IMPLEMENTATION AND EVALUATION OF DISPERSION-INVARIANT FEATURES FOR SIGNAL CLASSIFICATION

by

Greg Okopal

B.S., Villanova University, 2002

Submitted to the Graduate Faculty of

the School of Engineering in partial fulfillment

of the requirements for the degree of

Master of Science

University of Pittsburgh

2007

UNIVERSITY OF PITTSBURGH

SCHOOL OF ENGINEERING

This thesis was presented

by

Greg Okopal

It was defended on

December 14, 2006

and approved by

P. J. Loughlin, Ph. D., Professor

J. R. Boston, Ph. D., Professor, Associate Chairman, Undergraduate Coordinator

A. A. El-Jaroudi, Ph. D., Associate Professor

Thesis Advisor: P. J. Loughlin, Ph. D., Professor

ABSTRACT

IMPLEMENTATION AND EVALUATION OF DISPERSION-INVARIANT FEATURES FOR SIGNAL CLASSIFICATION

Greg Okopal, M.S.

University of Pittsburgh, 2007

When a sound wave interacts with an object, the acoustic energy may excite resonances in the object corresponding to its natural modes of vibration. The backscattered wave will then contain information which can be used to distinguish among different objects. As the wave propagates, it can be changed by the propagation channel, which complicates automatic classification of the echo. For example, in a dispersive channel, the duration of the wave increases with propagation distance. Our goal is to identify features of propagating waves that may be used for automatic classification.

In this work, we implement and test a class of moment-like features that are invariant to specific propagation effects, in particular dispersion. Our tests of the classification utility of the "dispersion-invariant moments" (DIMS) are performed on numerical models of dispersive propagation and acoustic scattering from steel shells. We consider the case of real dispersion relations and in the conclusion discuss the implementation of complex dispersion and a future direction for research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1.0 INTRODUCTION

The primary tool for the exploration of, and communication in, underwater environments is **SONAR** (**SO**und **N**avigation **A**nd **R**anging) because sound waves propagate far more efficiently in water than do electromagnetic waves. Modern sonar systems can be classified into two main types: passive sonar and active sonar. Passive sonar simply listens to the ocean environment and attempts to identify entities in the sea by the sounds emanating from them. Active sonar creates a sound in the water via an explosive source or transducer, and then listens for acoustic energy that is reflected back. Characteristics of the echo may be used to identify entities in the sea because the interaction of acoustic energy with an object is unique to the object's size, shape, and material composition.

Sounds propagating through the ocean may exhibit nonstationarities, that is, time- or spatially-varying spectral changes. These changes complicate the task of identifying the object that reflected the sound. In shallow water channels, *dispersion* is one of the primary phenomena that affects propagating sound. Dispersion means that acoustic energy at different frequencies propagates at different velocities, resulting in a spatial and temporal spreading of the sound wave. A significant consequence of dispersion is that the wave changes as it propagates, and hence observers at two different locations will receive two different waves even if they had a common origin.

The ocean acoustic environment is essentially a complicated waveguide, bounded by the surface of the water and the seafloor, with a non-homogeneous propagation medium in between. In order to study a specific environment, we use simplified waveguide models, such as the parallel-plate ideal fluid waveguide, or the slightly more complicated Pekeris waveguide.

It is a primary interest of the US Navy to develop and improve methods for automatic classification of sound in the sea, obtained via active or passive sonar, so that the fleet may be aware of potential threats lurking in the water. Human listeners were the original sonar classifiers, but for many reasons (e.g., fatigue, limits of human hearing) it is desirable to assist them with an automated system. Two main tasks are involved in automatic classification: development of a machine learning algorithm, and feature selection. Our concern here lies in feature selection, which is the art of choosing quantifiable characteristics of the received sound wave that are robust to environmental and other factors not related to the scattering of the wave by the object of interest in order to facilitate easy identification of the entity from which the wave has been reflected.

Statistical moments, computed on the time series or the spectrum of an echo, constitute a standard set of features used in sonar classification. Moments quantify such characteristics as duration, skewness, bandwidth, and kurtosis. These features do not account for the physics of ocean acoustic propagation, and therefore the effects of the ocean environment may alter the values of the moments. Thus, the moments may reflect qualities of the environment, which is inherently detrimental to classification, since the goal is to identify the source of the echo and not the environment through which it propagated. It is thereby desirable to use features which account for the physics of ocean acoustic propagation. Such features have recently been given [16] for classification in dispersive shallow-water environments, and it is the purpose of this work to implement and evaluate these features in a simulated ocean environment.

In this work, we implement and test a class of moment-like features that are invariant to specific propagation effects, in particular dispersion. Our tests of the classification utility of the "dispersion-invariant moments" (DIMS) are performed on numerical models of dispersive propagation and acoustic scattering from steel shells. We consider the case of real dispersion relations, and in the conclusion discuss the implementation of complex dispersion and a future direction for research.

2.0 BACKGROUND

2.1 THE WAVE EQUATION

The acoustic wave equation may be formulated in terms of pressure, velocity, or displacement and in any coordinate system. The general form is derived from basic equations of motion and is expressed as [9]

$$\nabla^2\Phi = \frac{1}{c^2}\frac{\partial^2\Phi}{\partial t^2} \tag{2.1}$$

where Φ is the wave, c is the speed of sound in the medium, and t is time. The Laplacian, ∇^2 is defined as

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \tag{2.2}$$

in Cartesian coordinates or

$$\nabla^2 = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2}{\partial \theta^2} + \frac{\partial^2}{\partial z^2} \tag{2.3}$$

in cylindrical coordinates. The Cartesian (x, y, z) and cylindrical (r, θ, z) coordinates are related by

$$r = \sqrt{x^2 + y^2} \tag{2.4}$$

$$\theta = \tan^{-1}\left(\frac{y}{x}\right) \tag{2.5}$$

$$z = z. \tag{2.6}$$

The solution in Cartesian coordinates will be used in this work; the solution in cylindrical coordinates is given in the Appendix. In order to simplify the solution to the wave equation, it is often assumed that the speed of sound is either constant or a function of only the depth coordinate, z. Such an environment is called *horizontally stratified*. Propagation from source

to receiver occurs in the two-dimentional $x-z$ plane. We then seek a solution of the form [21]

$$\Phi(x,z,t) = \phi(z)\, e^{j(k_x x - \omega t)}, \tag{2.7}$$

where k_x is the horizontal wavenumber. The vertical wavenumber is defined as k_z and the two are related to the wavenumber in the direction of propagation (k) by

$$k_x = k\sin\theta \tag{2.8}$$

$$k_z = k\cos\theta \tag{2.9}$$

$$k^2 = k_x^2 + k_z^2, \tag{2.10}$$

where $k = \frac{\omega}{c}$ and θ is the angle of k with respect to the z axis (Figure 1).

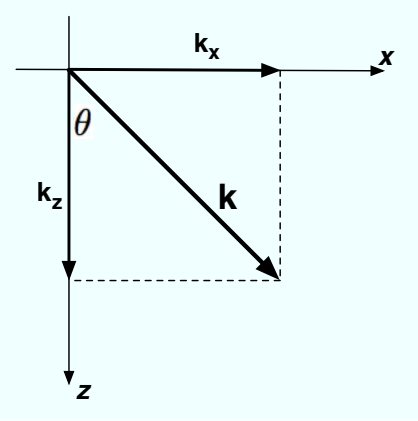

Figure 1: Components of the wavenumber in the direction of propagation.

Substituting Equation (2.7) into Equation (2.1) gives the Helmholtz equation [14],

$$\frac{\partial^2\phi}{\partial z^2} + k_z^2\phi = 0, \tag{2.11}$$

which does not depend upon time. If c and k_z are constant, the general solution is given by

$$\phi = C_1 e^{jk_z z} + C_2 e^{-jk_z z} = C_3\cos(k_z z) \tag{2.12}$$

where $C_{1,2}$ are arbitrary constants of integration. In a homogeneous medium, $C_1 = C_2$, and the expression in Equation (2.12) follows. The solution represents a vertical standing

wave pattern and, in the absence of boundary conditions, k_z is continuous and there are an infinite number of solutions to the Helmholtz equation (i.e., all vertical wavenumbers k_z are possible). These solutions are called modes and are analogous to the modes of a vibrating string.

The total time-dependent solution to the original wave equation (Eq. 2.1) in an infinite homogeneous medium is then given, per mode, by

$$\Phi(x, z, t) = C_3 \cos(k_z z)\, e^{j(k_x x - \omega t)}, \tag{2.13}$$

which shows how the modes propagate in the x direction in a time-dependent manner.

2.2 PROPAGATION IN WAVEGUIDES AND DISPERSION

Wave propagation in bounded media introduces boundary conditions on the wave equation (Eq. 2.1) and thus results in more complicated solutions. The ocean acoustic environment is often modeled as a waveguide, and the solution of the wave equation for specific types of waveguides can give insight into the behavior of a pulse propagating in the ocean. Two of the most common types of waveguides used in ocean acoustic modeling are the two-plate (ideal fluid) waveguide and the Pekeris waveguide.

2.2.1 Dispersion

Dispersion is a phenomenon in wave propagation whereby the velocity of propagation is dependent upon frequency. Two types of dispersion are relevant to ocean acoustics: geometrical dispersion and intrinsic dispersion. *Geometrical dispersion* is caused by waveguide-like structures, and the magnitude of the effect depends mostly upon the specific geometry of the environment. *Intrinsic dispersion* is caused by the frequency-dependent response of the medium and can be present even in a boundary-free environment. In shallow-water ocean acoustic environments, however, the effect of geometrical dispersion can be significant while the effect of intrinsic dispersion is negligible.

The concepts of group velocity and phase velocity can be helpful when describing dispersion. The *phase velocity* of a signal is the speed of propagation of points of constant phase and is given by [9]

$$v_m = \frac{\omega}{k_{xm}}. \tag{2.14}$$

Group velocity is the velocity at which variations in the envelope of a wave travel through the medium. It is given by [9]

$$u_m = \frac{\partial \omega}{\partial k_{xm}}. \tag{2.15}$$

The plot of group velocity versus frequency is called a *dispersion curve* and is used to characterize the dispersion for a given propagation environment (e.g., Fig. 2). For a wave propagating without dispersion, $k_{xm} = \frac{\omega}{c}$, and the group velocity is constant and independent of frequency. It follows that the wave propagates as

$$\Phi(x, z, t) = \phi_m(z)\, e^{jk_{xm}(x-ct)} \qquad \text{[no dispersion]} \tag{2.16}$$

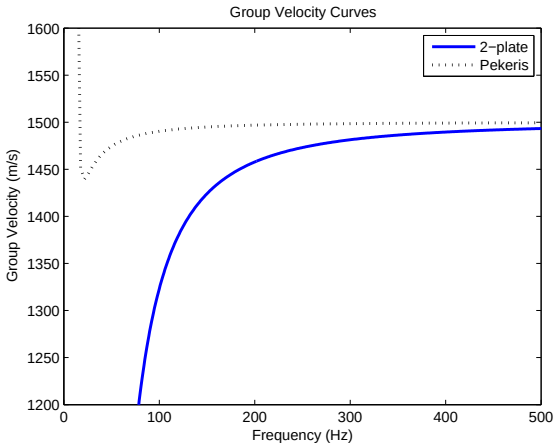

Figure 2: Group velocity curves for the two-plate and Pekeris waveguides, depth=100m.

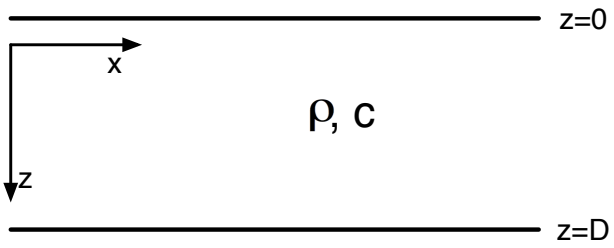

Figure 3: Schematic diagram of the two-plate waveguide.

2.2.2 Two-plate Waveguide

One of the simplest models of a shallow water acoustic propagation channel is the two-plate or ideal fluid waveguide. This model has a homogeneous, isovelocity fluid layer bounded by two perfectly reflecting parallel plates located at $z = 0$ and $z = D$ (Figure 3).

For the ideal fluid waveguide, the surface and bottom are modeled as pressure-release surfaces. Total reflection at these surfaces requires that the solution of the Helmholtz equation (Eq. 2.11) must go to zero at both boundaries. Accordingly, the solution is of the form

$$\phi_m(z) = A \sin(k_z z) \tag{2.17}$$

per mode m, where

$$k_z = \frac{m\pi}{D} \tag{2.18}$$

are the possible vertical wavenumbers, and A is a constant. Thus, not all wavenumbers are possible, and the spectrum is said to be *discrete*. The solution for $\phi_m(z)$ is a vertical standing wave with nodes at the boundaries (Figure 4).

