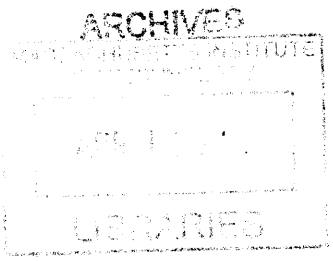
Real Time Active Sonar Simulation in a Deep Ocean Environment



by

Sheida Anya Danesh

BS, University of Rhode Island (2010)

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2013

Author .....
Department of Mechanical Engineering
October 1, 2012

Certified by .....
Prof. Henrik Schmidt
Professor of Mechanical and Ocean Engineering
Massachusetts Institute of Technology
Thesis Supervisor

Accepted by .....
Dr. David E. Hardt
Chair, Committee on Graduate Students - Mechanical Engineering
Massachusetts Institute of Technology

Real Time Active Sonar Simulation in a Deep Ocean Environment

by

Sheida Anya Danesh

Submitted to the Department of Mechanical Engineering
on October 1, 2012, in partial fulfillment of the
requirements for the degree of
Master of Science in Mechanical Engineering

Abstract

Accurate Simulations are important in all fields of science and engineering where experiments and real data are time consuming to obtain and are not easily accessible and/or free of cost. Particularly when operating in environments such as the deep ocean where it could take 2-3 hours for an autonomous underwater vehicle (AUV) to get to a specific depth and where power is very limited. A new module called uSimActiveSonar has been developed in order to simulate an active sonar system with hydrophone data acquisition. This has been done through the use of middleware Mission Oriented Operating Suite (MOOS), ray tracing code BELLHOP as well as using the method outlined by the Naval Research Laboratory to simulate the effects of surface reverberation.

Thesis Supervisor: Prof. Henrik Schmidt
Title: Professor of Mechanical and Ocean Engineering
Massachusetts Institute of Technology

Dedication

This thesis is dedicated to my parents Dr. Abol Hassan Danesh and Mrs. Sholeh Ghedari (Danesh), for their love and constant support. Thanks for always believing in me, usually more than I ever believed in myself.

Acknowledgments

First and foremost, I would like to thank Dr. Henrik Schmidt for his contagious excitement in the field as well for his constant patience for my infinite questions. I also would like to thank Professor Schmidt along with Erin Fischell and Kevin Cockrell for writing the baseline Matlab version of the active sonar simulator for use inthe SWAMSI '11 experiments. Thirdly, I would like to thank the other members of LAMSS: Toby Schneider, Stephanie Petillio, Stephanie Fried, Arjuna Balasuriya, Ian Katz, Alon Yaari, Keja Rowe and Raymund Lum for helping me debug my problems one step at a time, as well as Vanessa D'Amico her editorial comments

And lastly I would like to thank you, Robert, for your unconditional love and patience for me while I lost my mind.

Contents

List of Figures

Chapter 1

Introduction

How inappropriate to call this planet Earth when it is quite clearly Ocean.

Arthur C. Clarke

1.1 Motivation

It is very important to construct accurate simulations, when testing modules before experimentation is necessary. At the Laboratory for Autonomous Marine Sensing Systems (LAMSS) at MIT there are currently many modules in place for use on autonomous underwater vehicles (AUVs). If these modules are to be used in deep ocean, a credible simulator is needed that will output the effects of an active sonar system in such an environment. Simulations save precious ship time and also help to improve other existing modules.

1.2 Objectives

In creating an accurate deep ocean active sonar simulator, the main objectives are as follows:

1. Accurately simulate environmental effects

2. Simulate effects of having a target at any range and depth.

3. Create time series

1.3 Organization

Chapter 2 will present background information about AUVs and use of the middleware MOOS. Chapter 3 will present different techniques of modeling sound propagation in the ocean with an emphasis on raytracing. Chapter 4 will present the method and logic for implementing the simulator. The two main subsections in Chapter 4 will be the way in which the active sonar simulator deals with effects from a target and the effects from the surrounding environment. Chapter 5 will discuss resuls, conclusions and areas for futures improvements and research.

Chapter 2

AUVs and MOOS

2.1 AUVs

Increasingly Autonomous Underwater Vehicles (AUVs) are replacing remotely operated vehicles (ROVs) and submarines in underwater applications. AUVs are commonly used for applications such as mine counter measures, anti-submarine warfare, bathymetry mapping, and area surveillance. AUVs assist with mine counter measures through their ability to classify targets that are either buried, partially buried, or sitting on the sea floor as either hazardous mines or benign objects (rocks). This eliminates the need for divers and could ultimately save lives. AUVs also assist with seabed mapping via regular sonar and side scan sonar. Mapping out bathymetric features this way is very beneficial because it can prevent accidents such as ships running aground. AUVs also eliminate the need to both construct expensive submarines and risk the lives of submariners during applications such as anti-submarine warfare in target tracking. AUVs can search, detect and track using various algorithms, and have the ability to go to very low depths without the need for a tether (Lynch, 2012) and (Benjamin and Schmidt, 2012).

Popular middlewares for AUVs include Robotic Operating System (ROS), Mission Orienented Operating Suite (MOOS), Lightweight Communicating and Marshalling (LCM), and Microsoft Robotics Studio. At LAMSS at MIT, MOOS is the middleware of choice.

2.2 MOOS

As mentioned in the previous section, Mission Oriented Operating Suite, or MOOS, is the middleware used on the AUVs at LAMSS to control autonomy and the interfaces of other processes. Developed by Paul Newman in 2001, MOOS middleware has a publish-subscribe architecture by which different applications connect to one central MOOS database, or MOOSDB. The MOOSDB then acts like a mailman of sorts and passes information back and forth between different applications. There is no communication between applications directly. In fact, each application has no idea what the other applications publish to the DB or even if other applications exist at all! A publish-subscribe architecture means that applications can "publish" different variables (in the form of a double or a string) to the MOOSDB. They can also "subscribe" to other variables that may be published by different applications (Newman). For example, Figure 2-1 demonstrates the publish-subscribe architecture

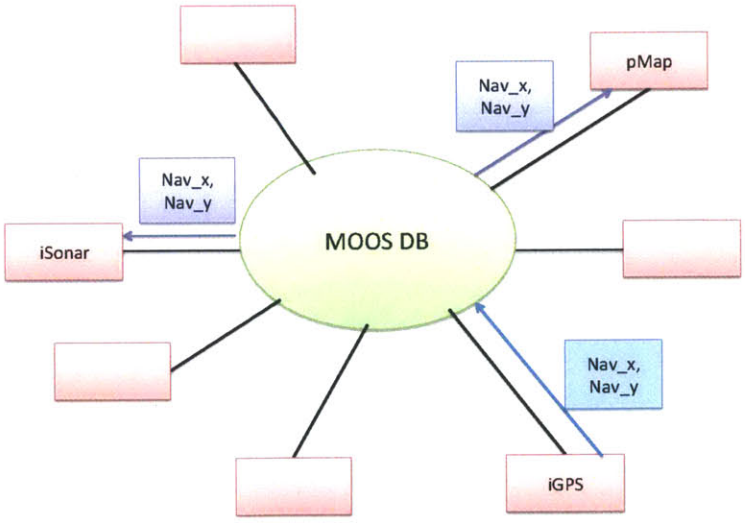


Figure 2-1: MOOS Publish Subscribe Architecture

and the functionality of the MOOSDB. In this case,there is one central MOOSDB with serveral applications/clients, or "MOOSApps", communicating through it. Although there may be more applications, let focus on three main MOOSApps: pMap, iGPS, and iSonar. In this simple example, iGPS connects to an actual GPS and is capable

of knowing the vehicle position. NAV-X and NAV-Y are the MOOS variables for the x and y location of the vehicle. pMap is a theoretical application that shows the vehicle position on a map and iSonar is a theoretical app that fires a sonar when the vehicle is at a certain (x,y) position. Here, pMap and iSonar need to know the vehicle position in order to perform their respective functions. Every time iGPS gets new NAV data, it will publish it to the MOOSDB. Meanwhile, iSonar and pMap have subscribed to variables called "NAV-X" and "NAV-Y". Every time iGPS (or any other application for that matter) refreshes this variable value, iSonar and pMap will get new information. The elegance of this set up is that any and all applications can have a need for NAV data but these needs do not affect the functionality of iGPS at all. Also, pMap will not be compromised if iGPS no longer exists and thus, NAV data must come from a different app (e.g. iCompass). The only thing that must be consistent is the name of the MOOS variables.

2.3 Current Simulation Environment

The current simulation setup launches various existing MOOS applications when certain switches are activated. The simulation environment that is applicable to the new active sonar MOOSApp is the active sonar simulation launch. Table 2.1 displays the names and functions of the relevant MOOSApps that are launched in this mode.

Such advanced simulation capabilities enable us to remove any of the uSim-modules and replace them with actual interface systems without affecting any other processes. In this case, the module that is being augmented is uSimActiveSonar. (It is important to note that a baseline MATLAB script that took care of some simulated active sonar effects with a target in shallow water was in place already. The new version of uSimActiveSonar, has been completely rewritten in C++ with more functionality, especially for deep ocean. The detailed updates to this application can be found in Chapter 4).

uSimActiveSonar uses ray tracing as its method of acoustic prorogation. The

MOOSApp	Function
pHelmIvP	Control's vehicle autonomy
pAcommsHandler	Deals with acoustic communication coming into and out of the vehicle
iBellhop	Interface for raytracing code BELLHOP
pActiveSonarControl	Publishes information about desired sonar beamwidth, source strength, waveform etc.
pRBTracker	Target Tracker
uSimTargets	Publishes target location, speed, heading, target strength, and other information
uSimTowedArray	Simulates a towed array at the end of the vehicle and publishes location of each hydrophone in (x,y,z)
uSimActiveSonar	Simulates sonar source and hydrophone receiver output
pActiveSonarProcess	Processes data published by uSimActiveSonar in order to calculate Time of Arrival and Direction of Arrival of a target (simulated by uSimTargets)

Table 2.1: MOOSApps in Active Sonar Simulation Environment

raytracing code BELLHOP is used through an interface called iBellhop. iBellhop subscribes to what is called a "BELLHOP-REQUEST". Within this request the user can define the type of output file desired (.ray file, .arr file etc) as well as the sound speed profile, depth, maximum and minimum value of the launch angle, number of rays, source and receiver positions, etc (Schneider). Usually all of this is included in an environmental file that BELLHOP reads (Porter, 2011). iBellhop, however, is made to interface with MOOS and other MOOSApps. The two main file types that are needed for this simulator are .ray files and .arr files. The .ray files contain information about the range and depth (r,z) location of each ray as it moves through time. Here, no receiver position needs to be defined since the .ray file is just showing how each ray propagates; however, the arrival file needs defined receiver positions. At each receiver, BELLHOP determines if there are any eigenrays depending on where the source is (this also depends on the number of rays. For instance, if there are more rays, then there will be a higher likely hood of an eigenray if the rays cross paths with the receiver). For each eigenray, the arrival file contains information such as travel time, amplitude (transmission loss), source angle, receiver angle, number of top bounces, and number of bottom bounces (Porter, 2011). This information is crucial to the implementation of the simulator.

Chapter 3

Propagation Techniques

The simulator models underwater sound propagation using ray tracing. In this section we will discuss a brief history of ray tracing, show a derivation of its major equations as well as its advantages and disadvantages. A short description of other propagation methods will also be presented in this section.

3.1 Ray Tracing

3.1.1 History and Intuition

Ray theory gives us a very intuitive feel for how sound propagates in the ocean. Snell's law, developed in the 1600s for optics, describes the direction of light when there are two media with different indexes of refraction. In the ocean, sound waves behave the same way thus we can substitute the sound speed of the medium for the index of refraction

As seen in equation 3.1, the relationship between the sound speed of the medium, c, and the cosine of the grazing angle, θ is a constant.

$$\frac{c_1}{cos\theta_1} = \frac{c_2}{cos\theta_2} = const. \tag{3.1}$$

This makes a lot of intuitive physical sense. If we modify 3.1 and use the definitions in 3.2 and 3.3, taking ω to be the angular frequency and k to be the

wavenumber, we get the result in 3.4, where V_x is the horizontal phase speed. Physically, the horizontal phase speed needs to be a constant. If not, the boundary would be moving! (Dyer)

$$\omega = 2\pi f \tag{3.2}$$

$$k = \frac{\omega}{c} \tag{3.3}$$

$$\frac{\omega}{k_1 cos\theta_1} = \frac{\omega}{k_2 cos\theta_2} = V_x \tag{3.4}$$

In order to get a better feel for how the rays are propagating, we add a few more layers to our initial image. If we look at figure 3-2, we notice that when each progressive layer has a faster sound speed than the one before it, the rays tend to bend up.

The basic equations of ray tracing show that for a linearly increasing sound speed profile, rays bend in circular arcs. Equations 3.5 - 3.7 demonstrate this. Here $r(s)$ and $z(s)$ are range and depth respectively of each ray path, R is the radius of the arc, θ_0 and θ are the launch angle and the incident angle at the the end of the trajectory, g is the gradient of the linear sound speed profile, $c(z_0)$ is the sound speed at the launch depth, and $c(0)$ is the sound speed at the surface. If the sound speed profile is not linear, a general rule of thumb is that rays will bend towards lower sound

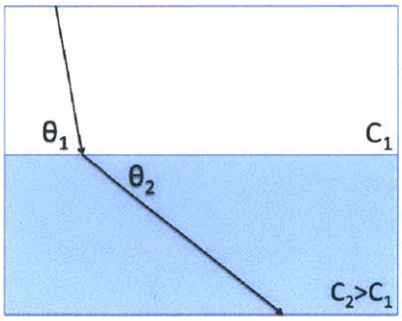


Figure 3-1: Snell's Law

speed (Jensen et al., 2011).

$$r(s) = Rsin\theta_0 - Rsin\theta \tag{3.5}$$

$$z(s) = Rcos\theta - \frac{c(0)}{g} \tag{3.6}$$

$$R = \frac{c(z_0)}{gcos\theta_0} \tag{3.7}$$

There are a few different paths that a ray of sound can follow due to the boundaries set in place by the sea. They are: the direct path, the path reflected from surface (not too much loss), and the path reflected from the seabed (much more loss). Each path will produce a different result depending on the type of environment the sonar system is operating in. In deep sea environments,with the source near the sea floor, sound can travel for very long distances (45km) without incurring any bottom loss. However, in shallow water environments, the rays will not be able to travel as far because they continuously bounces from the surface to the bottom. Of course, the distance the ray travels, not only depends on the environment, but also on where the source itself is placed as well as frequencies used. Sound will travel farthest when the source is placed in the deep sound channel axis because this is where sound speed is at a minimum. The deep sound channel axis is usually found about 1000m below

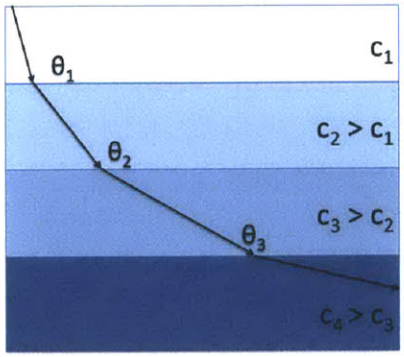


Figure 3-2: Snell's Law with Multiple Layers

the sea surface. When a source is placed here, the propagating ray will turn down before it hits the surface and turn up before it hits the seafloor. Lower frequencies attenuate less and have much longer ranges (Marage and Mori, 2010).

3.1.2 Derivation

In deriving the major ray tracing equations I will be following the method outlined in (Jensen et al., 2011).

$$\nabla^2 P + \frac{\omega^2}{c^2(\mathbf{x})} P = -\delta(\mathbf{x} - \mathbf{x}_0) \tag{3.8}$$

If we start by trying to solve the Helmholtz equation above (3.8), we will end up with two equations to solve. One of these equations solves for the actual ray path (position: range and depth), also known as the Eikonal equation. The second one, known as the Transport equation, solves for the amplitude across the ray tube.

