NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

FORWARD-LOOKING SONAR SIMULATION MODEL FOR ROBOTIC APPLICATIONS

by

Andreina Rascon

September 2020

Thesis Advisor: Brian S. Bingham
Co-Advisor: Derek Olson

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2020	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE
FORWARD-LOOKING SONAR SIMULATION MODEL FOR ROBOTIC APPLICATIONS

5. FUNDING NUMBERS

6. AUTHOR(S) Andreina Rascon

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

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

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

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

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

Underwater simulators are less common due to the complexity of underwater acoustics. Simulation is an effective tool for rapid testing of autonomous vehicles and complements the test and evaluation process. The goal of this thesis is to present a computationally efficient forward-looking sonar simulation model for robotic applications. A model for a single sonar beam is developed using a point-scattering model, applying both Fourier synthesis and a correction for beam forming. The single sonar beams are concatenated to simulate a forward-looking sonar system field of view. The result is a sonar simulation model that can be used in the established ROS Gazebo robotic framework as a tool for effective testing of autonomous underwater vehicles. Future improvements in the acoustics of the sonar model include the addition of reverberation, multi-path propagation, and interference.

14. SUBJECT TERMS forward-looking sonar, underwater simulation environment, sonar model, acoustics, autonomous vehicles	**15. NUMBER OF PAGES** 65
	16. PRICE CODE

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

NSN 7540-01-280-5500

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

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

FORWARD-LOOKING SONAR SIMULATION MODEL FOR ROBOTIC APPLICATIONS

Andreina Rascon
Lieutenant, United States Navy
BS, University of Notre Dame, 2013

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2020

Approved by: Brian S. Bingham
Advisor

Derek Olson
Co-Advisor

Garth V. Hobson
Chair, Department of Mechanical and Aerospace Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Underwater simulators are less common due to the complexity of underwater acoustics. Simulation is an effective tool for rapid testing of autonomous vehicles and complements the test and evaluation process. The goal of this thesis is to present a computationally efficient forward-looking sonar simulation model for robotic applications. A model for a single sonar beam is developed using a point-scattering model, applying both Fourier synthesis and a correction for beam forming. The single sonar beams are concatenated to simulate a forward-looking sonar system field of view. The result is a sonar simulation model that can be used in the established ROS Gazebo robotic framework as a tool for effective testing of autonomous underwater vehicles. Future improvements in the acoustics of the sonar model include the addition of reverberation, multi-path propagation, and interference.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

1D	one-dimensional
2D	two-dimensional
3D	three-dimensional
AUV	autonomous underwater vehicle
dB	decibel
EL	echo-level
FLS	forward-looking sonar
FOV	field of view
GPU	graphics processing unit
MBS	multi-beam sonar
MSIS	mechanical-scanning imaging sonar
ROS	Robot Operating System
SB	single beam
SL	source level
SONAR	sound navigation and ranging
TL	transmission loss
TS	target strength
USN	United States Navy
UUV	unmanned underwater vehicle
XLUUV	extra-large unmanned underwater vehicle

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

My sincerest gratitude goes out to all my family and friends, whose support through this program helped me stay positively motivated. To Brian and Derek, I cannot thank you enough for your dedication and incredible patience when helping me with this thesis. And to my dearest cats, Esme and Ares, because with your constant companionship, I was able to survive the work through the pandemic.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. MOTIVATION

The United States Navy (USN) recognizes that it requires a widely distributed fleet architecture, and the addition of unmanned underwater vehicles (UUVs) are the force multipliers necessary to achieve that goal. The USN requested $579.9 million for FY2021 for the acquisition of various types of unmanned vehicles. The UUV outline in Figure 1 illustrates the categories of unmanned vehicles the USN is interested in procuring in the near and far future [1]. The request also includes funding for research and development of enabling technologies which are identified from [1] as the following:

- endurance
- autonomy and precision navigation
- command, control, and communications
- payloads and sensors
- platform integration

The technology developed in these areas are significant for air and surface unmanned vehicle operations. However, in the underwater operating environment, the complexity of the sea still poses difficulties in effective communication and navigation which makes it difficult to achieve full autonomy in UUVs. The risk that unreliable communications or navigation posed by unmanned vehicles continues to be an issue; therefore, the investment in research and development addressing the listed enabling technologies for UUVs is necessary.

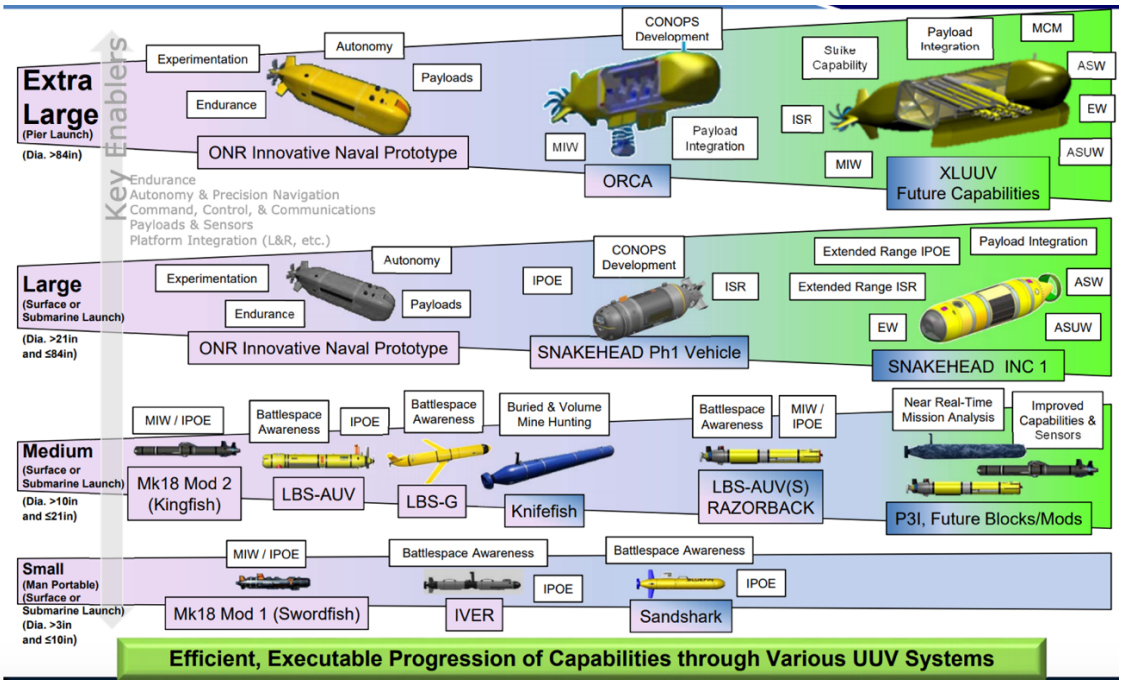

Figure 1. USN UUV Systems Vision. Adapted from [1].

In-situ or field testing is required to validate a new technology, control algorithm, or proof-of-concept. Acquisition and implementation of the new technology depends on the execution of the field testing. Difficult environments, funding, or unresolvable risks sometimes cause delays in testing, creating long gaps of time between design concept to full acceptance of the project. A promising addition for test and evaluation (T&E) programs is simulation. Simulation environments are useful tools for testing autonomous vehicles. They provide a cost-effective capability to provide rapid feedback on system updates, proof-of-concepts, etc. Simulation is not a substitute for field testing, but rather envisioned to complement the T&E of autonomous systems. A notable example of a program that could have benefitted from simulation is ExoMars Lander, which crashed on approach and cost approximately $350 million. The vehicle was recreated in simulation, and upon demonstration, the simulated vehicle also crashed in similar circumstances [2].

A study done by [2] evaluates the effectiveness of simulation capability in detecting bugs early in robotic systems. The empirical study focuses on bugs that were fixed in the open source ArduPilot system. They provide a quantitative and qualitative evaluation

characterization on bug detection, delineating how to identify bugs early and prior to field testing [2]. These bugs were defined as system faults that can cause a system to crash or behave unexpectedly. Their results show a “majority of bugs can be reproduced using software-in-the-loop simulation” and that only a few of the bugs are dependent on environmental inputs. This is contrary to the belief that simulation does not provide the fidelity required for identifying faults within a system. Complete representation of the environment in simulation is not necessary for faults that do not require environmental data inputs. While the best simulation models cannot capture real world physical models in its entirety, simulation can still benefit the T&E process in identifying faults that are not dependent on environmental inputs. Field testing will continue to be a critical step in quality assurance but including simulation can increase the probability of a successful field test.

Capturing the physics of the underwater environment is computationally expensive and it must be noted that simulation for underwater environments will not provide a substitute for a real-world environment. The complexities of the underwater environment have resulted in few simulators that have the capability of simulating an underwater environment, and even less so for underwater sensors. Sensors typically used for perception include optical cameras, lidar systems, doppler velocity logs, and sonar systems. Turbidity limits the range and effectiveness of optical cameras therefore limiting the capability for UUV sensing and navigation. Acoustic imaging for underwater perception is hence emphasized as essential for UUV perception and operations. The simulation of active sonar would therefore greatly benefit development and testing of UUVs and autonomous operations.

The motivation for this thesis is to provide the underwater simulation community a sonar sensor model that can be used in research and development efforts for UUVs. With few underwater simulators and even fewer underwater sensors simulators, the efforts of this thesis aim to fill this research gap. Assuring a capable active sonar sensor simulation supports advancements in autonomous vehicles and operations while minimizing expensive field testing. In the USN efforts to procure UUVs to create a more

distributed fleet, simulation tools should be inclusive in the research and development efforts.

B. CONTRIBUTIONS

This thesis presents a computationally efficient sonar sensor model and prototype implementation that generates data consistent with experimental measurements typical of high-frequency forward-looking (FLS) and multibeam sonar (MBS) sensors. In an effort to fill a research gap that exists in sensors for underwater simulation environments, the sonar sensor model includes several acoustic properties to capture the underwater physical phenomena such as target strength, transmission loss, speckle noise, and scattering. These properties are captured in a point-scattering method developed in [3]. The point-scattering method is coupled with a beamforming application and is verified in MATLAB. These two methods have not been implemented in any sonar sensor model researched and the combination of point-scattering and beamforming provides a unique approach in simulating sonar. We clearly delineate in our sonar model by providing an accurate representation of active sonar transmissions. This is confirmed by analyzing the frequency response of the sonar model and displaying the results in an acoustic image. Qualitative comparisons are drawn using sonar images collected from a sonar similar to the one utilized in this thesis and validate the fidelity of our sonar sensor model. The overall implementation of the point-scattering and beamforming application is balanced between achieving a high-fidelity model and computational efficiency.

C. RELATED WORK

Several works have been published describing the development of a sonar simulation model, but only two have truly integrated the model in a supporting robotics framework. The following paragraphs details their efforts and how the work provided in this thesis compares to their efforts. Table 1 summarizes each sonar model features.

In [4], a Gazebo sonar sensor model is developed using ray tracing. Gazebo is robotic simulation framework and is commonly integrated with the robotics middleware Robot Operating System (ROS). A BlueView P900-45 2D imagining sonar is used for experimental tests and simulated using ROS and Gazebo. The Gazebo ray-tracing

functionality generates a 3D point cloud and is passed to the ROS node and where it is then transformed into a sonar image. We employ a similar process in pursuing a Gazebo sensor model with the use of ROS nodes to pass data information. However, upon further inspection of the model's code, the acoustic properties were either hard coded or commented out. The acoustic model also did not include speckle noise simulation.

In [5], a GPU-based sonar simulator is developed using rasterization. They model two types of sonar, mechanically scanned imaging sonar (MSIS) and FLS. The acoustics features provided in their model as shown in Table 1 are accurate and representative of sound propagation. The robotics platform used is the ROCK-Gazebo framework, and the sonar simulator is written in C++ with OpenCV support as ROCK packages. The sonar sensor model is modeled using a virtual camera and utilizes the following three parameters to render the camera as a sonar: pulse distance, echo intensity, and field of view [5]. Physical tests were also conducted to compare the simulation results with the physical imaging sonar.

