NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

INVESTIGATION OF ACOUSTIC VECTOR SENSOR DATA PROCESSING IN THE PRESENCE OF HIGHLY VARIABLE BATHYMETRY

by

Timothy D. Kubisak

June 2014

Thesis Advisor:	Kevin B. Smith
Second Reader:	Daphne Kapolka

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704–0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 2014	Master's Thesis

4. TITLE AND SUBTITLE
INVESTIGATION OF ACOUSTIC VECTOR SENSOR DATA PROCESSING IN THE PRESENCE OF HIGHLY VARIABLE BATHYMETRY

5. FUNDING NUMBERS

6. AUTHOR(S) Timothy D. Kubisak

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research, Code 321MS/322OA, 875 N. Randolph St., Arlington, VA 22203.

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

Data has been collected on acoustic vector sensors mounted on autonomous underwater gliders in the Monterey Bay during 2012–2013. Previous processing work computed the acoustic vector intensity to estimate bearing to impulsive sources of interest.

These sources included small explosive shots deployed by local fishermen and humpback whale vocalizations. While the highly impulsive shot data produced unambiguous bearing estimations, the longer duration whale vocalizations showed a fairly wide spread in bearing.

In this work, causes of the ambiguity in bearing estimation are investigated in the context of the highly variable bathymetry of the Monterey Bay Canyon, as well as the coherent multipath interference in the longer duration calls.

Sound speed data collected during the previous experimental effort, along with a three-dimensional bathymetric relief of the Monterey Bay Canyon, are incorporated into a three-dimensional version of the Monterey-Miami Parabolic Equation Model. Propagation results are computed over a frequency band from 336–464 Hz in order to provide predictions of pulse arrival structure. This data is analyzed using conventional pressure plane-wave beamforming techniques in order to highlight horizontal coupling caused by the canyon bathymetry. The data is also analyzed using the previously developed acoustic vector intensity processing string and shown to exhibit a qualitatively similar spread in the estimated bearing.

14. SUBJECT TERMS Acoustic vector sensors, bearing estimation, intensity processing, parabolic modeling, three dimensional propagation, unmanned underwater vehicles, UUV, vector intensity

15. NUMBER OF PAGES
83

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540–01-280-5500

Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

INVESTIGATION OF ACOUSTIC VECTOR SENSOR DATA PROCESSING IN THE PRESENCE OF HIGHLY VARIABLE BATHYMETRY

Timothy D. Kubisak
Lieutenant, United States Navy
B.S., The Citadel, 2007

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN APPLIED PHYSICS

from the

NAVAL POSTGRADUATE SCHOOL
June 2014

Author: Timothy D. Kubisak

Approved by: Kevin B. Smith
Thesis Advisor

Daphne Kapolka
Second Reader

Andres Larraza
Chair, Department of Physics

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Data has been collected on acoustic vector sensors mounted on autonomous underwater gliders in the Monterey Bay during 2012–2013. Previous processing work computed the acoustic vector intensity to estimate bearing to impulsive sources of interest.

These sources included small explosive shots deployed by local fishermen and humpback whale vocalizations. While the highly impulsive shot data produced unambiguous bearing estimations, the longer duration whale vocalizations showed a fairly wide spread in bearing.

In this work, causes of the ambiguity in bearing estimation are investigated in the context of the highly variable bathymetry of the Monterey Bay Canyon, as well as the coherent multipath interference in the longer duration calls.

Sound speed data collected during the previous experimental effort, along with a three-dimensional bathymetric relief of the Monterey Bay Canyon, are incorporated into a three-dimensional version of the Monterey-Miami Parabolic Equation Model. Propagation results are computed over a frequency band from 336–464 Hz in order to provide predictions of pulse arrival structure. This data is analyzed using conventional pressure plane-wave beamforming techniques in order to highlight horizontal coupling caused by the canyon bathymetry. The data is also analyzed using the previously developed acoustic vector intensity processing string and shown to exhibit a qualitatively similar spread in the estimated bearing.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

3D	three-dimensional
DOA	direction of arrival
FT	Fourier transform
FFT	fast Fourier transform
MMPE	Monterey-Miami Parabolic Equation
PE	parabolic equation
SNR	signal-to-noise ratio
SSF	split-step Fourier
TL	transmission loss
UNCA	uncoupled azimuth
USW	undersea warfare
UUV	unmanned underwater vehicle

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

Professor Kevin Smith, I am so deeply thankful for the knowledge, wisdom, and insight that you have shared with me during the process of developing this thesis. I was fortunate to have learned from you in the classroom, in the lab, and even out to sea during my time at NPS. Humbled by your expertise and motivated by your drive to advance this research, I stepped out of my own comfort zone and gained a greater understanding of the underwater environment. Thank you for the commitment you have graciously shown to me and to my classmates.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Passive detection of underwater acoustic signals is critical to maintaining dominance in the Undersea Warfare (USW) domain. As the Navy looks to develop and employ new technologies, the acoustic vector sensor has been utilized in recent research. The research conducted for this paper is based on the data processing of an acoustic vector sensor with an omnidirectional pressure sensor and three orthogonal accelerometers.

A typical towed array is a straight line array that uses multiple, spatially distributed omnidirectional pressure sensors to determine the direction of arrival (DOA) for an acoustic source. The straight line array is known to be limited by its broadside port/starboard ambiguity and ambiguity from its front and back lobes [1]. An array of acoustic vector sensors, however, does not experience this ambiguity. An additional advantage of the acoustic vector sensor is that a single sensor can determine DOA while providing a directivity gain equivalent to that of a four-element line array [1].

Single sensor operation is ideal in littoral regions where deployment of straight line arrays is impractical. Using acoustic vector sensors deployed on small unmanned underwater vehicles (UUVs), such as the Exocetus Littoral Glider, the Navy could perform local area monitoring for the presence of marine mammals, unfriendly tonals, or other signals of interest. Multiple UUVs deployed in the same operating area would be able to triangulate the location of an acoustic source.

This paper demonstrates the ability to determine DOA for different broadband sources in the presence of highly variable bathymetry. In support of future experimentation, the effects of horizontal and vertical multipath interference are addressed to determine if sources with broader bandwidth or tonals of shorter duration are required to improve source bearing estimation.

The remainder of this paper is organized as follows:

- Chapter II: Theoretical background and previous research
- Chapter III: Three-Dimensional Monterey-Miami Parabolic Equation Model
- Chapter IV: Environmental description and data processing
- Chapter V: Results and analysis
- Chapter VI: Conclusions and recommendations

II. BACKGROUND

DOA can be determined by combining the pressure and the particle velocity of an acoustic wave. Leveraging the relationships between these two measurements, intensity processing is an incoherent processing approach that can be used to estimate the direction of the source of interest [2]. While intensity processing does not provide the gain that conventional additive or adaptive processing yield, it is a very simple technique for combining acoustic vector sensor data into a directional vector.

A. ACOUSTIC VECTOR FIELDS

1. Euler's Equation

For a fluid element with differential volume $dV = dxdydz$ and mass dm that is acted upon by a force f in the x-direction, Newton's Second Law is applied in Equation (2.1) and the force experienced by the fluid element is shown in Equation (2.2). p represents acoustic pressure.

$$d\vec{f} = \vec{a}dm, \tag{2.1}$$

$$d\vec{f} = \left[p - \left(p + \frac{\partial p}{\partial x}\mathrm{d}x \right) \right] dydz \tag{2.2}$$

In three dimensions, we can formulate equations for the y- and z-directions and combine them as shown in Equation (2.3).

$$d\vec{f} = -\nabla p dV \tag{2.3}$$

From the velocity defined in Equation (2.4), the fluid element's acceleration, a, is expressed in Equation (2.5) which then reduces to Equation (2.6).

$$\vec{u} = \frac{d}{dt}(x\hat{i} + y\hat{j} + z\hat{k}) \tag{2.4}$$

$$\vec{a} = \frac{\partial \vec{u}}{\partial t} + u_x \frac{\partial \vec{u}}{\partial x} + u_y \frac{\partial \vec{u}}{\partial y} + u_z \frac{\partial \vec{u}}{\partial z} \tag{2.5}$$

$$\vec{a} = \frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla)\vec{u} \tag{2.6}$$

In small amplitude acoustic processes, the assumption that $\left|(\vec{u} \cdot \nabla)\vec{u}\right| << \left|\frac{\partial \vec{u}}{\partial t}\right|$ is applied. Following the preceding assumption, and with ρ as the density of the fluid, ρdV is substituted for dm. The combination of Equations (2.1), (2.3), and (2.6) produces the linear Euler's equation in Equation (2.7) [3].

$$\rho_0 \frac{\partial \vec{u}}{\partial t} = -\nabla p \tag{2.7}$$

2. Linear Wave Equation

For the same element used to derive Euler's Equation (2.7), conservation of mass requires the fluid flow rate into the volume dV must equal the rate at which the mass dm increases. The Mass Continuity Equation is formed to first order and shown in Equation (2.8).

$$\frac{\partial \rho}{\partial t} = -\rho \nabla \cdot \vec{u} \tag{2.8}$$

The divergence of Euler's Equation (2.7) is combined with the time derivative of the Mass Continuity Equation (2.8) to resolve the differential relationship of pressure pof and density ρ. This relationship is shown in Equation (2.9).

$$\nabla^2 p = \frac{\partial^2 \rho}{\partial t^2} \tag{2.9}$$

For small perturbations, Equation (2.9) is rewritten with a single independent variable in Equation (2.10) where c is the speed of sound in water. This equation is known as the linear wave equation.

$$\nabla^2 p = \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2} \tag{2.10}$$

3. Acoustic Intensity

From Kinsler, et al., the instantaneous intensity associated with a sound wave can be described as the "instantaneous rate per unit area at which work is done by one fluid element on an adjacent element" [3]. The energy flux in the direction of sound propagation is found from the time average of the instantaneous intensity. Thus, the intensity vector yields a magnitude and direction for the associated wave.

With acceleration data provided by the vector sensors, it is necessary to expand Equation (2.6) and apply the relationship between the complex acceleration and particle velocity as shown in Equation (2.11).

$$\frac{d}{dt}\left[\vec{\hat{u}}(t)\right] = \vec{\hat{a}}(t) \Rightarrow \vec{\hat{u}}(t) = \frac{i}{\omega}\vec{\hat{a}}(t) \tag{2.11}$$

After taking the dot product of Euler's Equation (2.7) with the complex conjugate of the particle velocity, $\vec{u}^*$ in Equation (2.12) and combining the result with the conjugate of the mass continuity equation (2.8) multiplied by the complex pressure as shown in Equation (2.13), the differential of Equation (2.14) can be found.

$$\rho_0 \vec{u}\frac{\partial u}{\partial t} = -\vec{u}^*\nabla\hat{p} \tag{2.12}$$

$$\frac{1}{\rho_0 c^2}\hat{p}\frac{\partial p^*}{\partial t} = -\hat{p}\nabla\vec{u}^* \tag{2.13}$$

$$\frac{\partial}{\partial t}\left(\frac{1}{2}\rho_0\hat{u}^2 + \frac{1}{2}\frac{\hat{p}^2}{\rho_0 c^2}\right) + \nabla\left(\hat{p}\vec{u}^*\right) = 0 \tag{2.14}$$

The rate of change in the energy density of the acoustic field is given by the first term of Equation (2.14). The energy density is composed of the kinetic energy due to fluid flow and the potential energy due to pressure perturbations. The acoustic energy flow is given by the second term of Equation (2.14) and defines the acoustic intensity vector.

From the acoustic energy flow given by the second term in Equation (2.14), the acoustic intensity is shown in Equation (2.15).

$$\vec{I}_c = \frac{1}{2} p\vec{u}^* \tag{2.15}$$

B. INTENSITY PROCESSING

Intensity processing is a common method for determining DOA using individual vector sensors. Formally, we measure the real, instantaneous acoustic intensity as defined in Equation (2.16).

$$\vec{I}(t) = \mathrm{Re}\left\{\hat{p}(t)\right\} \cdot \mathrm{Re}\left\{\vec{\hat{u}}(t)\right\} \tag{2.16}$$

To resolve DOA, the time average of the instantaneous acoustic intensity must be found. It can be shown, however, that the real portion of the complex acoustic intensity in Equation (2.15) yields the time average of the instantaneous intensity in Equation (2.16) [4].

The real portion of the complex acoustic intensity is a vector normal to constant phase surfaces while the imaginary portion is a vector normal to constant pressure surfaces. Also known as reactive intensity, the imaginary portion goes to zero when it is averaged over time [4]. Therefore, only the real portion is used to determine DOA.

C. PREVIOUS RESEARCH

In March 2012 and September 2013, the Naval Postgraduate School deployed a UUV with an integrated acoustic vector sensor to observe ambient noise conditions in the Monterey Bay area of California. The vector sensor provided four channels of data, one from the omnidirectional hydrophone and one each from the three orthogonal accelerometers. The acoustic channels were sampled at 39.0625 kHz. Numerous broadband signals were collected, including boat noise, marine mammal vocalizations, and even several impulsive signals that were transmitted from “seal bombs” used by local fisherman to discourage harbor seals and sea lions from taking their catch. These “seal bombs” are referred to as “shots” in this paper. Figure 1 shows a UUV (LG-16) being deployed from the R/V Fulmar during the data acquisition for this paper.

Figure 1. Glider deployment.

In 2013, research by LT James Upshaw processed numerous signals using two different approaches to determine DOA [5]. The research for this paper was prompted by the results from the intensity processing which revealed interesting behavior for different noise sources. Short duration shots recorded in 2012 were analyzed along with humpback whale vocalizations recorded in 2013. Figure 2 shows a significant difference in the signals when analyzing their respective spectrograms. The spectrogram for a string of five whale vocalizations displayed in the left panel shows that the signals had a bandwidth of approximately 450–550 Hz, duration of about 1 sec, and a signal-to-noise ratio (SNR) of only 10 dB at best. The spectrogram for two shots displayed in the right panel shows broadband signals covering the entire processing band (350-650 Hz), short duration of approximately 50 msec, and SNRs of about 30 dB.

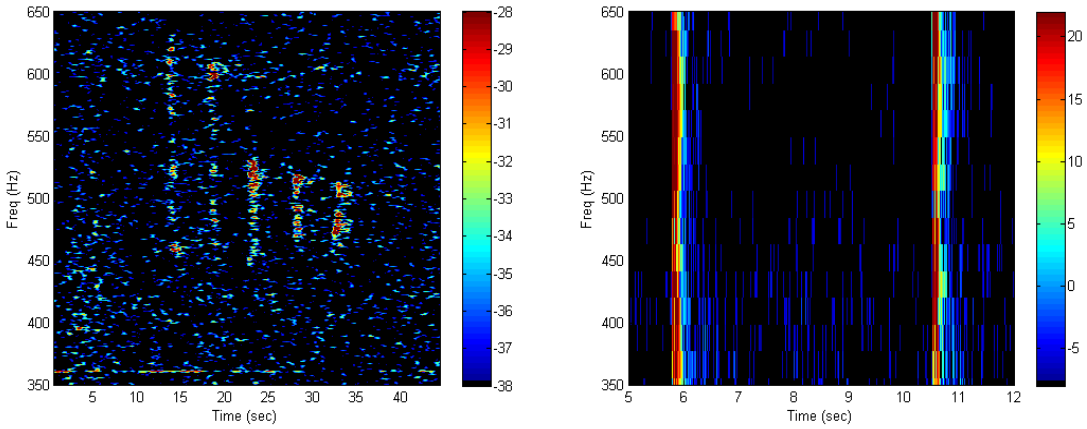

Figure 2. Pressure spectrograms of impulsive signals; five whale vocalizations (left panel) and two explosive shots (right panel), after [5].