These two equations come from taking the second derivative of a series solution to the Helmholtz equations, plugging them back into the Helmholtz equations and making a first order approximation. Due to the first order approximation (where only the first term in the series is used) ray tracing is considered to be a high frequency approximation. A situation can be considered "high frequency" if the wavelength is small compared to the depth and the bathymetric features of interest. In the Eikonal equation 3.9, $\tau(x)$ represent the equation of the wavefronts, which, by definition, are considered to be perpendicular to the ray path. (See Figure 3-3). Because of this, we can reduce 3.9 to 3.10.

$$(\nabla\tau)^2 = \frac{1}{c^2(x)} \tag{3.9}$$

$$\frac{dx}{ds} = c\nabla\tau \tag{3.10}$$

Here $\mathbf{x}(s)$ is the path that the ray follows. After manipulating this a few steps

further we get the general solution to the Eikonal equation in 3.11.

$$\frac{d}{ds}(\frac{1}{c}\frac{dx}{ds}) = -\frac{1}{c^2}\nabla c \tag{3.11}$$

We now simplify 3.11 by writing it in cylindrical coordinates (we assume everything is axis-symmetric). In order to make this simplification we also have to assume that the sound speed is only a function of depth and not range. (3.12 - 3.15)

$$\frac{dr}{ds} = \xi(s) \tag{3.12}$$

$$\frac{dz}{ds} = \zeta(s) \tag{3.13}$$

$$\frac{d\xi}{ds} = -\frac{1}{c^2}\frac{dc}{dr} \tag{3.14}$$

$$\frac{d\zeta}{ds} = -\frac{1}{c^2}\frac{dc}{dz} \tag{3.15}$$

These are the general ray tracing equations. More specific results can be obtained once the sound speed profile and some initial conditions are known.

Moving on to the Transport equation, 3.16, we are able to find the amplitude along the ray path. If we first reduce the original transport equation to the form seen

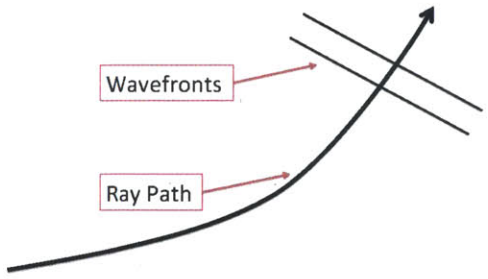


Figure 3-3: Ray Path and Wavefronts

in 3.17 and the apply the divergence theorom, we get the result seen in 3.18.

$$2\nabla\tau \bullet \nabla A_0 + (\nabla^2\tau)A_0 = 0 \tag{3.16}$$

$$\nabla(A_0^2\nabla\tau) = 0 \tag{3.17}$$

$$\int_{dV} A_0^2\nabla\tau \mathbf{n} dS \tag{3.18}$$

Using (Jensen et al., 2011)'s definition of a ray tube as "the volume enclosed by a family of rays" we can define $\mathbf{n}$, the normal, as pointing in the same direction as each of the rays. Therefore we can now say that $\mathbf{n} = \frac{dx}{ds}$. We can also rewrite 3.10 as $\nabla\tau \bullet \mathbf{n} = \frac{1}{c}$. Now we finally have the Transport equation in the form of 3.19.

$$\int_{dV} A_0^2 \frac{1}{c} dS \tag{3.19}$$

It is important to note that although the volume changes as we move along the ray bundle, the value of this integral stays constant. Jensen et al. (2011) tells us that the final result will have look like 3.20, where J is related to the raytube's cross-sectional area by 3.21

$$A_0(s) = A_0(0)\sqrt{\frac{c(s)J(0)}{c(0)J(s)}} \tag{3.20}$$

$$J = r\sqrt{\frac{dx}{d\theta_0}^2 + \frac{dr}{d\theta_0}^2} \tag{3.21}$$

We include the extra r factor because we are assuming everything to be axis-symmetric about the source.

3.1.3 Advantages and Disadvantages

One of the main advantages of using ray tracing is its transparency with regard to the physics of sound propagation. Intuitively, we expect circular arcs when there

is a linear sound speed profile, when we have a non-linear sound speed profile we can still visually see what is going on in range. Another very important advantage of using ray tracing is that it is much more computationally efficient than other methods such as normal modes and wavenumber integration, especially for long ranges and high frequencies.

Ray tracing has disadvantages as well. For instance, when using ray tracing methods, one must make heavy use of approximations. As discussed in the previous section, one of these is the high frequency approximation. This illustrates that ray tracing is not accurate for any and all frequencies (Jensen et al., 2011). Additionally, depending on the trajectory of the ray, there may be areas where intensity soars off to infinity. This occurs because in traditional ray tracing the intensity of the ray is inversly proportional to the cross-sectional area of the ray tube. If the ray were to bend and the cross-sectional area were to temporarily go to zero, then the intensity at that particular point would go to infinity. Clearly this is not what happens in real life. A third shortcoming of ray tracing is its ability to handle shadow zones. There are sharp discontinuities between a region of ray tubes and the shadow zone. In reality this would be much more continuous (Porter and H.P, 1987).

3.1.4 Gaussian and Geometric Beams

In order to offset most of the shortcomings discussed in the previous subsection, we can use certain secondary approximations. One method is to develop an intensity distribution for each ray. In figure 3-4, we can see two different types of intensity profiles. The top one (blue) is known as a geometric beam while the bottom one (red) is known as a Gaussian beam. Here, the intensity distribution of each ray is a Gaussian bell curve (Porter and H.P, 1987).

These Gaussian Beams smooth out the transitions between rays and shadow zones and also smooth out intensities at caustics. The main challenge with Gaussian Beams is picking the correct Gaussian profile with the optimal width (Jensen et al., 2011). Gaussian beams actually provide us with an approximation that is closer to the exact solution of the wave equation. Porter and H.P (1987) developed a method

to solve for the optimum parameters of the Gaussian beam profile. This Gaussian beam approximation, although is closer to a precise physical solution than traditional ray tracing, also requires a high frequency approximation.

3.2 Other Methods

3.2.1 Normal Modes

Normal Modes is a rather popular propagation technique. It uses no high frequency approximation and it does not necessarily need to make a far field approximation. If we define a Green's function as a solution to the Helmholtz equation, Normal Modes creates this Green's function through eigenfunction expansion. Any other system that vibrates and has some semblence of waves and frequency associated with it also has some natural modes associated with it. A violin string for example can have many modes. Depending on where the string is plucked, different modes will be excited. The first three modes are shown in Figure 3-5.

In this respect, acoustics behave the same way. If we reduce the wave equation, assuming cylindrical symmetry with variations only in depth, according to Frisk (1994), we can put it into the form of a Sturm-Liouville eigenvalue equation (which has a known solution). It is important to note the assumptions of a proper Sturm-Liouville equation. In terms of ocean acoustic boundary conditions, the Sturm-Louville equa-

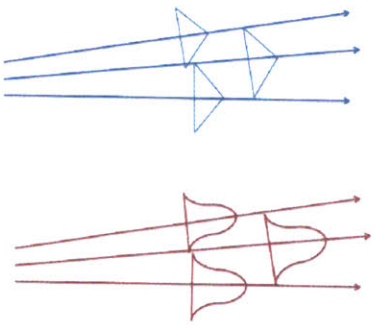


Figure 3-4: Geometric and Gaussian Beams

tions will satisfy the velocity boundary conditions but will not satisfy the impedance boundary conditions. This is because in these Sturm-Louville equations, all energy is trapped and therefore attenuation to the seabed can not be taken into account directly. The general form of a normal mode equation can be seen in equation 3.22.

$$p(r,z) = \Sigma a_n(z_0)u_n(z)R_n(r) \tag{3.22}$$

Here $a_n(z_0)$ refers to the amplitude at each mode, $u_n(z)$ refers to each vertical mode function and $R_n(r)$ represents the radial function. For a simple situation in which we have a waveguide with a pressure release surface, a perfectly reflecting seabed with a fixed sound speed and density, and a point source, our pressure equation in range and depth will look like 3.23.

$$p(r,z) = \frac{2i\pi}{h}\Sigma sin(k_{zn})z_0 sin(k_{zn}z)H_0^1(k_n,r) \tag{3.23}$$

Here (r,z) are range and depth. Just as in equation 3.22, we have our amplitude function, our vertical eignenfunction equation (the sine functions) as well as a radial function. Here the radial function is the Hilbert transform. If we make a farfield approximation, we can remove the Hilbert transform and replace it with $\frac{e^{ik_n r}}{\sqrt{k_n r}}$.

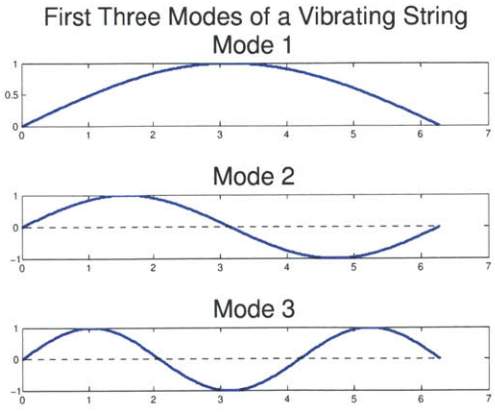


Figure 3-5: String Modes

We can see that even for a very simple case, these equations require us to find the entire field in order to see where the sound is propagating (Frisk, 1994). As mentioned before, we are assuming a constant sound speed profile. If we were to assume a more complicated sound speed profile, we would probably have to use either some form of finite difference method or perturbation theory. Again, this method is computationally exhaustive and would not work well for the needs of the desired simulator. It is important to note, however, that rays and modes generate similar results. The

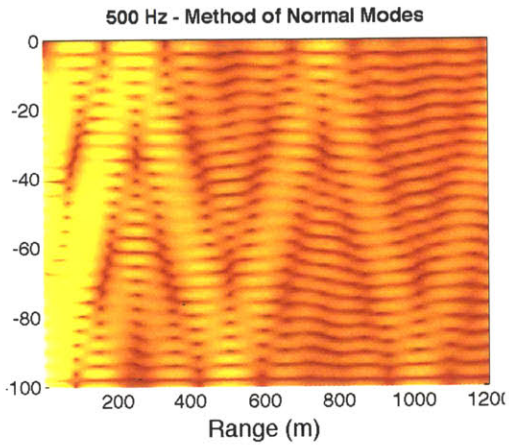


Figure 3-6: 500 Hz Normal Mode Propagation acting like Rays

higher the frequency, the more the mode results resemble rays. We can use a very simple example to demonstrate this phenomena. Again, we assume constant sound speed with a pressure release surface and a perfectly reflecting bottom with a point source at the bottom of the waveguide. For display purposes, we only take into account those eigenfunctions with 20 degree grazing angles (eigenangles of 70 degrees). Once we have these eigenfunctions and eigenvalues, we only take the ones that add constructively (with cycle distances only off by a quarter of a wavelength). With a 500 Hz source, we find the result displayed in Fgure 3-6. If we increase the frequency to 1000 Hz and apply the same method, we end up with the result displayed in figure 3-7.

As we increase the frequency the mode solution looks more and more like the ray solution. This confirms the rigor of ray tracing as a high frequency approximation.

3.2.2 Wavenumber Integration

The wavenumber integration approach is only applicable to horizontally stratified media. That is, the physical characteristics of the halfspace must only change in depth, not range. Wavenumber integration divides the halfspace into a number of horizontal slices and solves each slice. Each slice is solved independently from the other slices; however, boundary conditions between the slices much be upheld (Jensen et al., 2011).

Wavenumber integration is sometimes referred to as the Hankel transform. This approach is just a different way to solve the Helmholtz equation and find out Green's function. Using this method, a 1-D Hankel transform in cylindrical coordinates can express the same result as a 2-D Fourier transform in Cartesian coordinates. If we define the Hankel transform, as in Frisk (1994), we get equation 3.24.

$$I.H.T[\bullet] = \int_0^\infty [\bullet] J_0(k_r r) r dr \tag{3.24}$$

If we apply this operator to the the inhomogeneous time-independent wave equation, we get the expected result of the 1-D Hankel transform, where the expressions for pressure and Green's function are zeroth order Hankel transforms of one another. This result is seen in 3.25. (Frisk, 1994)

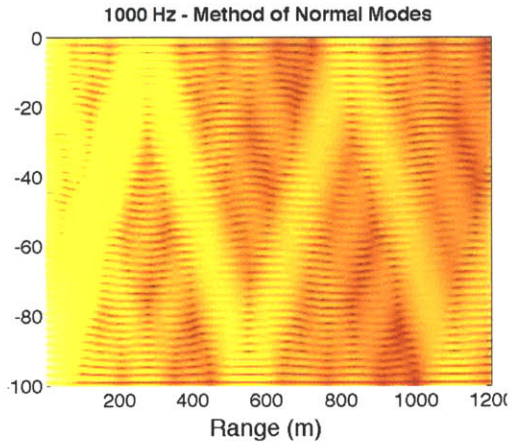


Figure 3-7: 1000 Hz Normal Mode Propagation acting like Rays

$$I.H.T(\frac{1}{r}\frac{\delta}{\delta r}[r\frac{\delta p(r;z,z_0)}{\delta r}]) = k_r^2 g(r;z,z_0) \tag{3.25}$$

Because wavenumber integration solves each horizontally stratified level separately, more complicated sound speed profiles will require more layers (and thus more computation). Also, for higher frequencies wavenumber integrations requires higher resolution, and therefore will also take too long.

Although ray tracing has its own shortcomings, for the purposes of this simulator, the computational advantages largely outweigh the inaccuracies set in place by our assumptions (especially at the frequencies being used).

Chapter 4

uSimActiveSonar

4.1 The Problem

The basic setup of a deep ocean problem is displayed below. As in Fgure 4-1, the depth of the ocean is presumed to be somewhere between 5,000 to 6,000 meters and the AUV is somewhere close to the sea floor. In Figure 4-1 it is shown at a depth of 4,000 to 5,000 meters. The goal of this example mission is to find and track a given target (or targets). Here the target is shown to be relatively close to the sea surface at a depth of somewhere between 0 to 200m. However, it should be noted

Figure 4-1: The Problem

that the simulation environment is completely generic. The AUV and targets can be

simulated at any depth. The AUV is assumed to be equipped with a sonar system which includes a directional mid-frequency (1-5kHz) sonar source as well as an array of hydrophones. The active simulator, or uSimActiveSonar, receives a command from an module called pActiveSonarControl (see Chapter 2) which specifies the source waveform, source strength, elevation angle from the horizontal and beamwidth of the sonar beam. Figure 4-2 illustrates this.

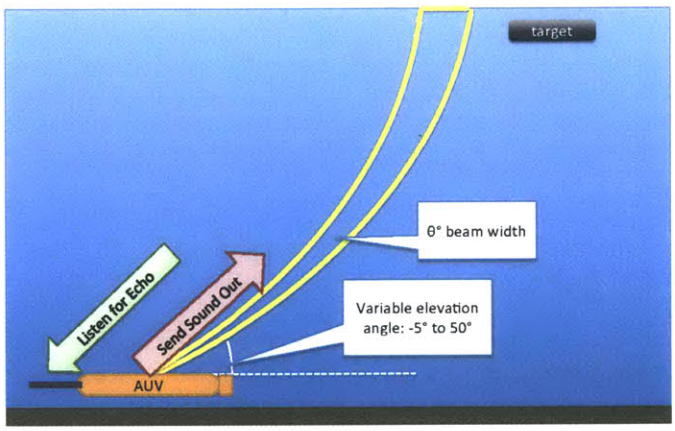


Figure 4-2: Elevation Angle and Beam Width of the Sonar Source

The function of this entire simulator example is to find the target. Ultimately uSimActiveSonar will output the timeseries of what is heard at the hydrophones. This data is then sent to an active sonar processor, or pActiveSonarProcess, which does some processing (using matched filters and other techniques) in order to deduce time of arrival and direction of arrival of the target. The result from pActiveSonarProcess is then sent to a third module, a target tracker (pRBTracker). The tracker then conducts further processing in order to find the (x,y) location of the target. Figure 4-3 demonstrates this. If uSimActiveSonar is to be replaced by a real active sonar system, the rest of the processing chain will remain unchanged.

When creating this simulator, sound pressure such as sound from the target, sound from the environment, and ambient noise must be taken into consideration. When using an active sonar system, the main impact of the environmental noise comes from reverberation. There are three main types of reverberation: sea surface

reverberation, bottom surface reverberation and volume reverberation (due to fish schools).

In this simulator, we only consider sea surface reverberation. If we assume the the aim of this entire processing chain is to try and locate a sea surface target, the main lobe of the sonar source should never directly hit the sea floor. However, in cases where there is some bottom bounce, (if the vehicle is close to the sea floor), the reverberation from the sea floor will come much sooner in time than the time frame with which we are concerned (close to the target).

That being said, even if the main lobe of the source beam pattern is pointed away from the sea floor, we will still encounter leakage in other directions (side lobes).