Table 1. Summary of Sonar Sensor Models

Model	DeMarco et al. (2015)	Cerquiera et al. (2017)	Rascon Thesis
Sonar Type	FLS	FLS MSIS	FLS MB
Rendering	Rasterization	Rasterization	Rasterization
Features	Reflection[1] Speckle	Reflection Surface Irregularities Attenuation Speckle Reverberation[1]	Reflection Scattering Attenuation Speckle Beamforming
Evaluation	Qualitative Computation Time	Qualitative Quantitative	Qualitative Computation Time

[1] Partially provided feature

D. APPROACH

First, a single-beam model is presented to approximate the measurements from one of the many beams that make up the field of view for an FLS or MB sonar. The model is based upon a ray-based spatial discretization of the model facets, beampattern appropriate for a line array and a point scattering model of the echo level. A MATLAB model is developed using the point-scattering method and beamforming applications. The MATLAB model verifies the sonar model and provides the shell for implementation as a Python node which converts the simulation output (depth buffer and array of normal vectors) to simulated sonar measurements. The remainder of this thesis is structured as follows: Chapter II provides the reader the requisite acoustic background necessary for understanding the simulated sonar model. Chapter III describes the development of the single beam sonar. Chapter IV provides the verification tests, qualitative, and quantitative methods used to analyze the sonar simulation model. Chapter V discuss the experimental results and analysis. Chapter VI concludes the thesis and provides directions for future work.

II. BACKGROUND

Sonar (sound navigation and ranging) can be used to detect object underwater by exploiting sound propagation. This thesis focuses on the use of active sonar systems, specifically FLS and multi-beam sonar (MBS). For an active sonar system, an acoustic pulse is transmitted at a specific frequency for a short duration. The pulse ensonifies an object, and the scattered acoustic wave is captured by a receiver, where it is either recorded or processed by an onboard computer. The background information in this chapter is essential in modeling acoustic transmissions.

A. SOUND PROPAGATION

Underwater sound is composed of superimposed small pressure fluctuations that propagate away from a source into the fluid. The sound wave is modeled as a traveling plane wave. Combining the time and spatial variations in pressure, a plane wave is written as

$$p(x,t) = \mathrm{A}\sin\left(2\pi\mathrm{t} - \frac{2\pi x}{\lambda}\right) = \mathrm{A}\sin(\omega t - \mathrm{k}x) \qquad 1$$

where A is the peak amplitude of the acoustic pressure of a plane wave, f is the frequency in Hz, λ is the wavelength, ω is the angular frequency in rad/s and k is the wavenumber in rad/m [6].

Sound waves carry energy, and the rate at which power flows through a unit of cross-sectional area is defined as the intensity. The time averaged intensity I is given by

$$\mathrm{I} = \frac{P_{rms}^2}{\rho c} \qquad 2$$

where ρ is the density of seawater with a value of 1026 kg/m^3, c is the speed of sound in seawater with a value of 1500 m/s, and P_{rms} is the root mean square amplitude of acoustic pressure [6]. The units of intensity are W/m^2.

In an effort to maintain computational efficiency, only the far field is considered when modeling a single sonar beam. The far field begins at a distance of two wavelengths

away from the source [7]. The wave front in the far field can be modeled as the traveling plane wave previously defined [7]. A sonar beam is considered fully developed in the far field, therefore using the standard sonar equations is applicable. Figure 2 and Figure 3 are examples of a sonar beam pattern in the far field.

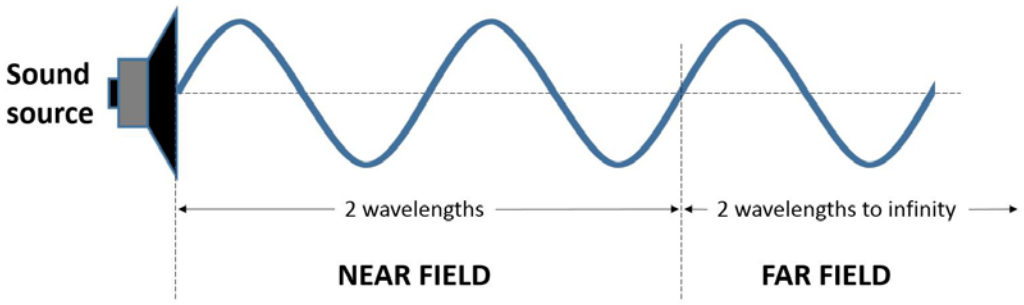

Figure 2. Near Field versus Far Field. Source: [8].

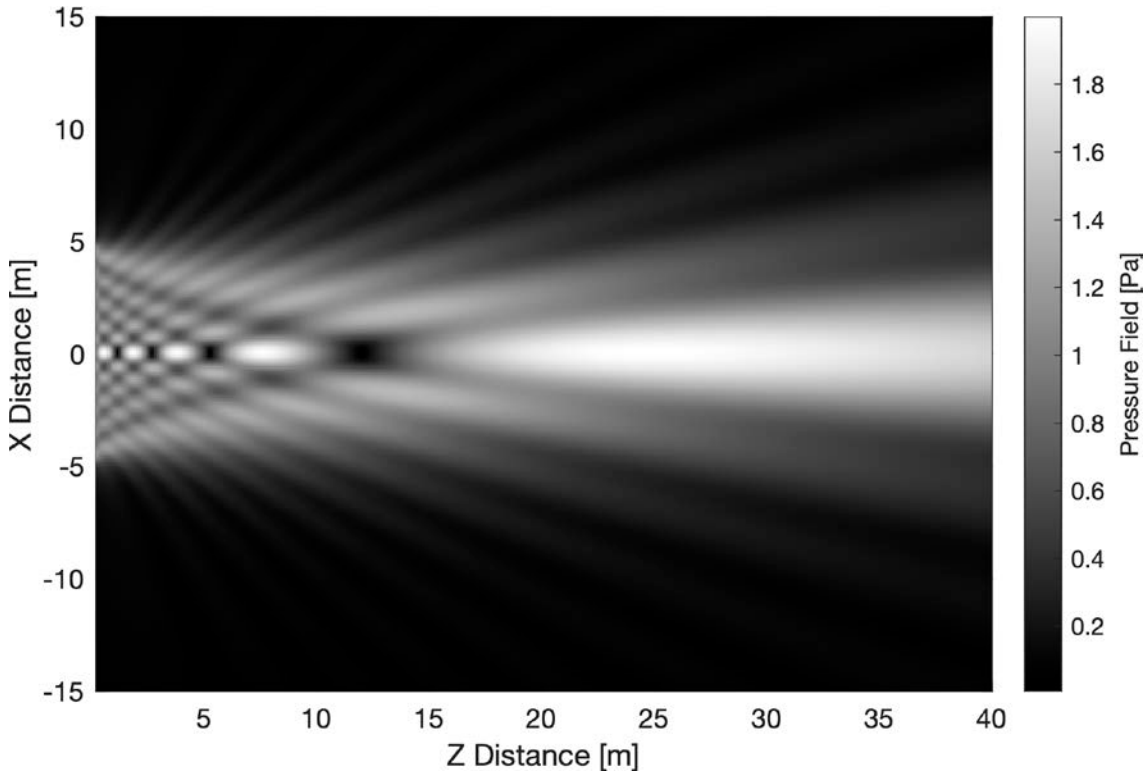

Figure 3. Wave Fronts in the Far Field. Source: [9].

B. ACTIVE SONAR EQUATION

1. Source Level

The source level, *SL*, is the sound pressure level at a distance of 1 m from a sonar and is defined as the decibel of the ratio of the signal intensity, I_0, and a reference intensity, I_{ref}. A common I_{ref} used in underwater acoustics is a reference pressure of 1 μPa.

$$SL = 10\log\left(\frac{I_0}{I_{ref}}\right). \quad (3)$$

2. Transmission Loss

Transmission loss, *TL*, is defined as the attenuation of sound as it propagates a distance 1 m from a sonar source to a range *r* from a source. Sources of transmission loss include spreading loss, refraction, and attenuation loss [10].

Spreading loss represents the weakening of a signal as it spreads away from a source. It is a geometrical effect that commonly takes the form of spherical spreading or cylindrical spreading. Spherical spreading is the decrease in intensity of a signal as the square of the range. In dB, this is

$$TL = 20\log\left(\frac{r}{r_o}\right) \quad (4)$$

where r_o is a range reference typically given as 1 m.

Cylindrical spreading is modeled when the signal is bounded by the surface and the bottom, and therefore intensity varies inversely as range increases. Spherical spreading is an appropriate approximation since the range between the sonar and target are very short paths.

Refraction is present when the sound speed varies with depth and range. Specifically, refraction is an acoustic property more applicable in low-frequency sonar, since the range is typically much larger than the range for a high-frequency sonar. For the purposes of this thesis and its focus on high-frequency sonar, refraction is not considered

in the model because this thesis focuses on high-frequency sonar and very short paths between the sonar system and the target.

Attenuation loss can be due to absorption, scattering, and leakage our of sound channels [10]. Absorption loss is the conversion of acoustic energy into heat and can be modeled by the absorption coefficient, a_c.

$$a_c = \frac{A_1 P_1 f_1 f^2}{f_1^2 + f^2} + \frac{A_2 P_2 f_2 f^2}{f_1^2 + f^2} + A_3 P_3 f^2. \quad 5$$

In Equation 5, f is the transmitted frequency, f_1 and f_2 are the boric acid and magnesium sulfate relaxation frequencies, A_1, A_2, and A_3 are weighting amplitudes, and P_1, P_2, and P_3 are nondimensional pressure correction factors [11]. The absorption coefficient used in this thesis is calculated using a frequency of 900 kHz and standard values for the remaining variables, resulting in a value of 0.0354 dB/m. The attenuation coefficient is calculated by

$$\alpha_p = \frac{a_c \log 10}{20}. \quad 6$$

The wave vector k_w can be related to the attenuation. The wave vector is given by

$$k_w = \frac{2\pi f}{c}. \quad 7$$

Overall attenuation is modeled using a complex frequency-dependent relationship between the wave vector and the attenuation coefficient [12].

$$K = k_w + i\alpha_p. \quad 8$$

According to [10], the combination of spherical spreading and absorption provides a reasonable approximation under a wide variety of conditions.

$$TL = 20 \log(r) + a_c r. \quad 9$$

3. Target Strength

Target strength is defined as the ratio of intensity reflected by an object at a distance 1 m from the object to the incident intensity,

$$TL = 10 \log\left(\frac{I_r}{I_i}\right). \quad (10)$$

The true target strength of an object in water depends on a variety of parameters, to include but not limited to specular reflection, object aspect, surface irregularities, acoustic impedance, and internal structure and composition. Considering how quickly calculating target strength can become computationally inefficient, only the following parameters are considered: object surface area, object surface reflectivity, and reflection using Lambert's law and derivations provided in [10]. Lambert's law is an empirical model that assumes the scattering is perfectly diffuse [12]. It is commonly used to model acoustic scattering at low grazing angles on rough seafloors as described in [13].

4. Echo Level

The echo level, *EL*, is the intensity of the signal that is returned to the receiver by a target. The echo level is a function of source level, two-way transmission loss, and target strength,

$$EL = SL - 2(TL) + TS. \quad (11)$$

This is the simplest form of the active sonar equation with respect to a desired target.

C. SONAR TYPES

Active sonar systems use a transducer to transmit acoustic pulses and can either receive the transmission known as bistatic operation or use a separate receiver in monostatic operation. A single transducer is a single beam sonar that emits only one pulse and can be used in depth measurement, mechanical scanning profiling, and mechanical scanning imaging [14]. Transducers can also be arranged in a series of arrays and emit multiple beams of acoustic pulses. A FLS and MBS consist of an array of beams that have a specific field of view (FOV) defined in the horizontal and vertical aspects. FLS/MB sonars can be very useful when conducting UUV operations and are therefore the type of active sonar system modeled in this thesis.

D. SONAR SIMULATION MODELS

The simulator architecture is comprised of three parts: a rendering engine, a physics engine, and a simulation framework [15]. The rendering engine provides the software for producing 2D or 3D images in real-time. The physics engine is responsible for simulating the physical interactions with objects in the rendered scene. The simulation framework bridges the physics engine and rendering engine and provides an interface to robotics middleware. Figure 4 illustrates the system architecture for a robot simulator.