To examine the vector intensity bearing estimation, a bubble plot was created in MATLAB as a function of time that displayed the relative amplitude of the intensity vector proportional to the bubble size, centered at the bearing of the signal relative to the glider's position. These plots are shown in Figure 3 with the whale vocalizations again in the left panel and the shots in the right panel.

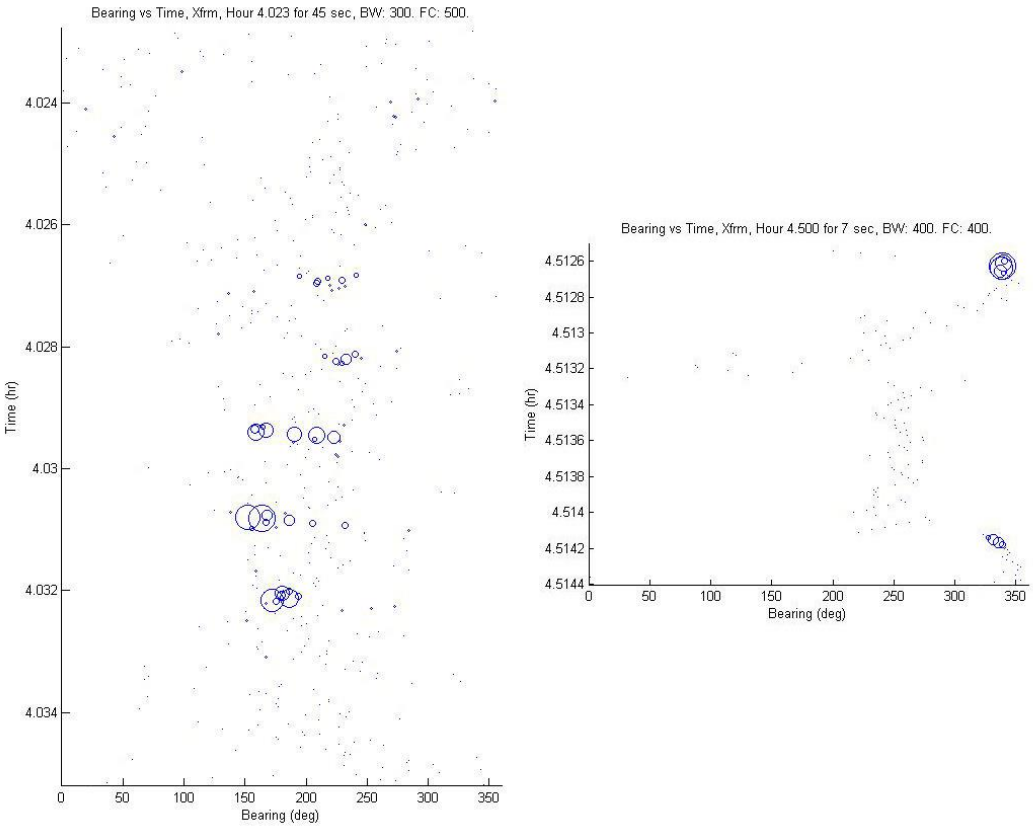

Figure 3. Bubble plots of intensity vector response; 5 whale vocalizations (left panel) and 2 explosive shots (right panel), after [5].

As shown in Figure 3, the bearing estimation of the 5 whale vocalizations showed a directional ambiguity of 40 degrees or larger. The shot data, however, showed unambiguous bearing estimations with a spread of 10 degrees or less. The lower SNR of the whale vocalizations may be a factor, but examination of the bearing ambiguity showed that the evolution of the intensity vector was not random. The bearing ambiguity seemed to sweep continuously over the range of bearing uncertainty. This suggests that the ambiguity could be caused by something fundamental in the signal propagation.

After displaying the glider locations at the time of the data recording and illustrating the general directions toward the sources, as shown in Figure 4, it was noted that there was a significant difference in the bathymetry along the propagation paths. The whale vocalizations were transmitted across the highly variable bathymetry of the Monterey Bay Canyon while the shots were transmitted across a relatively stable shelf

region to the north of the canyon. The impact of this 3-dimensional (3D) variable bathymetry, which may be combined with the effects of more narrow band or longer duration signals, are potential causes of the horizontal bearing ambiguity.

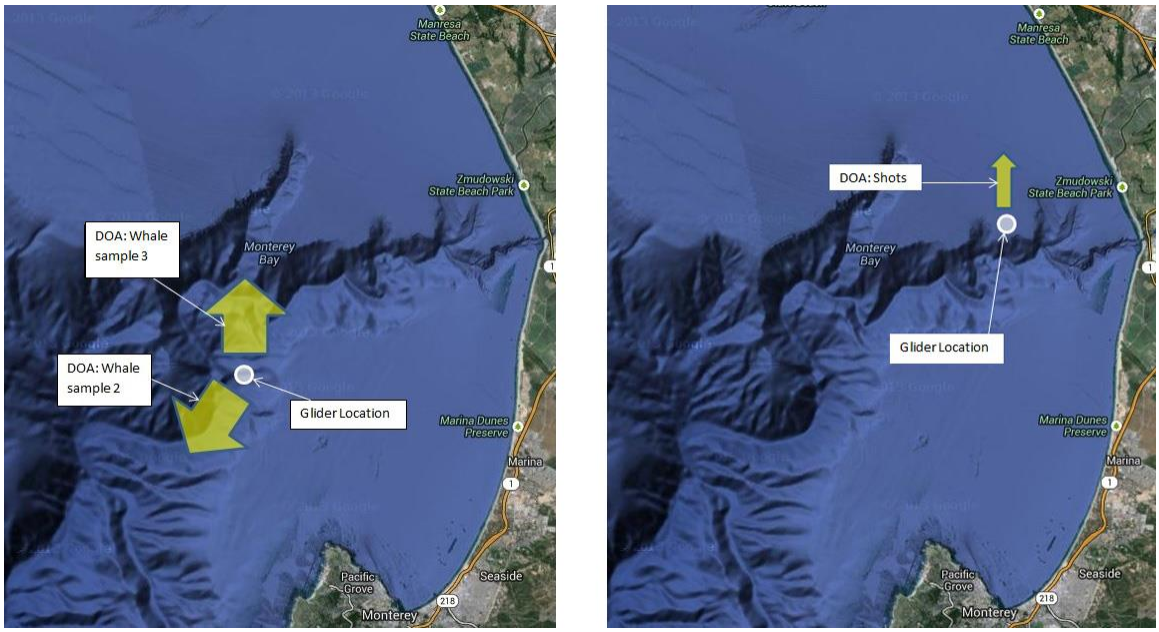

Figure 4. Geospatial direction of intensity vector bearing results relative to glider position in Monterey Bay; 5 whale vocalizations (left panel) and 2 explosive shots (right panel), after [5].

D. DATA PROCESSING

The intensity processing for this paper was conducted by modifying the method employed during LT Upshaw's research. The signals of interest in this case were centered at 400 Hz with a bandwidth of 127.75 Hz. All four acoustic signals (pressure and particle velocity (3)) were simulated using the Monterey-Miami Parabolic Equation (MMPE) Model at NPS. This model is described in more detail in a subsequent section. In order to facilitate the use of LT Upshaw's previous processing string, the MMPE model data was extracted at various spatial locations of interest and processed to produce arrival time for different signal types (i.e., impulsive or chirp-like signals). The data was then treated as measured sensor data to determine acoustic intensity vectors.

All four acoustic channels were transformed from the time domain to the frequency domain using a fast Fourier Transform (FFT) and any data outside the frequency band would then be discarded. This transformation was performed over a user defined Hanning window (research for LT Upshaw's paper used a 0.3 second processing window and 0.1 second overlap time) that advanced incrementally throughout the sample being analyzed.

For each time step, the intensity vector's x-, y-, and z-components were calculated along with the magnitude of the vector. Prior to calculating the magnitude, each vector component was averaged over the frequency bandwidth being analyzed. The calculations for the vector components and magnitude were then completed with Equation (2.17) and (2.18), respectively.

$$I_{x,y,z} = \mathrm{Re}\left\{\frac{1}{2}\,\hat{p}u^{*}_{x,y,z}\right\} \tag{2.17}$$

$$\left|\vec{I}\right| = \sqrt{I_x^2 + I_y^2 + I_z^2} \tag{2.18}$$

The complex pressure and particle velocity were derived from the frequency domain response.

Next, the horizontal bearing angle of the intensity vector was. Since the intensity vector points in the direction of wave propagation, the bearing was rotated 180 degrees to point in the DOA. The DOA was then plotted on a bearing versus time plot using markers that were sized proportional to the magnitude of the intensity for each time step. The scaling of the marker size was determined by the ratio of the intensity magnitude for each time step to the maximum magnitude for the sample. A sensitivity factor could also be adjusted by the user to enhance or reduce the scaling.

THIS PAGE INTENTIONALLY LEFT BLANK

III. 3D MONTEREY-MIAMI PARABOLIC EQUATION MODEL

This section has been adapted from reports by Smith [6], Smith and Colosi [7], and Smith and Tappert [8]. The following sub-sections describe the fundamentals of the MMPE Model approximation and its implementation.

A. PARABOLIC EQUATION MODEL

To begin, the time-harmonic acoustic field is represented in a Cartesian coordinate system (*x*,*y*,*z*) in Equation (3.1) where *P* is the pressure in the time-domian and *p* represents the spatially varying pressure.

$$P(x, y, z, \omega t) = p(x, y, z)e^{-i\omega t} \tag{3.1}$$

Substituting Equation (3.1) into the linear wave equation (2.9) gives the Helmholtz equation in Equation (3.2).

$$\nabla^2 p(x, y, z) + k_0^2 n^2(x, y, z)\, p(x, y, z) = 0 \tag{3.2}$$

The reference wavenumber $k_0 = \frac{\omega}{c_0}$, the index of refraction is n, and the reference sound speed is c_0. The starting field is modeled as a point source at coordinates $(x = 0, y = 0, z = z_s)$ with reference source level P_0 defined as the pressure amplitude at a distance of $R_0 = 1m$.

By introducing the operator notation in Equations (3.3) through Equation (3.7), the homogeneous form of the Helmholtz equation becomes Equation (3.8).

$$P_{op} = \frac{\partial}{\partial x} \tag{3.3}$$

$$Q_{op} = \sqrt{(\mu + \varepsilon + \nu + 1)} \tag{3.4}$$

$$\varepsilon = n^2 - 1 \tag{3.5}$$

$$\mu = \frac{1}{k_0^2}\frac{\partial^2}{\partial z^2} \tag{3.6}$$

$$\nu = \frac{1}{k_0^2}\frac{\partial^2}{\partial y^2} \tag{3.7}$$

$$\left(P_{op}^2 + k_0^2 Q_{op}^2\right) p = 0 \tag{3.8}$$

Defining the x-axis as the primary "forward" direction we can write Equation (3.9)

$$p = \frac{P_0}{\sqrt{Q_{op}}}\Psi e^{ik_0 x} \tag{3.9}$$

With proper factorization the outward propagating field is obtained by defining Equation (3.10) [9].

$$\frac{\partial \Psi}{\partial x} = ik_0(Q_{op} - 1)\Psi = -ik_0 H_{op}\Psi \tag{3.10}$$

H_{op} is a Hamiltonian-like operator defined in Equation (3.11).

$$H_{op} = 1 - Q_{op} \tag{3.11}$$

B. SPLIT-STEP FOURIER ALGORITHM

The split-step Fourier (PE/SSF) method is one of three common methods for computing PE solutions [10]. The speed and simplicity of the PE/SSF method make it advantageous over other methods in range-dependent media [11]. Generating solutions to the PE is largely dependent on approximations of the pseudo-differential operator Q_{op}. If the Thomson-Chapman Wide Angle PE approximation is made, the operator takes the form of Equation (3.12).

$$Q_{op} \approx \sqrt{1+\mu+\nu} + \sqrt{1+\varepsilon} - 1 = 1 - T_{2DTCWAPE} - U_{TCWAPE} \tag{3.12}$$

$$T_{2DTCWAPE} = 1 - \sqrt{1+\mu+\nu} = 1 - \sqrt{1 + \frac{1}{k_0^2}\frac{\partial^2}{\partial z^2} + \frac{1}{k_0^2}\frac{\partial^2}{\partial y^2}} \tag{3.13}$$

$$U_{TCWAPE} = 1 - \sqrt{1+\varepsilon} = -(n-1) \tag{3.14}$$

U_{op} is a multiplication operator in z-space and, therefore, a diagonal matrix. Since T_{op} is a differential operator, different depth and horizontal eigenfunctions are coupled. However, in the vertical and horizontal wavenumber space, T_{op} is a diagonal symmetric, constant matrix. Thus, we separate the application of each operator. The PE/SSF algorithm in Cartesian coordinates now takes the form of Equation (3.15).

$$\Psi(x+\Delta x, y, z) = e^{-ik_0 \frac{\Delta x}{2} U_{op}(x+\Delta x)} F^{-2D}\left\{ e^{-ik_0 \Delta x \left[\hat{T}_{op}(k_y, k_z)\right]} \times F^{2D}\left[e^{-ik_0 \frac{\Delta x}{2} U_{op}(x)} \Psi(x, y, z) \right] \right\} \quad (3.15)$$

$$\left[\hat{T}_{op}(k_z) + \hat{V}_{op}(k_y)\right]_{2DTCWAPE} = 1 - \sqrt{1 - \left(\frac{k_z^2 + k_y^2}{k_0^2} \right)} \quad (3.16)$$

The general algorithm for the PE/SSF implementation can be summarized as follows. The PE field function Ψ is specified at a range x in the x-domain. Multiplication of the (y,z)-space operator $e^{-ik_0 \frac{\Delta x}{2} U_{op}(x)}$ defined at the beginning of the range step is applied. Then a transformation into the (k_y, k_z)-domain is made before a multiplication of the (k_y, k_z)-space operator $e^{-ik_0 \Delta x \left[\hat{T}_{op}(k_y, k_z)\right]}$ is applied. The solution is transformed back to the (y,z)-space then multiplied by the (y,z)-space operator $e^{-ik_0 \frac{\Delta x}{2} U_{op}(x+\Delta x)}$ defined at the end of the range step. Our final result is a field function at $x + \Delta x$. The discrete FFT subroutine in the numerical code assumes the convention of Equation (3.17) and (3.18).

$$\Psi(y, z) = FFT(\Psi(k_y, k_z)) \quad (3.17)$$

$$\Psi(k_y, k_z) = IFFT(\Psi(y, z)) \quad (3.18)$$

C. GRID SIZES

To complete the PE model, we select grid sizes with steps in range, Δx, cross-range, Δy, and depth, Δz. Accurate solutions can be obtained when the steps are on the order of a few acoustic wavelengths. The mesh size is much larger than that needed by

other numerical algorithms based on finite differences or finite approximations to the differential operators in range-dependent environments.