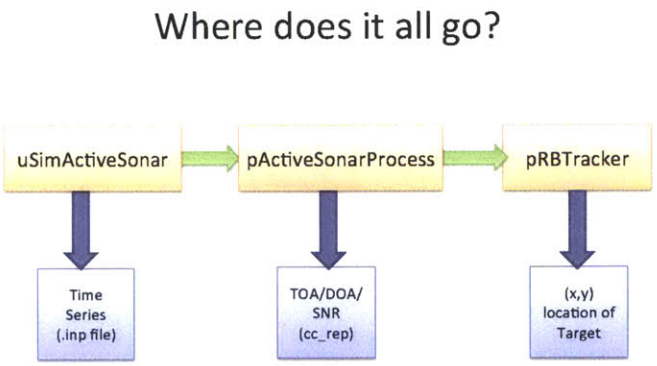


Figure 4-3: Processing Chain of Time Series Information

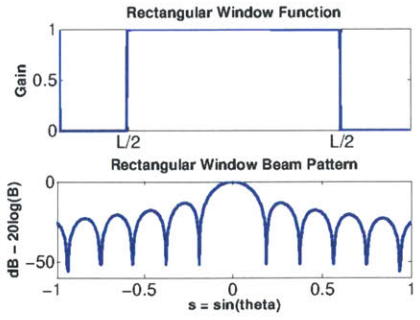


Figure 4-4: Rectangular Window Taper Function and Beam Pattern

These side lobes now could possibly come into contact with the sea floor, causing an echo; however, if we use a Hanning window, we can significantly decreases our side lobes. This is because beampattern and source taper function are Fourier transforms of one another (Dyer). Assuming a line array, Figures 4-4 and 4-5 demonstrate the difference between the beampatterns of a rectangular taper function (constant amplitude) and that multiplied by a Hanning window.

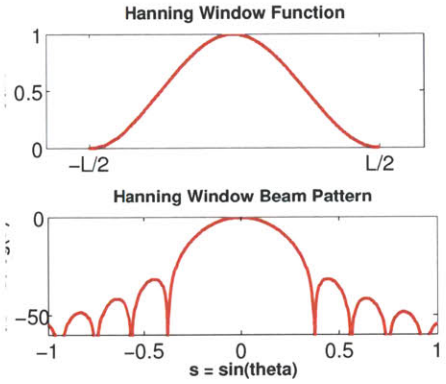


Figure 4-5: Hanning Window Taper Function and Beam Pattern

A rectangular taper function has a desired narrow mainlobe, but also has undesirable high side lobes (only -13 dB down). With side lobes this high, we would almost definitely encounter meaningful reverberation from the sea floor as well as take away from the directionality of the source. The Hanning window will fatten the mainlobe by about two times, but it will also lower the amplitude of the first side lobe to -24 dB (Oppenheim and Schafer, 2010). This amplitude is significantly less than the rectangular window and therefore we can ignore the reverberation coming form the sea floor. A Hanning window will also enhance the desired directionality of the source with less sound going into unwanted directions.

In this case, we have increased the width of our mainlobe by using a Hanning window. However, we are not required to simulate a specific source array. Additionally we know that, being Fourier transforms of one another, the length of the window/source and the width of the main lobe and inversely proportional. Therefore we can assume a longer line array in order to get whatever desired beamwidth that

we need.

4.2 Simulating Surface Reverberation

Surface reverberation is caused by sound that is scattered due to trapped air bubbles and roughness on any 2-D surface. This surface could be the sea floor or the sea surface. On the sea surface scattering changes with angle, frequency, and roughness, which is caused by wave heights and wind above the surface. The effects of surface reverberation become especially important when analyzing deep ocean environments because even at horizontal source launch angles, the rays will still refract up and hit the surface. The grazing angle at the surface is very important because reverberation at different angles will be caused by different phenomena. For grazing angles less than 30 degrees, the main cause of surface reverberation is scattering due to the air bubbles just below the surface. At angles above 70 reverberation is caused reflection off of surface waves. Meanwhile angles from 30 to 70 degrees have very high reverberation caused by a combination of the two reasons (Urick, 1983). In coastal regions higher reverberations results due to winds hitting the coast creating more subsurface bubbles. (McDaniel, 1993).

There are many ways to determine sea surface reverberation. I chose to model the reverberation as a number of discrete sources (or "targets") at the sea surface. In order to do this, just like a regular target, I needed to define a target strength (TS). In this case, I will define this as the Surface Target Strength (STS). Each "source" has a different STS. In order to get this STS, we need to define the area of this source as well as the Scattering Strength (Ss) per unit area.

$$STS = Ss + 10log_{10}(A) \tag{4.1}$$

The "area" in this case is assumed to be the insonified annulus at each range. See Figure 4-6 for clarification. In order to find these areas/ranges, the ray-tracing code BELLHOP is used. BELLHOP works by using the correct beam width and elevation angle and splitting the ray bundle into a definable number of rays (200 in

most cases). A request is sent to BELLHOP to find if and where each of these 200 rays hits the surface. Each of these locations represents a virtual receiver. Figure 4-7 shows a plot of this ray trace with different elevation angles. Then, another request to BELLHOP is sent so the output is in the form of an "arrival file". The "arrival file" contains information for each eigenray that hits each "receiver"(Porter, 2011). As mentioned in Chapter 2, information contained includes source angle, received angle, transmission loss, travel time, etc. In order to keep things simple, only one eignenray is followed. The ray chosen to be followed for each "hit" is the one with the maximum amplitude. Because we have a monostatic configuration, the path from source to each surface hit is assumed to be the same path from the surface to the receiver.

The major challenge in determining the reverberation level is developing a method of calculating the scattering strength, Ss. In the 1960s, Chapman and Harris developed a fully empirical equation (see equation 4.2) to determine Ss using omni-directional explosive sources (Chapman and Harris, 1962).

$$Ss = 3.3\beta log_{10}\frac{\theta}{30} - 42.4log_{10}\beta + 2.6 \tag{4.2}$$

Here, $\beta = 158(vf^{\frac{1}{3}})^{-0.58}$ where v is the wind speed in knots, f is the frequency

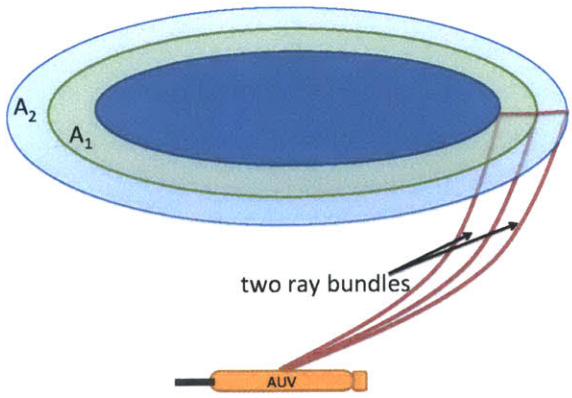


Figure 4-6: Insonified Area for Each Ray Bundle

and θ is the grazing angle (Chapman and Harris, 1962). In creating this model, Chapman and Harris considered wind speeds less than 15 m/s and frequencies ranging from 400-6400 Hz. Their model only predicts back-scattering strength, and therefore is only applicable to monostatic systems (Jensen et al., 2011). Furthermore, at low grazing angles (less than 30 degrees), this model over-predicts the scattering strength, Ss (McDaniel, 1993). This could be due to its empirical nature and the fact that it does not distignuish whether or not the reverberation is coming from the surface interface or the interaction with the sub-surface bubbles. That being said, this model is still deemed to be accurate when there are rough seas, high frequencies, and plumes as the main source of scattering strength (Etter, 2003). However, because it is not deemed accurate at low grazing angles, a better method was needed. Low grazing angles will occur at longer ranges. Given the simulator's current configuration, a target at 45 km results in a grazing angle with the sea surface that is very small. At this range the rays are almost horizontal because there are just about to turn back down towards the lower sound speed. (See the image on the far right in Figure 4-7). Under these circumstances a physics based model would provide the simulator with

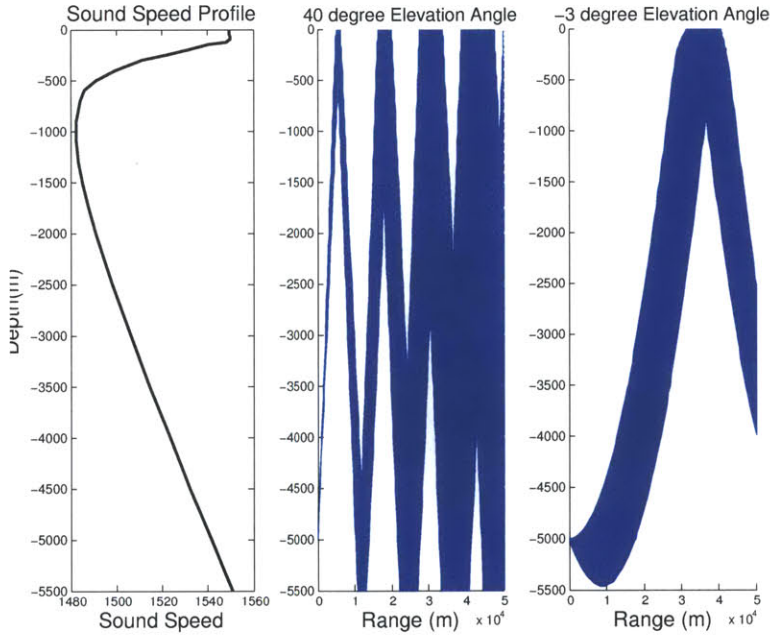


Figure 4-7: Ray Trace with Different Elevation Angles

better accuracy

Some improvements have been made to the Chapman-Harris Model. Ogden and Erskine constructed a new empirical algorithm that decouple effects from bubbles and the air-water interface. However, these improvements are only valid for frequencies less than 1 kHz (Ogden and Erskine, 1994). As mentioned earlier, the simulator should be as generic as possible, so only using frequencies less than 1 kHz would vastly limit its functionality.

Zhang further improved the Chapman-Harris model by extending it to 3-dimension so that it could be used in bistatic systems as well as monostatic ones. His improvements consisted of including forward scatter (instead of just back scatter) through the use of a Kirchhoff approximation as well as including a shadow factor. However, when looking through Zhang's paper it is evident that he is comparing the revised version of the Chapman-Harris model to a more ideal one. This ideal one is the model developed by the Naval Research Laboratory (Zhang, 2004). The Naval Research Laboratory (NRL) compared the Chapman-Harris, Ogden-Erskine and NRL methods with actual data and found that the residual error with each method was (for low frequencies) 4.1 dB error for Chapman-Harris, 3.1 dB error for Ogden-Erskine and only 2.2 dB for the NRL model. (Low frequencies were used because the Ogden-Erskine formula is only valid for low frequencies). An additional test for all frequencies was conducted in order to compare Chapman-Harris with NRL. For this test, NRL had 2.5 dB error and Chapman-Harris still had 4.1 dB error (Gauss et al., 2002). Since NRL consistency resulted in the least error when compared with actual data, our simulator used the NRL method as discussed below.

4.2.1 Method Developed by Naval Research Laboratory

The model developed by the Naval Research Laboratory in 2002 is semi-empirical in nature. Using this method we find Surface Scattering Strength (SSS) by incoherently adding the effects from the sonar's interaction with the surface interface as well as its interaction with the sub-surface bubbles (See Equation 4.3). The equations in this model are derived from data collected during various sea tests, including

deep sea tests conducted by Chapman and Harris, Chapman and Scott, Crowther, Ogden and Erskene, and others. Additional data was gathered by from the Space and Naval Warefare Systems Command's Critical Sea Test program (1988-1996) and Office of Naval Research's Littoral Warfare Advanced Development program (1996-present).

$$SSS = 10log_{10}(\sigma_{int}^{surf} + \sigma_{bub}); \tag{4.3}$$

Gauss et al. (2002) has already plotted different scenarios so that one may see the effects of grazing angle, frequency, and wind speed. Here, we will discuss the effects of changing each of these variables. Grazing angle, as one would expect, determines which source to is responsible for most of the reverberation. At low grazing angles, the scattering strength from sub-surface bubbles increases very quickly and then drops off when the angle is greater than 20 degrees or so. The scattering strength from the surface interaction, however, more or less increases linearly with the grazing angle. In this case, at lower grazing angles we have very low scattering strength. The combined effect of these two sources is that the scattering strength monotonically increases; however the increase no longer is linear due to the sharp increase at low elevation angles from the subsurface bubbles.

Any increase in frequency has a very small effect on σ_{int}. The scattering strength due to the surface interface slightly increases with frequency but only by a few dB when ranging from 10Hz to 10,000Hz. Frequency's effect on the scattering strength due to bubbles, however, is a little more complicated. The effect on bubbles is coupled with wind speed. At low wind speeds (5 m/s), for example, the bubbles only contribute if the frequency is greater than 1 kHz. At higher wind speeds, however, (e.g. 20 m/s) the scattering strength also increases with frequency up to a certain maximum. Once it hits this maximum (which is smaller at lower wind speeds), the scattering strength starts to decrease with increasing frequency.

In order for wind speed to have any effect on the scattering strength from bubbles, the wind speed needs to at least 5 m/s. After this point, the scattering

Variable	Definition
σ	Dimensionless parameter (scattering crossection/unit area)
f	Source Frequency
c	Sound Speed
c_0	Sound Speed at surface boundary
k_0	Wavenumber: $\frac{2\pi f}{c_0}$
$\lvert\mathbf{Q}_h\rvert$	Difference in horizontal wavevector magnitude
$W(\mathbf{Q}_h)$	2D roughness spectrum density
β	An "algebraic form" dependent on boundary conditions
$\boldsymbol{I}$	Spatial spectrum integral dependent on $\mathbf{Q}_h$ and Q_z
U	Wind speed 10 m above sea surface (m/s)
w_2	Spectral Strength
A_s and γ_2	Free parameters that contain frequency dependence

Table 4.1: Variables needed in developing NRL Semi-Empirical Surface Reverberation Model

strength from bubbles increases with increasing wind speed up until wind speed is about 17m/s, when it starts to plateau at its maximum scattering strength. Increasing the wind speed does not have a significant effect on the backscattering strength from the interface, however. Although it is monotonically increasing, there is a sharp increase in scattering strength before 5m/s. Once the wind speed increases above 5m/s, the backscattering strength appears to plateau, but it is still slightly increasing with increased wind speed. (Gauss et al., 2002)

Finding σ_{int}^{surf} is physics based while finding σ_{bub} is almost entirely empirical. Table 4.1 includes a list of variables and their definitions.

4.2.2 σ_{int}

In order to derive the effects of the interaction between the rough surface and air (equation 4.4), we will start with the big picture and then discuss each component

part.

$$\sigma_{int} = |\frac{\beta}{2}^2 |W(\mathbf{Q}_h) \tag{4.4}$$

Here β is rather simple to calculate because we can assume that the total field vanishes on the surface (Dirichlet boundary condition).

$$\beta = -4k_0^2 sin(\theta_{inc}) sin(\theta_{scat}) \tag{4.5}$$

$$\sigma_{int} = \frac{1}{8\pi} |\frac{\beta}{|\mathbf{Q}_h| Q_z}|^2 \cdot \boldsymbol{I} \tag{4.6}$$

The form we see in equation 4.4 is a numerical approximation of the equation's exact form, equation 4.6. The integral in this expression involves a zeroth order Bessel function and is rather difficult to evaluate because in order to do so, we would need to consider all possible incident angles. The approximations that were made in order to obtain equation 4.4 include a small slope approximation that puts to equations into the form of a series. If we take only the first term of this series (make a first order approximation - just like perturbation theory), we get the desired result. Of course, if we wanted to make the model more accurate we could include more terms from the sum. However, this would make the calculations more complicated than need be for this simulator.

$$W(\mathbf{Q}_h) = \frac{w_2}{(h_0 |\mathbf{Q}_h|)^{\gamma_2}} \tag{4.7}$$

$$|\mathbf{Q}_h| = k_0 \sqrt{cos^2\theta_{inc} + cos^2\theta_{scat} - 2cos\theta_{inc} cos\theta_{scat} cos\phi_{bi}} \tag{4.8}$$

$$w_2 = A_s U \tag{4.9}$$

At this point we have most of the information we need to calculate the reverberation effects due to the surface roughness. The difference in horizontal wavenumber magnitude can be found by applying equation 4.8. Then we need the spectral strength, or w_2. The NRL model assumes that the surface is isotropic and there-

e-folding depth d	**Wind Speed** U
$d = 0.557 - 0.117 \cdot U + 0.0109 \cdot U^2$	$U > 7.5m/s$
$d = -0.19509 + 0.06503 \cdot U$	$3 \leq U \leq 7.5m/s$
$d = 0$	$U < 3m/s$

Table 4.2: Farmer and Vagle's Method for Determining e-Folding Depth

fore there is no directionality in this term. Once we have these two terms, all we need is the 2D roughness spectral density in order to get the input from the surface interference (Gauss et al., 2002).