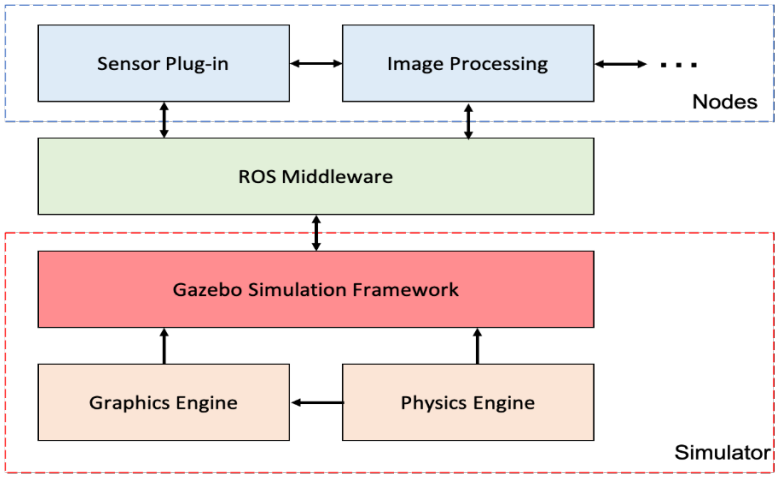

Figure 4. ROS Gazebo Simulation Architecture. Adapted from [15].

Robotics middleware manages the low-level control algorithms used in robots or vehicles and interfaces with the high-level simulation framework. It allows developers the flexibility to design specific sensors and control algorithms for robotic applications while remaining separate from the simulation framework. Robot Operating System (ROS) is a common robotics middleware that allows the creation of plug-ins known as ROS nodes. These individual nodes can control specific aspects of a robot vehicle such as a sonar sensor, vehicle controller, or image processor. ROS enables the nodes to communicate via a common inter-message passing system. Robotic vehicles that use middleware also have the advantage of interfacing with the simulator via hardware-in-the-loop simulation [15].

Ray tracing, tube tracing, and rasterization are common rendering techniques used to display images in computer graphics. Ray tracing consists of a series of rays extending until the ray comes into contact with an object. If a constant water temperature and salinity is assumed, then a constant speed of sound can be used, and the resulting straight-line ray trace is a good approximation of a sonar beam [4]. A similar approach is tube tracing, although the rays involved are used to detect a boundary of an object to form a polygon [4]. Rasterization is a process where an image is represented by polygons, where the vertices are projected onto the screen, and is then broken down into triangles. Using various techniques, a pixel is filled if it is covered by a triangle [16]. Rasterization is integrated with many contemporary GPUs, and therefore more GPUs are used for ray tracing—and are very efficient. Our model proposes the sole use of rasterization to achieve maximum computational efficiency and therefore produce a sonar sensor model that runs in real time.

Gazebo is a general purpose robot simulator that is compatible and commonly integrated with ROS [17]. Gazebo has a variety of readily available sensors such as the depth camera sensor that utilizes rasterization techniques. There is also the added advantage of allowing users to create and integrate ROS nodes without hard coding into the simulator itself. Complementing the sensors are Gazebo ROS packages, providing ROS wrappers around a stand-alone Gazebo. UUV Simulator is an example of this type of package that is implemented with Gazebo ROS nodes and dedicated to simulating UUVs [18]. Overall, Gazebo provides the platform necessary to achieve a specialized sonar sensor model.

THIS PAGE INTENTIONALLY LEFT BLANK

III. SONAR SIMULATION MODEL

Forward looking sonar (FLS) and multibeam (MB) sonar sensors use arrays of discrete acoustic elements to generate 2D and 3D representations of the environment based on received acoustic energy. At the lowest level the received acoustic signal consists of a time history of pressure for each channel of the receiver, where each receive channel corresponds to a discrete element in the array. Acoustic beamforming could be applied to this multi-channel record to produce a time history for each sensor beam. We model the sensory feedback from FLS/MB sensors at beam level by generating a time history for the acoustic intensity that would be estimated as the output of the beamforming operation of a multi-channel array. In this chapter we describe the mathematical model for single-beam (SB) sonar. This model can then be extended to generate the array of beams necessary for simulating FLS/MB sensors.

Figure 4 shows the field of view for a 2D imaging sonar with the FOV for a single sonar beam highlighted in yellow. The following sections detail the single beam sonar development process.

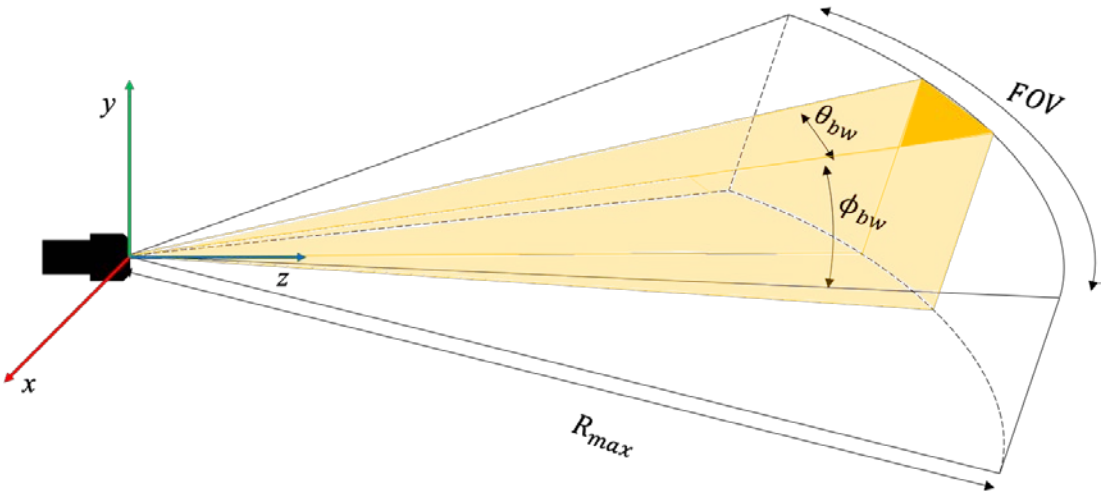

Figure 5. FLS Field of View

A. SINGLE-BEAM SONAR MODEL

1. Ray-Based Beam Model

A single sonar beam is modeled using discrete rays. The individual rays are indexed as i={1, 2, … N} for N. Based on the visual model of the scene the following information is generated for each ray within an individual beam:

- The range r_i as the distance from the origin of the sonar reference frame to the first intersection between the ray and a target in the field of view. The azimuth angle of the ray is fixed in the sensor frame as $\theta_i$and the inclination angle of the ray is ϕ_i. See Figure 5, Figure 6, and Figure 7.
- The incident angle α_i as the difference formed between the ray vector (**z**) and the normal vector (**n**) of the target surface at the location of intersection between the ray vector and the target surface, see Figure 5.
- The reflectivity of the target intersected by the i^{th} ray, $\mu_i \cos(\theta)$, which is a property of the target object model.

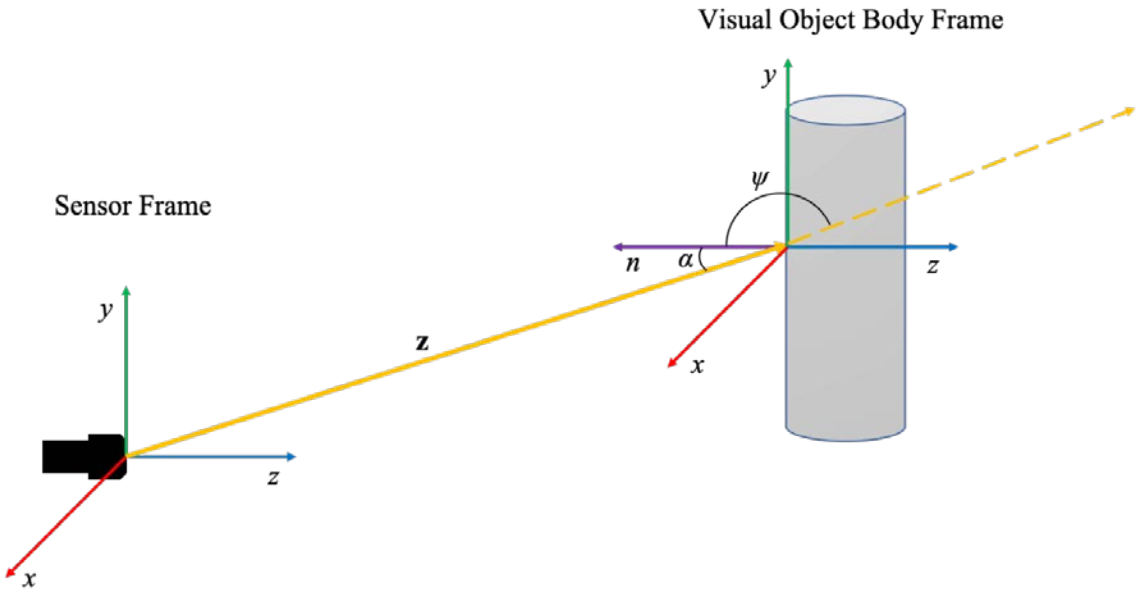

Figure 6. Coordinate Frame System of Sensor Frame and Visual Object Body Frame

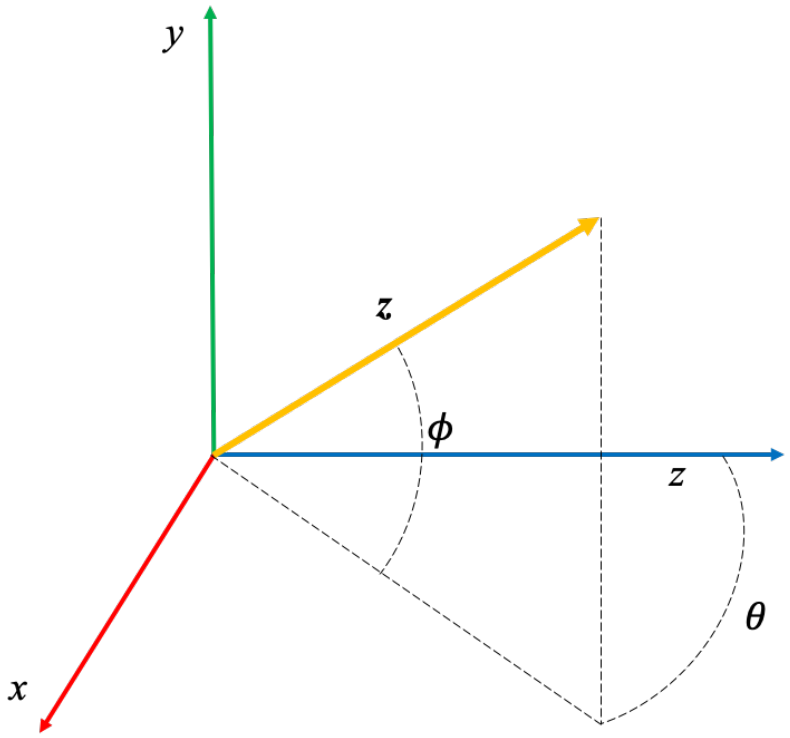

Figure 7. Ray Vector Defined from Sensor Origin Frame

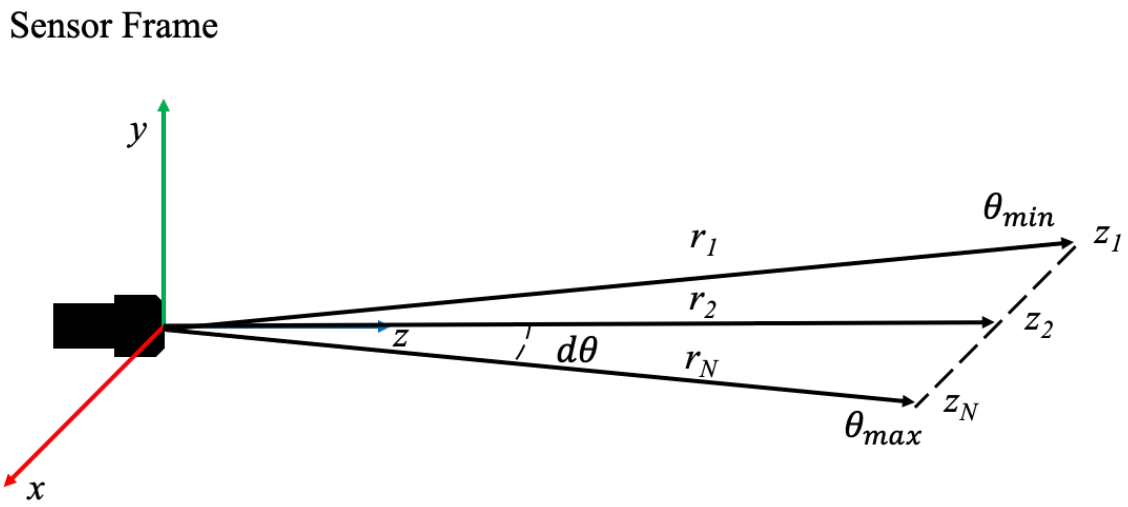

Figure 8. Set of Three Rays Forming a Single Sonar Beam

a. Ray Vector

For our purposes a ray is defined as a vector, **z**, from the sensor frame origin to the first intersection with a visual object within the scene. The location of the intersection is determined based on a fixed orientation of the vector within the sensor frame. The vector is expressed in the sensor frame in 3D rectangular coordinates. Figure 5 illustrates the coordinate frame system for the sensor and the visual object.