The vertical wavenumber is related to the horizontal wavenumber and the wavenumber in the direction of propagation by Equation (2.10). In terms of the discrete spectrum, we can solve for the horizontal wavenumber (per mode m)

$$k_{xm}^2 = k^2 - \left(\frac{m\pi}{D}\right)^2 \tag{2.19}$$

$$k_{xm} = \frac{1}{c}\sqrt{\omega^2 - \omega_{0m}^2} \tag{2.20}$$

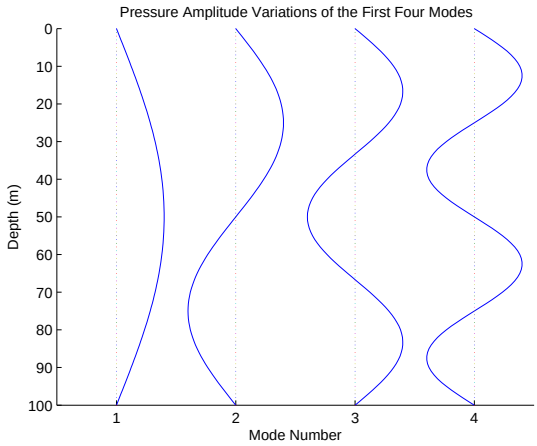

Figure 4: Plot of the first four mode shapes for a two-plate isovelocity waveguide, D=100 meters, pressure-release bottom.

where $\omega_{0m} = \frac{cm\pi}{D}$ is the low cutoff frequency for mode m. A mode whose cutoff frequency is greater than the source frequency is called an *evanescent mode*, the intensity of which decays exponentially with range [21]. Evanescent modes are not relevant to this work, as they do not propagate to significant ranges. Equation (2.20) is called the *dispersion relation* and it describes how spatial frequency k and radial frequency w are coupled. The dispersion relation may also be formulated in terms of ω as a function of wavenumber,

$$\omega = c\sqrt{k_{xm}^2 + k_{0m}^2}, \tag{2.21}$$

where $k_{0m} = \frac{m\pi}{D}$.

The total time-dependent solution for the acoustic field in a two-plate waveguide is given by

$$\Phi(x, z, t) = \phi_m(z)\, e^{j(k_{xm}(\omega)x - \omega t)}, \tag{2.22}$$

per mode, where $k_{xm}(\omega)$ is given by Equation (2.20) and $\phi(z)$ is given by Equations (2.17) and (2.18). The total wave at a given depth z is then obtained by summing the contribution from each of the modes [14].

Alternatively, the bottom may be modeled as a rigid, perfectly reflecting surface, instead of pressure-release. In this situation, the solution of the Helmholtz equation (Eq. 2.11) goes to zero at the surface and is at a maximum at the bottom. This solution is easily obtained by replacing the mode number m with $m - \frac{1}{2}$ in Equations (2.18), (2.20), and (2.21). (Fig. 5) [2]. This approach would be used to model a shallow water channel with a very rigid bottom.

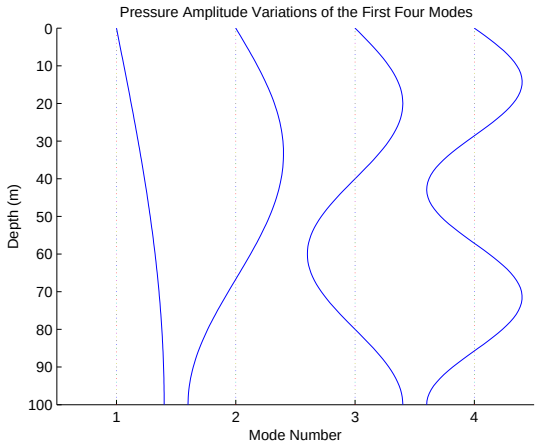

Figure 5: Plot of the first four mode shapes for a rigid bottom two-plate isovelocity waveguide, D=100 meters.

2.2.3 Pekeris Waveguide

Total internal reflection is a fairly good approximation for the behavior of sound waves acting upon the ocean surface, a fluid-air interface. For the ocean bottom, however, total reflection is usually a poor approximation, because the amount of energy transferred into the bottom is non-negligible, except for very hard bottoms.

Often the bottom is made up of wet silt or sand and possesses acoustic properties resembling a dense fluid. For this reason, a two-layered waveguide can be used to more accurately model propagation of sound in the ocean. This model, first introduced by Pekeris [7], consists of a water layer of finite depth, constant density, and constant sound speed atop an

infinitely deep fluid layer of constant density and sound speed (Figure 6). The lower fluid layer typically has a higher density and sound speed than the water layer. The acoustic source and receiver are located in the water layer.

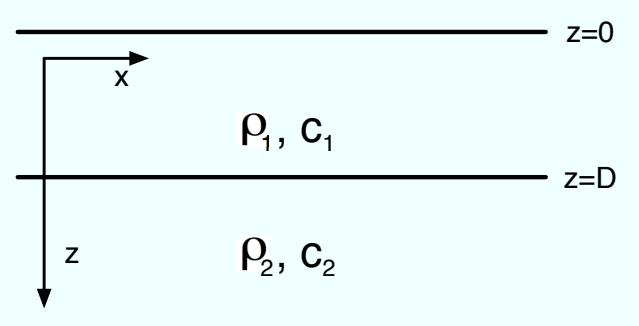

Figure 6: Schematic diagram of the Pekeris waveguide.

Acoustic energy interacting with the boundary of the two fluid layers, $z = D$, may either be totally reflected into the water layer or partially reflected and partially transmitted into the lower fluid layer. The potential for transmission is proportional to the respective densities of the layers, ρ_1 and ρ_2. In terms of the modes, the equations

$$\rho_1\phi_1 = \rho_2\phi_2 \tag{2.23}$$

$$\frac{\partial\phi_1}{\partial z} = \frac{\partial\phi_2}{\partial z} \tag{2.24}$$

describe the bounds on the transfer of energy from layer to layer.

For the Pekeris waveguide, two families of modes coexist. The first, referred to as *guided modes*, consists of modes trapped in the top layer (identical to the two-plate case) and decreasing exponentially in the bottom layer. The second, called *continuous-spectrum modes*, consists of modes which vary sinusoidally throughout the entire medium. The guided modes exhibit dispersive propagation [21].

The solution of the Helmholtz equation is subject to the boundary condition at the ocean surface, that is, at $z = 0$ the vertical standing wave must have a node. For the guided modes,

the modes decay exponentially in the lower fluid layer, $z > D$, and are subject to Equations (2.23) and (2.24); thus, the solution is formulated as [21]

$$\phi_m(z) = A \begin{cases} \sin(k_{z1m} z), & 0 \le z \le D \\ \frac{\rho_1}{\rho_2} \sin(k_{z1m} D)\, e^{j k_{z2m}(z-D)}, & z > D \end{cases} \tag{2.25}$$

In order to obtain the characteristic equation for the guided modes, we must examine the conditions at the boundaries. The total field near the boundary is a sum of the incident and reflected waves

$$\phi = e^{j k_z z} + R e^{-j k_z z} \tag{2.26}$$

where R is the coefficient of reflection for the boundary. Assuming total reflection at the free surface ($z = 0$), there can only be a phase change, and thus R is given by

$$R = R_s = -e^{2j\theta} \tag{2.27}$$

Evaluating Equation (2.26) for the boundary at $z = 0$, we obtain

$$\phi = 1 - e^{2j\theta} \tag{2.28}$$

By definition, the pressure must vanish, therefore,

$$\theta = m\pi. \tag{2.29}$$

At the fluid-fluid boundary ($z = D$), we employ the Rayleigh coefficient of reflection [2]

$$R_b = \frac{\rho_2 k_{z_1} - \rho_1 k_{z_2}}{\rho_2 k_{z_1} + \rho_1 k_{z_2}}, \tag{2.30}$$

which, using the trigonometric identity

$$\tan^{-1} t = \frac{1}{2j} \ln \frac{1 + jt}{1 - jt}, \tag{2.31}$$

is given as

$$R_b = e^{-2j \tan^{-1}\left(\frac{\rho_2 k_{z_1}}{\rho_1 k_{z_2}}\right)}. \tag{2.32}$$

The reflection coefficient at the bottom R_b is related to the reflection coefficient at the surface R_s by [21]

$$R_b = R_s e^{2j k_{z_1} D}. \tag{2.33}$$

Therefore, from Equations (2.27), (2.32), and (2.33) we obtain

$$\theta = k_{z1}D + \tan^{-1}\left(\frac{\rho_2}{\rho_1}\frac{k_{z1}}{jk_{z2}}\right) \tag{2.34}$$

at the bottom boundary. Thus, from Equations (2.29) and (2.34), the characteristic equation for the guided modes is

$$k_{z1}D + \tan^{-1}\left(\frac{\rho_2}{\rho_1}\frac{k_{z1}}{jk_{z2}}\right) = m\pi. \tag{2.35}$$

Using Equation (2.10), we can write the relations

$$k_{z1} = \sqrt{k^2 - k_{x1}^2} \tag{2.36}$$

$$jk_{z2} = \sqrt{k_{x2}^2 - k^2} \tag{2.37}$$

which can be substituted into Equation (2.35) to obtain the dispersion relation

$$k_{xm} = \frac{\frac{m\pi}{D} - \frac{1}{D}\tan^{-1}\left(\frac{\rho_2}{\rho_1}\sqrt{\frac{v_m^2/c_1^2 - 1}{1 - v_m^2/c_2^2}}\right)}{\sqrt{v_m^2/c_1^2 - 1}} \tag{2.38}$$

where v_m is the phase velocity of the mode, c_1 is the velocity of sound in the water layer, and c_2 is the velocity of sound in the sediment layer.

The propagation of the guided modes in a Pekeris waveguide is therefore dispersive. The total time-dependent solution for the acoustic field in a Pekeris waveguide is given by

$$\Phi(x, z, t) = \phi_m(z)\, e^{j(k_{xm}(\omega)x - \omega t)}, \tag{2.39}$$

per guided mode, where $k_{xm}(\omega)$ is given by Equation (2.38) and $\phi_m(z)$ is given by Equation (2.25).

3.0 CLASSIFICATION

The purpose of automatic signal classification in sonar applications is to identify the source of an echo without human input. Typically, the task involves distinguishing *targets*, such as submarines or mines, from *clutter*, such as rocks or schools of fish. Classification is complicated by the ocean environment because of ambient noise and propagation effects (such as attenuation and dispersion) that change the reflected sound as it propagates. Accordingly, different sounds may be due to different targets or different propagation environments. The latter is undesirable, motivating our efforts to obtain features that are (ideally) unaffected by propagation effects.

3.1 STATISTICAL LEARNING

Statistical learning uses a training set of data to build probability distributions of features that may be used to classify future signals [18]. The selection of features is fundamental to statistical classification of signals. A *feature* is a measurable property of a signal, such as bandwidth or signal-to-noise ratio (SNR). Other features describe the shape of a signal, such as skewness and kurtosis.

Ideally, in the chosen feature space, the distributions of the features for targets and clutter over the training set will be distinct from one another, allowing a classifier to determine an effective decision threshold. This threshold will then be used to classify future echoes as targets or clutter.

3.2 MOMENTS

Statistical moments may be used to numerically quantify characteristics of a traveling wave and have been widely used as features for signal detection and classification. Temporal moments numerically describe attributes of a wave such as duration, which is the length of a pulse; skewness, which measures the temporal symmetry of a wave; and kurtosis, which provides information about the shape of the wave in the "tails" (i.e., far away from the centroid). The concept of moments is not specific to wave propagation, and the general formulas for calculating the moments of a traveling wave do not account for specific propagation effects.

3.2.1 Propagation Effects on Temporal Moments

For ideal classification purposes, the wave received from a source should produce moment values which are similar in any propagation environment. In practice, of course, this is rarely the case, because the environment affects a propagating wave and changes it. Hence, it is likely that features derived from the wave will likewise change. Dispersion, frequency-dependent attenuation, and environmental noise are three of the most significant propagation effects which complicate detection and classification. In this paper, our focus is on dispersion and its effects upon temporal moments.

The spectrum of the wave given by $u(x,t)$ at $x = 0$ is

$$F(0,\omega) = \frac{1}{\sqrt{2\pi}} \int u(0,t)\, e^{j\omega t} dt \tag{3.1}$$

For propagation without attenuation, the spectrum of the wave at some other location x is given by

$$F(x,\omega) = F(0,\omega)\, e^{jk(\omega)x} \tag{3.2}$$

where $k(\omega)$ is the dispersion relation.

The temporal moments of a wave $u(x,t)$ at location x are given by [4]

$$\langle t^n \rangle_x = \int t^n \left| u(x,t) \right|^2 dt. \tag{3.3}$$

The *central* temporal moments (or temporal moments centered about the mean time) are given by

$$\mu_{t|x}^{n} = \int \left(t - \langle t \rangle_x\right)^n \left|u\left(x,t\right)\right|^2 dt, \tag{3.4}$$

which can also be written in the frequency domain as [16]

$$\mu_{t|x}^{n} = \int F^*\left(x,\omega\right) \left(j\frac{\partial}{\partial\omega} - \langle t \rangle_x\right)^n F\left(x,\omega\right) d\omega \tag{3.5}$$

To illustrate the effect of dispersion upon the temporal moments, we examine the classical duration moment, defined as the central temporal moment with $n = 2$:

$$\sigma_{t|x}^{2} = \int \left(t - \langle t \rangle_x\right)^2 \left|u\left(x,t\right)\right|^2 dt \tag{3.6}$$

which can be written in the frequency domain as [4]

$$\sigma_{t|x}^{2} = \int \left|\left(j\frac{\partial}{\partial\omega} - \langle t \rangle_x\right) F\left(x,\omega\right)\right|^2 d\omega. \tag{3.7}$$

Use the identity from Equation (3.2) to obtain

$$\sigma_{t|x}^{2} = \int \left|\left(j\frac{\partial}{\partial\omega} - \langle t \rangle_x\right) F\left(0,\omega\right) e^{jk(\omega)x}\right|^2 d\omega, \tag{3.8}$$

and simplify to arrive at [5]

$$\sigma_{t|x}^{2} = \int \left[\left|F'\left(0,\omega\right)\right|^2 + \left|-k'\left(\omega\right)x + \langle t \rangle_x\right|^2 \left|F\left(0,\omega\right)\right|^2\right] d\omega. \tag{3.9}$$

In a dispersive environment,

$$\langle t \rangle_x = \langle t \rangle_0 + \int \psi'\left(x,\omega\right) \left|F\left(0,\omega\right)\right|^2 d\omega \tag{3.10}$$

where $\psi\left(x,\omega\right)$ is the spectral phase. Thus, the value of the duration moment depends upon the x coordinate, which potentially increases the variability of this moment and, in turn, reduces its discriminatory utility.