4.2.3 σ_{bub}

As discussed earlier, the bubbles cause high reverberation levels at low frequencies, low grazing angles and high wind speeds. The first step in finding the contributions of bubble cloud to scattering strength is to find the e-folding depth (depth by which the amplitude is decreased by a factor of e) of the bubble cloud. Farmer and Vagle developed a method of finding this depth, seen in table 4.2, from the known wind speed. NRL assumes that this method is true for all of their data because there were no actual measurements of this e-folding depth.

Physically, the effect of the bubbles is dependent on how fast this air-void fraction decreases. Another physically important characteristic of the bubble intensity is that the maximum intensity is found a few meters below the sea surface. As mentioned earlier, there is no acoustic field at the surface; therefore this maximum is due the constructive interference of the incident and scattered waves. NRL assumes that, within the bubble cloud, each bubble acts as a point source, completely uncorrelated to the other bubbles in the cloud. Finding an equation for σ_{bub} was found by curve fitting the data. The result is equation 4.12.

$$k_{v,i} = -k_0 sin(\theta_{inc}) \tag{4.10}$$

$$k_{v,s} = -k_0 sin(\theta_{scat}) \tag{4.11}$$

$$\sigma_{bub} = \frac{0.0019 d^{5.15} k_0^{-0.6} k_{v,i}^2 k_{v,s}^2 [6 + 3(k_{v,i}^2 + k_{v,s}^2) d^2 + (k_{v,i}^2 - k_{v,s}^2)^2 d^4]}{2(1 + k_{v,i}^2 d^2)(1 + k_{v,s}^2 d^2)[1 + (k_{v,i} - k_{v,s})^2 d^2][1 + (k_{v,i} + k_{v,s})^2 d^2]} \tag{4.12}$$

Because this method is almost purely empirical, there are a few problems in its implementation. First of all, Farmer and Vagle found that when the wind changes direction, it actually creates more bubbles because there are more waves. However, equation 4.12 does not take this into consideration. Also, as mentioned earlier, none of the NRL experiments measure the depth, d, directly. This depth can also change depending on season and location. Basically, the relationship between depth, d, and wind speed, U, needs to be strengthened. Another shortcoming lies in that the resonant frequencies of the bubbles and bubble clouds are not taken into consideration, at least not in a physically rigorous way. There also may be a correlation between the effect of bubbles and the temperature and salinity of the ocean.

Once we have the surface target strength, we can treat the surface as a number of separate targets. Our chosen method of modeling the targets is then applied to each of the surface areas. See the next section for details.

4.3 Modeling the Target

Modeling the effects from the target is done in two parts: simulating what happens between source to target and simulating what happens between target to receiver. First, two BELLHOP requests are sent. In the first request, the position of the vehicle is the source and the position of the target is the receiver. The angle and beam width are as defined by the actual sonar (given by pActiveSonarControl). This first bellhop request gives us the transmission loss from source to target, or TL_{ST} as well as the travel time for each eigenray, t_{st}. Figures 4-8 and 4-9 show two different cases. In both cases, the green dot represents the target. In the first ray trace we can

see that the target was not insonified, while in the second ray trace we can see that it was. In the first case, we will only expect to hear reverberation back.

In the second request the source and receiver are redefined., the position of the target is now the source and the average position of the array elements is the receiver. The position of the array elements comes from a module called uSimTowedArray which publishes the (x,y,z) location of each hydrophone. The simulator is capable of processing horizontal arrays, vertical arrays, and towed arrays (which could be neither horizontal nor vertical). Here the target is assumed to be a more or less omnidirectional source. This BELLHOP response will give us the transmission loss from target to receiver, TL_{TR}, the travel time, t_{tr}, as well as the received angle for each eigenray. Figure 4-10 shows the trajectory of each ray. The black dot is the position of the vehicle. Note the rays in red and black. These do not have any bottom bounces and will therefore have the highest intensity. In order to create the time series, it is

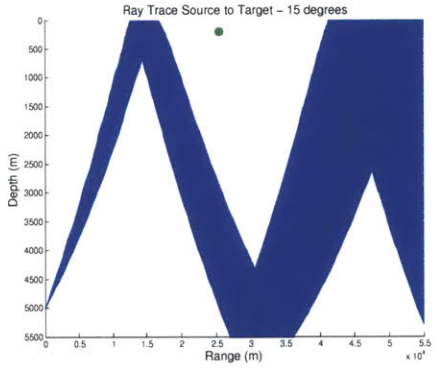


Figure 4-8: No Target Found

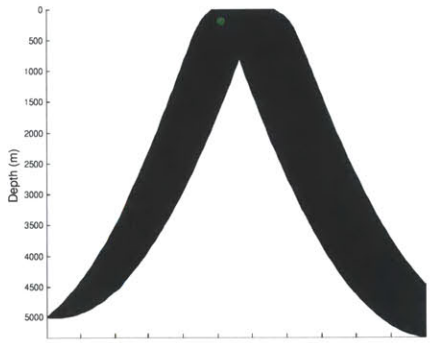


Figure 4-9: Target Found

assumed that the target response acts as plane waves near the hydrophones.

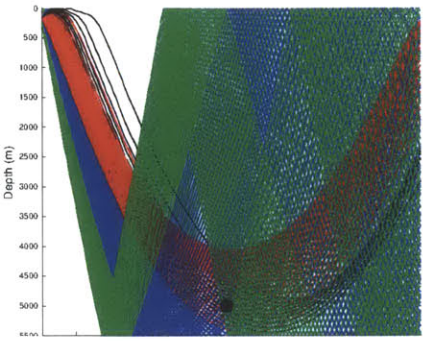


Figure 4-10: Ray Trace from Target to Receiver

4.3.1 Plane Wave Moveout over Hydrophones

As mentioned in the previous section, once we know what the target signal looks like at one of the receivers, we use plane wave expansion over the rest of the hydrophones. The first step is to determine the difference in travel time for an eigenray to each hydrophone. Because the "receiver" that BELLHOP used was in the average position of all other hydrophones we have to extrapolate out for both range and depth. Figure 4-11 demonstrates this.

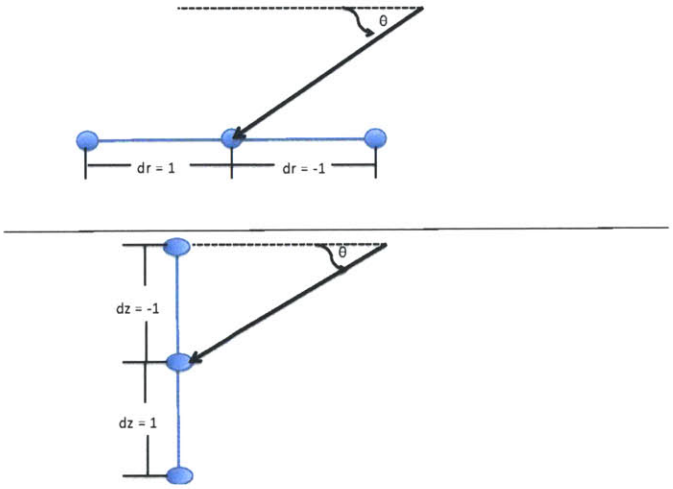


Figure 4-11: Top: Horizontal Array, Bottom: Vertical Array

In this figure, dr and dz are just the differences in range to the target and depth from the center hydrophone, while θ is the incoming grazing angle of the

eigenray. With this configuration we can define the arrival times at each hydrophone as: $newtime = time + (drcos\theta + dzsin\theta)\frac{1}{1500}$, where 1500 m/s is the sound speed. The propagation time between source and target is assumed to be the same for all hydrophones.

The next step is the determine the amplitude of the incoming wave. To do this we multiply the target strength (in pressure) by the pressure from the transmission loss form source to target and the pressure from the transmission loss from target to receiver. This comes from the sonar equation. Equation 4.13 assumes everything has units of decibels. In pressure, everything must be multiplied, not added (Jensen et al., 2011). The effects of reverberation and noise are added in later and the source level is taken care of by the source waveform.

$$Signal = SourceLevel - TL_{ST} - TL_{TR} - (Reverb + Noise) \tag{4.13}$$

Depending on where the source angle is we also include a different factor in order to account for the Hanning window. We multiply this amplitude factor by the Doppler shifted (see next section) source waveform and then insert the amplitude modified waveform into the correct temporal location in the time series for each hydrophone. Once the time series is created it is written as a binary file and passed along the processing chain. At this point a sonar processor analyzes the data in order to get time of arrival, direction of arrival and other information. Because all the sonar processor sees is the binary file, it has no way of knowing whether the files comes from uSimActiveSonar or from the output from hydrophones in an actual active sonar configuration.

4.3.2 Doppler Shift of a Moving Target

Correctly simulating Doppler shift of a moving target is especially important in an active sonar system because the source waveform frequency is known. Further processing of the signal heard by the hydrophones can tell us how fast the target is moving and whether it is moving towards or away from the vehicle.

$$\Omega_{mn} = \omega + (k_m + k_n)(v_{TS}cos\varphi_{TS} - v_S cos\varphi_S) \tag{4.14}$$

We can modify Jensen et al. (2011)'s monostatic normal mode Doppler equation, 4.14, in order to get it in an accessible form. k_m and k_n are the horizontal wave numbers. If we take this into consideration and divide both side by ω our final result for the Doppler factor will look like that of equation 4.15.

$$Doppler = 1 + \frac{1}{1500}(cos\theta_S + cos\theta_R)(v_{TS}cos\varphi_{TS} - v_S cos\varphi_S) \tag{4.15}$$

In this equation θ_S and θ_R represent the vertical launch angle at the source and the received angle at the hydrophone respectively. These values change with each arrival. For example if we were to have 2 arrivals from source to target and 10 arrivals from target to receiver then we would have 20 different Doppler shifts (corresponding to 20 different arrival times). $v_{TS}cos\varphi_{TS}$ represents the horizontal velocity of the target in the direction pointing away from the source while $v_S cos\varphi_S$ is the horizontal velocity of the source in the direction of the target.

In order to correctly implement this, we need to shift the frequency of the original source. For each arrival we create a new waveform (see next section) by multiplying both the original frequency and pulse duration by the Doppler factor. As mentioned in the previous section once we have this Doppler shifted waveform, we put it into its correct temporal locations in the time series.

4.4 Other Functionality

In addition to adding in scattering effects from the target and the sea surface, uSimActiveSonar also incorporates ambient noise. This is done by adding white noise at a user defined level to the time series.

4.4.1 Creating Waveforms and Replica Files

uSimActiveSonar can also create a source waveform based on changeable user defined parameters such as: center frequency, pulse duration, minimum frequency, maximum frequency, source sampling frequency, type, and source strength. The two types of signals that the simulator is capable of creating are constant waveform (CW) and linear frequency modulated (LFM) waveforms. uSimActiveSonar can also format each waveform to include a Hanning window. An additional "replica file" is created and written for the active sonar processor (pActiveSonarProcess) to use. This replica file represents the source waveform sampled at the sampling frequency of the hydrophones. The active sonar processor uses matched filtering and therefore needs the original source signal.

Chapter 5

Results and Conclusion

5.1 Results

5.1.1 General Overview

In order to visualize what the binary file containing the time series information looked like, we created another module, called pActiveSonarPlot. This module reads the binary file and plots various parameters including pressure and a crude beamformed result. Figure 5-1 shows the result of the situation seen in Figure 4-9. The top images depict the pressure at each hydrophone plotted on top of one another. The center image shows the beamformed time series. Here we can see that the target is coming at an angle of 5-10 degrees from the horizontal. The third image shows the average dB levels over all hydrophones. In all three, we can see the differences among the effects of the ambient noise, the surface reverberation and the target. As expected, the arrival time of the target falls somewhere within the time span of reverberation. This image was generated via a 16 element horizontal array, with a target at a distance of 30 km with a target strength of 30 dB and a source strength of 230 dB. When creating this image, only direct paths were taken into consideration.

In order to visualize the multipath (different ray paths including bottom and surface bounces) of the target, paths with bounces must be taken into consideration. Figure 5-2 was created using a similar set up as Figure 5-1 except a target strength

of about 60 dB was used with a much lower source strength. In addition, the time series has fewer samples so that we may concentrate on a specific section of it. The pulse duration was also reduced from 1 second to 0.1 seconds. From the Figure, we are able to see the multipath with two distinct arrivals about 1 second apart from one another.

pActiveSonarPlot also generates another plot that shows the history of the actions performed by the simulator. Figure 5-3 shows an example of this type of plot. Here, the source is sweeping the sea surface in order to find the target. The target is moving at about 4 m/s. A sweep is defined here as a monotonically decreasing change

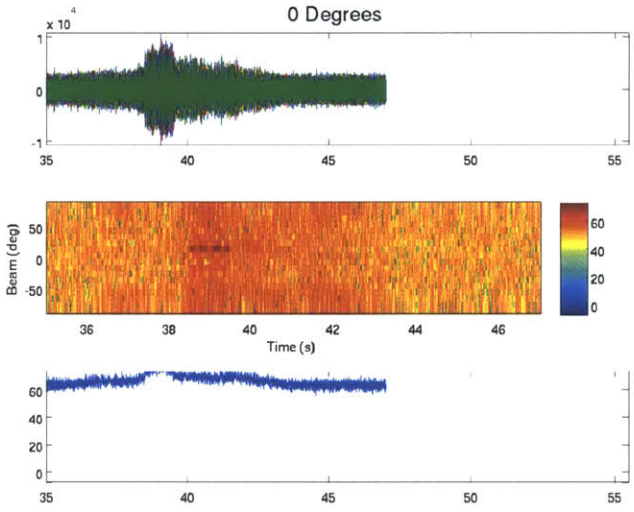


Figure 5-1: Target Found

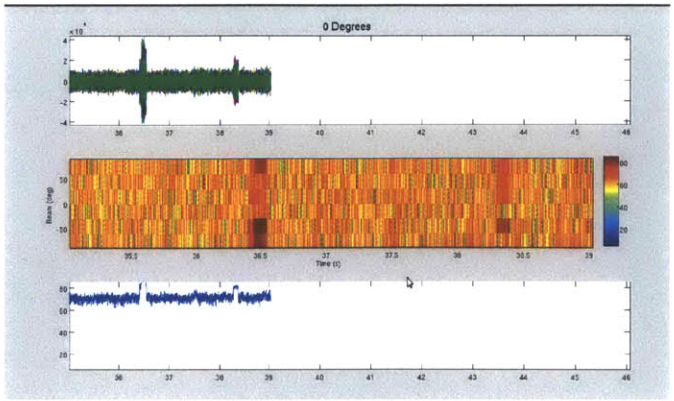


Figure 5-2: Multipath

in elevation angle (from 50 degrees to -1 degrees) followed by a monotonic increase in elevation angle (back to 50 degrees). We can see that when the sonar sweeps back in the direction of the target, the target signal arrives earlier. The first time we can see the target, its echo returns to the hydrophones after about 45 seconds. After a second full sweep we find the target again. Its echo returns at around 35 seconds, which clearly suggests that the target is moving closer.

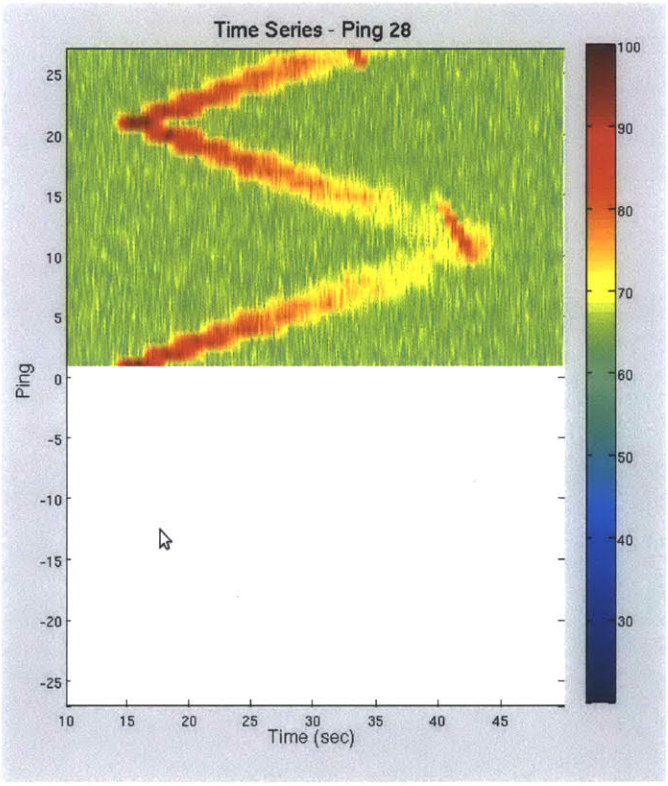


Figure 5-3: Sweep

The generated plot also keeps track of the ping number.

