FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Synthesis of acoustic images of underwater targets - Síntese de Imagens Acústicas de Alvos Subaquáticos

Duarte Nuno Reimão Borges Lopes Silva

FOR JURY EVALUATION

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Nuno Alexandre Lopes Moreira da Cruz

Second Supervisor: José Carlos dos Santos Alves

March 2, 2015

Abstract

This thesis describes a software tool that simulates data generated by different sonars during AUV (Autonomous Underwater Vehicle) surveys. Analyzing sonar data can be a challenging task. Large amounts of information are easily obtained from a sonar mission and manually interpreting it in order to find specific targets can be difficult. Searching for an automated process for this task could be an appropriate solution. After creating those automated processes they need to be tested and validated with controlled sonar data, but obtaining it from real sonars is expensive. Generating it from a computer would be cheaper and more convenient. That is the main scope of this thesis, synthesizing sonar images of underwater targets through computer simulation.

There are already some tools for this (e.g. SonarSim, SST) but not all are freely available for everyone. Another use for this software is to train human operators in sonar mission planning, as it allows for evaluating different combinations of vehicle trajectories and sonar configurations. MATLAB was the chosen tool to develop FSS (FEUP Sonar Simulator) because of its vast supply of simulation tools that help achieve the intended purpose.

In order to use FSS, some inputs have to be defined. These are separated in three categories, where the first one is related to the sonars in the vehicle, the second one deals with the trajectory of the AUV and the last one is related to the environment. The software processes all the input data and then returns bathymetry information and signal strength, which, when combined, make up the synthesized sonar image. In order to get this response from the software several methods were used. The main ones are the kinematic model which allows us to calculate the AUV's trajectory through the points given by the user, the intersection of sound waves with bottom and targets which provides the bathymetry information and the sonar equation which is used to calculate the return signal strength.

This is a work in progress and there is still some room for improvements, several approximations were made and some factors were not considered. Special attention was given to the modularity of this system, future expansion and improvements were always an important part so the software is split in several modules that can be altered independently of each other.

Agradecimentos

Agradeço à minha família por tudo. Nunca teria chegado até aqui sem o apoio e compreensão que me foram dados.

Ao Professor Nuno Cruz, agradeço a orientação neste trabalho, encaminhou-me para a rota certa e motivou-me nos momentos em que o trabalho estava a dar mais luta.

Agradeço também a todos os que conheci durante o meu percurso académico e que de uma maneira ou de outra ajudaram a que estes últimos anos fossem dos melhores que tenho recordação. Em especial ao Rui Feio por me ter acompanhado desde o inicio, nos momentos de lazer e também nos momentos de estudo. Ao Hugo Carvalho e ao Diogo Queirós por se terem juntado a mim na ultima fase do projeto e me terem motivado a não desistir. Obrigado também ao João Machado e ao Miguel Ribeiro, a vossa experiência e conselhos foram fundamentais para o meu trabalho.

Por fim agradeço aos meus amigos de longa data, Tiago Torres, Nuno Dias, Daniel Santos, Tiago Teixeira, André Sousa, Filipa Carregal, Sofia Durão, Joana Loureiro, Vânia Gonçalves, Patrícia Ferreira, Marisa Ferreira, Marina Pereira, Cátia Teixeira, Mariana Rocha e Rita Mota. Obrigado pela companhia e amizade que dura há tantos anos.

Duarte Silva

“Everyone thinks of changing the world,
but no one thinks of changing himself.”

Leo Tolstoy

Contents

List of Figures

Abbreviations and Symbols

Abbreviations list:

AUV	Autonomous Underwater Vehicle
FSS	Feup Sonar Simulator
ROV	Remotely Operated Vehicles
Sonar	SOund Navigation And Ranging

Chapter 1

Introduction

1.1 Motivation

In underwater surveys, as well as in several other fields (Automobile Industry, Home Automation, Video Surveillance, etc.), automation is becoming more and more prevalent [5]. Therefore, Autonomous Underwater Vehicles (AUVs) are becoming more widely used in tasks that were formerly performed by Remotely Operated Vehicles (ROVs). Some of the advantages of an AUV over a ROV are the fact that two or three people are enough to deploy them and once deployed they perform their task with almost no human input, they can travel at greater speeds and cover greater distances[5].

Although there have been considerable developments in data recollection technology and automated processes have been created, data analysis and treatment is still a hard and lengthy manual process. Very few algorithms have been developed to automatically analyze that data, such as the work done by Wei et al [6]. To conceive and validate those algorithms a controlled dataset of sonar data is required. Retrieving this dataset from real surveys would become expensive.

1.2 Context

This dissertation is to be developed under the Robotics and Intelligent Systems Unit of INESC-TEC.

The main objective of this work is to produce a software able to synthesize acoustic images of underwater targets that are close to real sonar images. This software is to be used to test and validate methods for automatic detection of targets, so that this task can be performed in a faster and cheaper way.

To better understand this work, some background could be useful. In the next sections (1.2.1 and 1.2.2) some basis of the Sonar operation are given.

1.2.1 Sonar Theory

The objective of this thesis builds upon the underwater Sonar (SOund Navigation And Ranging) technology. Due to that fact, background information of its basic principle of operation and different types is provided in the following subsections.

There are two main types of sonar:

- Passive sonar: only listens to sound emitted by objects in the water.
- Active sonar: produces a sound pulse and waits for the echo to return.

Only the latter is relevant to this document. Assessing how long it takes for the echo to return and measuring the strength of the returned signal allows to make inferences about the distance to the target and its composition.

1.2.1.1 Basic Principle

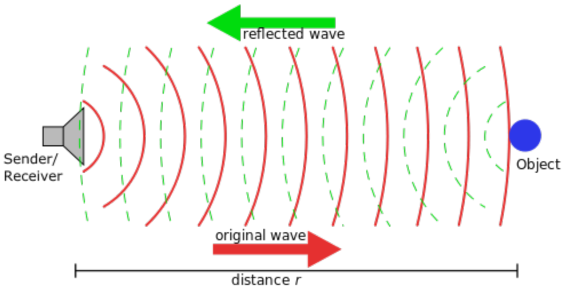


Figure 1.1: Sonar basic principle

Although it was used in acoustic location in the air, before the introduction of Radar, nowadays it is mostly used in underwater environments since sound is the only way of transmitting energy in a water environment for a considerable distance [7].

A source sends a ping (short pulse of sound) and then waits for the return signal. Knowing the local speed of sound (knowing that the speed of sound in the water is, approximately, 1500 m/s [1]) and the time it took the sound to return it is easy to determine the distance to the object:

$$range = (\frac{1}{2}) \times speedofsound \times echotime \tag{1.1}$$

A sound propagates spherically, from its source, into all directions spreading its energy. The further it goes the more energy it loses, this phenomenon is called spreading loss. The signal is also affected by some attenuation that its called absorption loss. These two losses form the transmission loss (TL). When the sound pulse hits an object some energy is transmitted into the object, varies depending on the object's material. the amount of energy per unit are that is returned in the direction of the projector is called the backscattering strength of the object [1]. From the moment a ping is produced to the moment it is received several interferences affect the signal (boats, animals, etc.). The sum of this interferences is called noise level (NL). In order to measure the strength of the echo return, called signal excess (SE), an equation was deduced:

$$SE = SL - 2TL + BS - NL + TA \tag{1.2}$$

With SL being the signal level as it leaves the source of the ping, BS the backscattering strength and TA the target area. The target area is the area affected by the sound wave. All the values above are in decibels (dB).

1.2.2 Sonar Systems

1.2.2.1 Single Beam Echo Sounder

This specific type of sonar is used mainly for bathymetry and fisheries-resource surveys [8]. While it is a simple and inexpensive system, it has some problems. Having only one transducer, each ping will propagate in every direction. Because of that fact it cannot be assumed that the first echo to arrive is from the point directly beneath our transducer as shown in figure 1.2.

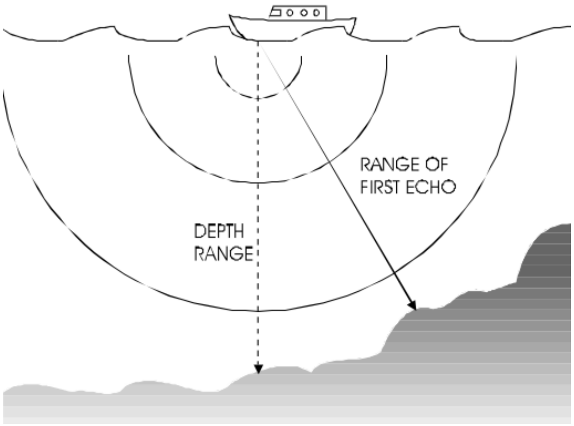


Figure 1.2: Diagram showing how a Single Beam Echo Sounder works. Figure from [1]

Single beam echo sounders can deal with this problem, making its beam narrower, but not completely because narrowing the beam requires larger transducers, that are more expensive. Another problem with this echo sounder is that it can only produce one reading at a time, as it needs to wait for a ping to return before sending one more. This makes the process slow which in turn increases the cost of the survey.

1.2.2.2 Sidescan Sonar

Contrary to the two previous sonars this type is not used for bathymetry but for the acquisition of sea floor composition information. As previously explained, different materials have different effects in a sound pulse, the sidescan sonar takes advantage of this characteristic. The combination of bathymetry information with the sidescan sonar information provides a good image of the oceanic bottom.

This sonar has the same hardware as the multibeam sonar but with a different configuration (the arrays are positioned on the sides of the survey vessel, as shown in figure 1.3) and the data is processed in a different way.

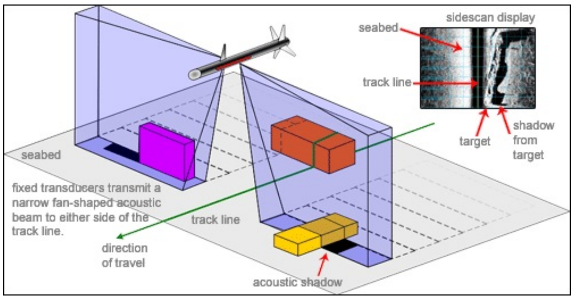


Figure 1.3: Sidescan Sonar figure from [2]

1.2.2.3 MultiBeam Echo Sounder

The multibeam echo sounder solves the problems that exist in the single beam. This sonar projects several beams at the same time covering various locations, that are normally a continuous area perpendicular to the survey vessel trajectory, as seen in figure 3.9.