b. Ray Incidence Angle

In order to calculate the incident angle α_i for each ray, the supplementary angle, ψ_i, as illustrated in Figure 6 must first be solved by doing the following:

$$\psi_i = \cos^{-1}\left(\frac{\boldsymbol{z_i} \cdot \boldsymbol{n_i}}{\|\boldsymbol{z_i}\|\|\boldsymbol{n_i}\|}\right). \quad 12$$

Since only the angle ψ_i is required, unit vectors $\hat{\boldsymbol{z}}_i$ for the ray direction and $\hat{\boldsymbol{n}}_i$ for the normal direction are used to simplify the calculations and ψ_i can be solved by

$$\psi_i = \cos^{-1}(\hat{\boldsymbol{z}}_i \cdot \hat{\boldsymbol{n}}_i). \quad 13$$

The incident angle α_i for each ray is finally resolved by doing

$$\alpha_i = 180^{\circ} - \psi_i. \quad 14$$

c. Ray Surface Area

The projected ray surface area, *dA*, is the area projected onto the visual object by the individual ray. If the changes in both $d\theta_i$ and $d\phi_i$angles for each ray are assumed to be infinitesimally small, i.e., both are much smaller than 1, then the projected area ray scene can be calculated by doing

$$dA = r_i^2 d\theta_i d\phi_i. \quad 15$$

However, as shown in Figure 8, *dS* is the true surface area patch of the visual object. The following two relationships are used to calculate *dS* for a more accurate surface area approximation.

$$dS_i = \frac{dA}{\cos(\alpha_i)}. \quad 16$$

$$dS_i = \frac{r_i^2 d\theta_i d\phi_i}{\cos(\alpha_i)}. \quad 17$$

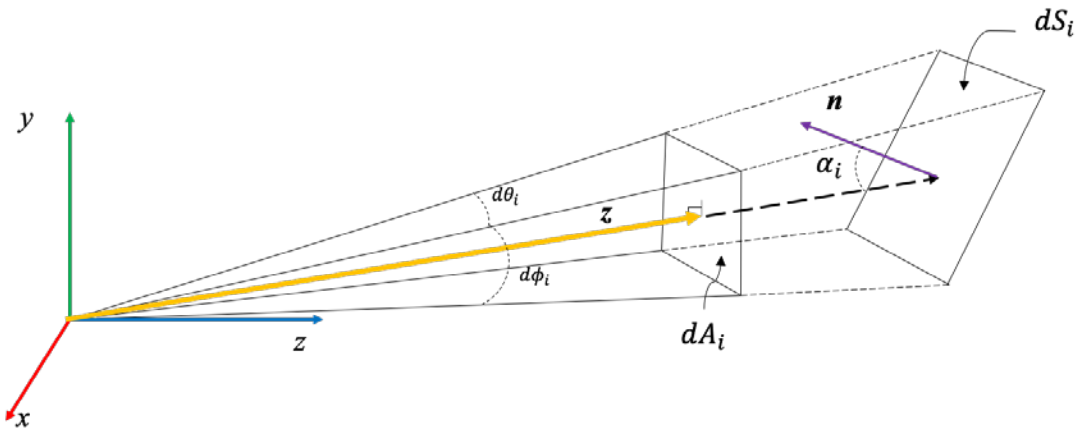

Figure 9. Infinitesimal Surface Area of a Single Ray

d. Target Strength

The target strength model is given by

$$TS_i = 10 \log\left(\frac{I_{r_i}}{I_{I_i}}\right). \quad 18$$

Using Lambert's reflection law, the ratio of intensities is

$$\frac{I_{r_i}}{I_{I_i}} = \mu \cos^2(\alpha_i)\, dS_i. \quad 19$$

Substituting for dS_i using the equation derived earlier, the intensity ratio becomes

$$\frac{I_{r_i}}{I_{I_i}} = \mu \cos^2(\alpha_i)\, r_i^2 d\theta_i d\phi_i. \quad 20$$

e. Point Scattering Model

The physical model given by Equation 8 is simulated using a point-based scattering model developed in [3]. The point-based scattering model generates a spatially

coherent time-series that is useful in simulating narrowband sonar applications such as the FLS and MBS systems. The point-scattering model is adapted to meet the requirements of this thesis. For example, the point-scattering model uses discrete scatterers distributed over a surface [3]. These scatterers are representative of the number of surfaces a ray intersects based on the object's surface mesh. In our model, this approach is computationally inefficient. Therefore, we define each ray in a beam as a scatterer. The overall spectrum of the signal received from each ray, $P(f)$, is computed by

$$P(f)_i = \frac{S(f)\sum_{i=1}^{N} a_i e^{iKr_i}}{r_i^2} \quad 21$$

where $S(f)$ is transmited spectrum of the source, *N* is the number of scatters, a_i is the complex scatterer amplitude, *f* is a frequency vector, and *K* is the complex wavenumber from Equation 8. $P(f)$ is a combination of the physical model for echo level and a complex random scale factor for speckle noise resolved in the frequency domain.

The source level is a user defined input and remains constant for each ray and is modeled in the frequency domain by the transmit spectrum $S(f)$. The transmit spectrum is calculated by doing the following:

$$S(f)_i = S_0 e^{-(f-f_c)^2 b^2 \pi^2}. \quad 22$$

The source level in Equation 22 is defined by S_0 and has units of Pa*m. The frequency vector, *f*, is a linearly spaced vector from f_{min} to f_{max} and centered on the central frequency, f_c. The full width of the transmit spectrum is the bandwidth, *b*, a user provided input based on the sonar specifications. For example, the bandwidth for a BlueView P900-45 FLS is 2.95 kHz.

$$f = f_{min} \cdots f_{max}. \quad 23$$

$$f_{min} = f_c - \frac{b}{2}. \quad 24$$

$$f_{max} = f_c + \frac{b}{2}. \quad 25$$

Spherical spreading is an appropriate assumption for modeling transmission loss. The two-way transmission loss for incoherent scattering in $P(f)$ is captured in the denominator, r_i^2.

The scatter amplitude a_i is calculated by iterating for each ray the following:

$$a_i = \frac{(\xi_{xi} + i * \xi_{yi})}{\sqrt{2}} \sqrt{\mu \cos^2(\alpha_i)\, r_i^2 d\theta_i d\phi_i}. \quad 26$$

Although the random variable, ξ_i, is indexed by i to represent the ray index, the real random variable and the complex random variable must both be generated and different from each other, hence the x and y notation. Overall, the random variables are representative of Gaussian noise and for our purposes, satisfies the speckle noise requirement [3]. The variables under the square root represent the target strength of an incident ray on an object.

Finally, the discrete inverse Fourier transform of $P(f)$ transfers the function from the frequency domain to the time domain and results in a time-series of the received signal [19].

$$P(t_n) = \sum_{m=1}^{N} e^{-i2\pi f_m t_n} P(f_m) \Delta f \quad 27$$

2. Beam Pattern Geometry

We combine the ray-based discretization of the single-beam with a linear array beam pattern model. The beam pattern of an array is defined in polar coordinates where the acoustic intensity is the distance along the radial axis and the angle is relative to the transducer axis. The beam pattern is visualized as one main lobe in the center with smaller side lobes radiating away from the main axis. The main lobe contains most of the energy and is along the main axis as shown in Figure 9. By inspection, the highest return will be along the main axis as the response decreases off axis. Therefore, the echo level depends on the size and position of a target within a beam. The beam width, θ_{bw}, or also referred to as the half intensity beam angle, is marked at -3 dB on the main lobe. Half of the beam width will be annotated as

$$\theta_w = \frac{\theta_{bw}}{2}. \quad 28$$

The beam pattern, $B(\theta)$, expresses the EL as a function of θ, the angle relative to the main axis,

$$EL = EL|B(\theta)|^2. \quad 29$$

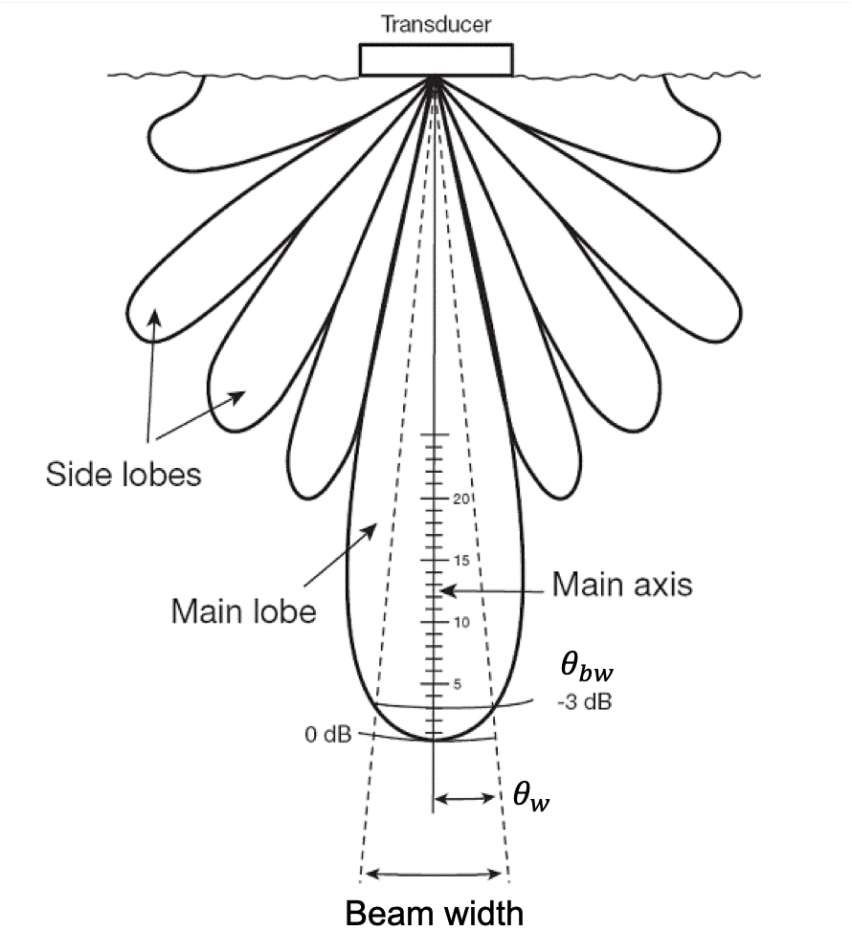

Figure 10. Beam Pattern Schematic. Adapted from [20] and [21].

3. Linear Array Beam Pattern

For a continuous line array of length L, radiating energy at a wavelength λ, the beam pattern is that of a uniform aperture function. The radiated power is modeled as a normalized sinc function

$$|B(Lu)|^2 = |\mathrm{sinc}(Lu)|^2 = \begin{cases} 1 & \text{for } Lu = 0 \\ \left|\dfrac{\sin(\pi Lu)}{\pi Lu}\right|^2 & \text{otherwise} \end{cases} \quad 30$$

where u is the electrical angle

$$u = \frac{\sin(\theta)}{\lambda}. \quad 31$$

The half intensity point, θ_w , can be solved by setting

$$|B(\theta_w)|^2 = \left|\frac{\sin\left(\pi \frac{L}{\lambda}\sin(\theta_w)\right)}{\pi \frac{L}{\lambda}\sin(\theta_w)}\right|^2 = \frac{1}{2} \quad 32$$

where $k = 1$. For high frequency sonar, we can assume $L \gg \lambda$, and then θ_w becomes

$$\frac{L}{\lambda} = \frac{0.884}{\theta_{bw}} = \frac{0.442}{\theta_w}. \quad 33$$

Using the relationship from Equation 28 the final beam pattern is

$$|B(\theta_w)|^2 = \left|\frac{\sin\left(\pi \frac{0.884}{\theta_{bw}}\sin(\theta)\right)}{\pi \frac{0.884}{\theta_{bw}}\sin(\theta)}\right|^2 \quad 34$$

where θ_{bw} is in radians. Figure 11 illustrates this beam pattern for Equation 34.