If there is no dispersion, $k\left(\omega\right) \to k$, $k'\left(\omega\right)x \to \frac{x}{c}$,

$$\langle t \rangle_x = \langle t \rangle_0 + \frac{x}{c} \tag{3.11}$$

and the duration moment reduces to

$$\sigma_{t|x}^{2} = \int \left[\left|F'\left(0,\omega\right)\right|^2 + \left|\langle t \rangle_0 F\left(0,\omega\right)\right|^2\right] d\omega. \tag{3.12}$$

The right-hand side of the equation has no x dependence, and thus it is clear that $\sigma_{t|x}^{2} = \sigma_{t|0}^{2}$ in the absence of dispersion.

3.3 DISPERSION-INVARIANT MOMENTS

In dispersive environments, the velocity of propagating sound is dependent upon frequency, a phenonemenon which manifests itself as a spreading of energy in the time and spatial domains. In the time-frequency plane, however, dispersion does not induce time-spreading per frequency; instead, the energy at a given frequency is merely delayed by the quantity $t_g(x,\omega)$, which is known as the *group delay*. The group delay is given by

$$t_g(x,\omega) = -\psi'(x,\omega), \tag{3.13}$$

where $\psi(x,\omega)$ is the spectral phase of the wave at x. This property can be exploited to formulate new temporal moments which are invariant to dispersion, as has been shown by Loughlin and Cohen [16]. The formulation is similar to the central temporal moments given in Equation (3.5), except the moment is centered about the group delay instead of the mean time,

$$A_n(x) = \int F^*(x,\omega)\left(j\frac{\partial}{\partial\omega} - t_g(x,\omega)\right)^n F(x,\omega)\,d\omega. \tag{3.14}$$

The most important property of these moments is that in a dispersive envrionment with no damping (real dispersion relation), the value of the moment does not depend upon position per mode. To illustrate this property, we expand the spectrum of the wave at x into its amplitude and phase components, and make use of the identity [6]

$$\left(j\frac{\partial}{\partial\omega} + \psi'(x,\omega)\right)^n F(x,\omega) = j^n B^{(n)}(x,\omega)\,e^{j\psi(x,\omega)} \tag{3.15}$$

to obtain

$$\begin{aligned} A_n(x) &= \int B(x,\omega)\,e^{-j\psi(x,\omega)} j^n B^{(n)}(x,\omega)\,e^{j\psi(x,\omega)}d\omega && (3.16)\\ &= \int B(x,\omega)\,j^n B^{(n)}(x,\omega)\,d\omega && (3.17)\\ &= \int B(0,\omega)\,j^n B^{(n)}(0,\omega)\,d\omega && (3.18) \end{aligned}$$

where the final equation is true because the spectral amplitude is constant in the absence of damping. Thus, $A_n(x) = A_n(0)$, which means that the value of the moment is invariant to dispersion, per mode.

3.3.1 Implementation

Equation (3.18) may be written in operator notation as

$$A_n(x) = \int B(0,\omega) \left(j \frac{\partial}{\partial \omega} \right)^n B(0,\omega)\, d\omega \tag{3.19}$$

which can be transformed into the time-domain to obtain

$$A_n(x) = \int t^n |u_a(t)|^2\, dt \tag{3.20}$$

where u_a is a signal constructed from the spectral amplitude,

$$u_a(t) = \frac{1}{2\pi} \int |F(x,\omega)|\, e^{j\omega t} d\omega. \tag{3.21}$$

Since the dispersion relation is real, $|F(x,\omega)| = |F(0,\omega)|$, by which the invariance of the moments to dispersion follows. The procedure for calculating $A_n(x)$ is straightforward: calculate the magnitude spectrum, construct the signal u_a by inverse transforming to the time domain, and compute A_n as given in Equation (3.20).

The signal u_a is symmetric about zero; therefore, the value of A_n for odd values of n is identically zero, which offers no classification utility. This situation may be remedied by one of two approaches. Either replace time vector t in Equation (3.20) by its absolute value, or (equivalently) compute a one-sided integral by changing the lower bound of integration from $-\infty$ to 0. In this work, we computed one-sided moments.

4.0 IMPLEMENTATION OF CLASSIFICATION SIMULATION

This section describes some classification simulations conducted to compare the performance of the dispersion-invariant moments to the performance of traditional moments that are used for classification. The simulations were conducted using numerical methods to compute the sonar backscatter from air-filled steel shells and the subsequent propagation of the wave through a simulated dispersive shallow-water channel (Figure 7).

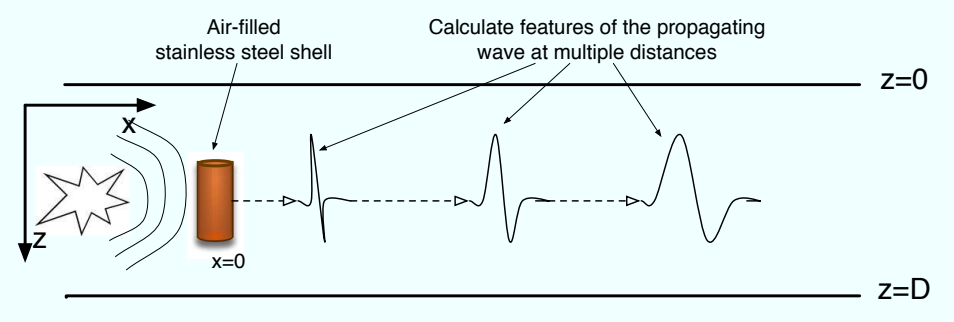

Figure 7: Graphical overview of the simulation.

The goal was to distinguish between two different shells (sphere vs. cylinder) from the moment features extracted from the propagated echoes. The shells were insonified by an impulsive interrogating waveform, and the backscattered pressure obtained by resonance scattering theory (RST) [8] was used as the initial waveform $u(0,t)$ in the channel at position $x = 0$. This initial backscattered wave was then propagated to several distances in a dispersive channel model.

The classification features extracted from each simulated propagated echo were ordinary central temporal moments given in Equation (3.5) for $n = 2, 3, 4$, and the corresponding

dispersion-invariant moments given in Equation (3.20). Receiver operating characteristic (ROC) curves were generated in order to compare classification performance of the two features.

The geometry of the shells (i.e., size and thickness) was chosen to represent real-world objects and to generate feature values which facilitate easy classification prior to propagation through a dispersive channel. For this simulation, the depth of the channel (D) was fixed at 50 meters, and the maximum propagation distance (x) was 15 kilometers. Feature values were calculated at 25 meter increments starting at $x = 25$, resulting in a feature vector of length 599. A feature vector for each moment was calculated for each shell, resulting in 12 total feature vectors per trial. The feature values were calculated from the first propagating mode in the waveguide ($m = 1$). The sampling frequency was fixed at 10 kHz. The inner and outer radii of the steel shells are given in Figure 8.

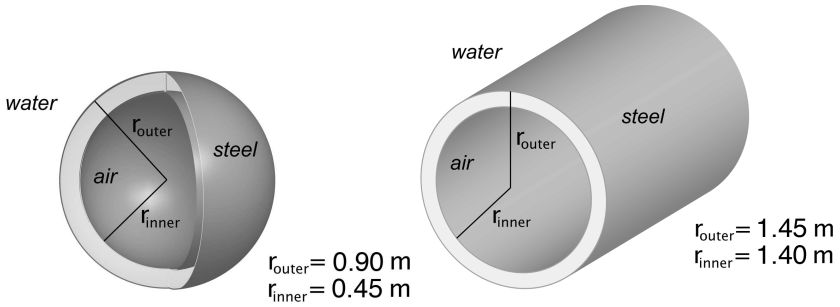

Figure 8: Diagrams of the sphere and cylinder.

4.1 RESONANCE SCATTERING THEORY

When an object is insonified, the acoustic energy may excite resonances in the object corresponding to its natural modes of vibration. Resonance scattering theory (RST) is a well-known method for computing the resonances present in objects that form simple geometric shapes. [8, 10, 12, 13, 22]

The total field is a sum of the incident and scattered fields,

$$\Phi\left(\mathbf{r}\right) = \Phi^{incident}\left(\mathbf{r}\right) + \Phi^{scattered}\left(\mathbf{r}\right) \tag{4.1}$$

where the incident field is a plane wave

$$\Phi^{incident}\left(\mathbf{r}\right) = p_0 e^{j(kr\cos\theta - \omega t)} \tag{4.2}$$

and, in the continuous-wave case, the scattered field is given by

$$\Phi^{scattered}\left(\mathbf{r}\right) = \frac{p_0}{\mathbf{r}} e^{-jx\tau} \frac{a}{2} \left[\sum_{n=0}^{\infty} \left(-1\right)^n \left(2n+1\right) T_n\left(x\right)\right] \tag{4.3}$$

where $x = ka$, and $T_n\left(x\right)$ is known as the T-matrix. The coefficients of the T-matrix for spherical and cylindrical shells are given in the Appendix. If the interrogating waveform is broadband, we make the substitution [11]

$$p_0 e^{-jx\tau} \rightarrow \frac{1}{2\pi} \int G\left(x'\right) e^{-jx'\tau} dx' \tag{4.4}$$

where $G(x')$ is the spectrum of the interrogating waveform. The backscattered pressure is then given by

$$\Phi^{scattered}\left(\mathbf{r}\right) = \frac{1}{2\pi} \int f_\infty\left(\pi, x'\right) G\left(x'\right) e^{-jx'\tau} dx' \tag{4.5}$$

where

$$f_\infty\left(\pi, x\right) = \frac{a}{jx} \sum_{n=0}^{\infty} \left(-1\right)^n \left(2n+1\right) T_n\left(x\right). \tag{4.6}$$

Implementation of Equation (4.6) clearly requires that a finite number of modes be used. We investigated an appropriate upper limit to use by examining the simulated waveforms for upper limits of n ranging from 1 to 50. Our simulations showed that for $n > 5$, the differences in the waveforms were slight; examples of simulated waveforms for a cylinder computed with $n = 5$ and $n = 25$ are shown in Figure 9, along with the respective spectrograms. Hence the upper limit was taken to be 5 in our implementation of Equation (4.6).

For the material properties given in Table 1 and the geometries given in Figure 8, the backscatter computed by RST (with $n = 5$) for each shell is shown in Figure 10.

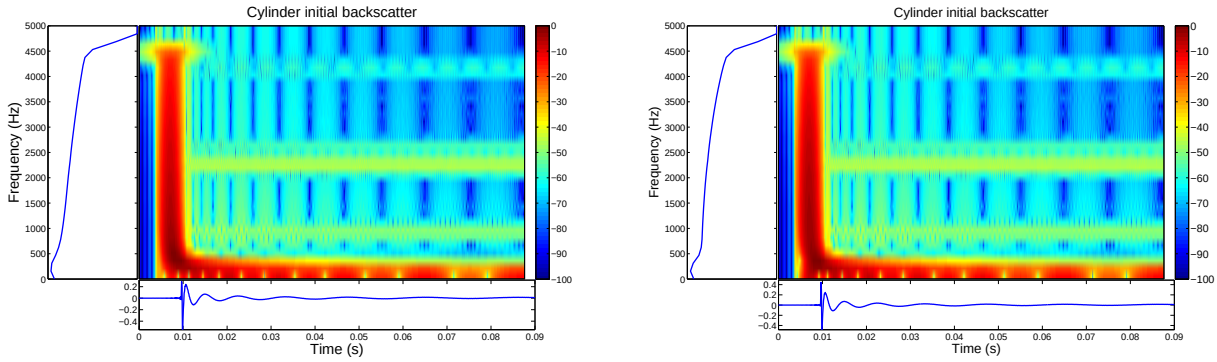

Figure 9: Spectrograms of the inital backscattered waves from a cylinder for $n = 5$ (left) and $n = 25$ (right).

Table 1: Material properties of the shells and environment.

	Density $\left(\frac{kg}{m^3}\right)$	**Sound Speed** $\left(\frac{m}{s}\right)$
Water layer	1000	1500
Sediment layer	1800	1800
Air	1.2	340
Steel	7800	5880 (dilatational) 3140 (shear)

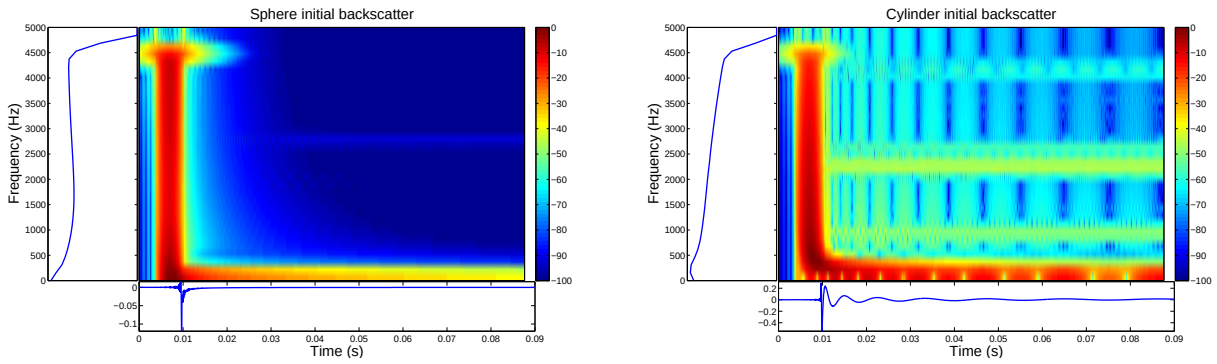

Figure 10: Spectrograms of the inital backscattered waves from the sphere and cylinder.