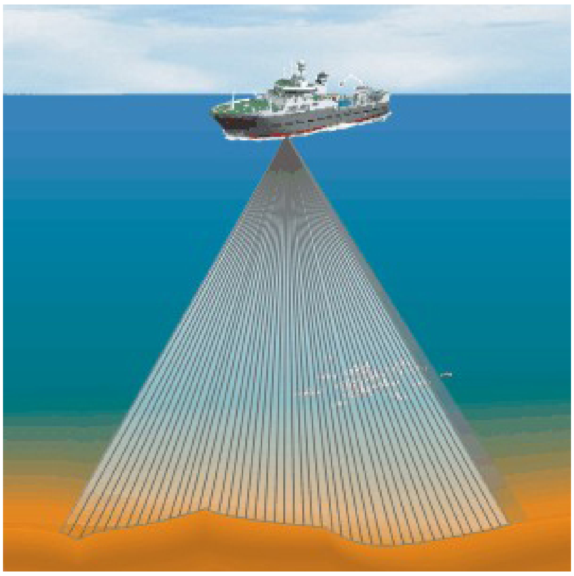


Figure 1.4: MultiBeam Echo Sounder figure from [3]

This technology is possible by combining several transducers in two perpendicular arrays in a Mills Cross arrangement, as done by SeaBeam Instruments [1].

1.3 Objectives

The objectives of this project are:

- Investigate the acoustic propagation phenomena on predefined targets.
- Develop a software that is able to synthesize sonar images.
- Allow the user to define the sea floor characteristics.
- Allow the user to place a target and define its characteristics.
- Be able to define the route and speed of the AUV.
- Validate the proposed solution through data acquisition in the field.

1.4 Dissertation Structure

Besides the introduction, this dissertation contains four more chapters.

In chapter 2 a few tools that are pertinent to this subject are presented.

Chapter 3 explains all the inputs used in this project as well as the methods, from data acquisition and processing to the final stage of synthesizing sonar images.

Chapter 4 presents the results obtained through the applied methods and also a critical analysis of those results.

Chapter 5 presents the conclusions obtained from the work done for this project and also gives some ideas on the future work that can be done in order to improve what has been done.

Chapter 2

State of the Art

This section details some information about the state of the art.

2.1 SonarSim

SonarSim is an Irish company that has developed a software that simulates an underwater sonar. This product is called PHYSicS (Portable Hydrographic Survey Simulator) [9]. It simulates Sidescan and Multibeam sonar in several types of operational scenarios.

Another product developed by SonarSim is ECHO (E-learning on the Cloud for Hydrographic Operations) [9]. This web portal gives remote access to Hydrographic survey simulation servers as well as integration of the National archive survey datasets with PHYSicS.

The information regarding this software is very limited, there is only a summarized text on the company website.

2.2 SST

Developed by Robert Goddard and sponsored by the US Navy, the Sonar Simulation Toolset (SST) is a computer program that allows the simulation of an oceanic environment and the use of survey vessels in that environment and it is used to design new sonar systems, test existing sonars, predict performances, develop tactics, train operators, plan experiments and interpret measurements. This project started in 1989 and is still being improved regularly. SST is only available to companies that have contracts with the United States Department of Defense. Still, in [10] a vast and detailed description of the code and methods is provided.

In Goddard et al [10] a description of the following models is provided:

- The Eigenray Model - describes how sound is propagated and transformed as it travels in the ocean.
- Ocean Model - this model contains the properties of the ocean, this properties are used as inputs for the eigenray model.

- Sonar and Source Models - describes any object that can produce and/or receive sound.
- Direct Sound Propagation Models - this model inputs are all the sounds emitted by a source and the output is the portion of those signals that does not scatter on its way to the receiver.
- Target Echo Model - the output of this model is a signal that has been received, altered and re-transmitted by a target.

2.3 MB-System

MB-System^{TM} is an open source collection of tools that is able to process several format types of swath mapping sonar data and is usually used with the support of the Generic Mapping Tools (GMT). It was created in 1993 at the Lamont-Doherty Earth Observatory of Columbia University and has now the collaboration of the Monterey Bay Aquarium Research Institute. MB-System^{TM} does not support every possible data type but it accepts the majority of the multibeam data formats. This toolset is still in development to keep adding support to more variations of sonar data formats [11].

In order to achieve satisfying results, a series of steps has to be taken. Considering that this system accepts different types of data formats the steps taken can be slightly different but the center core is the same. The general process starts by analyzing and organizing the data, then, if not provided by the survey ship, physical constants, for instance roll and pitch bias, are calculated from the acquired data. Afterward automated and manual editing of the data is executed in order to remove erroneous segments and the navigation data is edited to ensure continuity. Next, high quality sound speed profiles for regions in the data set are identified and, for sonar that produce sidescan data, amplitude and grazing angles tables are developed. In the end all the aforementioned physical constants, bathymetry and navigation editing changes, sound speed profiles and grazing angles tables are applied to the original data and the final data set is created[11].

2.4 GMT

GMT stands for *Generic Mapping Tools* and is a free software package (about 80 command-line tools) that is used to manipulate columns of tabular data, time-series and gridded data sets (including filtering, trend fitting, gridding, projecting), and display these data in a variety of forms such as x-y plots to maps and color, perspective, and shaded-relief illustrations. The first version of *GMT* was released in 1988, it is mostly used by Earth or ocean scientists but there are applications in other areas such as medical research, engineering or physics.

2.5 Conclusions

From the sections above it is possible to observe that there is no available open-source software that simulates the sonar data and then displays it as a real sonar would. MB-System uses data created externally and GMT is only used to manipulate data. An open-source tool that could compete with Sonarsim and SST would be extremely useful.

Chapter 3

Methodology

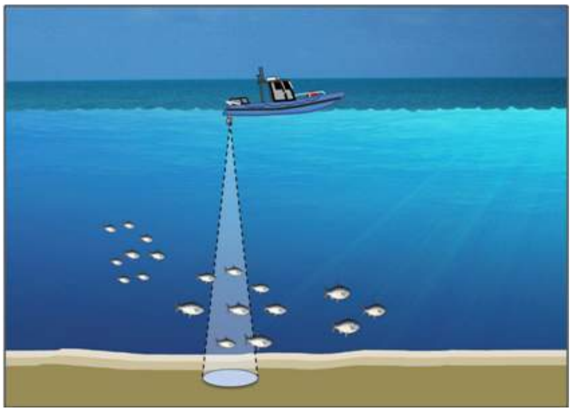
This chapter explains most of the work done for this Thesis and it is divided in three sections. The first one offers some background, providing a more practical approach to the different types of sonar, from a simulation point of view. The second section defines the inputs used to reach the desired solution. And the final section describes the methods used to successfully arrive at the simulated sonar images through the use of MATLAB.

3.1 Approach

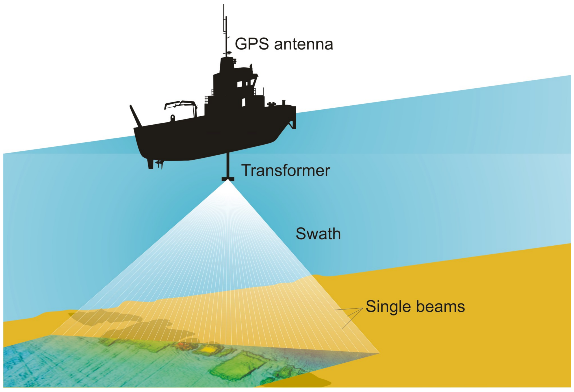
Chapter 1 gives a brief explanation of the sonar theory of operation but for this project it is not enough to explain how the sonar works, it is also important to explain how everything is going to be handled during the simulation. This section aims to clarify how some aspects of the sonar survey were translated to this simulator, so that it is easier to understand the methods used.

3.1.1 Sonars

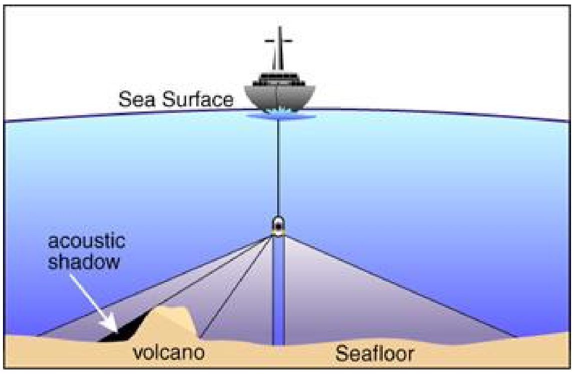
As was explained earlier, each type of sonar has a different way of collecting data, this means that they have different requirements in terms of simulation. Figure 3.1 is a compilation of the methods of operation for each sonar type considered in this work.



(a) Echosounder operation from [12]



(b) Multibeam operation from [13]



(c) Sidescan operation from [14]

Figure 3.1: Different Sonar operation patterns

An echosounder has only one beam (figure 3.1a) in the form of a cone, that can have different angles of aperture. For this equipment, only the first return ping is considered but several arrive at the ship. To accurately simulate this sonar, the beam was divided in several different beams and all the information from those beams is stored during the data acquisition phase but only the first return will be considered in the image synthesis. A 2D image results from this data, it is a depth in terms of distance traveled.

The multibeam has several beams (figure 3.1b), where all of them collect data. In the simulation, several beams with a variable angle between each other are created and and all the information collected is used to construct the simulation image, which is a 3D image.

Finally, a sidescan sonar has two beams one pointed at each side (Figure 3.1c) with a variable angle between beams and a variable aperture. To simulate this sonar the two beams are treated independently and the information stored in separate variables. Each beam is split into several beams, like in the echosounder pattern, but the information is treated differently.

3.1.2 Rotations

A vehicle, either a ship or an AUV, in an aquatic environment has its position defined not only by the coordinates of the center of mass but also by a set of rotation angles that define its orientation. As shown in picture 3.2, the rotation of the x-axis is called roll (α), the rotation of y-axis is called pitch (β) and the rotation of the z-axis is called yaw (γ).