As shown in Figure 8, the 2D beam model the azimuth angle, θ, is defined as rotation about the z sensor frame axis and the main axis is defined as coincident with the sensor frame z axis. The use of *N* rays, with angular location distributed evenly between in the range $[-\theta_{min}, \theta_{max}]$ effectively discretizes the beam in the azimuth dimension. The beam width is the interval $[-\theta_{min}, \theta_{max}]$ and the intensity is a function of θ for the main lobe only and is illustrated in Figure 11.

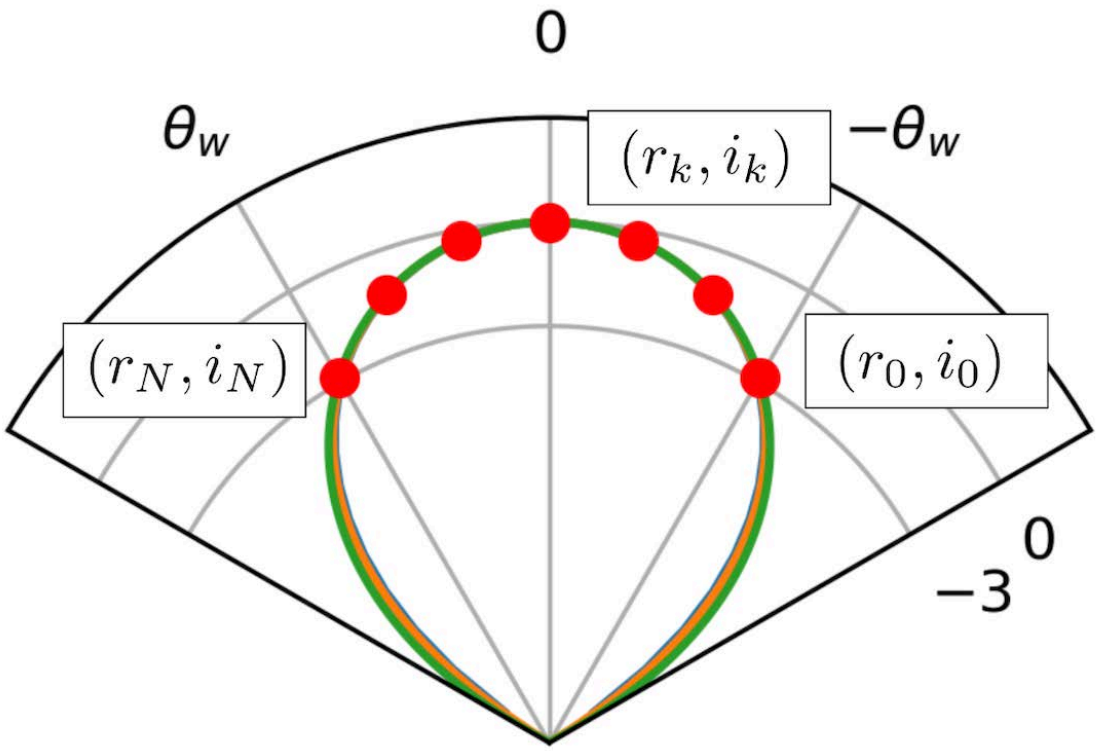

Figure 11. Example of Weights in a Sampled Beam Pattern

The radial dimension is discretized by considering *M* samples within the interval $[R_{min}, R_{max}]$. The number of samples in the radial dimension is a function of the range interval and the range resolution (dR) and can be expressed as

$$M = \left\lceil \frac{R_{max} - R_{min}}{dR}. \right\rceil \quad 35$$

For example, the BlueView P900-45 FLS, has a specified range interval of 2–60 m with a range resolution of $dR = 0.0254$ m resulting $M = 2283$ samples for each beam.

4. Single Beam Sampling Application

The single sonar beam model generates a single pair of range and intensity values: (r, I). The pair is generated by taking a uniform sample of the beam pattern. Figure 11 is an example of the beam pattern sample. Each sample is a ray at θ_k in the range $[-\theta_w, \theta_w]$ with the associated range and intensity pair (r_k, i_k). The set of ordered pairs from all the rays is $R = \{(r_0, i_0), (r_1, i_1), \ldots, (r_N, i_N)\}$.

Using a uniform weighted average to model the beam pattern effect, the echo level for the beam can be calculated as following:

$$\overline{EL} = \bar{\imath} = \frac{\sum_{k=0}^{N} w_k i_k}{\sum_{k=0}^{N} w_k}. \quad 36$$

The weights are determined from the beam pattern function and calculated as follows:

$$w_k = |B(\theta_k)|^2. \quad 37$$

Figure 10 demonstrates the beam pattern sampling.

B. MATLAB IMPLEMENTATION

MATLAB is used to develop a prototype implementation of the single-beam sonar model. Once verified with respect to consistency of the mathematical model, this prototype will then be the basis for implementing the model in Gazebo/ROS framework (typically accomplished using C++ and/or Python). As an example and for the purposes of verification, we use parameters consistent with the BlueView P900-45D instrument. The pertinent information of the sonar is given in Table 2.

Figure 12 demonstrates the general algorithm used in MATLAB for implementing the sonar model. First a single beam is constructed using three beams, where this spatial discretization is parameterized by the input parameter N. Each ray is representative of a single scatterer.

Table 2. Sonar Model Beam Inputs

BlueView P900-45 Parameters	
Frequency	900 kHz
Bandwidth	295 kHz
Resolution	0.0254 m
Beam Width	0.1 rad
Source Level	220 dB re 1 μPa
Environmental Parameters	
Sound Speed	1500 [m/s]
Lambert Parameter, μ	10^{-4} [dimensionless]
Absorption	0.0354 [dB/m]

The transmit spectrum, wave vector, and attenuation are defined as described in Chapter III. The outer loop generates the beam data and applies the beam forming, and the inner loop calculates the ray information by applying the point-scattering model. Lastly, the sound pressure level is calculated for each beam.

The output of the sonar model is a time-history of the received sound pressure level. This information is plotted and the signal behavior is analyzed. The resultant figures are provided and discussed in Chapter IV.

The MATLAB code that is used to implement this model can be found at the GitHub repository for the MATLAB code at https://github.com/Field-Robotics-Lab/matlab_sonar_prototype [22].

Algorithm 1 Sonar Model

$ray \leftarrow calculateRay(distance, \alpha)$
$S(f) \leftarrow calculateTransmitSpectrum(frequency)$
for k = 1:nBeams **do**
  for i = 1:nRays **do**
    $noise \leftarrow Gaussian(x, y)$
    $amplitude \leftarrow calculateAmplitdue(noise, \alpha, distance)$
    $P(f) \leftarrow calculateReceiveSpectrum(S(f), amplitude, distance)$
  end for
  $beampattern \leftarrow calculateBeamPattern(\theta)$
  $P(t) \leftarrow transformReceiveSpectrum$
  $P(t)modified \leftarrow calculateModifiedReceiveIntensity(P(t), beampattern)$
end for
$SPL \leftarrow calculateSoundPressureLevel(P(t)modified)$

Figure 12. Sonar Model MATLAB Algorithm

THIS PAGE INTENTIONALLY LEFT BLANK

IV. SONAR MODEL EVALUATION

The results of the MATLAB sonar model prototype are presented in this chapter. The MATLAB model is executed using the computer system provided in Table 3. The MATLAB version used in this thesis is 2019b. No special toolboxes are required.

Table 3. Computer System Information

Subsystem	Version
Operating System	Ubuntu 18.04.4 LTS 64-bit
CPU	AMD® Ryzen 9 3900 12-core processor
GPU	NVIDIA GeForce RTX 2070 SUPER
Memory	62.8 GiB

A matrix of diagonal lines in range - angle space, equally separated and extending a distance of 1.5 m is created and illustrated in Figure 13. This object model is easier to represent in MATLAB versus an object such as a cylinder, which consists of many surface facets of varying normal angles and distances. Each line in the MATLAB object model is representative of a scattering surface where the sonar model is applied. Once the MATLAB prototype is implemented in the Gazebo ROS framework, the data for a ray intersecting an object in a rendered scene will be provided to the sonar model.

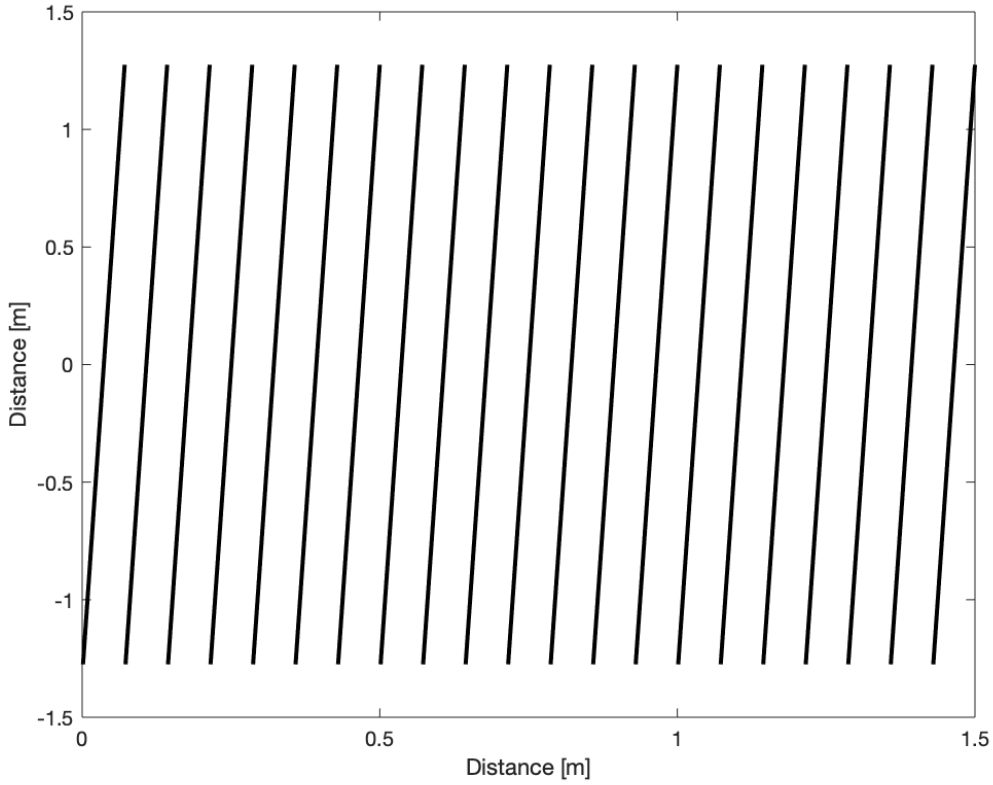

Figure 13. Diagonal Lines Representing a Scattering Object

The first experimental case consists of 51 beams and 21 rays per beam. The frequency response of one beam without beamforming is shown in Figure 14. The travel time expected for a signal to travel 1.5 m is 0.002 sec. Two-way time response from Figure 14 is 0.001929 s, 3 % faster than the anticipated time. The signal behaves as expected, with the sound pressure level decaying as the signal detects an object that is further in distance.

