expected equilateral triangle shape, however with a slight rotation from the nominal array. Although technically a correct solution, there may be a need to include errors in the distance measurements to account for the fact that some measurements may be more accurate, and thus considered more importantly, than others.

4.3.2 Theory with Distance Measurement Uncertainty and Prior Estimate Uncertainty

To include distance measurement uncertainty, the parameter vector solution should instead minimize the objective function

$$\phi = \left| G\left(J\,\overrightarrow{\boldsymbol{\delta m}} - \overrightarrow{\boldsymbol{\delta d}} \right) \right|^2 \tag{89}$$

where a distance error matrix G is defined by

$$G = \mathrm{diag}\left(\frac{1}{\sigma_{12}}, \frac{1}{\sigma_{13}}, \ldots, \frac{1}{\sigma_{(N-1)N}} \right) \tag{90}$$

(S. E. Dosso et al., 2004). Here, the value σ_{ij} denotes the standard deviation of the error in the corresponding distance measurement d_{ij} between sensors *i* and *j*. Now, the increment in solution estimate is given by

$$\overrightarrow{\boldsymbol{\delta m}} = [J^T G^T G\, J]^{-1} J^T G^T G\, \overrightarrow{\boldsymbol{\delta d}} \tag{91}$$

As before, the solution requires iterating from a starting point until a stopping criterion is met. In this case, the criterion is given by

$$\chi^2 \equiv \left|G(D(\vec{\boldsymbol{m}}) - \vec{\boldsymbol{d}}\right|^2 \lesssim N \tag{92}$$

where N is the total number of sensors. If the solution should account for prior estimate uncertainty in addition to distance measurement uncertainty, the parameter vector solution should minimize the objective function

$$\phi = \left|G\left(J\,\overrightarrow{\boldsymbol{\delta m}} - \overrightarrow{\boldsymbol{\delta d}}\right)\right|^2 + \mu\left|H\left(\vec{\boldsymbol{m}} - \vec{\widehat{\boldsymbol{m}}}\right)\right| \tag{93}$$

where μ is an arbitrary scaling parameter, $\vec{\widehat{\boldsymbol{m}}}$ is the prior/initial parameter vector estimate, and

$$H = \text{diag}\left(\frac{1}{\xi_{1x}}, \frac{1}{\xi_{1y}}, \ldots, \frac{1}{\xi_{Nz}}\right) \tag{94}$$

is the matrix of uncertainties in the prior estimate (S. E. Dosso et al., 2004). The standard deviation of the *i*-th sensor's x, y, or z location component is given by ξ_{ic}.

At each iteration, the parameter μ is selected such that the χ^2 value decreases by a factor of 10. This ensures the solution is approaching the minimum at a reasonable rate. If μ is too small, the solution overemphasizes the importance of the distance measurements, and if μ is too large, it overemphasizes the importance of the prior estimates. The updated parameter vector at each iteration is obtained by (S. E. Dosso et al., 2004)

$$\vec{\boldsymbol{m}} = [J^T G^T G\, J + \mu H^T H]^{-1}\left[J^T G^T G\left(\vec{\boldsymbol{\delta d}} + J\vec{\boldsymbol{m}}_{\mathbf{0}}\right) + \mu H^T H\vec{\widehat{\boldsymbol{m}}}\right] \tag{95}$$

where the value of μ was selected such that $\chi^2 \equiv \left|G(D(\vec{\boldsymbol{m}}) - \vec{\boldsymbol{d}}\right|^2$ is 10% of the previous iteration's value. The same stopping criterion is used as before.

The example nominal array shown in Figure 86 was passed into the AEL algorithm described above using two different sets of uncertainties. The first, whose solved array is shown in Figure 88, used very precise distance measurements and imprecise prior estimates. The result is nearly identical to that of Figure 87, since in that case the distance measurements were assumed perfect, and no prior measurement was considered. The second set of uncertainties, whose solved array is shown in Figure 89, used similar precision of the distance measurements and prior estimates, both imprecise enough to allow a solution which feasibly satisfies both the distance measurements and the prior location estimates. The solution is thus a hybrid of the prior array and equilateral triangle.

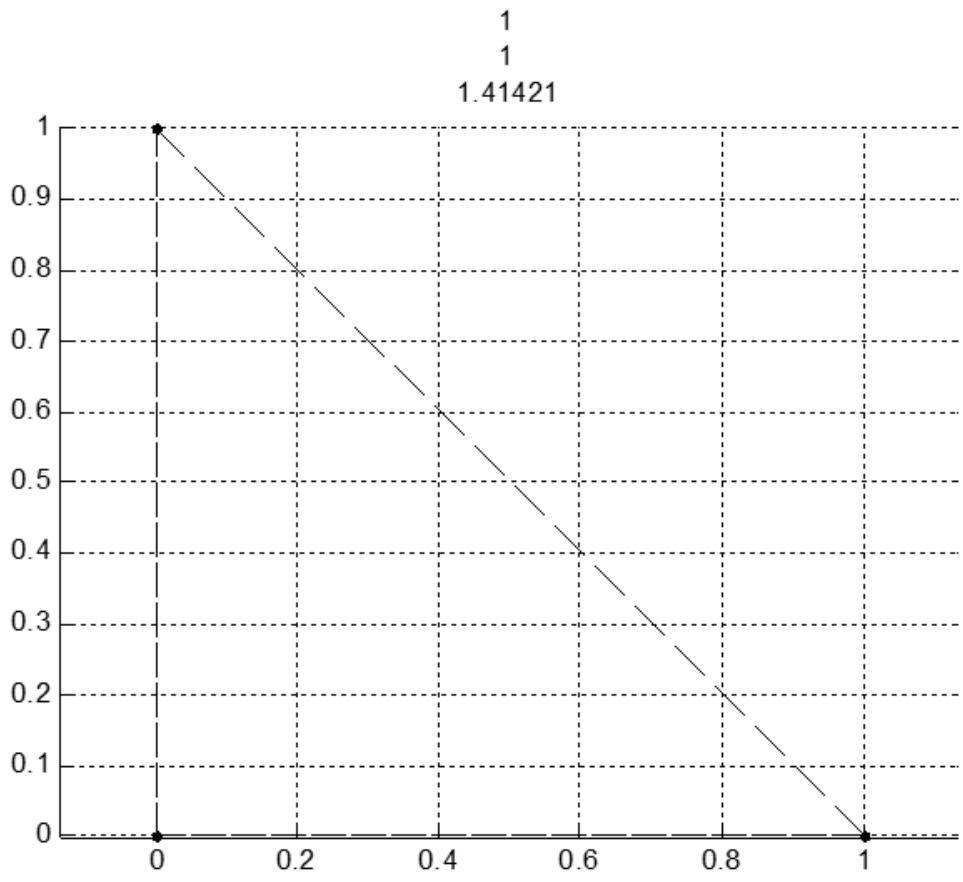


Figure 86: Nominal test array, with sensors shown as black dots, and their separation distances shown in the title, and drawn with dashed lines.

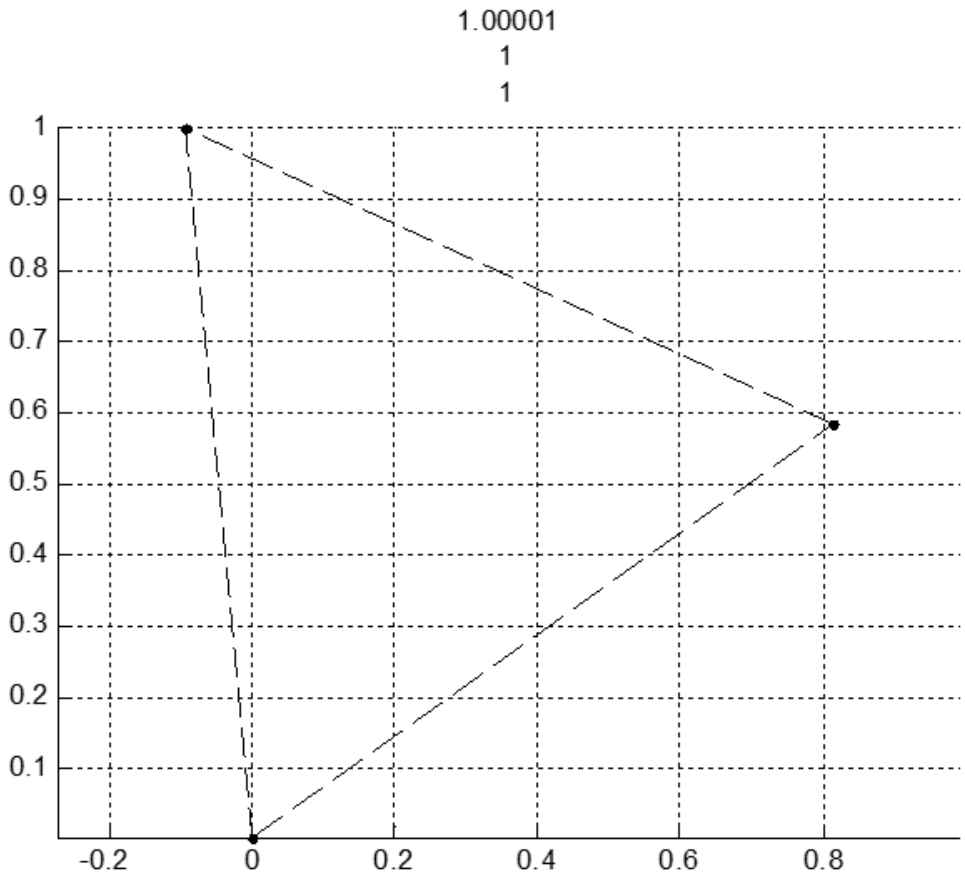


Figure 87: Sensor positions after three iterations. Sensors are shown as black dots, and their separation distances shown in the title, and drawn with dashed lines.

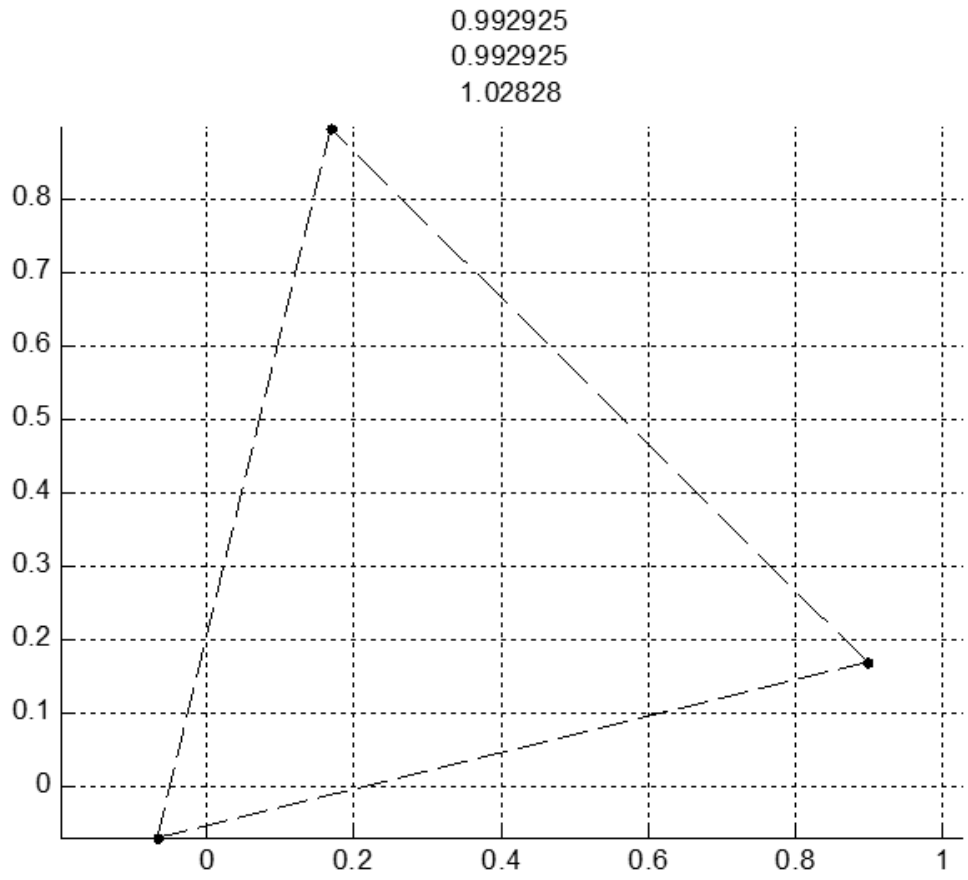


Figure 88: Sensor positions after solving regularized least-squares with σ=0.01 (precise distance measurements) and ξ=0.5 (very inaccurate initial estimates). Sensors are shown as black dots, and their separation distances shown in the title, and drawn with dashed lines.

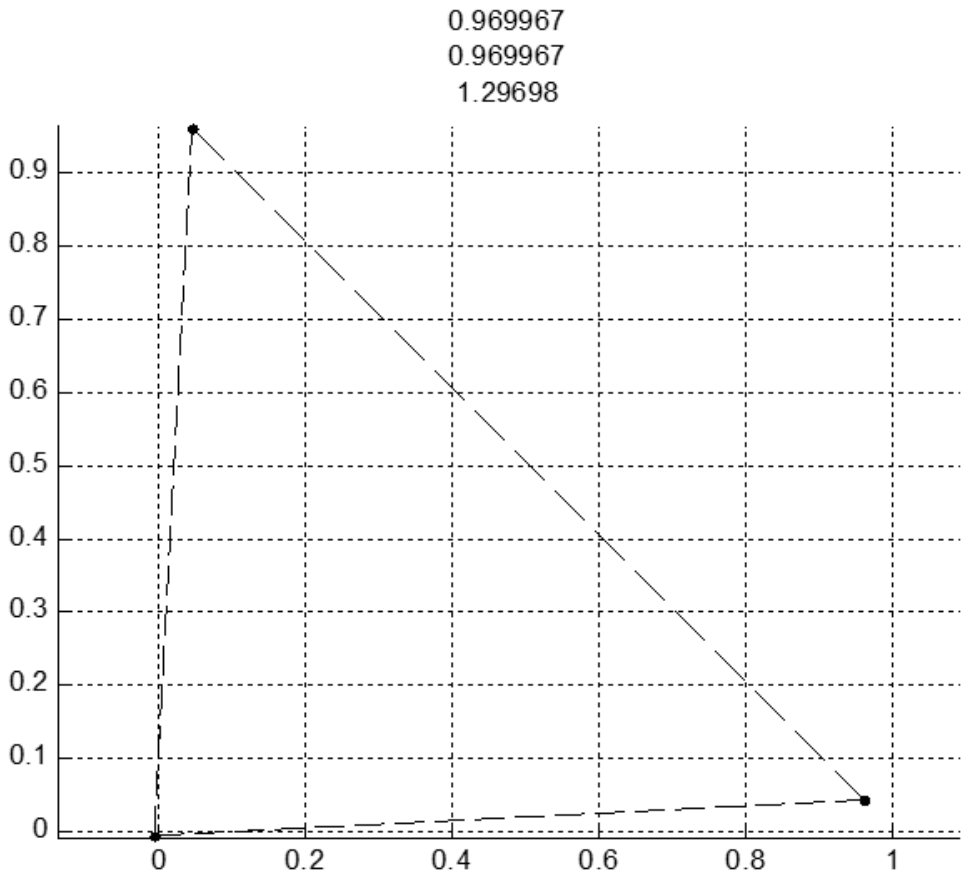


Figure 89: Sensor positions after solving regularized least-squares with σ=0.1 (accurate distance measurements) and ξ=0.1 (accurate initial estimates). Sensors are shown as black dots, and their separation distances shown in the title, and drawn with dashed lines.

The results shown above demonstrate the AEL algorithm's ability to consider the measurement and prior uncertainties appropriately. The prior uncertainty is rather well-known from GPS performance (Grimes, 2008). However, the measurement uncertainty of the ambient noise correlation estimation is not as well known. It is only known that the accuracy of the ambient noise estimation is comparable to that of GPS, assuming the conditions described in Section 4.2.6 are met. Furthermore, the experimental data limitations are such that only 2-3 elements can be used, and as such, only their 2D locations

corrected. Indeed, the simple case of a pair of elements means only the distance between them can be corrected.

4.4 Conclusion

Correlations of ambient noise recorded on separated vector sensors contains information on the separation distance between them. The goal is thus to leverage this information to correct the locations of sensors in an array, improving their source localization performance. However, the correlation peaks which need to be identified are buried in noise, since the drifting sensors limit the allowed correlation averaging length. A stochastic search method employing the genetic algorithm was successfully applied to identify the hidden correlation peak locations. When properly constrained, its performance meets or exceeds that of GPS. Furthermore, the method performs best when estimating peaks from the correlation derivative. Using the estimated peak locations, the distance between pairs of sensors is obtained, and can be passed into a simple regularized non-linear optimization algorithm for array element localization. An array whose sensor locations are corrected in this manner should have improved source localization performance (see Chapter 5).

CHAPTER 5
COHERENT SOURCE LOCALIZATION USING SPARSE VECTOR SENSOR ARRAYS

5.1 Introduction

Locating and tracking sound sources in an ocean environment is useful in a myriad of scenarios such as military surveillance, submarine warfare, wildlife monitoring, natural resource mining or seismic event tracking. Regardless of the application, source localization relies on measurements of the acoustic field to determine where a possible source is located. These measurements are taken on a number of separated sensors, referred to as an array. Array signal processing is the field which concerns the extraction of useful information from the array.

A subset of array signal processing, referred to as beamforming, generally attempts to invert the propagation process which created the recorded sound in the first place. For example, the traditional beamforming method guesses a source location, virtually propagates the sound to each sensor and then compares this expectation with the actual data recorded. A good match indicates the source was likely at the guessed location. Repeating for many different possible source locations will result in one with the highest output, marking the most likely source location. The first uses of beamforming for source localization were limited to direction-finding using narrowband hydrophone arrays (Van Trees, 2004). Many advancements have since been made to generalize the process for broadband processing, optimize the performance under a variety of conditions, and adaptively process the data (Van Trees, 2004).

Vector sensor beamforming was introduced later with the advent of sensors capable of measuring acoustic particle velocity (G. L. D'Spain, Hodgkiss, & Edmonds, 1991; Nehorai & Paldi, 1994). Some of the first developments utilize an intensity-based approach, where the product of pressure and velocity is used to determine the propagation direction (G. L. D'Spain et al., 1992). Another early development was an extension of the scalar (hydrophone) methods, which scales and adds pressure and velocity components (Gerald L. D'Spain et al., 2006; Hawkes & Nehorai, 1998). Just as in the scalar array case, these methods rely on a replica (or weight) vector to spatially filter the array data, and the replica which yields a large output is assumed to be indicative of the true source location.

However, a fundamental difference between vector sensor beamforming and hydrophone beamforming is that the replica contains the directional response of the sensor, as well as the propagation delays experienced by the assumed source. Having a directional component to the beamforming process means a single sensor is able to roughly determine the direction of a source, and an array of vector sensors typically performs better than an identical hydrophone array. This performance gain comes first from the directionality, which effectively carries more information about the source location than a hydrophone array. Secondly, the increased number of data channels helps improve the signal-to-noise ratio (SNR) in the presence of uncorrelated channel noise (Gerald L. D'Spain et al., 2006). Lastly, a vector sensor array is able to better reject spurious peaks in the beamformer output. These peaks could be the result of interfering sources, or the result of array geometry limitations such as left-right ambiguity in line arrays or grating lobes in sub-sampled arrays (Cox & Lai, 2009). The vector sensor replica ensures that sound arriving from these angles does not have as large a beamformer output due to the mismatched angles

(Chen & Zhao, 2004). The benefits afforded by a vector sensor array are not without cost though.

Noise is a more serious problem when using vector sensors, either caused by flow sensitivity, integration noise, electronic noise, or orientation measurement noise. Inertial-based types of vector sensors measure fluid acceleration rather than velocity, so they are very sensitive to local currents or movement of the sensor through the water (Lauchle et al., 2002). This effect manifests itself either through low frequency corrupting noise induced by currents/flow, or by impulsive events which are believed to be caused by sudden movement of the sensor through the water (see Section 2.2.3). Thirdly, the sensor technology may be such that the acceleration channels carry more electronic noise than pressure channels (J. C. Shipps & Deng, 2003). Finally, although not a form of noise recorded explicitly by the acceleration channels, there exists sensor orientation measurement noise. Both the gravitational vector and magnetic field vector are measured to obtain the orientation of the vector sensor so its components may be rotated into a common coordinate system for processing. Any noise or errors in the measurement of the orientation will manifest themselves as errors in the acceleration components. Regardless of the source of noise, vector sensors may experience a drastic decrease in localization performance, especially if only a single sensor is used.

One of the methods which has been used to mitigate these problems is to weight the pressure and velocity components depending on the amount of noise present (Gerald L. D'Spain et al., 2006). This method aims to maximize the detection performance of a vector sensor array. It also only allows control of the pressure and velocity inputs through the selection of their weights. If localization performance is the primary concern, the

optimum weights may differ. The weights could also be more general and encompass all components of the data covariance matrix to account for noise which is correlated for same-sensor components and uncorrelated across sensors. Furthermore, the positional uncertainty of the sensor locations is not accounted for in the prior work by D'Spain et al. Some past work has been done on optimal processing of perturbed hydrophone arrays (Schultheiss, 1980), and it was found that the optimum processing is not much better than traditional processing for small perturbations, and no benefit is gained for large perturbations. These results indicate that a perturbed array could benefit from an array processing scheme which is robust to both sensor noise and perturbations.

Section 5.2 introduces such a beamforming framework, which weights the data covariance matrix element-wise. It also proposes a novel weighting method which is both robust to local noise (e.g. flow noise) and sensor location errors. Section 5.3 examines the performance of traditional beamforming methods and the novel method using simulated and experimental data. Section 5.4 demonstrates the performance of the localization methods when the sensor locations are derived from various sources. The AEL methodology proposed in Chapter 4 is also validated in Section 5.4, where the localization performance is compared in relation to GPS-derived sensor locations.

5.2 Weighted Array Signal Processing

5.2.1 Vector Sensor Beamforming Theory

Assume N vector sensors are located in free space, and they record a time-domain data vector

$$\vec{\boldsymbol{d}}(t) = [p_1(t) \quad v_{1x}(t) \quad v_{1y}(t) \quad v_{1z}(t) \quad \dots \quad p_N(t) \quad v_{Nx}(t) \quad v_{Ny}(t) \quad v_{Nz}(t)]^T \quad 96$$

where a numerical subscript denotes the *i*-th sensor, and the *p, x, y, z* subscript denotes the pressure, or corresponding axis of velocity. Note that all components of velocity are assumed to be in the same coordinate system (see Section 2.3.2). Furthermore, assume all preprocessing steps have already been applied beforehand (see Section 2.3.4). The frequency domain data vector is obtained by taking the Fourier transform of the time-domain data vector, and produces

$$\vec{\boldsymbol{d}}(\omega) = [p_1(\omega) \quad v_{1x}(\omega) \quad v_{1y}(\omega) \quad v_{1z}(\omega) \quad \dots \quad p_N(\omega) \quad v_{Nx}(\omega) \quad v_{Ny}(\omega) \quad v_{Nz}(\omega)]^T \tag{97}$$

Pairwise correlation of each component of data can then be performed in the frequency domain by multiplying the spectra by their complex conjugate, which in matrix form is represented by the data covariance matrix

$$R(\omega) = d(\omega)d(\omega)^H = \begin{bmatrix} C_{11}(\omega) & C_{12}(\omega) & \dots \\ C_{21}(\omega) & C_{22}(\omega) & \dots \\ \vdots & \vdots & \ddots \end{bmatrix} \tag{98}$$

where the *H* superscript denotes Hermitian transposition (complex conjugate transpose) (Nichols & Sabra, 2015). The submatrices $C_{ij}(\omega)$ are 4x4 submatrices pertaining to the cross-spectra of data from sensor *i* to sensor *j* and is defined by

$$C_{ij} = \begin{bmatrix} p_i p_j^* & p_i v_{jx}^* & p_i v_{jy}^* & p_i v_{jz}^* \\ v_{ix} p_j^* & v_{ix} v_{jx}^* & v_{ix} v_{jy}^* & v_{ix} v_{jz}^* \\ v_{iy} p_j^* & v_{iy} v_{jx}^* & v_{iy} v_{jy}^* & v_{iy} v_{jz}^* \\ v_{iz} p_j^* & v_{iz} v_{jx}^* & v_{iz} v_{jy}^* & v_{iz} v_{jz}^* \end{bmatrix} \tag{99}$$

where the * superscript denotes complex conjugation and the explicit dependence on frequency is omitted for clarity. The diagonal block matrices C_{ii} are referred to as the incoherent matrices, as they contain only cross-spectra of components on the same sensor.

Off-diagonal block matrices $C_{ij},\ i \neq j$ are referred to as coherent matrices, as they require coherent cross-spectra of data across different sensors.

To locate a single source in free space, the classical Bartlett beamformer is employed (Van Trees, 2004)

$$B(\omega, \hat{\theta}) = w(\omega, \hat{\theta})^H d(\omega) d(\omega)^H w(\omega, \hat{\theta}) \qquad 100$$

where the search space vector $\hat{\theta}$ is an arbitrary vector describing the unknown source parameters, w is the steering (or replica) vector, and B is the beamformer output power. For simple direction of arrival (DOA) estimation, the parameter vector is simply the unknown angle from the array to the source. The angle which maximizes the output power is the assumed source bearing. In the case of an arbitrary sparse vector sensor array, the source may not be at the same bearing for each sensor, so the parameter vector must take this into account. For this chapter, the source's 3D location is the unknown, so the parameter vector is given by

$$\hat{\theta} = \vec{\boldsymbol{r}}_{\boldsymbol{s}} = [r_{sx} \quad r_{sy} \quad r_{sz}]^T \qquad 101$$

where the position of the source is $\vec{\boldsymbol{r}}_{\boldsymbol{s}}$.

The steering, or replica, vector in Equation 100 is formulated such that a correct match of the parameter vector to the parameters that generated the data vector d produces the maximum beamformer output power (Hawkes & Nehorai, 1998). In other words, the replica vector represents the data vector that would be recorded when the source is located at the estimated location $\hat{\theta}$. For a simple spherical wave emanating from the source without spherical spreading losses, the weight vector is given by

$$w(\omega, \hat{\theta}) = \begin{bmatrix} e^{j\omega\tau_1} & e^{j\omega\tau_1}\cos\theta_1\cos\phi_1 & e^{j\omega\tau_1}\sin\theta_1\cos\phi_1 & e^{j\omega\tau_1}\sin\phi_1 \\ e^{j\omega\tau_2} & e^{j\omega\tau_2}\cos\theta_2\cos\phi_2 & e^{j\omega\tau_2}\sin\theta_2\cos\phi_2 & e^{j\omega\tau_2}\sin\phi_2 \end{bmatrix}^T \quad (102)$$

where the relative time delay of the wave arriving at sensor *i* is τ_i, defined by the distance of sensor *i* to the source divided by an assumed constant sound speed c. The azimuth, θ_i, and elevation, ϕ_i, angles describe the angle of the source to the *i*-th sensor (see Figure 90 and Figure 91).

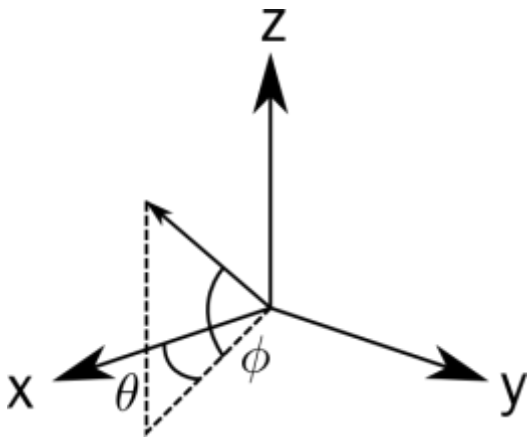


Figure 90: Vector sensor propagation direction convention, with azimuth angle θ and elevation angle ϕ.

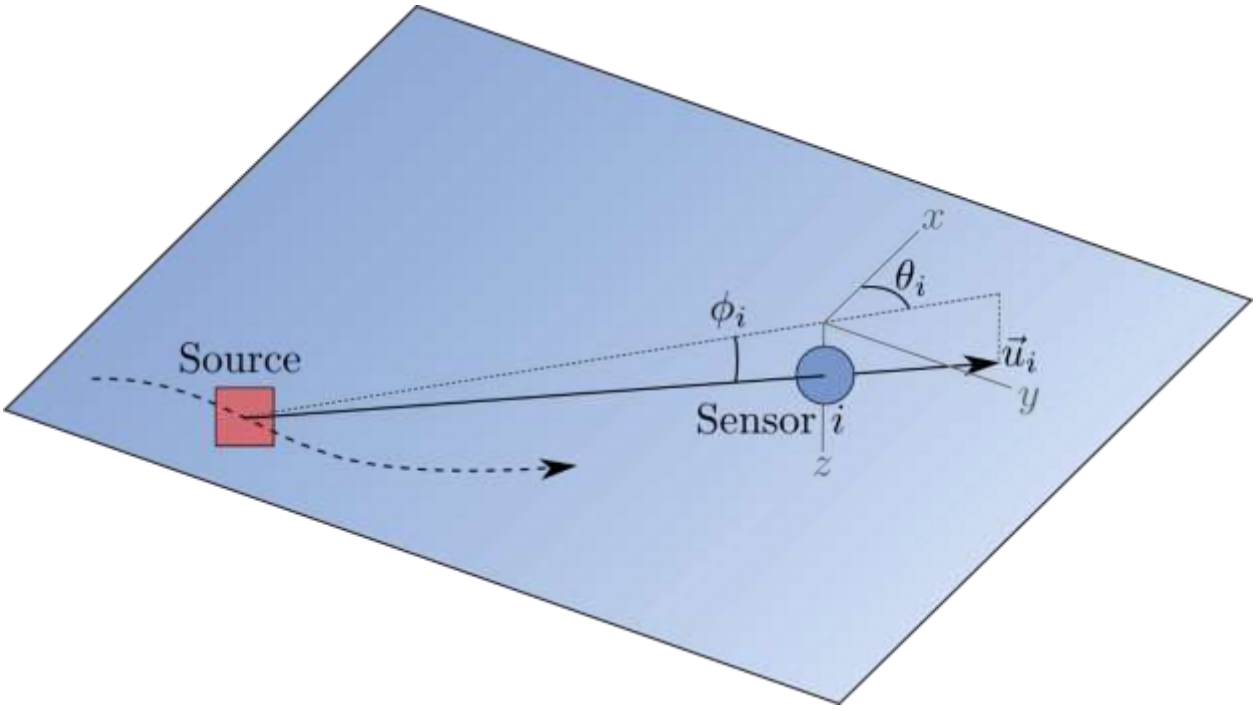


Figure 91: Propagation from source to sensor i, with azimuth angle θ_i and elevation angle ϕ_i. The x, y, z, coordinate system is assumed to be aligned with North, East, and Down respectively (NED coordinate system).

For any frequency and selection of a parameter vector, the beamforming process (see Equation 100) can be simply written as

$$B = w^H R w \tag{103}$$

where the covariance matrix R is defined in Equation 98. To locate a source using a wide band of frequencies, the beamformer output is calculated for each frequency as in Equation 100. Its time-domain response $B(t)$ is obtained through an inverse Fourier transform, and should contain a single peak centered at $t = 0$ if the parameter vector estimate matched the true parameters. For this reason, the wide-band beamformer output power is defined as

$$\Pi(\hat{\theta}) = \sqrt{\int \left(f(t)B(t,\hat{\theta})\right)^2 dt} \tag{104}$$

where $f(t)$ is a narrow time window centered at $t = 0$ whose width is inversely proportional to the signal bandwidth (Nichols & Sabra, 2015). A typical choice of $f(t)$ is a rectangular window whose width is $2/B$. This encapsulates the main lobe of the cross-spectrum peaks. Thus, an incorrect parameter vector won't shift the cross-spectral peaks to reside within the window, and the wide-band beamformer power will be smaller than that for the correct parameter estimate.

A typical application of the beamformer described here is to define a search space of likely candidates for the parameter vector, in this case a 3D search space for the source location. Then, for a discretized set of parameter vectors within the search space, the beamformer output power is computed. The estimated source location is given by the parameter vector which maximizes the output power, i.e.

$$\hat{\theta}_{\boldsymbol{s}} = \max_{\hat{\theta} \in S} \Pi(\hat{\theta}) \qquad 105$$

where S denotes the search space. In the case of tracking a surface vessel, the source's z position is known to be zero (i.e. at the surface). This allows the search space to be the 2D plane defined by $z = 0$. When plotted over this plane, the beamformer output power $\Pi(\hat{\theta})$ is also referred to as an ambiguity surface. An example of such an ambiguity surface is shown in Figure 92, where the beamformer output power is normalized between 0 and 1 and plotted as a function of the 2D search space.