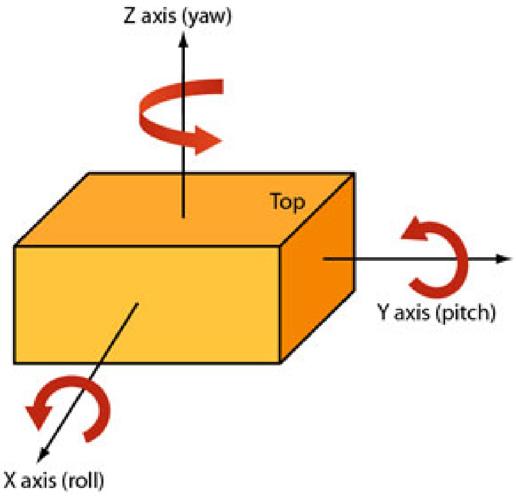


Figure 3.2: Visual representation of the rotations around the X, Y and Z axis. Figure from [[4]]

These rotations are important because they will affect the position of the beams, therefore the measurements will also be influenced. To easily apply the rotations three matrices were used, called *rotation matrices*, one for each axis.

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

$$R_y = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix}$$

$$R_z = \begin{bmatrix} \cos(\gamma) & -sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

And if a point P (x,y,z) was rotated by (α, β, γ) to a point P' (x',y',z') the rotation would be applied by the following equation:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R_z \times R_y \times R_z \times \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{3.1}$$

3.2 Inputs

In order to run the simulation and retrieve the desired information a set of inputs has to be defined. A proper assignment of those components is vital to reach the desired outputs. The following command has to be typed into MATLAB to start the process:

$$fss(Environment, Vehicle, Sonar, BackscateringStrength); \tag{3.2}$$

The next four subsections explain each one of those inputs.

3.2.1 Environment

The *Environment* input is, perhaps, the most important of all the inputs. It contains all the data related to the oceanic environment, with that being bathymetry information and terrain composition information. It is organized in a $n \times m \times 2$ matrix, bathymetry data in the first level and terrain composition information in the second level.

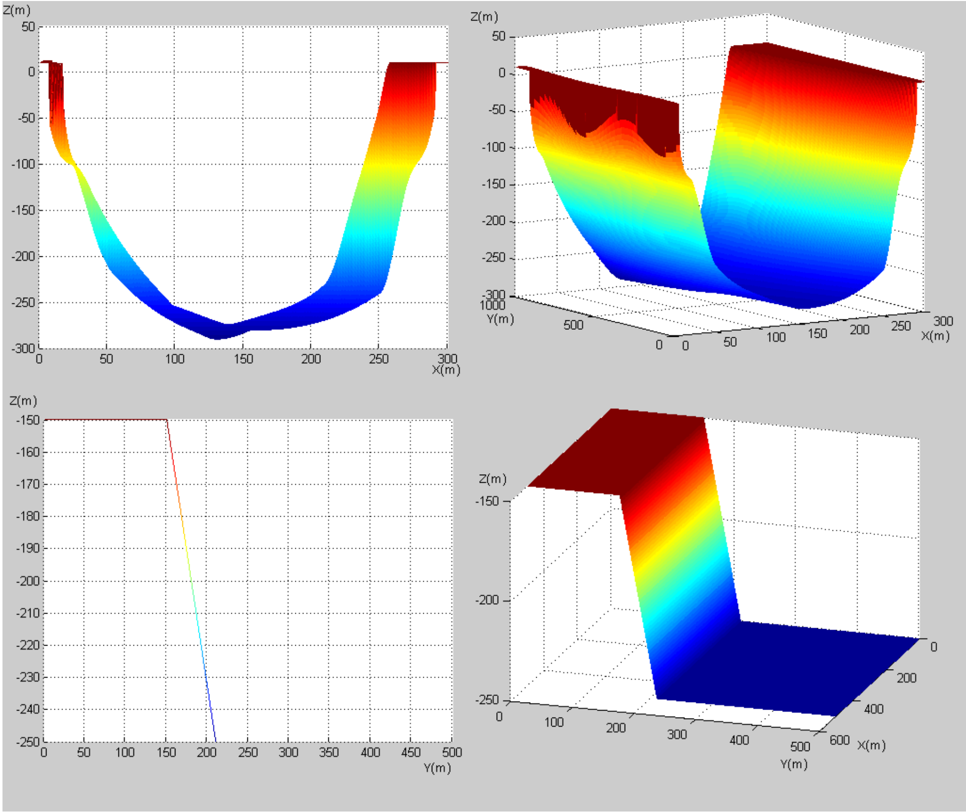


Figure 3.3: Examples of Heightmaps

For the bathymetry information, each point can take any value below zero since it represents the oceanic floor height for that point. Terrain composition is represented by an integer greater than zero, where each value represents a different terrain type. That information is in a lookup table that will be explained in section 3.2.4.

3.2.2 Vehicle

Vehicle represents the position or positions of the AUV or Ship through time. In case of it being static, *Vehicle* is a seven element array that contains position coordinates (x,y,z), rotation angles (roll, pitch, yaw) and the instant at which that pose occurs.

$$Vehicle = \begin{bmatrix} x & y & z & roll & pitch & yaw & time \end{bmatrix}$$

In case of a moving vehicle, the *Vehicle* input is a $n \times 7$ matrix with each line representing a different position.

$$Vehicle = \begin{bmatrix} x_1 & y_1 & z_1 & roll_1 & pitch_1 & yaw_1 & t_1 \\ x_2 & y_2 & z_2 & roll_2 & pitch_2 & yaw_2 & t_2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x_n & y_n & z_n & roll_n & pitch_n & yaw_n & t_n \end{bmatrix}$$

This information does not need to be manually introduced by the user, it can come from the output of a dynamic simulation of a moving vehicle, following a specific trajectory or even from the log file of a vehicle.

3.2.3 Sonar

This input is a four or five cell array that determines the type of sonar used in the simulation.

$$\begin{aligned} Sonar &= \begin{bmatrix} echosounder & SignalLevel & Frequency & \sigma \end{bmatrix} \\ Sonar &= \begin{bmatrix} sidescan & SignalLevel & Frequency & \sigma & \zeta \end{bmatrix} \\ Sonar &= \begin{bmatrix} multibeam & SignalLevel & Frequency & \zeta \end{bmatrix} \end{aligned}$$

The first cell contains sonar type information (echosounder, sidescan or multibeam), the second has the level of the signal in dB (decibels) and the third the frequency of the signal in KHz. The following cells vary according to sonar type. For an echosounder there is only one more cell and it contains the beam aperture angle (σ), in degrees. In the case of a sidescan sonar, there are two more cells, the first one contains beam aperture angle (σ) and the second one the angle between beams (ζ), both in degrees. Multibeam as only one more cell and it contains the angle between beams (ζ) in degrees.

3.2.4 Backscattering Strength

This input is a lookup table that contains the terrain information. The first column contains the terrain type (sand, rock, etc.) and the second one contains a value, in dB, of the backscattering strength of those materials. The value in the second level of *Environment* (as explained in section 3.2.1) represents the line, from the lookup table, from which this information is extracted. This information could be directly in the *Environment* input but being in a lookup table makes it easier to change values for a determined terrain composition, without having to change every value of the *Environment*.

3.3 Methods

This section explains how the methods are processed and how they interact with each other. Figure 3.4 gives an overview of the software architecture. The inputs (Environment, Vehicle, Sonar and Backscattering Strength as explained in section 3.2) are passed as arguments to FSS and then, based on the type of sonar chosen, FSS calls one of the three methods (Echo, Sidescan, Multibeam) after this step the return signal strength is calculated and finally PrintRes is called with some of the original Inputs and the outputs of the previous methods as arguments.

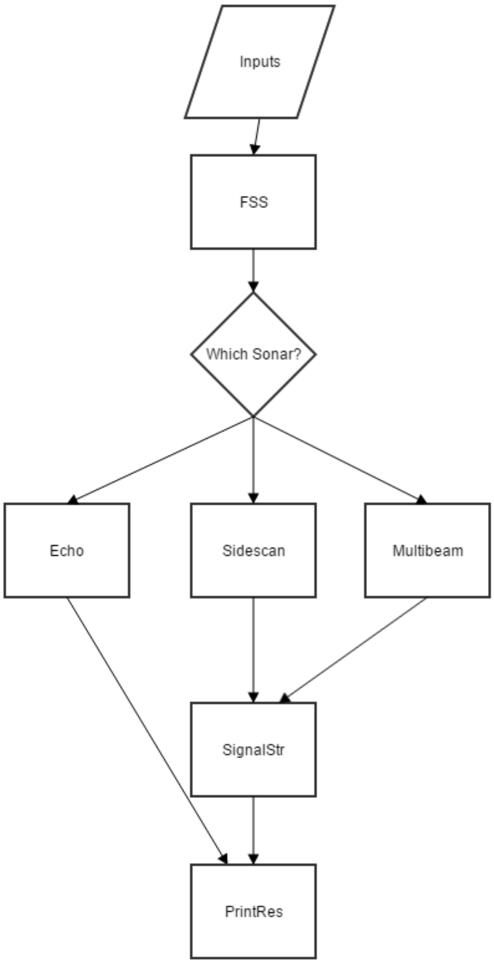


Figure 3.4: Functions overview

3.3.1 FSS

FSS is the main function of the project, all other methods are called within this one. It organizes the data from the inputs and then calculates the vehicle trajectory. Afterwards, the appropriate data acquisition function is called and finally *PrintRes* is called to obtain the synthesized image.