5.2 Verification

The previous figures show that the target, reverberation, and noise are all combined in the time series, written to a binary file, and then read and plotted by pActiveSonarPlot. However, we still need to verify that the amplitudes associated with the target, noise, and reverberation, plane wave move-out across vertical and

horizontal arrays, and Doppler shift are calculated correctly.

5.2.1 Amplitude Validation

In order to verify that the amplitude is correct in uSimActiveSonar, we must consult the sonar equation (Jensen et al., 2011):

$$RL = SL + TS + TL_{ST} + TL_{TR} \tag{5.1}$$

where RL is the received level, TS is the target strength TL_{ST} is the transmission loss from source to target and TL_{TR} is the transmission loss from target to receiver. Note that for a monostatic system $TL_{ST} = TL_{TR}$. Transmission loss is defined below:

$$TL = 10log_{10}(range^2) \tag{5.2}$$

Target and Noise Amplitude

In order to validate the target amplitude, we assume ambient noise of 40 dB and a very low sea state (wind speed of 0 m/s) so that the effects of surface reverberation do not interfere with the results. First, we validate that the target strength (TS) and the source level (SL) are correctly being implemented. The source waveform is 4 kHz CW with a duration of 1 second. With the target fixed at a range of 30 km, we are able to vary only SL and TS. See Table 5.1. Referring to equation 5.2, we can determine that the transmission loss for a target at 30 km is about 180 dB. Using the expected values for SL, TS, and TL in equation 5.1, the expected level at the receivers can be calculated.

Note that the user defined 40 dB ambient noise is a spectral noise (noise per Hz). In raw time series it will show up much too high (70 dB). However, if we look at Figures 5-5, 5-7, 5-9, and 5-13 we notice that the noise is a constant 40 dB across the entire spectrum as expected, regardless of source strength, target strength and target range. We also notice that, in the spectrograms, the received level of the target is

Source Strength	Target Strength	Expected RL (dB)	Result
230 dB	30 dB	80 dB	Figure 5-4
230 dB	40 dB	90 dB	Figure 5-6
240 dB	40 dB	100 dB	Figure 5-8

Table 5.1: Testing Target Amplitude - 30 km target

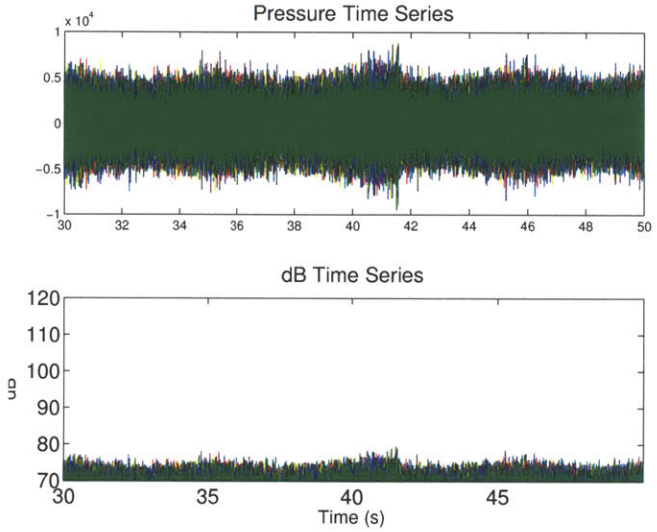


Figure 5-4: SL:230 dB, TS: 30dB,

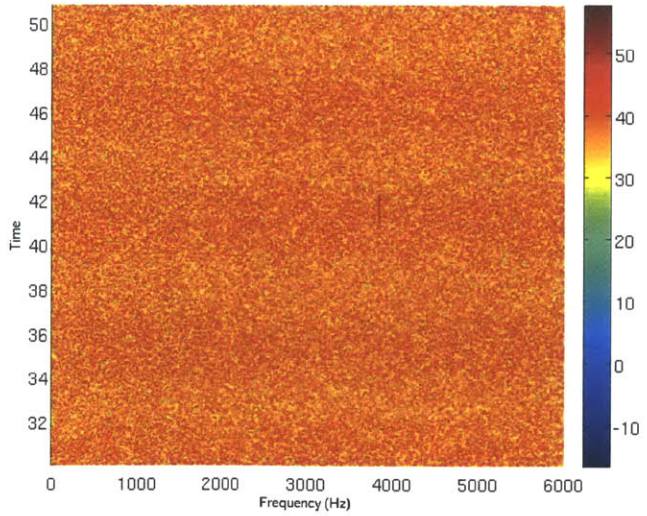


Figure 5-5: Spectrogram with SL:230 dB, TS: 30dB,

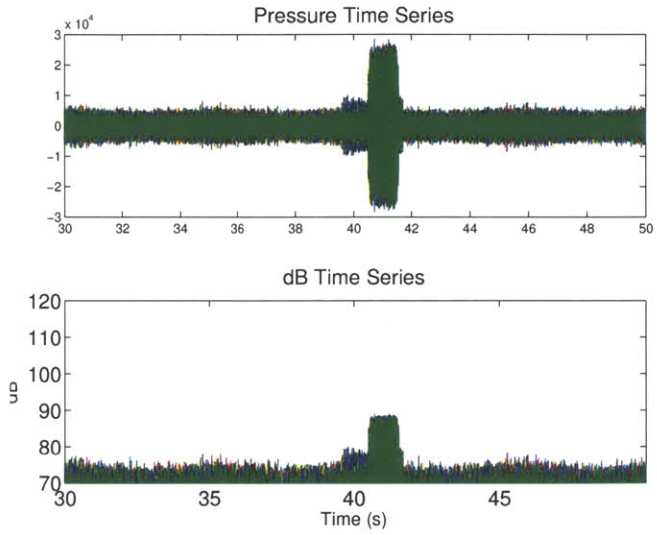


Figure 5-6: SL:230 dB, TS: 40dB,

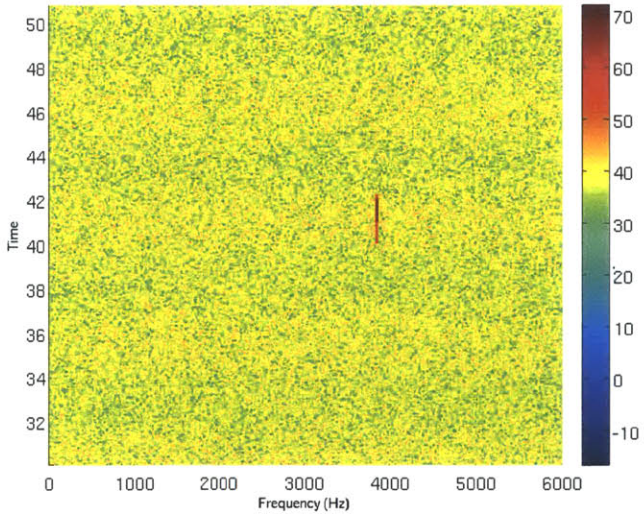


Figure 5-7: Spectrogram with SL:230 dB, TS: 40dB,

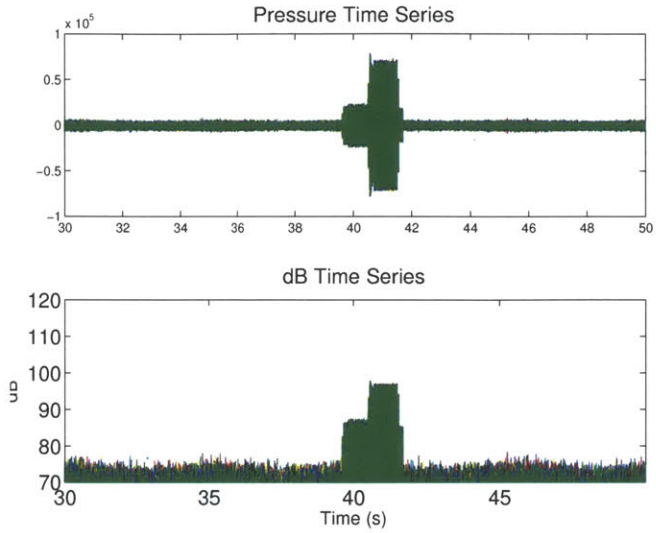


Figure 5-8: SL:240 dB, TS: 40dB,

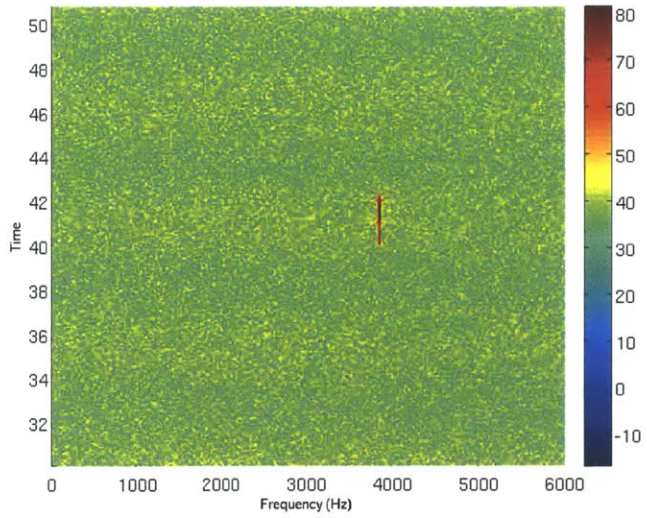


Figure 5-9: Spectrogram with SL:240 dB, TS: 40dB,

Target Range	Expected TL	Expected RL (dB)	Result
30 km	180 dB	100 dB	Figure 5-8
20 km	172 dB	108 dB	Figure 5-10
10 km	160 dB	120 dB	Figure 5-11
5 km	148 dB	132 dB	Figure 5-12

Table 5.2: Inputs and Expected Results for Testing Target Amplitude

much lower than expected. This is because, the one second long 4 kHz CW sound pulse is finite in length so energy will dissipate to all other frequencies. The target levels shown in Figures 5-4, 5-6, and 5-8 are consistent with the expected results.

To validate that the transmission loss is correct, TS and SL are kept at a constant value of 40 dB and 240 dB respectively. Table 5.2 shows the range of the target, calculated expected two-way TL (using equation 5.2) as well as the expected maximum target level and simulated result.

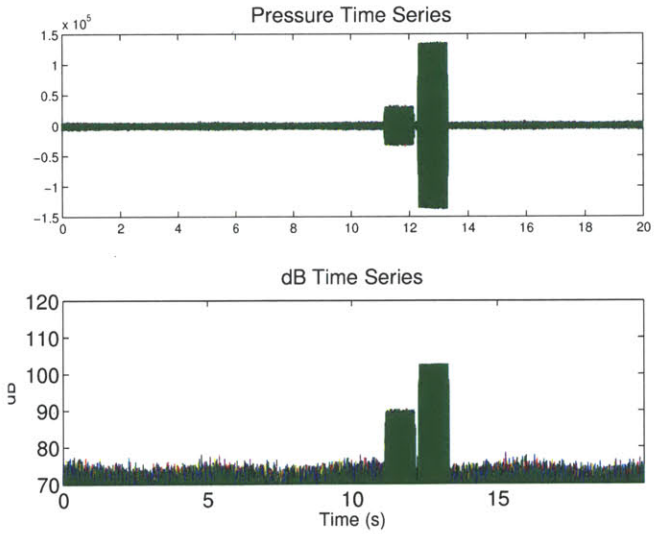


Figure 5-10: 20 km

In each of the previous examples the actual level outputted by the simulator is in agreement with the theoretical level one would expect from implementing the sonar equation. The spectral noise level is also consistent with the expected level regardless of changes to the source level, target level, or transmission loss from the

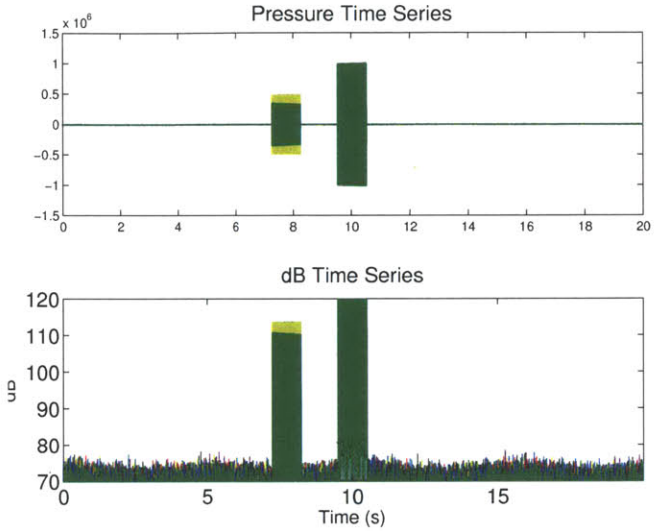


Figure 5-11: 10 km

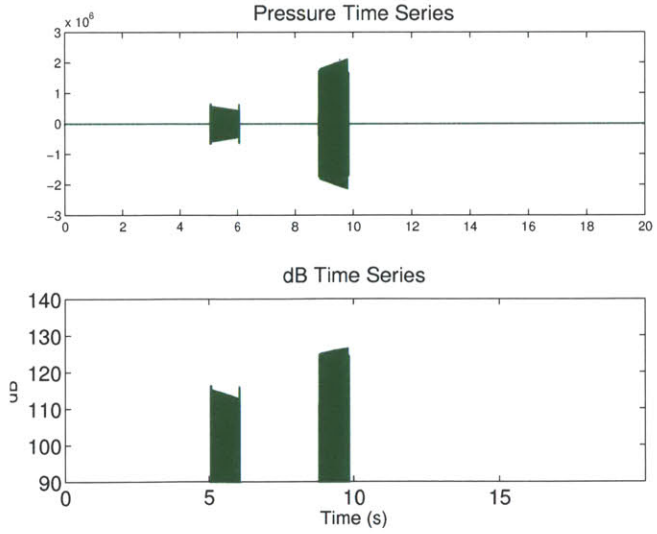


Figure 5-12: 5 km

target.

Surface Reverberation

In this section we validate that the surface scattering strength calculated by the simulator is consistent with the NRL model. Here, changes in angle and wind speed need to be taken into consideration because they are major contributing factors to reverberation. Table 5.3 displays theoretical scatterings strength (Gauss et al., 2002) and actual scattering strengths outputted by the simulator when wind speed is changed. In these examples, the elevation angle is kept constant (8 degrees). The incident grazing angles on the surface vary from 7 degrees to 12.Because the grazing angle has a range of values, the scattering strength (Ss) will also have a range of values. The results provided in Table 5.3 show conclusively that the reverberation levels are calculated correctly for each of the given wind speeds.

In order to test that the grazing angle is being implemented correctly by setting the wind speed to a constant 10 m/s and varying all of the other parameters. See Table 5.4.

Table 5.4 shows that the expected levels are consistent with the actual levels.

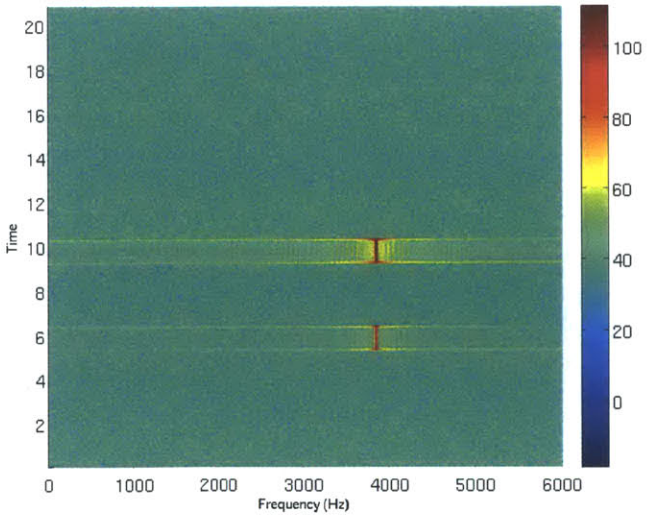


Figure 5-13: Spectrogram with Target at 5 km Range

Wind Speed	Expected Ss	Simulated Ss
5 m/s	-62 dB	-53 through -61 dB
10 m/s	-45 dB	-40 through -43 dB
15 m/s	-35 dB	-34 through -35 dB
20 m/s	-30 dB	-31 dB

Table 5.3: Expected and Actual Surface Scattering Strength at an 8 degree elevation angle.