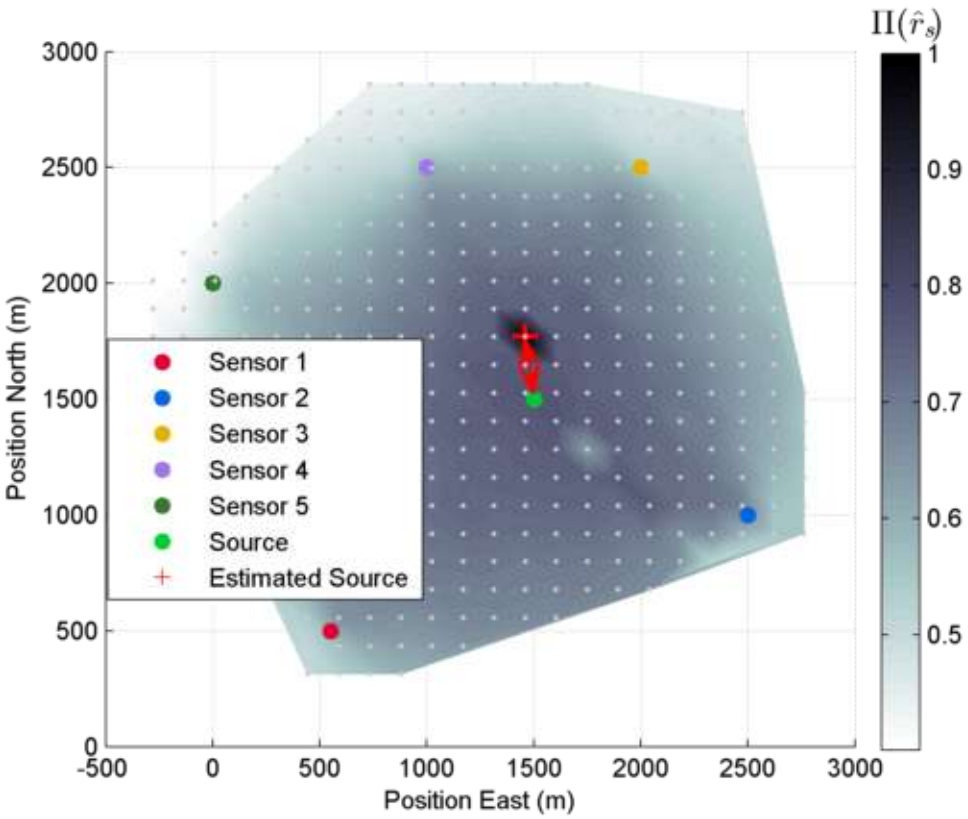


Figure 92: Sample ambiguity surface normalized to unit amplitude. The evaluation points are shown as gray dots, and the surface is interpolated between them. The estimated source location is given by the maximum of the ambiguity surface (red cross), and its error shown with a red arrow.

The gray dots in Figure 92 denote the discrete locations at which the beamformer output power was calculated, and the color of the surface is interpolated between these vertices. The darker the surface, the higher the beamformer power and thus, the more likely a source resides at that location. Shown as a red cross, the estimated source position $\hat{\boldsymbol{r}}_{\boldsymbol{s}}$ is located at the maximum of the ambiguity surface. The accuracy of the estimate can be represented as its distance to the actual source (red arrow), and is defined by

$$\epsilon_a = |\hat{\boldsymbol{r}}_{\boldsymbol{s}} - \vec{\boldsymbol{r}}_{\boldsymbol{s}}| \qquad 106$$

In addition to an accuracy metric, a precision metric is introduced which describes the probable error in the source location, or in other words, how much larger the beamformer

output is for the estimated source location relative to other locations. Thus, the precision of an estimate is defined by

$$\epsilon_p = 10\log_{10}\left(\frac{(\mathrm{P}-1)\,\Pi(\hat{\boldsymbol{r}}_{\boldsymbol{s}})}{\sum_{\vec{\boldsymbol{r}}_{\boldsymbol{s}}\neq\hat{\boldsymbol{r}}_{\boldsymbol{s}}}\Pi(\vec{\boldsymbol{r}}_{\boldsymbol{s}})}\right) \tag{107}$$

where P is the number of discrete points in the ambiguity surface. Thus, the precision gives a ratio of the peak beamformer output relative to the average non-peak output (that is equivalent to a peak-to-sidelobe ratio for the beamformer output). Together, the accuracy and precision define a beamforming method's ability to accurately locate a source and how precisely the location is known. Alternatively, the precision is a useful metric for source detection, as a precision of 0 dB indicates a constant ambiguity surface (i.e. no source present). Values much higher than 0 dB indicate the strong presence of a source.

5.2.2 Weighted Correlation Matrices

Performing beamforming as described in Section 5.2.1 with the covariance matrix equal to

$$R = \begin{bmatrix} C_{11} & C_{12} & \dots \\ C_{21} & C_{22} & \dots \\ \vdots & \vdots & \ddots \end{bmatrix} \tag{108}$$

is hereafter referred to as coherent beamforming, where all components of sensor cross-spectra are present in the beamformer output. The reason the covariance matrix is represented in the block-matrix format of Equation 108/98 is to highlight the properties of each block matrix. For example, the incoherent matrices C_{ii} represent correlations of data on the same sensor. If the sensor is in the presence of localized acoustic noise, it is possible the correlations within are corrupted by the noise. Likewise, if time-synchrony across

sensors is not guaranteed, the correlations in the coherent matrices $C_{ij}, i \neq j$, would not yield much useful information. For these reasons, one might be interested in optimizing the beamformer output to account for noise, time-synchrony errors, sensor positional errors, etc. To this end, the covariance matrix from Equation 108 can be element-wise weighted to produce a weighted covariance matrix

$$R_w = W \odot R \tag{109}$$

where the $\odot$ operator denotes an element-wise (Hadamard) product, and W is an arbitrary matrix of weighting components. The covariance-weighted beamformer output is then calculated for each frequency and estimated parameter vector as

$$B_w = w^H R_w w \tag{110}$$

As stated before, fully coherent processing uses all terms in the covariance matrix, thus its weight matrix is

$$W_C = \mathbf{1}_{4N} \tag{111}$$

where $\mathbf{1}_n$ denotes an *n* by *n* square matrix of ones. Incoherent processing only uses same-sensor correlations, thus its weight vector is

$$W_I = \begin{bmatrix} \mathbf{1}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \cdots \\ \mathbf{0}_4 & \mathbf{1}_4 & \mathbf{0}_4 & \\ \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{1}_4 & \\ \vdots & & & \ddots \end{bmatrix} \tag{112}$$

where $\mathbf{0}_n$ denotes an *n* by *n* square matrix of zeros.

In the event the vector sensors are subject to large amounts of local noise (e.g. flow, surface noise or biological noise), the incoherent terms may be unusable for source

localization (Hawkes & Nehorai, 2001). Although far from optimal, a novel method of weighting is proposed which nullifies the contribution of the incoherent terms, and is referred to as cross-coherent processing. Its weight matrix is given by

$$W_{CC} = \begin{bmatrix} \mathbf{0}_4 & \mathbf{1}_4 & \mathbf{1}_4 & \cdots \\ \mathbf{1}_4 & \mathbf{0}_4 & \mathbf{1}_4 & \\ \mathbf{1}_4 & \mathbf{1}_4 & \mathbf{0}_4 & \\ \vdots & & & \ddots \end{bmatrix} \tag{113}$$

Such a method uses only correlations across sensors, which are less affected by localized noise at each sensor. This is especially true for sparse arrays where the distances between sensors may be extremely large. Thus, the performance of the cross-coherent method should outperform the incoherent method as the noise levels increase. However, it will require more precise sensor location estimates as it is still a coherent processing method (Culver & Hodgkiss, 1988; S. E. Dosso et al., 2004). To evaluate the performance of each of these methods, simulated and experimental data are processed in the following section, and the performance metrics of Equations 106 and 107 are compared.

5.3 Localization Performance with Positional Uncertainty

5.3.1 Simulation Methodology and Results

Simulations were carried out in a simple free-space environment in which the source emits a spherical wave that is received by each sensor as a plane wave propagating radially (see Figure 93). Spherical spreading and impedance were neglected since the receivers are all assumed to be in the far field as well as whitened to normalize their received amplitudes. Furthermore, the simulations restricted the sensor and source locations to the surface plane, since the distances between sensors and source are much larger than any possible sensor depth.

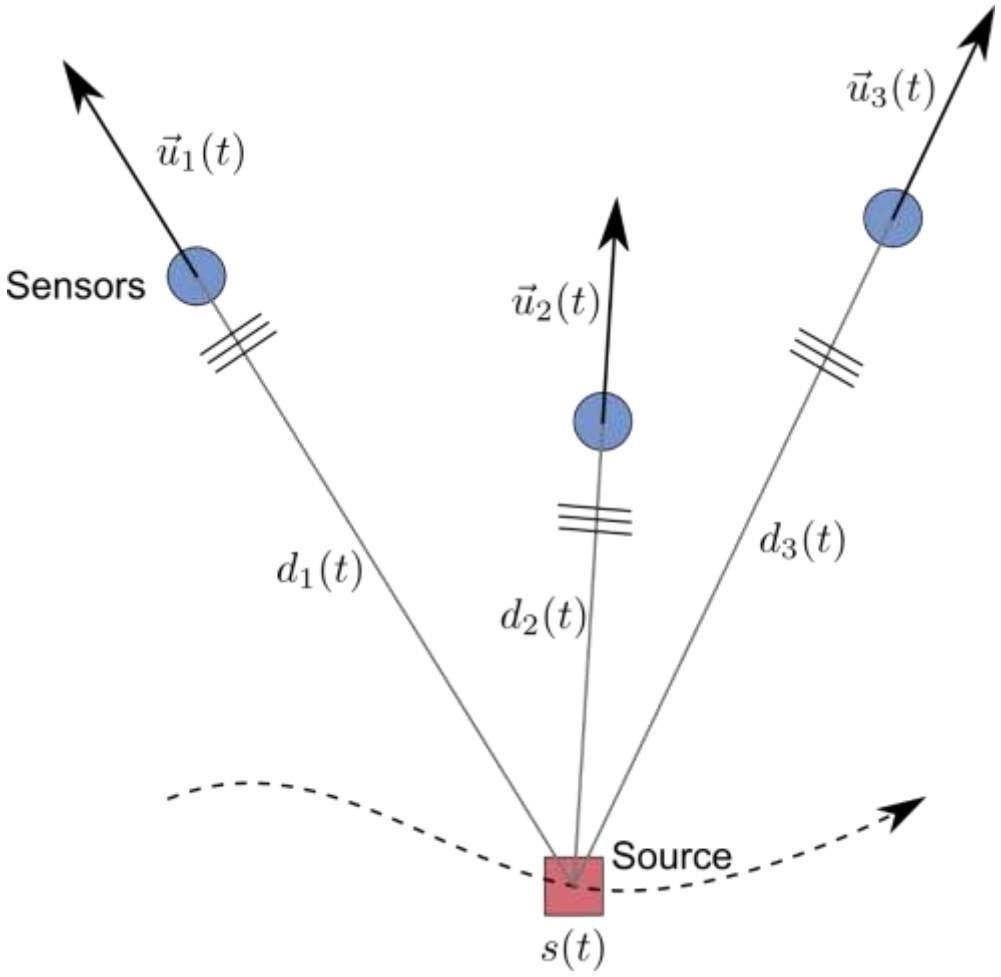


Figure 93: Simulation model, where the source signal $s(t)$ is broadcast and received by each sensor as a plane wave. The source signal is delayed by the distance to the source ($d_i(t)$) divided by a constant sound speed c. The velocity received by each sensor is $u_i(t)$.

The source is assumed to be a random point source radiator, broadcasting a signal $s(t)$ which is Gaussian white noise with unit power (or variance). Each sensor receives a delayed version of this signal plus additive noise whose SNR is defined by

$$SNR = \frac{\sigma_s^2}{\sigma_n^2} \qquad 114$$

where the signal power σ_s^2 is arbitrarily chosen to be unity, and the noise standard deviation is σ_n. The signal is assumed to propagate at a constant sound speed c across a distance $d_i(t)$ to reach the *i*-th sensor. Thus, the pressure received at the *i*-th sensor is

$$p_i(t) = s\left(t - \frac{d_i(t)}{c}\right) + SNR_p^{-1/2} n_{ip}(t) \qquad 115$$

where SNR_p is the SNR of the pressure channels, and $n_{ip}(t)$ is Gaussian white noise with unit power which is assumed uncorrelated to the source signal or any other additive noise signals. Note that the distance from source to sensor *i*, $d_i(t)$, should in fact be evaluated at a retarded time which indicates the distance when the signal was emitted. However, it is a fair assumption that the source and sensors are moving much slower than the speed of sound, and thus the distance does not change appreciably during the propagation time. Assuming a plane wave arrival at the *i*-th sensor, the velocity received is

$$\vec{\boldsymbol{v}}_{\boldsymbol{i}}(t) = \frac{\vec{\boldsymbol{u}}_{\boldsymbol{i}}(t)}{\rho_0 c} s\left(t - \frac{d_i(t)}{c}\right) + (3SNR_v)^{-1/2} \vec{\boldsymbol{n}}_{\boldsymbol{iv}}(t) \qquad 116$$

where $\vec{\boldsymbol{u}}_{\boldsymbol{i}}(t)$ is the vector pointing from source to sensor at time *t*, ρ_0 is the ambient density, c the sound speed, SNR_v the velocity channel SNR, and $\vec{\boldsymbol{n}}_{\boldsymbol{iv}}(t)$ the velocity Gaussian white noise signal with diagonal, unit covariance. The factor of three is introduced such that the power of the velocity signal magnitude to noise magnitude is given by SNR_v (Gerald L. D'Spain et al., 2006).

To simulate positional uncertainty in the sensor locations, a positional noise vector is added to the sensor locations after the data simulation has taken place with the nominal positions. The error vector is given by

$$\vec{\boldsymbol{e}}_{\boldsymbol{i}} = \sigma \cdot \left[e_x \; e_y \; 0\right]^T \tag{117}$$

with σ defining the standard deviation in position error, and e_x, e_y being Gaussian random variables with zero-mean and unit variance. Positional error is only added in the surface plane, since future simulations are neglecting the vertical positions. Furthermore, each sensor receives added positional noise which is uncorrelated in the *x* and *y* directions, as well as for different sensors. Although a simplification of the often-complex positional errors which may arise, including correlated errors across different sensors or the *x* and *y* axes, this analysis will prove useful as a worst-case scenario if σ is chosen to match the maximum value experienced in any direction.

To test the localization performance of the methods introduced in Section 5.2.2, simulated data were generated for 10 second snapshots with a sample rate of 5512.5 Hz and filtered in the band 100 – 800 Hz. The assumed sound speed was 1494 m/s and ambient density 992 kg/m^3. The source and sensor locations were constant across the 10 second simulation snapshot and are plotted in Figure 94.

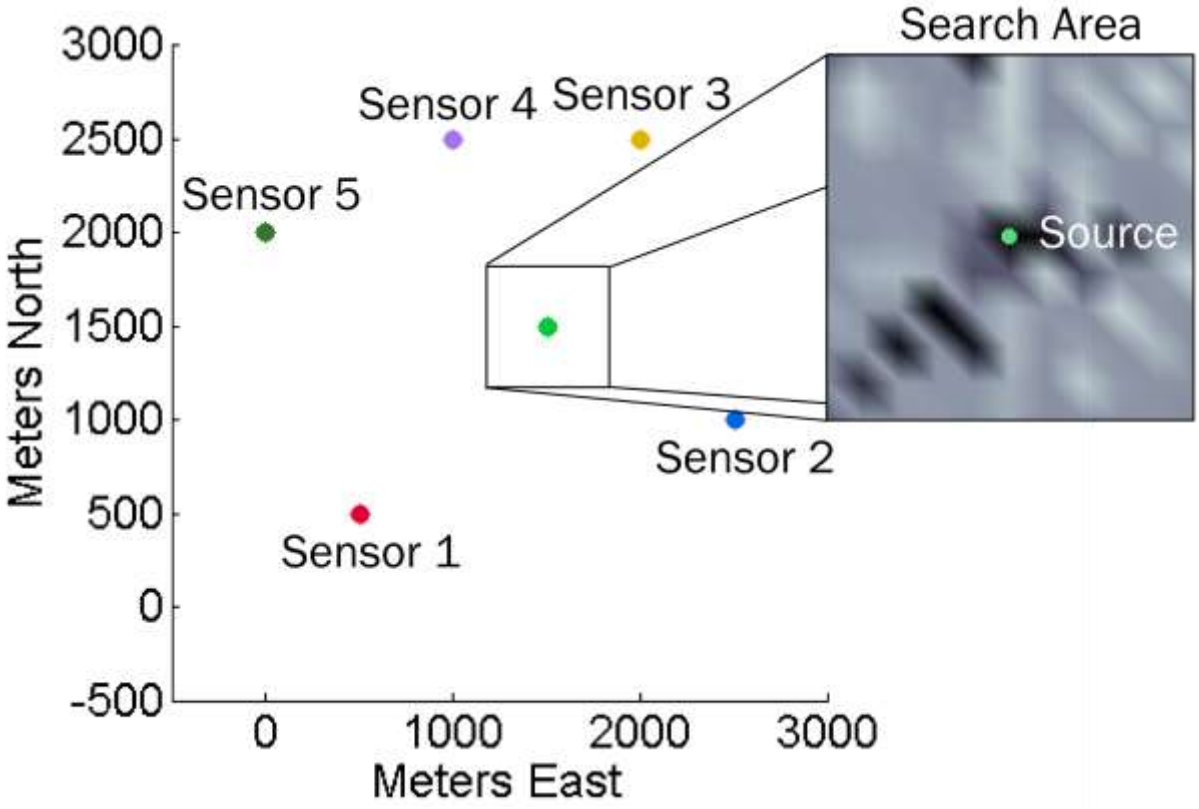


Figure 94: Simulation methodology, showing locations of each sensor surrounding the source. The search area is shown in the inset and is a 51 x 51 grid with 4-meter spacing, with an example interpolated ambiguity surface.

The ambiguity surface was evaluated on an equally-spaced grid centered on the true source location containing 2601 points (51 x 51), with 4 meters horizontally and vertically between points, for a total search area 204 x 204 meters large. All simulated acoustic data was preprocessed by filtering/whitening, clipping, then filtering again (see Section 2.3.4) in order to match the steps taken in experimental analyses. The pressure and velocity SNR (SNR_p, SNR_v) as well as the standard deviation of positional error σ were varied in each trial. A sample set of ambiguity surfaces (generated for a larger set of points for demonstration purposes) is shown for each beamforming method in Figure 95.

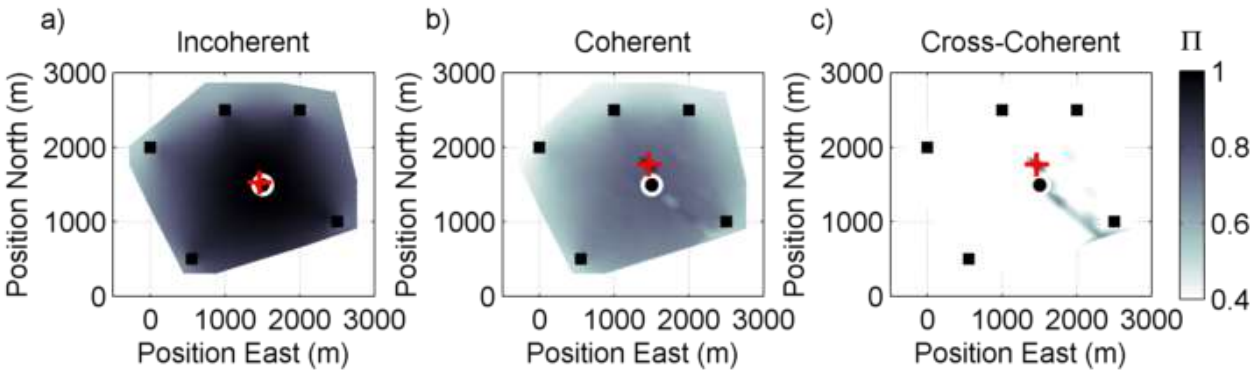


Figure 95: Sample ambiguity surfaces for simulated localization using positional standard deviation of 10 meters and 15 dB SNR. The surface is normalized to unit amplitude, and the estimated source location is displayed as a red cross. The true source location is a black circle with white border and the sensor locations are black squares.

The sample ambiguity surfaces were generated using $\sigma = 10$ meters and $SNR_p = SNR_v = 15$ dB. Beamformer output powers were normalized to the range 0 to 1 for each ambiguity surface. Note that these results are only for a single statistical realization and do not reflect the overall performance of each method. However, some properties of each method become apparent, such as the relative imprecision the incoherent method displays. This indicates the presence of noise is likely to be able to vastly skew the estimated source location. Furthermore, note the high precision of the source estimate in the cross-coherent case, although its estimate is inaccurate. This is due to the positional error which was added to the sensor positions. Figure 95 gives a sense of what is to be expected from the simulated results: the optimum method will depend on the amount of acoustic and positional noise present.

The simulation results presented in Figure 96 and Figure 97 reflect the average performance of 50 statistical trials, each with different random number generator seeds. This ensures the signal, noise and positional noise vectors are different for each trial.

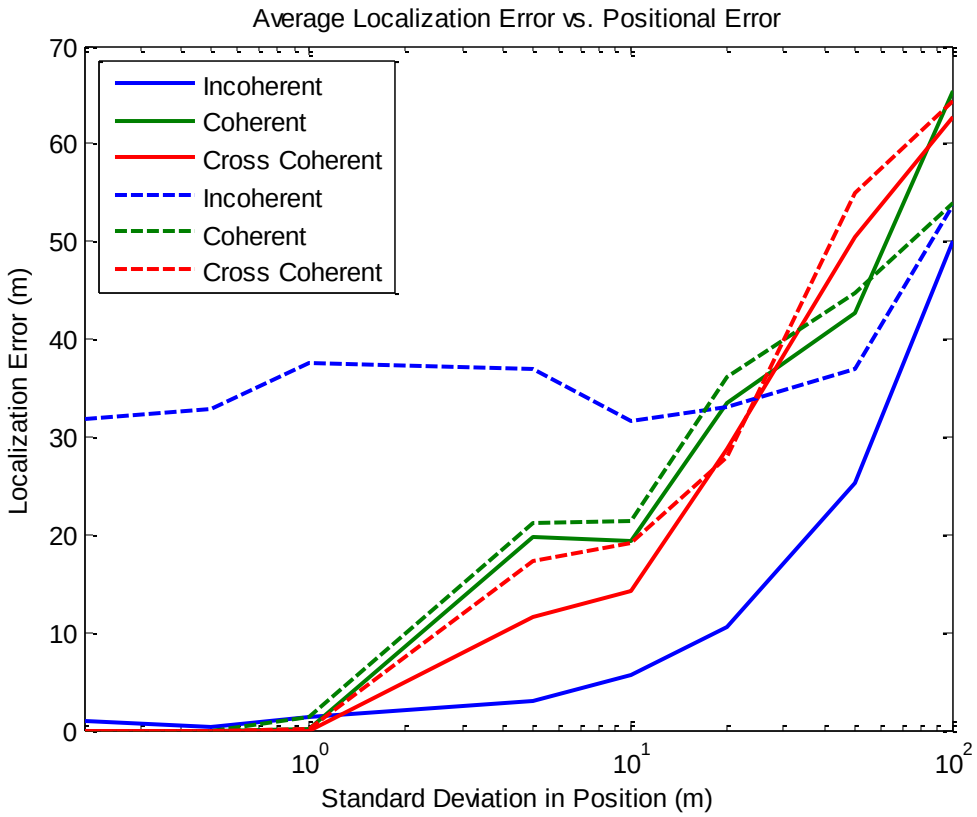


Figure 96: Localization error for high SNR (15 dB pressure, 15 dB velocity) in solid lines and low SNR (-5 dB pressure, -10 dB velocity) in dashed lines. The incoherent weighting is shown in blue, the coherent in green, and the cross coherent in red.

The solid lines in Figure 96 show the high SNR results, with $SNR_p = SNR_v = 15$ dB. The dashed lines correspond to a low SNR case, with $SNR_p = -5$ dB and $SNR_v = -10$ dB. The high SNR case shows that the incoherent method outperforms the coherent methods when the positional errors increase beyond a meter or so. However, for negligible positional error (< 1 m), the coherent methods have no error. In the case of very low SNR, the incoherent method no longer outperforms the coherent methods. Both coherent methods have very similar performance for low SNR, only slightly worsened with the increased noise. Also worth noting is that the cross-coherent method outperforms the coherent method slightly for modest positional errors (< 20 m) and both noise cases.

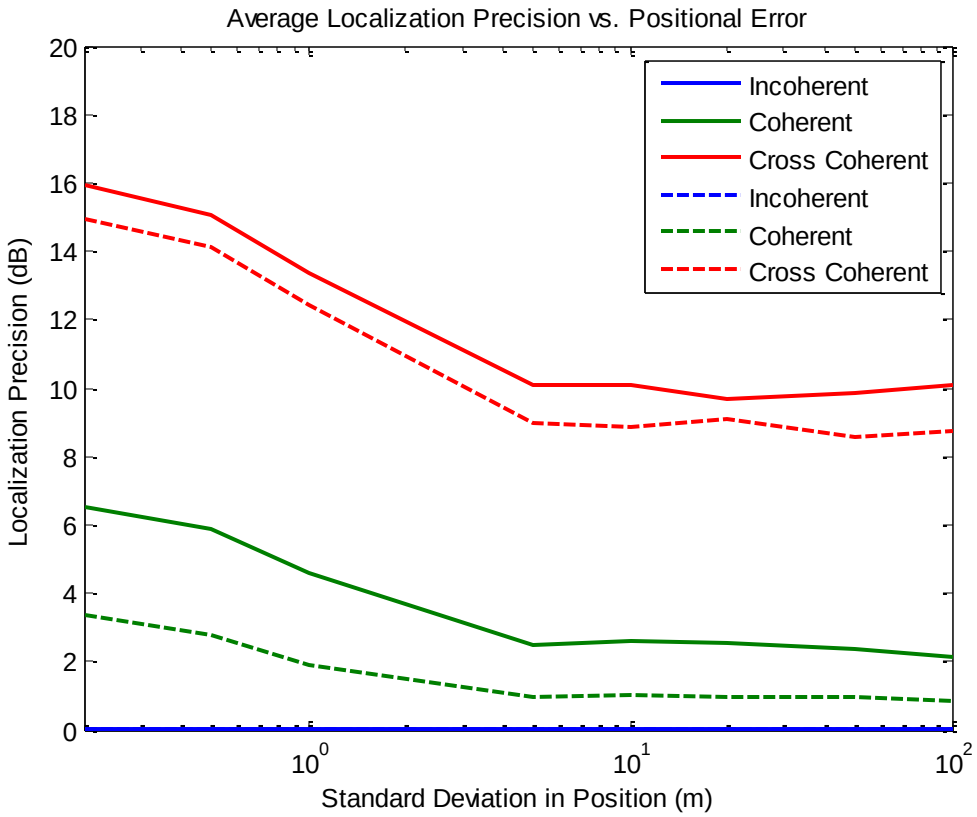


Figure 97: Localization precision for high SNR (15 dB pressure, 15 dB velocity) in solid lines and low SNR (-5 dB pressure, -10 dB velocity) in dashed lines. The incoherent weighting is shown in blue, the coherent in green, and the cross coherent in red.

Examining the localization precision, plotted in Figure 97, the cross-coherent method performs the best for both high and low SNR, and across all positional errors. The coherent method performs second-best, but shows a more noticeable decrease in precision for low SNR (dashed line) than the cross-coherent, which seems more robust to decreased SNR. Lastly, the incoherent method has very low precision (almost 0 dB), indicating that the ambiguity surface is approximately the same value as the maximum value. This property is likely why the error increases so drastically when noise is added; any slight increase in the ambiguity surface at the wrong location could easily cause a new maximum to be found there.

5.3.2 Experimental Methodology and Results

To validate the proposed beamforming framework and weighting methods from Section 5.2, experimental data collected from the October 30 dataset were analyzed. The source tracked is the research vessel (R/V), whose location is known from GPS measurements taken during the experiment. Since the vessel is moving, the experimental data is segmented into windows of length T = 5 seconds (i.e. snapshot duration). The maximum vessel speed observed was 13.5 kts (~7 m/s), which means the vessel travels no more than 35 meters in a given window. The segment length was chosen to be as short as possible, but long enough that the beamforming process is robust to noise. Shorter segments often resulted in erratic localization behavior, and longer segments usually were acceptable, but could degrade localization performance if the source moves significantly over a period.

Acoustic data from three periods, hereafter referred to as P1, P2, and P3, were analyzed. Each period consisted of the research vessel either approaching or retreating from the array. Each period also contained a section of data where the vessel performed maneuvers around the array, either circling it, passing through it, or a combination of both. The acoustic data were preprocessed using a filtering step, followed by integration and pressure unit conversion, then finally another filtering step (see Section 2.3.4). The filtering was performed in the widest band possible of 200 – 1500 Hz to avoid low frequency acceleration noise and integration noise, as well as the higher frequency accelerometer resonant peak. A sample of the source power spectral density (PSD), filtered to the stated band, is plotted in Figure 98.

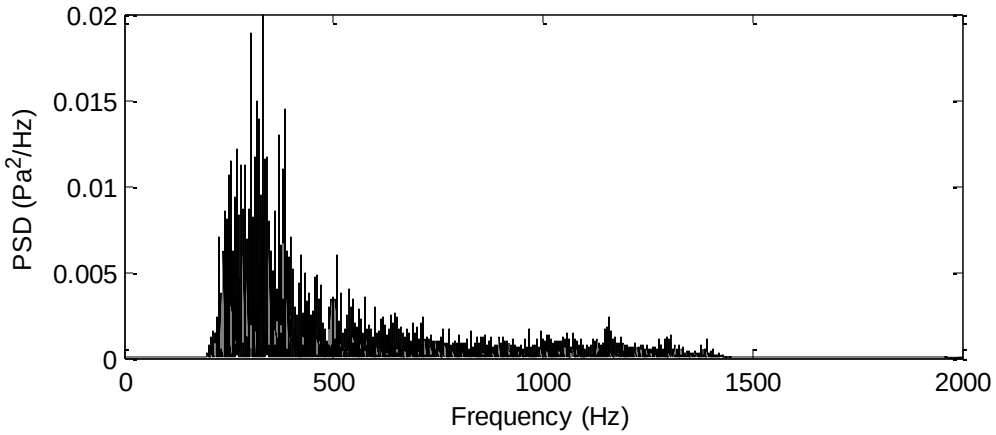


Figure 98: Power spectral density of source at 11:33:00-11:33:10

Clipping was unnecessary as a preprocessing step, as the source signal did not contain significant peaks. Whitening was avoided since it tends to skew the relative amplitudes of the velocity channels, causing incoherent methods which rely solely on this information for localization to perform poorly.

Period 1 lasted from 11:33:00 to 11:38:10 while the R/V made maneuvers around the array and then retreated (see Figure 99).

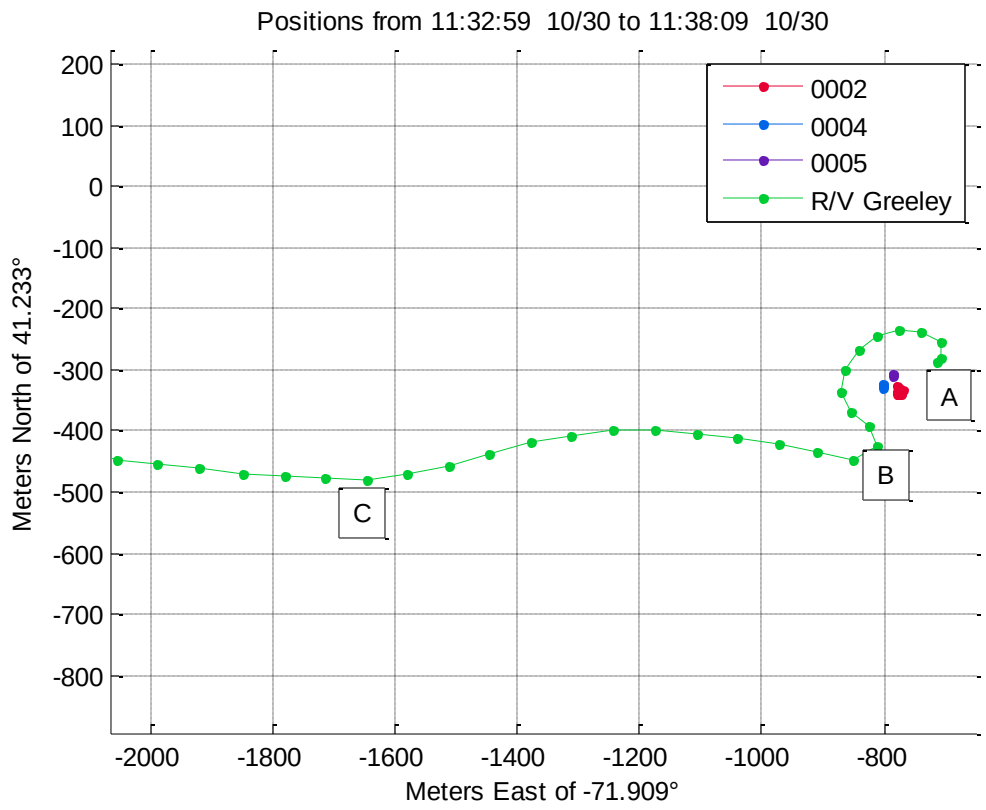


Figure 99: Period 1 sensor locations and source (R/V Greeley) locations.