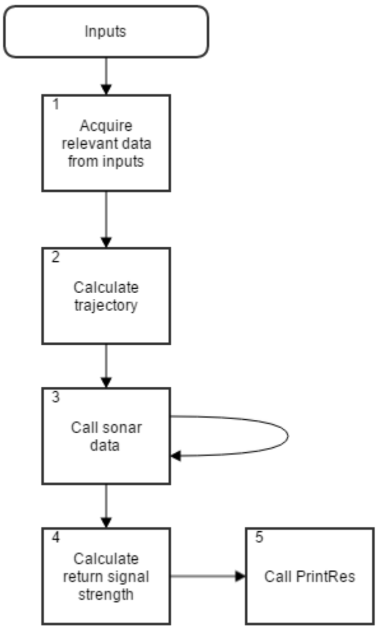


Figure 3.5: FSS workflow diagram

The first step in this process is to take all the input data (*Environment*, *Vehicle*, *Sonar*, *Backscatering Strength*) and extract the relevant information. Next the path of the vehicle has to be calculated, which requires an analysis of the input *Vehicle*. If there is only one set of coordinates it means that the vehicle is static, and in that case this step is over. If there is more than one set of coordinates, the straight line between each two consecutive points is calculated through the following set of equations:

$$\begin{aligned} x_n &= x_0 + t \times (x_1 - x_0) \\ y_n &= y_0 + t \times (y_1 - y_0) \\ z_n &= z_0 + t \times (z_1 - y_0) \end{aligned} \tag{3.3}$$

With the variable t increment from 0 to 1, the increment (inc) depends on the time set by the user for those two points.

$$\begin{aligned} \Delta t &= t_1 - t_0 \\ inc &= \tfrac{1}{2 \times \Delta t} \end{aligned} \tag{3.4}$$

This way two sonar pulses are sent per second. Although at this point the frequency of the pulses is not controlled through user inputs, that feature can be added in later versions. Between each increment of equation 3.3 the two following steps are executed. The first one is to call the bathymetry acquisition function (*Echo*, *Sidescan* or *Multibeam*), in order to choose which function should be called the *Sonar* input is evaluated. And the second one is to calculate the return signal strength. Finally, the results are printed with function *PrintRes*.

Given the following input to *Vehicle*:

$$Vehicle = \begin{bmatrix} 50 & 200 & -50 & 0 & 0 & 0 & 0 \\ 100 & 200 & -50 & 0 & 0 & 0 & 25 \\ 100 & 350 & -50 & 0 & 0 & 0 & 100 \\ 150 & 400 & -50 & 0 & 0 & 0 & 130 \end{bmatrix}$$

The trajectory followed would be as seen in figure 3.6

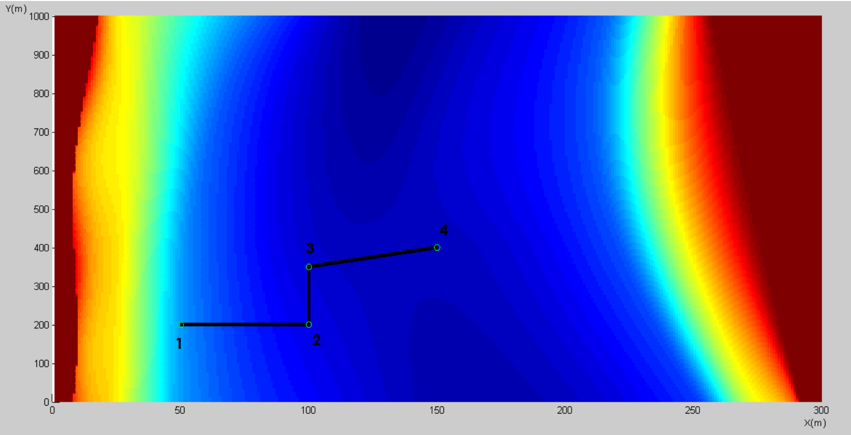


Figure 3.6: Trajectory example

3.3.2 Data Acquisition

This section presents the functions responsible for data collection. There are three functions that retrieve sonar range information, one for each sonar type (*Echo*, *Sidescan*, *Multibeam*), and one that calculates the return signal strength based on the sonar range data (*SignalStr*). *Sidescan* and *Multibeam* have similar processing steps, the differences happen only because of the different nature of the two types of sonar. *Echo* was originally similar to the other two but now has a more efficient algorithm implemented.

At this point, besides the x, y, z coordinates, the euler angles, $roll(\alpha)$, $pitch(\beta)$ and $yaw(\gamma)$, used to describe the vehicle orientation have also to be taken into account since they will influence the outcome of the following functions.

3.3.2.1 Echo

Echo is the function responsible for retrieving bathymetry data for an echosounder sonar, the output is a matrix as seen in equation 3.5. The first line contains the information about depth, the second and third ones contain, respectively, the x and y positions and the fourth one contains the time it took the signal to return to the vehicle. This time is obtained through the use of equation 1.1, where $t_{rn} = \frac{d_n}{c}$, c is the local speed of sound).

$$depth = \begin{bmatrix} d_1 & d_2 & d_3 & \dots & d_n \\ x_1 & x_2 & x_3 & \dots & x_n \\ y_1 & y_2 & y_3 & \dots & y_n \\ t_{r1} & t_{r2} & t_{r3} & \dots & t_{rn} \end{bmatrix} \quad (3.5)$$

Figure 3.7 showcases a block diagram of the processes involved in this function.

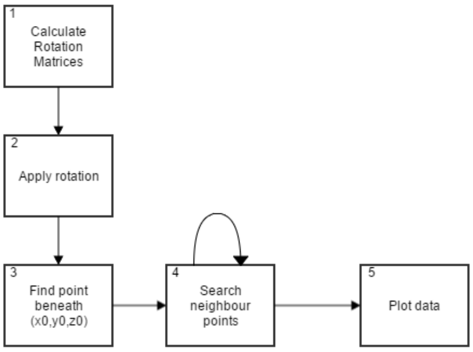
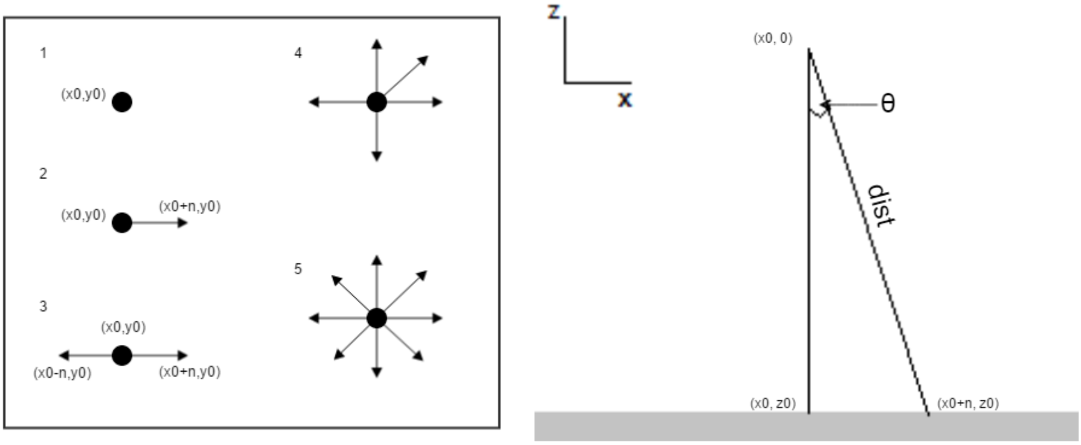


Figure 3.7: Block diagram of the steps involved in the function Echo

The first step is to calculate the rotation matrices and then finding the *Environment* value in x_0 and y_0 .

$$z_0 = Environment(x_0, y_0) \quad (3.6)$$

Afterwards the rotation is applied. This yields a new x, y and z, if α and/or β are *not zero*. At this point starts the search of the neighboring points in a star shape as seen in figure 3.8a.



(a) Star shape pattern - XY plane

(b) XZ plane

Figure 3.8: Search pattern

Starting from the original point and increasing one unit at a time, values of sonar range are acquired until the angle between the original line and the new line (θ), as seen in figure 3.8b, is equal to or greater than the sonar aperture angle. When all the eight sides have been inspected this step ends. The final step is to print all the acquired points over the *Environment* map.

3.3.2.2 MultiBeam

The algorithm starts by filling an array with the angles of interest (θ) for the multibeam sonar, in this case -60$^{o} \leq \theta \leq$+60^{o} with a β increment (β is the angle between beams that is defined in the *sonar* input, as explained in section 3.2.3). Each of this angles defines a sonar beam (A1, A2, A3, etc.) as seen in figure 3.9. Then calculates the rotation matrices. At this point a loop is started, for each angle θ, a variable that represents the depth (*dpos*) is decreased and a new x is calculated.

$$n_x = dpos \times \tan(\theta) \tag{3.7}$$

After that the rotation is applied, and new values of n_x (n_{x2}), y (n_y) and z (n_z) are obtained. Now the following comparison is made:

$$n_z \leq Environment(n_{x2},n_y) \tag{3.8}$$

If it returns true, the loop ends and starts again for another angle. If it returns false, z is decremented and the process is repeated.

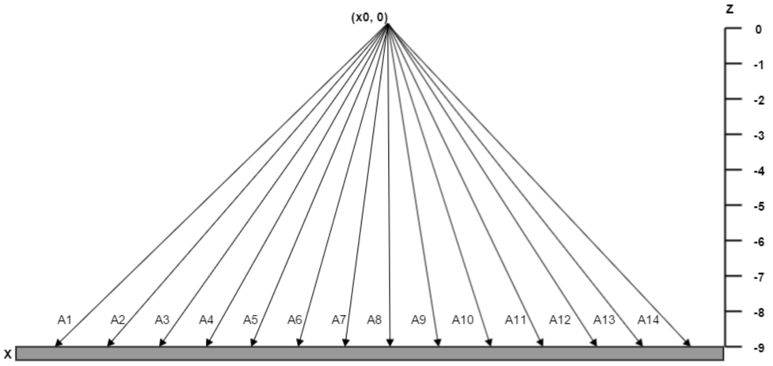


Figure 3.9: Multibeam Sonar example

3.3.2.3 SideScan

This method is similar to the one described in subsection 3.3.2.2 but, instead of several beams, there are only two as seen in figure 3.10. Despite that difference, the only noticeable change is in the first step of the process. Instead of one array being created, there are two arrays, one for each beam. Each beam is then processed separately and saved in different variables.