D. IMPLEMENTATION

1. Broadband Parameters

To estimate the broadband, impulsive arrival structure, the model was run for 512 frequencies over a bandwidth of 127.75 Hz centered at 400 Hz. This resulted in a frequency bin size of 0.25 Hz and a total time window of 4 sec. The source depth, consistent with the depth of the glider at the time of the whale vocalizations, was 92 m. Reciprocity allows for the acceptance of calculations outward from the receiver location. While this time window is long enough to capture the primary structure of the measured impulse response, the smaller bandwidth was chosen to reduce computational run time. To further reduce the computational run time, the frequency band was divided into four equally sized parts, and each sub-band was processed on a separate computer processor. The output data was combined into a single file using the "Combine Frequencies" script found in Appendix A: PE output code.

2. Boundary Filters

An important tactic in employment of the MMPE model relates to undefined filters, or "sponges" to remove acoustic energy from very deep depths in the bottom (from which no energy is expected to return) and from very high angles of propagation. First, we recognize the radiation condition $\Psi(z) \to 0$ as $z \to \pm\infty$. Since the computational depth is finite, however, we must force the field amplitude to zero at the maximum depth. The MMPE model applies a sine-squared filter function, designed to range from unity down to 0.5, to the bottom third of the computational depth. After multiple applications of such a filter, the deepest part of the signal is greatly reduced. A similar filter is applied in the cross-range dimension to force $\Psi(y) \to 0$ as $y \to \pm\infty$

Additionally, some type of filtering may be needed in the k_z- and k_y-domain to remove angles beyond 90 degrees. The wavenumber domain propagator function of

Equation (3.16) naturally provides a filter for these angles by making them evanescent. Therefore, energy is attenuated beyond this limit.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. ENVIRONMENTAL DESCRIPTION AND DATA PROCESSING

A. BATHYMETRY EXTRACTION

Additional analysis conducted for this paper was completed in MATLAB R2012b.

Bathymetry data for the Monterey Bay Canyon was extracted from the Southern California Coastal Ocean Observing System [12], and gridded at roughly (1/20) min in latitude and longitude. The bathymetry extraction code can be found in Appendix B: Bathymetry extraction code.

The bathymetry extraction produced a MMPE input file that was used in the 3D-MMPE model. The SSP and geoacoustical properties were defined as independent of range and bearing while the bottom was assumed to be a fluid-like homogeneous half-space with typical coastal sediment values: sound speed 1700 m/s, density 1.8 g/cm^3, and attenuation of 0.2 dB/m/kHz. The surface was assumed to be a flat, pressure release boundary. The SSP was based on average profiles collected in the upper water column during the September 2013 experiment, and extrapolated to depth using historical data. The SSP is shown in Figure 5.

Figure 6 and Figure 7 show the bathymetry profile for the northern and southwestern runs referenced in Figure 4. The bathymetry was extracted on a square grid of approximate size 10 km x 10 km, sampled at 40 m in range and 40 m in cross-range for a gridded bathymetry of 250 points in range and 250 points in cross-range. The calculations were performed with the starting field at a range of $x = 0$ and cross-range of $y = 0$, in the lower center portion of each figure. The bathymetry utilized in the calculations was a subset of the gridded bathymetry shown, based on the maximum range of the calculation and a total cross-range of $8192*\Delta y$, where Δy was the computational cross-range grid size. For the frequencies investigated in this thesis, the cross-range grid size was defined as $\Delta y = 0.83$m. All of the calculations were done out to a maximum range of 5 km for a computational region that was 5 km in range by 6.825 km in cross-range. The computational region is boxed in Figures 6 and 7.

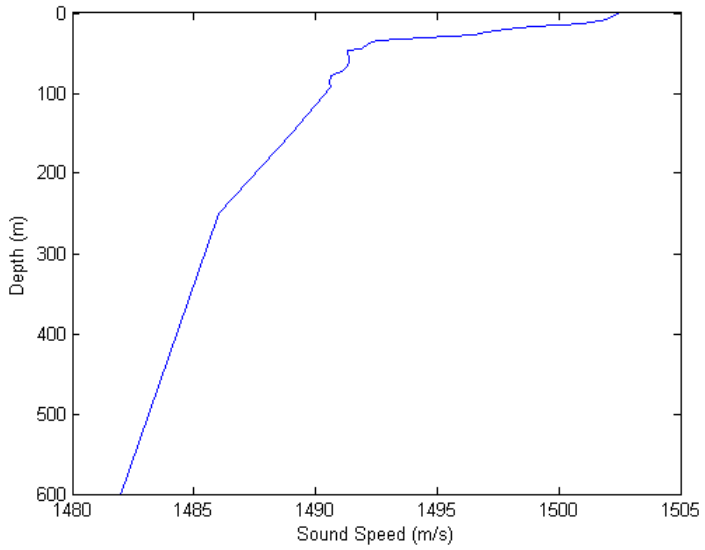

Figure 5. September 2013 Sound Speed Profile for the Monterey Bay Canyon

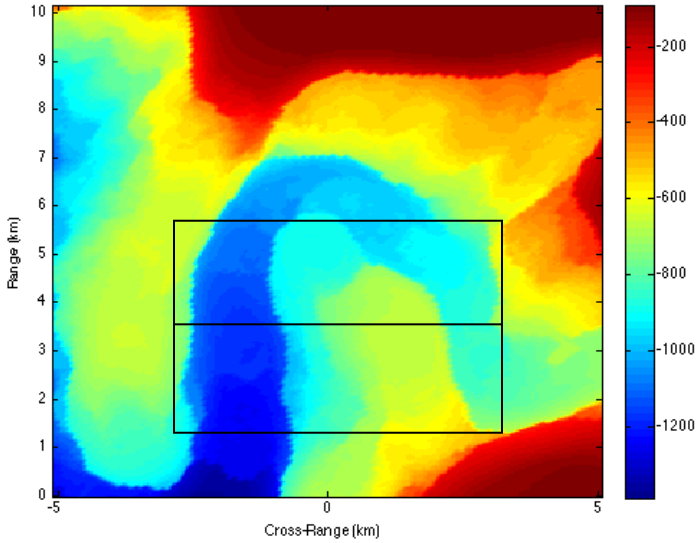

Figure 6. Northern bathymetry profile for the Monterey Bay Canyon (DOA Sample 3); canyon present in negative cross-ranges with canyon wall in positive cross-ranges.

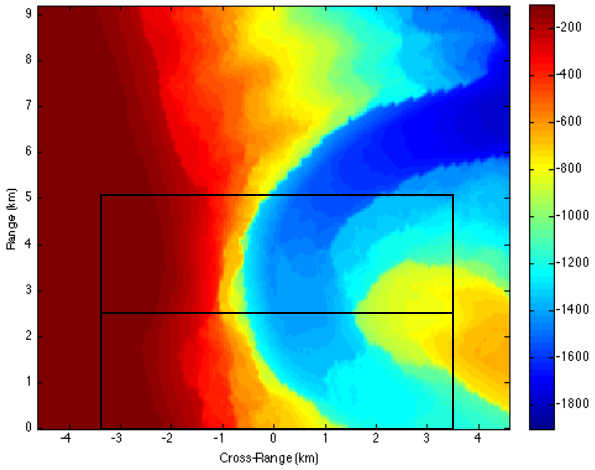

Figure 7. Southwestern bathymetry profile for the Monterey Bay Canyon (DOA Sample 2); canyon present in positive cross-ranges with canyon wall and shelf in negative cross-ranges.

B. INTENSITY PROCESSING

The binary output from the 3D-MMPE Model was then post-processed with an MMPE script that allows a user to extract field data for range (fixed *x*), single cross-range (fixed *y*), single depth, or single interface. For most of this paper, acoustic pressure and particle velocity calculations were made for single ranges of x = 2.5 and 5 km. For single range calculations, the user may select from single frequency, single cross-range, or single depth options. A depth of 50 m was used for this paper.

Intensity processing was then used to determine DOA, as previously developed by LT Upshaw. This technique processes time-domain acoustic wave pressure and particle velocity to resolve the direction of wave propagation. The time-domain arrival structure generated by the MMPE model was user selected as a short-duration impulsive signal (based on a simple Hanning window in the frequency domain) or a longer duration chirp-like signal, as described below. The analysis was conducted by applying a user selected Hanning window or Chirp window before using an FFT with a user defined sliding processing window (0.1 seconds) that advanced at user defined time steps (0.05 seconds) throughout the sample.

After the signals were transformed into the frequency domain, they were used to compute the DOA. The "intensity" script was used to calculate and display the bearing information. To examine the vector intensity bearing estimation, a bubble plot was created in MATLAB as a function of time that displayed the relative amplitude of the intensity vector proportional to the bubble size, centered at the true bearing of the signal relative to the glider's position. The code used in this research can be found in Appendix C: Processing code.

C. SIGNAL TYPE

The intensity processing code allows the user to examine the response of two different signal types: an impulsive signal (similar to the shots) or a chirp signal (similar to the whale vocalizations). To generate the impulsive signal, the frequency domain output of the MMPE Model was filtered with a simple Hanning window over the frequency bandwidth. This frequency data is then defined as the response of the field at the positive frequency bins corresponding to the bandwidth 336 – 464 Hz. The total number of frequency bins is then scaled up to a power of 2 above the maximum frequency (464 Hz), and zero-padded everywhere except in the computational band. This results in a frequency domain vector of length 4096, which corresponds to a sample frequency of $f_s = 1024\,\text{Hz}$. After transforming to the time-domain and removing the imaginary part of the signal, the real time series used for subsequent analysis was a 4 sec long time series sampled at 1024 Hz.

The synthetic chirp signal is generated in the time-domain over a user defined time. The chirp rate is found by dividing the signal bandwidth by the chirp time as shown in Equation (4.1).

$$r_{chirp} = \frac{f_{\max} - f_{\min}}{t_{chirp}} \tag{4.1}$$

The chirp signal is created as a sinusoid of the phase defined in Equation (4.2).

$$\theta_{chirp} = 2\pi\left(f_{\min} * (t - t_{\min}) + \frac{r_{chirp}}{2}(t - t_{\min})^2 \right) \tag{4.2}$$

A Hanning window is applied to the generated time-domain chirp signal such that at all other times outside the processing window, the signal is set to 0. It is then transformed to the frequency domain. The chirp signal is then normalized to the maximum value in the frequency domain and becomes the complex amplitude spectrum of the chirp signal, which may be applied to the MMPE frequency domain data to create the response of such a signal. This data is then transformed back to the time domain before applying the intensity processing algorithm.

D. PLANE-WAVE BEAMFORMING

Plane-wave beamforming is a form of spatial processing which assumes the acoustic signal can be decomposed into simple plane waves [13]. This type of processing requires an extended spatial sampling which is usually accomplished with a linear array of receivers. Although not utilized in the experimental tests, the numerical analysis of the response on a linear array will help separate the multipath structure and aid in the interpretation of any observed bearing ambiguity at the location of a single vector sensor receiving element.

For this paper, a 400 m array of pressure elements, corresponding to 128 points in cross-range, was selected for use in a FFT beamforming script in MATLAB. The use of an FFT beamformer was ideal for the broadband signals of interest. The code for the plane-wave beamformer can be found within the intensity processing script in Appendix C: Processing code.

THIS PAGE INTENTIONALLY LEFT BLANK

V. RESULTS AND ANALYSIS

Whale vocalizations were analyzed in two regions with varying bathymetry. The bathymetry for the northern and southwestern regions is shown in Chapter IV.A.

A. NORTHERN RUN

Before investigating the presence of horizontal coupling, the time-domain response as a function of depth is shown along the central cross-range at a range of 1 km, 2.5 km, and 5 km in Figure 8, Figure 9, and Figure 10, respectively. At a range of 1 km, the observed arrival is the combination of the direct and surface-reflected paths. At this distance and these depths, there is not enough temporal resolution in the signal to clearly discriminate the two arrivals (although some separation can be noted at 500 m depth). In addition, because the bottom depth was approximately 1km at the source location, any bottom-reflected energy would correspond to reflections at grazing angles of roughly 60 deg, well above the critical angle. Because such angle of propagation are not expected to contribute significantly to the solution, the model does not incorporate angles much above 45 degrees. Thus, no bottom-reflected energy is observed at this short distance.

However, at ranges of 2.5 km and 5 km, later arrivals are observed from bottom interactions. Specifically, in Figure 9 the bottom-reflected energy is observed to arrive roughly 10 msec behind the direct/surface-reflected paths. In Figure 10, the first bottom-reflected energy arrives shortly behind the direct/surface-reflected paths, and a secondary bottom-reflected path is seen to arrive roughly 1.5 sec after the initial arrival.

It is also worth noting that at ranges of 2.5 km and 5 km, the direct and surface-reflected paths are completely indistinguishable. More importantly, due to the downward refracting nature of the sound speed profile at these depths, a direct/surface-reflected path shadow zone is being formed at the shallowest depths, above about 10 m at a range of 2.5 km and above about 25 m at a range of 5 km. Because our analysis presented here is at a depth of 50 m, the results will always include the influence of the strong direct/surface-reflected paths.

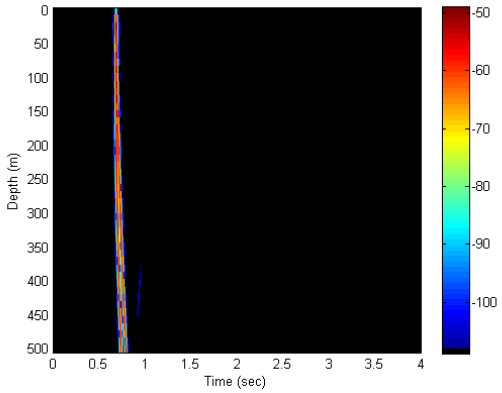

Figure 8. Time-domain response as a function of depth for northern run, range = 1 km, cross-range = 0m, for the impulsive source.

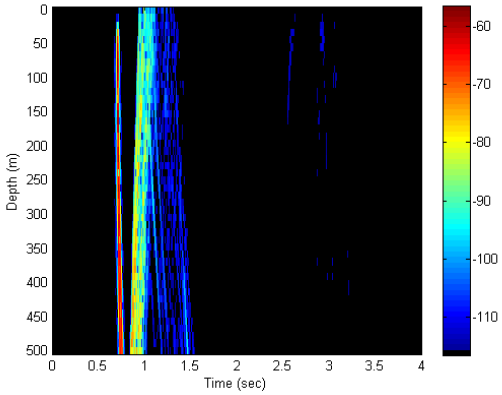

Figure 9. Time-domain response as a function of depth for northern run, range = 2.5 km, cross-range = 0m, for the impulsive source.

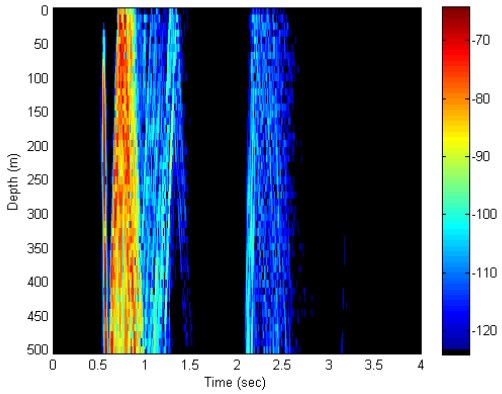

Figure 10. Time-domain response as a function of depth for northern run, range = 5 km, cross-range = 0m, for the impulsive source.