The sensors with synchronized data (0002, 0004, 0005) were in a triangle formation, not drifting much throughout the period (see Figure 100). It was during this time the tides were changing direction from West to East.

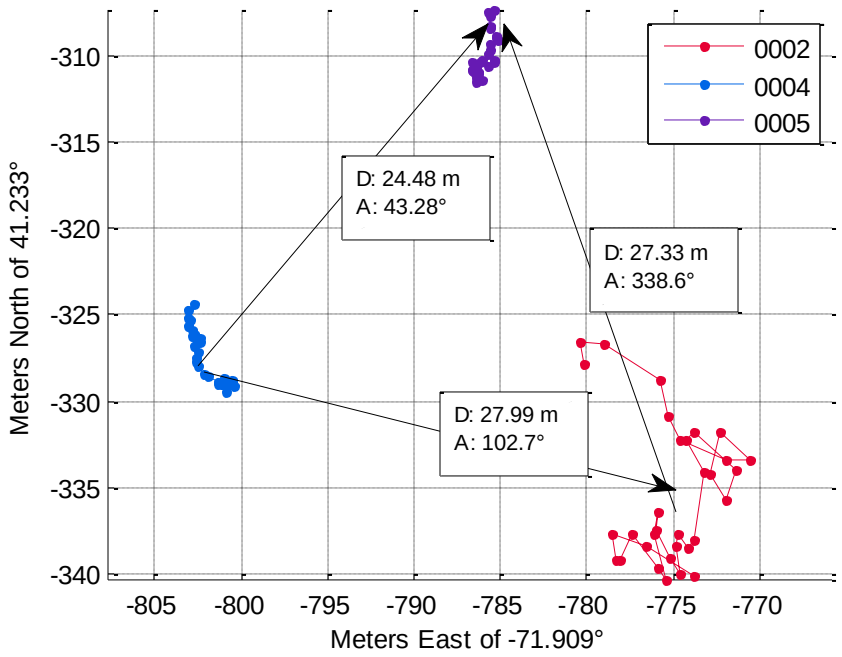


Figure 100: Period 1 sensor positions with sample separation distances. Note sensor 0002's GPS antenna malfunctioned, causing erratic measurements.

Note that sensor 0002 had a malfunctioning GPS antenna, and thus its position appears to be erratic. This is not actually the case, but will unfortunately add additional sources of error during localization. A correlation ensemble (see Section 4.2.1) between pressure channels on sensors 0004 and 0005 shows a clear coherent peak due to the source (see Figure 101). Each correlation is normalized to its own maximum, and the magnitude is obtained using the magnitude of the Hilbert transform. The difference in distance from the source to each sensor divided by the sound speed (1500 m/s) is overlaid with a black line, and represents the estimated location of the coherent peak which should result. The bottom horizontal color bar indicates the relative maximum of each correlation with respect to the total maximum. As time goes on, the correlation maxima become smaller due to the boat receding from the array and getting quieter.

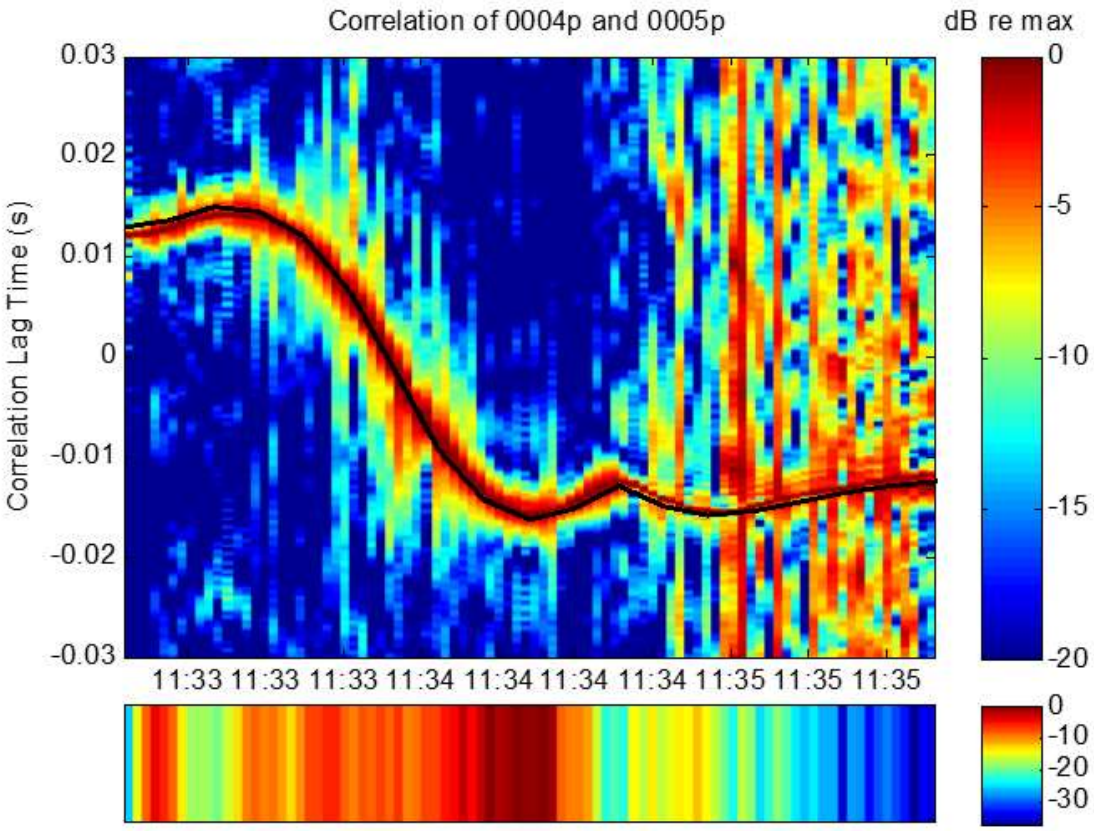


Figure 101: Correlation ensemble for 5 second intervals over the course of Period 1. Each correlation in the ensemble is normalized to its own maximum. These maxima are plotted relative to the overall maximum in the bottom panel. The black line denotes the estimated correlation delay given the source and sensor positions and a sound speed of 1500 m/s.

A grid of test points for the beamforming algorithm to evaluate (see Section 5.2.1) was chosen to encompass the possible source locations (see Figure 102). It attempts to strike a balance between being too broad and taking too much time to evaluate and being too narrow such that the localization has no choice but to "succeed".

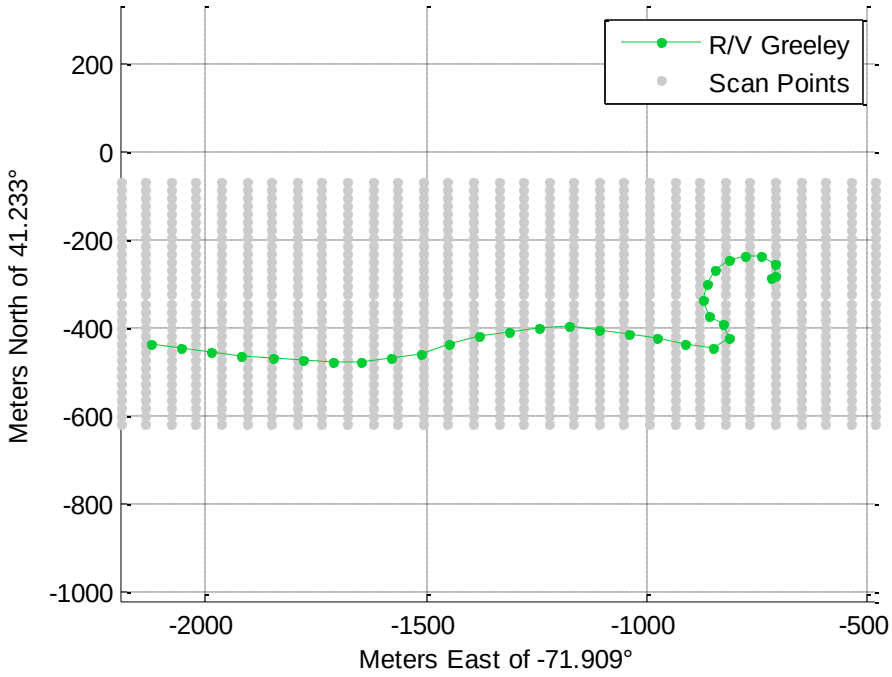


Figure 102: Ambiguity surface grid points for Period 1.

A sample normalized ambiguity surface for each beamforming method at 11:33:10 is plotted using a color scale in which black denotes the maximum (i.e. unity) and white indicates the minimum. The actual source and sensor locations are plotted as well as the estimated source location obtained from the maximum of the ambiguity surface (see Figure 103, Figure 104, Figure 105).

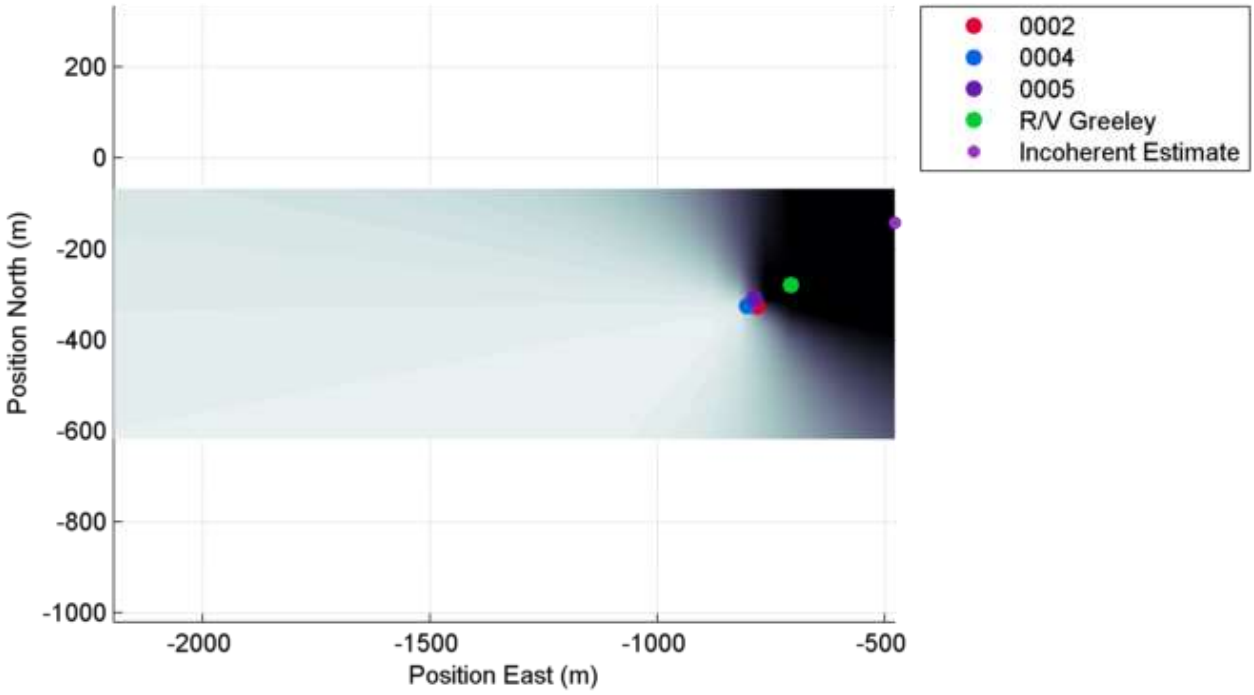


Figure 103: Sample incoherent ambiguity surface for one 5-second segment. The true source is shown as a green dot (R/V Greeley) and the estimated source as a purple dot.

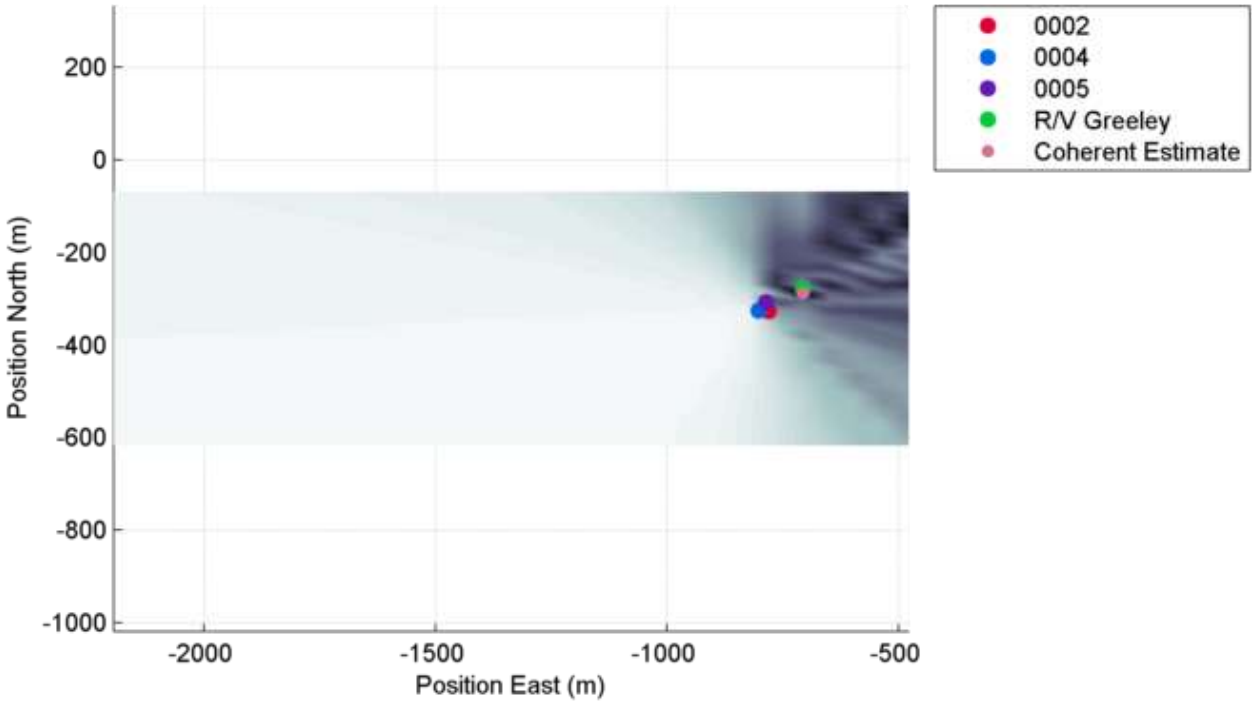


Figure 104: Sample coherent ambiguity surface for one 5-second segment. The true source is shown as a green dot (R/V Greeley) and the estimated source as a pink dot.

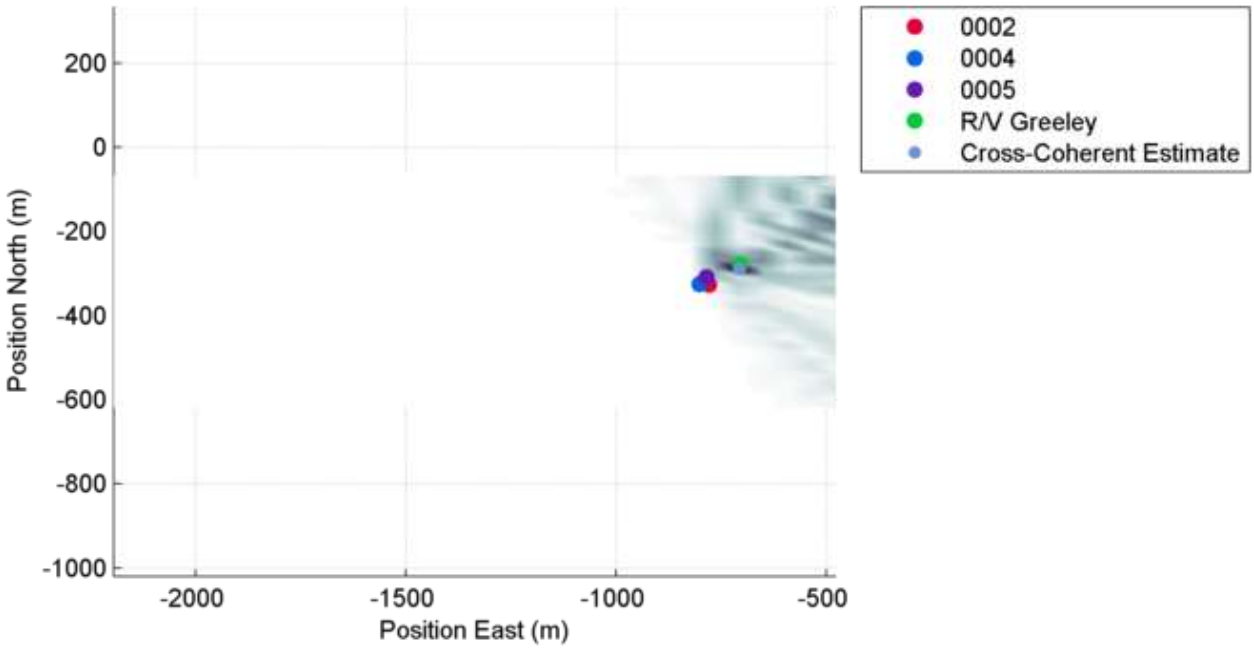


Figure 105: Sample cross coherent ambiguity surface for one 5-second segment. The true source is shown as a green dot (R/V Greeley) and the estimated source as a sky blue dot.

These results are similar to the simulation results seen in Section 5.3.1: the incoherent is accurate but not precise, the cross-coherent is precise and accurate, and the coherent falls somewhere in between. The accuracy of the incoherent method indicates the SNR of the source is high, which makes sense given its proximity (~100 meters) to the sensors. Another aspect of these results shows that the range-resolution of the array is limited, and will especially be strained when the source is much further from the array. For this reason, only the bearing to the source will be considered in the following results, and the error metric is modified to be

$$\epsilon_b = \theta_s - \hat{\theta}_s \tag{118}$$

where θ_s is the bearing from the array centroid to the source and $\hat{\theta}_s$ is the bearing to the estimated source location. The precision metric is identical to that in Equation 107,

however it is worth noting that the metric is not search-space invariant (i.e. the precision changes as the search space changes). Figure 106a shows the bearing error for 5-second snapshots over the duration of P1, for each of the three weighting methods. Figure 106b shows the localization precision for each snapshot.

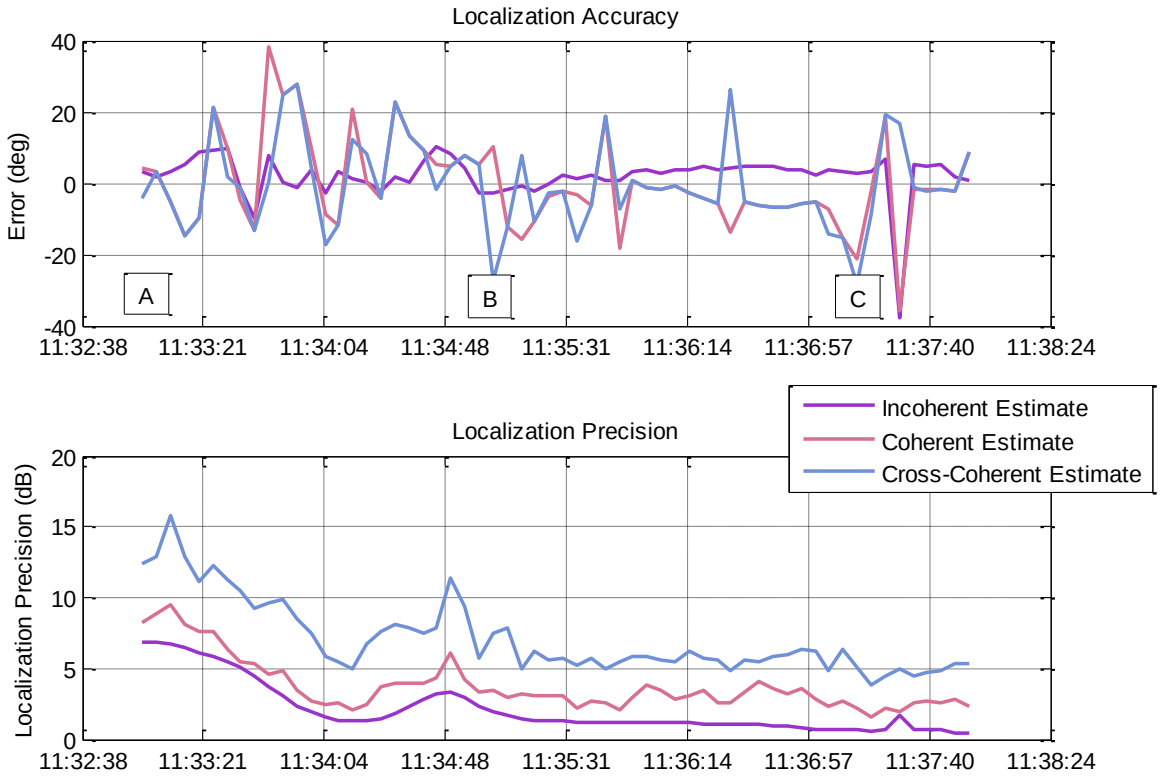


Figure 106: Localization accuracy and precision over Period 1 for the incoherent (purple), coherent (pink) and cross-coherent (blue) methods. The letter markers (A, B, C) match the times shown in Figure 99.

The results show that the incoherent method is relatively accurate, and consistently so over the entire period. The coherent methods do not fare as well; on average they are accurately locating the source, but occasionally display a large error for a single snapshot. This could be easily avoided using a source tracking algorithm implemented with Kalman filters or the like. It is most likely the result of sensor 0002's malfunctioning GPS, which effectively

puts the positional error much higher than it would be for accurate GPS (~ 10 m instead of 1.5 m). As seen in the simulation results (see Section 5.3.1), the coherent methods have higher error than the incoherent method for high SNR cases. This is experimentally verified since the vessel was close by for the entire period, thus having relatively large SNR (see correlation ensemble in Figure 101).

Furthermore, the precision of the localization is somewhat similar to simulation results. The cross-coherent method obtains the highest precision which decreases as the source moves further away, decreasing SNR. The coherent method also shows a similar effect, but with lower overall precision. The only difference between the experimental results and simulation is the higher precision of the incoherent method, which is due to the difference in the search space. In simulation, the search space was very localized to around the stationary source. In this case, the search space encompasses many points not near the source. This ensures there are points in the space which will be in a direction different to the true source direction, thus having smaller output even for the incoherent beamformer. Ideally, the precision metric should be invariant of the search space, or be defined for a search space which is nearly infinite. However, computational limitations dictated the search space be as small as possible.

The second period, P2, lasted from 13:30:30 to 13:51:10 local time of October 30th. The R/V was returning to the array from a standoff distance of ~4.5 km (see Figure 107) to reposition the sensors, as they had drifted far from each other. Before approaching any sensors, the vessel performed a maneuver through the array, this time sharper and more sudden than that of P1.

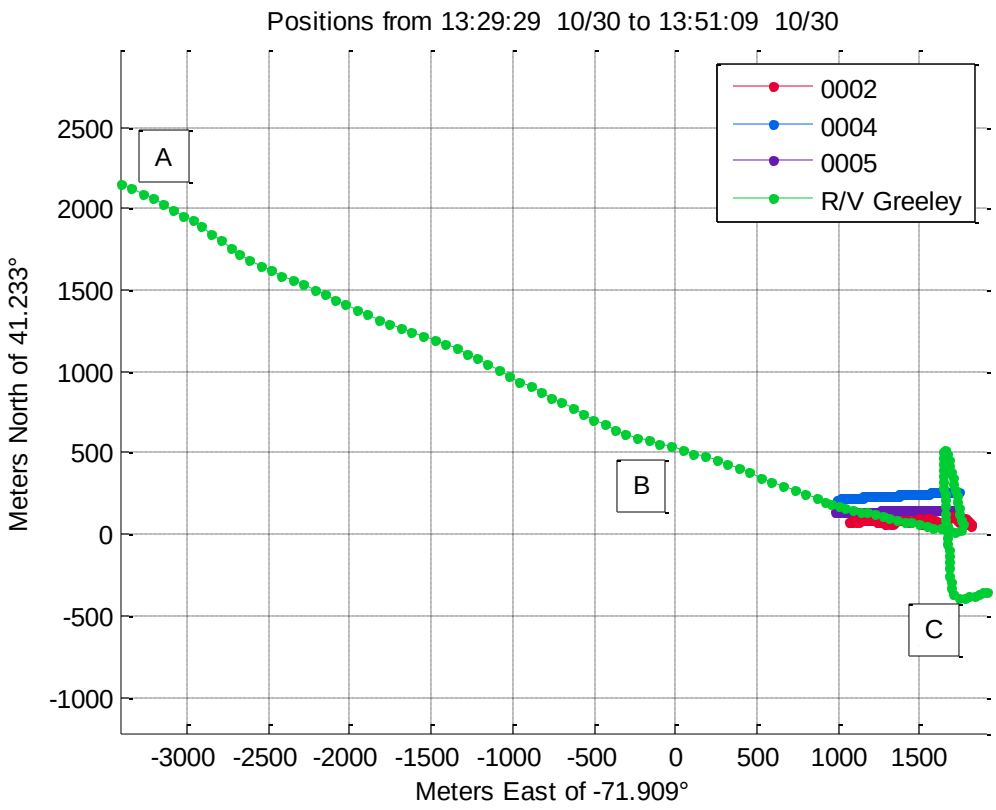


Figure 107: Period 2 sensor locations and source (R/V Greeley) locations.

The sensors had drifted from West to East over this period, diverging from their initial close separations to over 80 meters apart (see Figure 108). The array also became more linear from North to South, rather than their triangular configuration seen during P1. The aperture of this array was approximately 150 meters, increasing to almost 200 meters by the end of the period. Again, note that sensor 0002 displayed poor GPS locations due to its antenna malfunction.

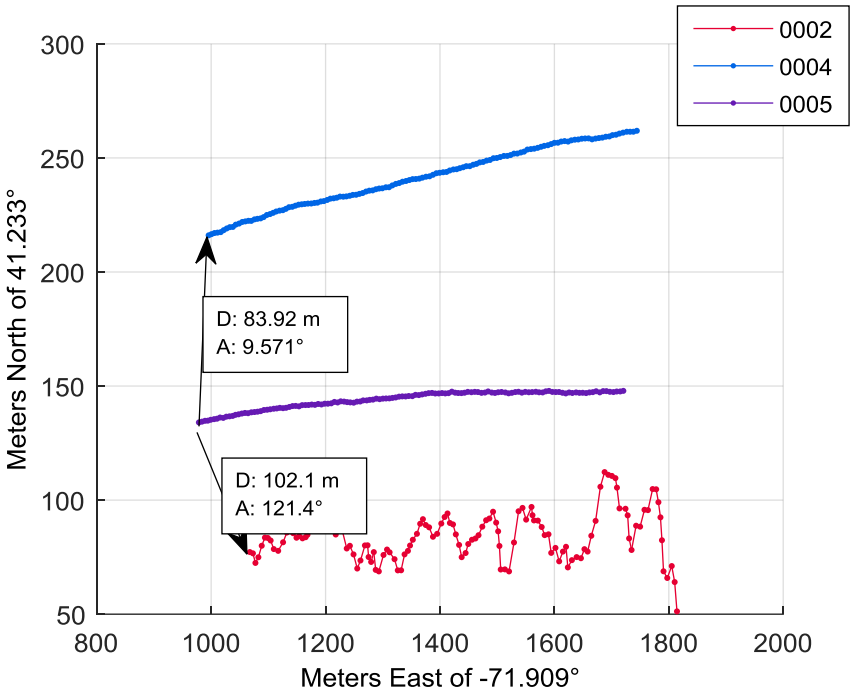


Figure 108: Period 2 sensor positions with sample separation distances. Note sensor 0002's GPS antenna malfunctioned, causing erratic measurements.

The search space required for this period is much larger than that used in P1, since there was a larger number of possible source locations across the entire period. This could be remedied in the future if the search space is limited to where the source is believed to be, given its history (e.g. using Kalman filtering). In any case, as a proof of concept, the search space is shown in Figure 109, encompassing a large area around the source and sensors.

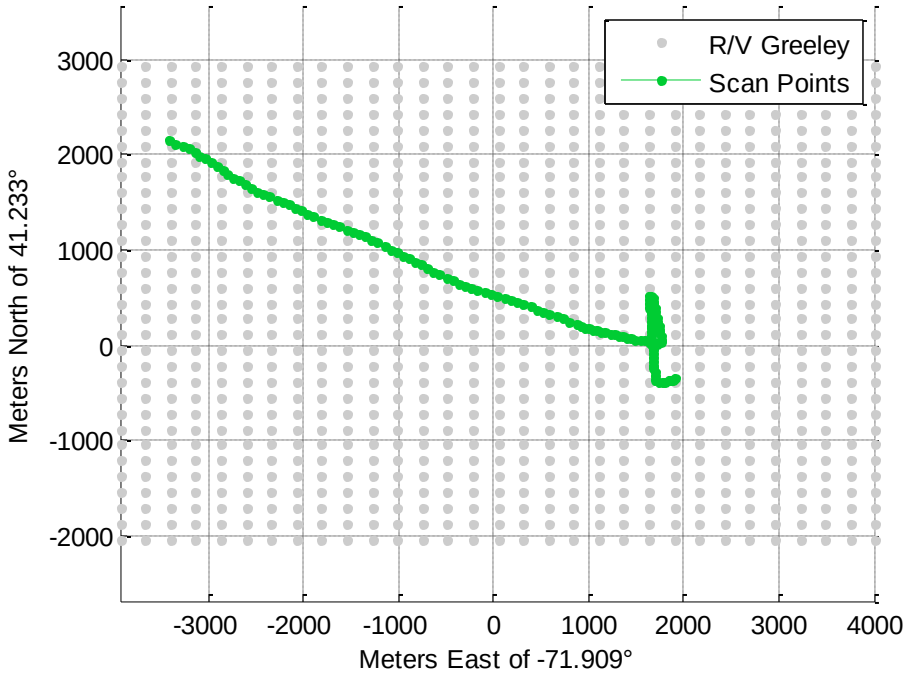


Figure 109: Ambiguity surface grid points for Period 2.

An example ambiguity surface is shown for each method at 13:38, denoted by marker B in Figure 107. The incoherent surface is plotted in Figure 110, the coherent in Figure 111, and the cross-coherent in Figure 112.

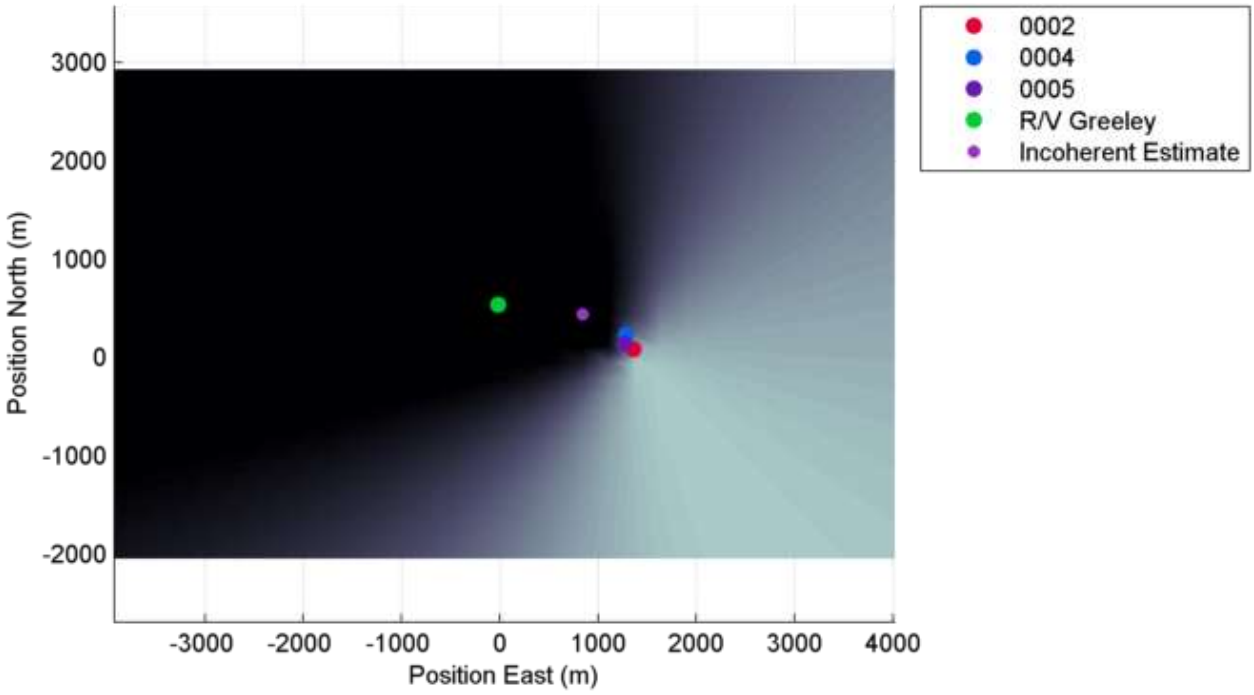


Figure 110: Sample incoherent ambiguity surface for one 5-second segment. The true source is shown as a green dot (R/V Greeley) and the estimated source as a purple dot.