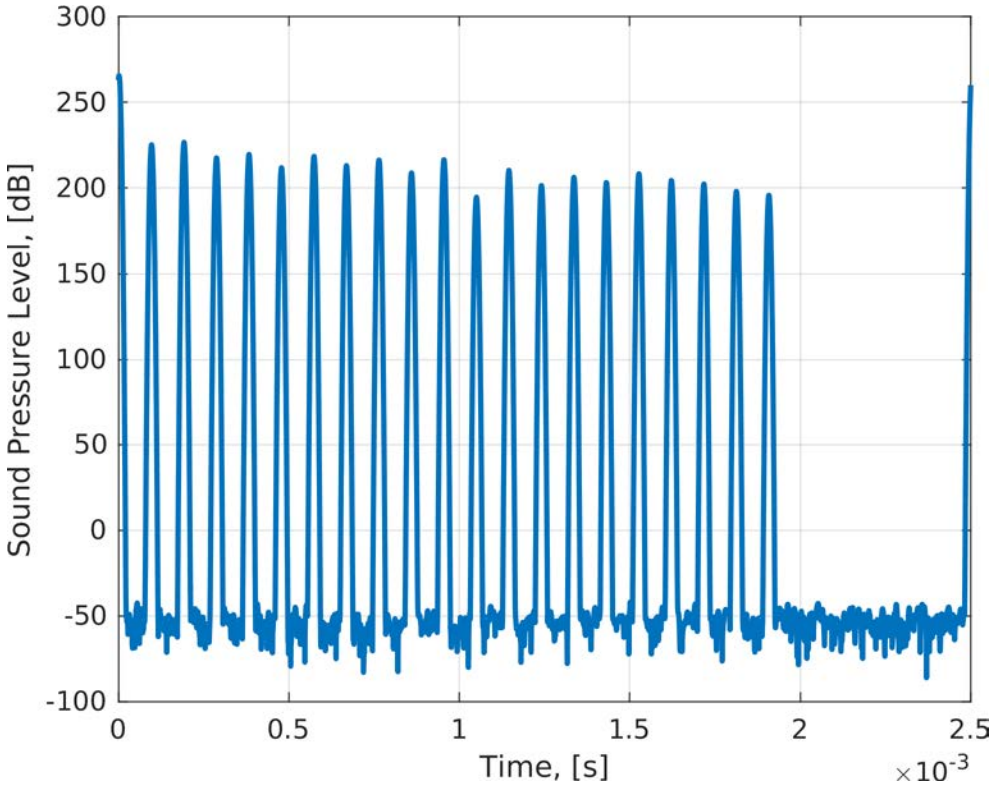

Figure 14. Received Sound Pressure Level Response for One Beam

Figure 15 illustrates the same signal in Figure 14 although includes beam forming. The travel time for the beamformed signal is 0.002002, responding within 0.1 % for the expected travel time. The decay of the signal is also more pronounced than the decay of the signal for a beam without beamforming. This is indicative that as a scattering surface is located at an azimuth angle further away from the center beam of a sonar, the weighted signal response of that beam is going to result in a lower sound pressure level. This is an indication that the beamforming simulation behaves as expected.

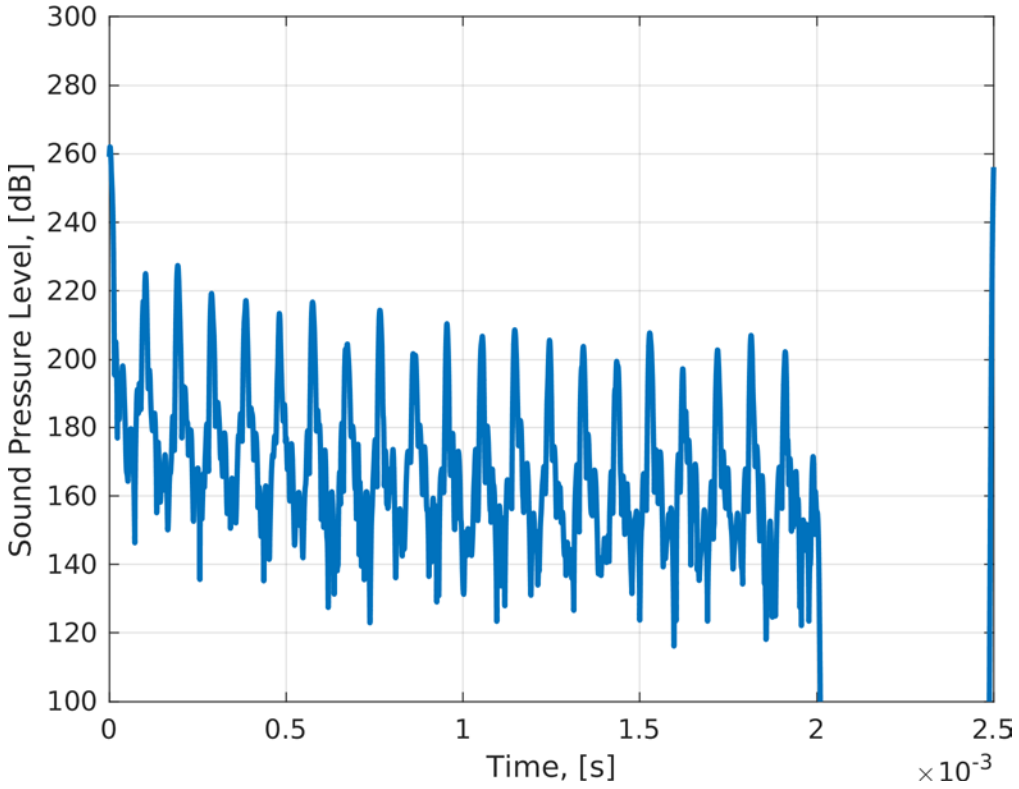

Figure 15. Received Sound Pressure Level with Beamforming for One Beam

Figure 16 illustrates the development of the sonar image for 51 beams. The Y axis is the azimuth angle centered at the sonar sensor origin in rectangular coordinates. The top image plots the response for a signal without beamforming applied and the bottom image includes beamforming. For the top image, the response strictly correlates to the location of the scatterers on the object lines. The sound pressure level decreases as the range increases as expected and confirms the frequency response from Figure 14. The bottom image in Figure 16 depicts a response that is not restricted to the location of an object line, but rather has a smeared effect that resembles an image typical of a FLS. This image demonstrates how the behavior of one scatterer affects all the beams, and that is a direct result of beam forming simulation.

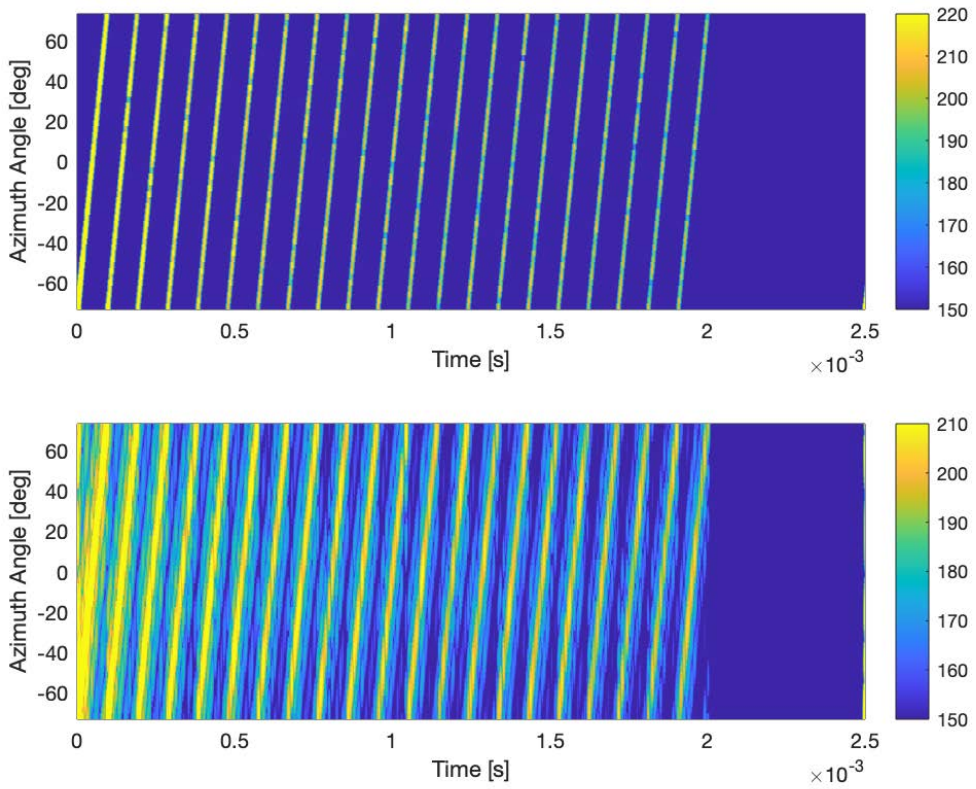

Figure 16. Sonar Image for 51 Beams

Figure 17 is created by taking a vertical slice from the bottom image of Figure 16. This slice depicts the information of one scatterer in the 51 beams. A sinc pattern is overlayed to compare the accuracy of the simulated beam forming. The simulated beam pattern mimics a similar behavior to that of the plotted sinc function, demonstrating the correct application of beamforming in the sonar model.

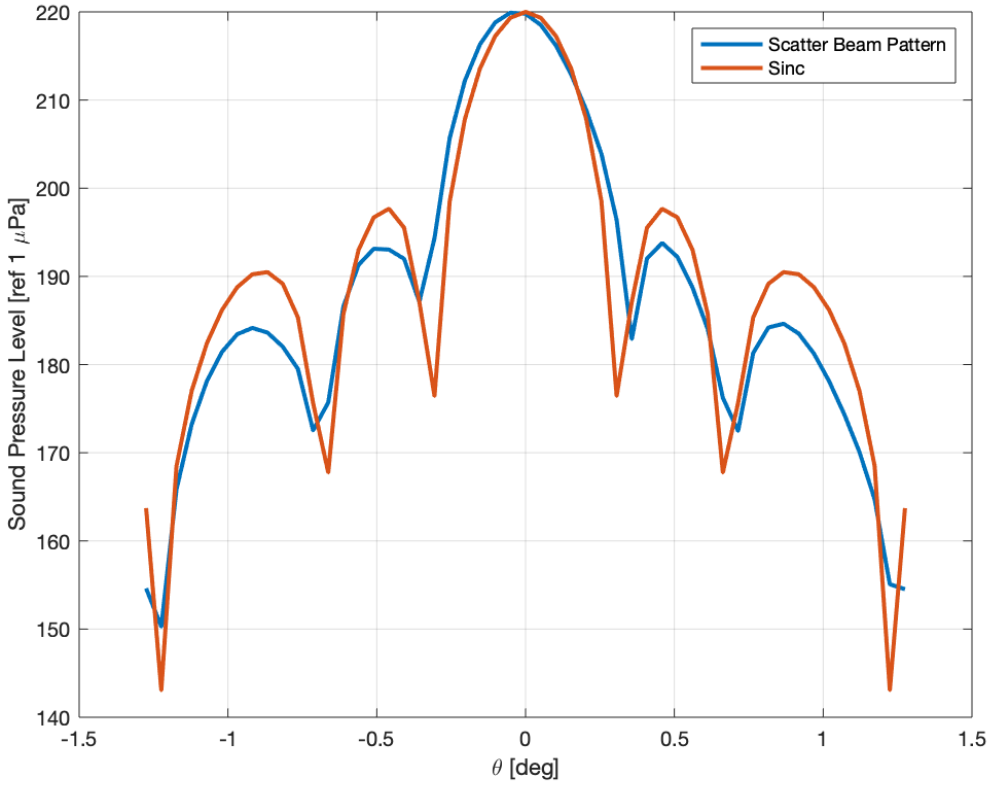

Figure 17. Beam Pattern Comparison for One Scatterer in 51 Beams

The second experimental case analyzes 101 beams with 21 rays per beam. Figure 18 illustrates the sonar image for increased beam count. The bottom image in Figure 18 demonstrates a common effect in acoustic imaging, which is the mirroring of a sonar image and appearance of replicated or “ghost” reflections. In designing a sonar transducer, efforts are made to minimize the effect of side lobe grating to produce a more accurate acoustic image. Our efforts aim to simulate acoustic phenomena, and the result in the bottom image of Figure 18 demonstrates that achievement. Figure 19 illustrates the beam pattern response taken from a vertical slice of Figure 18. The behavior of this scatter generally follows the plotted sinc function. The beampattern also shows the scatterers are too close together, and add coherently.