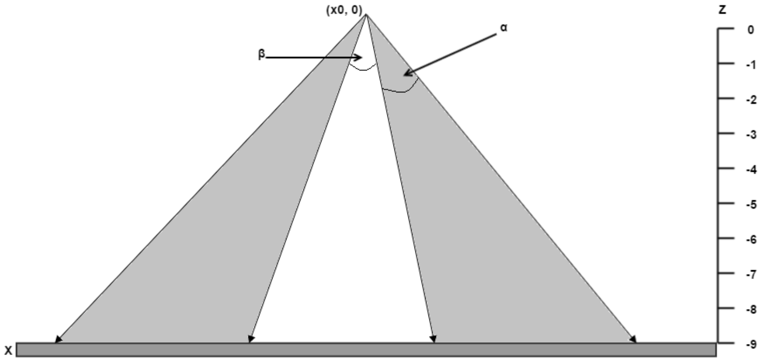


Figure 3.10: Sidescan Sonar example

3.3.2.4 SignalStr

$$ReturnSignal = SignalStr(Environment, depth, BackscatteringStrength, sonar) \tag{3.9}$$

SignalStr is a direct application of the Sonar equation (equation 1.2), but with two modifications. The first one is a simplification, at this point the *TA* (target area) is not taken into account and the second one is the addition instead of the subtraction of the *NL* (noise level) because what we are measuring is the strength of the signal that arrives back at the vehicle. The noise level is, for now, constant at 60 dB. This leaves three parcels to be determined, Transmission Loss (*TL*), Backscattering Strength (*BS*) and Signal Level (*SL*). Signal Level is part of the *Sonar* input and Backscattering Strength is an input. The only thing that has to be calculated is the Transmission Loss, and this was done with the equations from the work done by François and Garrisson [15]

3.3.3 Image Synthesis

3.3.3.1 PrintRes

This last method is also the final stage of the process. It is used to print the information gathered in the previous methods but at this point that data is not ready to be displayed, some treatment has to be done. This step was only concluded for the echosounder and multibeam data. For the sidescan, this feature was not fully implemented.

For the echosounder sonar, a 2D print is made. For each reading the minimum return time value is found and the distance corresponding to that time is printed over the Environment map corresponding to that area.

For the multibeam things are not as simple. Through the observation of figure 3.9, it is possible to see that beams A1 and A8 are of different lengths, but the terrain being surveyed is flat. This will cause a distortion in the printed image, so before it is printed a correction has to be applied.

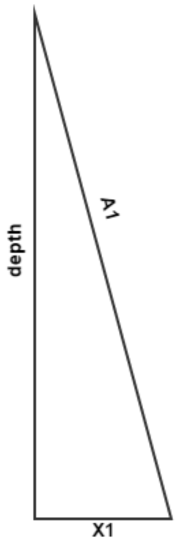


Figure 3.11: Visual representation of the use of pytagoras theorem for the calculation of depth.

To correct this a simple Pythagorean theorem has to be a applied. Figure 3.11 represents the concept, *A1* and *x1* are the information available and *depth* is the correct value that will be printed, therefore:

$$depth = \sqrt{A1^2 - x1^2} \tag{3.10}$$

From this point the bathymetry map can be plotted. Next, *SignalStr* is called again to recalculate the new values of the return signal strength. Finally, the signal strength map is plotted.

Chapter 4

Results

In order to validate the proposed solution some tests were performed. This chapter presents the tests made as well as the results from those tests followed by some comments.

4.1 EchoSounder

4.1.1 Environment 1

Figure 4.1 displays the *Environment* used in this test, it is a flat terrain with a half cylinder elevation in it. This will be used to test how the echosounder sonar reacts in this situation.

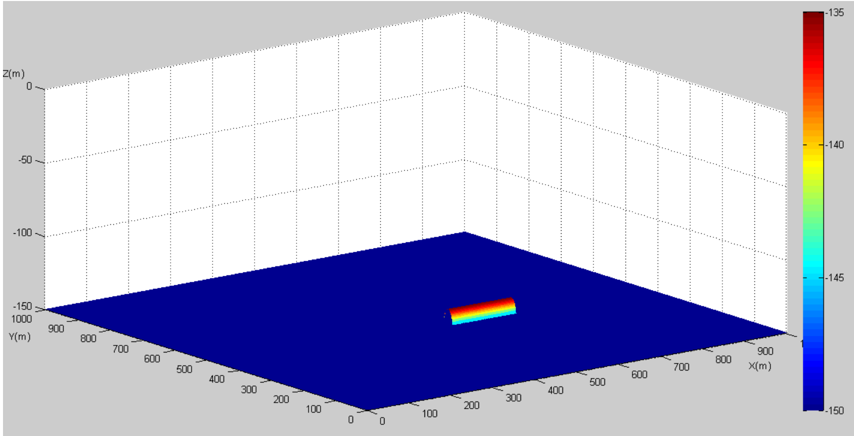


Figure 4.1: Environment for the first test with an echosounder

The following *Vehicle* input was used:

$$Vehicle_1 = \begin{bmatrix} 600 & 300 & -50 & 0 & 0 & 0 & 0 \\ 600 & 500 & -50 & 0 & 0 & 0 & 100 \end{bmatrix}$$

The first output is the trajectory made by the vehicle plotted over the environment, as seen in picture 4.2. The green circle represents the starting point and the black marks the path of the vehicle.

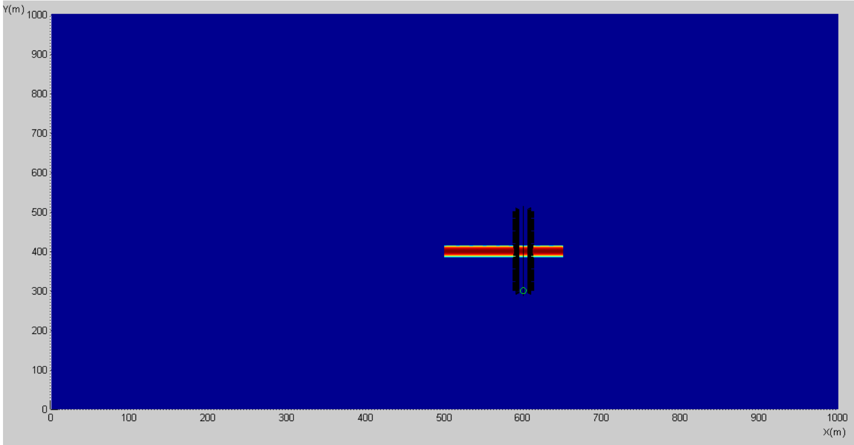


Figure 4.2: Vehicle path

The last output is the bathymetry measurements, as it is possible to observe on figure 4.3. Each red circle represents a measurement and the blue line is the real value of the terrain.

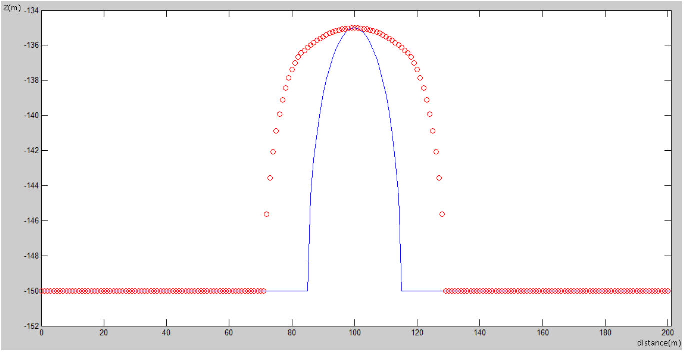


Figure 4.3: Passage at a depth of -50 and beam aperture angle of 5º

By observing figure 4.3 it is possible to see that while the sonar is over a flat piece of land the measurement and the real value match, however the measurement starts to climb before the real ascent and only declines a while after the real descent. This error is to be expected, an echosounder sonar will advance the ascents and delay the descents.

The explanation to this is in figure 1.2, it is possible to see that when approaching an elevation in the oceanic floor the first return echo will not be the one directly beneath the vehicle but from the elevated area, resulting in the difference visible in the simulation image. On the other hand, when passing over a descent the first return will be from the highest point resulting in a delay of the descent on the simulated image.

This error in the measure will be affected by the radius of the base of the cone. The user controlled factors that directly influence the radius are the beam aperture angle and the vehicle depth. So the next tests will compare the same path from figure 4.2 firstly with different beam aperture angles and then with different vehicle depths.

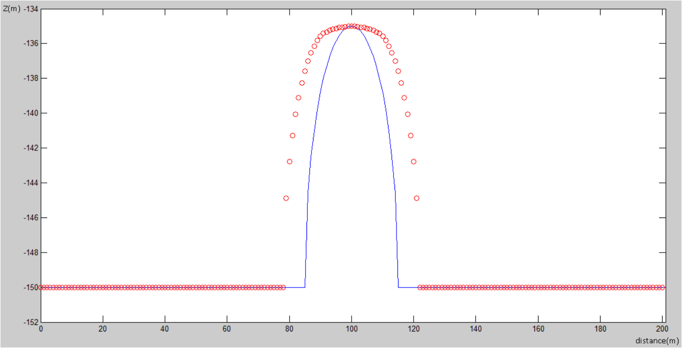


Figure 4.4: Passage at a depth of -50 and beam aperture angle of 2.5º

It is possible to see, in figure 4.4, by comparing the two graphics that decreasing the aperture angle will also decrease the survey error.

By making a comparison between the original survey (figure 4.3) and a survey with a smaller angle (figure 4.4) it is possible to see that error was reduced throughout all the protuberance but a greater influence is noticed during the rise and descent.

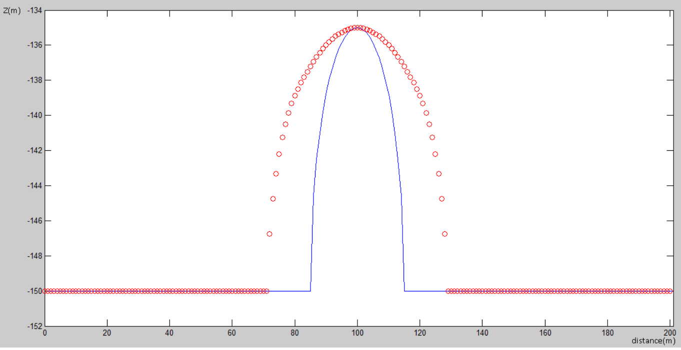


Figure 4.5: Passage at a depth of -100 and 5º beam aperture angle

Comparing the original survey with a survey at a lower depth (figure 4.5) the effect is contrary to the previous. A greater influence is noticed at the top of the protuberance and nearly none at the early stages of rising and late stages of falling.