After running the post-script to obtain data at a range of 2.5 km and depth of 50 m, the broadband processing for the impulsive signal provided the time structure shown in Figure 11. This is pre-processing within the intensity processing script.

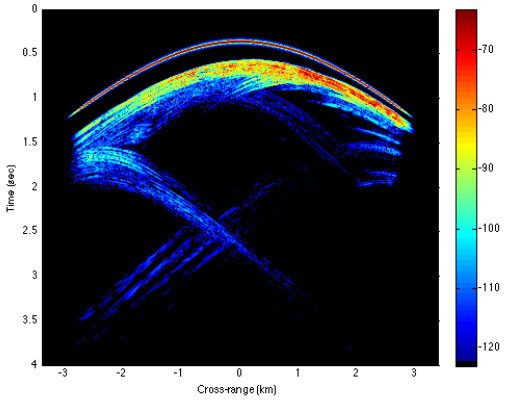

Figure 11. Time-domain response as a function of cross-range for northern run, range = 2.5 km, depth = 50 m, for the impulsive source.

This structure illustrates varying bottom-bounce interactions as a function of cross-range. The initial arrival is seen to be well-defined, and corresponds to the direct/surface-reflected path. Because the model assumes the surface to be perfectly flat, there is no degradation in signal structure for this path. At positive cross-ranges, significantly more scattering is observed, and bottom-reflected paths enter the image for negative cross-ranges. This is expected from the canyon wall present on the right and greater depths to the left of the central cross-range in Figure 6. The multipath structure appears to illustrate multiple angles of arrival which suggests the presence of horizontal coupling.

For this data set and each subsequent set to be processed, analysis is conducted for data at the central cross-range, corresponding to $y = 0$ km, and cross-ranges off-center at $y = \pm 1$ km.

The plane-wave beamforming output for each cross-range assessed in this region is shown in Figures 12 through 14.

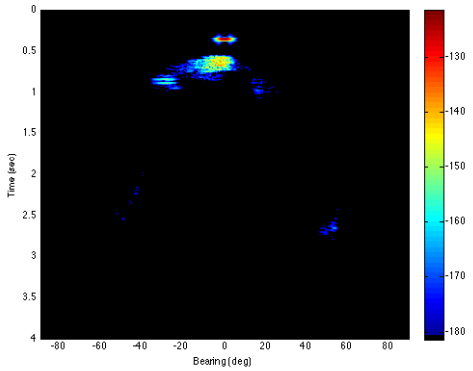

Figure 12. Beamformer output for northern run, range = 2.5 km, cross-range = 0 km, depth = 50 m, for the impulsive source.

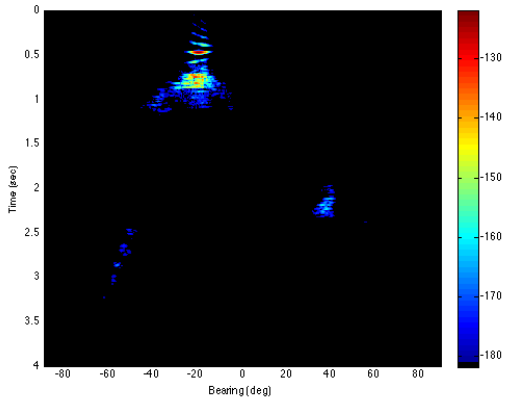

Figure 13. Beamformer output for northern run, range = 2.5 km, cross-range = -1 km, depth = 50 m, for the impulsive source.

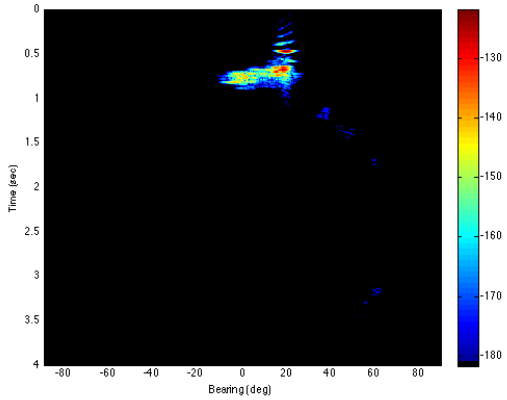

Figure 14. Beamformer output for northern run, range = 2.5 km, cross-range = +1 km, depth = 50 m, for the impulsive source.

The output centered at y = 0 km is dominated by the strong direct/surface-reflected path arrival at 0 degrees (broadside). The subsequent arrival is more diffuse but remains concentrated near broadside. There are weaker subsequent arrivals that occur at negative angles on the array (going from positive to negative cross-ranges), which are consistent with the general features observed in Figure 11. Finally, the weak arrival at around +50 degrees indicates a large horizontal scattering feature to the left of the source. This feature should be considered with caution due to its lower intensity compared to other arrival structures as numerical noise may play a role at these levels.

The outputs centered at $y = \pm$ 1 km demonstrate similar behavior with even greater angular spread for the secondary arrivals at + 1 km. The secondary arrivals tend toward negative angles, consistent with the earliest bottom reflections to the right of the source and generating horizontal scattering toward the negative cross-ranges. In both cases, there are late, weak arrivals observed at positive angles. As before, the late arrivals are considered with caution based on their relatively weak strength.

Bubble plots generated by the intensity processing produced similar results with the greatest horizontal spread found in the +1 km cross-range as shown in Figure 15. Because the bubbles are proportional to the magnitude of the intensity vector, only the surface-reflected path and stronger secondary arrivals are observed. There is approximately 30 degrees of bearing ambiguity from the secondary arrival which coincides with that observed in the beamformer output at this cross-range.

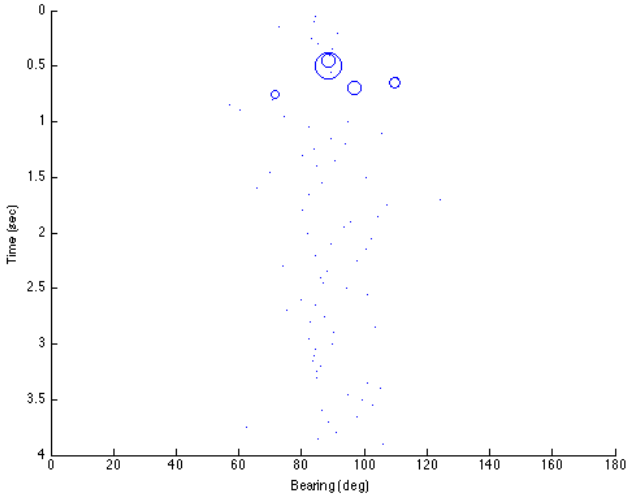

Figure 15. Bubble plot for northern run, range = 2.5 km, cross-range = +1 km, depth = 50m, for the impulsive source; bearing ambiguity is shown to be approximately 30 degrees.

Figure 16 displays the arrival time structure for the impulsive source at a range of 5 km. Again, the earliest arrival corresponds to the direct/surface-reflected path and shows no degradation in the structure. Compared to the response at a range of 2.5 km, there is even more defined scattering at positive cross-ranges and bottom-reflected paths for negative cross-ranges.

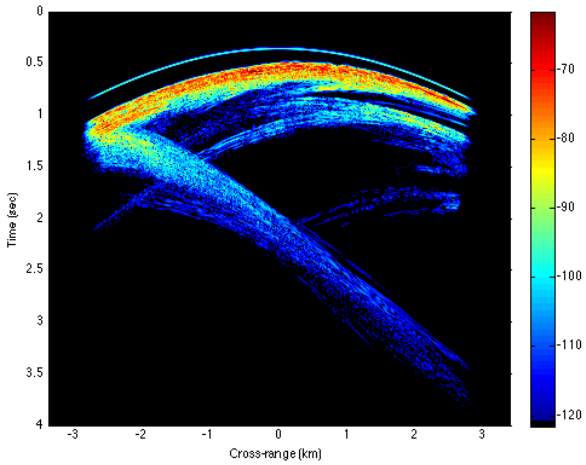

Figure 16. Time-domain response as a function of cross-range for northern run, range = 5 km, depth = 50m, for the impulsive source.

With a similar response to that observed at a range of 2.5 km, we expect the beamformer outputs shown in Figure 17 through 19 to illustrate arrival patterns consistent with those observed at 2.5 km. The secondary arrivals tend toward negative angles, generating horizontal scattering toward negative cross-ranges and there are weaker, late arrivals at large positive angles.

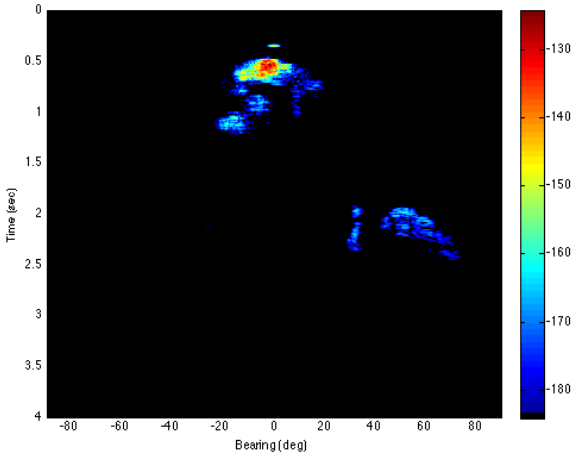

Figure 17. Beamformer output for northern run, range = 5 km, cross-range = 0 km, depth = 50 m, for the impulsive source.

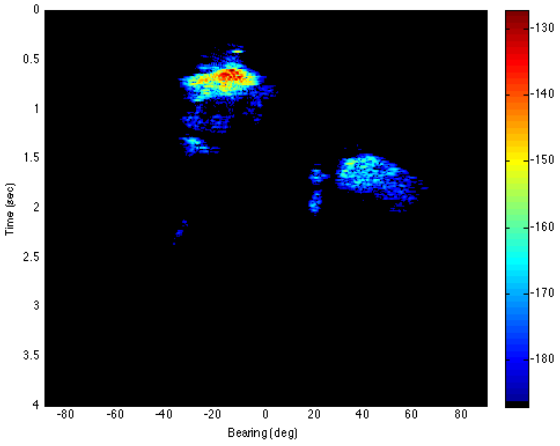

Figure 18. Beamformer output for northern run, range = 5 km, cross-range = -1 km, depth = 50 m, for the impulsive source.

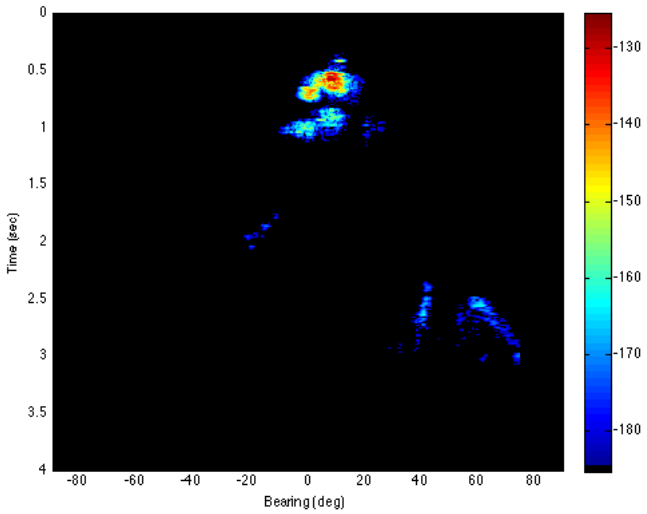

Figure 19. Beamformer output for northern run, range = 5 km, cross-range = +1 km, depth = 50 m, for the impulsive source.

The bubble plots for each data set produced similar results in bearing and the bubble plot shown in Figure 20 demonstrates the ambiguity of over 20 degrees that is consistent with the horizontal angle spread observed in Figure 19.

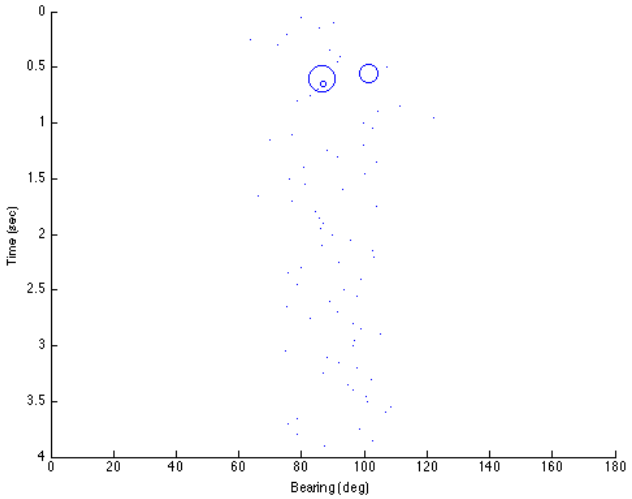

Figure 20. Bubble plot for northern run, range = 5 km, cross-range = +1 km, depth = 50 m, for the impulsive source; bearing ambiguity is shown to be approximately 20 degrees.

For further comparison, a 0.5 sec chirp signal was processed at the +1 km cross-range for the 2.5 km and 5 km ranges. The time structures and bubble plots for each range are shown in Figures 21 through 24.

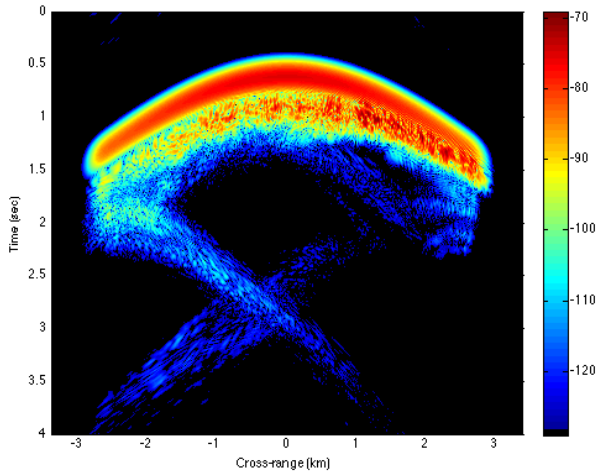

Figure 21. Time-domain response as a function of cross-range for northern run, range = 2.5 km, depth = 50 m, for the chirp signal.

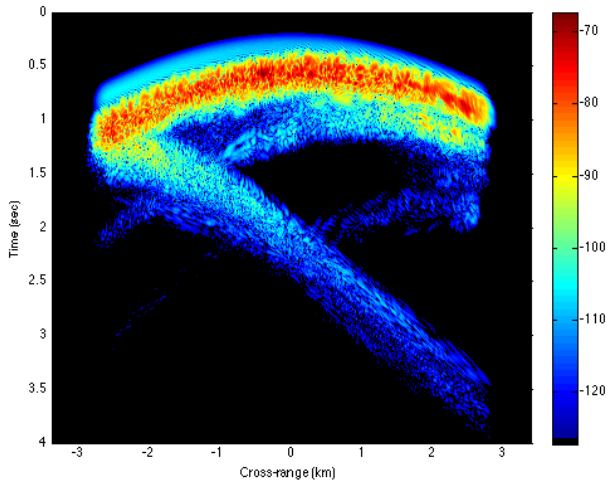

Figure 22. Time-domain response as a function of cross-range for northern run, range = 5 km, depth = 50 m, for the chirp signal.