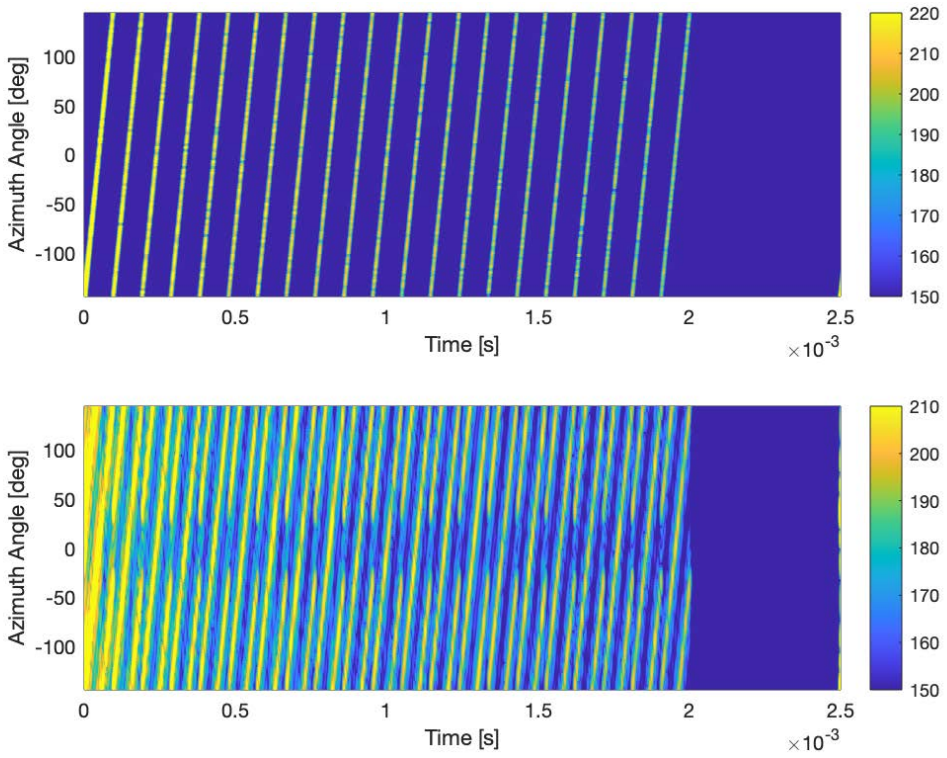

Figure 18. Sonar Image 101 Beams

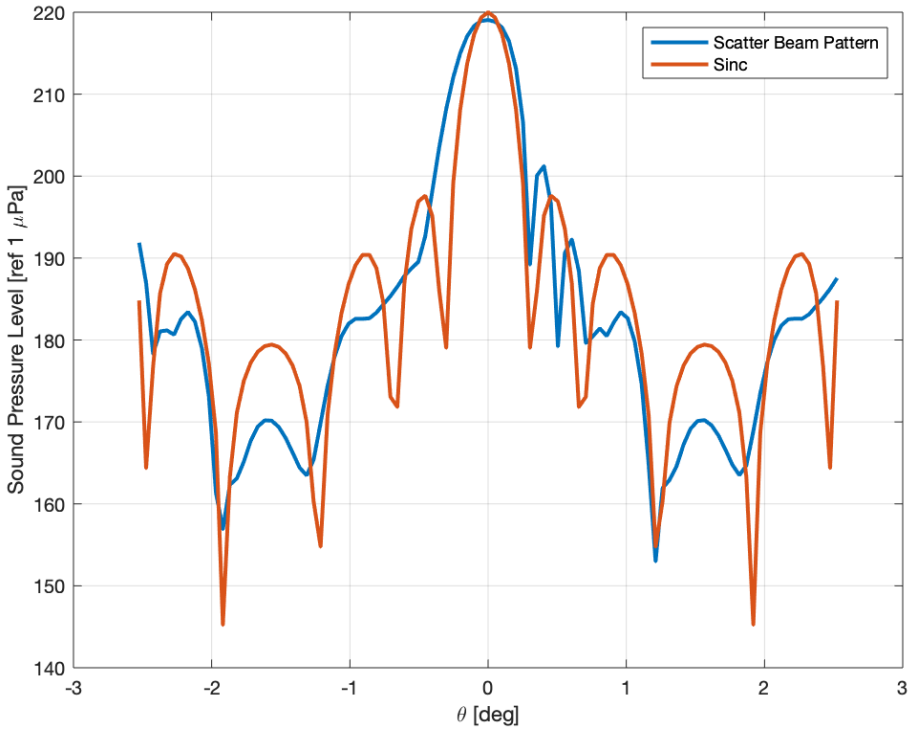

Figure 19. Beam Pattern Comparison for One Scatterer in 101 Beams

The third experimental case simulated 256 beams with 21 rays per beam. This number of beams is the same for the BlueView P900-45. The sonar image in Figure 20 illustrates an image that is "washed out." The number of beams in this scenario demonstrates the effect of grating lobes, which are clearly demonstrated in Figure 21. The grating lobes are identified as the major peaks with a similar amplitude to the main lobe located at 0°. This phenomenon is a common issue associated with transducer arrays and their appearance and effect on the sonar image contributes to the fidelity of the sonar model.

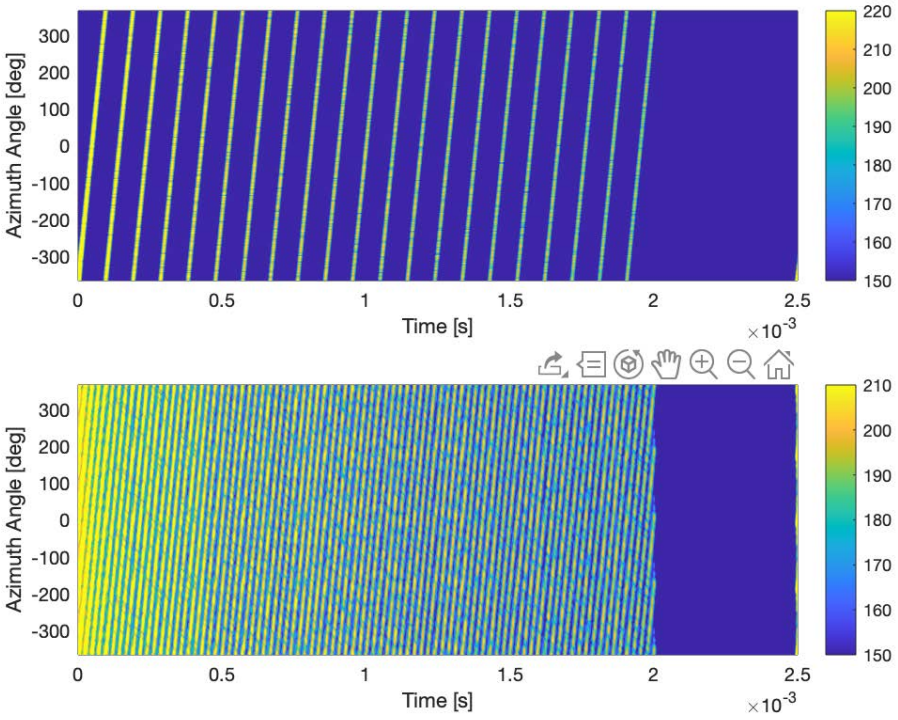

Figure 20. Sonar Image for 256 Beams

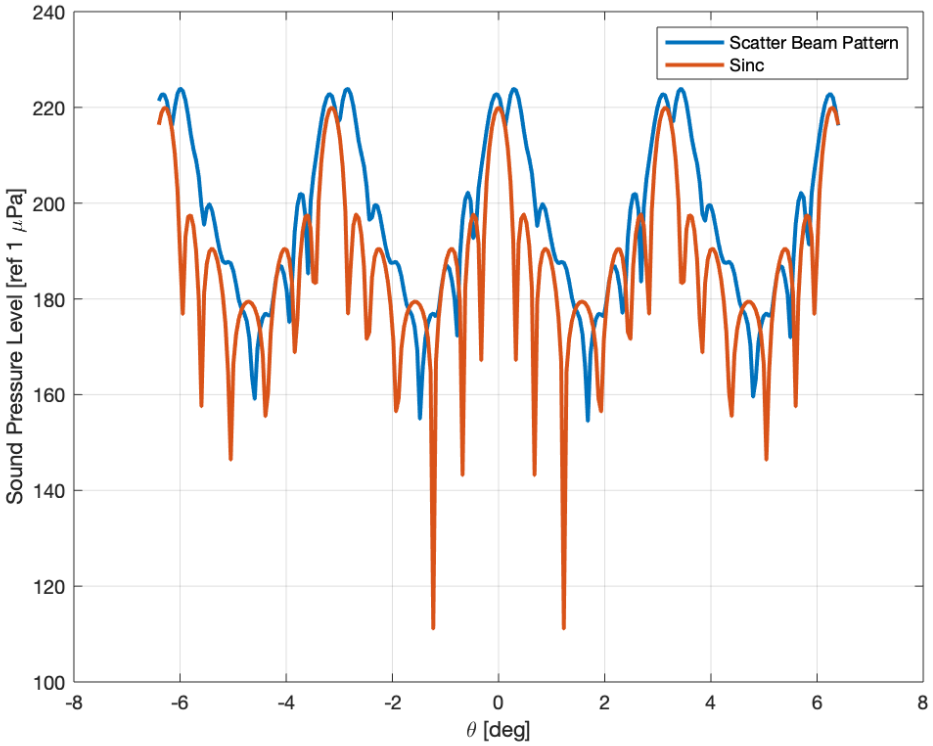

Figure 21. Beam Pattern Comparison for One Scatterer in 256 Beams

As a qualitative comparison, we took an actual acoustic image produced by a BlueView M900-2250 which uses the same frequency as the BlueView P900-45 of 900 kHz. Figure 22 is a screen capture of data collected of bridge pilings from the BlueView M900-2250. The purpose of this image is to demonstrate the effect of the multiple reflections that a sonar sometimes produces. At a distance of 30 m, the bright spot is actually a mirrored response that is located at the 20 m mark. Similar effects are located at the 10 m mark, although the object itself is not in the sonar image.

Figure 23 is a sonar image captured by a BlueView P900-45, the same model used in this thesis. This sonar image ensonifies a box located on the sea floor. The bright returns are the reflections from the side of a box closest to the sonar and the top of the box is shadowed. Comparing our model from Figure 14 to the sonar image in Figure 23, our sonar model provides an acceptable acoustic image representation via simulation. One major difference that is noted is the visualization of shadowing. Our model exhibits no returns beyond the simulated objects reflections otherwise the simulation can become

computationally inefficient. In a real underwater environment, hard shadows are not necessarily the case since there are acoustic returns due to various acoustic properties such as multi-path.

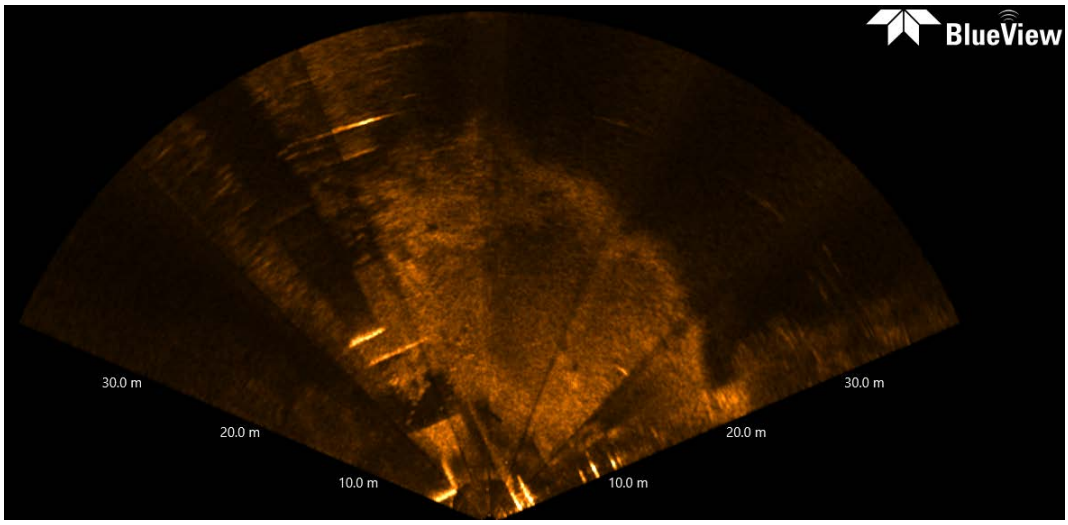

Figure 22. BlueView M900-2250 Sonar Image. Source: [23].