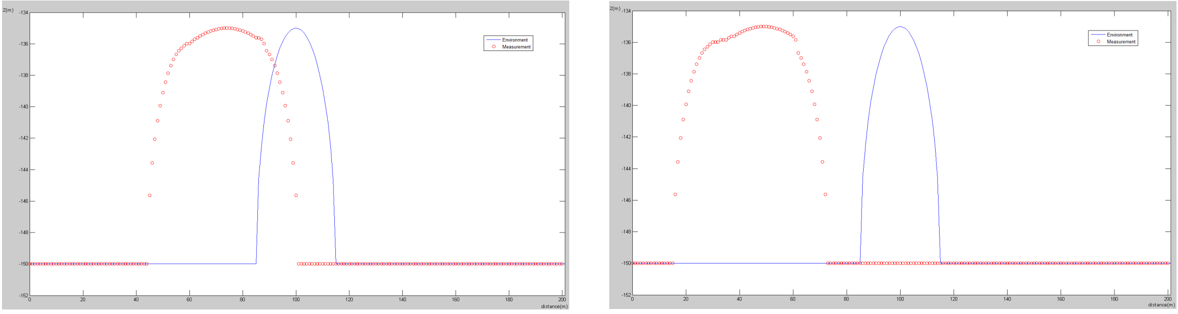
Finally, the last test will run the same trajectory but with two different roll (α) angles. So, the *Vehicle* inputs are:

$$Vehicle_2 1 = \begin{bmatrix} 600 & 300 & -50 & 20 & 0 & 0 & 0 \\ 600 & 500 & -50 & 20 & 0 & 0 & 100 \end{bmatrix}$$

and

$$Vehicle_2 2 = \begin{bmatrix} 600 & 300 & -50 & 10 & 0 & 0 & 0 \\ 600 & 500 & -50 & 10 & 0 & 0 & 100 \end{bmatrix}$$

Since the vehicle is moving in the y-axis and the rotation is applied in the x-axis, it will make the sonar point ahead (depending on the rotation angle) instead of straight down. With this rotation, it is expected that the measurement be advanced of the real value. And, as it is possible to see in figure 4.6, that is exactly what happens. A greater angle (figure 4.6b) makes a greater advance.

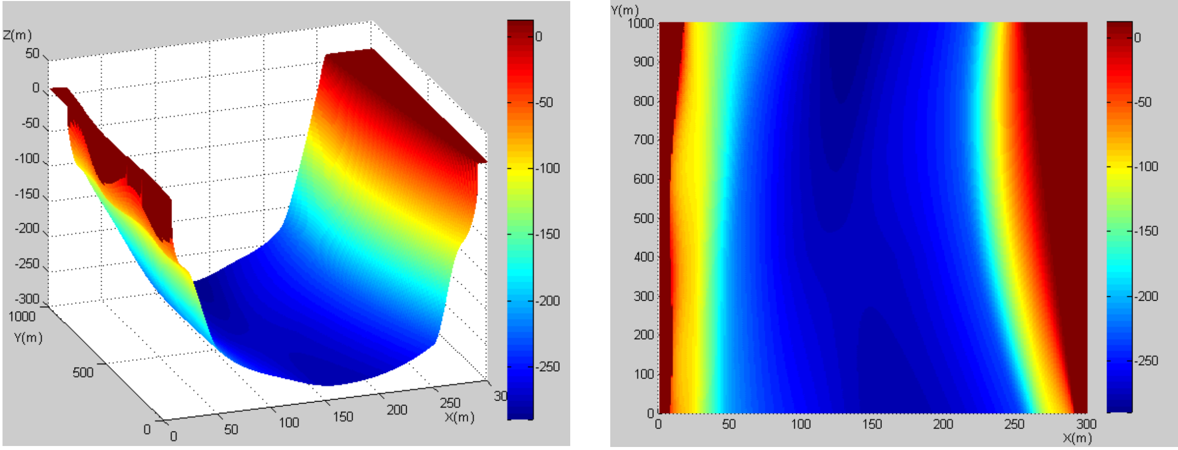


(a) Passage with α=10^{o} (b) Passage with a α=20^{o}

Figure 4.6: Survey with different roll angles

4.1.2 Environment 2

In the previous section the *Environment* input can be considered ideal, it is flat with a limited and well defined cylindrical bulge. In a real world survey (it rarely exists) nothing of the kind is normally found, so in this section a more realistic ocean floor model is going to be used, the one in figure 4.7.



(a) Environment 2 - Side view (b) Environment 2 - Over view

Figure 4.7: Environment for the second test with an echosounder

For this *Environment* a passage is going to be made along with the x-axis and another one along the y-axis. The first one has the following *Vehicle* input:

$$Vehicle_{21} = \begin{bmatrix} 50 & 750 & -20 & 0 & 0 & 0 & 0 \\ 250 & 750 & -20 & 0 & 0 & 0 & 100 \end{bmatrix}$$

Figure 4.8 shows the trajectory made over the oceanic floor. The black line is the path made by the vehicle and as it is possible to see in the input definition, it goes from the left to the right.

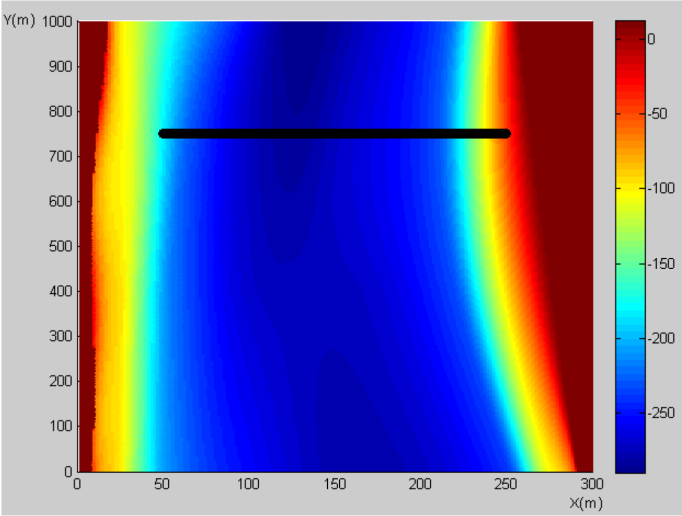
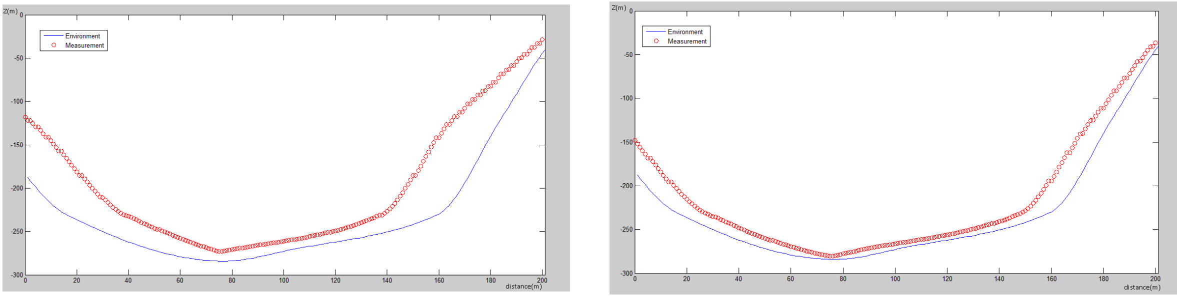


Figure 4.8: Trajectory for $Vehicle_{21}$



(a) Passage with a 5^{o} beam aperture

(b) Passage witha a 2.5^{o} beam aperture

Figure 4.9: Results from $Vehicle_{21}$ on Environment 2

Analyzing the graphs in figure 4.9 it is possible to see that, contrary to what happened in section 4.1.1, there is no point where the measurements and the actual values match. Considering that the *Environment 2* is much more irregular than *Environment 1*, this is not unexpected. Still, all the observations from the previous section still apply. There is still a delay in the the descents and an anticipation of the ascents. Also, decreasing the beam aperture angle significantly decreased the error. This test can be considered satisfactory, because although there is some error in the measurement an approximation of the seafloor depth was made.

In the next test it is going to be made a passing through the y-axis, with the following input:

$$Vehicle_{22} = \begin{bmatrix} 175 & 900 & -20 & 0 & 0 & 0 & 0 \\ 175 & 300 & -20 & 0 & 0 & 0 & 300 \end{bmatrix}$$