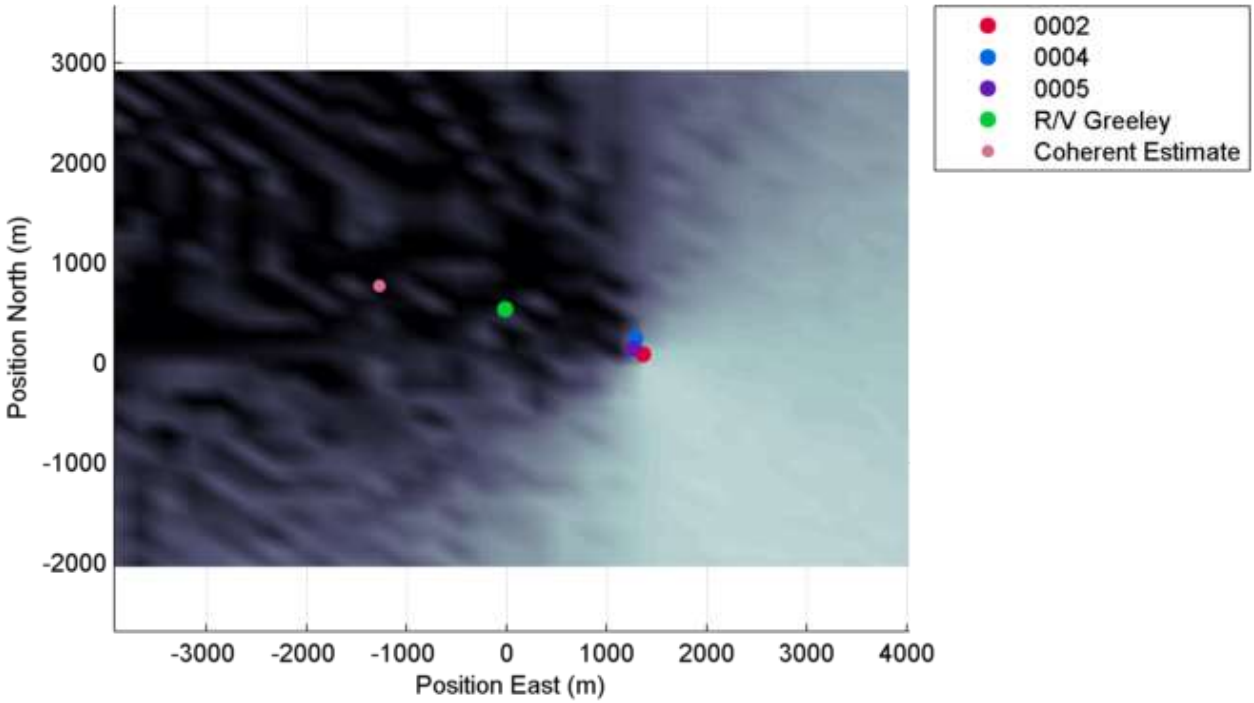


Figure 111: Sample coherent ambiguity surface for one 5-second segment. The true source is shown as a green dot (R/V Greeley) and the estimated source as a pink dot.

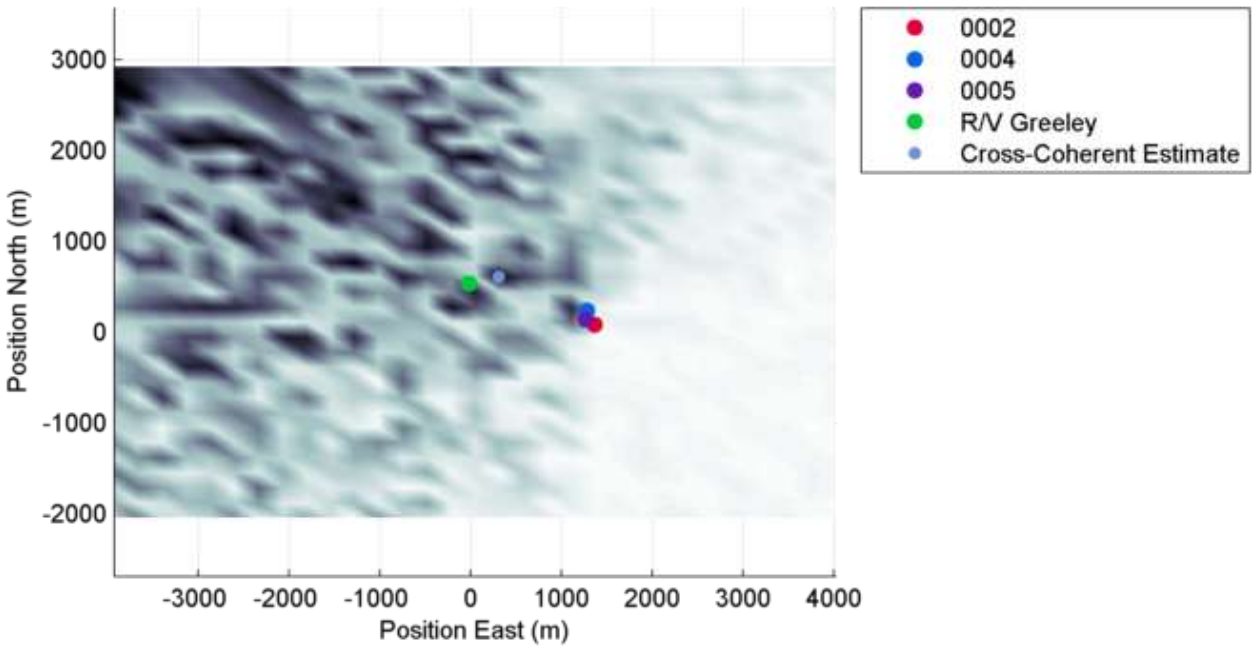


Figure 112: Sample cross-coherent ambiguity surface for one 5-second segment. The true source is shown as a green dot (R/V Greeley) and the estimated source as a sky blue dot.

These ambiguity surfaces are evocative of the simulation results again, in that the incoherent displays a wide beam (i.e. dark area) in the direction of the source. The cross-coherent method shows increased noise as the background "clutter" in the ambiguity surface. It is also apparent that the cross-coherent method still leverages the sensor directionality, as the clutter noise amplitude is larger in the direction of the source (i.e. in the same areas the incoherent method shows high output). The coherent method is again somewhere between the incoherent surface and cross-coherent surface, as it is simply the combination of the two. Also, worth noting is the slight bearing error of the coherent methods compared to a more severe error seen in the incoherent case. This effect corroborates the simulation findings where the incoherent method fails for low SNR, even when the coherent methods still function properly.

As was the case in P1, the range resolution of the array is limited due to the closely-spaced sensors. Again, the localization accuracy will be displayed using the bearing error (Equation 118). A future experiment which is not opportunistic in nature should attempt to position the sensors further from each other to test absolute error. Figure 113a shows the localization accuracy over the 5-second snapshots of P2, and Figure 113b shows the localization precision for each method.

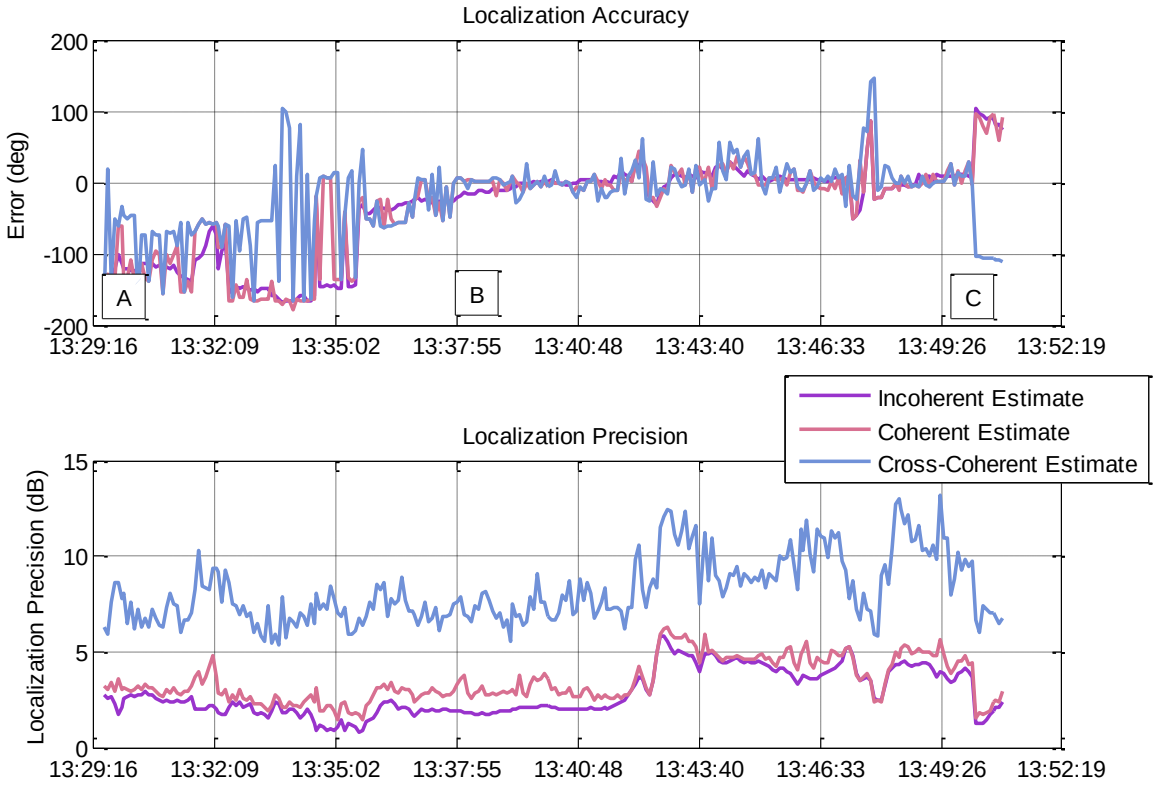


Figure 113: Localization accuracy and precision over Period 2 for the incoherent (purple), coherent (pink) and cross-coherent (blue) methods. The letter markers (A, B, C) match the times shown in Figure 107.

The results shown in Figure 113 show a clear period between marker A and B where the localization fails. This is simply due to the source being too far from the array for it to

be detected (as was intended when the 4.5 km standoff distance was selected). At point B, and possibly even before, the source is being properly localized. One might even argue that the coherent methods were detecting the source before point B, but with a small positive bearing error (which could be simply due to the GPS errors present in sensor 0002). From point B to point C, the results are very similar to P1, where all methods locate the source on average. The coherent methods show more spurious peaks, and all methods fail when the source is too close (i.e. source driving through the array) and the free-space, far-field assumptions of the beamformer are no longer upheld as multipath propagation effects start to become more significant when the source moves closer to the array in this shallow water environment.

The localization precision of each method follows a similar trend to P1. The cross-coherent method again has the highest precision, followed by the coherent and closely behind by the incoherent. The precision increases during this period however, as the source is moving closer to the array, thus increasing the SNR.

Finally, the third period, P3, lasted from 14:27:30 to 14:45:10 local time. The R/V had just finished repositioning the array elements and performed a simple circular maneuver (marker A) around the array, then proceeded to the 4.5 km standoff distance (marker C, see Figure 114).

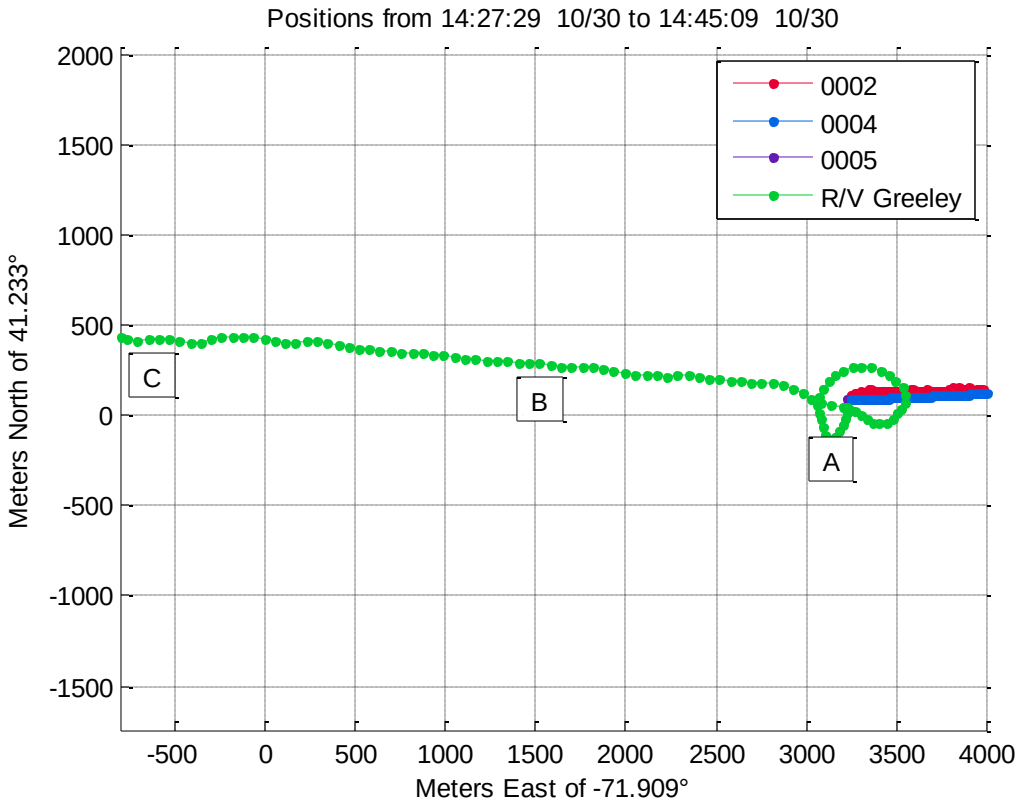


Figure 114: Period 3 sensor locations and source (R/V Greeley) locations.

During P3, the array remained somewhat linear while drifting from West to East (see Figure 115). The apparent separation distance is small when all locations are plotted simultaneously as in Figure 115, however the separation distance mainly is due to differing easterly positions, and varied from 30 meters to 60 meters. Sensor 0002's GPS malfunction appears to be slightly better-behaved during this period, which may help the coherent methods' accuracy.

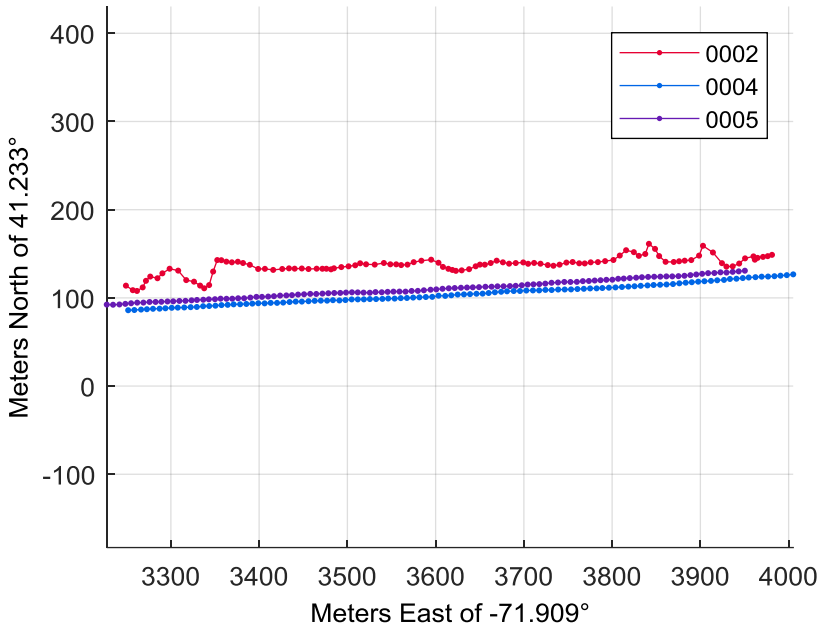


Figure 115: Period 3 sensor positions. Note sensor 0002's GPS antenna malfunctioned, causing erratic measurements.

The search space for P3 is shown in Figure 116, and was chosen to encompass the source and sensors with enough added area to ensure the success of localization was not due to a restrictively small search space.

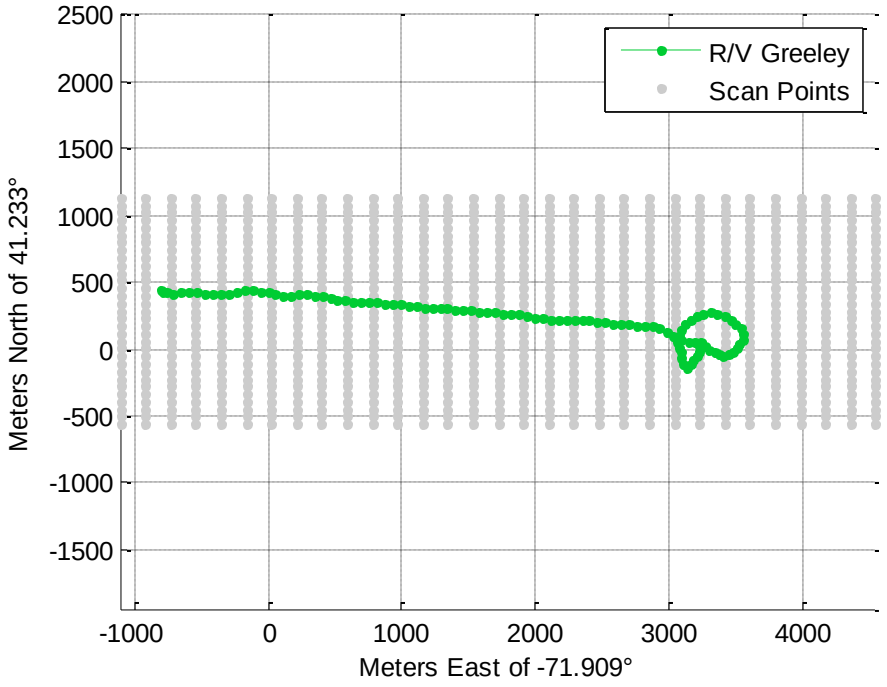


Figure 116: Ambiguity surface grid points for Period 3.

Ambiguity surfaces for each method are shown for 14:38:20 (marker B in Figure 114). The incoherent surface is shown in Figure 117, the coherent in Figure 118, and the cross-coherent in Figure 119.

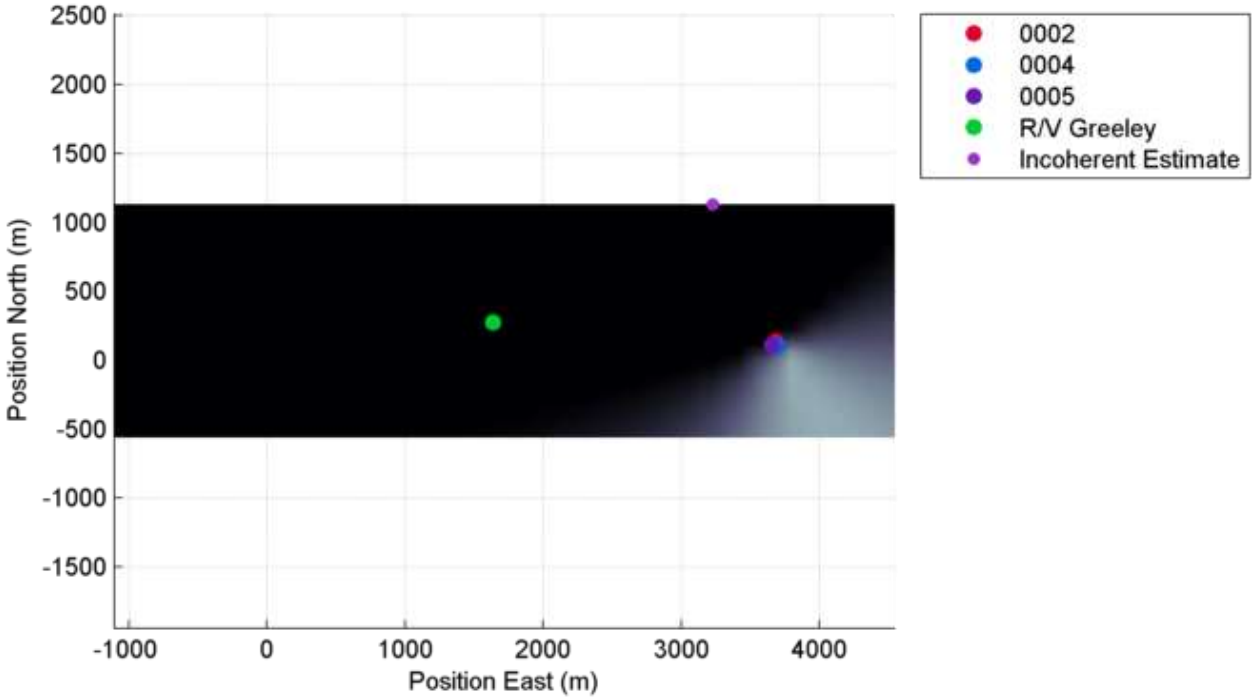


Figure 117: Sample incoherent ambiguity surface for one 5-second segment. The true source is shown as a green dot (R/V Greeley) and the estimated source as a purple dot.

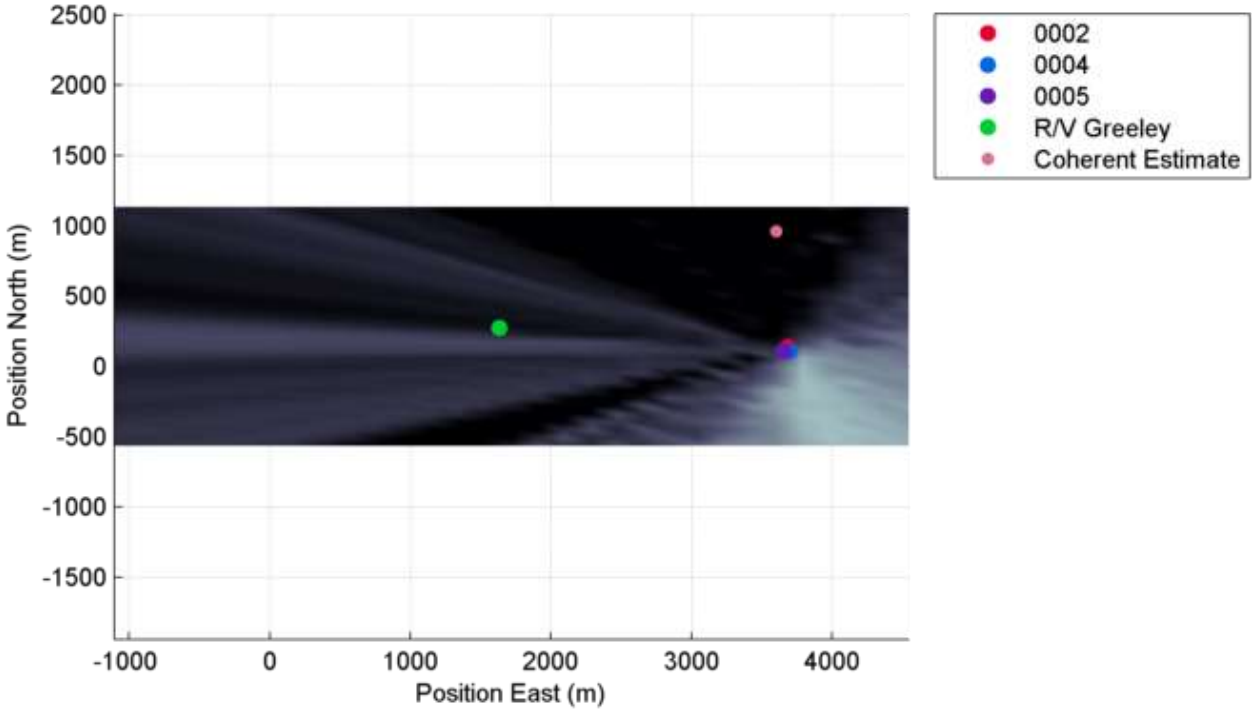


Figure 118: Sample coherent ambiguity surface for one 5-second segment. The true source is shown as a green dot (R/V Greeley) and the estimated source as a pink dot.

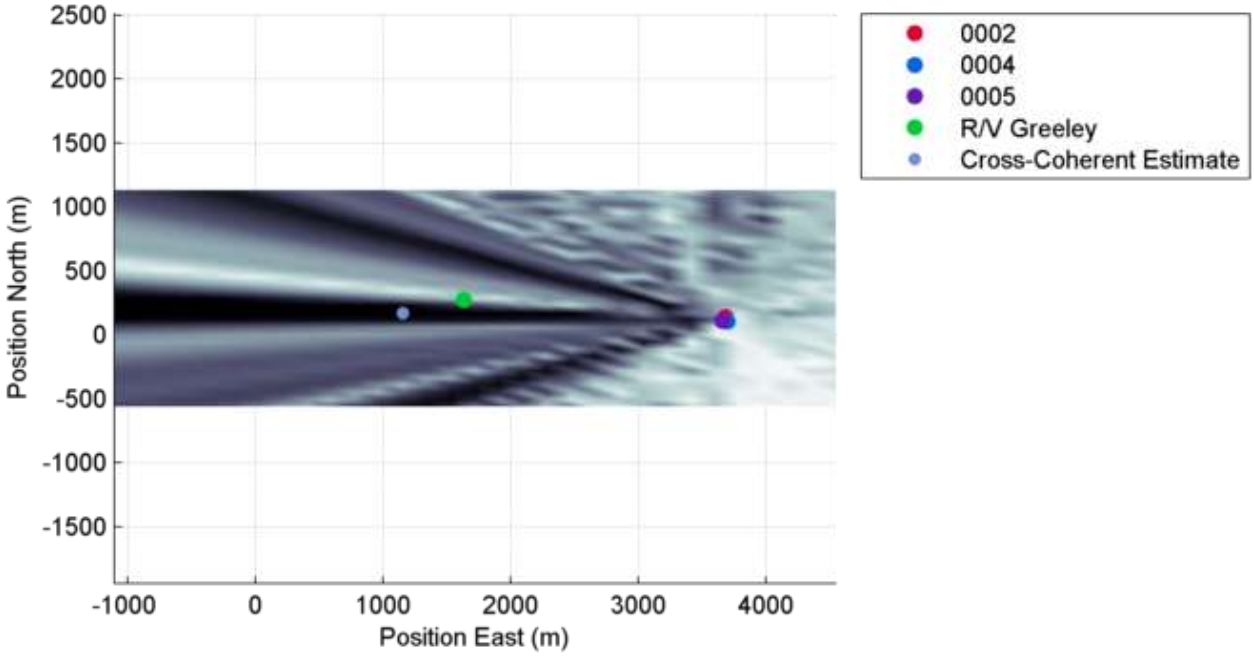


Figure 119: Sample cross-coherent ambiguity surface for one 5-second segment. The true source is shown as a green dot (R/V Greeley) and the estimated source as a sky blue dot.

The sample surfaces demonstrate a case where the cross-coherent method maintains an accurate estimate when the incoherent and coherent do not. This could be the result of an interfering source to the North which has biased the incoherent beam quite far from the R/V (see Figure 117). The coherent method also shares this bias, as its cross-coherent component is not large enough to overrule the incoherent component. In other words, the coherent method shares a dark sidelobe in the direction of the source with the cross-coherent method, however the incoherent main lobe is larger (see Figure 118). The cross-coherent method eschews the incoherent information, which allows this coherent lobe to accurately locate the source. Furthermore, note that the cross-coherent surface displays three lobes. These are the result of intersecting hyperbolae which correspond to the correlation time delay of the source. Each pair of sensors produces a hyperbola for the

source's differential arrival time. The hyperbolae are scaled by the sensor directionality, and their intersection should align with the true source location. This indicates the cross-coherent method acts as a method which solves for the source location from time-difference-of-arrival (TDOA) information using intersecting hyperbolae. This makes sense, since the fundamental source of information in the cross-coherent method is the cross-correlation between sensors, yielding peaks at the TDOA times.

Figure 120a plots the localization accuracy over P3 for each 5-second segment. Figure 120b plots the localization precision for each method.

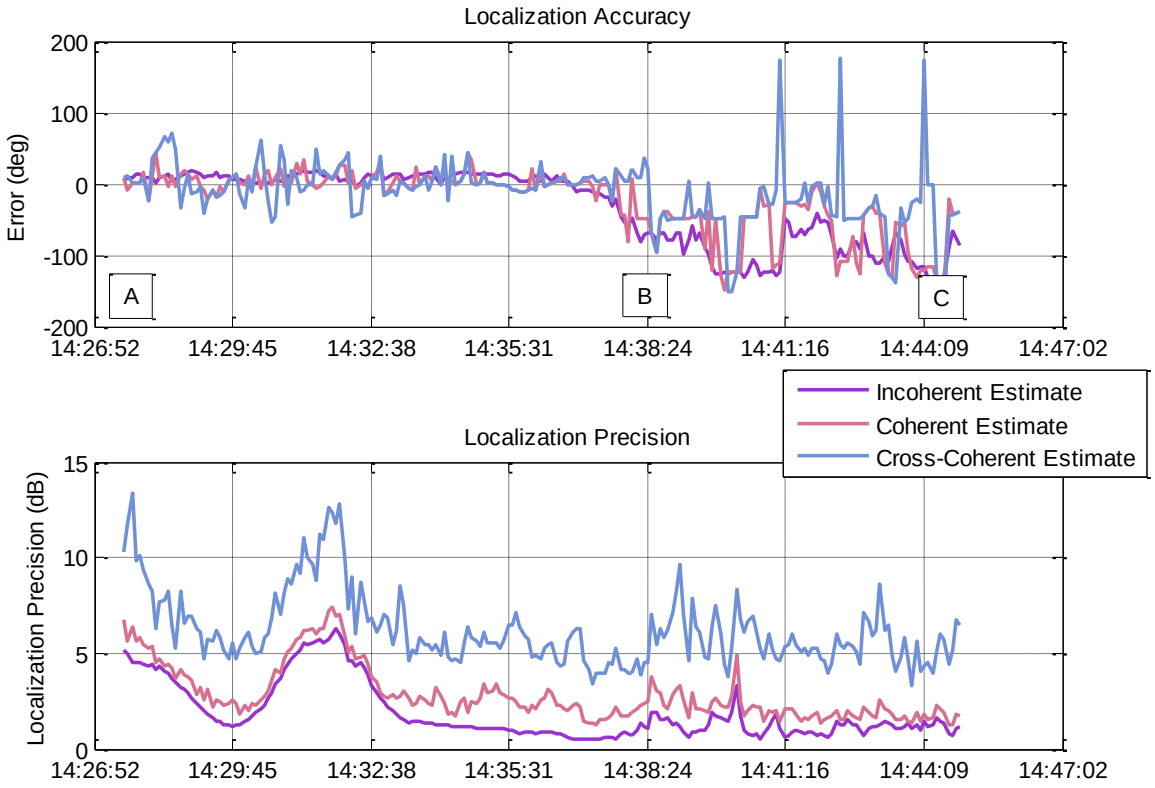


Figure 120: Localization accuracy and precision over Period 3 for the incoherent (purple), coherent (pink) and cross-coherent (blue) methods. The letter markers (A, B, C) match the times shown in Figure 114.

Similar results to P1, P2 and simulation are obtained for P3. The localization error is erratic for the coherent methods, and more stable for the incoherent method from marker A to B. Just before marker B, the coherent and incoherent methods begin to fail, and a short time later, the cross-coherent method fails as well. At this point, the source has moved too far from the array to be reliably located. A few snapshots between B and C show small error for the coherent methods, which might simply be random luck if the localization is so erratic. Again, a tracking algorithm could improve the bearing error, as well as improved GPS accuracy (e.g. fixing sensor 0002), or spreading the sensors further apart as done in simulation.