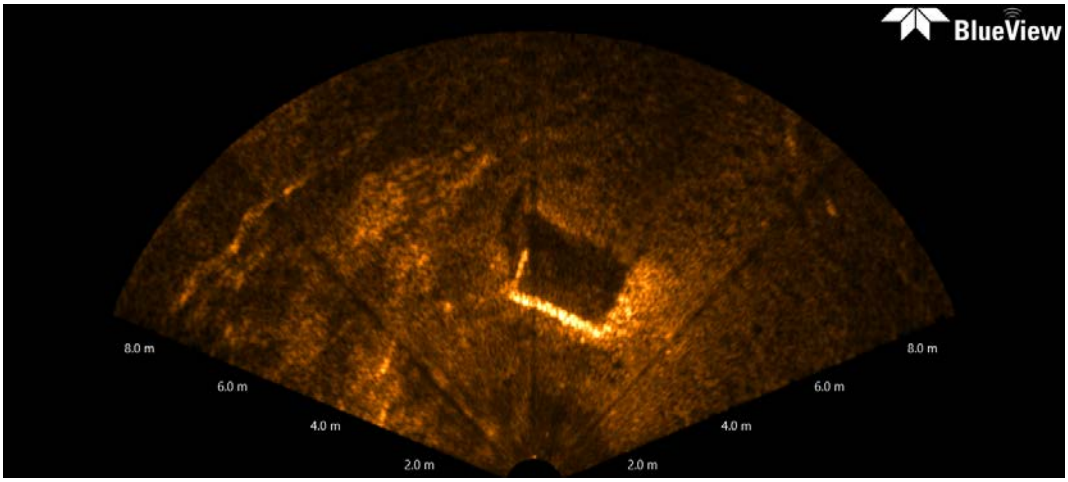

Figure 23. BlueView P900 Sonar Image. Source: [23].

Finally, the sonar model was executed 50 times and the computation time for each run was averaged. Table 4 provides the average execution time for various number of beams. The execution of the MATLAB code only included the production of the sonar image. The results indicate that as the number of beams is increased, the computation time also increases. For the purposes of this thesis, these times are acceptable as a prototype on a CPU. This sonar model is intended to be implemented in the Gazebo ROS framework, where the model is integrated with a GPU and the computational processing is significantly faster. Although efforts can be made continuously during implementation in the robotics framework to refine the model and ensure real-time execution.

Table 4. Computation Time for Various Number of Beams

Number of Beams	**Average Time [s]**	**Standard Deviation**
51	0.3075	0.0699
101	0.7699	0.0124
256	1.8380	0.0401

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSION

A. DISCUSSION

This thesis presents a sonar model to generate data comparable to active sonar transmissions of a FLS or MBS. The acoustic background knowledge provided the mathematical groundwork necessary for accurately representing acoustic propagation, target strength, transmission loss, beam pattern, and speckle noise. Understanding the various types of sonars aided in selecting an appropriate sensor best geared for underwater robotic applications. Research into simulation frameworks resulted in choosing a robot simulator that allowed for the development for our own sensor, with the added capability of employing the sensor with UUVs. The knowledge gained from researching how simulators work and how to best employ them aided in the design for a sonar model that is computationally efficient once inserted into a simulation framework. Overall, the background research helped significantly guide the development of our sonar model.

Starting with a single sonar beam proved extremely useful when troubleshooting the sonar model. Once the single beam sonar executed correctly, the model was analyzed for areas where computational efficiency could benefit. The MATLAB model included pre-allocation for vectors, the use of matrix and vector math, and the minimal use of loops or functions. Adding the conditional statement to avoid calculations for empty or zero distance vectors minimized the amount of calculations required. A second conditional statement limiting the range of the simulated sonar also minimized the amount of calculations required. These minor yet impactful statements helped to add finesse to the final product.

The sonar model analyzed at two stages in the formation of the acoustic image, without beamforming and with beamforming, confirms that the addition of beamforming produces a sonar image that is acceptable for sonar simulation. Effects such as repeated reflections are consistent with actual acoustic images, and the formation of grating side lobes accurately portray the physical behavior of a sonar transducer array. The MATLAB

prototype supports the implementation of a computationally efficient sonar sensor model that can produce simulated data in real time.

The sonar model provided in this thesis fulfills the requirements for a sonar sensor simulator that produces realistic data and exhibits underwater acoustic phenomena. The sonar MATLAB model verifies through time-history plots of the sensors that the signal behaves as expected for an active sonar transmission. The comparison of the simulated sonar images also confirm that the sonar sensor model produces acoustic images that are acceptable for simulation. Although input into Gazebo ROS framework was not accomplished during the time of this thesis, the groundwork is sufficiently laid out for seamless transition into programming the ROS nodes.

B. FUTURE WORK

Additional acoustics that can be included in the sonar model are multi-path, reverberation, and surface irregularities. Multi-path and reverberation could potentially become computationally inefficient, yet details in [24] provide a method for partially simulating reverberation to achieve a similar effect. Surface irregularities can be added by adding different surface reflectance values for different scattering surfaces. The value used in this thesis assumed a smooth cylinder, therefore we had a very small reflectance value.

The MATLAB implementation provided in this thesis is in the process of being transformed into a Python ROS node integration with Gazebo. Once the ROS node is created, the sound pressure levels can be viewed in the ROS rqt viewer. When the sonar model is verified operational and accurate within ROS and Gazebo, the data needs to be transformed into a 3D point cloud where the Gazebo uuv_sim sonar viewer can be used to display the sonar image in polar coordinates.

UUV_sim does provide a sonar viewer provided it has a 3D point cloud to read into the node. However, as demonstrated in [4], they created their own sonar viewer that provides a higher fidelity image when compared to the sonar viewer from uuv_sim. The use of the sonar viewer provided in [4] is sufficient for current projects, but the ultimate

goal is to feed sonar data in a specific sensor format. Therefore, a specialized sonar viewer will require development.

The sonar sensor data format for many of the commercially available sonars are proprietary. However, there are a few open-source sonar data formats that could potentially be used to develop a ROS node that takes the information from the FLS and converts the 3D point cloud into the specific format. For example, Leidos manages a generic sensor format (GSF) which is a standard file format designed to efficiently store sensor data [25]. GSF has proven to be very useful for sonars such as the FLS and MBS due to the large volume of data collection. Its modular design makes it a prime candidate for implementation into a ROS node.

Once a sonar viewer is implemented, experimental tests can be conducted to compare the simulation results and the accuracy of the model. NPS has a BlueView P900-45D sonar and two water tanks for executing field tests. A physical cylinder model has already been constructed out of PVC for these field tests. NPS also has access to the BlueView Software Development Kit for further analysis of the collected sonar data. The images produced by experiments and the simulation can be compared using common computer vision methods such as the peak signal to noise ratio (PSNR), mean square error (MSE), and structural similarity index (SSIM). The computational efficiency of the sonar simulation model can also be verified against its computational time and the real time acquisition of sonar data.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] "UnmannedMaritimeSys-Small.pdf." Accessed Aug. 13, 2020. [Online]. Available: https://www.navsea.navy.mil/Portals/103/Documents/Exhibits/SNA2019/UnmannedMaritimeSys-Small.pdf?ver=

[2] C. S. Timperley, A. Afzal, D. S. Katz, J. M. Hernandez, and C. Le Goues, "Crashing simulated planes is cheap: Can simulation detect robotics bugs early?" In *2018 IEEE 11th International Conference on Software Testing, Verification and Validation (ICST)*, Apr. 2018, pp. 331–342, doi: 10.1109/ICST.2018.00040.

[3] D. C. Brown, S. F. Johnson, and D. R. Olson, "A point-based scattering model for the incoherent component of the scattered field," *J. Acoust. Soc. Am.*, vol. 141, no. 3, pp. EL210–EL215, Mar. 2017, doi: 10.1121/1.4976584.

[4] K. J. DeMarco, M. E. West, and A. M. Howard, "A computationally-efficient 2D imaging sonar model for underwater robotics simulations in Gazebo," in *OCEANS 2015 – MTS/IEEE Washington*, Washington, DC, Oct. 2015, pp. 1–7, doi: 10.23919/OCEANS.2015.7404349.

[5] R. Cerqueira, T. Trocoli, G. Neves, S. Joyeux, J. Albiez, and L. Oliveira, "A novel GPU-based sonar simulator for real-time applications," *Comput. Graph.*, vol. 68, pp. 66–76, Nov. 2017, doi: 10.1016/j.cag.2017.08.008.

[6] L. M. Brekhovskikh and I. P. Lysanov, *Fundamentals of ocean acoustics*, 3rd ed. New York: Springer, 2003.

[7] L. E. Kinsler, Ed., *Fundamentals of acoustics*, 4th ed. New York: Wiley, 2000.

[8] "Sound fields: Free versus diffuse field, near versus far field." Accessed Aug. 12, 2020. [Online]. https://community.sw.siemens.com/s/article/sound-fields-free-versus-diffuse-field-near-versus-far-field

[9] D. Olson, email, Aug. 21, 2020.

[10] R. J. Urick, *Principles of underwater sound*, 3rd ed. Westport, CT: Peninsula Publishing.

[11] R. E. Francois and G. R. Garrison, "Sound absorption based on ocean measurements. Part II: Boric acid contribution and equation for total absorption," *J. Acoust. Soc. Am.*, vol. 72, no. 6, pp. 1879–1890, Dec. 1982, doi: 10.1121/1.388673.

[12] D. R. Jackson and M. D. Richardson, *High-frequency seafloor acoustics*. New York: Springer, 2007.

[13] H. Boehme et al., “Acoustic backscattering at low grazing angles from the ocean bottom. Part I. Bottom backscattering strength,” *J. Acoust. Soc. Am.*, vol. 77, no. 3, pp. 962–974, Mar. 1985, doi: 10.1121/1.392064

[14] R. G. Cerqueira, L. R. de Oliveira, and J. C. Albiez, “Ficha catalográfica elaborada pelo Sistema Universitário de Bibliotecas (SIBI/UFBA), com os dados fornecidos pelo(a) autor(a),” [A multi device sonar simulator for real time robotic underwater robotic applications], Federal University of Bahia, 2019.

[15] D. Cook, A. Vardy, and R. Lewis, “A survey of AUV and robot simulators for multi-vehicle operations,” in *2014 IEEE/OES Autonomous Underwater Vehicles (AUV)*, Oct. 2014, pp. 1–8, doi: 10.1109/AUV.2014.7054411

[16] Scratchapixel, “Rasterization: A practical implementation,” *Scratchapixel*. Accessed May 11, 2020. [Online]. https://www.scratchapixel.com//lessons/3d-basic-rendering/rasterization-practical-implementation

[17] “Gazebo.” Accessed Aug. 23, 2020. [Online]. http://gazebosim.org/

[18] Musa M. Marcusso, S. Scherer, M. Voss, L. Douat, and T. Rauschenbach, “UUV Simulator: A Gazebo-based package for underwater intervention and multi-robot simulation,” presented at the OCEANS 2016 MTS/IEEE Monterey, Sep. 2016, doi: 10.1109/oceans.2016.7761080.

[19] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-time signal processing*, 2nd ed. Upper Saddle River, N.J: Prentice Hall, 1999.

[20] “Acoustics unpacked.” Accessed Jun. 13, 2020. [Online]. http://acousticsunpacked.org/AcousticBackground/AcousticTransducers.html

[21] “3. Principles of instruments for fisheries acoustics.” Accessed Aug. 26, 2020. [Online]. http://www.fao.org/3/x5818e/x5818e04.htm#3.1.3% 20transducers%20and%20acoustic%20beams

[22] *Field-Robotics-Lab/matlab_sonar_prototype*. Monterey, CA, Field-Robotics-Lab, 2020.

[23] “ProViewer™ – Forward looking imaging sonars (2D) – BlueView.” Accessed Aug. 31, 2020. [Online]. http://www.teledynemarine.com/ProViewer?BrandID=3

[24] R. Cerqueira, T. Trocoli, J. Albiez, and L. Oliveira, “A rasterized ray-tracer pipeline for real-time, multi-device sonar simulation,” *ArXiv200103539 Cs Eess*, Jan. 2020, Accessed: May 11, 2020. [Online]. Available: http://arxiv.org/abs/ 2001.03539

[25] “Ocean & marine systems,” *Leidos*. [Online]. https://www.leidos.com/products/ ocean-marine (accessed Aug. 24, 2020).

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California