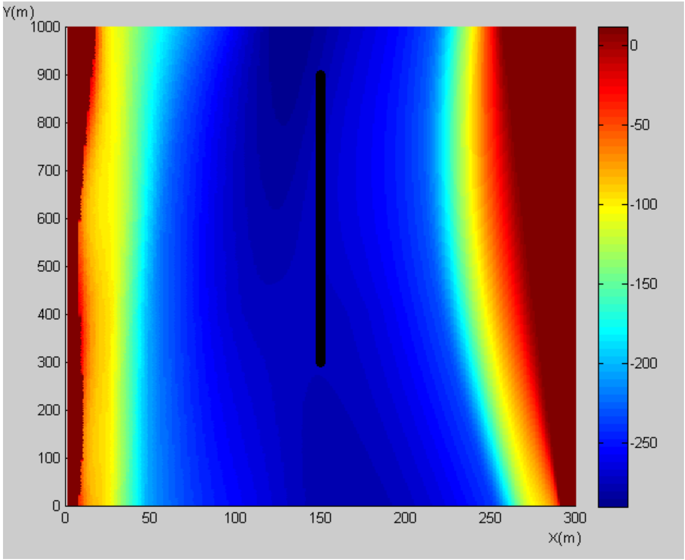
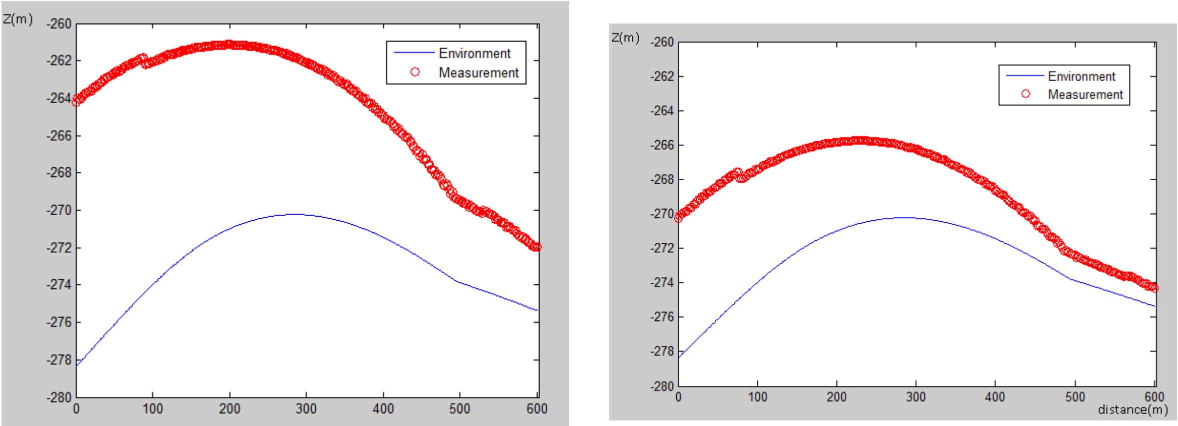


Figure 4.10: Trajectory for $Vehicle_{22}$

Figure 4.10 shows the path taken by the vehicle for this test and figure 4.9 shows the results of the survey. Once again a decrease in the measurement error is observed when the beam aperture angle is reduced. In this case the advances of the rises in the oceanic floor and the delays on the descents of the oceanic floor, in the measurements, are not as evident.



(a) Passage with a 5º beam aperture

(b) Passage witha a 2.5º beam aperture

Figure 4.11: Results from $Vehicle_{22}$ on Environment 2

4.2 MultiBeam

4.2.1 Environment 1

For the first test with the multibeam sonar an *Environment* similar to the one from subsection 4.1.1 is going to be used.

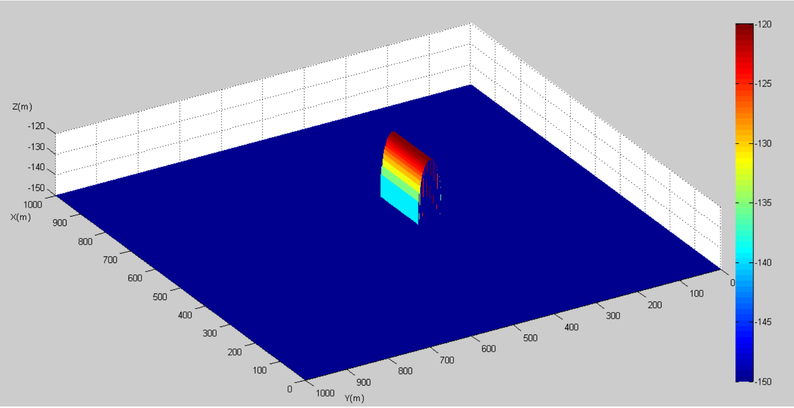


Figure 4.12: Environment for the first test with a multibeam

In this test the first *Vehicle* input is going to be $Vehicle_{31}$, making the path in picture 4.13.

$$Vehicle_{31} = \begin{bmatrix} 600 & 300 & -100 & 0 & 0 & 90 & 0 \\ 600 & 500 & -100 & 0 & 0 & 90 & 100 \end{bmatrix}$$

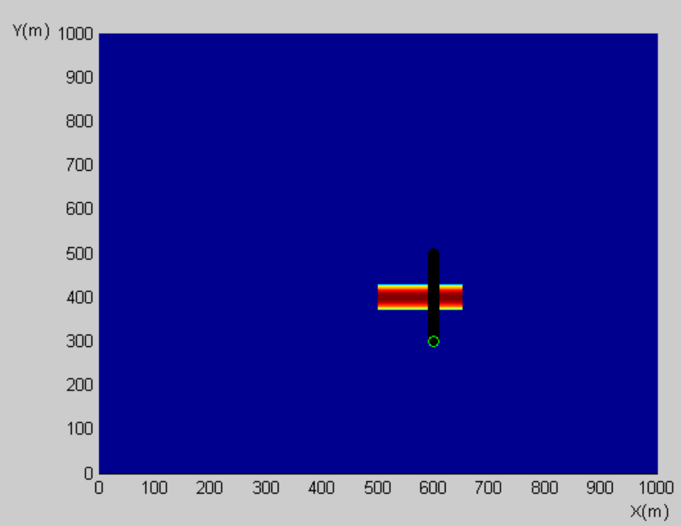
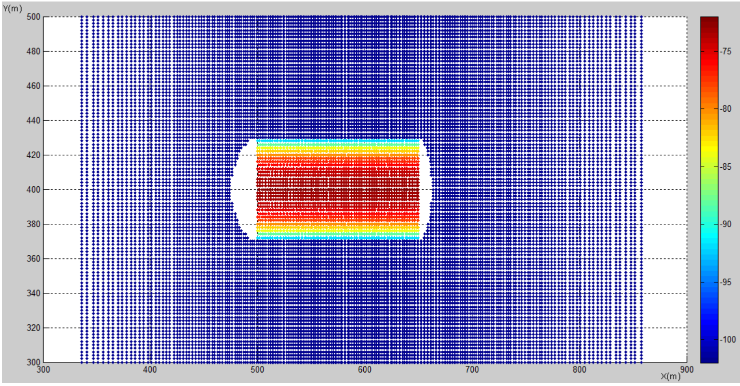
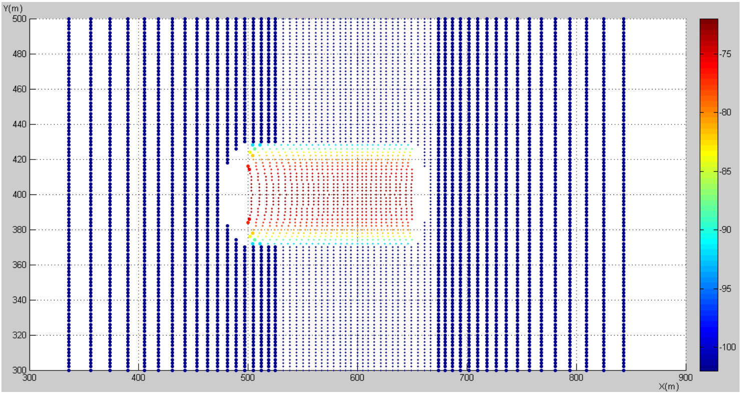


Figure 4.13: Trajectory for $Vehicle_{31}$

Two parameters will be altered in the test in order to compare the results, the first one is the angle between beams (β) and the second one is the time for the trajectory.



(a) $Vehicle_{31}$ with ζ=0.5°



(b) $Vehicle_{31}$ with ζ=2°

Figure 4.14: Results from a survey with $Vehicle_{31}$

By increasing the angle β the number of beams will decrease resulting in less information acquired. This is exactly what is possible to observe in figure 4.14, there is more information in figure 4.14a than in figure 4.14b, and that results in less blank spaces and in a better synthesized image. Still, even with more information available there are two blank areas next to both bases of the cylinder. This is an expected development, since the vehicle is traveling perpendicularly to the cylinder those areas won't be caught by the sonar beams, making a shadow behind that target. The one on the right is smaller because the vehicle is closer to that side.

For the next part the *Vehicle* input it is going to be:

$$Vehicle_{32} = \begin{bmatrix} 600 & 300 & -100 & 0 & 0 & 90 & 0 \\ 600 & 500 & -100 & 0 & 0 & 90 & 50 \end{bmatrix}$$

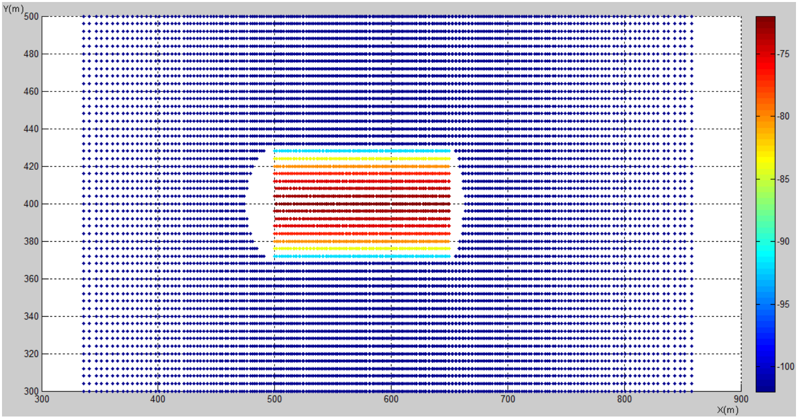


Figure 4.15: Result from survey with $Vehicle_{32}$ with ζ=0.5°

Comparing picture 4.14a with picture 4.15 the only change is the intervals in which the measurements were taken, making the first one more complete than the second. Unsurprisingly, it is clear that the slower the survey the better the information recovered. Still, at some point the information becomes redundant.

4.2.2 Environment 2

Once again, for the next series of tests a more authentic *Environment* will be used. It is the same as the one used in subsection 4.1.2, only bigger.

This test will show how the generated image for a multibeam sonar reacts in a environment closer to a real one, as well as how *FSS* responds to several way points being defined.

The *Vehicle* input used is:

$$Vehicle_4 = \begin{bmatrix} 400 & 500 & -60 & 0 & 0 & 0 & 0 \\ 400 & 700 & -60 & 0 & 0 & -16.7 & 100 \\ 500 & 730 & -60 & 0 & 0 & -31 & 150 \\ 600 & 790 & -60 & 0 & 0 & -31 & 210 \end{bmatrix}$$

Through that input the vehicle will follow a path with three way points. The software does not calculate the angles needed for the course to be correctly followed, they were calculated manually. In picture 4.16 it is possible to see the output of that simulation, the transitions between way points are not smooth as they should be.