The localization precision is highest for the cross-coherent method, and varies as the source moves further from the array. It is unclear why the precision peaks again after marker A, but this could be the result of more complicated bathymetry present in the area of P3. As seen in the previous periods, the incoherent has the smallest precision, closely followed by the coherent method. The large search space relative to simulation results allows the incoherent precision to nearly match that of the coherent. This near-matching also corroborates the finding displayed in Figure 118, where the coherent ambiguity surface appears to be more heavily favoring the incoherent components. The result is that the ambiguity surface is not drastically different from the incoherent, thus its precision is almost identical.

5.3.3 Conclusion

Source localization using weighted correlation matrices was tested on simulated and experimental vector sensor data. Incoherent processing was shown to be accurate in high SNR cases, and robust to sensor position errors. However, its accuracy suffers greatly

when more noise is present. Furthermore, its precision is usually quite low. The coherent methods display higher precision, but at the cost of accuracy if the positional noise is high. They also perform better under low SNR conditions, with the cross-coherent method retaining high precision and accuracy, as long as positional errors are well-managed. Further improvements could be made to include adaptive weighting of the correlation matrix, or adding tracking algorithms to improve sequential localization performance.

5.4 Localization Performance Using Ambient Noise AEL

5.4.1 Methodology

Sensor location accuracy was shown to be vital to localization performance of coherent methods (see Section 5.3). Leveraging the ambient noise array element localization (AEL) algorithm presented in Chapter 4 to decrease sensor location error should improve localization performance. To test this theory, sensor pair 0002/0005 from October 30th experimental data (see Section 2.3.5.5) was examined between 15:12 and 15:17 local time. The pair contained the sensor with erratic GPS measurements, which provides opportunity for improvement using ambient noise AEL. The GPS error may also reflect a scenario in which dead-reckoning or other low-accuracy method was used to locate the source.

To correct the sensor positions, the difference-constrained genetic algorithm with a stochastic search segment length of 10 was applied to the positive time delays of a 5-second correlation ensemble. The ensemble was obtained from beamformed axial velocity correlations where the vector sensor data was preprocessed using a filter step, integration and pressure unit conversion step, followed by a whitening step (see Section 2.3.4). The filter band limits used were 200 – 1500 Hz. All parameters match those of the results

presented in Section 4.2. The separation distances were obtained by multiplying the genetic algorithm time delays by an assumed sound speed of 1500 m/s. Then, the AEL algorithm was applied using an assumed measurement error σ equal to 0.01 meters and an assumed prior (i.e. GPS) measurement error ξ equal to 1.5 or 10 meters for the valid and invalid GPS sensors, respectively. This effectively ignores the GPS prior in order to fix the distances between sensors to those measured by the genetic algorithm. A plot of the corrected sensor locations is shown in Figure 121. The thin lines denote the GPS measurements, and the thick lines denote the genetic algorithm's corrected locations.

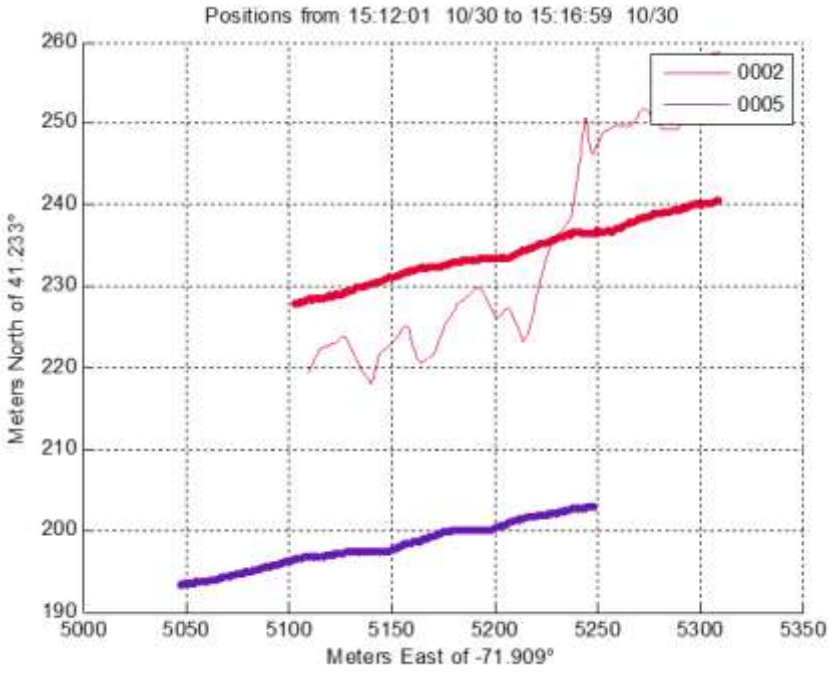


Figure 121: Sensor locations using GPS (thin lines) estimates and positions obtained from regularized AEL obtained from constrained genetic algorithm distance estimates (thick lines). Note the GPS and GA positions for sensor 0005 are nearly identical.

The location of sensor 0005 does not change at all, since its prior error is an order of magnitude smaller than that of 0002. Furthermore, the erratic GPS measurements (thin

line) are effectively smoothed out by the genetic algorithm distance estimation, resulting in feasible corrected locations (thick line).

In addition to comparing the GPS and genetic algorithm sensor locations, a third method is used which assumes the sensor locations are constant across the five-minute period. This simulates a scenario in which the sensors' locations were known until they submerged, at which point no further GPS measurements were obtained. Table 4 outlines the different parameters which were tested in the following experimental results.

Table 4: Experimental Results Parameter Table

Parameter	**Values**	**Comments**
Sensor Pair	0002/0005	Tests whether the GA/AEL method can correct GPS errors
Sensor Locations	Constant, GPS, GA	Tests worst-case scenario, current best-case scenario, and proposed AEL method
Beamforming Methods	Incoherent, Coherent, Cross-Coherent	Tests each of the three weighting methods introduced in Section 5.2.2

Localization was performed using the same methodology employed in Section 5.3.2. The localization snapshots were each 5 seconds long. Data was preprocessed using a filter step, integration and pressure unit conversion step, and finally a clipping and whitening step (see Section 2.3.4). The band limits for filtering and whitening were kept at 200 – 1500 Hz to match the parameters used in the previous section. Since only two closely-spaced sensors were used for these tests, only the source bearing can be accurately ascertained (see Section 5.3.2). As such, the search space was chosen to be a simple angular

scan from 0 to 360 degrees in 1-degree increments. The beamformer output power was normalized to its maximum to be able to compare the three different weighting methods.

The estimate accuracy (see Equation 118) was obtained assuming the strong source present in the negative lags of the correlation ensemble (see Figure 72, Section 4.2.4) was the tug "Pocomoke" whose GPS location was obtained from the automatic identification system (AIS) database (see Figure 122). The only evidence for this being the case is the measured time-difference-of-arrival from the tug's GPS location to the sensors matches somewhat well with the peak locations of the correlation ensemble (see Figure 123).

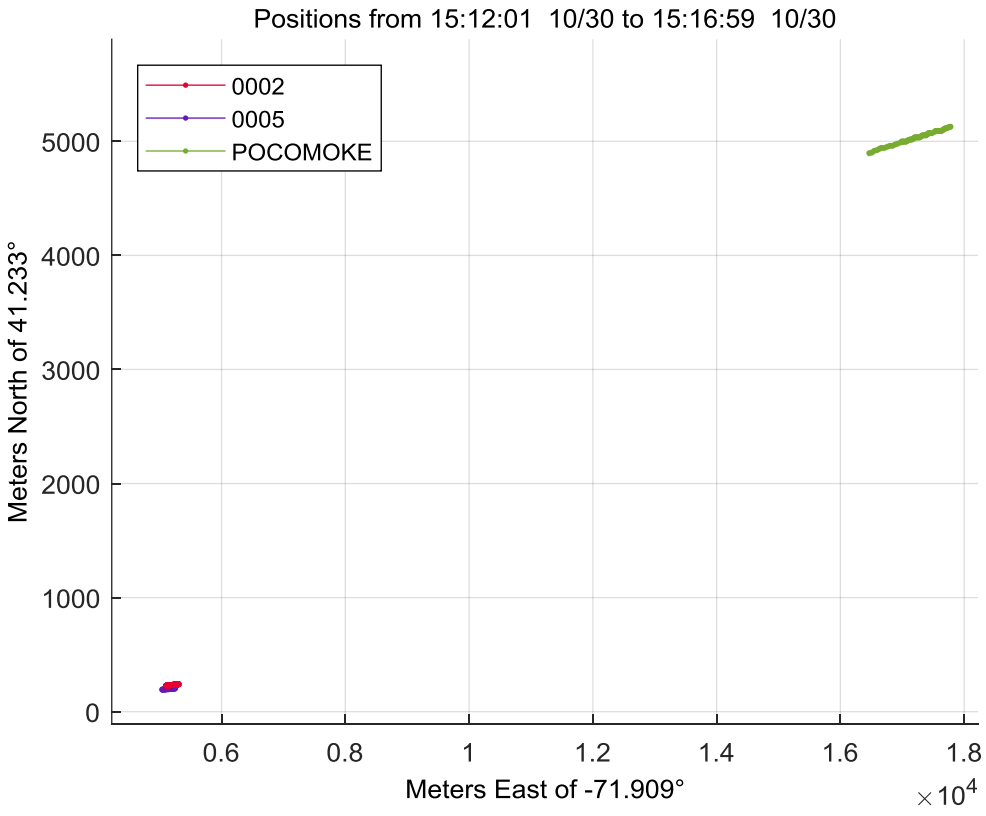


Figure 122: Sensor locations and estimated source "Pocomoke" tug whose location was obtained from AIS data.

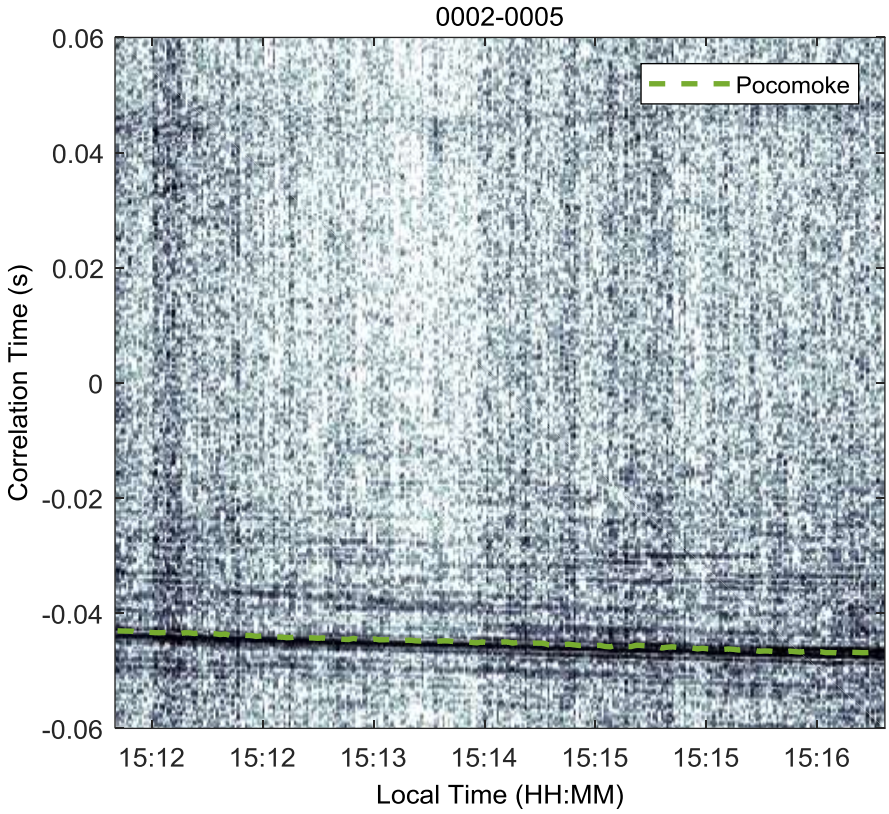


Figure 123: Velocity correlation ensemble amplitude for sensors 0002/0005 normalized to the overall maximum. Black corresponds to 0 dB and white to -10 dB. The TDOA of the Pocomoke is overlaid as a green dashed line, computed assuming a sound speed of 1500 m/s.

The estimate precision metric is identical to that presented in Equation 107. However, due to the search space being one-dimensional, the precision values will be somewhat smaller than the two-dimensional results presented in the previous section.

5.4.2 Results

For the 0002/0005 sensor pair, sample beamformer outputs at 15:15 local time are plotted for each beamforming method in Figure 124. The estimated source direction is shown as a black dotted line.

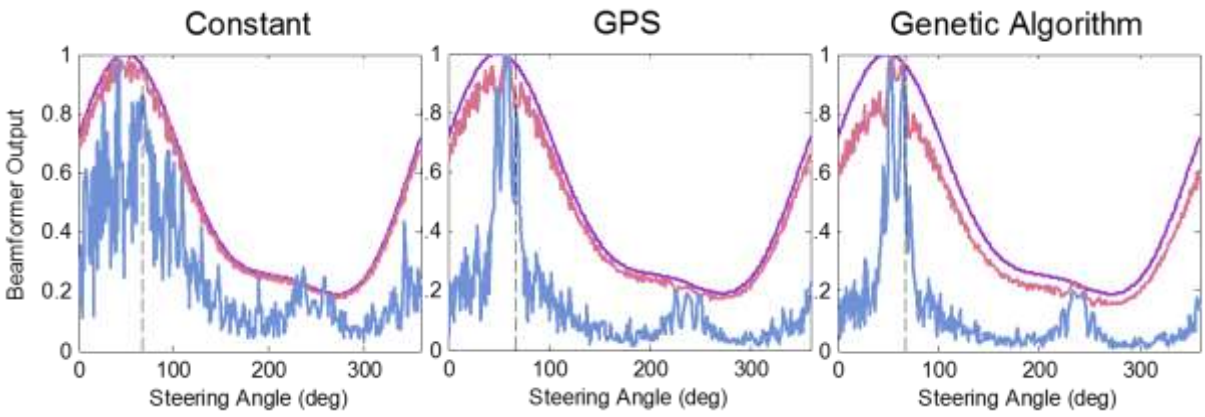


Figure 124: Sample beamformer outputs at 15:15 using a) constant sensor positions, b) GPS-measured sensor positions, and c) genetic algorithm AEL positions. The incoherent output is shown in purple, the coherent in pink, and the cross-coherent in blue.

Each method of obtaining sensor locations (constant, GPS, and genetic algorithm) have very similar incoherent results. This is to be expected, since the bearing to a far-field source does not change much for small changes in sensor location. The coherent method generally follows the incoherent trend, but with small variations because of the cross-coherent components present. The cross-coherent method shows the greatest difference between the three sensor location methods. Using constant separation distance is obviously not ideal, as it carries too much error for the cross-coherent method to work well. The cross-coherent method carries a bias for the GPS and GA results, however there is a strong peak at the true source direction when using the GA sensor locations. This indicates the GPS locations are still somewhat inaccurate, but the GA locations improved upon them enough to improve the source localization.

Plots of the localization error for each method across the entire period are given in Figure 125 for each of the source location methods.

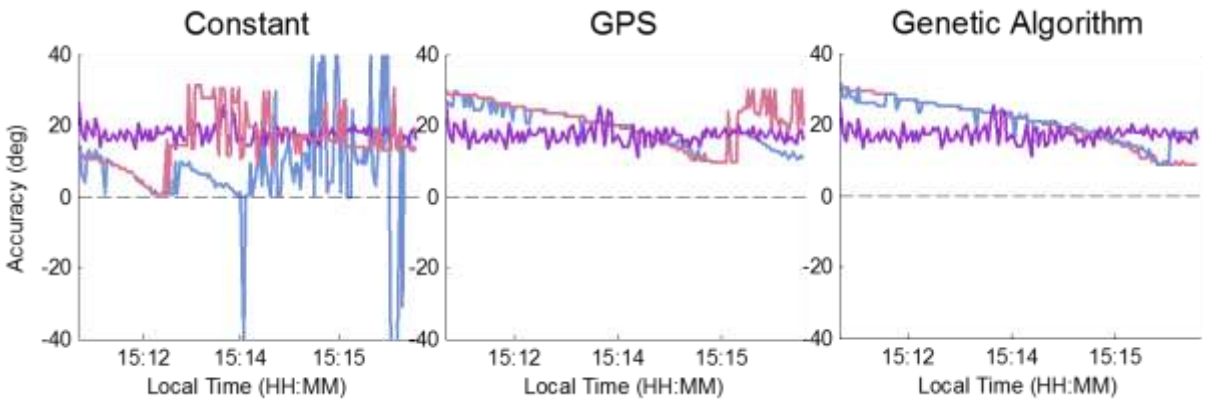


Figure 125: The localization error assuming the source is the "Pocomoke" tug. The localization was performed using a) constant sensor positions, b) GPS-measured sensor positions, and c) genetic algorithm AEL positions. The incoherent error is shown in purple, the coherent in pink, and the cross-coherent in blue.

The most obvious feature of the localization accuracy plot is the presence of approximately 19-degree bias in all the results. This was seen in the sample beamformer outputs in Figure 124, and could stem from many causes. The first cause could be an incorrect assumption of the true source location (i.e. the source was not the Pocomoke tug). The second cause could be a systematic error in the vector sensor orientation measurements, causing the apparent direction to the source to be shifted by the sensor orientation bias. In any case, it is most likely that the beamformers are locating an actual source, and that source is either actually or apparently 19 degrees off of the Pocomoke bearing. However, not much can be determined about the different beamforming methods or sensor location methods from these results. Plots of the localization precision (see Figure 126) will show how precise each method was in locating whichever source they are tracking.

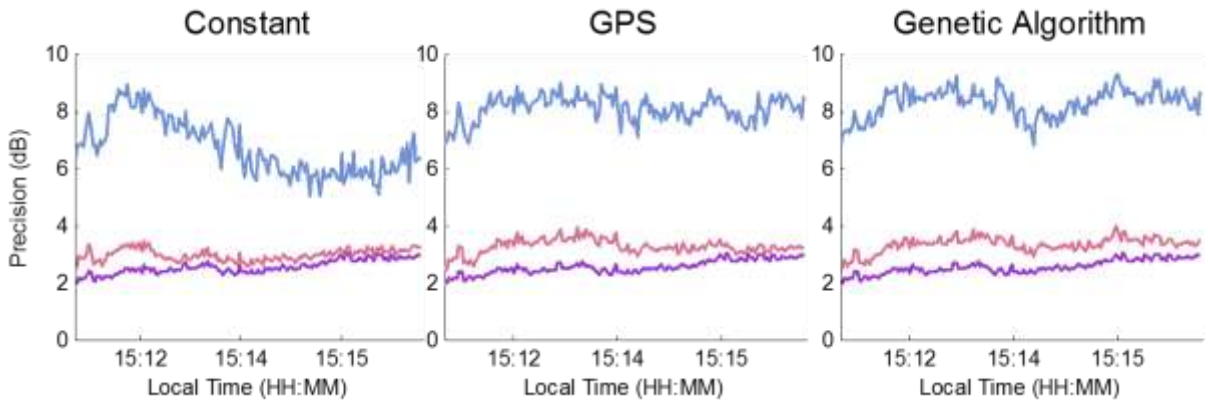


Figure 126: The localization precision assuming the source is the "Pocomoke" tug. The localization was performed using a) constant sensor positions, b) GPS-measured sensor positions, and c) genetic algorithm AEL positions. The incoherent precision is shown in purple, the coherent in pink, and the cross-coherent in blue.

The precision results show that the coherent and incoherent methods perform similarly for any choice of sensor locations. However, the cross-coherent method displays different precision for the different sensor location methods. A higher precision indicates the cross-coherent method obtains a sharper peak at the estimated source bearing, which results from having the correct sensor separation distance. It also indicates that weaker sources (i.e. with lower SNR) are more likely to be detected with the cross-coherent methods than the other 2 methods. The constant sensor locations become more inaccurate over time, thus the cross-coherent precision drops. The GPS and GA estimates have comparable cross-coherent precision, with the GA sometimes slightly higher than the GPS, indicating the GA measurements may be superior to GPS at times.

5.4.3 Conclusion

Time delays identified by the constrained genetic algorithm presented in Chapter 4 were used to correct erratic GPS measurements. The corrected sensor locations resulted in more precise localization results, indicating the distance measurements were accurate and

the sensor location corrections valid. Possible improvements to the analysis performed here would be to attempt array localization using ambient noise while simultaneously tracking a known source, possibly even one which is not purely in the endfire direction.

5.5 Conclusion

Beamforming using weighted correlation matrices provides opportunities for novel methods of locating sources, with varying degrees of sensitivity to noise and sensor location errors. A novel cross-coherent weighting method was proposed which maintains localization accuracy and precision even in the presence of noise. Simulated and experimental data showed that coherent beamforming requires accurate knowledge of sensor locations, and this was corroborated by applying the ambient noise AEL method described in Chapter 4 to improve sensor location estimates and thus, the localization performance.

APPENDIX A
VECTOR SENSOR CALIBRATIONS

As described in Chapter 2, the vector sensor IMU components showed different scaling and orientation depending on the serial number. To confirm the acoustic sensitivities, each sensor was tested with a reference measurement to obtain the hydrophone sensitivity and accelerometer sensitivity. All sensors had a hydrophone sensitivity of -162 dB re 1 V/µPa, however the accelerometer sensitivities varied.

The first method mounted the sensor rigidly to a mechanical shaker with a reference accelerometer also attached. The shaker was operated from 40 to 240 Hz and the reference accelerometer used to find the VS-301 sensitivity. The second method mounted the sensor to a piezoelectric shaker stack and the displacement measured with a Laser Doppler Vibrometer (LDV). The frequency range tested was between 50 and 2500 Hz, however there were suspected resonances of the setup at this range which may have caused the results to be somewhat less accurate. Figure 127 and Figure 128 show the axial and radial configurations of the sensor mounted to the piezoelectric stack.

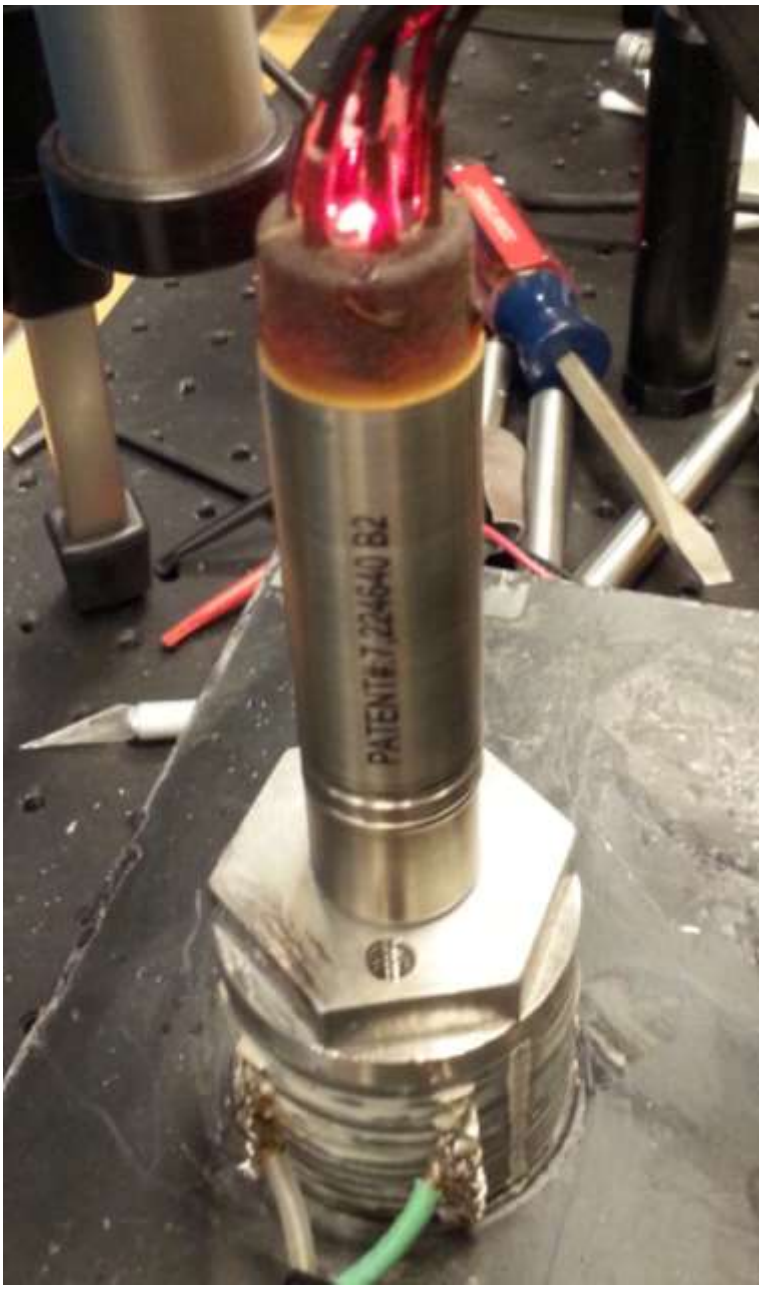


Figure 127: Axial accelerometer calibration setup using piezoelectric shaker (bottom) and LDV (red laser dot).

Figure 128: Radial accelerometer calibration setup using piezoelectric shaker (bottom) and LDV (laser dot not shown).

The calibration results are shown below for the shaker calibration and the axial (z-axis) LDV calibration. Each sensor which was tested is shown, using the notation SN VSXX to denote the serial number of the VS-301.

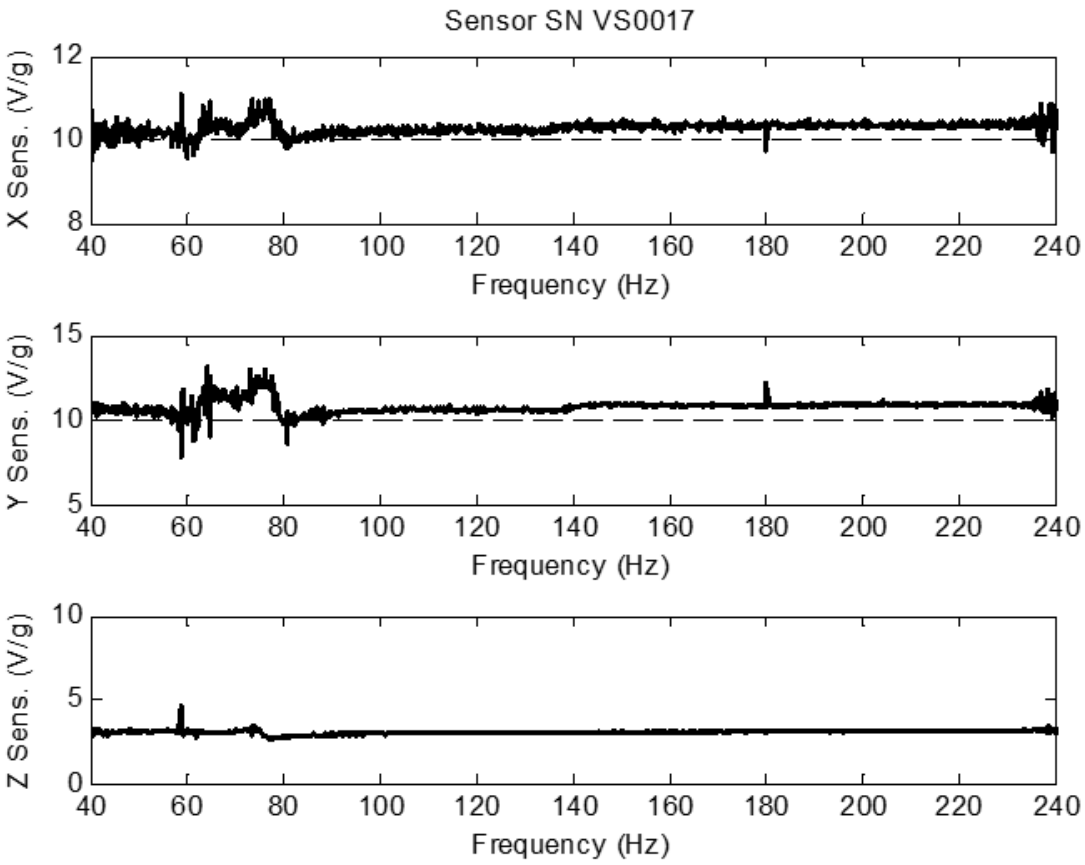


Figure 129: Mechanical shaker calibration curves for X, Y, Z axes of sensor #0017. Assumed reference sensitivities are shown in black dashed lines.

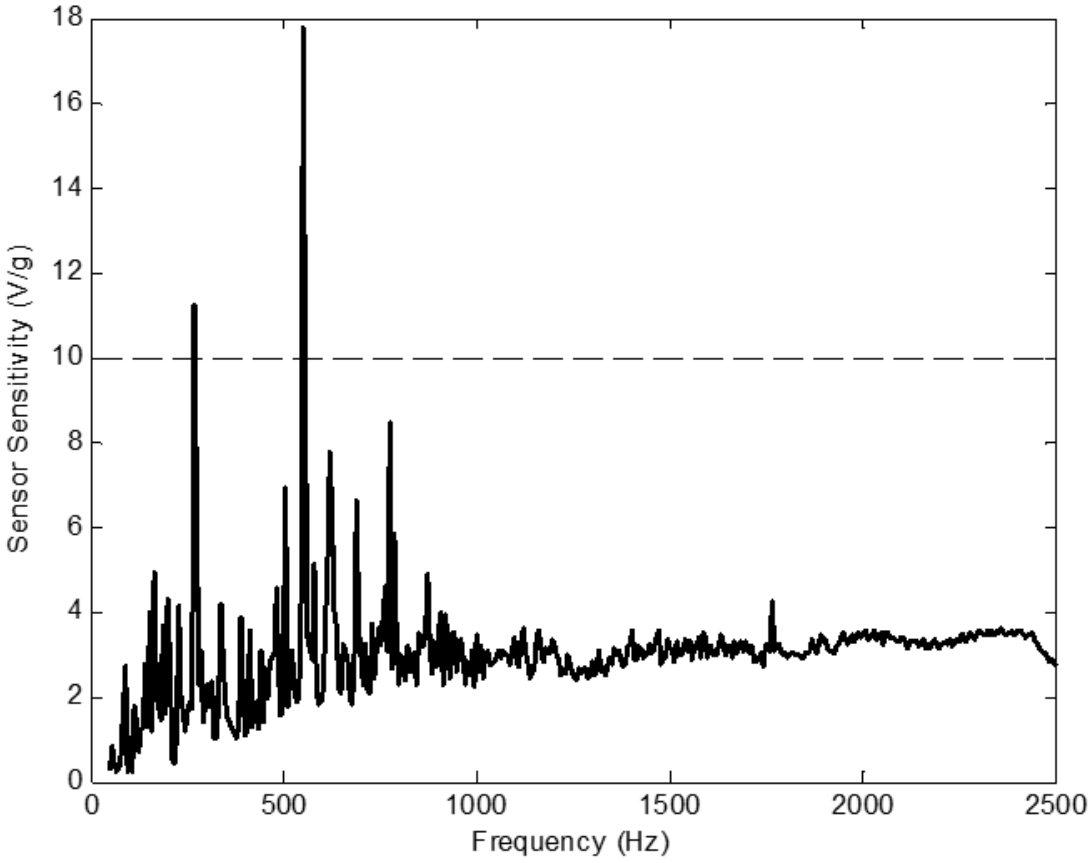


Figure 130: Piezoelectric shaker/LDV calibration curve for Z-axis of sensor #0017. Assumed reference sensitivity is shown in black dashed line.