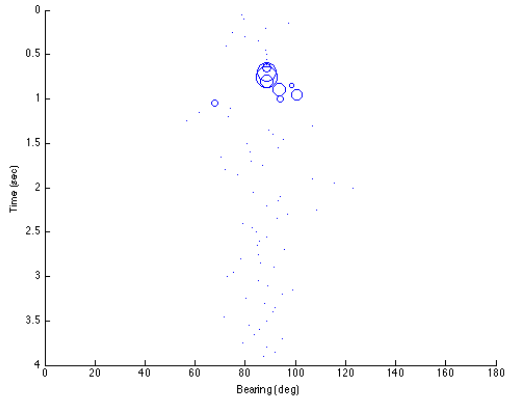

Figure 23. Bubble plot for northern run, range = 2.5 km, cross-range = +1 km, depth = 50 m, for the chirp signal; bearing ambiguity is shown to be approximately 20 degrees.

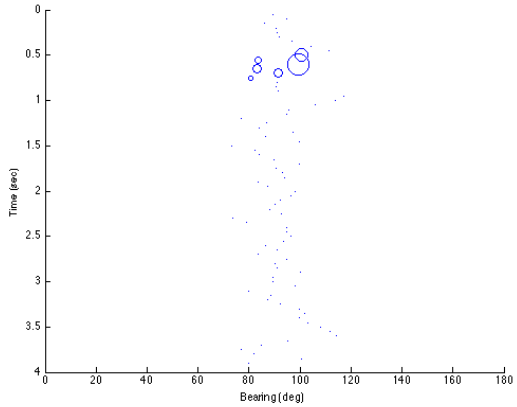

Figure 24. Bubble plot for northern run, range = 5 km, cross-range = +1 km, depth = 50 m, for the chirp signal; bearing ambiguity is shown to be approximately 30 degrees.

The chirp signals, expectedly, have a broader arrival structure. It was originally anticipated that the interfering multi-path structure of the surface and bottom-reflected paths might be the cause of the intensity bearing ambiguity. However, the ambiguity observed in Figures 23 and 24 does not appear to increase from the previous observations of the impulsive signal response. While further analysis may be warranted, these results suggest it is the direct spread of the horizontal arrival angles by up to 30 degrees that is causing the intensity bearing ambiguity, and not the coherent interference of the horizontal multipath structure.

B. SOUTHWEST RUN

In contrast to the bathymetry in the northern run, the southwest run has a canyon wall to the left and deeper depths to the right of the central cross-range. From the results of the northern run, we now anticipate scattering in the negative cross-ranges and bottom-reflected paths entering the image in the positive cross-ranges of the southwestern run. The time structure for a range of 2.5 km is shown in Figure 25.

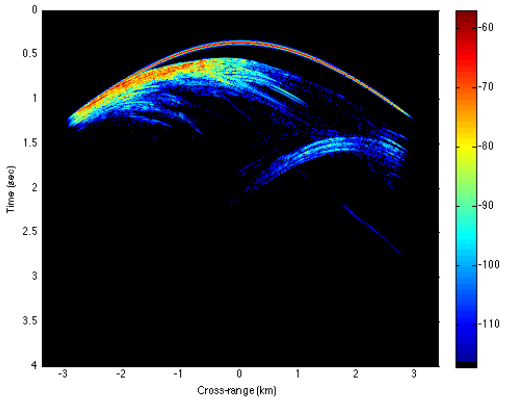

Figure 25. Time-domain response as a function of cross-range for southwestern run, range = 2.5 km, depth = 50 m, for the impulsive source.

Because the canyon wall to the left of the central-cross range runs near the surface, the direct/surface-reflected arrival structure becomes scattered in the negative cross-ranges. The bottom-reflected arrivals do appear in the positive cross-ranges as predicted.

To investigate the horizontal coupling, the beamforming outputs for this range are shown in Figure 26 through 28.

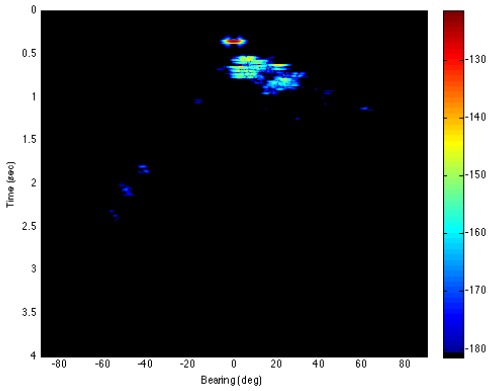

Figure 26. Beamformer output for southwestern run, range = 2.5 km, cross-range = 0 km, depth = 50 m, for the impulsive source.

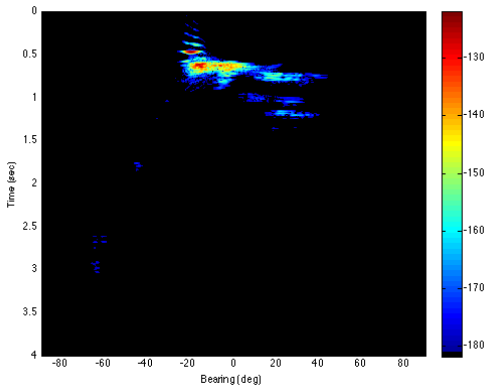

Figure 27. Beamformer output for southwestern run, range = 2.5 km, cross-range = -1 km, depth = 50 m, for the impulsive source.

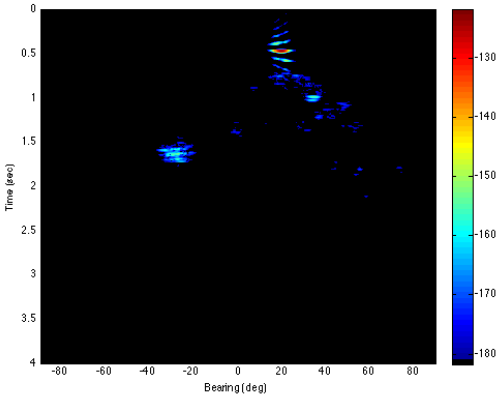

Figure 28. Beamformer output for southwestern run, range = 2.5 km, cross-range = +1 km, depth = 50 m, for the impulsive source.

Along the central cross-range, we again have a strong, direct/surface-reflected signal that is broadside to the array. The secondary arrivals now tend toward positive angles (going from negative to positive cross ranges) as compared to the northern run, but are consistent with the features of the time structure in Figure 25. The presence of late, weaker arrivals at large negative angles indicate a horizontal scattering feature to the right of the source in this run. While the off-center cross-ranges demonstrate similar patterns, a significant angular spread for the secondary arrivals at -1 km is observed. This indicates strong, horizontal coupling in the vicinity of the canyon wall.

The bubble plots continued to track with the bearing results from the beamformer output. Figure 29 is the bubble plot for the -1 km cross-range which had nearly 30 degrees of bearing ambiguity.

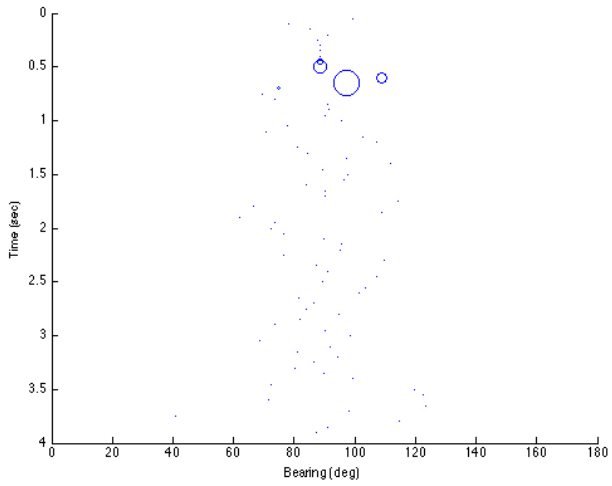

Figure 29. Bubble plot for southwestern run, range = 2.5 km, cross-range = -1 km, depth = 50 m, for the impulsive source; bearing ambiguity is shown to be approximately 30 degrees.

Looking at a range of 5 km we begin again with the time response of the impulsive signal in Figure 30.

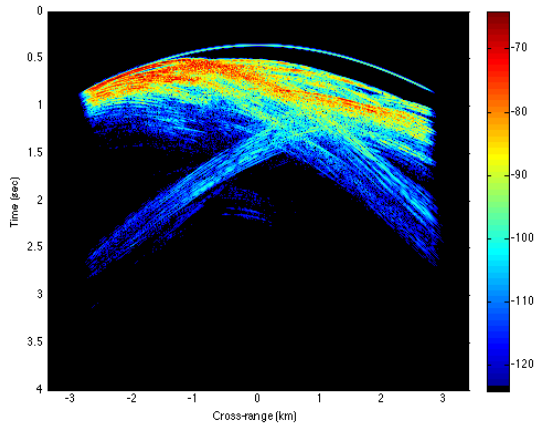

Figure 30. Time-domain response as a function of cross-range for southwestern run, range = 5 km, depth = 50 m, for the impulsive source.

Similar to the response at a range of 2.5 km, the direct/surface-reflected arrival becomes scattered at cross-ranges to the left of -1 km. We now observe significant scattering in the secondary arrivals from negative to positive cross-ranges. Also significant at this range is the bottom-reflected arrivals that appear from both the right and the left. As shown in beamforming outputs in Figure 31 through 33, this introduces secondary arrivals with stronger intensity and greater bearing ambiguity.

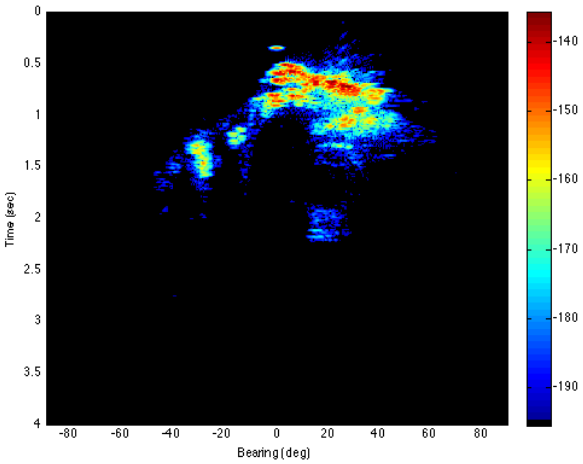

Figure 31. Beamformer output for southwestern run, range = 5 km, cross-range = 0 km, depth = 50 m, for the impulsive source.

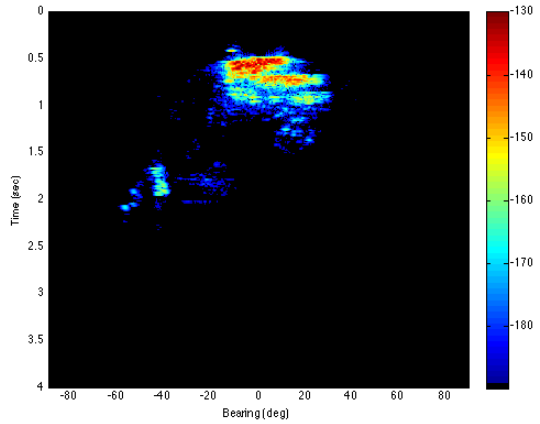

Figure 32. Beamformer output for southwestern run, range = 5 km, cross-range = -1 km, depth = 50 m, for the impulsive source.

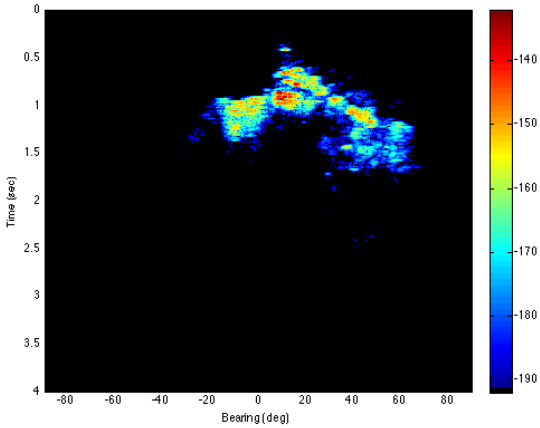

Figure 33. Beamformer output for southwestern run, range = 5 km, cross-range = +1 km, depth = 50 m, for the impulsive source.

Beginning with the array at the central cross-range, the direct/surface-reflected arrival is broadside with secondary arrivals tending toward positive angles. The secondary arrivals are relatively stronger than at a range of 2.5 km and also demonstrate a greater spread in angular ambiguity for the central cross-range and subsequent off-center cross-ranges. At cross-ranges of 0 and – 1 km, we observe the late, weaker arrivals at large negative angles, consistent with the earliest bottom reflections on the left.

The impact of the bottom-path reflections from the left and right at this range are demonstrated by the greatest bearing ambiguity along the +1 km cross-range. Following

the trends observed in both northern run samples and the southwestern run at 2.5 km, the -1 km cross-range was expected to have the greatest spread in angular ambiguity. However, the influence of bottom-path arrivals crossing the central cross-range and into the positive cross-ranges creates secondary signals of greater intensity and spread in bearing.

The bubble plot for the +1 km cross-range in Figure 34 shows approximately 30 degrees of bearing ambiguity which is consistent with the angular spread observed in Figure 31. Compared to the bubble plots of the other impulsive sources, we note the greater relative magnitude for this sample.

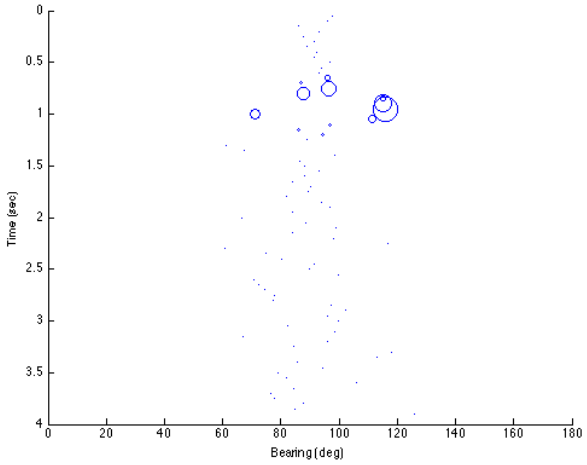

Figure 34. Bubble plot for southwestern run, range = 5 km, cross-range = +1 km, depth = 50 m, for the impulsive source; bearing ambiguity is shown to be approximately 40 degrees.

For the southwestern run, the -1 km cross-range is of interest at a range of 2.5 km while the +1 km cross-range is of interest at 5 km. Thus, a 0.5 sec chirp signal was processed for each combination of range and cross-range with the time structures and bubble plots shown in Figures 35 through 38.

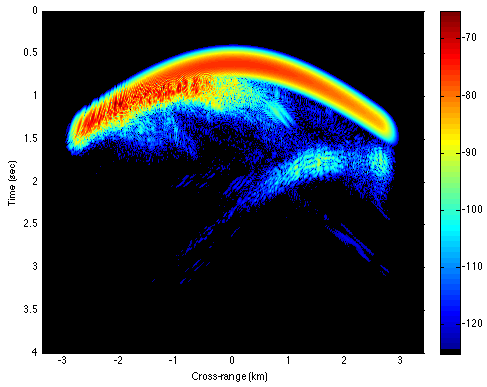

Figure 35. Time-domain response as a function of cross-range for southwestern run, range = 2.5 km, depth = 50 m, for the chirp signal.