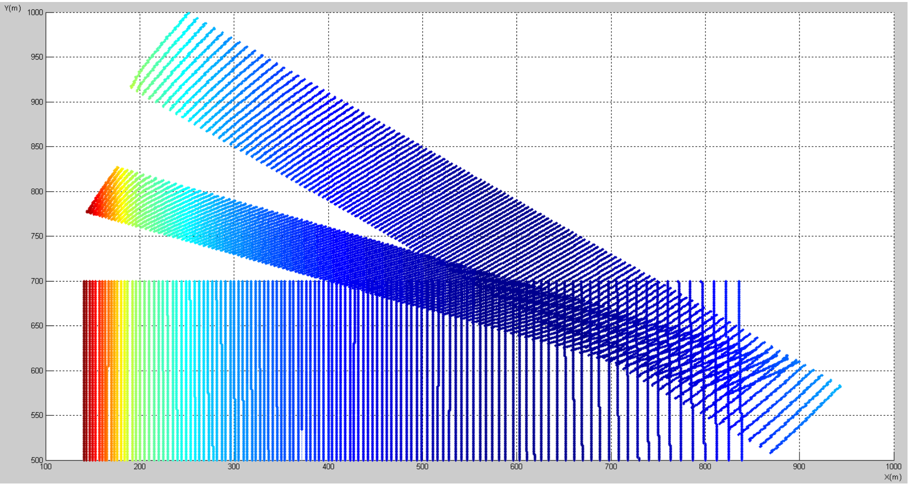


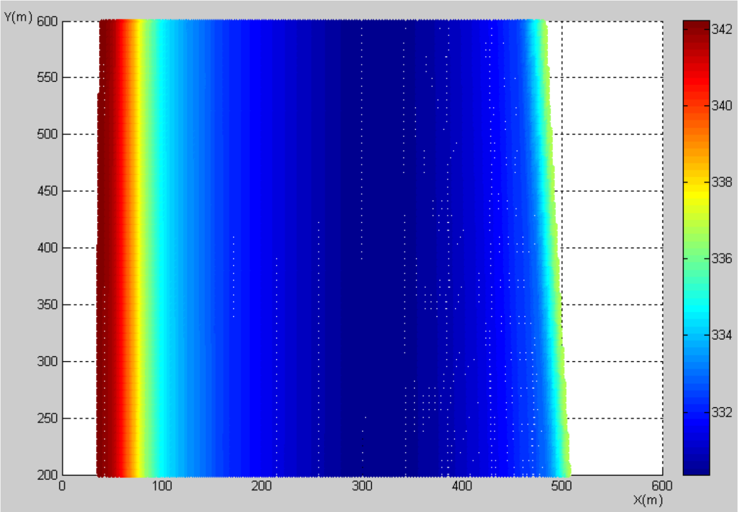
Figure 4.16: Result from survey with $Vehicle_4$ with ζ=1°

If the vehicle input came from a dynamic simulation of a moving vehicle more way points would have been calculated and the transitions would have been smoother. Still, it is possible to see that the simulation reacts well to this type of inputs and that the areas in which there is a superposition of acquired information become more complete than the rest of the survey area.

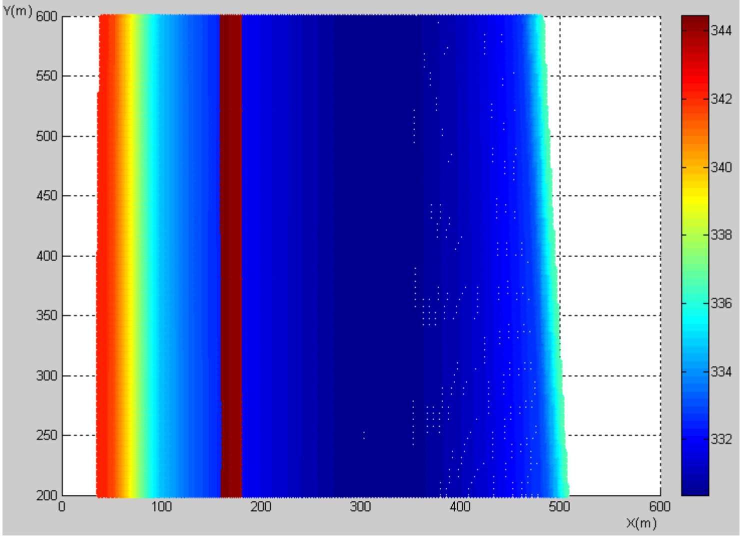
The last test to be performed will demonstrate the output reached when the *Environment* has different types of terrain compositions. In order to notice this changes the output image is formed with the return signal strength information. In this test the *Vehicle* input used is:

$$Vehicle_5 = \begin{bmatrix} 200 & 200 & -50 & 0 & 0 & 0 & 0 \\ 200 & 600 & -50 & 0 & 0 & 0 & 200 \end{bmatrix}$$

Two situations were used in this test, in the first one all the terrain has the same composition and the second has a "pipe" of a different material present. Figure 4.17 shows the results of those tests. In figure 4.17b there is a a line with a different return strength than the ones surrounding it and that is not present in figure 4.17a. This is due to the fact that the backscattering strength of that material is superior to the ones around it.



(a) $Vehicle_5$ all terrain type equal



(b) $Vehicle_5$ with a pipe of different material in the environment variable

Figure 4.17: Return signal strength results

Chapter 5

Conclusions and Future Work

5.1 Conclusions

Considering the objectives proposed in chapter 1, this project is not completely successful. The main objective, of creating a tool to simulate sonar surveys, was only partially completed. The sidescan sonar was not fully implemented since no images were generated from it. It is also not possible to add targets to the environment, which would be a really interesting feature. Still, there were some satisfying results obtained from the echosounder and the multibeam sonars. Also, the modular characteristics of the proposed solution allow an easy way to make upgrades since it is possible to upgrade each module individually.

5.2 Future Work

This project was intended as the first step in the creation of a tool that is not only useful but needed. So, it is only logical that improvements of the current project be made in the future.

Considering that it was one of the main objectives, the first improvement that should be made is the fully implementation of the sidescan sonar. Being able to generate sidescan images would be a great addition. The possibility of adding any number of targets to an Environment would also bring great value to this tool, testing automatic search algorithms can only be possible by having this option. Increasing the number of sonar types available would be something to be considered, widening the range of this software will only add value to it. A graphical interface would simplify greatly the use of this tool and should also be a high priority improvement. Finally, a migration to another programming language would be a great asset. Matlab was an usefull tool to start this project, since it has great number of toolboxes that make some tasks simpler but it is also somewhat restrictive. Using C++ (as is used in all the tools in chapter 2) would allow this project to achieve fantastic results.

References

[1] L-3 Communications SeaBeam Instruments. Multibeam Sonar Theory of Operation, 2000.

[2] SEASCAPE SUBSEA TECHNOLOGY. SIDESCAN SONAR, 2014. URL: `http://www.seascape.nl/support/sidescan-sonar`.

[3] Kongsberg Maritime AS. SIMRAD, 2014. URL: `http://www.simrad.com/me70`.

[4] CompWilde. rpy, 2014. URL: `http://www.computersrwilde.com/Projects/hexacopter/design1.html`.

[5] Andrew Hoggarth and Juan Carballini. The evolution of offshore survey technology for pipeline inspections. *2013 IEEE/OES Acoustics in Underwater Geosciences Symposium*, pages 1–2, July 2013. URL: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6683981`, `doi:10.1109/RIOAcoustics.2013.6683981`.

[6] Shuang Wei and Henry Leung. An Automated Change Detection Approach for Mine Recognition Using Sidescan Sonar Data. (October):553–558, 2009.

[7] Angela D'Amico and Richard Pittenger. A Brief History of Active Sonar. *Aquatic Mammals*, 35(4):426–434, December 2009. URL: `http://www.aquaticmammalsjournal.org/index.php?option=com_content&view=article&id=146:a-brief-history-of-active-sonar&catid=35:volume-35-issue-4&Itemid=93`, `doi:10.1578/AM.35.4.2009.426`.

[8] David A Demer and Josiah S Renfree. Variations in echosounder – transducer performance with water temperature. *ICES Journal of Marine Science*, 65:1021–1035, 2008.

[9] SONARSIM. PHYSicS. *http://www.sonarsim.com/Pages/home.html*, 2014. URL: `http://www.sonarsim.com/Pages/home.html`.

[10] Robert P Goddard. The Sonar Simulation Toolset , Release 4 . 6 : Science , Mathematics , and Algorithms APL-UW TR 0702 October 2008. Technical Report October, 2008.

[11] Val Schmidt. The MB-System Cookbook. 2004.

[12] BioSonics. BioSonic, 2014. URL: `http://www.biosonicsinc.com/product-mx-habitat-echosounder.asp#learnthebasics`.

[13] Wessex Archaeology. MultiEx, 2014. URL: `http://ets.wessexarch.co.uk/recs/how-we-study-the-seafloor/geophysical-survey/02-multibeamsurvey/`.

[14] Puna Ridge. SidescanEx, 2014. URL: `http://www.punaridge.org/doc/factoids/digitaldata/default.htm`.

[15] G.R. Francois, R.E., Garrison. Sound absorption based on ocean measurements. I. Pure water and magnesium sulfate contributions. *Journal of the Acoustical Society of America*, 72(3):896–907.

U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

MIEEC - MESTRADO INTEGRADO EM ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES **2014/2015**

A Dissertação intitulada

"Síntese de Imagens Acústicas de Alvos Subaquáticos"

foi aprovada em provas realizadas em 19-02-2015

o júri

Presidente Professor Doutor Paulo José Cerqueira Gomes da Costa
Professor Auxiliar do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

Professor Doutor José Miguel Soares Almeida
Professor Adjunto Departamento de Engenharia Electrotécnica do Instituto Superior de Engenharia do Porto

Mestre Nuno Alexandre Lopes Moreira da Cruz
Assistente do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projeto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extratos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são corretamente citados.

x Duarte Silva

Autor - Duarte Nuno Reimão Borges Lopes Silva

Faculdade de Engenharia da Universidade do Porto