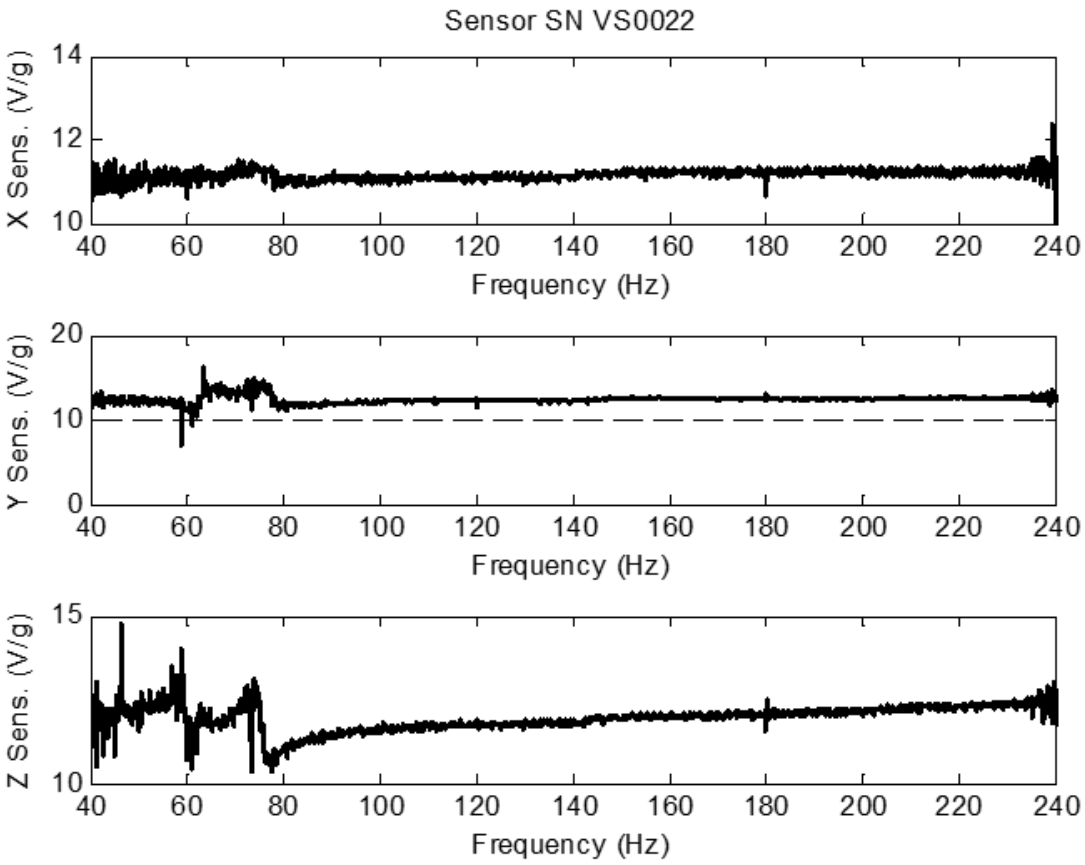


Figure 131: Mechanical shaker calibration curves for X, Y, Z axes of sensor #0022. Assumed reference sensitivities are shown in black dashed lines.

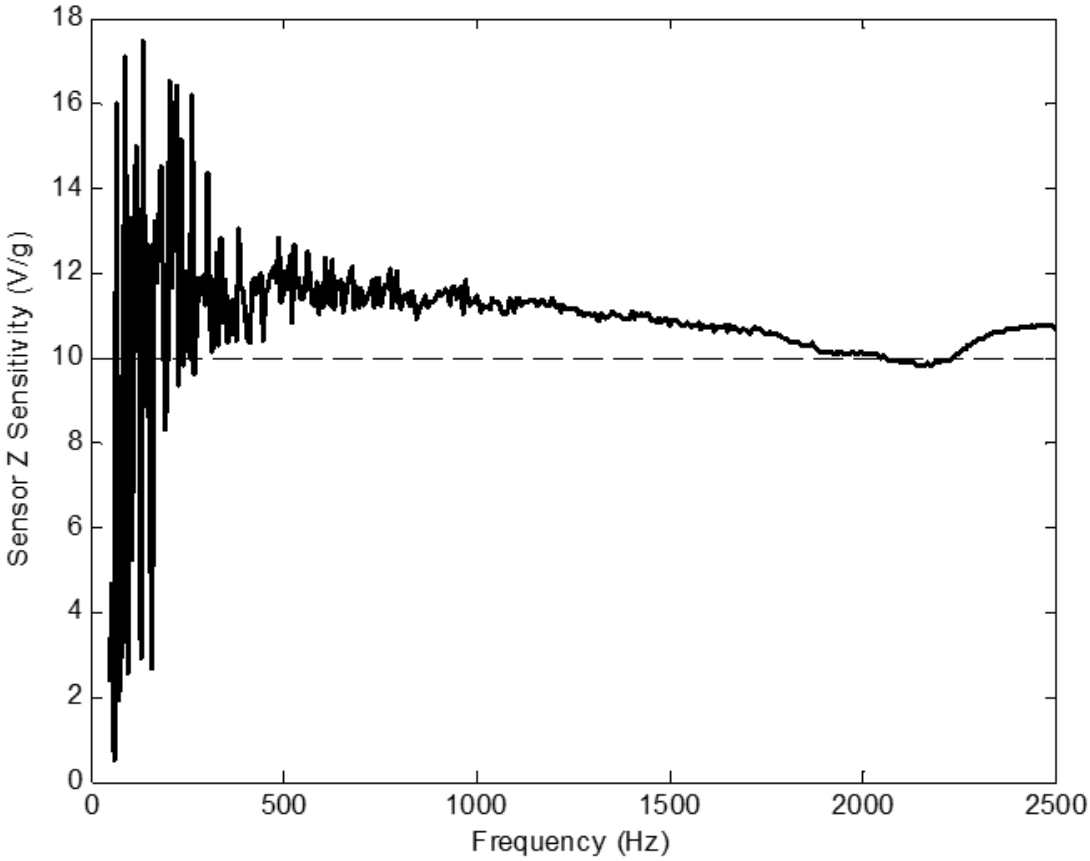


Figure 132: Piezoelectric shaker/LDV calibration curve for Z-axis of sensor #0022. Assumed reference sensitivity is shown in black dashed line.

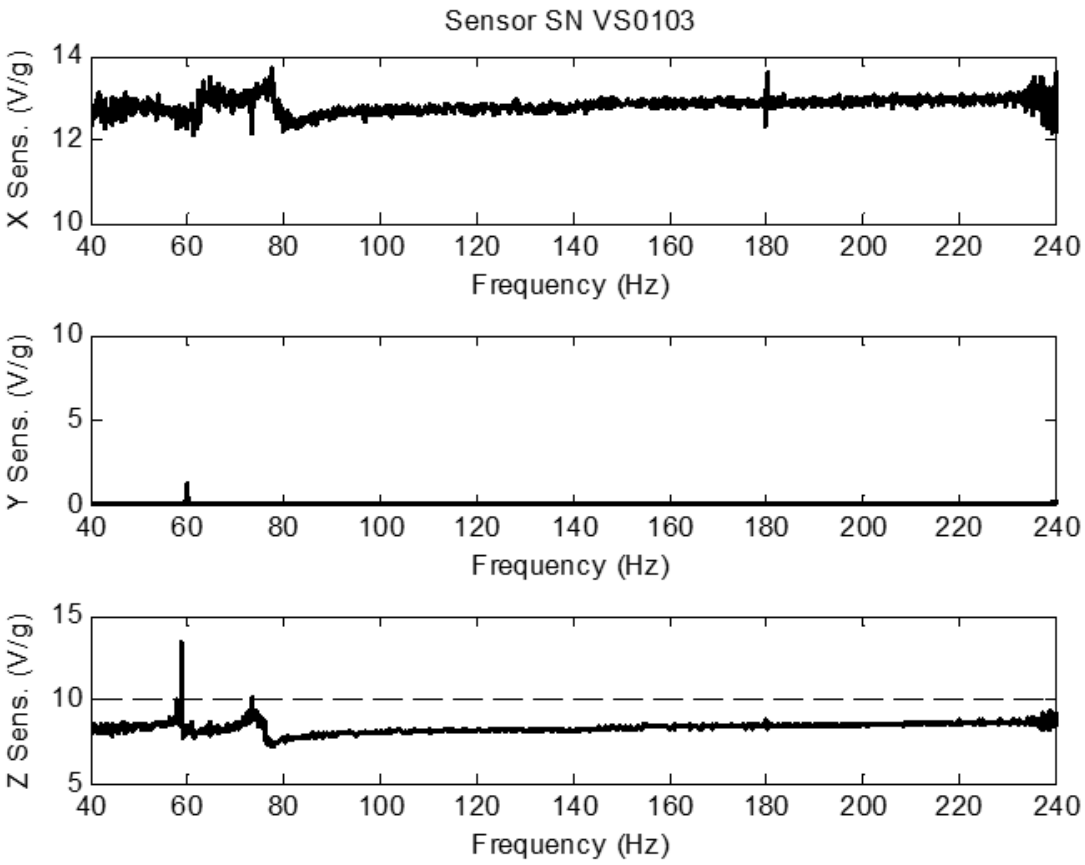


Figure 133: Mechanical shaker calibration curves for X, Y, Z axes of sensor #0103. Assumed reference sensitivities are shown in black dashed lines.

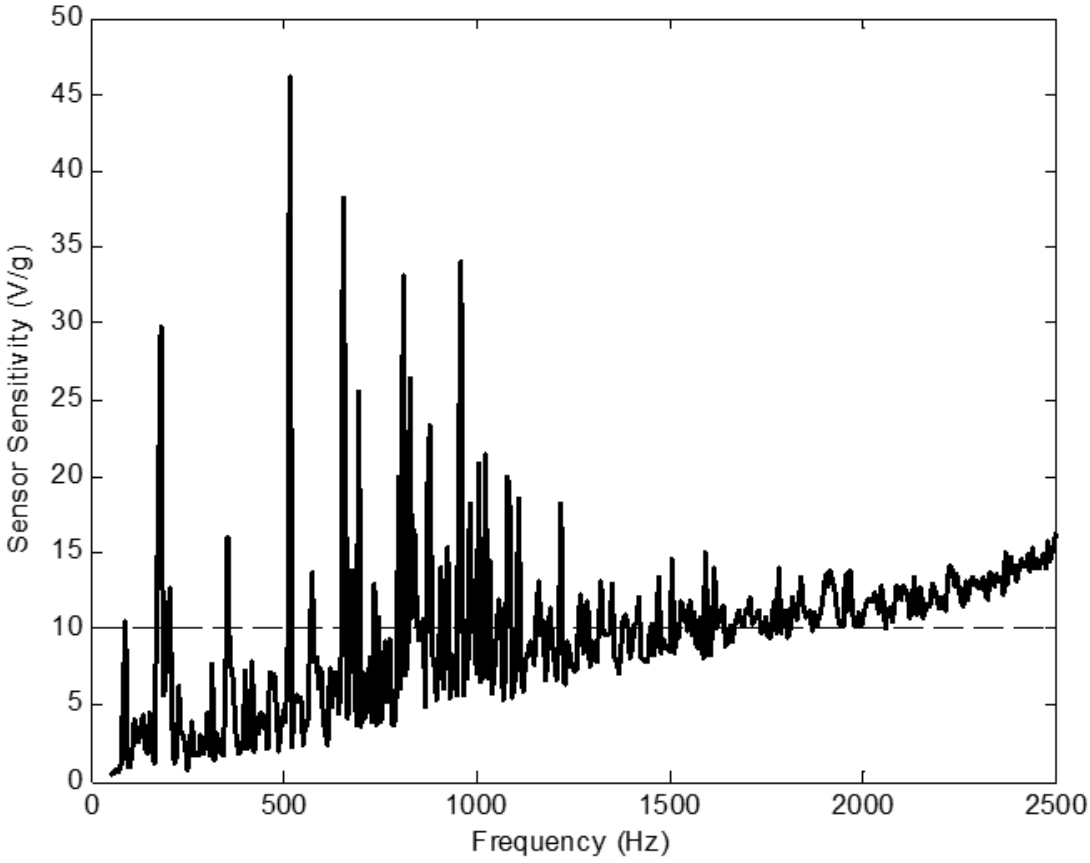


Figure 134: Piezoelectric shaker/LDV calibration curve for Z-axis of sensor #0103. Assumed reference sensitivity is shown in black dashed line.

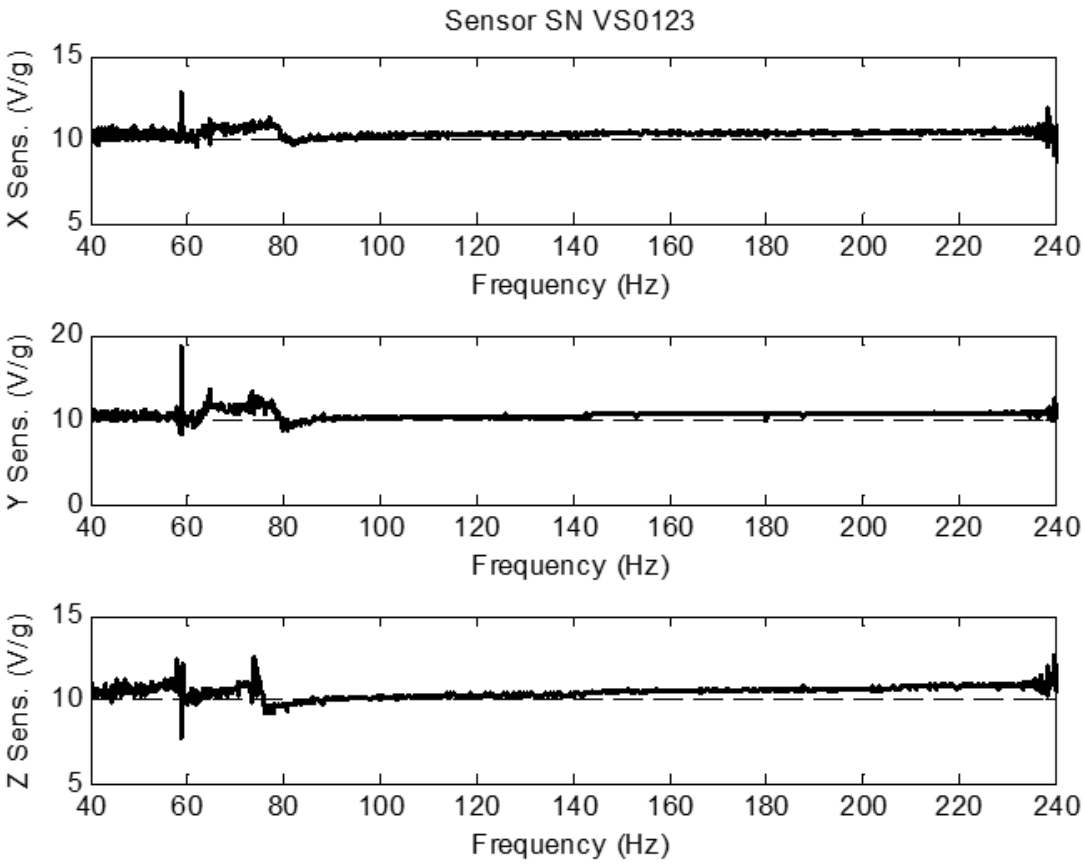


Figure 135: Mechanical shaker calibration curves for X, Y, Z axes of sensor #0123. Assumed reference sensitivities are shown in black dashed lines.

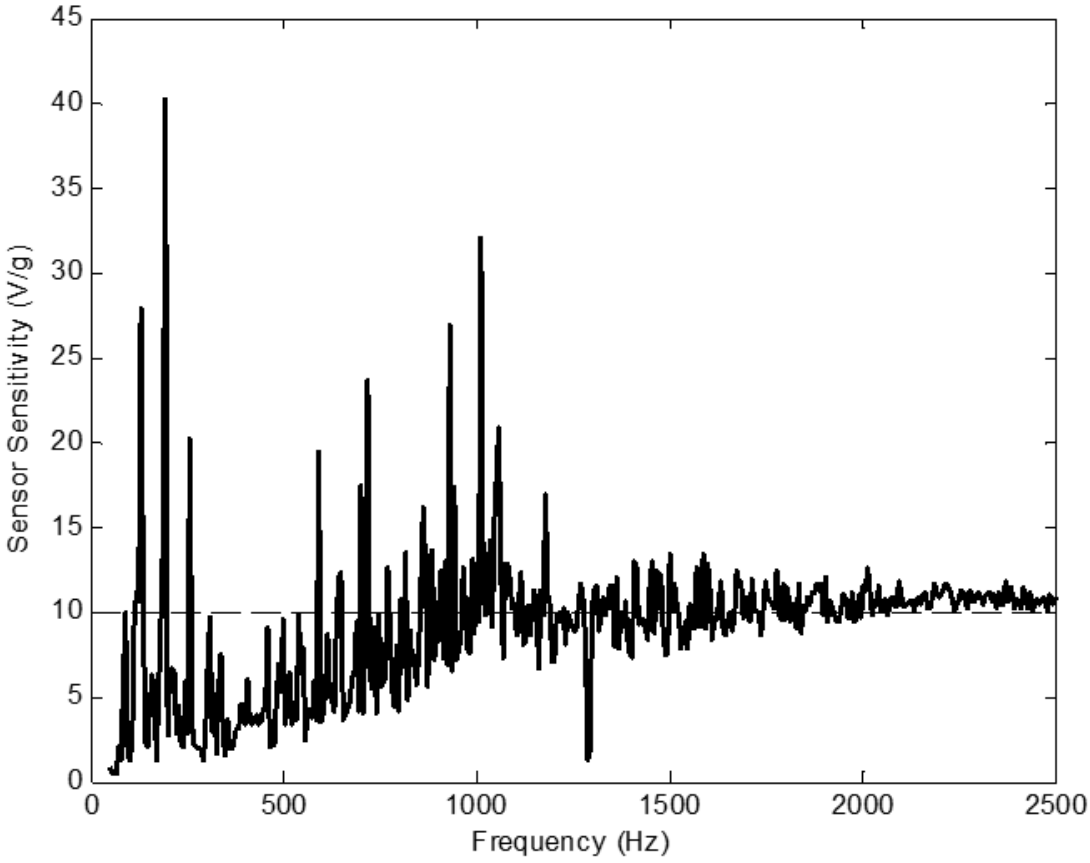


Figure 136: Piezoelectric shaker/LDV calibration curve for Z-axis of sensor #0123. Assumed reference sensitivity is shown in black dashed line.

The calibrations show that there may have been some errors in obtaining the sensitivities using the LDV method. However, the mechanical shaker method appears to have more consistent results. The sensitivities of the sensors appear to vary depending on the serial number, with the most common being around 10 V/g. One sensor showed 3 V/g sensitivity on just one component, and another appears to have a malfunctioning axis. This axis was oriented vertically in experiments whenever possible, to limit its effect on localization performance. Figure 137 shows the hydrophone calibration performed in water using a reference hydrophone and a chirp signal played through a 15 cm diameter source.

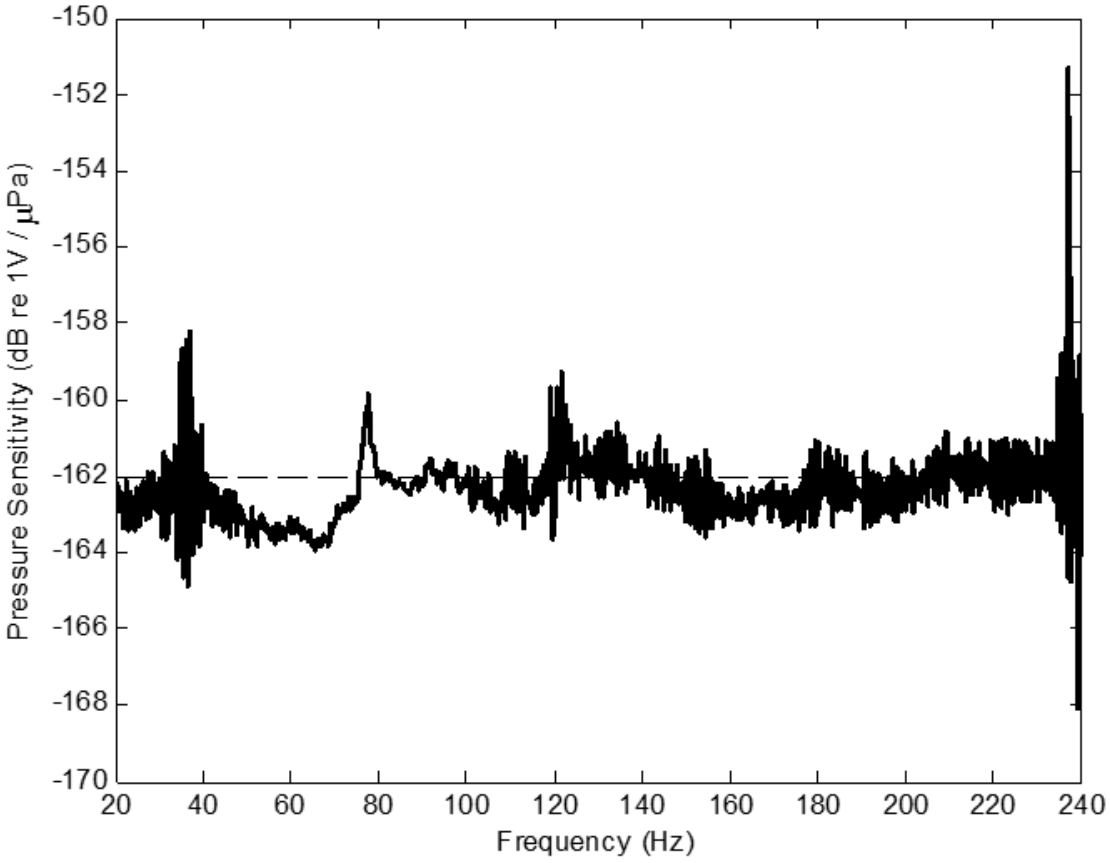


Figure 137: Reference hydrophone calibration curve for hydrophone of sensor #0123. Assumed reference sensitivity is shown in black dashed line.

APPENDIX B
DATA SPECTROGRAMS

Plotted below are sample spectrograms of the acoustic data collected for various at-sea tests. The power spectral density computed from 1-second bins is plotted over 15-minute windows, plotted in logarithmic scale. No preprocessing (filtering, downsampling, rotation, etc.) was performed on the acoustic data. Only a pressure and horizontally-oriented velocity channel are shown for each sensor. The sample rate was 39.0625 kHz, but only frequencies up to 2 kHz are shown to match the sensor's usable frequency range.

Figure 138 shows the pressure and X-axis velocity for Sensor 0005 on October 29th, 2016 when the sea-state was higher than on the following day, plotted in Figure 139. The spectra during the river-bottom deployment on the 29th is plotted in Figure 140. For comparison, the spectra are plotted for the June 2016 deployment which employed the 2nd revision floats. Figure 141 shows the spectra for June 20th, which had the highest sea-state, and Figure 142 shows the spectra for June 22nd , which had a calmer sea-state.

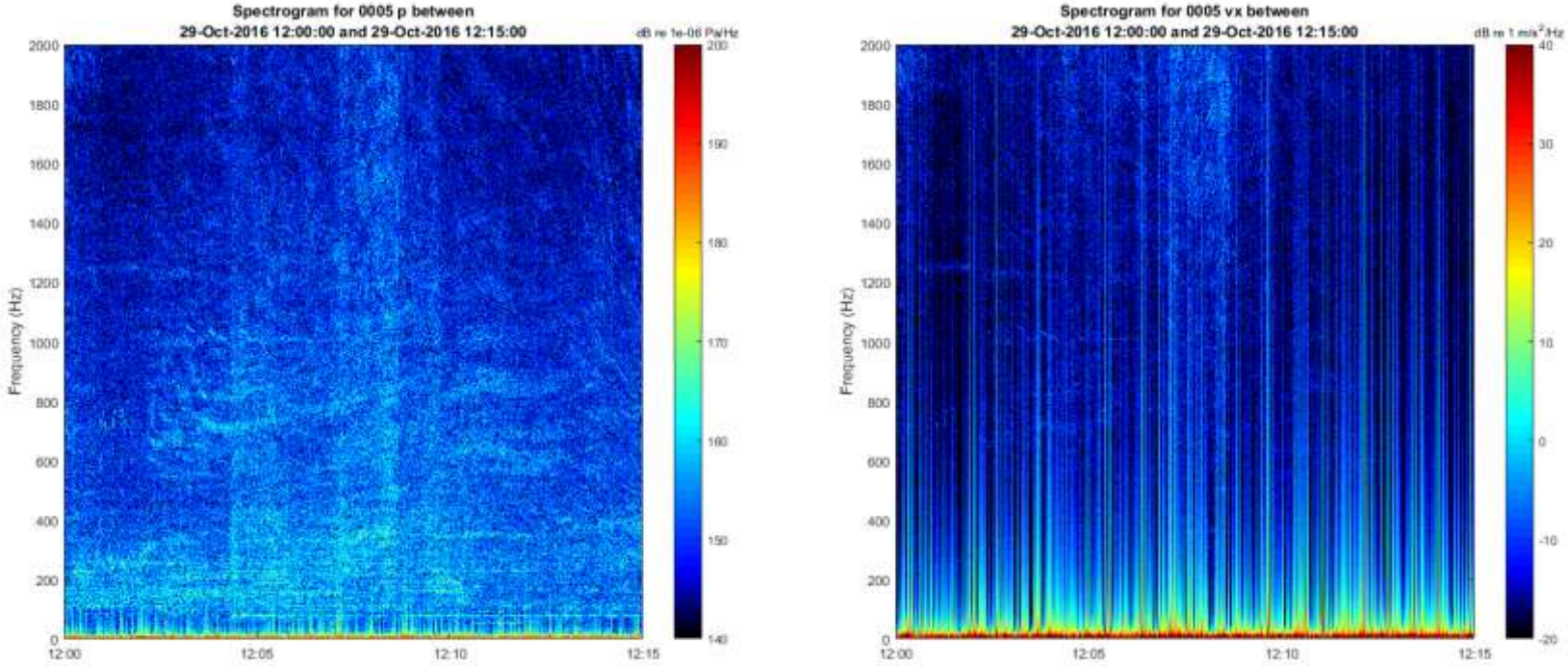


Figure 138: Representative spectrograms for Sensor 0005's pressure and X-axis velocity for October 29, 2016 from 12:00 – 12:15 local time.

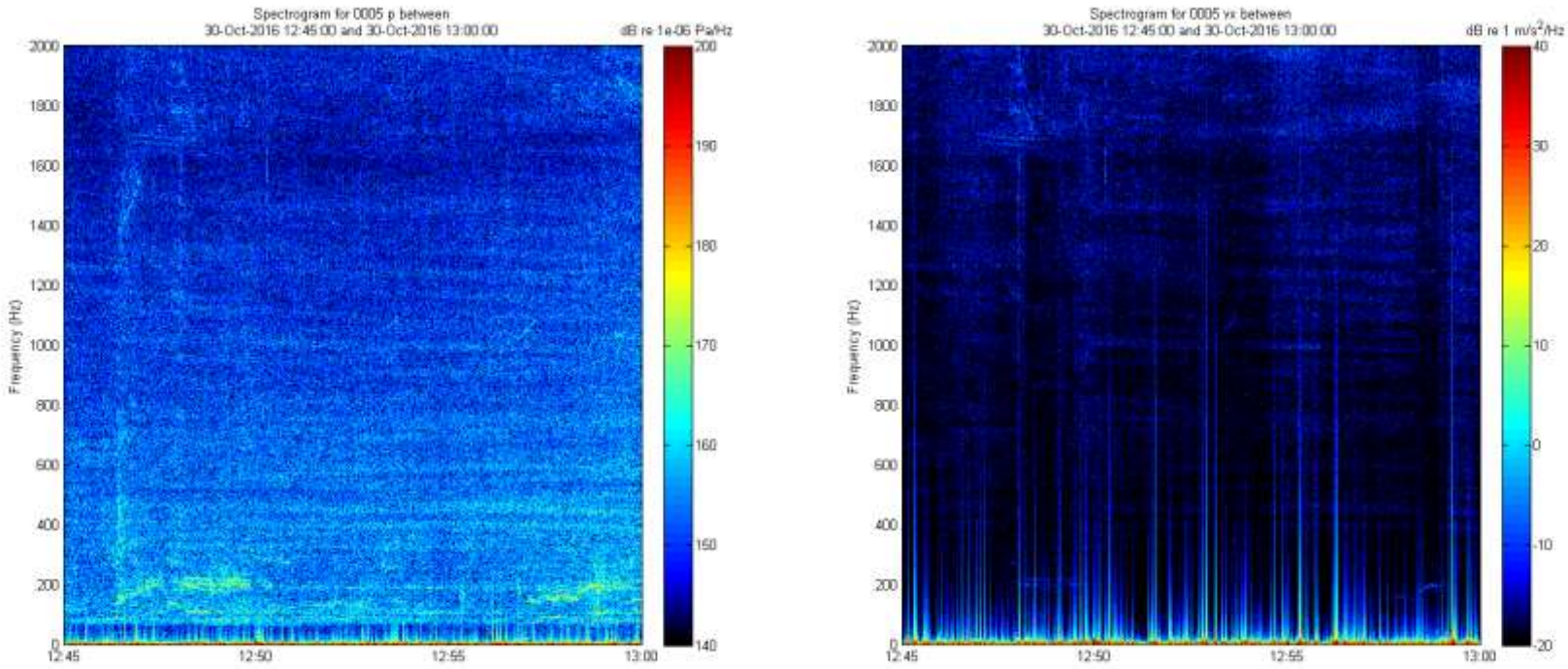


Figure 139: Representative spectrograms for Sensor 0005's pressure and X-axis velocity for October 30, 2016 from 12:45 – 13:00 local time.

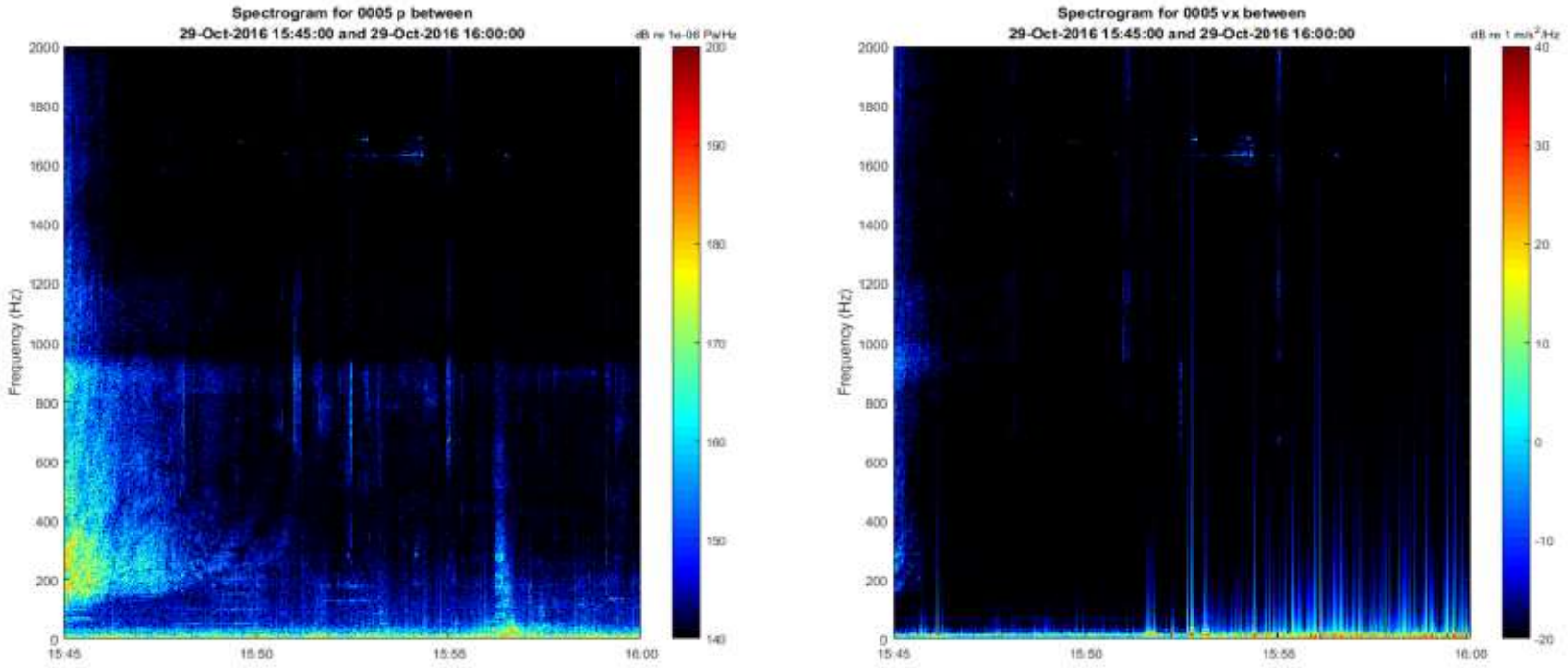


Figure 140: Representative spectrograms for Sensor 0005's pressure and X-axis velocity for October 29, 2016 from 15:45 – 16:00 local time. The sensor frame was stationary on the bottom of the river for this period of time.