Elevation Angle	Incoming grazing angles	Expected Ss	Actual Ss range
0 degrees	5 - 6 degrees	-46 dB	-44 through -46 dB
8 degrees	7- 12 degrees	-44 dB	-40 through -43 dB
17 degrees	13- 19 degrees	-43 dB	-39 through -40 dB
26 degrees	23- 29 degrees	-40 dB	-35 through -37 dB
40 degrees	37- 43 degrees	-30 dB	-28 through -31 dB

Table 5.4: Expected and Actual Surface Scattering Strength at an wind speeds of 10 m/s.

5.2.2 Validation of Horizontal Array

In order to test that the move out of the horizontal line array is being carried out correctly, we use a 16 element horizontal array with a spacing of 0.1875 m between each hydrophone, a Hanning windowed 4 kHz source, and a linear beamformer. The 0.1875 separation between hydrohpones satisfies the Nyquist rate ($\Delta z \leq \frac{\lambda}{2}$) in order to avoid grating lobes at broadside (Dyer). The AUV is simulated to be at (x, y, z) coordinates of (0,0,4500). The target is kept at a depth of 125 m and a range of 25 km.

The following table, Table 5.5, displays the actual location of the target, actual bearing of the target with respect to the horizontal array, and the figure number which corresponds to the output from the simulator.

Each of the following images follows the ray path shown in Figure 5-14 from source to target. Here we notice that the elevation angle (and thus the dominant incoming angle at the receiver) is not exactly zero. Horizontal arrays are sensitive

Actual Bearing	Figure	Spacing ¡ Nyquist Rage
0 degrees	Figure 5-15	Figure 5-21
10 degrees	Figure5-16	Figure 5-22
30 degrees	Figure 5-17	Figure 5-23
75 degree	Figure 5-18	Figure 5-24
-20 degree	Figure 5-19	Figure 5-25
-40 degree	Figure 5-20	Figure 5-26

Table 5.5: Actual Bearing and Simulator Output

to range if the bearing of the target is anything but zero. The move out across the hydrophones is proportional to $sin\phi cos\theta$ where θ is the vertical angle (determined by range) and ϕ is the horizontal angle (bearing). Here we notice that if bearing is zero degrees, then there is no contribution from the vertical angle. As the target moves closer to endfire (90 degrees to the array), not only will we have a loss in resolution, but we will also have a higher influence from the vertical angle. In the images displayed, we notice that the beampattern has it's main lobe at an angle slightly higher than the expected one. This is caused by the offset by the vertical angle.

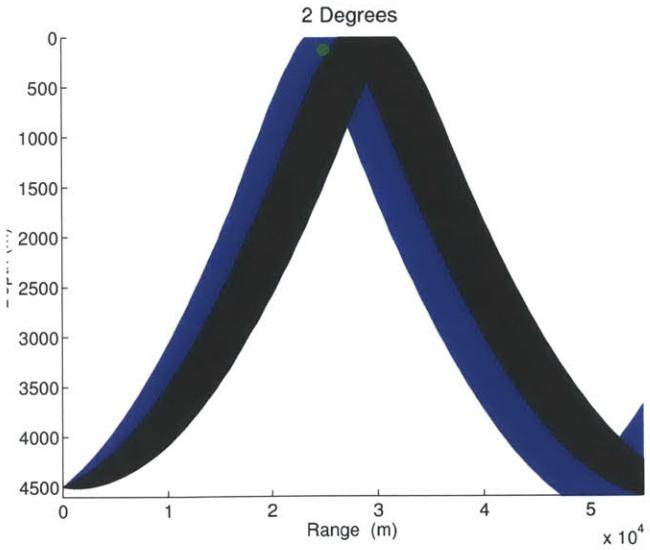


Figure 5-14: Ray Path Followed (from source to target) for a target at 25 km

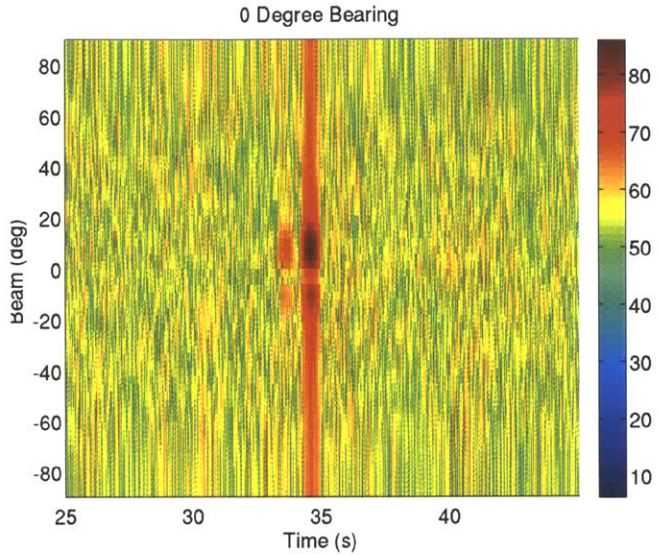


Figure 5-15: Target Location- x: 25 km y: 0 km

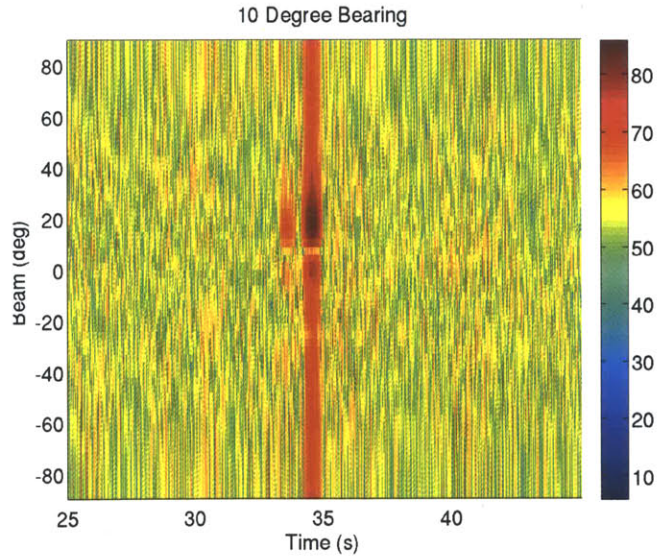


Figure 5-16: Target Location- x: 24.62 km y: 4.34 km

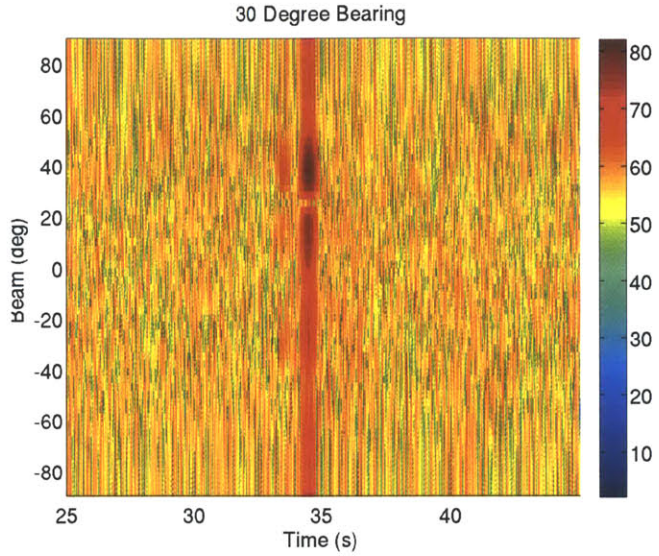


Figure 5-17: Target Location- x: 21.65 km y: 12.5 km

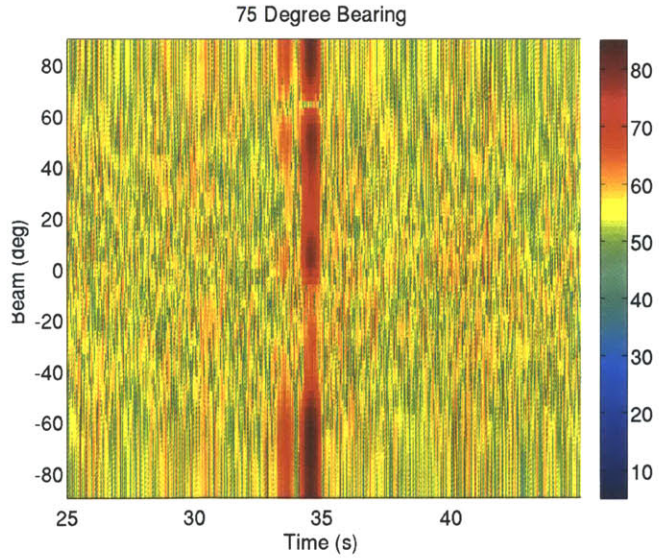


Figure 5-18: Target Location- x: 6.47 km y: 24.15 km

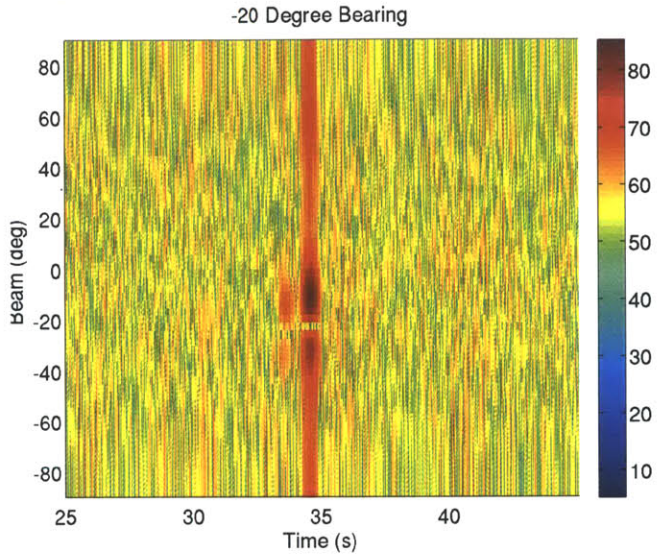


Figure 5-19: Target Location- x: 23.49 km y: -8.55 km

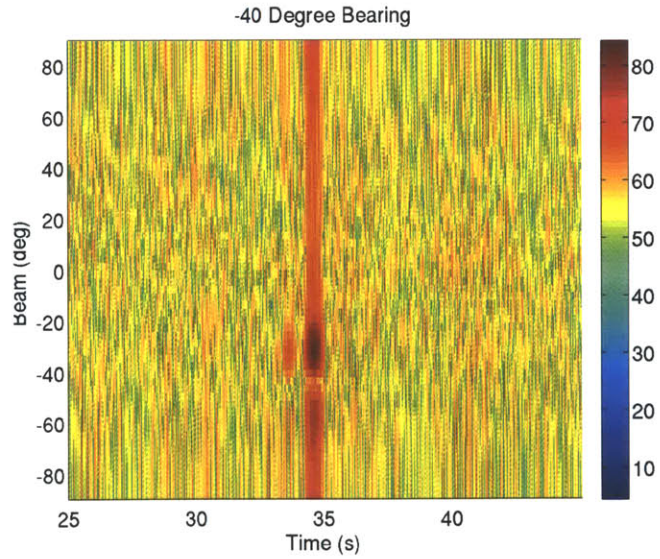


Figure 5-20: Target Location- x: 19.15 km y: -16.06 km

As seen in the following figures, the main lobe of the beampattern is in the expected direction. However, as we move out away from broadside (defined here as 0 degrees), the main lobe gets wider and we start to see grating lobes. This is to be expected because the relationship between the beam domain and the phone domain is a Fourier transform. In the beam domain, the beanpattern is related to $sin\theta$ where θ is the steering direction of the array. In the best case scenario (uniform line array) as we move away from boradside and towards endfire, we expect the resolution to go down by a factor of $\frac{\lambda}{L}cos\theta$, where λ is the wavelength of the source and L is the length of the array.

In order to get rid of these grating lobes, the distance between elements on the array must be less than the Nyquist rate. Figures 5-21, 5-22, 5-23, 5-24, 5-25, 5-26 display the results with an element seperation of 0.15 m. From these figures we can conclude that the horizontal move-out of the target arrivals are correct.

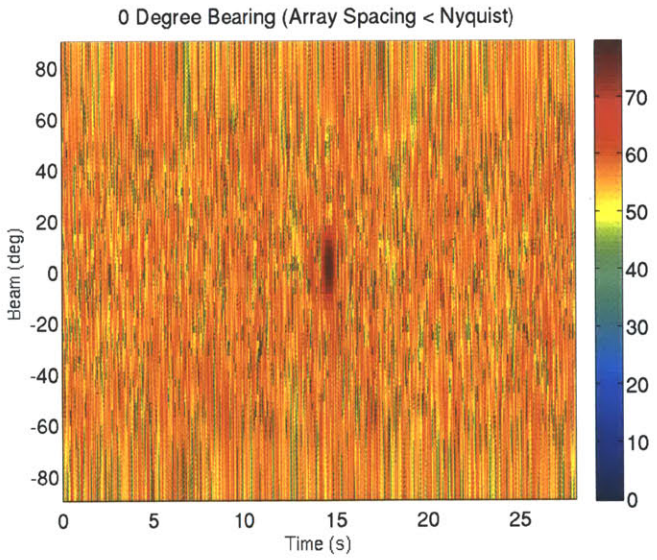


Figure 5-21: Target Location- x: 19.15 km y: -16.06 km

5.2.3 Validation of Vertical Line Array

In order to validate that the move-out across a vertical line array is carried out correctly, we simulate a vertical line array with 20 elements spaced 0.15 m apart.