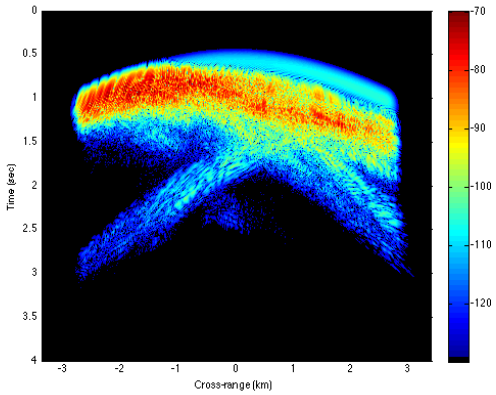

Figure 36. Time-domain response as a function of cross-range for southwestern run, range = 5 km, depth = 50 m, for the chirp signal.

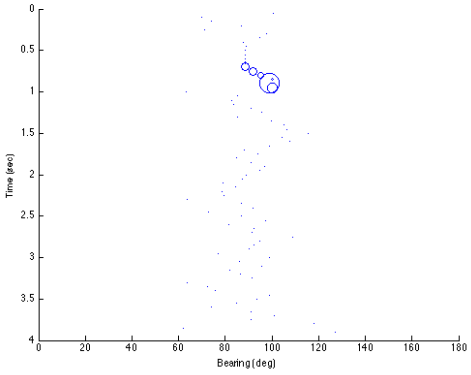

Figure 37. Bubble plot for southwestern run, range = 2.5 km, cross-range = -1 km, depth = 50 m, for the chirp signal; bearing ambiguity is shown to be approximately 20 degrees.

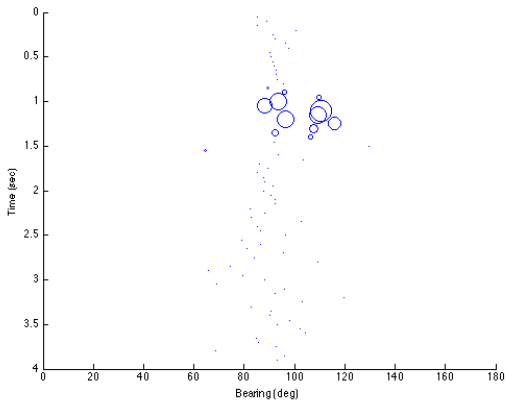

Figure 38. Bubble plot for southwestern run, range = 5 km, cross-range = +1 km, depth = 50 m, for the chirp signal; bearing ambiguity is shown to be approximately 30 degrees.

The chirp signals, as in the northern run, have a broader arrival structure. Again, the bearing ambiguity for the chirp signals does not appear to increase from the observations of the impulsive signal response in the southwestern run. This further suggests that the spread of the horizontal arrival angles is the cause of the intensity bearing ambiguity vice the coherent interference of the horizontal multipath structure.

VI. CONCLUSIONS AND RECOMMENDATIONS

Analysis of the data generated by the 3D-MMPE model, extracted at a combination of ranges and cross-ranges, illustrated significant horizontal coupling effects as a result of the varying bathymetry within the Monterey Bay Canyon. In the time arrival structures from impulsive signals in both the northern and southwestern runs, significantly stronger scattering was observed in regions of shallow depth within the respective sample.

Inspection of the plane-wave beamforming output for the impulsive signals found secondary arrivals and late, weaker arrivals that tended toward the direction of horizontal scattering features in each case. Intensity processing of the impulsive signals was used to compare the bearing ambiguity in each case. The bubble plots provided similar results for the bearing ambiguity measured in the impulsive signals.

It was originally anticipated that the interfering multi-path structure of the surface and bottom-reflected paths might be the cause of the intensity bearing ambiguity. However, the ambiguity observed in the intensity processing of 0.5 sec chirp signals did not appear to increase from the observations of the impulsive signal responses. When observing the chirp signal response in the southwestern run, the ambiguity appeared to decrease with the longer duration signal. Observations in both runs suggest it is the direct spread of the horizontal arrival angles that is causing the intensity bearing ambiguity, and not the coherent interference of the horizontal multipath structure. Although not directly confirmed through modeling, these results suggest that the lack of ambiguity in the previously analyzed shot data was due to the benign bathymetry in the region, rather than the highly impulsive nature of the source.

To efficiently generate the MMPE data, a 127.75 Hz bandwidth was processed about a center frequency of 400 Hz. Future consideration may be given to analyzing the entire processing band (350-650 Hz) used in LT Upshaw's research. Further, the depth grid was limited to 50–150 m to reduce data storage requirements. Longer processing runs over a greater range of depths may also be considered. Investigation of depths above

50 m would provide insight for whale vocalizations closer to the surface (consider 10 m depth).

The presence of shadow zones created by the direct/surface-reflected paths at shallow depths gives cause for investigation of the signal response based only on bottom-reflected interactions. The arrival structure of bottom-reflected energy alone may increase the ambiguity in the bearing estimation. This may suggest that accurate bearing estimates require deeper receivers, to insure reception of the direct/surface-reflected paths, as well as longer processing windows that would be dominated by the stronger, early arrivals, thereby overwhelming the variability introduced in the bottom-reflected arrivals. Finally, consideration may be given to investigating ambient noise and any quantifiable effects based on SNR.

APPENDIX A. PE OUTPUT CODE

A. COMBINE FREQUENCIES SCRIPT

```
clear
disp('Combining files');
fid1=fopen('352\press_400_1.bin');
fid2=fopen('384\press_400_2.bin');
fid3=fopen('416\press_400_3.bin');
fid4=fopen('448\press_400_4.bin');
fid=fopen('press_400.bin','w');
% fid1=fopen('352\apvx_400_1.bin');
% fid2=fopen('384\apvx_400_2.bin');
% fid3=fopen('416\apvx_400_3.bin');
% fid4=fopen('448\apvx_400_4.bin');
% fid=fopen('apvx_400.bin','w');
% fid1=fopen('352\apvy_400_1.bin');
% fid2=fopen('384\apvy_400_2.bin');
% fid3=fopen('416\apvy_400_3.bin');
% fid4=fopen('448\apvy_400_4.bin');
% fid=fopen('apvy_400.bin','w');
% fid1=fopen('352\apvz_400_1.bin');
% fid2=fopen('384\apvz_400_2.bin');
% fid3=fopen('416\apvz_400_3.bin');
% fid4=fopen('448\apvz_400_4.bin');
% fid=fopen('apvz_400.bin','w');

% Read in first record of data
nrecs=fread(fid1,1,'int32');
c0=fread(fid1,1,'float32');
nf=fread(fid1,1,'int32');cfreq=fread(fid1,1,'float32');freqbw=fread(fid
1,1,'float32');
irng=1;
nrout=fread(fid1,1,'int32');rngmin=fread(fid1,1,'float32')/1000;
rngmax=fread(fid1,1,'float32')/1000;dr=fread(fid1,1,'float32')/1000;
nzout=fread(fid1,1,'int32');depmin=fread(fid1,1,'float32');depmax=fread
(fid1,1,'float32');
ny=fread(fid1,1,'int32');dy=fread(fid1,1,'float32');
bdint(irng)=fread(fid1,1,'float32');dbdint(irng)=fread(fid1,1,'float32'
);
sd=fread(fid1,1,'float32');itype=fread(fid1,1,'int32');
% Read in bathymetry
fseek(fid1,(4*(2*nzout*(1+(1+nrout)*ny*nf)-18)),0);
for ir=1:nrout+1,
    bath1(ir,:)=fread(fid1,ny,'float32');
%    bath1(ir,:)=fftshift(bath1(ir,:));
end
% trick for constant interface depths for xyz problem
dbath1(:,1)=fread(fid1,nrout+1,'float32');
surf1(:,1)=fread(fid1,nrout+1,'float32');

% determine size of new file
nf=4*nf;
```

```
nrecs=2*nzout*(1+(1+nrout)*ny*nf)+(nrout+1)*ny+2*(nrout+1);
% create new file full of zeros (to be overwritten later)
disp('Generating new file with zeros');
temp=zeros(1,2*nzout);
for it=1:(1+(1+nrout)*ny*nf),
    fwrite(fid,temp,'int32');
end
temp2=zeros(1,(nrout+1));
for it=1:(ny+2),
    fwrite(fid,temp2,'int32');
end

% write new file header
disp('Writing new file header');
frewind(fid);
nf=512; cfreq=400.; freqbw=127.75;
fwrite(fid,nrecs,'int32');
fwrite(fid,c0,'float32');
fwrite(fid,nf,'int32');fwrite(fid,cfreq,'float32');fwrite(fid,freqbw,'f
loat32');
fwrite(fid,nrout,'int32');fwrite(fid,rngmin*1000.,'float32');
fwrite(fid,rngmax*1000.,'float32');fwrite(fid,dr*1000.,'float32');
fwrite(fid,nzout,'int32');fwrite(fid,depmin,'float32');fwrite(fid,depma
x,'float32');
fwrite(fid,ny,'int32');fwrite(fid,dy,'float32');
fwrite(fid,bdint(1),'float32');fwrite(fid,dbdint(1),'float32');
fwrite(fid,sd,'float32');fwrite(fid,itype,'int32');

% now read from old files and write to new file
disp('Processing 1st file');
header=2*nzout;
fseek(fid,(4*header),'bof');
fseek(fid3,(4*(header+64*(2*nzout*(nrout+1)*ny))),'bof');
for ifr=1:64,
  for ir=1:nrout+1,
    for iy=1:ny,
      data=fread(fid3,2*nzout,'float32');
      fwrite(fid,data,'float32');
    end
  end
end
fseek(fid3,(4*header),'bof');
for ifr=1:64,
  for ir=1:nrout+1,
    for iy=1:ny,
      data=fread(fid3,2*nzout,'float32');
      fwrite(fid,data,'float32');
    end
  end
end

disp('Processing 2nd file');
fseek(fid4,(4*(header+64*(2*nzout*(nrout+1)*ny))),'bof');
for ifr=1:64,
  for ir=1:nrout+1,
```

```
    for iy=1:ny,
      data=fread(fid4,2*nzout,'float32');
      fwrite(fid,data,'float32');
    end
  end
end
fseek(fid4,(4*header),'bof');
for ifr=1:64,
  for ir=1:nrout+1,
    for iy=1:ny,
      data=fread(fid4,2*nzout,'float32');
      fwrite(fid,data,'float32');
    end
  end
end

disp('Processing 3rd file');
fseek(fid1,(4*(header+64*(2*nzout*(nrout+1)*ny))),'bof');
for ifr=1:64,
  for ir=1:nrout+1,
    for iy=1:ny,
      data=fread(fid1,2*nzout,'float32');
      fwrite(fid,data,'float32');
    end
  end
end
fseek(fid1,(4*header),'bof');
for ifr=1:64,
  for ir=1:nrout+1,
    for iy=1:ny,
      data=fread(fid1,2*nzout,'float32');
      fwrite(fid,data,'float32');
    end
  end
end

disp('Processing 4th file');
fseek(fid2,(4*(header+64*(2*nzout*(nrout+1)*ny))),'bof');
for ifr=1:64,
  for ir=1:nrout+1,
    for iy=1:ny,
      data=fread(fid2,2*nzout,'float32');
      fwrite(fid,data,'float32');
    end
  end
end
fseek(fid2,(4*header),'bof');
for ifr=1:64,
  for ir=1:nrout+1,
    for iy=1:ny,
      data=fread(fid2,2*nzout,'float32');
      fwrite(fid,data,'float32');
    end
  end
end
```

```
% write new file bathymetry data
for ir=1:nrout+1,
    fwrite(fid,bath1(ir,:),'float32');
end
fwrite(fid,dbath1(:,1),'float32');
fwrite(fid,surf1(:,1),'float32');


fclose(fid1);
fclose(fid2);
fclose(fid3);
fclose(fid4);
fclose(fid);
```

APPENDIX B. BATHYMETRY EXTRACTION CODE

B. DEGREE TO UTM FUNCTION

```
function  [x,y,utmzone] = deg2utm(Lat,Lon)
% ------------------------------------------------------------------------
----
% [x,y,utmzone] = deg2utm(Lat,Lon)
%
% Description: Function to convert lat/lon vectors into UTM coordinates
(WGS84).
% Some code has been extracted from UTM.m function by Gabriel Ruiz
Martinez.
%
% Inputs:
%    Lat: Latitude vector.   Degrees.  +ddd.ddddd  WGS84
%    Lon: Longitude vector.  Degrees.  +ddd.ddddd  WGS84
%
% Outputs:
%    x, y , utmzone.   See example
%
% Example 1:
%    Lat=[40.3154333; 46.283900; 37.577833; 28.645650; 38.855550;
25.061783];
%    Lon=[-3.4857166; 7.8012333; -119.95525; -17.759533; -94.7990166;
121.640266];
%    [x,y,utmzone] = deg2utm(Lat,Lon);
%    fprintf('%7.0f ',x)
%       458731  407653  239027  230253  343898  362850
%    fprintf('%7.0f ',y)
%      4462881 5126290 4163083 3171843 4302285 2772478
%    utmzone =
%       30 T
%       32 T
%       11 S
%       28 R
%       15 S
%       51 R
%
% Example 2: If you have Lat/Lon coordinates in Degrees, Minutes and
Seconds
%    LatDMS=[40 18 55.56; 46 17 2.04];
%    LonDMS=[-3 29  8.58;  7 48 4.44];
%    Lat=dms2deg(mat2dms(LatDMS)); %convert into degrees
%    Lon=dms2deg(mat2dms(LonDMS)); %convert into degrees
%    [x,y,utmzone] = deg2utm(Lat,Lon)
%
% Author:
%   Rafael Palacios
%   Universidad Pontificia Comillas
%   Madrid, Spain
% Version: Apr/06, Jun/06, Aug/06, Aug/06
```

```
% Aug/06: fixed a problem (found by Rodolphe Dewarrat) related to
southern
%    hemisphere coordinates.
% Aug/06: corrected m-Lint warnings
%--------------------------------------------------------------------
---

% Argument checking
%
error(nargchk(2, 2, nargin));  %2 arguments required
n1=length(Lat);
n2=length(Lon);
if (n1~=n2)
   error('Lat and Lon vectors should have the same length');
end


% Memory pre-allocation
%
x=zeros(n1,1);
y=zeros(n1,1);
utmzone(n1,:)='60 X';

% Main Loop
%
for i=1:n1
   la=Lat(i);
   lo=Lon(i);

   sa = 6378137.000000 ; sb = 6356752.314245;

   %e = ( ( ( sa ^ 2 ) - ( sb ^ 2 ) ) ^ 0.5 ) / sa;
   e2 = ( ( ( sa ^ 2 ) - ( sb ^ 2 ) ) ^ 0.5 ) / sb;
   e2cuadrada = e2 ^ 2;
   c = ( sa ^ 2 ) / sb;
   %alpha = ( sa - sb ) / sa;             %f
   %ablandamiento = 1 / alpha;   % 1/f

   lat = la * ( pi / 180 );
   lon = lo * ( pi / 180 );

   Huso = fix( ( lo / 6 ) + 31);
   S = ( ( Huso * 6 ) - 183 );
   deltaS = lon - ( S * ( pi / 180 ) );

   if (la<-72), Letra='C';
   elseif (la<-64), Letra='D';
   elseif (la<-56), Letra='E';
   elseif (la<-48), Letra='F';
   elseif (la<-40), Letra='G';
   elseif (la<-32), Letra='H';
   elseif (la<-24), Letra='J';
   elseif (la<-16), Letra='K';
   elseif (la<-8), Letra='L';
```

```
   elseif (la<0), Letra='M';
   elseif (la<8), Letra='N';
   elseif (la<16), Letra='P';
   elseif (la<24), Letra='Q';
   elseif (la<32), Letra='R';
   elseif (la<40), Letra='S';
   elseif (la<48), Letra='T';
   elseif (la<56), Letra='U';
   elseif (la<64), Letra='V';
   elseif (la<72), Letra='W';
   else Letra='X';
   end

   a = cos(lat) * sin(deltaS);
   epsilon = 0.5 * log( ( 1 +  a) / ( 1 - a ) );
   nu = atan( tan(lat) / cos(deltaS) ) - lat;
   v = ( c / ( ( 1 + ( e2cuadrada * ( cos(lat) ) ^ 2 ) ) ) ^ 0.5 ) *
0.9996;
   ta = ( e2cuadrada / 2 ) * epsilon ^ 2 * ( cos(lat) ) ^ 2;
   a1 = sin( 2 * lat );
   a2 = a1 * ( cos(lat) ) ^ 2;
   j2 = lat + ( a1 / 2 );
   j4 = ( ( 3 * j2 ) + a2 ) / 4;
   j6 = ( ( 5 * j4 ) + ( a2 * ( cos(lat) ) ^ 2) ) / 3;
   alfa = ( 3 / 4 ) * e2cuadrada;
   beta = ( 5 / 3 ) * alfa ^ 2;
   gama = ( 35 / 27 ) * alfa ^ 3;
   Bm = 0.9996 * c * ( lat - alfa * j2 + beta * j4 - gama * j6 );
   xx = epsilon * v * ( 1 + ( ta / 3 ) ) + 500000;
   yy = nu * v * ( 1 + ta ) + Bm;

   if (yy<0)
       yy=9999999+yy;
   end

   x(i)=xx;
   y(i)=yy;
   utmzone(i,:)=sprintf('%02d %c',Huso,Letra);
end
```