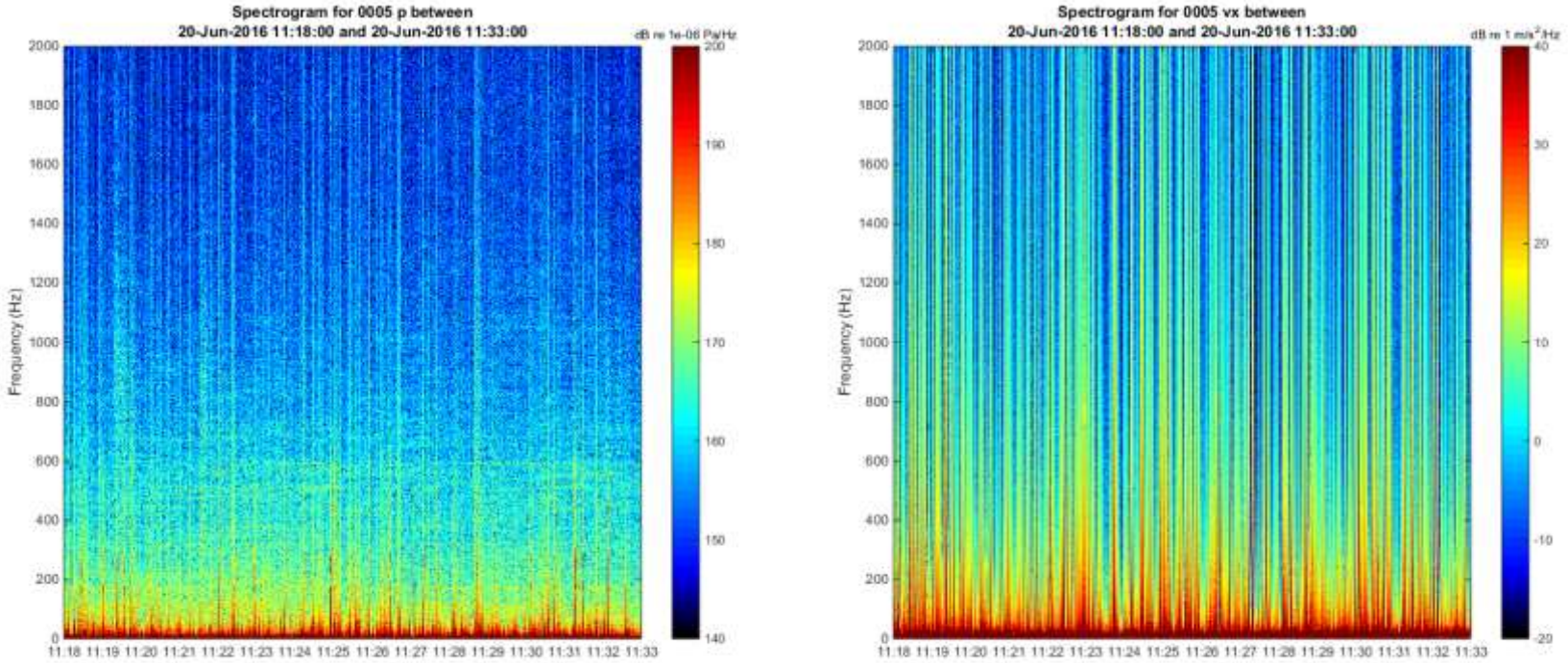


Figure 141: Representative spectrograms for Sensor 0005's pressure and X-axis velocity for June 20, 2016 from 11:18 – 11:33 local time. The second generation of floats was in use here, and the recorder gains set too high, resulting in clipped data.

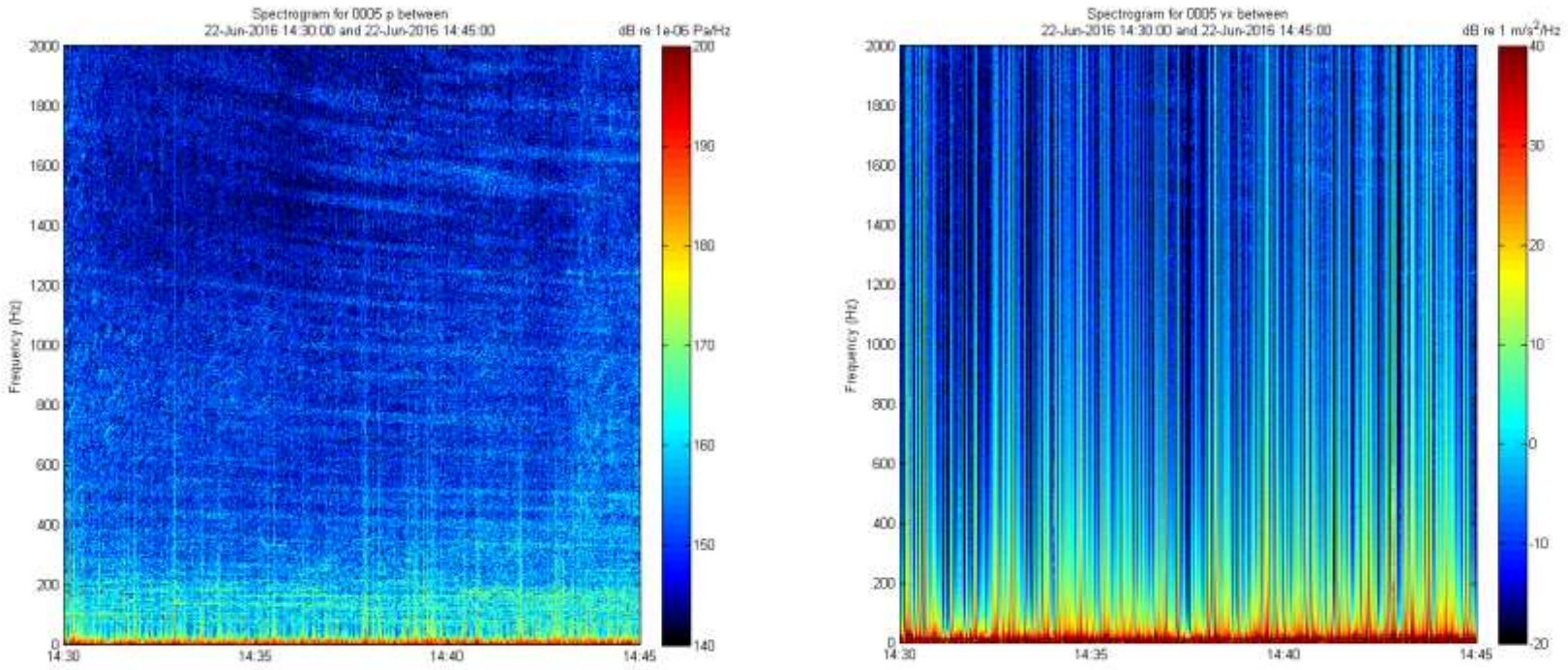


Figure 142: Representative spectrograms for Sensor 0005's pressure and X-axis velocity for June 22, 2016 from 14:30 – 14:45 local time. The second generation of floats was in use here, resulting in clipped data on the velocity channels.

APPENDIX C
TIME-DOMAIN / FREQUENCY-DOMAIN COMPARISON

Using the material from Hawkes and Nehorai 2001 and Cox et al. 2009, the correlation between two separated vector sensors in the presence of isotropic noise is analyzed for the special case where the vector between the sensors is aligned with their x-axes. This assumption will work for any two sensors, provided their data is rotated into a baseline-x coordinate system, and the noise is isotropic. At a given frequency ω, the correlation matrix between the two sensors in the presence of only isotropic noise is given by

$$R_{12}(\omega) = \begin{bmatrix} j_0 & i\,j_1 & 0 & 0 \\ i\,j_1 & \frac{1}{3}j_0 - \frac{2}{3}j_2 & 0 & 0 \\ 0 & 0 & \frac{1}{3}j_0 + \frac{1}{3}j_2 & 0 \\ 0 & 0 & 0 & \frac{1}{3}j_0 + \frac{1}{3}j_2 \end{bmatrix} \qquad 119$$

Note the argument *ks*, where *s* is the separation distance, is omitted after each j_n (spherical Bessel functions). Plotting the non-zero terms in the correlation matrix against *ks*/2π, which gives the ratio of the separation distance relative to a wavelength, shows the magnitude of each component relative to each other (see Figure 143).

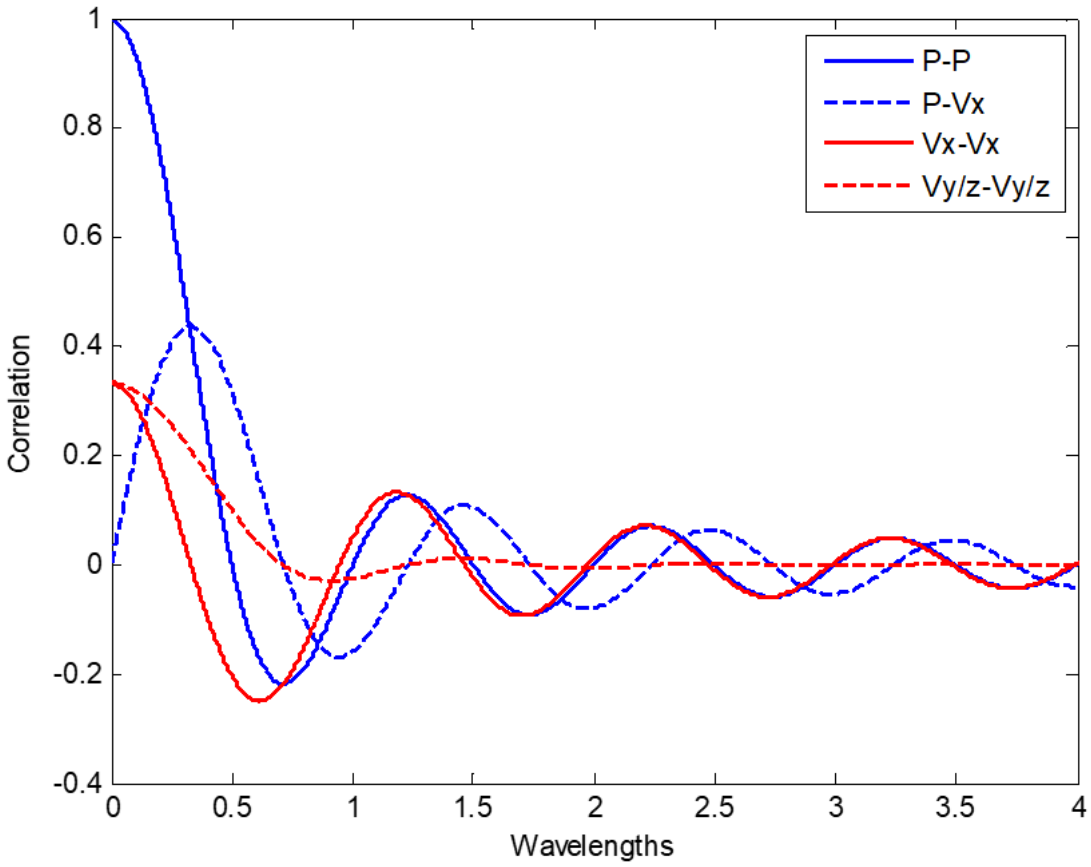


Figure 143: Narrowband spatial correlation for various components of a vector sensor. The abscissa is represented in multiples of the narrowband wavelength $\lambda = c/f$.

For large separation distances relative to a wavelength, the correlation of the ambient noise decreases, as is expected. However, a surprising consequence is that correlations between pressure channels and correlations between velocity channels are nearly identical for large separations.

The above narrowband correlation functions are used to simulate wideband correlations by taking an IFFT on a band-limited set of narrowband noise correlations. With a spacing of 10 meters, sound speed of 1500 m/s, sample rate of 20 kHz, and a band between 1 and 9000 Hz, the theoretical "infinite"-band result of a step function is obtained

for pressure-pressure correlation, whose derivative gives the Green's function estimates. The velocity-velocity correlation is markedly different in this case, however, when only higher frequencies are used, it approaches the pressure-pressure correlation.

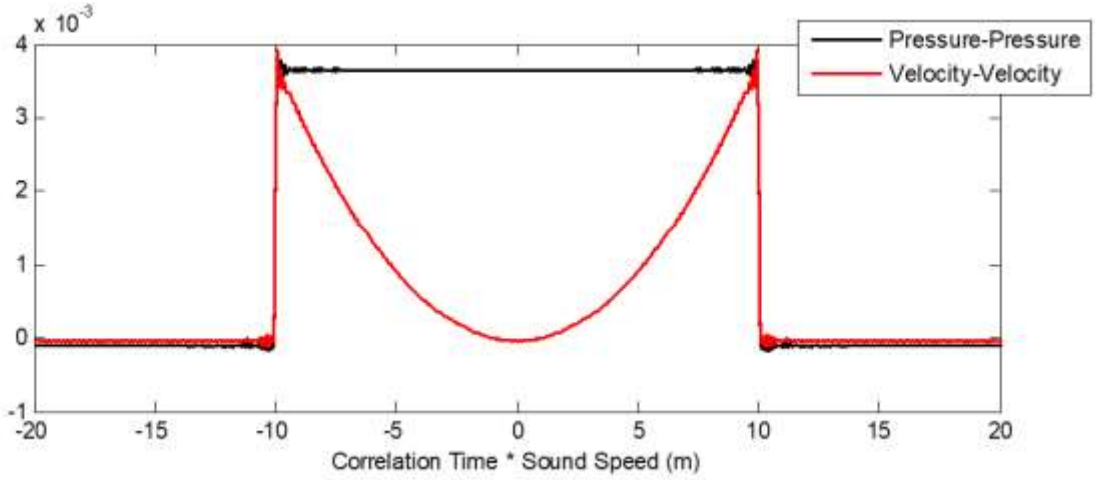


Figure 144: Simulated wideband correlations of pressure and velocity components for "infinite" band (i.e. no filtering).

For a more limited band of 20-1500 Hz, the correlations become more similar:

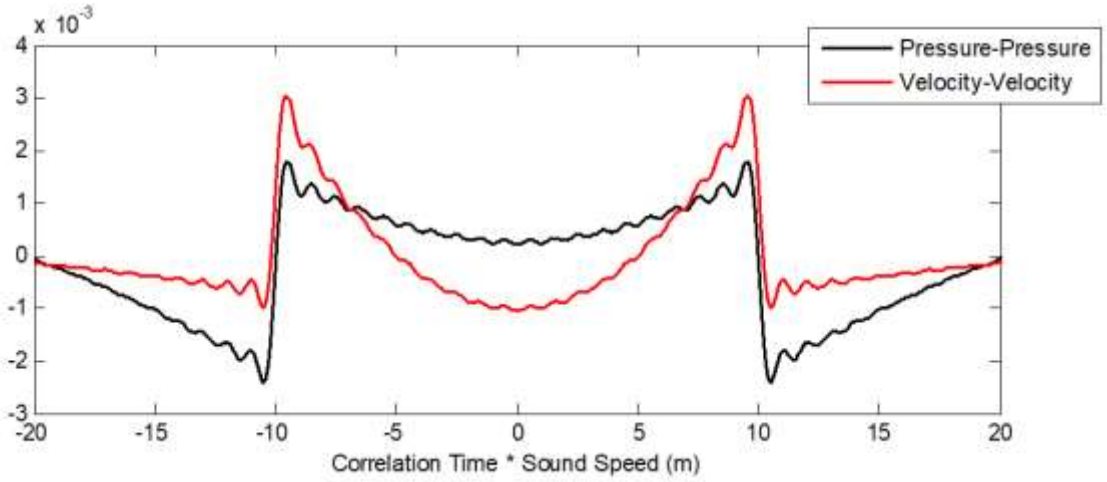


Figure 145: Simulated wideband correlations of pressure and velocity components for 20-1500 Hz band.

And finally, for the typically-used band of 200-1500 Hz:

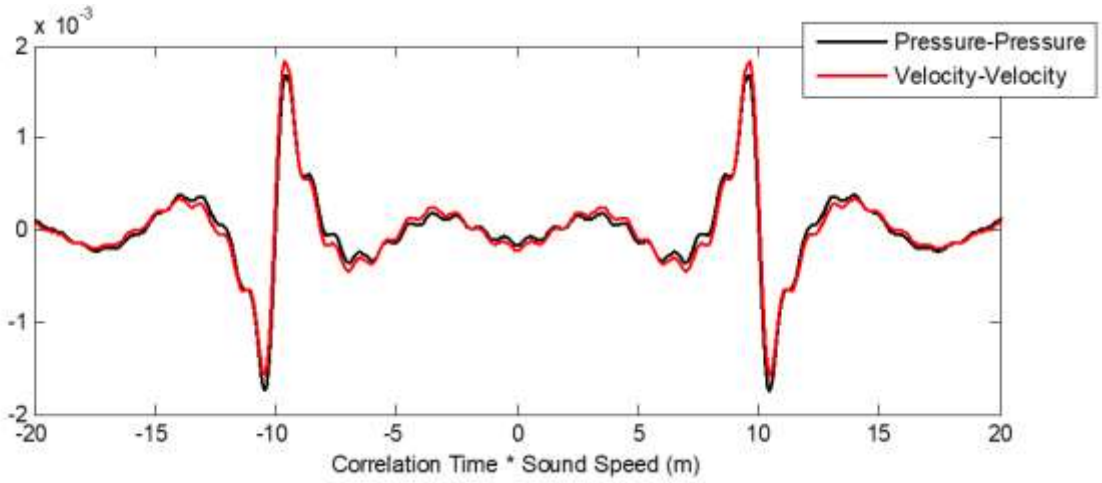


Figure 146: Simulated wideband correlations of pressure and velocity components for 200-1500 Hz band.

The reason the pressure-pressure and velocity-velocity converge is seen in Figure 143, because their spectral correlation functions are nearly identical for large frequencies, so when filtered above any frequency whose wavelength is about equal to the spacing, the functions are nearly identical. Thus, for the typical inter-element spacing experienced at sea (20-60 m), the pressure-pressure and velocity-velocity correlations will be identical if only frequencies above 25-75 Hz are used.

To confirm the time-domain results presented in Chapter 3 match the frequency-domain results presented by Hawkes and Nehorai 2001 and Cox et al. 2009, the Bessel function method plotted in Figure 146 is compared to the convolution formulation of Chapter 3 in Figure 147.

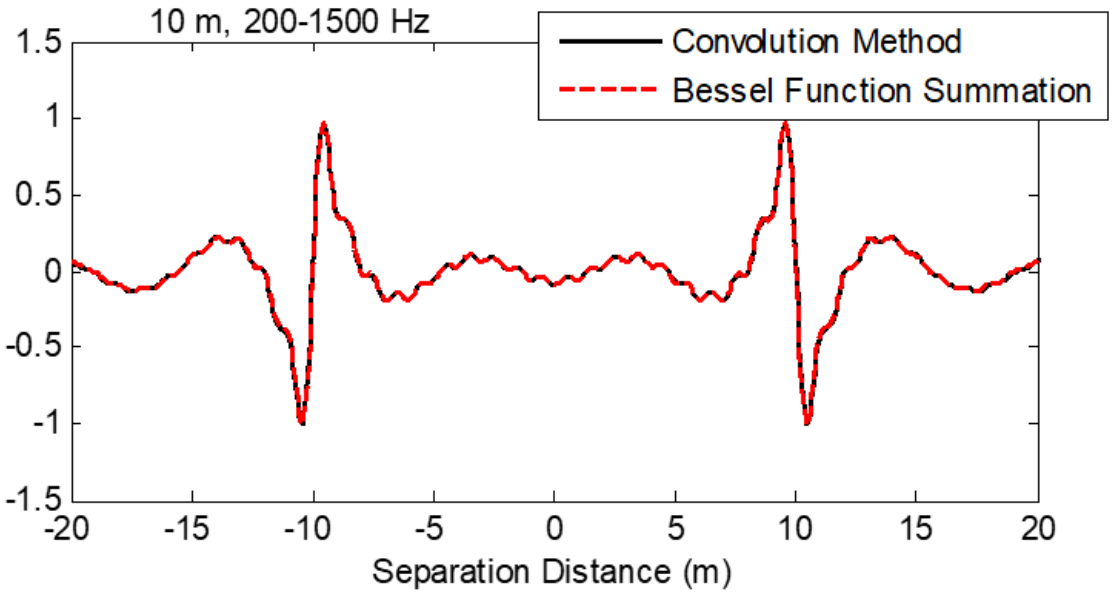


Figure 147: Comparison of pressure correlations computed with the convolution method presented in Chapter 3 (black solid line) and the wideband summation of the Bessel function expressions (red dashed lines). The sensor separation distance is 10 meters, and the bandwidth 200-1500 Hz.

The results match exactly, indicating the equivalence of the two interpretations. Lastly, the axial velocity correlations are compared for 10 meter spacing, 20-1500 Hz bandwidth in Figure 148.

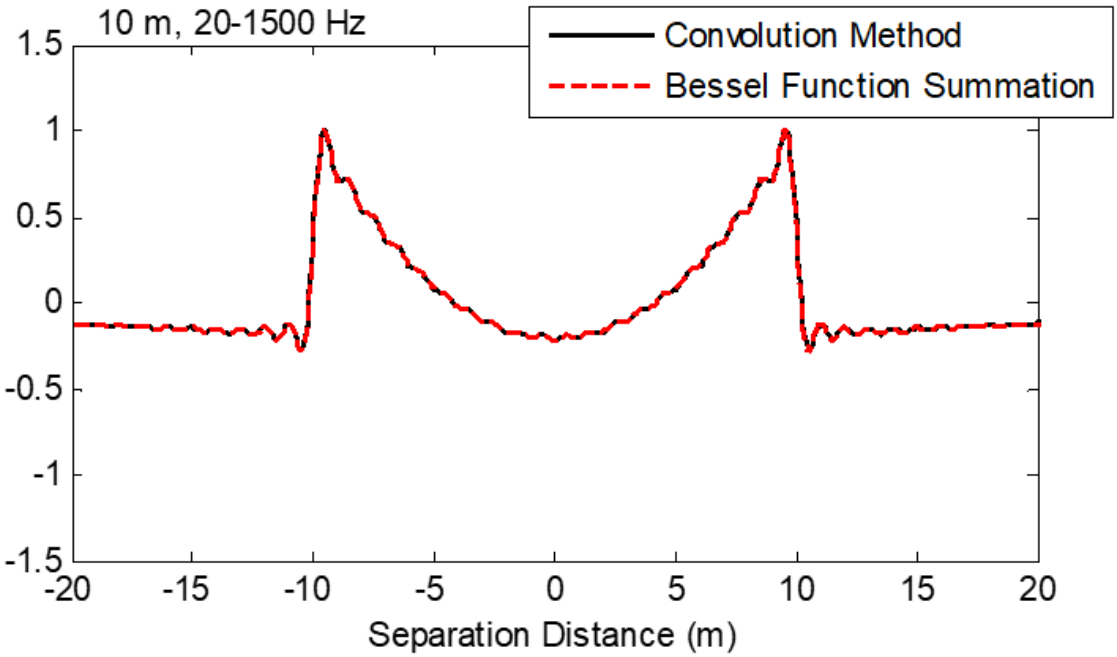


Figure 148: Comparison of velocity correlations computed with the convolution method presented in Chapter 3 (black solid line) and the wideband summation of the Bessel function expressions (red dashed lines). The sensor separation distance is 10 meters, and the bandwidth 20-1500 Hz.

Again, the results are exactly identical. The equivalence of the inverse Fourier transform of the Bessel solution and the convolution method indicates they are time/frequency Fourier pairs.

APPENDIX D
NOISE AUTOCORRELATION DERIVATION

The autocorrelation is related to the power spectral density by the Fourier transform

$$C_{ss}(\tau) = \int_{-\infty}^{\infty} S(f) \cdot e^{j2\pi f\tau}\, df \tag{120}$$

where the power spectral density is given by $S(f)$. If a noise process has equal power across a bandwidth B centered on a frequency f_c, its power spectral density is given by

$$S(f) = \begin{cases} \dfrac{\sigma_s^2}{2B}, & f_c - \dfrac{B}{2} \le |f| \le f_c + \dfrac{B}{2} \\ 0\ , & \text{otherwise} \end{cases} \tag{121}$$

Thus, the integral in Equation 120 becomes

$$C_{ss}(\tau) = \int_{f_c - B/2}^{f_c + B/2} \frac{\sigma_s^2}{2B} \cdot e^{j2\pi\tau f}\, df + \int_{-f_c - B/2}^{-f_c + B/2} \frac{\sigma_s^2}{2B} \cdot e^{j2\pi\tau f}\, df \tag{122}$$

Integrating and simplifying the expression results in

$$C_{ss}(\tau) = \frac{\sigma_s^2}{2B} \frac{1}{2j\pi\tau} [e^{j2\pi\tau(f_c + B/2)} - e^{j2\pi\tau(f_c - B/2)} + e^{j2\pi\tau(-f_c + B/2)} - e^{j2\pi\tau(-f_c - B/2)}] \tag{123}$$

which further simplifies to

$$C_{ss}(\tau) = \frac{\sigma_s^2}{\pi B\tau} \frac{1}{2j} \left[e^{j2\pi\tau f_c} + e^{-j2\pi\tau f_c}\right]\left[e^{j\pi\tau B} - e^{-j\pi\tau B}\right] \tag{124}$$

Applying the Euler formula for sine and cosine functions yields

$$C_{ss}(\tau) = \frac{\sigma_s^2}{\pi B \tau} \cdot \cos(2\pi f_c \tau) \cdot \sin(\pi B \tau) \quad 125$$

which is finally simplified using the definition of $\mathrm{sinc}(x) = \sin(x)/x$ to yield

$$C_{ss}(\tau) = \sigma_s^2 \cdot \cos(2\pi f_c \tau) \cdot \mathrm{sinc}(\pi B \tau) \quad 126$$

Thus, the noise process autocorrelation is obtained from its speculated power spectral density.

APPENDIX E
ADDITIONAL STOCHASTIC SEARCH SIMULATION RESULTS

Shown in Figure 149 is the average RMSE of 20 different realizations of base correlations having varying PSNR from 0.1 to 2 (-10 dB to 3 dB). Any results for SNR higher than 2 (3 dB) are effectively redundant as the correlation peak is clearly visible even in the base correlations, thus extended averaging times are unnecessary. The true time delays (blue circles) were obtained from linear interpolations of the experimental delays described above. The estimated time delays from the genetic algorithm are superimposed as red crosses. The averaged correlations are plotted in black underneath.

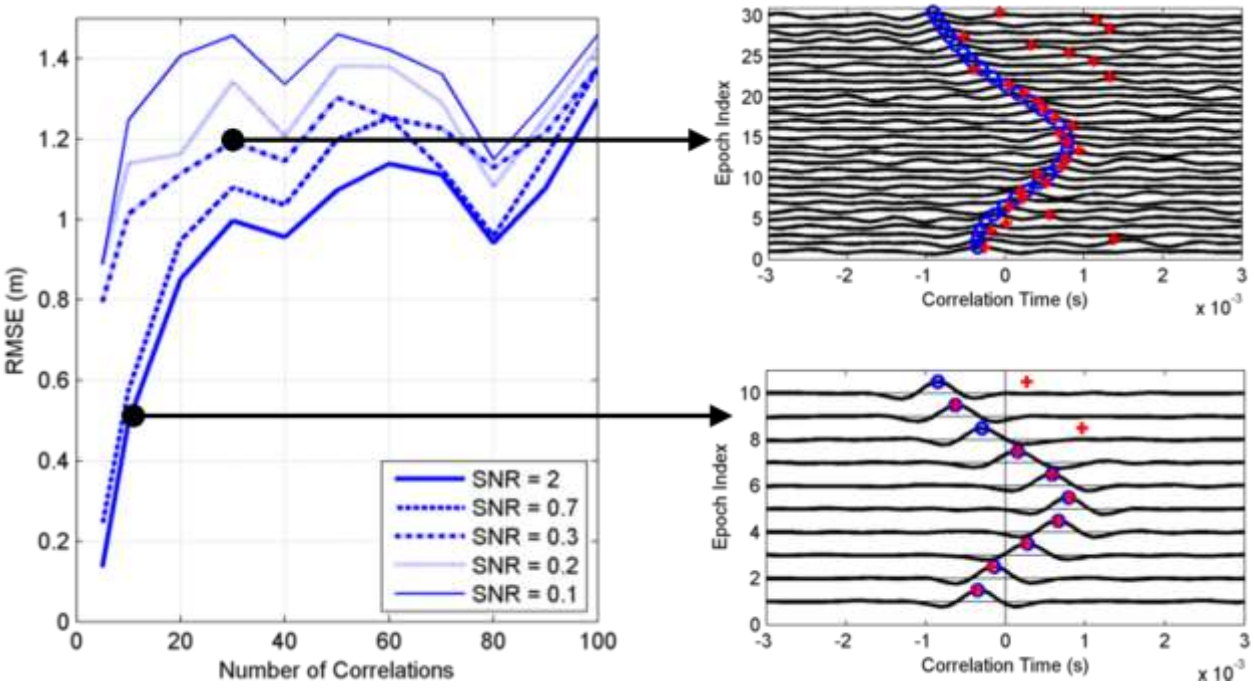


Figure 149: Genetic algorithm RMSE vs total number of correlations (left) for varying base correlation SNR. Two sample solutions are shown for SNR = 0.3 (top right) and SNR = 2 (bottom right), where the true delays are plotted as blue circles and the estimated delays as red crosses.

The results show an increase in the error as the number of correlations increases. This effect is due to the decreased averaging duration, and thus increased noise hiding the peak. It may also be indicative of the algorithm having too many degrees of freedom. This may certainly be the case when using more than 100 correlations, as on several occasions the genetic algorithm reached the 1000 generation limit and was forced to return a solution prematurely. This limitation in estimating many time delays and getting all correct could be remedied in two ways: the first involves filtering the incorrect estimates afterward and the second reduces the number of estimated parameters using a model.

The first method, post-filtering, allows the genetic algorithm to estimate the large number of time delays, fully expecting some or many of them will not have converged on the true solution. Then, a model with a reduced number of parameters is fit to the estimates, and the model which best fits is used as the final solution. However, because of the prevalence of outliers, a simple least-squares fit will generally be quite poor. Instead, RANSAC is used, a method which attempts to find the best fit while ignoring outliers (Fischler & Bolles, 1981). This is achieved by running a model-fit on a small, randomly selected set of points for a fixed number of iterations K. On each iteration, a random subset of N points is selected and the traditional least-squares model obtained for the subset. Then, using a threshold D, the number of inliers is estimated as the number of points with a residual to the model fit of D or less. After all K iterations are run, the iteration with the highest number of inliers (and lowest model fit error in the event of a tie) is assumed to be a decent fit of the data, ignoring outliers. Lastly, the model is fit to the inliers from the winning iteration, again using a traditional least-squares fit. The

RANSAC method relies on the assumption that at least one of the K iterations will select N subset points which are all inliers, and the resulting model-fit will result in the remainder of inliers being identified. Assuming the percentage of points that are inliers is w, the probability of selecting all inliers on at least one iteration is

$$p = 1 - (1 - w^N)^K \tag{127}$$

The probability p is plotted in Figure 150 as a function of w. A nominal choice for K, N is shown in black. Choices of K, N which increase the probability are shown in blue, and choices which decrease the probability are shown in red. It can be seen that decreasing the number of iterations decreases the probability, as expected. Also, choosing more points in the subset selection also reduces the probability. For this reason, the number of subset points N should be as small as possible, but no smaller than the number of model parameters so that a least-squares fit can be performed. Likewise, increasing the number of iterations increases the probability, as does decreasing the number of subset points.

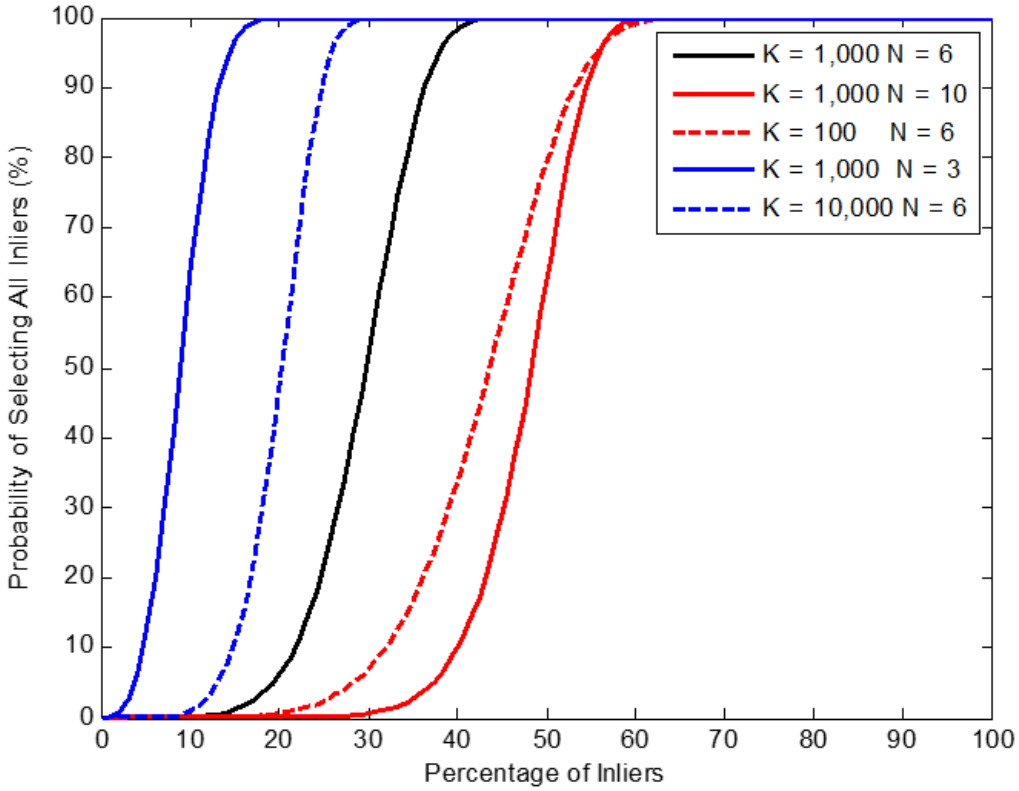


Figure 150: RANSAC performance curves

A takeaway of Figure 150 is that there exists some cutoff inlier percentage where RANSAC is no longer very likely to select all inliers on at least one iteration. For the nominal choice of parameters ($K = 1000, N = 6$), this cutoff is about 40%. Running ten times as many iterations reduces the cutoff to 30%, whereas reducing the number of subset points by half decreases the cutoff to 20%. Likewise, reducing the iterations by a factor of 10 or nearly doubling the number of subset points increases the cutoff to 60%. Since the unconstrained genetic algorithm generally has at least 50% inliers, the nominal parameter set was deemed sufficient.