4.2 PROPAGATION MODELS

The propagation models used in the simulations calculate the impulse response of a waveguide, per mode, at a given range (horizontal distance). Both a two-plate and Pekeris waveguide are used, as implemented by Aluko [1]. The two-plate waveguide has a depth of 50 meters, a rigid bottom, and an isovelocity sound speed profile of $c = 1500\frac{m}{s}$. The Pekeris waveguide has a water layer depth of 50 meters, density $1000\frac{kg}{m^3}$, and isovelocity sound speed profile of $c = 1500\frac{m}{s}$. The sediment layer has a density of $1800\frac{kg}{m^3}$ and a sound speed of $1800\frac{m}{s}$. The group velocity curves for the two waveguides are given in Figure (11). In both waveguides, white Gaussian noise was added to the propagated waveforms to arrive at an SNR of 20 dB.

4.3 EVALUATION

The features are evaluated for classification utility by means of a receiver operating characteristic (ROC) curve. ROC curves are generated by computing distributions of feature values for the two shells and sweeping a decision threshold across the distributions. At each threshold, the probability of correct classification and the probability of false alarm (false

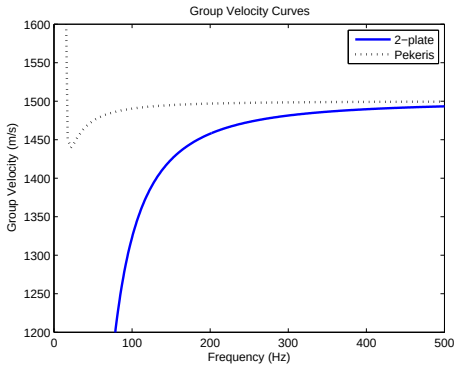

Figure 11: Group velocity curves for the waveguides used in this simulation.

classification) are computed. The probability of correct classification is then plotted against the probability of false alarm.

The ROC curve provides a graphical representation of the separability of the underlying distributions that facilitates easy comparison of the classification utility of the features. Better classification is indicated by higher probabilities of correct classification at lower probabilities of false alarm. Graphically, this is represented by the curve trending toward the upper-left corner of the figure. Two ROC curves are plotted on each figure, one for the dispersion-invariant moment (DIM), and one for the corresponding ordinary temporal moment (MOM). The curve which lies closer to the upper-left corner indicates the feature which has greater class separation and hence would be more useful for classification.

4.4 RESULTS

As the waves propagate through the dispersive two-plate waveguide, the energy at low frequencies is delayed relative to higher frequencies (Figures 12 and 13). This affects the value of the ordinary temporal moment, causing the value of the moment at any point x to contain information about both the steel shell and the channel itself. For example, the duration

moment, $n = 2$, will increase as the wave propagates, which is clear from the figures because the wave stretches in time. In contrast, the dispersion-invariant temporal moment will not vary (in the absence of noise), allowing the value of the moment at any point to fully reflect information about the target shell.

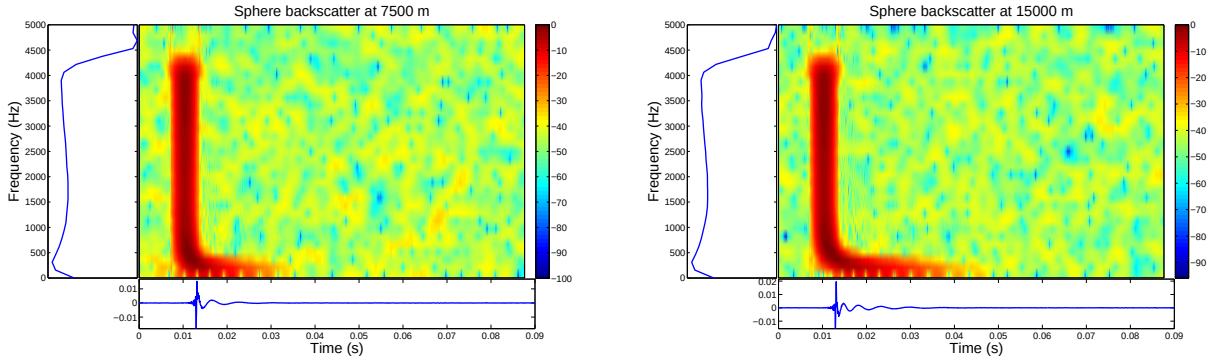

Figure 12: Spectrograms of the wave propagating in the two-plate waveguide from the sphere at 7.5 km and 15 km.

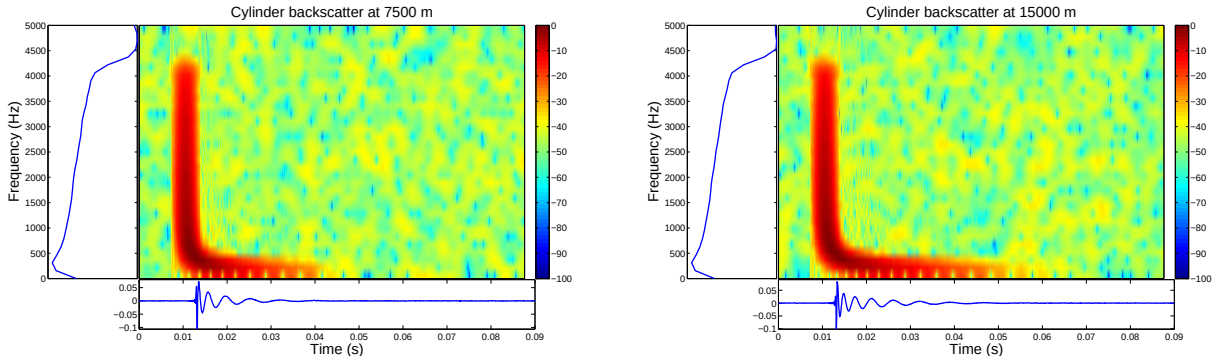

Figure 13: Spectrograms of the wave propagating in the two-plate waveguide from the cylinder at 7.5 km and 15 km.

ROC curves for the dispersion-invariant moments and the ordinary temporal moments show that the dispersion-invariant moments are more useful for classification (Figures 14, 15, and 16).

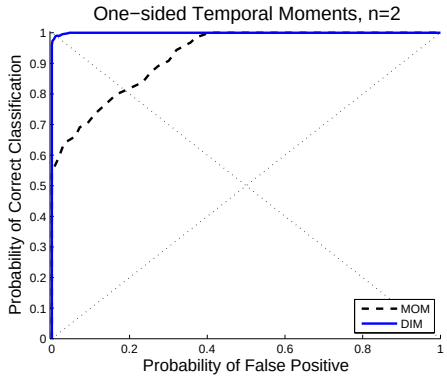

Figure 14: ROC curves comparing DIM and MOM for n=2, two-plate waveguide.

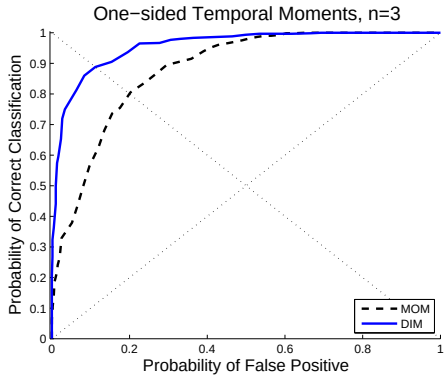

Figure 15: ROC curves comparing DIM and MOM for n=3, two-plate waveguide.

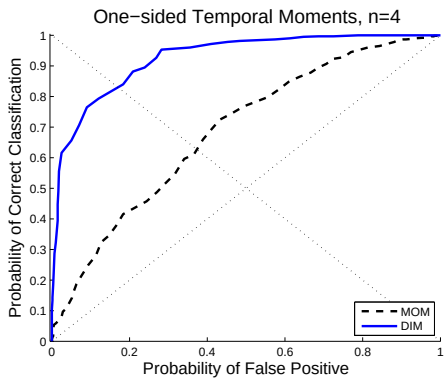

Figure 16: ROC curves comparing DIM and MOM for n=4, two-plate waveguide.

Waves propagating in the Pekeris waveguide also show the effects of dispersion, with lower frequencies delayed in time relative to higher frequencies (Figures 17 and 18).

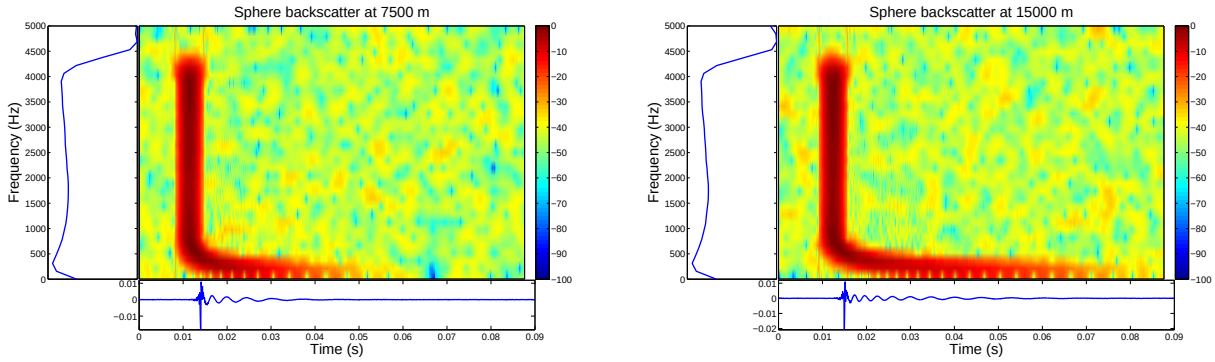

Figure 17: Spectrograms of the wave propagating in the Pekeris waveguide from the sphere at 7.5 km and 15 km.

ROC curves showing the separability of the distributions of the two moments for the waves propagating in the Pekeris waveguide indicate that the dispersion-invariant temporal moments are more useful for classification (Figures 19, 20, and 21).

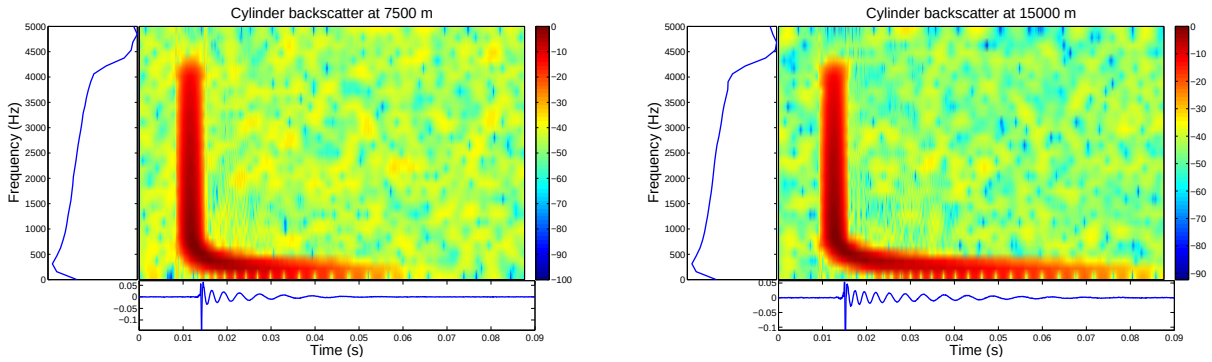

Figure 18: Spectrograms of the wave propagating in the Pekeris waveguide from the cylinder at 7.5 km and 15 km.

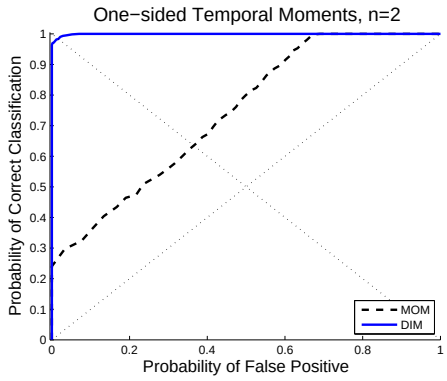

Figure 19: ROC curves comparing DIM and MOM for n=2, Pekeris waveguide.

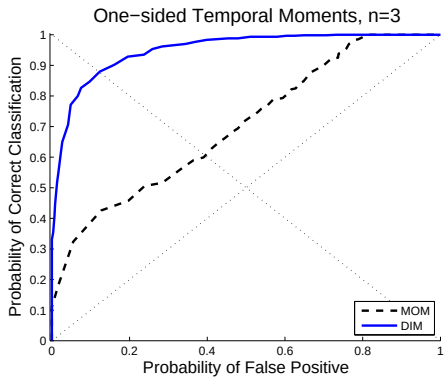

Figure 20: ROC curves comparing DIM and MOM for n=3, Pekeris waveguide.

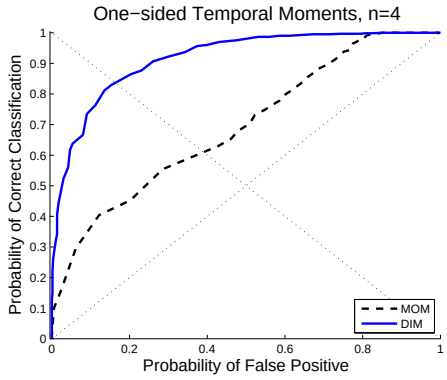

Figure 21: ROC curves comparing DIM and MOM for n=4, Pekeris waveguide.

5.0 CONCLUSIONS AND FUTURE WORK

We have implemented dispersion-invariant temporal moments [16] and evaluated their utility for classification of sonar signals in shallow-water ocean environments via simulation. For propagation with a real dispersion relation (no damping), it is clear that the dispersion-invariant temporal moments show greater separability than the corresponding ordinary temporal moments.