A. GRID INTERPOLATION SCRIPT

```
%R = 6367449;

Lat = reshape(lat.',1,numel(lat));
Lon = reshape(lon.',1,numel(lon));
%Bat = reshape(bath.',1,numel(bath));

[x,y,utmzone] = deg2utm(Lat,Lon);
for ii=1:1201,
    for jj=1:1321,
        kk=(ii-1)*1321+jj;
        X(ii,jj)=x(kk);
        Y(ii,jj)=y(kk);
```

```
    end
end

Lat1=input('Enter starting Lat: ');
Lon1=input('Enter starting Lon: ');

Lat2=input('Enter ending Lat: ');
Lon2=input('Enter ending Lon: ');

%crng=input('Enter cross range: ');
%crng=pi/180*Mr*abs(Lat1-Lat1);
%crng=R*sqrt((cosd(Lon1)*sind(Lat1)-cosd(Lon2)*sind(Lat2))^2 +
(sind(Lon1)*sind(Lat1)-sind(Lat2)*sind(Lon2))^2 + (cosd(Lat1)-
cosd(Lat2))^2);

[x0,y0,utmzone] = deg2utm(Lat1,Lon1);
[x1,y1,utmzone] = deg2utm(Lat2,Lon2);

%x0=755670; y0=4363500; x1=739450; y1=4398700; crng=45000;

rmax=sqrt((x1-x0).^2+(y1-y0).^2); crng=rmax;
%ang=-atan((x1-x0)./(y1-y0));
%ang=atan((x1-x0)./(y1-y0));
ang=-atan2((x1-x0),(y1-y0));

N=250;
dr=rmax/(N-1); ynew=[0:dr:rmax];
dxr=crng/(N-1); xnew=[-(N/2)*dxr:dxr:(N/2-1)*dxr];
[xg,yg]=meshgrid(xnew,ynew);

xrot=xg.*cos(ang)-yg.*sin(ang);
yrot=xg.*sin(ang)+yg.*cos(ang);
xrot=xrot+x0; yrot=yrot+y0;

zrot=griddata(X,Y,bath,xrot,yrot);
```

B. CREATE MMPE INPUT FILE SCRIPT

```
icnt=0;
for iy=1:250,
for ix=1:250,
icnt=icnt+1;
znew(icnt)=zrot(iy,ix);
end
end
figure;plot(zrot(1,:));
figure;plot(znew(1:250));
data=[250 250 xnew ynew -znew]';
save -ascii MBay_MMPE_SW.inp data
```

APPENDIX C. PROCESSING CODE

C. CHIRP TIME FUNCTION

```
% clear
% filin=input('Enter a filename to process: ','s');
function chirptime(filin,filout)

eval(['load ' filin]);
if exist('press')==1
    data=press;
elseif exist('apvx')==1
    data=apvx;
elseif exist('apvy')==1
    data=apvy;
elseif exist('apvz')==1
    data=apvz;
end

nz=size(data,1);
nf0=size(data,2); df=freq(2)-freq(1);
cfreq=freq(nf0/2+1); ncf=floor(cfreq/df);
freq0=freq;

nf=2*(ncf+nf0/2);
lnf=floor(log10(nf)/log10(2.)+0.99); nf=2^lnf;
freq=-nf/2*df:df:(nf/2-1)*df;

dt0=1/(max(freq0)-min(freq0));
time0=-nf0*dt0/2:dt0:(nf0/2-1)*dt0;
DT=1/(max(freq)-min(freq));
time=-nf*DT/2:DT:(nf/2-1)*DT;

hanwinf=hanning(nf0+1);
hanwinf=hanwinf(1:nf0)/sum(hanwinf(1:nf0));

cr=(max(freq0)-min(freq0))/(max(time0)-min(time0))/2;
phs=2*pi*(min(freq0)*time0+cr/2*time0.^2);
chirpwinf=hanwinf.*exp(-1i*phs');

dataf=zeros(size(data));
datat=zeros(size(data,1),nf);

for iz=1:nz,

    dataf(iz,:)=data(iz,:).*chirpwinf';

    datat(iz,(nf/2+ncf-nf0/2+1):(nf/2+ncf+nf0/2))=dataf(iz,:);

    datat(iz,:)=fftshift(fft(fftshift(datat(iz,:))));
```

```
end

figure;imagesc(time,dep,20*log10(abs(datat)))
% filout=input('Enter a filename to save: ','s');
if exist('press')==1
    presst=datat;
    eval(['save ' filout ' time freq dep presst']);
elseif exist('apvx')==1
    apvxt=datat;
    eval(['save ' filout ' time freq dep apvxt']);
elseif exist('apvy')==1
    apvyt=datat;
    eval(['save ' filout ' time freq dep apvyt']);
elseif exist('apvz')==1
    apvzt=datat;
    eval(['save ' filout ' time freq dep apvzt']);
end

clear
```

D. CHIRP TIME SCRIPT

```
chirptime('pr_xp1km_y0km','prct_xp1km_y0km');
chirptime('pr_x1km_y0km','prct_x1km_y0km');
chirptime('pr_x2p5km_y0km','prct_x2p5km_y0km');
chirptime('pr_x5km_y0km','prct_x5km_y0km');
chirptime('vx_xp1km_y0km','vxct_xp1km_y0km');
chirptime('vx_x1km_y0km','vxct_x1km_y0km');
chirptime('vx_x2p5km_y0km','vxct_x2p5km_y0km');
chirptime('vx_x5km_y0km','vxct_x5km_y0km');
chirptime('vy_xp1km_y0km','vyct_xp1km_y0km');
chirptime('vy_x1km_y0km','vyct_x1km_y0km');
chirptime('vy_x2p5km_y0km','vyct_x2p5km_y0km');
chirptime('vy_x5km_y0km','vyct_x5km_y0km');
chirptime('vz_xp1km_y0km','vzct_xp1km_y0km');
chirptime('vz_x1km_y0km','vzct_x1km_y0km');
chirptime('vz_x2p5km_y0km','vzct_x2p5km_y0km');
chirptime('vz_x5km_y0km','vzct_x5km_y0km');
```

E. CHIRP INTENSITY SCRIPT

```
% clear
% filid=input('Enter a filename identifier (e.g., after "prct_" or
"vxct_"): ','s');
filpr=['prct_' filid];
filvx=['vxct_' filid];
filvy=['vyct_' filid];
filvz=['vzct_' filid];

eval(['load ' filpr]);
eval(['load ' filvx]);
```

```
eval(['load ' filvy]);
eval(['load ' filvz]);

Jx=real(presst).*real(apvxt);
Jy=real(presst).*real(apvyt);
Jz=real(presst).*real(apvzt);

theta=atan2d(Jz,sqrt(Jx.^2+Jy.^2));
phi=atan2d(Jx,Jy);

filout=['Jct_' filid];
eval(['save ' filout ' Jx Jy Jz theta phi time dep']);

clear
```

F. HANNING TIME FUNCTION

```
% clear
% filin=input('Enter a filename to process: ','s');
function hantime(filin,filout)

eval(['load ' filin]);
if exist('press')==1
    data=press;
elseif exist('apvx')==1
    data=apvx;
elseif exist('apvy')==1
    data=apvy;
elseif exist('apvz')==1
    data=apvz;
end

nz=size(data,1);
nf0=size(data,2); df=freq(2)-freq(1);
cfreq=freq(nf0/2+1); ncf=floor(cfreq/df);
freq0=freq;

nf=2*(ncf+nf0/2);
lnf=floor(log10(nf)/log10(2.)+0.99); nf=2^lnf;
freq=[-nf/2*df:df:(nf/2-1)*df];

DT=1/(max(freq)-min(freq));
time=[-nf*DT/2:DT:(nf/2-1)*DT];


hanwinf=hanning(nf0+1);
hanwinf=hanwinf(1:nf0)/sum(hanwinf(1:nf0));

dataf=zeros(size(data));
datat=zeros(size(data,1),nf);

for iz=1:nz,
```

```
    dataf(iz,:)=data(iz,:).*hanwinf';

    datat(iz,(nf/2+ncf-nf0/2+1):(nf/2+ncf+nf0/2))=dataf(iz,:);

    datat(iz,:)=fftshift(fft(fftshift(datat(iz,:))));

end

figure;imagesc(time,dep,20*log10(abs(datat)))
% filout=input('Enter a filename to save: ','s');
if exist('press')==1
    presst=datat;
    eval(['save ' filout ' time freq dep presst']);
elseif exist('apvx')==1
    apvxt=datat;
    eval(['save ' filout ' time freq dep apvxt']);
elseif exist('apvy')==1
    apvyt=datat;
    eval(['save ' filout ' time freq dep apvyt']);
elseif exist('apvz')==1
    apvzt=datat;
    eval(['save ' filout ' time freq dep apvzt']);
end

clear
```

G. HANNING TIME SCRIPT

```
hantime('pr_xp1km_y0km','prt_xp1km_y0km');
hantime('pr_x1km_y0km','prt_x1km_y0km');
hantime('pr_x2p5km_y0km','prt_x2p5km_y0km');
hantime('pr_x5km_y0km','prt_x5km_y0km');
hantime('vx_xp1km_y0km','vxt_xp1km_y0km');
hantime('vx_x1km_y0km','vxt_x1km_y0km');
hantime('vx_x2p5km_y0km','vxt_x2p5km_y0km');
hantime('vx_x5km_y0km','vxt_x5km_y0km');
hantime('vy_xp1km_y0km','vyt_xp1km_y0km');
hantime('vy_x1km_y0km','vyt_x1km_y0km');
hantime('vy_x2p5km_y0km','vyt_x2p5km_y0km');
hantime('vy_x5km_y0km','vyt_x5km_y0km');
hantime('vz_xp1km_y0km','vzt_xp1km_y0km');
hantime('vz_x1km_y0km','vzt_x1km_y0km');
hantime('vz_x2p5km_y0km','vzt_x2p5km_y0km');
hantime('vz_x5km_y0km','vzt_x5km_y0km');
```

H. HANNING INTENSITY SCRIPT

```
% clear
% filid=input('Enter a filename identifier (e.g., after "prt_" or
"vxt_"): ','s');
filpr=['prt_' filid];
filvx=['vxt_' filid];
filvy=['vyt_' filid];
filvz=['vzt_' filid];

eval(['load ' filpr]);
eval(['load ' filvx]);
eval(['load ' filvy]);
eval(['load ' filvz]);

Jx=real(presst).*real(apvxt);
Jy=real(presst).*real(apvyt);
Jz=real(presst).*real(apvzt);

theta=atan2d(Jz,sqrt(Jx.^2+Jy.^2));
phi=atan2d(Jx,Jy);

filout=['Jt_' filid];
eval(['save ' filout ' Jx Jy Jz theta phi time dep']);
```