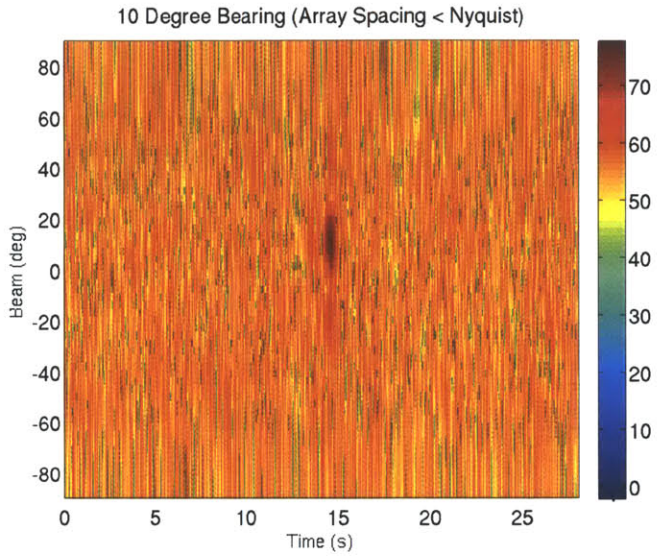


Figure 5-22: Target Location- x: 19.15 km y: -16.06 km

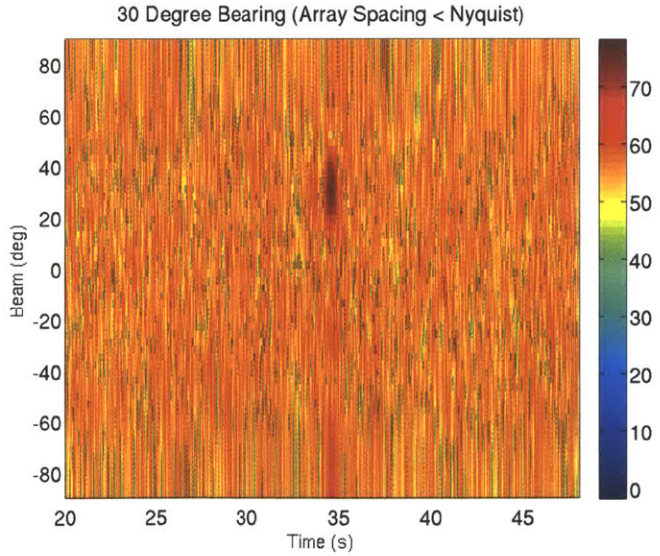


Figure 5-23: Target Location- x: 19.15 km y: -16.06 km

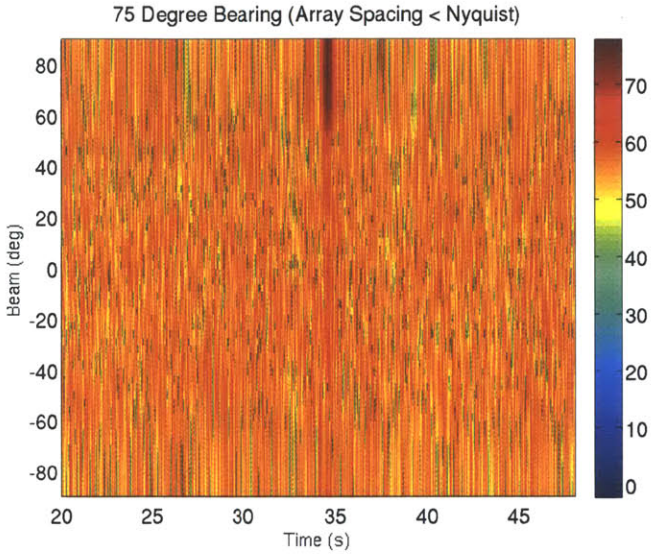


Figure 5-24: Target Location- x: 19.15 km y: -16.06 km

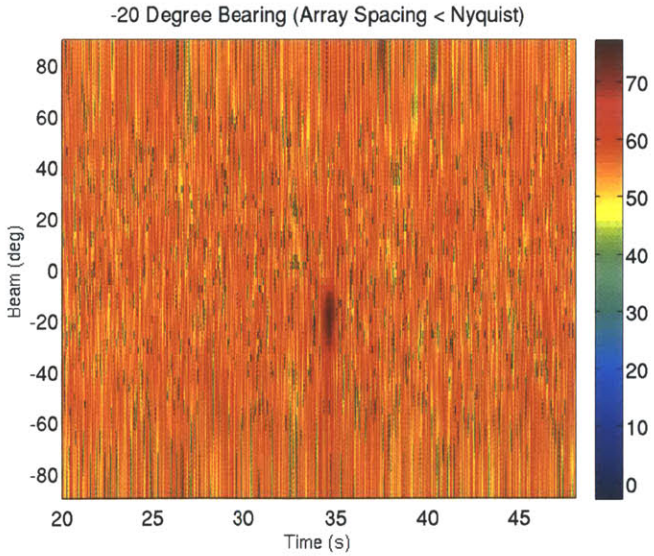


Figure 5-25: Target Location- x: 19.15 km y: -16.06 km

This element separation is chosen to be less than the Nyquist sampling rate so that we can bypass the grating lobe issue. The 4 kHz windowed source with a 6 degree beamwidth is again used. A vertical line array is insensitive to bearing and can only process range information. Therefore, for testing, bearing and depth are kept the same, while the target is moved from a range of 31 km to 5 km. The elevation angle of the source will change as the range of the target changes. We expect the majority of the sound heard back at the receiving array to be in the direction of the elevation angle (direct path). We also expect, however, that the target will seem as though it is coming from multiple directions due to bottom/surface bounces. Table 5.6 describes the input parameters, expected results and the figure displaying the actual result. Note that in the "Incoming Angles" section of this table, the negative angles indicate bottom bounce from the target to the hydrophone array.

The ray path from source to target and the beamformed time series at the vertical hydrophone array are shown in the following figures. As the target gets closer to the array, we notice that our directional resolution gets worse because as in the previous section as we steer the array towards endfire, we expect a loss in resolution. In all cases, the main lobe (maximum) is found, as expected, in the direction of the elevation angle. However, due to side lobes and bottom bounces, peaks are seen in

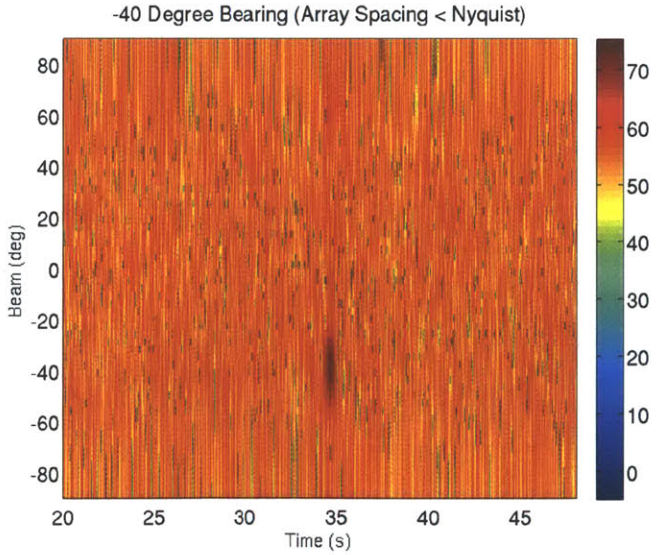


Figure 5-26: Target Location- x: 19.15 km y: -16.06 km

Target Range	Elevation Angle	Incoming Angles	Figure
31 km	-1 degrees	All 0 degree	Figure 5-28
22 km	5 degrees	6, 5, -7 degrees	Figure 5-30
17 km	9 degrees	11, 10, -11 degrees	Figure 5-32
9 km	23 degrees	25, 24, -25 degrees	Figure 5-34
7 km	30 degrees	32, 30, -32 degrees	Figure 5-36
5 km	40 degrees	41, 40, -42 degrees	Figure 5-38

Table 5.6: Actual Bearing and Simulator Output

other directions as well. Note that in addition to the direct path, the direction that has the largest contribution to the received signal is the negative of the direct path. This is due to bottom bounce.

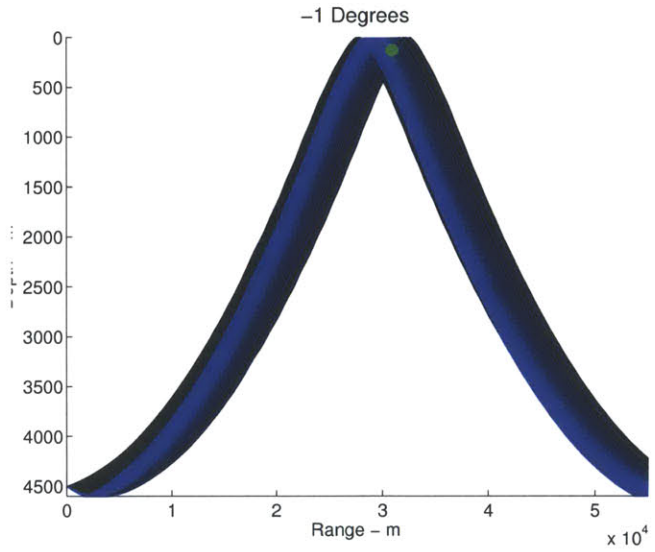


Figure 5-27: Target at 31 km range

5.2.4 Validate Multi-Target Capability

In order to validate that the simulator has the capability to handle more than one target, we start a simulation assuming one target at a range of 25 km with a target strength of 30 dB. Once there has been one ping, another target is simulated at a range of 30 km with a target strength of 40 dB. Figure 5-39 displays the result