The propagation of sound in the ocean can exhibit frequency-dependent attenuation [3], and this will cause the value of A_n to vary with x. Figures 22 and 23 show the waves after propagating in the two-plate waveguide with complex dispersion,

$$k(\omega) = k_r(\omega) + j\beta \tag{5.1}$$

where $\beta = 10^{-8}$ is the imaginary part of the dispersion relation. Notice that the higher frequencies are attenuated as the wave propagates (compare with Figures (12) and (13) for propagation without attenuation).

This attenuation negatively affects classification (Figures 24, 25, and 26). In the future, we intend to develop signal processing methods to compensate for this attenuation.

We can model the attenuation as

$$B(x,\omega) = B(0,\omega)\, e^{-\beta\omega x}, \tag{5.2}$$

and taking the natural logarithm of both sides, we obtain

$$\ln B(x,\omega) = \ln B(0,\omega) - \beta\omega x \tag{5.3}$$

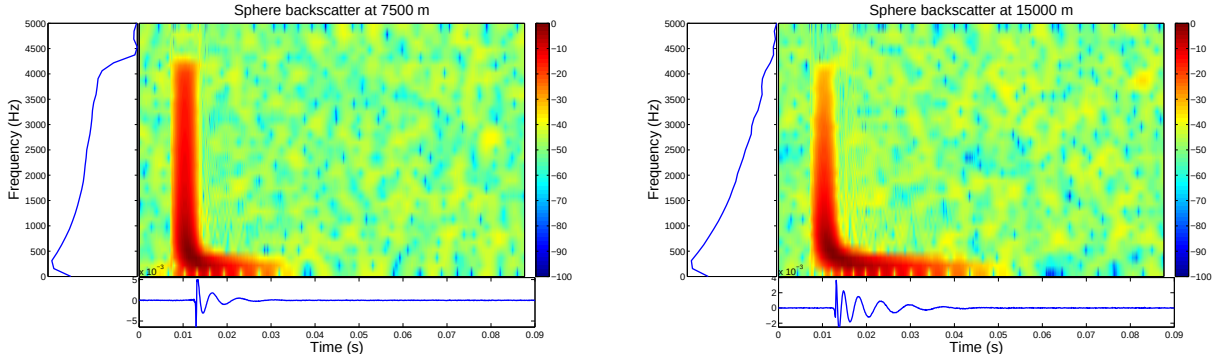

Figure 22: Spectrograms of the wave propagating with complex dispersion from the sphere at 7.5 km and 15 km.

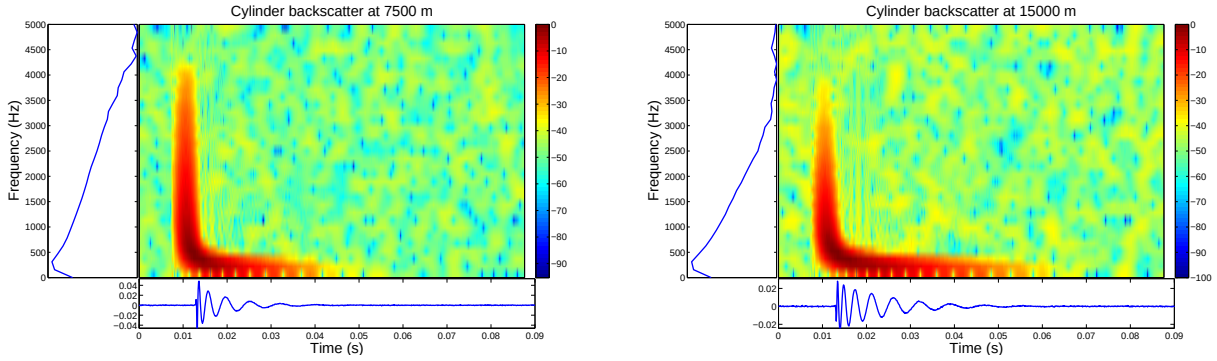

Figure 23: Spectrograms of the wave propagating with complex dispersion from the cylinder at 7.5 km and 15 km.

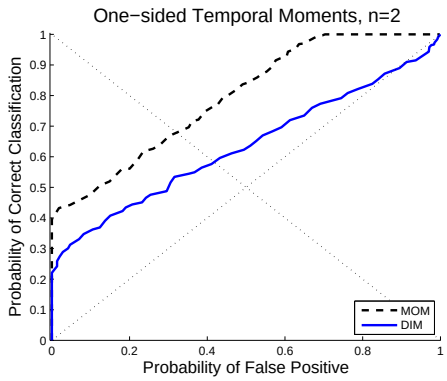

Figure 24: ROC curves comparing DIM and MOM for n=2, two-plate waveguide, complex dispersion.

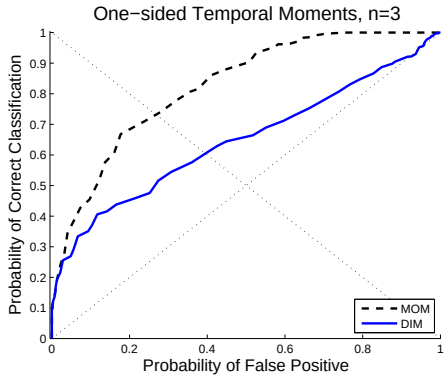

Figure 25: ROC curves comparing DIM and MOM for n=3, two-plate waveguide, complex dispersion.

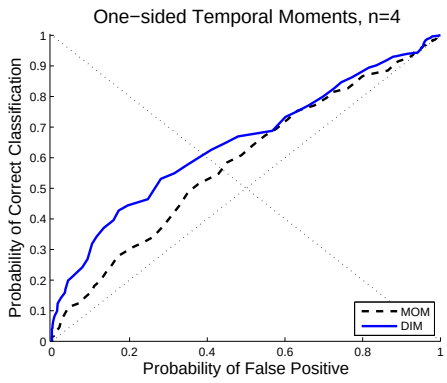

Figure 26: ROC curves comparing DIM and MOM for n=4, two-plate waveguide, complex dispersion.

The derivative with respect to ω gives

$$\frac{B'(x,\omega)}{B(x,\omega)} = \frac{B'(0,\omega)}{B(0,\omega)} - \beta x \tag{5.4}$$

which removes the frequency-dependence of the attenuation. This processing chain is akin to Cepstral processing [20]. In the future, we intend to explore this procedure and others in order to compensate for the effects of frequency-dependent attenuation.

APPENDIX A

CYLINDRICAL COORDINATES

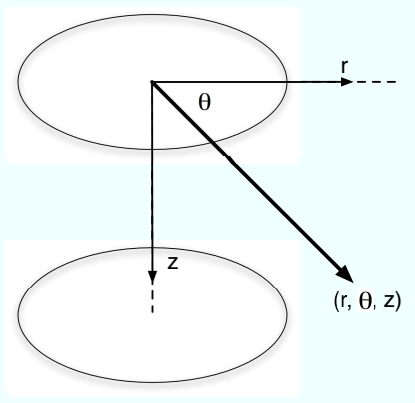

Figure 27: Cylindrical Coordinate System.

The Laplacian in cylindrical coordinates (Eq. 2.3) may be simplified by applying an assumption of *cylindrical symmetry*, that is, that a wave propagating from a source is identical at every value of θ for any given range r.

$$\Phi(r, \theta, z) = \Phi(r, z) \tag{A.1}$$

$$\Phi(r, z) = \Phi(-r, z) \tag{A.2}$$

This assumption is intuitively acceptable in light of the assumption that the environment is horizontally stratified, because the sound speed at a given depth is constant across range.

Equation (2.3) gives the formula for the Laplacian in cylindrical coordinates

$$\nabla^2 = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2}{\partial \theta^2} + \frac{\partial^2}{\partial z^2}. \tag{A.3}$$

Because of cylindrical symmetry, the value of Φ is constant over θ, and therefore the partial derivative with respect to θ goes to zero. The partial derivative with respect to r may be distributed using the product rule

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial}{\partial r}\right) = \frac{\partial^2}{\partial r^2} + \frac{1}{r}\frac{\partial}{\partial r} \tag{A.4}$$

Thus, Equation (A.3) reduces to

$$\nabla^2 = \frac{\partial^2}{\partial r^2} + \frac{1}{r}\frac{\partial}{\partial r} + \frac{\partial^2}{\partial z^2}, \tag{A.5}$$

and Equation (2.1) has the form

$$\frac{\partial^2 \Phi}{\partial r^2} + \frac{1}{r}\frac{\partial \Phi}{\partial r} + \frac{\partial^2 \Phi}{\partial z^2} = \frac{1}{c^2}\frac{\partial^2 \Phi}{\partial t^2}. \tag{A.6}$$

Applying the form of the desired solution (Equation 2.7), we obtain

$$\frac{\partial^2 \Phi}{\partial r^2} + \frac{1}{r}\frac{\partial \Phi}{\partial r} + \frac{\partial^2 \Phi}{\partial z^2} + k^2\Phi = 0, \tag{A.7}$$

an equation whose solution is well-known and given by

$$\Phi = \phi(z)\, H_0^{(1,2)}(k_r r)\, e^{-j\omega t}, \tag{A.8}$$

where $H_0^{(1,2)}$ are Hankel functions of the first and second kind. The asymptotic approximation, valid for $k_r r \gg 1$ and given by

$$H_0^{(1,2)}(k_r r) \approx \sqrt{\frac{2}{\pi k_r r}} e^{\pm j(k_r r - \pi/4)} \tag{A.9}$$

illustrates that $H_0^{(1)}$ corresponds to waves traveling in the $+r$ direction (outward) and thus we choose the Hankel function of the first kind in Equation (A.8).

APPENDIX B

T-MATRIX COEFFICIENTS

The T-matrix is given by

$$T_n(x) = \frac{|B_n(x)|}{|D_n(x)|} \tag{B.1}$$

where B and D are given in the following sections for the elastic sphere and elastic cylinder. In the definition of the coefficients, k_1 is the wavenumber in the water, k_{d_2} is the wavenumber for dilatational waves in the shell, k_{s_2} is the wavenumber for shear waves in the shell, and k_3 is the wavenumber for waves in air. The Hankel function of the first kind is denoted $H_n^{(1)}$, and J_n and Y_n are the Bessel functions of the first and second kind, respectively.

B.1 ELASTIC SPHERE

$$B_n(x) = \begin{bmatrix} \Re\{d_{11}\} & d_{12} & d_{13} & d_{14} & d_{15} & 0 \\ \Re\{d_{21}\} & d_{22} & d_{23} & d_{24} & d_{25} & 0 \\ 0 & d_{32} & d_{33} & d_{34} & d_{35} & 0 \\ 0 & d_{42} & d_{43} & d_{44} & d_{45} & d_{46} \\ 0 & d_{52} & d_{53} & d_{54} & d_{55} & d_{56} \\ 0 & d_{62} & d_{63} & d_{64} & d_{65} & 0 \end{bmatrix} \tag{B.2}$$

$$D_n(x) = \begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} & d_{15} & 0 \\ d_{21} & d_{22} & d_{23} & d_{24} & d_{25} & 0 \\ 0 & d_{32} & d_{33} & d_{34} & d_{35} & 0 \\ 0 & d_{42} & d_{43} & d_{44} & d_{45} & d_{46} \\ 0 & d_{52} & d_{53} & d_{54} & d_{55} & d_{56} \\ 0 & d_{62} & d_{63} & d_{64} & d_{65} & 0 \end{bmatrix} \tag{B.3}$$

$$d_{11} = \left(\frac{\rho_1}{\rho_2}\right) k_{s_2}^2 a^2 H_n^{(1)}(k_1 a) \tag{B.4}$$

$$d_{12} = \left[2n(n+1) - k_{s_2}^2 a^2\right] J_n(k_{d_2} a) - 4k_{d_2} a J_n'(k_{d_2} a) \tag{B.5}$$

$$d_{13} = \left[2n(n+1) - k_{s_2}^2 a^2\right] Y_n(k_{d_2} a) - 4k_{d_2} a Y_n'(k_{d_2} a) \tag{B.6}$$

$$d_{14} = 2n(n+1)\left[k_{s_2} a J_n'(k_{s_2} a) - J_n(k_{s_2} a)\right] \tag{B.7}$$

$$d_{15} = 2n(n+1)\left[k_{s_2} a Y_n'(k_{s_2} a) - Y_n(k_{s_2} a)\right] \tag{B.8}$$

$$d_{21} = k_1 a H_n'^{(1)}(k_1 a) \tag{B.9}$$

$$d_{22} = k_{d_2} a J_n'(k_{d_2} a) \tag{B.10}$$

$$d_{23} = k_{d_2} a Y_n'(k_{d_2} a) \tag{B.11}$$

$$d_{24} = n(n+1) J_n(k_{s_2} a) \tag{B.12}$$

$$d_{25} = n(n+1) Y_n(k_{s_2} a) \tag{B.13}$$

$$d_{32} = 2\left[J_n\left(k_{d_2}a\right) - k_{d_2}aJ_n'\left(k_{d_2}a\right)\right] \tag{B.14}$$

$$d_{33} = 2\left[Y_n\left(k_{d_2}a\right) - k_{d_2}aY_n'\left(k_{d_2}a\right)\right] \tag{B.15}$$

$$d_{34} = 2k_{s_2}aJ_n'\left(k_{s_2}a\right) + \left[k_{s_2}^2a^2 - 2n\left(n+1\right) + 2\right]J_n\left(k_{s_2}a\right) \tag{B.16}$$

$$d_{35} = 2k_{s_2}aY_n'\left(k_{s_2}a\right) + \left[k_{s_2}^2a^2 - 2n\left(n+1\right) + 2\right]Y_n\left(k_{s_2}a\right) \tag{B.17}$$