C. INTENSITY PROCESSING SCRIPT

```
clear;
figure(95);close(95);
figure(97);close(97);
figure(100);close(100);

filin=input('Enter a file designator to process (e.g., "_x1km_y0km"):
','s');
filpr=['press' filin];
filvx=['apvx' filin];
filvy=['apvy' filin];
filvz=['apvz' filin];
% filpr=['pr' filin];
% filvx=['vx' filin];
% filvy=['vy' filin];
% filvz=['vz' filin];
eval(['load ' filpr]);
eval(['load ' filvx]);
eval(['load ' filvy]);
eval(['load ' filvz]);

disp(' ');
sigtype=input('Enter 1 for Hanning, 2 for Chirp: ');
disp(' ');

nf0=size(press,2); df=freq(2)-freq(1);
cfreq=freq(nf0/2+1); ncf=floor(cfreq/df);
```

```
freq0=freq;

nf=2*(ncf+nf0/2);
lnf=floor(log10(nf)/log10(2.)+0.99); nf=2^lnf;
freq1=[-nf/2*df:df:(nf/2-1)*df];

dt0=1/(max(freq0)-min(freq0));
time0=-nf0*dt0/2:dt0:(nf0/2-1)*dt0;
DT=1/(max(freq1)-min(freq1));
time=[-nf*DT/2:DT:(nf/2-1)*DT];

hanwinf=hanning(nf0+1);
hanwinf=hanwinf(1:nf0)/sum(hanwinf(1:nf0));

if sigtype==2,
    ctim=input('Enter time extent of chirp (sec): ');
%     cr=(max(freq0)-min(freq0))/(max(time0)-min(time0))/2;
    cr=(max(freq0)-min(freq0))/ctim;
    phs=2*pi*(min(freq0)*(time-min(time))+cr/2*(time-min(time)).^2);

    chirp=sin(phs);
    chirp(find(time>(ctim+min(time))))=0;
    chhan=zeros(size(chirp));
    chhan(1:floor(ctim/DT))=hanning(floor(ctim/DT))';
    chirp=chirp.*chhan;
    chirpf=fftshift(ifft(fftshift(chirp)));
    chirpf=chirpf/max(abs(chirpf));

end

pressf=zeros(size(press));
prt=zeros(size(press,1),nf);
apvxf=zeros(size(apvx));
vxt=zeros(size(apvx,1),nf);
apvyf=zeros(size(apvy));
vyt=zeros(size(apvy,1),nf);
apvzf=zeros(size(apvz));
vzt=zeros(size(apvz,1),nf);

for ii=1:size(press,1),

    pressf(ii,:)=press(ii,:).*hanwinf';
    prt(ii,(nf/2+ncf-nf0/2+1):(nf/2+ncf+nf0/2))=pressf(ii,:);
    if sigtype==2
        prt(ii,:)=prt(ii,:).*chirpf;
    end
    prt(ii,:)=fftshift(fft(fftshift(prt(ii,:))));

    apvxf(ii,:)=apvx(ii,:).*hanwinf';
    vxt(ii,(nf/2+ncf-nf0/2+1):(nf/2+ncf+nf0/2))=apvxf(ii,:);
    if sigtype==2
        vxt(ii,:)=vxt(ii,:).*chirpf;
    end
    vxt(ii,:)=fftshift(fft(fftshift(vxt(ii,:))));
```

```
    apvyf(ii,:)=apvy(ii,:).*hanwinf';
    vyt(ii,(nf/2+ncf-nf0/2+1):(nf/2+ncf+nf0/2))=apvyf(ii,:);
    if sigtype==2
        vyt(ii,:)=vyt(ii,:).*chirpf;
    end
    vyt(ii,:)=fftshift(fft(fftshift(vyt(ii,:))));

    apvzf(ii,:)=apvz(ii,:).*hanwinf';
    vzt(ii,(nf/2+ncf-nf0/2+1):(nf/2+ncf+nf0/2))=apvzf(ii,:);
    if sigtype==2
        vzt(ii,:)=vzt(ii,:).*chirpf;
    end
    vzt(ii,:)=fftshift(fft(fftshift(vzt(ii,:))));

end

if exist('dep','var')
    jet2=jet;
    jet2(1,:)=[0 0 0];
    figure(95);imagesc(time+2,dep,20*log10(abs(prt)));
    colmax=max(max(20*log10(abs(prt))));colmin=colmax-60;
    caxis([colmin colmax]); colormap(jet2);colorbar;
    xlabel('Time (sec)');ylabel('Depth (m)');

    disp(' ');
    dosh=input('Is time shift desired? ','s');
    while dosh(1)=='y' | dosh(1)=='Y'
        timsh=input('Enter time shift (sec): ');
        binsh=floor(timsh/(time(2)-time(1)));
        prt=circshift(prt,[0 binsh]);
        vxt=circshift(vxt,[0 binsh]);
        vyt=circshift(vyt,[0 binsh]);
        vzt=circshift(vzt,[0 binsh]);
        figure(95);imagesc(time+2,dep,20*log10(abs(prt)));
        colmax=max(max(20*log10(abs(prt))));colmin=colmax-60;
        caxis([colmin colmax]); colormap(jet2);colorbar;
        xlabel('Time (sec)');ylabel('Depth (m)');
        disp(' ');
        dosh=input('Shift again? ','s');
        disp(' ');
    end

    depout=input('Enter depth (in meters positive downward) to extract
data: ');
    disp(' ');

    nz=size(press,1);
    if depout < dep(1) | depout > dep(nz)
        disp('Requested depth not contained in input data file.');
        disp(' ');
    else
        nzout=find((dep-depout)>=0);
    end
    if isempty(nzout)
        if depout<0
```

```
          nzout=1;
        else
          nzout=nz;
        end
    else
        nzout=nzout(1);
    end
    depout=dep(nzout);
    disp(['Outputting data for depth ' num2str(dep(nzout)) 'm.']);
    disp(' ');

    pr=prt(nzout,:);
    vx=vxt(nzout,:);
    vy=vyt(nzout,:);
    vz=vzt(nzout,:);

elseif exist('crng','var')
    jet2=jet;
    jet2(1,:)=[0 0 0];
    figure(95);imagesc(crng,time+2,20*log10(abs(prt')));
    colmax=max(max(20*log10(abs(prt))));colmin=colmax-60;
    caxis([colmin colmax]); colormap(jet2);colorbar;
    xlabel('Cross-range (km)');ylabel('Time (sec)');

    disp(' ');
    dosh=input('Is time shift desired? ','s');
    while dosh(1)=='y' | dosh(1)=='Y'
        timsh=input('Enter time shift (sec): ');
        binsh=floor(timsh/(time(2)-time(1)));
        prt=circshift(prt,[0 binsh]);
        vxt=circshift(vxt,[0 binsh]);
        vyt=circshift(vyt,[0 binsh]);
        vzt=circshift(vzt,[0 binsh]);
        figure(95);imagesc(crng,time+2,20*log10(abs(prt')));
        colmax=max(max(20*log10(abs(prt))));colmin=colmax-60;
        caxis([colmin colmax]); colormap(jet2);colorbar;
        xlabel('Cross-range (km)');ylabel('Time (sec)');
        disp(' ');
        dosh=input('Shift again? ','s');
        disp(' ');
    end



    yout=input('Enter cross-range (in km) to extract data: ');
    disp(' ');

    ny=size(press,1);
    if yout < crng(1) | yout > crng(ny)
        disp('Requested cross-range not contained in input data
file.');
        disp(' ');
    else
        nyout=find((crng-yout)>=0);
    end
```

```
    if isempty(nyout)
        if yout<0
          nyout=1;
        else
          nyout=ny;
        end
    else
        nyout=nyout(1);
    end
    yout=crng(nyout);
    disp(['Outputting data for cross-range ' num2str(crng(nyout))
'km.']);
    disp(' ');

    pr=prt(nyout,:);
    vx=vxt(nyout,:);
    vy=vyt(nyout,:);
    vz=vzt(nyout,:);

end

% run a sliding Hanning window of length X secs an Y secs steps, and
% process over a band BW Hz centered at FC Hz
% (note, overlap is X-Y secs)
fs=1024.;           % Sampling bandwidth
Nt0=length(pr);     % Number of pressure samples
% X=0.25;              % Processing Window
% Y=X/4.;             % Overlap Time
X=input('Enter time of processing window in secs (e.g., .5): ');
Y=input('Enter overlap time in secs (e.g., .25): ');
FC=400.;            % Center Frequency
BW=128.;            % Bandwidth
ntmax=floor(((Nt0-1)/fs-X)/Y);  % Number of Samples
disp(['Number of samples to average = ' num2str(ntmax)]);

freqlow=FC-BW/2;
freqhigh=FC+BW/2;

Nt=round(X*fs);
Hfltr=hann(Nt+1);
freq=[-fs/2:(1/X):(fs/2)]';

nt=0;
ntdisp=1;

nf1=min(find(freq>(FC-BW/2)))-1;
nf2=min(find(freq>(FC+BW/2)))-1;
nftot=nf2-nf1+1;
freqrangets=freq(nf1:nf2);

% Computes the Intensity response to signals.
% Calculate max(dirmag)
while ((nt*Y*fs+1+Nt) <= Nt0)

    istrt=round(nt*Y*fs+1);
```

```
    p=real(pr(istrt:istrt+Nt))';
    x=real(vx(istrt:istrt+Nt))';
    y=real(vy(istrt:istrt+Nt))';
    z=real(vz(istrt:istrt+Nt))';

    % transform to freq domain
    pf=fftshift(fft(fftshift(p.*Hfltr)));
    xf=fftshift(fft(fftshift(x.*Hfltr)));
    yf=fftshift(fft(fftshift(y.*Hfltr)));
    zf=fftshift(fft(fftshift(z.*Hfltr)));

    % take only the positive freq values within the bandwidth
    xf=xf(nf1:nf2);
    yf=yf(nf1:nf2);
    zf=zf(nf1:nf2);
    pf=pf(nf1:nf2);

    nt=nt+1;

    % for I=.5Re(pu*)
    dirx(nt)=mean(real((pf).*conj(xf)));
    diry(nt)=mean(real((pf).*conj(yf)));
    dirz(nt)=mean(real((pf).*conj(zf)));
    dirmag(nt)=sqrt(dirx(nt).^2+diry(nt).^2+dirz(nt).^2);

    tim(nt)=(istrt+Nt/2)/fs;

end

ntmax=nt;
dirmagmax=max(dirmag);

for nt=1:ntmax,
    msize=dirmag(nt)/dirmagmax;
    msizefac=.8;

    MAP=[dirx(nt);diry(nt);dirz(nt)];
    theta_prime(nt)=acosd(max(min(MAP(3)/dirmag(nt),1),-1));

phi_prime(nt)=acosd(max(min(MAP(1)/(dirmag(nt)*sind(theta_prime(nt))),1
),-1));

    figure(100);%subplot(1,2,1);
    hold
on;plot(theta_prime(nt),(tim(nt)),'o','MarkerSize',20*msize^msizefac);v
=axis;v(1)=0;v(2)=180;v(3)=0;v(4)=4;axis(v);axis ij;xlabel('Bearing
(deg)');ylabel('Time (sec)');
    %figure(100);subplot(1,2,2);hold
on;plot(phi_prime(nt),(tim(nt)),'o','MarkerSize',20*msize^msizefac);v(1
)=0;v(2)=360;v(3)=0;v(4)=4;axis(v);axis ij;

    % Plot arrow graph
    %qx(nt)=sind(phi_prime(nt))*msize^msizefac*0.9;
    %qy(nt)=cosd(phi_prime(nt))*msize^msizefac*0.9;
```

```
    % Record the movie
    %figure(97);hold on;mfset(nt)=quiver(0,0,qx(nt),qy(nt),'b');
    %title(sprintf('Time %2.4f',tim(nt)));
    %axis([-1 1 -1 1]);

    %pause
end



data = press;

nf=nf0;
freq=freq0;
ccr=yout;
al=0.4;

nyc=find((crng-ccr)>=0,1);
ccr=crng(nyc);
nymin=find((crng-(ccr-al/2))>=0,1);
nymax=find((crng-(ccr+al/2))>=0,1);
ny=nymax-nymin+1;
pow_y=ceil(log10(ny)/log10(2));
ny=2^(pow_y);
nymin=nyc-ny/2; nymax=nyc+ny/2-1;
disp(['Computing response of ' num2str(ny) ' element array from '
num2str(crng(nymin)) 'km to ' num2str(crng(nymax)) 'km.']);

dtime=1/(max(freq)-min(freq));
time=[-nf/2*dtime:dtime:(nf/2-1)*dtime];

dely=1000*(crng(2)-crng(1));
delk=2*pi/(1000*(crng(nymax)-crng(nymin)));
ky=[-ny/2*delk:delk:(ny/2-1)*delk];
c0=1500;

hanwinf=hanning(nf+1);
hanwinf=hanwinf(1:nf)/sum(hanwinf(1:nf));
cfreq=freq(nf/2+1);
phs=2*pi*cfreq*time;

yvsfreq=zeros(ny,nf);
for iy=nymin:nymax,
    yvsfreq(iy-nymin+1,:)=data(iy,:).*hanwinf';
end

hanwiny=hanning(ny+1);
hanwiny=hanwiny(1:ny)/sum(hanwiny(1:ny));
theta=[-90:0.5:90]*pi/180;
angvsfreq=zeros(length(theta),nf);

kvsfreq=zeros(ny,nf);

for n=1:nf,
    kvsfreq(:,n)=fftshift(fft(fftshift(yvsfreq(:,n).*hanwiny)));
```

```
    kyb=2*pi*freq(n)*sin(theta)/c0;
    angvsfreq(:,n)=interp1(ky,kvsfreq(:,n),kyb);
end

%  Transform to time domain
for m=1:length(theta),

pressbeam(m,:)=fftshift(fft(fftshift(angvsfreq(m,:).*hanwinf'))).*exp(-
j*phs);
end

theta=theta*180/pi;

disp(' ');
    dosh=input('Was arrival time shift desired? ','s');
    while dosh(1)=='y' | dosh(1)=='Y'
        timsh=input('Enter first time shift (sec): ');
        binsh=floor(timsh/(time(2)-time(1)));
        pressbeam=circshift(pressbeam,[0 binsh]);
        disp(' ');
        dosh=input('Shift again? ','s');
        disp(' ');
    end

tlpressbeam=20*log10(max(abs(pressbeam),1.e-20));
tlmax=max(max(tlpressbeam));tlmin=tlmax-60;

%figure;imagesc(time,theta,tlpressbeam);caxis([tlmin
tlmax]);colormap(flipud(jet));
jet2=jet;
jet2(1,:)=[0 0 0];
figure;imagesc(theta,time+2,tlpressbeam');caxis([tlmin tlmax]);
colormap(jet2);colorbar;xlabel('Bearing (deg)');ylabel('Time (sec)')
```

LIST OF REFERENCES

[1] B. A. Cray and A. H. Nuttall, "Directivity factors for linear arrays of velocity sensors," *J. Acoust. Soc. Am.*, no. 110, pp. 324–331, July 2001.

[2] F. B. Jensen, W. A. Kuperman, M. B. Porter and H. Schmidt, *Computational Ocean Acoustics*, 2nd ed. New York, NY: Springer, 2011.

[3] L. E. Kinsler, A. R. Frey, A. B. Coppens and J. V. Sanders, *Fundamentals of Acoustics*, 4th ed. New York, NY: John Wiley & Sons, 1999.

[4] I. J. A. Mann, J. Tichy and A. J. Romano, "Instantaneous and time-averaged energy transfer in acoustic fields," *J. Acoust. Soc. Am.*, no. 82, pp. 17–30, July 1987.

[5] J.M. Upshaw, Bearing Estimation from a Single Vector Sensor Mounted on an Underwater Autonomous Glider, *M.S. Thesis, Dept. of Physics, Naval Postgrad. School*, Monterey, CA, 2013.

[6] K. B. Smith, "Convergence, stability, and variability of shallow water acoustic predictions using a split-step Fourier parabolic equation model," *Shallow Water Acoustic Modeling Workshop*, Monterey, CA, 1999.

[7] K.B. Smith and J.A. Colosi, "Effects of Solitions on Acoustic Energy Flow in Three Dimensions," *155th Meeting of the Acoustical Society of America*, Paris, France, 2008.

[8] K.B. Smith and F.D. Tappert, "UMPE: The University of Miami Parabolic Equation Model, Version 1.0," Marine Physical Laboratory Technical Memo 432, 1993.

[9] F.D. Tappert, "The parabolic approximation method," in Lecture Notes in Physics, Vol. 70, *Wave Propagation and Underwater Acoustics*, edited by J. B. Keller and J. S. Papadakis, Springer-Verlag, New York, pp. 224–287, 1977.

[10] R.H. Hardin and F.D. Tappert, "Applications of the split-step Fourier method to the numerical solution of nonlinear and variable coefficient wave equations," *SIAM* Rev. 15, p. 423, 1973.

[11] D. Lee, G. Botseas and J.S. Papadakis, "Finite-difference solution to the parabolic wave equation," *J. Acoust. Soc. Am.* 70, pp. 795–800, 1981.

[12] D.L. Divins and D. Metzger, NGDC Coastal Relief Model, Monterey Bay, http://www.ngdc.noaa.gov/mgg/coastal/coastal.html. Download provided by the Southern California Coastal Ocean Observing System, http://sccoos.org/data/bathy.

[13] F.B. Jensen, W.A. Kuperman, M.B. Porter and H. Schmidt, *Computational Ocean Acoustics*. New York, NY: Sheridan Books, 2000.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California