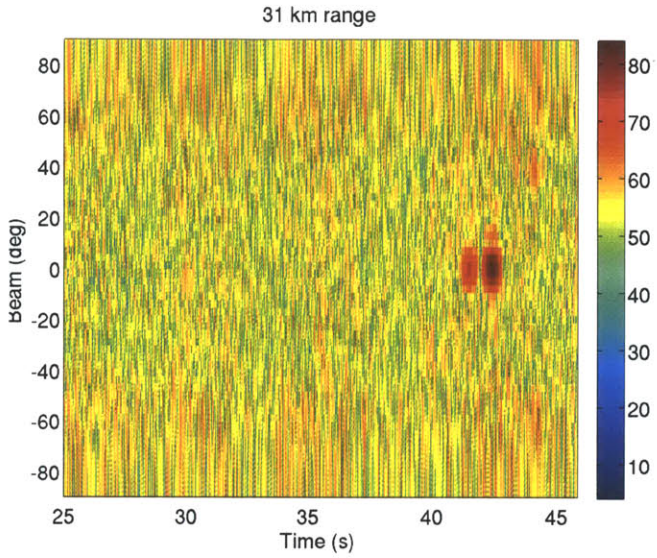


Figure 5-28: -1 degree Elevation Angle

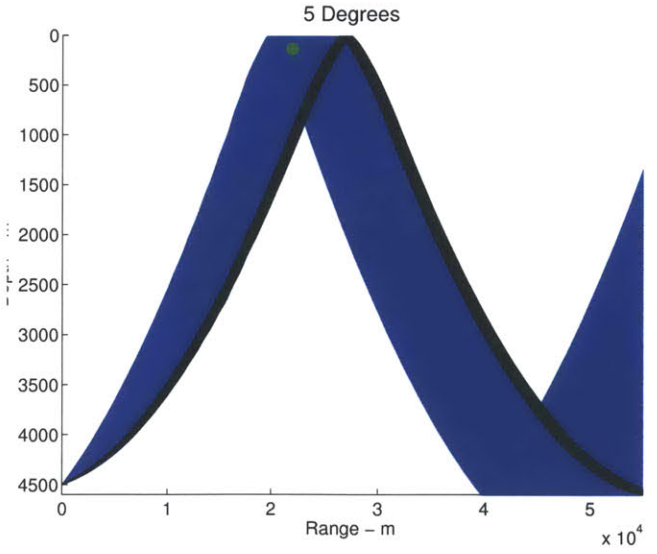


Figure 5-29: Target at 22 km range

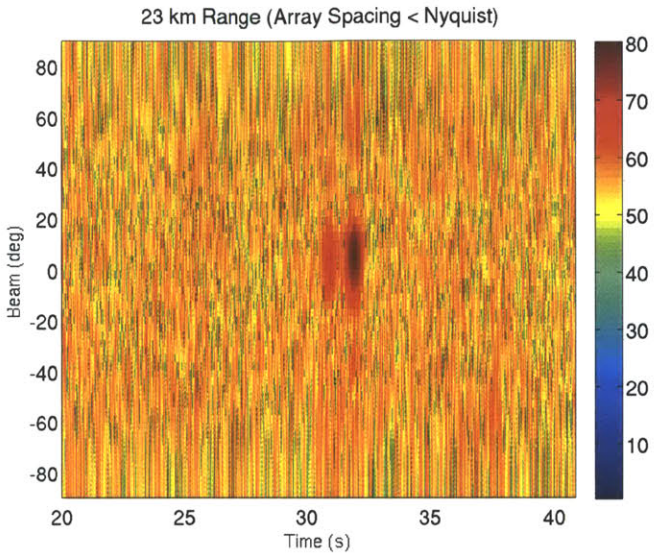


Figure 5-30: 5 degree Elevation Angle

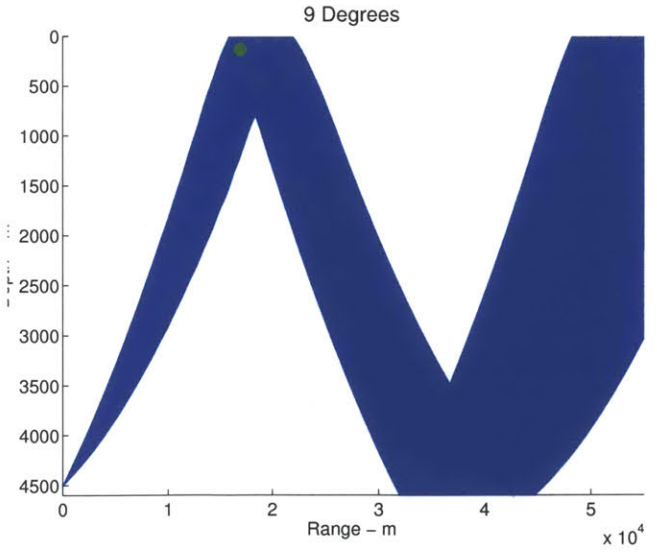


Figure 5-31: Target at 17 km range

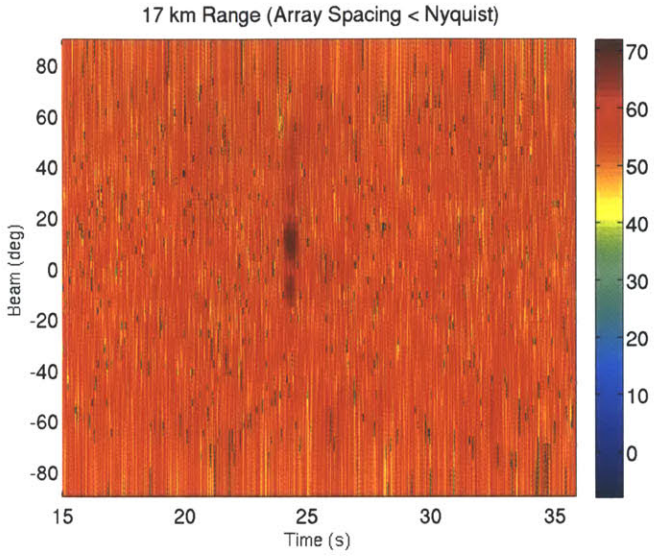


Figure 5-32: 9 degree Elevation Angle

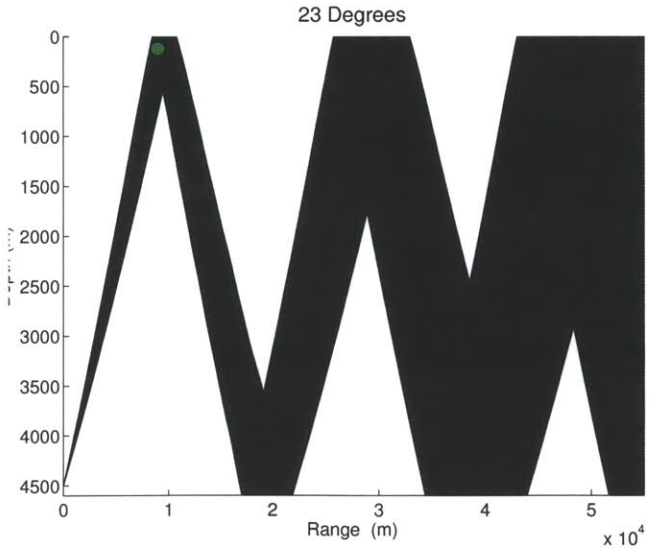


Figure 5-33: Target at 9 km range

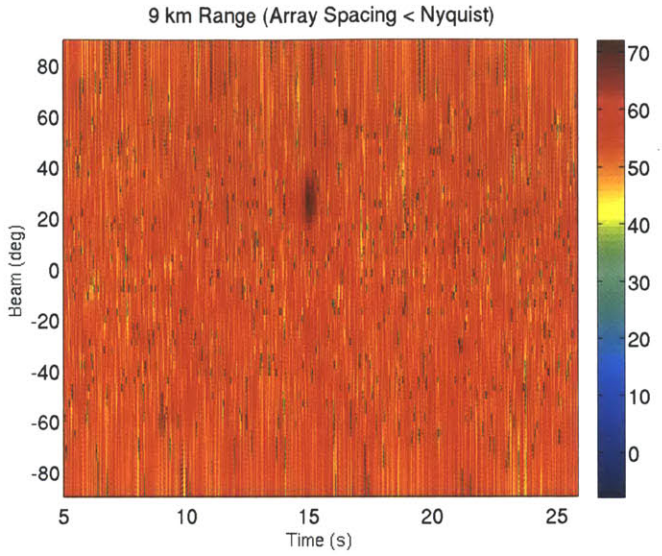


Figure 5-34: 23 degree Elevation Angle

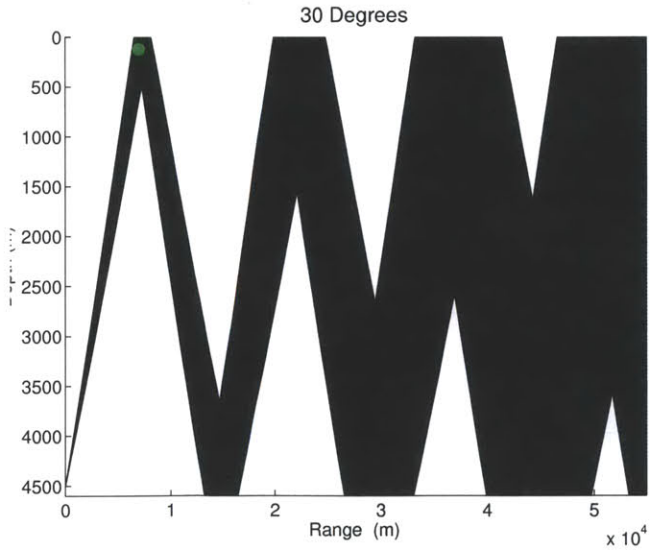


Figure 5-35: Target at 7 km range

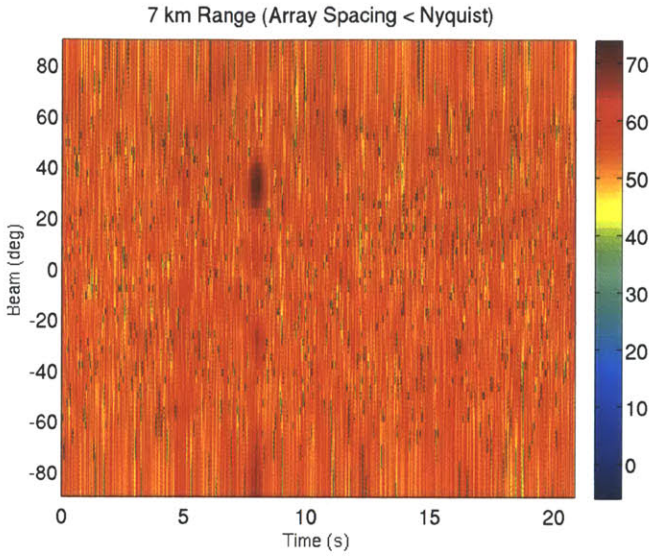


Figure 5-36: 30 degree Elevation Angle

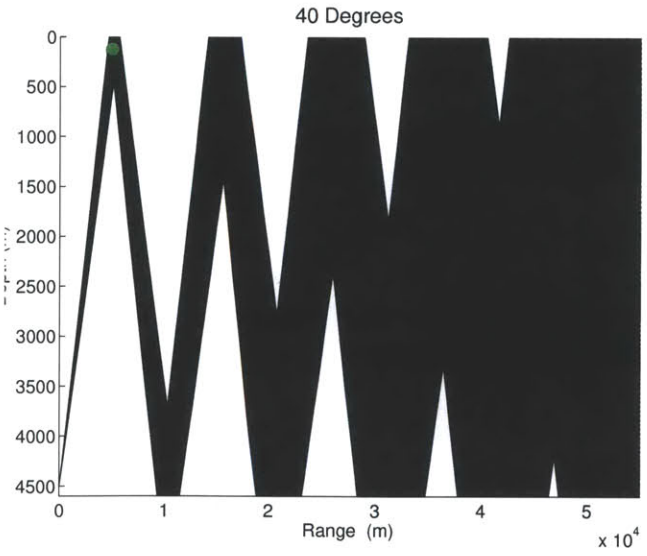


Figure 5-37: Target at 5 km range

of just one target and Figure 5-40 displays the result of simulating a second target once the simulator is already running.

In the second timeseries we can clearly see two targets.

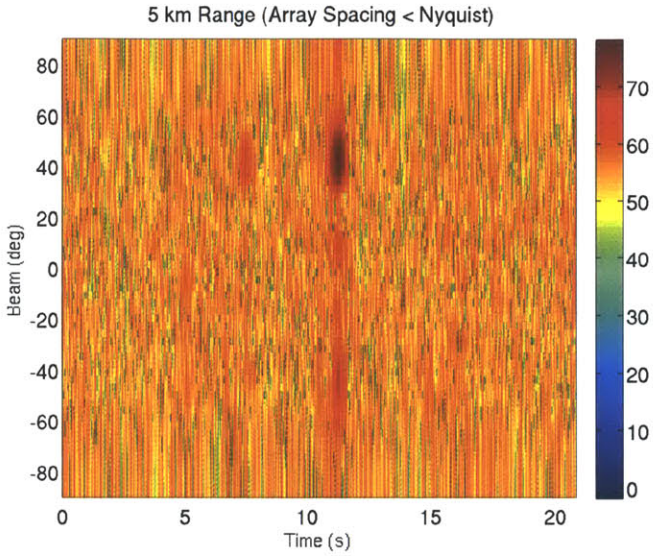


Figure 5-38: 40 degree Elevation Angle

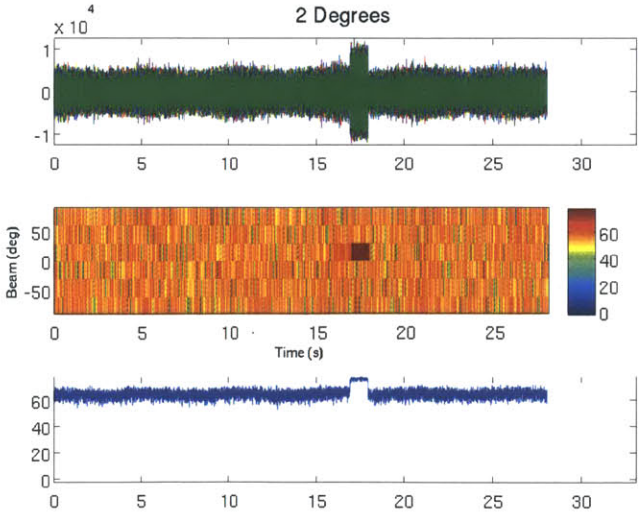


Figure 5-39: 1 Simulated Target

5.2.5 Validating Frequency Content

Frequency content in the simulator is important when it comes to creating the source waveform and simulating the Doppler shift of a moving target.

As mentioned earlier, the simulator is capable of creating two types of waveforms: CW and LFM. Input parameters such as pulse width, center frequency (when

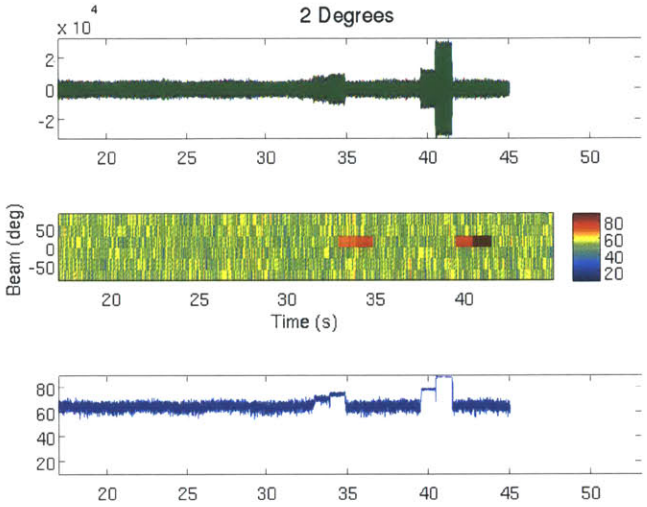


Figure 5-40: 2 Simulated Targets

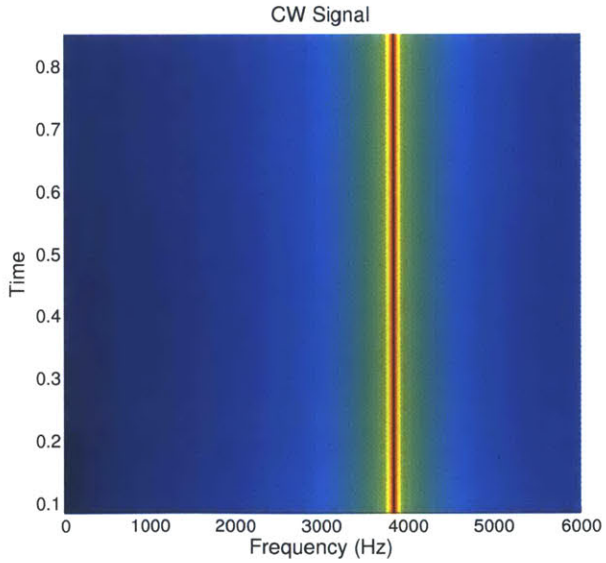


Figure 5-41: CW source waveform spectogram

dealing with CW), maximum frequency and minimum frequency (when dealing with LFM) are user definable. Figure 5-41 displays a spectrogram of a 1 second long CW signal at 4 kHz. Figure 5-42 displays a spectrogram of a 1 second long LFM signal with a minimum frequency of 3950 Hz and a maximum frequency of 4050 Hz. We can see that the CW signal remains at its constant frequency while the LFM has the expected, linear shift in frequency.

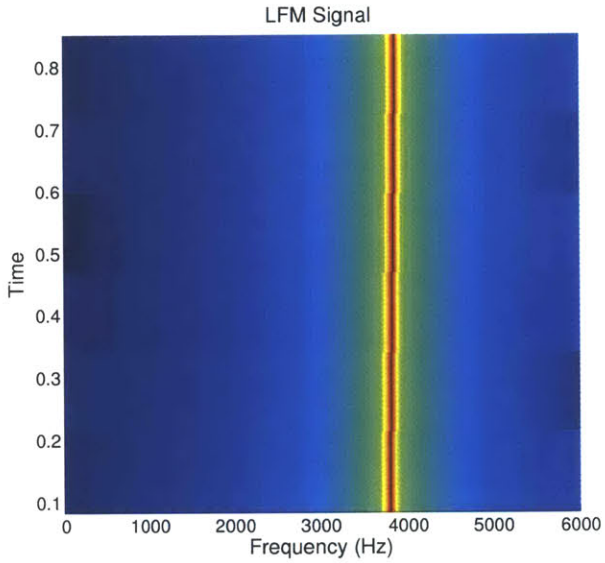


Figure 5-42: LFM source waveform spectrogram

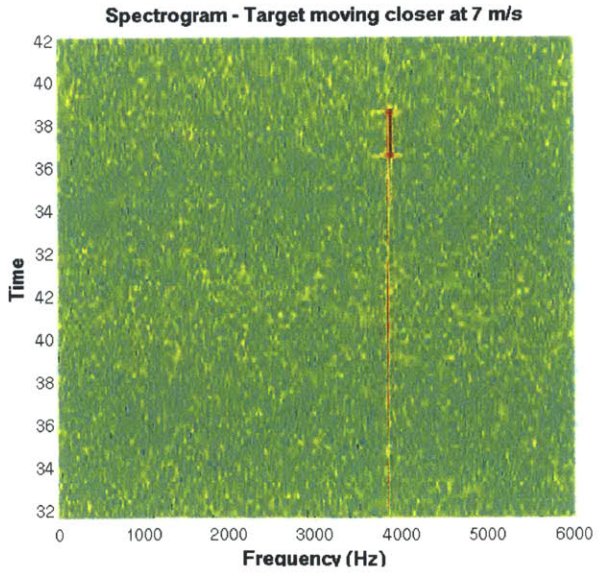


Figure 5-43: Spectrogram of Target Moving at 7 m/s Closer to the Vehicle

Doppler shift can also be visualized through the use of spectrograms. Figure 5-43 is a spectrogram that demonstrates the simulator's ability to deal with Doppler using a surface target at a range of 30 km moving closer to the vehicle at a rate of 7 m/s. A 4000Hz CW waveform and a launch angle of 0 degrees was used the generate this plot. As expected, we hear reverberation at a constant frequency the entire time we are recording. Here the target is moving closer to the vehicle and the Doppler shifted target frequency is higher than the original frequency (as compared to the surface reverberation).

Figure 5-44 shows the same configuration as Figure 5-43 but with the target moving away from the vehicle at 7 m/s. As expected, the target has a lower frequency than that of the reverberation.

5.3 Conclusion and Future Improvements

Currently, the new simulator has been implemented and the processing chain is being tested.

uSimActiveSonar has made it possible to test a sonar processor, target tracker, and beamformer along with other modules. Each of these secondary modules depends

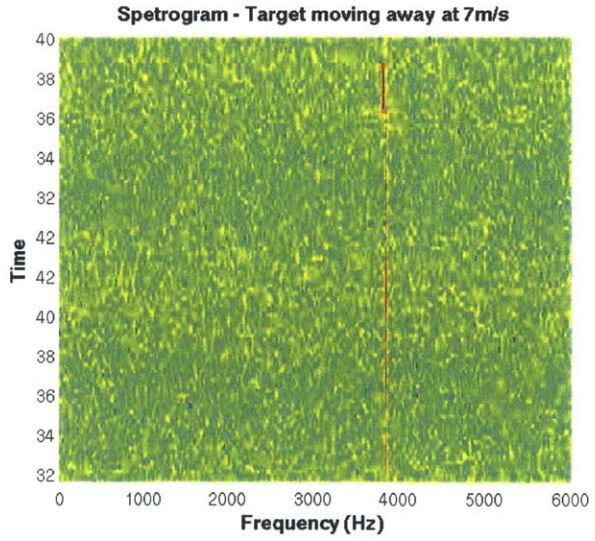


Figure 5-44: Spectrogram of Target Moving at 7 m/s Away from the Vehicle

on a different aspect of the simulator. Among others, these aspects include accurate noise (reverberation), accurate Doppler, and accurate move-out over the hydrophones.

One improvement to this simulator would be to augment it to have bistatic capabilities. The NRL method for surface reverberation is fully bistatic and therefore this is within the realm of possibilities. However, challenges may arise including bottom reverberation. Depending on vehicle configuration, the time it takes for bottom reverberation to reach the receiver hydrophones may be close to the time at which we expect to hear the target signal.

Such advanced simulation capabilities are very powerful tools. Anyone who has been on a research vessel will tell you that ship time is incredibly short and valuable. Often, there is only one chance to run a mission and if something is going to go wrong, it is bound to go wrong while you are on the actual vessel. Simple problems such as a glitch in someone's code should be solved before an experiment. The ability to simulate accurately makes it possible to reduce the number of these problems that one may run into while out at sea.

Bibliography

M. Benjamin and H. Schmidt, *2.S998 - Unmanned Marine Autonomy, Sensing and Communications- Course Notes*, 2012.

R. Chapman and J. Harris, "Surface backscattering strengths measured with explosive sound sources," *JASA*, vol. 34, no. 10, pp. 1592–1597, 1962.

P. I. Dyer, *Fundamentals and Applications of Underwater Sound*, Massachusetts Institute of Technology.

P. C. Etter, *Underwater Acoustic Modelling and Simulation*, 3rd ed. Spon Press, 2003.

G. Frisk, *Ocean and Seabed Acoustics: A Theory of Wave Propagation*, 1st ed., M. P. Rottino, Ed. Prentice Hall, 1994.

R. C. Gauss, R. F. Gragg, D. Wurmser, and J. M. Fialkowski, "Broadband models for predicting bottom, surface, and volume scattering strengths," *Naval Research Laboratory*, 2002.

F. B. Jensen, W. A. Kuperman, M. B. Porter, and H. Schmidt, *Computational Ocean Acoustics*, 2nd ed., W. M. Hartmann, Ed. Springer Science+Business Media, 2011.

J. Lynch, *2.682 - Acoustical Oceaongraphy - Course Notes*, 2012.

J.-P. Marage and Y. Mori, *Sonar and and Underwater Acoustics*. Wiley, 2010.

S. T. McDaniel, "Sea surface reverberation: A review," *JASA*, vol. 94, no. 4, pp. 1905–1922, 1993.

P. M. Newman, *MOOS: Mission Oriented Operating Suite*, Massachusetts Institute of Technology.

P. M. Ogden and F. T. Erskine, "Surface scattering measurements using broadband explosive charges in the critical sea test experiments," *JASA*, vol. 95, no. 2, pp. 746–751, 1994.

A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 3rd ed. Prentice Hall, 2010.

M. Porter and B. H.P, "Gaussing beam tracing for computing ocean acoustic fields," *JASA*, vol. 82, no. 4, pp. 1349–1359, 1987.

M. Porter, *The BELLHOP Manual and Users Guide: PRELIMINARY DRAFT*, Heat, Light, and Sound Research, Inc, 2011.

T. Schneider, *iBellhop lib-bellhop QuickStart*, Laboratory for Autonomous Marine Sensing Systems at MIT.

R. J. Urick, *Principles of Underwater Sound*, 3rd ed., D. Heiberg and J. Davis, Eds. McGraw-Hill, 1983.

Z. Y. Zhang, "Modeling bistatic surface scattering strength including a forward scattering lobe with shadowing effects," *Proceedings of ACOUSTICS*, pp. 171–174, 2004.