$$d_{42} = -4k_{d_2}bJ_n'\left(k_{d_2}b\right) + \left[2n\left(n+1\right) - k_{s_2}^2b^2\right]J_n\left(k_{d_2}b\right) \tag{B.18}$$

$$d_{43} = -4k_{d_2}bY_n'\left(k_{d_2}b\right) + \left[2n\left(n+1\right) - k_{s_2}^2b^2\right]Y_n\left(k_{d_2}b\right) \tag{B.19}$$

$$d_{44} = 2n\left(n+1\right)\left[k_{s_2}bJ_n'\left(k_{s_2}b\right) - J_n\left(k_{s_2}b\right)\right] \tag{B.20}$$

$$d_{45} = 2n\left(n+1\right)\left[k_{s_2}bY_n'\left(k_{s_2}b\right) - Y_n\left(k_{s_2}b\right)\right] \tag{B.21}$$

$$d_{46} = \left(\frac{\rho_3}{\rho_2}\right)k_{s_2}^2b^2J_n\left(k_3b\right) \tag{B.22}$$

$$d_{52} = k_{d_2}bJ_n'\left(k_{d_2}b\right) \tag{B.23}$$

$$d_{53} = k_{d_2}bY_n'\left(k_{d_2}b\right) \tag{B.24}$$

$$d_{54} = n\left(n+1\right)J_n\left(k_{s_2}b\right) \tag{B.25}$$

$$d_{55} = n\left(n+1\right)Y_n\left(k_{s_2}b\right) \tag{B.26}$$

$$d_{56} = -k_3bJ_n'\left(k_3b\right) \tag{B.27}$$

$$d_{62} = 2\left[J_n\left(k_{d_2}b\right) - k_{d_2}bJ'\left(k_{d_2}b\right)\right] \tag{B.28}$$

$$d_{63} = 2\left[Y_n\left(k_{d_2}b\right) - k_{d_2}bY'\left(k_{d_2}b\right)\right] \tag{B.29}$$

$$d_{64} = 2k_{s_2}bJ_n'\left(k_{s_2}b\right) + \left[k_{s_2}^2b^2 - 2n\left(n+1\right) + 2\right]J_n\left(k_{s_2}b\right) \tag{B.30}$$

$$d_{65} = 2k_{s_2}bY_n'\left(k_{s_2}b\right) + \left[k_{s_2}^2b^2 - 2n\left(n+1\right) + 2\right]Y_n\left(k_{s_2}b\right) \tag{B.31}$$

$$A_1^* = -\left(\frac{\rho_1}{\rho_2}\right)k_{s_2}^2a^2J_n\left(k_1a\right) \tag{B.32}$$

$$A_2^* = k_1aJ_n'\left(k_1a\right) \tag{B.33}$$

B.2 ELASTIC CYLINDER

$$B_n(x) = \begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} & d_{15} & A_1^* \\ d_{21} & d_{22} & d_{23} & d_{24} & d_{25} & A_2^* \\ 0 & d_{32} & d_{33} & d_{34} & d_{35} & 0 \\ 0 & d_{42} & d_{43} & d_{44} & d_{45} & 0 \\ 0 & d_{52} & d_{53} & d_{54} & d_{55} & 0 \\ 0 & d_{62} & d_{63} & d_{64} & d_{65} & 0 \end{bmatrix} \tag{B.34}$$

$$D_n(x) = \begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} & d_{15} & 0 \\ d_{21} & d_{22} & d_{23} & d_{24} & d_{25} & 0 \\ 0 & d_{32} & d_{33} & d_{34} & d_{35} & 0 \\ 0 & d_{42} & d_{43} & d_{44} & d_{45} & d_{46} \\ 0 & d_{52} & d_{53} & d_{54} & d_{55} & d_{56} \\ 0 & d_{62} & d_{63} & d_{64} & d_{65} & 0 \end{bmatrix} \tag{B.35}$$

$$d_{11} = \left(\frac{\rho_1}{\rho_2}\right) k_{s_2}^2 a^2 H_n^{(1)}(k_1 a) \tag{B.36}$$

$$d_{12} = \left(2n^2 - k_{s_2}^2 a^2\right) J_n(k_{d_2} a) - 2k_{d_2} a J_n'(k_{d_2} a) \tag{B.37}$$

$$d_{13} = \left(2n^2 - k_{s_2}^2 a^2\right) Y_n(k_{d_2} a) - 2k_{d_2} a Y_n'(k_{d_2} a) \tag{B.38}$$

$$d_{14} = 2n\left[k_{s_2} a J_n'(k_{s_2} a) - J_n(k_{s_2} a)\right] \tag{B.39}$$

$$d_{15} = 2n\left[k_{s_2} a Y_n'(k_{s_2} a) - Y_n(k_{s_2} a)\right] \tag{B.40}$$

$$d_{21} = -k_1 a H_n^{(1)}(k_1 a) \tag{B.41}$$

$$d_{22} = k_{d_2} a J_n'(k_{d_2} a) \tag{B.42}$$

$$d_{23} = k_{d_2} a Y_n'(k_{d_2} a) \tag{B.43}$$

$$d_{24} = n J_n(k_{s_2} a) \tag{B.44}$$

$$d_{25} = n Y_n(k_{s_2} a) \tag{B.45}$$

$$d_{32} = 2n\left[J_n\left(k_{d_2}a\right) - k_{d_2}aJ_n'\left(k_{d_2}a\right)\right] \tag{B.46}$$

$$d_{33} = 2n\left[Y_n\left(k_{d_2}a\right) - k_{d_2}aY_n'\left(k_{d_2}a\right)\right] \tag{B.47}$$

$$d_{34} = 2k_{s_2}aJ_n'\left(k_{s_2}a\right) + \left[k_{s_2}^2a^2 - 2n^2\right]J_n\left(k_{s_2}a\right) \tag{B.48}$$

$$d_{35} = 2k_{s_2}aY_n'\left(k_{s_2}a\right) + \left[k_{s_2}^2a^2 - 2n^2\right]Y_n\left(k_{s_2}a\right) \tag{B.49}$$

$$d_{42} = -2k_{d_2}bJ_n'\left(k_{d_2}b\right) + \left[2n^2 - k_{s_2}^2b^2\right]J_n\left(k_{d_2}b\right) \tag{B.50}$$

$$d_{43} = -2k_{d_2}bY_n'\left(k_{d_2}b\right) + \left[2n^2 - k_{s_2}^2b^2\right]Y_n\left(k_{d_2}b\right) \tag{B.51}$$

$$d_{44} = 2n\left[k_{s_2}bJ_n'\left(k_{s_2}b\right) - J_n\left(k_{s_2}b\right)\right] \tag{B.52}$$

$$d_{45} = 2n\left[k_{s_2}bY_n'\left(k_{s_2}b\right) - Y_n\left(k_{s_2}b\right)\right] \tag{B.53}$$

$$d_{46} = \left(\frac{\rho_3}{\rho_2}\right)k_{s_2}^2b^2J_n\left(k_3b\right) \tag{B.54}$$

$$d_{52} = k_{d_2}bJ_n'\left(k_{d_2}b\right) \tag{B.55}$$

$$d_{53} = k_{d_2}bY_n'\left(k_{d_2}b\right) \tag{B.56}$$

$$d_{54} = nJ_n\left(k_{s_2}b\right) \tag{B.57}$$

$$d_{55} = nY_n\left(k_{s_2}b\right) \tag{B.58}$$

$$d_{56} = -k_3bJ_n'\left(k_3b\right) \tag{B.59}$$

$$d_{62} = 2n\left[J_n\left(k_{d_2}b\right) - k_{d_2}bJ_n'\left(k_{d_2}b\right)\right] \tag{B.60}$$

$$d_{63} = 2n\left[Y_n\left(k_{d_2}b\right) - k_{d_2}bY_n'\left(k_{d_2}b\right)\right] \tag{B.61}$$

$$d_{64} = 2k_{s_2}bJ_n'\left(k_{s_2}^2b^2 - 2n^2\right)J_n\left(k_{s_2}b\right) \tag{B.62}$$

$$d_{65} = 2k_{s_2}bY_n'\left(k_{s_2}^2b^2 - 2n^2\right)Y_n\left(k_{s_2}b\right) \tag{B.63}$$

$$A_1^* = -\left(\frac{\rho_1}{\rho_2}\right)k_{s_2}^2a^2J_n\left(k_1a\right) \tag{B.64}$$

$$A_2^* = k_1aJ_n'\left(k_1a\right) \tag{B.65}$$

APPENDIX C

MATLAB CODE USED FOR SIMULATIONS

C.1 MOMENTS.M

```
function [DIM,MOM,y] = moments(x,fs,n,flag)
%
% USAGE:  [DIM,MOM,y] = moments(x,fs,n,flag)
%
% Matlab function to compute dispersion-invariant and ordinary central
% temporal moments of time series x
%
% Parameters:
% x - input time series [dimension 1xN]
% fs - sampling frequency [Hz]
% n - order of moment
% flag - 0=return ordinary central moments
%        1=return absolute value central moments
% (only effects odd moments, not DIMs)
%        2=return inverse-t moments
% DIM - dispersion-invariant moment value
% MOM - ordinary moment value
% y - augmented time series corresponding to dispersion-invariant moment
```

```
%
% Author: P. Loughlin (adapted from Greg Okopal's momgd147)
% Version 1.0, 10 March 2006
% Version 1.1, 16 March 2006: added inverse-time moments and changed DIMS
% to one-sided calculations (G. Okopal)
%

[r,c]=size(x);
if r>c, x=x'; end  % force x to 1xN vector

y = (ifft(abs(fft(x)))); % augmented signal for DIM calcuations
y = y(1:floor(end/2)); %use only the first half of the augmented signal
t = [0:length(x)-1]/fs;
t_dim=[0:length(y)-1]/fs; % t vector for dispersion invariant
% % *** SKIP this section -- only needed if doing symmetric
% % DIMs (i.e., using full ifft vector, rather than the first half for
% % one-sided DIMs as now being done starting with v1.1.) ***
% % check for odd / even length, and adjust vector to preserve symmetry. For
% % odd length, y(1)=y(end), y(2)=y(end-1),..., but for even length,
% % symmetry is such that y(2)=y(end), y(3)=y(end-1), etc. Set
% % y(1)=0 for even-length vector.  This will give zero for odd moments,
% % which is what you should get.
% %if ~mod(length(y),2), y(1)=0; end
y=y/norm(y);  % normalize to unit-energy for moment calculations
x=x/norm(x);
%t_a = sum(t_dim.*(abs(y).^2));  % compute mean time (Don't need for
%one-sided DIM)
t_x = sum(t.*(abs(x).^2));
if (flag==1)
    %absolute moments
```

```
    DIM = sum((abs(t_dim).^n).*(abs(y).^2));
    MOM = sum((abs(t-t_x).^n).*(abs(x).^2));
elseif (flag==2)
    %inverse t moments
    t_dim_inv= (1./(t_dim+1));
    t_inv= (1./(t+1));

    %compute mean inverse time
    %t_a_inv = sum(t_dim_inv.*(abs(y).^2));  %Don't need for one-sided DIM
    t_x_inv = sum(t_inv.*(abs(x).^2));

    DIM = sum(((t_dim_inv).^n).*(abs(y).^2));
    MOM = sum(((t_inv-t_x_inv).^n).*(abs(x).^2));
else
    %ordinary moments
    %one-sided DIMs are by default absolute moments
    DIM = sum(((t_dim).^n).*(abs(y).^2));
    MOM = sum(((t-t_x).^n).*(abs(x).^2));
end
return;
```