To filter the unconstrained genetic algorithm results, the RANSAC algorithm was applied using $K = 1000$ iterations with $N = 6$ subset points and $D = 0.5/c$ seconds. The model used for least-squares fitting was given by

$$\hat{\tau}(t) = a_0 + a_1 t + \sum_{n=1}^{4} b_n \sin\left(\frac{i\pi t}{T}\right) \qquad 128$$

where the parameters $a_0, a_1, b_1, b_2, b_3, b_4$ are to be determined by the RANSAC algorithm. The filtered solution is taken to be the function $\hat{\tau}(t)$ with the best-fit parameters of the inliers.

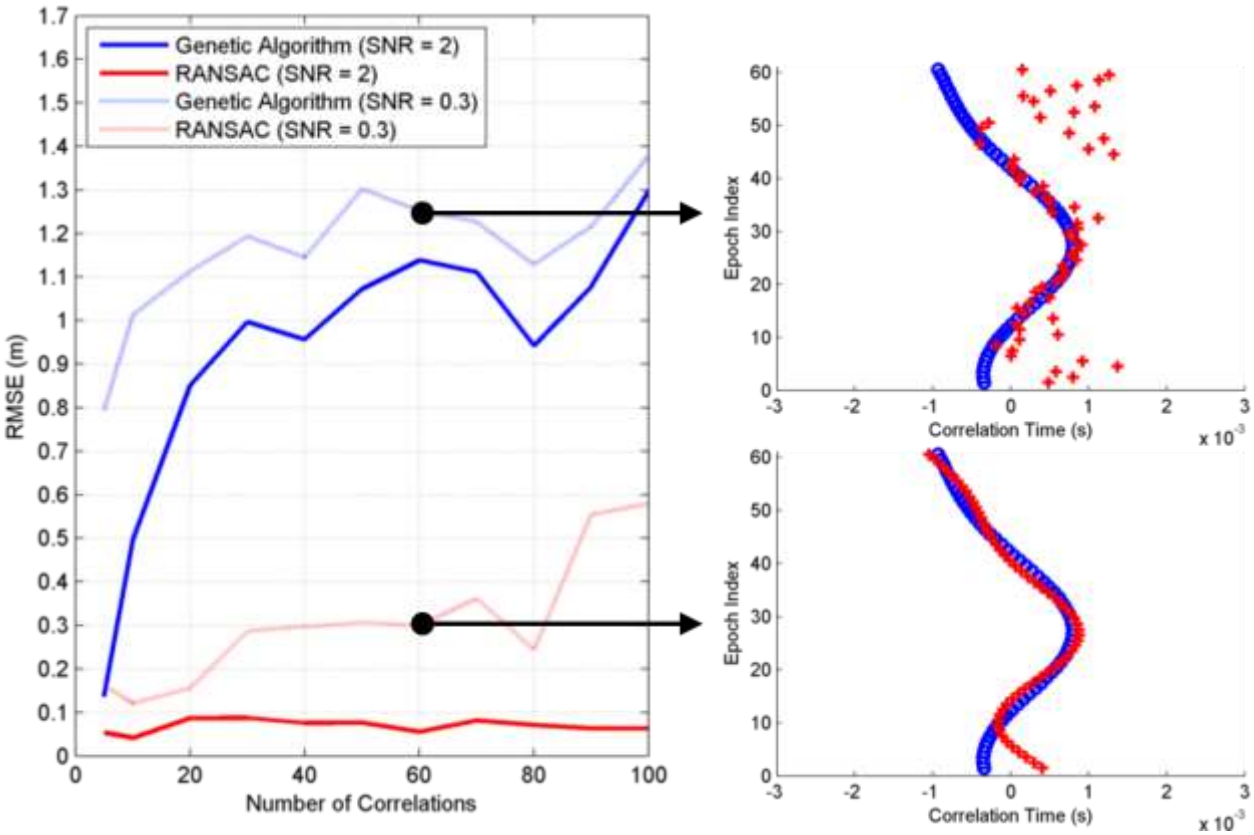


Figure 151: Genetic algorithm RMSE vs total number of correlations (left) for varying base correlation SNR and without (blue) and with (red) RANSAC filtering. Two sample solutions are shown for non-filtered (top right) and RANSAC-filtered (bottom right) for SNR = 0.3, where the true delays are plotted as blue circles and the estimated delays as red crosses. The correlation waveforms are not plotted on the right for clarity.

Shown in Figure 151 is the average RMSE of 20 statistical trials for a selected set of base PSNR's of 2 and 0.3 (3 dB and -5 dB). The genetic algorithm example shows that for 60 estimated time delays, there exist many outliers. A decent number of estimates were also accurate though, but the RMSE of the genetic algorithm suffers regardless. The RANSAC method uses the same estimates produced by the genetic algorithm, but finds the best-fit model (see Equation 128) while ignoring outliers. As seen in the left panel, the RMSE is dramatically decreased, even for the low SNR case. Also, the RMSE is now more consistent across the number of correlations, since the RANSAC model simply needs a small set of correct time delays to obtain a decent best-fit model. One of the drawbacks of the model-based approach such as RANSAC is the ability of a model to have too many degrees of freedom, allowing it to fit points which should be outliers. Likewise, the model could be too simplistic, and not capture the true dynamics of sensor drift accurately.

The second method of correcting error experienced when identifying many time delays is to only allow the stochastic search to estimate a smaller number of model parameters. For example, using the same model from Equation 128, the stochastic search would vary $a_0, a_1, b_1, b_2, b_3, b_4$ and evaluate the time delay model function $\hat{\tau}(t_i)$. Then, it evaluates the objective function identically to before, returning the time delays generated by the optimum parameter set.

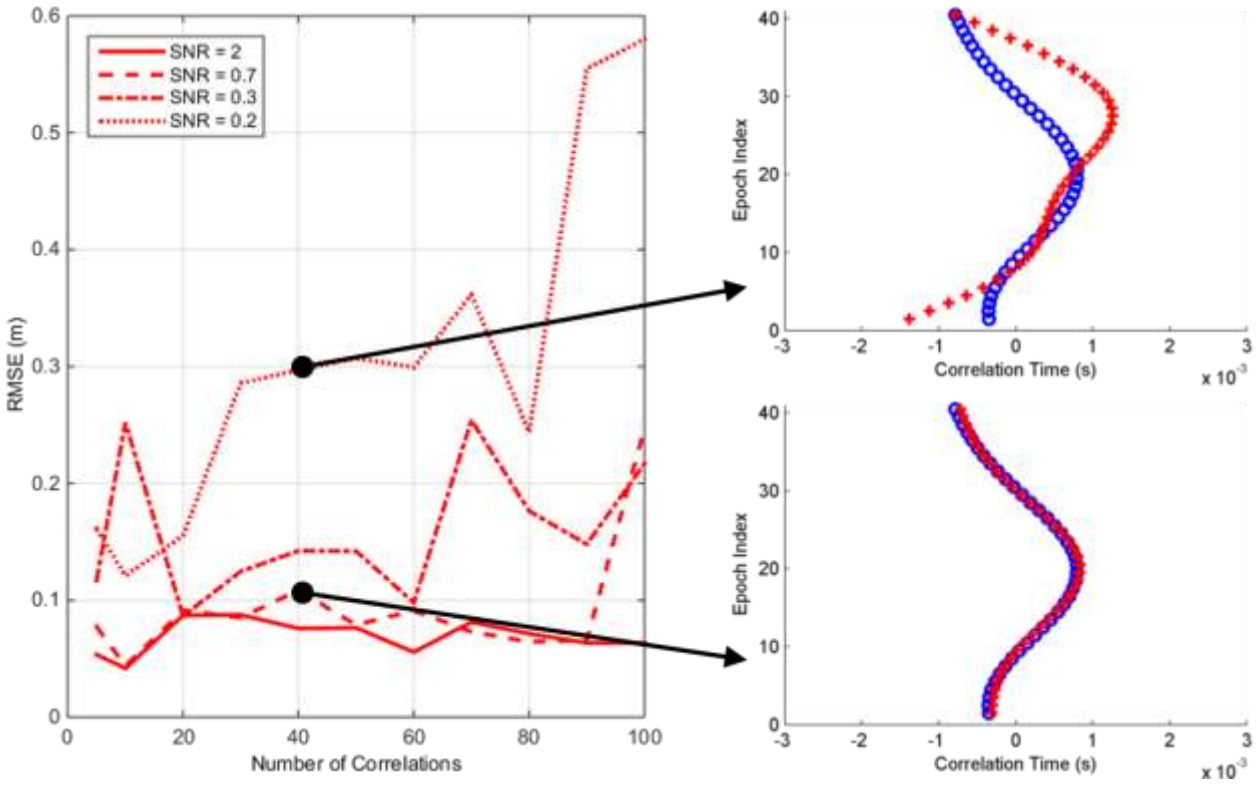


Figure 152: RANSAC-filtered genetic algorithm RMSE vs total number of correlations (left) for varying base correlation SNR. Two sample solutions are shown for SNR = 0.2 (top right) and SNR = 0.7 (bottom right), where the true delays are plotted as blue circles and the estimated delays as red crosses. The correlation waveforms are omitted on the right for clarity.

Figure 152 shows the results of using a genetic algorithm to identify the optimum parameter set of the model in Equation 128. The RMS error of this method is comparable to using RANSAC to post-filter the genetic algorithm results. Both have relatively constant error across the number of averaged correlations. However, the model-based identification suffers when a local minimum is obtained and the resulting model only fits a portion of the true time delays. There obviously exists a better solution which fits all the time delays, but the search was not guaranteed to find such a global optimum. This is a well-known limitation of stochastic search methods, and the post-filtering method handles the problem of local optimum solutions which contain outliers more gracefully.

APPENDIX F
ANALYSIS OF FIRST QUIET TIME, OCTOBER 30, 2016

During the first quiet time, the sensors drifted with a separation distance between 8 and 80 meters, first towards each other, then away. Their separation distance during this time period is plotted in Figure 153. The maximum drift speed seen during this time was approximately 0.05 m/s, or 180 m/hr (note the drift speed does not refer to the speed over ground of the individual floats, but their separation speed).

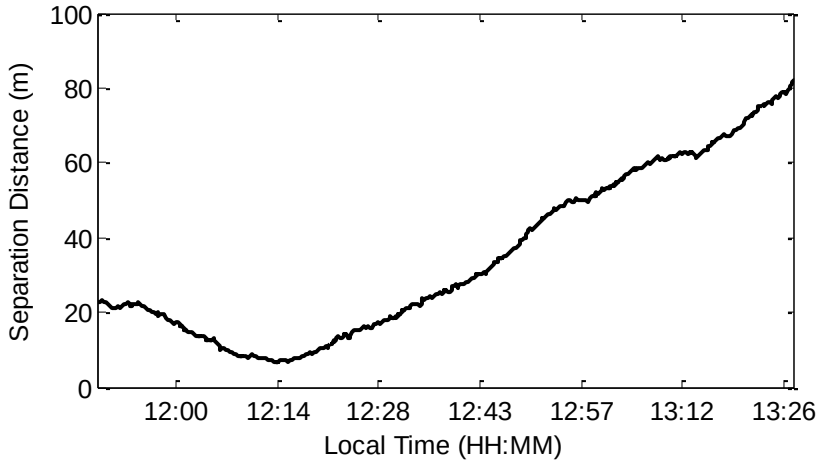


Figure 153: Sensor separation distance for first quiet time

An ensemble of short-time correlations of 1 second length are plotted in Figure 154 and Figure 155 for pressure and axial velocity components, respectively. They are normalized with respect to the total maximum observed throughout the quiet time. The preprocessing steps used in preparing the data for correlation was to first filter the data, then integrate the acceleration and convert to pressure units, then filter, clip, and whiten the resulting data (see Section 2.3.4 for more information on these processes). The bandwidth chosen for the filtering was obtained by time-windowing the full-bandwidth

correlations around the supposed ambient noise correlation peak and plotting the peak's spectrum. The resulting spectrum is nearly flat across 200 – 1500 Hz, with a small decrease in the amplitude of the 800-900 Hz band. Thus, the bandwidth was chosen to be maximally wide, and encompassed 200 – 1500 Hz.

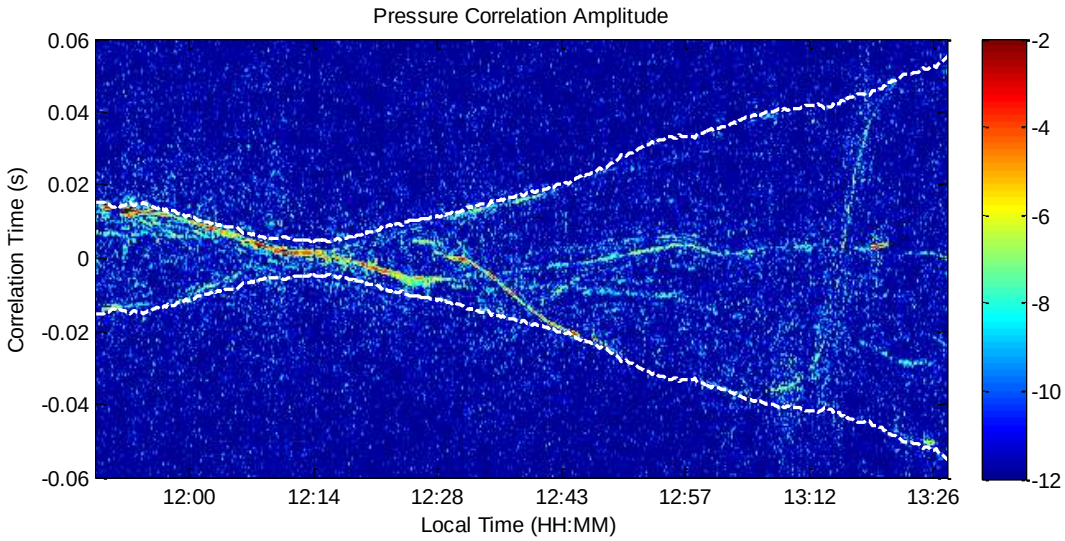


Figure 154: Pressure correlation ensemble amplitudes for quiet time 1. The amplitude is relative to the overall maximum amplitude, and is 10 $\log_{10}$ of the Hilbert transform magnitude.

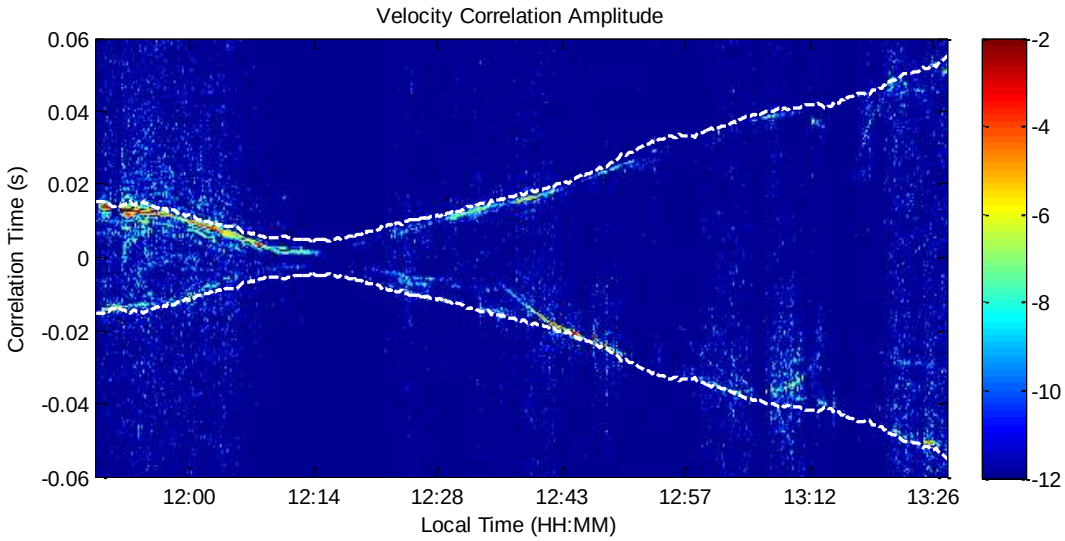


Figure 155: Velocity correlation ensemble amplitudes for quiet time 1. The amplitude is relative to the overall maximum amplitude, and is 10 $\log_{10}$ of the Hilbert transform magnitude.

From the figures, it is clear there were a number of interfering sources, each creating a ridge in the correlation ensemble. For example, the strong ridge seen from the beginning of the quiet time period to about 12:15 was most likely the result of the passenger vessel "Carol Jean", whose GPS location is provided by AIS data. Knowing the difference in distance between the vessel and the array elements, the expected time delay is plotted and nearly perfectly matches up. None of the other tracks appear to have a corresponding AIS vessel responsible, and could have been the result of the many smaller personal craft in the area not required to use AIS. For this reason, ambient noise correlations for this period may be corrupted with interfering noise.

APPENDIX G
AEL JACOBIAN MATRIX

Restating the sensor position vector

$$\vec{\boldsymbol{m}} = [\vec{\boldsymbol{r}}_1^T \quad \vec{\boldsymbol{r}}_2^T \quad \ldots \quad \vec{\boldsymbol{r}}_N^T]^T \tag{129}$$

as presented in Chapter 4, and linearizing the function for inter-sensor distances from sensor locations

$$\vec{\boldsymbol{d}} = D(\vec{\boldsymbol{m}}) \tag{130}$$

about an operating point $\vec{\boldsymbol{m}}_{\boldsymbol{0}}$ yields

$$\vec{\boldsymbol{d}} = D(\vec{\boldsymbol{m}}_{\boldsymbol{0}}) + J\,\overrightarrow{\boldsymbol{\delta m}} \tag{131}$$

where J is the Jacobian matrix, equal to the partial derivatives of the function D with respect to the components of $\vec{\boldsymbol{m}}$. For the distance between sensor i and j, the component of $\vec{\boldsymbol{d}}$ is d_{ij}, and the corresponding row of the Jacobian matrix is

$$J_{ij} = \left[\frac{\partial d_{ij}}{\partial m_1} \quad \frac{\partial d_{ij}}{\partial m_2} \quad \frac{\partial d_{ij}}{\partial m_3} \quad \ldots\right] \tag{132}$$

which has only non-zero partial derivatives for the components of $\vec{\boldsymbol{m}}$ which are part of sensor i or j's positions. Thus, the components of the Jacobian row are

$$J_{ij} = \frac{1}{d_{ij}}\left[(x_i - x_j)(\delta_i - \delta_j) \quad (y_i - y_j)(\delta_i - \delta_j) \quad (z_i - z_j)(\delta_i - \delta_j) \quad \ldots\right] \tag{133}$$

where the distance between the sensors is d_{ij} and δ_i is equal to one in a column which corresponds to a component of sensor *i*, and zero otherwise. The three components in Equation 133 repeat *n* times, where *n* is the number of sensors.

REFERENCES

Argo. (2000). Argo float data and metadata from Global Data Assembly Centre (ARGO GDAC).

Bendat, J. S., & Piersol, A. G. (2010). Random Data: Analysis and Measurement Procedures: Wiley.

Brooks, L. A., & Gerstoft, P. (2009). Green's function approximation from cross-correlations of 20-100 Hz noise during a tropical storm. J Acoust Soc Am, 125(2), 723-734. doi:10.1121/1.3056563

Chen, H. W., & Zhao, J. W. (2004). Wideband MVDR beamforming for acoustic vector sensor linear array. IEE Proceedings - Radar, Sonar and Navigation, 151(3), 158. doi:10.1049/ip-rsn:20040651

Cox, H., & Lai, H. (2009). Simultaneous grating lobe and backlobe rejection with a line array of vector sensors. 1320-1323. doi:10.1109/acssc.2009.5469917

Cox, H., Lai, H., & Bell, K. (2009). Spatial correlation for directional sensors in arbitrary noise fields. 459-463. doi:10.1109/acssc.2009.5469861

Culver, R. L., & Hodgkiss, W. S. (1988). Comparison of Kalman and least squares filters for locating autonomous very low frequency acoustic sensors. IEEE Journal of Oceanic Engineering, 13(4), 282-290. doi:10.1109/48.9241

D'Spain, G. L., Hodgkiss, W. S., & Edmonds, G. L. (1991). The simultaneous measurement of infrasonic acoustic particle velocity and acoustic pressure in the ocean by freely drifting Swallow floats. IEEE Journal of Oceanic Engineering, 16(2), 195-207. doi:10.1109/48.84136

D'Spain, G. L., Hodgkiss, W. S., Edmonds, G. L., Nickles, J. C., Fisher, F. H., & Harriss, R. A. (1992). Initial Analysis Of The Data From The Vertical DIFAR Array. 1, 346-351. doi:10.1109/oceans.1992.612715

D'Spain, G. L., Luby, J. C., Wilson, G. R., & Gramann, R. A. (2006). Vector sensors and vector sensor line arrays: Comments on optimal array gain and detection. J. Acoust. Soc. Am., 120(1), 171. doi:10.1121/1.2207573

Dosso, S. E., Collison, N. E. B., Heard, G. J., & Verrall, R. I. (2004). Experimental validation of regularized array element localization. The Journal of the Acoustical Society of America, 115(5), 2129-2137. doi:10.1121/1.1701897

Dosso, S. E., Fallat, M. R., Sotirin, B. J., & Newton, J. L. (1998). Array element localization for horizontal arrays via Occam’s inversion. J. Acoust. Soc. Am., 104(2), 846-859. doi:10.1121/1.423359

Dosso, S. E. E., Gordon R. (2006). Array element localization accuracy and survey design. Canadian Acoustics, 34(4).

Felisberto, P., Santos, P., & Jesus, S. M. (2010). Tracking Source azimuth Using a Single Vector Sensor. Paper presented at the Conference on Sensor Technologies and Applications.

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6), 381-395. doi:10.1145/358669.358692

Fried, S. E., Kuperman, W. A., Sabra, K. G., & Roux, P. (2008). Extracting the local Green's function on a horizontal array from ambient ocean noise. J Acoust Soc Am, 124(4), EL183-188. doi:10.1121/1.2960937

Godin, O. A., Zabotin, N. A., & Goncharov, V. V. (2010). Ocean tomography with acoustic daylight. Geophysical Research Letters, 37(13), n/a-n/a. doi:10.1029/2010gl043623

Greene, C. R., McLennan, M. W., Norman, R. G., McDonald, T. L., Jakubczak, R. S., & Richardson, W. J. (2004). Directional frequency and recording (DIFAR) sensors in seafloor recorders to locate calling bowhead whales during their fall migration. The Journal of the Acoustical Society of America, 116(2), 799-813. doi:10.1121/1.1765191

Grimes, J. G. (2008). Global Positioning System Standard Positioning Service Performance Standard. http://www.gps.gov/technical/ps/: GPS.gov.

Harrison, C. H., & Siderius, M. (2008). Bottom profiling by correlating beam-steered noise sequences. J. Acoust. Soc. Am., 123(3), 1282-1296. doi:10.1121/1.2835416

Hawkes, M., & Nehorai, A. (1998). Acoustic vector-sensor beamforming and Capon direction estimation. IEEE Transactions on Signal Processing, 46(9), 2291-2304. doi:10.1109/78.709509

Hawkes, M., & Nehorai, A. (1999). Effects of sensor placement on acoustic vector-sensor array performance. IEEE Journal of Oceanic Engineering, 24(1), 33-40. doi:10.1109/48.740154

Hawkes, M., & Nehorai, A. (2000). Acoustic vector-sensor processing in the presence of a reflecting boundary. IEEE Transactions on Signal Processing, 48(11), 2981-2993. doi:10.1109/78.875455

Hawkes, M., & Nehorai, A. (2001). Acoustic vector-sensor correlations in ambient noise. IEEE Journal of Oceanic Engineering, 26(3), 337-347. doi:10.1109/48.946508

Hawkes, M., & Nehorai, A. (2003). Wideband source localization using a distributed acoustic vector-sensor array. IEEE Transactions on Signal Processing, 51(6), 1479-1491. doi:10.1109/tsp.2003.811225

Hodgkiss, W., & Anderson, V. (1983). Acoustic positioning for an array of freely drifting infrasonic sensors. IEEE Journal of Oceanic Engineering, 8(3), 116-119. doi:10.1109/joe.1983.1145567

Konvalin, C. (2008). Compensating for Tilt, Hard Iron and Soft Iron Effects. Retrieved from

Lani, S. W., Sabra, K. G., Hodgkiss, W. S., Kuperman, W. A., & Roux, P. (2013). Coherent processing of shipping noise for ocean monitoring. J. Acoust. Soc. Am., 133(2), EL108-113. doi:10.1121/1.4776775

Lauchle, G. C., Wang, J., & Howe, M. S. (2002). Flow-induced noise on underwater pressure-vector acoustic sensors. 1906-1910. doi:10.1109/oceans.2002.1191921

Leroy, C., Lani, S., Sabra, K. G., Hodgkiss, W. S., Kuperman, W. A., & Roux, P. (2012). Enhancing the emergence rate of coherent wavefronts from ocean ambient noise correlations using spatio-temporal filters. J Acoust Soc Am, 132(2), 883-893. doi:10.1121/1.4731231

McEachern, J., McConnell, J., Jamieson, J., & Trivett, D. (2006). ARAP - Deep Ocean Vector Sensor Research Array. 1-5. doi:10.1109/oceans.2006.307082

Meggitt, I. Advanced Sensing Technologies for maritime monitoring and seismic sensing. Retrieved from https://wilcoxon.com/industries-and-applications/defense/

Mitchell, M. (1999). An Introduction to Genetic Algorithms (5th ed.): The MIT Press.

Morley, M. G., Dosso, S. E., & Chapman, N. R. (2009). Array element localization using ship noise. J. Acoust. Soc. Am., 125(3), 1403-1409. doi:10.1121/1.3076201

Naughton, P., Roux, P., Yeakle, R., Schurgers, C., Kastner, R., Jaffe, J. S., & Roberts, P. L. (2016). Ambient noise correlations on a mobile, deformable array. J Acoust Soc Am, 140(6), 4260. doi:10.1121/1.4971172

Nehorai, A., & Paldi, E. (1994). Acoustic vector-sensor array processing. IEEE Transactions on Signal Processing, 42(9), 2481-2491. doi:10.1109/78.317869

Nichols, B., & Sabra, K. G. (2015). Cross-coherent vector sensor processing for spatially distributed glider networks. J Acoust Soc Am, 138(3), EL329-335. doi:10.1121/1.4929615

Paull, L., Saeedi, S., Seto, M., & Li, H. (2014). AUV Navigation and Localization: A Review. IEEE Journal of Oceanic Engineering, 39(1), 131-149. doi:10.1109/joe.2013.2278891

Poulsen, A. J. (2009). Robust vector sensor array processing and performance analysis. (Ph.D.), Massachusetts Institute of Technology.

Rafaely, B. (2000). Spatial-temporal correlation of a diffuse sound field. The Journal of the Acoustical Society of America, 107(6), 3254-3258. doi:10.1121/1.429397

Raveon. RV-M7 Data Radio Modem Technical Overview. Retrieved from http://www.raveon.com/manual-rv-m7-specs.html

Roux, P., & Kuperman, W. A. (2004). Extracting coherent wave fronts from acoustic ambient noise in the ocean. The Journal of the Acoustical Society of America, 116(4), 1995-2003. doi:10.1121/1.1797754

Roux, P., Sabra, K. G., Kuperman, W. A., & Roux, A. (2005). Ambient noise cross correlation in free space: Theoretical approach. The Journal of the Acoustical Society of America, 117(1), 79-84. doi:10.1121/1.1830673

Sabra, K. G., Roux, P., & Kuperman, W. A. (2005a). Arrival-time structure of the time-averaged ambient noise cross-correlation function in an oceanic waveguide. The Journal of the Acoustical Society of America, 117(1), 164-174. doi:10.1121/1.1835507

Sabra, K. G., Roux, P., & Kuperman, W. A. (2005b). Emergence rate of the time-domain Green's function from the ambient noise cross-correlation function. J. Acoust. Soc. Am., 118(6), 3524. doi:10.1121/1.2109059

Sabra, K. G., Roux, P., Thode, A. M., D'Spain, G. L., Hodgkiss, W. S., & Kuperman, W. A. (2005). Using Ocean Ambient Noise for Array Self-Localization and Self-Synchronization. IEEE Journal of Oceanic Engineering, 30(2), 338-347. doi:10.1109/joe.2005.850908

Santos, P., Felisberto, P., & Jesus, S. M. (2010). Vector Sensor Arrays in Underwater Acoustic Applications.

Schultheiss, P. M. (1980). Optimum range and bearing estimation with randomly perturbed arrays. The Journal of the Acoustical Society of America, 68(1), 167. doi:10.1121/1.384616

Shipps, J. C., & Deng, K. (2003). A miniature vector sensor for line array applications. Paper presented at the OCEANS 2003.

Shipps, J. C. A., B.M. (2004). The use of vector sensors for underwater port and waterway security. Paper presented at the Sensors for Industry Conference.

Siderius, M., Harrison, C. H., & Porter, M. B. (2006). A passive fathometer technique for imaging seabed layering using ambient noise. J. Acoust. Soc. Am., 120(3), 1315-1323. doi:10.1121/1.2227371

Siderius, M., Song, H., Gerstoft, P., Hodgkiss, W. S., Hursky, P., & Harrison, C. (2010). Adaptive passive fathometer processing. J. Acoust. Soc. Am., 127(4), 2193-2200. doi:10.1121/1.3303985

Smith, K. B., & van Leijen, A. V. (2007). Steering vector sensor array elements with linear cardioids and nonlinear hippioids. J. Acoust. Soc. Am., 122(1), 370-377. doi:10.1121/1.2722054

SparkFun. SparkFun OpenLog. Retrieved from https://www.sparkfun.com/products/13712

Swartz, S. L. C., Tim; McDonald, Mark A.; Hildebrand, John A.; Oleson, Erin M.; Martinez, Anthony; Clapham, Phillip J.; Barlow, Jay; Jones, Mary Lou. (2003). Acoustic and Visual Survey of Humpback Whale (Megaptera novaeangliae) Distribution in the Eastern and Southeastern Caribbean Sea. Caribbean Journal of Science, 39(2), 195-208.

TASCAM. DR-680 Overview. Retrieved from http://tascam.com/product/dr-680/

Thode, A., Skinner, J., Scott, P., Roswell, J., Straley, J., & Folkert, K. (2010). Tracking sperm whales with a towed acoustic vector sensor. J. Acoust. Soc. Am., 128(5), 2681-2694. doi:10.1121/1.3495945

Van Trees, H. L. (2004). Detection, Estimation and Modulation Theory, Part IV, Optimum Array Processing: John Wiley & Sons.

Woolfe, K. F., Lani, S., Sabra, K. G., & Kuperman, W. A. (2015). Monitoring deep-ocean temperatures using acoustic ambient noise. Geophysical Research Letters, 42(8), 2878-2884. doi:10.1002/2015gl063438

Woolfe, K. F., & Sabra, K. G. (2015). Variability of the coherent arrivals extracted from low-frequency deep-ocean ambient noise correlations. J. Acoust. Soc. Am., 138(2), 521-532. doi:10.1121/1.4923447

Woolfe, K. F., Sabra, K. G., & Kuperman, W. A. (2015). Optimized extraction of coherent arrivals from ambient noise correlations in a rapidly fluctuating medium. J. Acoust. Soc. Am., 138(4), EL375-381. doi:10.1121/1.4931829

Worcester, P. F., Cornuelle, B. D., Dzieciuch, M. A., Munk, W. H., Howe, B. M., Mercer, J. A., . . . Baggeroer, A. B. (1999). A test of basin-scale acoustic thermometry using a large-aperture vertical array at 3250-km range in the eastern North Pacific Ocean. The Journal of the Acoustical Society of America, 105(6), 3185-3201. doi:10.1121/1.424649