C.2 CLASSIFICATION_SIMULATION.M

```
%This is the main script that runs the simulation

%*********************************************************
%GENERAL PARAMETERS
%*********************************************************
fs = 10000;%sampling frequency
```

```
max_n = 5;%number of scattering modes
freq_pts=fs;%number of pts used in RST calculation
flr = 100;%spectrogram floor
colorflag = 0;%0 color, 1 b&w

%*********************************************************
%CHANNEL PARAMETERS
%*********************************************************
MaxDist=15000;%total channel length
IncDist=25;%increment for moment calculation
D=50;%depth of channel
m=1;%propagated mode
snr=20;%determines power of white Gaussian noise added to signal
imaginary_dispersion = 0;%(10^(-8));

%*********************************************************
%MOMENT PARAMETERS
%*********************************************************
mom_flag=0;%mom_flag = 0 for ordinary, 1 for absolute, 2 for inverse
ord=4;%highest order of moment to calculate

%*********************************************************
%SHELL PARAMETERS
%*********************************************************
shell_type1=0;%shell types: 0 for sphere, 1 for cylinder
a_s1 = 0.90;%m =>outer radius of shell 1
b_s1 = 0.45;%m =>inner radius of shell 1

shell_type2=1;%shell types: 0 for sphere, 1 for cylinder
a_s2 = 1.45;%m =>outer radius of shell 2
```

```
b_s2 = 1.40;%m =>inner radius of shell 2

%data for an air filled steel shell
rho1=1000; %density of water
rho3=1.2;%   =>density of air inside shell
rho2=7800;%g/cm^-3 =>density of material
c_1 = 1500;%m/s => sound speed in water
c_3 = 340;%m/s =>sound speed in air
c_d2 = 5880;%cm s^-1 =>speed of dilational waves
c_s2 = 3140;%cm s^-1 =>speed of shear waves

%simulate propagation of shell 1
[d_shell1,A_x_shell1,f_t_shell1,disp_f_t_shell1,h_shell1,
mid_f_t_shell1,mid_dist_shell1,u_a_shell1]
=simulate_propagatio(ord,fs,freq_pts,snr,D,IncDist,MaxDist,m,
a_s1,b_s1,max_n,rho1,rho2,rho3,
c_1,c_d2,c_s2,c_3,mom_flag,shell_type1,imaginary_dispersion);

%simulate propagation of shell 2
[d_shell2,A_x_shell2,f_t_shell2,disp_f_t_shell2,h_shell2,
mid_f_t_shell2,mid_dist_shell2,u_a_shell2]
=simulate_propagation(ord,fs,freq_pts,snr,D,IncDist,MaxDist,m,
a_s2,b_s2,max_n,rho1,rho2,rho3,
c_1,c_d2,c_s2,c_3,mom_flag,shell_type2,imaginary_dispersion);



%Plot initial sphere backscatter
figtitle=['Sphere initial backscatter'];
plot_backscatter(f_t_shell1,fs,flr,figtitle,colorflag)
```

```
%Plot initial cylinder backscatter
figtitle=['Cylinder initial backscatter'];
plot_backscatter(f_t_shell2,fs,flr,figtitle,colorflag)

%Plot mid-distance sphere backscatter
figtitle=['Sphere backscatter at ', int2str(mid_dist_shell1), ' m'];
plot_backscatter(mid_f_t_shell1,fs,flr,figtitle,colorflag)
%Plot mid-distance cylinder backscatter
figtitle=['Cylinder backscatter at ', int2str(mid_dist_shell2), ' m'];
plot_backscatter(mid_f_t_shell2,fs,flr,figtitle,colorflag)

%Plot final sphere backscatter
figtitle=['Sphere backscatter at ', int2str(MaxDist), ' m'];
plot_backscatter(disp_f_t_shell1,fs,flr,figtitle,colorflag)
%Plot final cylinder backscatter
figtitle=['Cylinder backscatter at ', int2str(MaxDist), ' m'];
plot_backscatter(disp_f_t_shell2,fs,flr,figtitle,colorflag)

%set labels
if mom_flag==0
    mom_type='One-sided Temporal Moments';
elseif mom_flag==1
    mom_type='Absolute Value Moments';
else
    mom_type='Inverse-time Moments';
end

%Plot ROC curves
for ord_num = 2:ord
    roc3(d_shell1(ord_num,:),d_shell2(ord_num,:),75,'MOM',
```

```
    A_x_shell1(ord_num,:),A_x_shell2(ord_num,:),75,'DIM',
    [mom_type, ', n=', int2str(ord_num)],colorflag);
end

%************************************************************************
%************************************************************************
function [d,A_x,f_t,disp_f_t,h,mid_f_t,mid_dist,u_a]=simulate_propagation
(ord,fs,freq_pts,snr,D,IncDist,MaxDist,m,a,b,max_n,rho1,rho2,rho3,
c_1,c_d2,c_s2,c_3,mom_flag,shell_type,imag_disp)
w=fs/freq_pts:fs/freq_pts:fs/2;
%************************************************************************

%Compute initial shell backscatter
f_t = shell_backscatter(a,b,rho1,rho2,rho3,
c_1,c_d2,c_s2,c_3,w,max_n,shell_type);
f_t = f_t(1:round(length(f_t)/3));%crop excess signal length
f_t = [zeros(1,100) f_t];%zero pad

%generate impluse for channel response
imp = zeros(1,2*length(f_t));imp(1) = 1;

num = (MaxDist-IncDist)/IncDist;%calculate number of iterations

d = zeros(length(ord),num);%storage area for MOM
A_x = zeros(length(ord),num);%storage area for DIM

%anti-aliasing taper filter
f0m=(m-0.5)*c_1/(2*D*fs)
[Bcheb,Acheb]=cheby2(8,80,0.95,'low');
f_t=filtfilt(Bcheb,Acheb,f_t);
```

```
[B,A]=cheby2(4,60,10*f0m,'high');


for ind=1:num
    %2-plate waveguide code
    [h1,y] = invwaveguide3(imp,c_1,D,ind*IncDist,1,fs,imag_disp);
    h=fliplr(h1);

    %Polynomial waveguide code
    %[h,y] = waveguide(imp,c_1,D,ind*IncDist,m,fs,3);

    %Pekeris waveguide code
    %[h,y] = invpekeris2k(imp,c_1,1800,1000,1800,D,ind*IncDist,m,fs);
    %h=fliplr(h);


    h=filtfilt(B,A,h);%anti-aliasing taper filter
    disp_f_t = conv(f_t,h);%convolve backscatter with channel response

    disp_f_t = awgn(disp_f_t,snr,'measured');%add noise to the echo

    %crop excess
    disp_f_t = disp_f_t(round((length(disp_f_t)/6):
    (round(3*length(disp_f_t)/6))));

    %calculate moments
    for ord_num = 1:ord
        [A_x(ord_num,ind),d(ord_num,ind),u_a] =
        moments(disp_f_t,fs,ord_num,mom_flag);
    end
```

```
    %save midpoint echo for display
    if (ind==round(num/2))
        mid_f_t = disp_f_t;
        mid_dist=ind*IncDist;
    end
end

%************************************************************************
%************************************************************************
%************************************************************************
function f_t = shell_backscatter(a,b,rho1,rho2,rho3,c_1,c_d2,c_s2,c_3,w,
max_n,shell_type)
%************************************************************************
%Computes initial shell backscatter
%************************************************************************
f_n = zeros(max_n,length(w));%storage area for modes of the shell

k_1=w./c_1;%wavenumber in water

if (shell_type==0)
    %spherical shell
    for n = 1:max_n
        T=sphere_backscatter_coefficients
        (n,w,a,b,rho1,rho2,rho3,c_1,c_d2,c_s2,c_3);
        f_n(n,:)=a./(i.*k_1).*((-1)^n).*(2*n+1).*T;
        %only retain modes that don't throw NaNs
        if (sum(isnan(f_n(n,:)))~=0)
            f_n(n,:) = zeros(1,length(f_n(n,:)));
        end
```

```
    end
elseif (shell_type==1)
    %cylindrical shell
    for n = 1:max_n
        T=cylinder_backscatter_coefficients
        (n,w,a,b,rho1,rho2,rho3,c_1,c_d2,c_s2,c_3);
        f_n(n,:)=a./(i.*k_1).*((-1)^n).*(2*n+1).*T;
        %only retain modes that don't throw NaNs
        if (sum(isnan(f_n(n,:)))~=0)
            f_n(n,:) = zeros(1,length(f_n(n,:)));
        end
    end
end

%build interrogating impulse
g = zeros(1,length(w));g(1)=1;
[b,a]=butter(4,.1);
g_t=filter(b,a,g);

%compute backscatter due to interrogating impulse
f_t = real(ifft(sum(f_n,1)));
f_t = f_t(1:round(length(f_t)/2));
y=conv(g_t,f_t);



%************************************************************************
%************************************************************************
function plot_backscatter(f_t_signal,fs,flr,figtitle,colorflag)
%this function plots the spectrogram, time series, and frequency marginal
```

```
%compute normalized spectrogram
[B,faxis,taxis]=spectrogram(f_t_signal,64,63,64,fs);
B=abs(B).^2;
B=B./(max(max(B)));

%compute frequency marginal
fm = 10.*log10(sum(B,2));
fmin = find(fm>-flr);
fm2 = ones(size(fm)).*-flr;
fm2(fmin)=fm(fmin);
fm=fm2;

B = 10.*log10(B);
B2=ones(size(B)).*-flr;
Bin = find(B>-flr);
B2(Bin) = B(Bin);
tlabel='Time (s)';flabel='Frequency (Hz)';
epsflag=0;fname=0;
figure;tfdplot2(f_t_signal,B2',fm,t
axis,faxis,tlabel,flabel,figtitle,epsflag,fname,colorflag);
```

C.3 SPHERE_BACKSCATTER_COEFFICIENTS.M

```
function T=sphere_backscatter_coefficients
(n,w,a,b,rho1,rho2,rho3,c_1,c_d2,c_s2,c_3)
%*************************************************************************
%Coefficients for sphere backscatter
%*************************************************************************
w2 = [w w(end)+abs(w(1)-w(2))]-(abs(w(1)-w(2))/2);
```

```
h = 1-b/a;

k_1 = w./c_1; %wavenumber in water
k_d2 = w./c_d2; %dilatational wavenumber in shell
k_s2 = w./c_s2; %shear wavenumber in shell
k_3 = w./c_3; %wavenumber in air

k_1_2 = w2./c_1; %wavenumber in water
k_d2_2 = w2./c_d2; %dilatational wavenumber in shell
k_s2_2 = w2./c_s2; %shear wavenumber in shell
k_3_2 = w2./c_3; %wavenumber in air

hnk1a = besselh(n,k_1.*a);
d_hnk1a = diff(besselh(n,k_1_2.*a))./(diff(k_1_2.*a));

jnk1a = besselj(n,k_1.*a);
d_jnk1a = diff(besselj(n,k_1_2.*a))./(diff(k_1_2.*a));

jnkd2a = besselj(n,k_d2.*a);
d_jnkd2a = diff(besselj(n,k_d2_2.*a))./diff(k_d2_2.*a);

ynkd2a = bessely(n,k_d2.*a);
d_ynkd2a = diff(bessely(n,k_d2_2.*a))./diff(k_d2_2.*a);

jnks2a = besselj(n,k_s2.*a);
d_jnks2a = diff(besselj(n,k_s2_2.*a))./diff(k_s2_2.*a);

ynks2a = bessely(n,k_s2.*a);
d_ynks2a = diff(bessely(n,k_s2_2.*a))./diff(k_s2_2.*a);
```

```
jnkd2b = besselj(n,k_d2.*b);
d_jnkd2b = diff(besselj(n,k_d2_2.*b))./diff(k_d2_2.*b);

ynkd2b = bessely(n,k_d2.*b);
d_ynkd2b = diff(bessely(n,k_d2_2.*b))./diff(k_d2_2.*b);

jnks2b = besselj(n,k_s2.*b);
d_jnks2b = diff(besselj(n,k_s2_2.*b))./diff(k_s2_2.*b);

ynks2b = bessely(n,k_s2.*b);
d_ynks2b = diff(bessely(n,k_s2_2.*b))./diff(k_s2_2.*b);

jnk3b = besselj(n,k_3.*b);
d_jnk3b = diff(besselj(n,k_3_2.*b))./diff(k_3_2.*b);

d11=(rho1/rho2).*k_s2.^2.*a^2.*hnk1a;
d12=(2*n*(n+1)-k_s2.^2.*a^2).*jnkd2a-4.*k_d2.*a.*d_jnkd2a;
d13=(2*n*(n+1)-k_s2.^2.*a^2).*ynkd2a-4.*k_d2.*a.*d_ynkd2a;
d14=2*n*(n+1).*(k_s2.*a.*d_jnks2a-jnks2a);
d15=2*n*(n+1).*(k_s2.*a.*d_ynks2a-ynks2a);

d21=k_1.*a.*d_hnk1a;
d22=k_d2.*a.*d_jnkd2a;
d23=k_d2.*a.*d_ynkd2a;
d24=n*(n+1).*jnks2a;
d25=n*(n+1).*ynks2a;

d32=2.*(jnkd2a-k_d2.*a.*d_jnkd2a);
d33=2.*(ynkd2a-k_d2.*a.*d_ynkd2a);
```

```
d34=2.*k_s2.*a.*d_jnks2a+(k_s2.^2.*a^2-2*n*(n+1)+2).*jnks2a;
d35=2.*k_s2.*a.*d_ynks2a+(k_s2.^2.*a^2-2*n*(n+1)+2).*ynks2a;

d42=-4.*k_d2.*b.*d_jnkd2b+(2*n*(n+1)-k_s2.^2.*b^2).*jnkd2b;
d43=-4.*k_d2.*b.*d_ynkd2b+(2*n*(n+1)-k_s2.^2.*b^2).*ynkd2b;
d44=2*n*(n+1).*(k_s2.*b.*d_jnks2b-jnks2b);
d45=2*n*(n+1).*(k_d2.*b.*d_ynks2b-ynks2b);
d46=(rho3/rho2).*k_s2.^2.*b^2.*jnk3b;

d52=k_d2.*b.*d_jnkd2b;
d53=k_d2.*b.*d_ynkd2b;
d54=n*(n+1).*jnks2b;
d55=n*(n+1).*ynks2b;
d56=-k_3.*b.*d_jnk3b;

d62=2.*(jnkd2b-k_d2.*b.*d_jnkd2b);
d63=2.*(ynkd2b-k_d2.*b.*d_ynkd2b);
d64=2.*k_s2.*b.*d_jnks2b+(k_s2.^2.*b^2-2*n*(n+1)+2).*jnks2b;
d65=2.*k_s2.*b.*d_ynks2b+(k_s2.^2.*b^2-2*n*(n+1)+2).*ynks2b;

A_star_1=-(rho1/rho2).*k_s2.^2.*a^2.*jnk1a;
A_star_2=k_1.*a.*d_jnk1a;



T = zeros(1,length(w));
for ind = 1:length(w)

B=
[real(d11(ind))   d12(ind)     d13(ind)     d14(ind)     d15(ind)     0;
real(d21(ind))  d22(ind)     d23(ind)     d24(ind)     d25(ind)     0;
```

```
0               d32(ind)     d33(ind)     d34(ind)     d35(ind)     0;
0               d42(ind)     d43(ind)     d44(ind)     d45(ind)     d46(ind);
0               d52(ind)     d53(ind)     d54(ind)     d55(ind)     d56(ind);
0               d62(ind)     d63(ind)     d64(ind)     d65(ind)     0];

D=[d11(ind)     d12(ind)     d13(ind)     d14(ind)     d15(ind)     0;
d21(ind)     d22(ind)     d23(ind)     d24(ind)     d25(ind)     0;
0           d32(ind)     d33(ind)     d34(ind)     d35(ind)     0;
0           d42(ind)     d43(ind)     d44(ind)     d45(ind)     d46(ind);
0           d52(ind)     d53(ind)     d54(ind)     d55(ind)     d56(ind);
0           d62(ind)     d63(ind)     d64(ind)     d65(ind)     0];

    T(ind) = -det(B)/det(D);

end
```

C.4 CYLINDER_BACKSCATTER_COEFFICIENTS.M

```
function T=cylinder_backscatter_coefficients
(n,w,a,b,rho1,rho2,rho3,c_1,c_d2,c_s2,c_3)
%*************************************************************************
%Coefficients for cylinder backscatter
%*************************************************************************

w2 = [w w(end)+abs(w(1)-w(2))]-(abs(w(1)-w(2))/2);

h = 1-b/a;

k_1 = w./c_1; %wavenumber in water
```

```
k_d2 = w./c_d2; %dilatational wavenumber in shell
k_s2 = w./c_s2; %shear wavenumber in shell
k_3 = w./c_3; %wavenumber in air

k_1_2 = w2./c_1; %wavenumber in water
k_d2_2 = w2./c_d2; %dilatational wavenumber in shell
k_s2_2 = w2./c_s2; %shear wavenumber in shell
k_3_2 = w2./c_3; %wavenumber in air

hnk1a = besselh(n,k_1.*a);
d_hnk1a = diff(besselh(n,k_1_2.*a))./(diff(k_1_2.*a));

jnk1a = besselj(n,k_1.*a);
d_jnk1a = diff(besselj(n,k_1_2.*a))./(diff(k_1_2.*a));

jnkd2a = besselj(n,k_d2.*a);
d_jnkd2a = diff(besselj(n,k_d2_2.*a))./diff(k_d2_2.*a);

ynkd2a = bessely(n,k_d2.*a);
d_ynkd2a = diff(bessely(n,k_d2_2.*a))./diff(k_d2_2.*a);

jnks2a = besselj(n,k_s2.*a);
d_jnks2a = diff(besselj(n,k_s2_2.*a))./diff(k_s2_2.*a);

ynks2a = bessely(n,k_s2.*a);
d_ynks2a = diff(bessely(n,k_s2_2.*a))./diff(k_s2_2.*a);

jnkd2b = besselj(n,k_d2.*b);
d_jnkd2b = diff(besselj(n,k_d2_2.*b))./diff(k_d2_2.*b);
```

```
ynkd2b = bessely(n,k_d2.*b);
d_ynkd2b = diff(bessely(n,k_d2_2.*b))./diff(k_d2_2.*b);

jnks2b = besselj(n,k_s2.*b);
d_jnks2b = diff(besselj(n,k_s2_2.*b))./diff(k_s2_2.*b);

ynks2b = bessely(n,k_s2.*b);
d_ynks2b = diff(bessely(n,k_s2_2.*b))./diff(k_s2_2.*b);

jnk3b = besselj(n,k_3.*b);
d_jnk3b = diff(besselj(n,k_3_2.*b))./diff(k_3_2.*b);



d11=(rho1/rho2).*k_s2.^2.*a^2.^hnk1a;
d12=(2*n^2-k_s2.^2.*a^2).*jnkd2a-2.*k_d2.*a.*d_jnkd2a;
d13=(2*n^2-k_s2.^2.*a^2).*ynkd2a-2.*k_d2.*a.*d_ynkd2a;
d14=2*n.*(k_s2.*a.*d_jnks2a-jnks2a);
d15=2*n.*(k_s2.*a.*d_ynks2a-ynks2a);

d21=-k_1.*a.*d_hnk1a;
d22=k_d2.*a.*d_jnkd2a;
d23=k_d2.*a.*d_ynkd2a;
d24=n.*jnks2a;
d25=n.*ynks2a;

d32=2*n.*(jnkd2a-k_d2.*a.*d_jnkd2a);
d33=2*n.*(ynkd2a-k_d2.*a.*d_ynkd2a);
d34=2*k_s2.*a.*d_jnks2a+(k_s2.^2.*a.^2-2*n^2).*jnks2a;
d35=2*k_s2.*a.*d_ynks2a+(k_s2.^2.*a.^2-2*n^2).*ynks2a;
```

```
d42=-2.*k_d2.*b.*d_jnkd2b+(2*n^2-k_s2.^2.*b^2).*jnkd2b;
d43=-2.*k_d2.*b.*d_ynkd2b+(2*n^2-k_s2.^2.*b^2).*ynkd2b;
d44=2*n.*(k_s2.*b.*d_jnks2b-jnks2b);
d45=2*n.*(k_s2.*b.*d_ynks2b-ynks2b);
d46=(rho3/rho2).*k_s2.^2.*b^2.*jnk3b;

d52=k_d2.*b.*d_jnkd2b;
d53=k_d2.*b.*d_ynkd2b;
d54=n.*jnks2b;
d55=n.*ynks2b;
d56=-k_3.*b.*d_jnk3b;

d62=2*n.*(jnkd2b-k_d2.*b.*d_jnkd2b);
d63=2*n.*(ynkd2b-k_d2.*b.*d_ynkd2b);
d64=2.*k_s2.*b.*d_jnks2b+(k_s2.^2.*b.*2-2.*n.^2).*jnks2b;
d65=2.*k_s2.*b.*d_ynks2b+(k_s2.^2.*b.^2-2.*n.^2).*ynks2b;

A_star_1=-(rho1/rho2).*k_s2.^2.*a.^2.*jnk1a;
A_star_2=k_1.*a.*d_jnk1a;

T = zeros(1,length(w));
for ind = 1:length(w)

B=[real(d11(ind))   d12(ind)     d13(ind)     d14(ind)     d15(ind)     0;
real(d21(ind))  d22(ind)     d23(ind)     d24(ind)     d25(ind)     0;
0               d32(ind)     d33(ind)     d34(ind)     d35(ind)     0;
0               d42(ind)     d43(ind)     d44(ind)     d45(ind)     d46(ind);
0               d52(ind)     d53(ind)     d54(ind)     d55(ind)     d56(ind);
0               d62(ind)     d63(ind)     d64(ind)     d65(ind)     0];
```

```
D=[d11(ind)     d12(ind)     d13(ind)     d14(ind)     d15(ind)     0;
d21(ind)     d22(ind)     d23(ind)     d24(ind)     d25(ind)     0;
0            d32(ind)     d33(ind)     d34(ind)     d35(ind)     0;
0            d42(ind)     d43(ind)     d44(ind)     d45(ind)     d46(ind);
0            d52(ind)     d53(ind)     d54(ind)     d55(ind)     d56(ind);
0            d62(ind)     d63(ind)     d64(ind)     d65(ind)     0];

    T(ind) = -det(B)/det(D);

end
```

C.5 ROC.M

```
function [spec, sens] = roc(t, c, n,titlelabel,tlabel,clabel)
%%% computes and plots single feature roc curve based upon n-bin histogram,
%%% also plots the histograms
%%% plots are suppressed if output args are present
%
%
% t - feature value
% c - feature value
% n - number of bins in histogram
% titlelabel - title of figure
% tlabel - label corresponding to t values
% clabel - label corresponding to c values
% spec - specificity
% sens - sensitivity

[thist,tx] = hist(t,n);
```

```
[chist,cx] = hist(c,n);

mean(t)
mean(c)
if (mean(t)>mean(c))
    tmpx = tx;
    tx = cx;
    cx = tmpx;

    tmphist = thist;
    thist = chist;
    chist = tmphist;
end

xmin = min(min([tx cx]));
xmax = max(max([tx cx]));
step = ((xmax-xmin)./(n));
sens=zeros(1,n);
spec=zeros(1,n);
count = 1;
for ind = (xmin+step):step:xmax
    class1_right = sum(thist(tx<=ind));
    class1_wrong = sum(thist(tx>ind));
    class2_right = sum(chist(cx>ind));
    class2_wrong = sum(chist(cx<=ind));
    sens(count) = class1_right/(class1_right+class1_wrong);
    spec(count) = class2_right/(class2_right+class2_wrong);
    count=count+1;
end
spec=[1 spec];
```

```
sens=[0 sens];
if (nargout==0)
    figure;subplot(211)
    hist([t' c'],n);title(['Histograms of ', titlelabel]);
    legend(tlabel,clabel,'Location','NorthEast');

    subplot(212)
    plot(0:1/n:(1-1/n),0:1/n:(1-1/n),':');
    hold on;
    plot(1:-1/n:1/n,1/n:1/n:1,':');
    plot(1-spec,sens,'k');

    title(['ROC curve for ', titlelabel]);
    ylabel('Probability of Correct Classification');
    xlabel('Probability of False Positive');
end
```

C.6 ROC3.M

```
function [spec1, sens1, spec2, sens2] = roc3
(t1, c1, n1,momtype1,t2, c2, n2,momtype2,figtitle,colorflag)
%%% plots double roc curve and histogram
%
% t - feature value
% c - feature value
% n - number of bins in histogram
% titlelabel - title of figure
% tlabel - label corresponding to t values
% clabel - label corresponding to c values
```

```
% colorflag - 0 generate plots in color, 1 for b&w
% spec - specificity
% sens - sensitivity

[spec1,sens1]=roc(t1, c1, n1);
[spec2,sens2]=roc(t2, c2, n2);

figure;
hold on;

if colorflag==1
    plot(1-spec1,sens1,'--','LineWidth',2,'Color','k');
    plot(1-spec2,sens2,'-','LineWidth',2,'Color','k');
else
    plot(1-spec1,sens1,'--','LineWidth',2,'Color','k');
    plot(1-spec2,sens2,'-','LineWidth',2);
end

legend(momtype1, momtype2,18,'Location','SouthEast');
plot(0:1/n1:1,0:1/n1:1,':','Color','k');
plot(1:-1/n1:0,0:1/n1:1,':','Color','k');
axis([-0.005 1 0 1.005]);
ylabel('Probability of Correct Classification','FontSize',14);
xlabel('Probability of False Positive','FontSize',14);
title(figtitle,'FontSize',16);
```

BIBLIOGRAPHY

[1] O. Aluko, "Implementation and Application of Dispersion-Based Waveguide Models for Shallow-Water Sonar Processing." MS Thesis, University of Pittsburgh, 2004.

[2] L. Brekhovskikh and Y. Lysanov, *Fundamentals of Ocean Acoustics.* Springer-Verlag, New York, 1982.

[3] W. Burdic, *Underwater Acoustic Systems Analysis.* Peninsula Publishing, Los Altos, 2002.

[4] L. Cohen, *Time-Frequency Analysis,* Prentice-Hall, 1995.

[5] L. Cohen, "Pulse propagation in dispersive media," *IEEE Proc. SSAP 2000,* pp. 485-489, 2000.

[6] K. Davidson "Instantaneous Moments of a Signal," Ph. D. Dissertation, University of Pittsburgh, 2000.

[7] M. Ewing, J. L. Worzel, and C. L. Pekeris, *Propagation of Sound in the Ocean,* Geo. Soc. of Am., 1948.

[8] L. Flax and L. Dragonette, "Theory of elastic resonance excitation by sound scattering," *J. Acoustic. Soc. Am.*, Vol. 63, Issue 3, Mar 1978.

[9] A. P. French, *Vibrations and Waves,* W. W. Norton & Co., New York, 1971.

[10] G. Gaunaurd and D. Brill, "Acoustic spectrogram and complex-frequency poles of a resonantly excited elastic tube,"*J. Acoustic. Soc. Am.*, vol. 75, no. 6, pp. 1680-1693, 1984.

[11] G. Gaunaurd and H. Überall "Relation Between Creeping-wave Acoustic Transients and the Complex-frequency Poles of the Singularity Expansion Method." *J. Acoustic. Soc. Am.,* vol. 78, no. 1, July 1985.

[12] G. Gaunaurd and W. Wertman, "Transient Acoustic Scattering By Fluid-loaded Elastic Shells," *Int. J. Solids Structures*, vol. 27, no. 6, pp. 699-711, 1991.

[13] G. Gaunaurd and H. Strifors, "Frequency- and Time-Domain Analysis of the Transient Resonance Scattering Resulting from the Interaction of a Sound Pulse with Submerged Elastic Shells," *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 40, no. 4, pp. 313-224, 1993.

[14] F. Jensen, *et. al.*, *Computational Ocean Acoustics* Springer-Verlag, New York, 2000.

[15] J. Lighthill, *Waves in Fluids,* Cambridge University Press, chap. 3, sec. 3.7, 1978.

[16] P. Loughlin and L. Cohen, "Moment Features Invariant to Dispersion." *Proc SPIE Vol. 5426, Automatic Target Recognition XIV*, Sep 2004.

[17] P. Loughlin and L. Cohen, "Phase-space Approach to Wave Propagation with Dispersion and Damping," *Proc SPIE Vol. 5559, Advanced Signal Processing Algorithms, Architectures, and Implementations XIV*, Oct 2004.

[18] T. Mitchell, *Machine Learning,* McGraw-Hill, Singapore, 1997.

[19] G. Okopal, P. Loughlin, and L. Cohen, "Recognition of Propagating Vibrations and Invariant Features for Classification," *Proc SPIE Vol. 6234, Automatic Target Recognition XVI*, May 2006.

[20] A. V. Oppenheim and R. W. Schafer *Discrete-Time Signal Processing* Prentice-Hall, 1989.

[21] I. Tolstoy and C. Clay, *Ocean Acoustics: Theory and Experiments in Underwater Sound,* American Institute of Physics, NY, 1987.

[22] C. Tsui, G. Reid, and G. Gaunaurd, "Resonance scattering by elastic cylinders and their experimental verification," *J. Acoust. Soc. Am.*, vol. 80, no. 2, pp. 382-390, 1986.

[23] G. Whitham, *Linear and Nonlinear Waves,* J. Wiley and Sons, New York, 1974.