FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Multipath Acoustic Navigation

Diogo Minhava Lopes

FOR JURY EVALUATION

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Professor Aníbal Castilho Matos

Co-Supervisor: Professor José Carlos Alves

June 30, 2014

In loving memory of my Mother.

Resumo

Debaixo de água, as ondas eletromagnéticas não se propagam. Este facto leva a um problem sério para a robótica subquática: localizar veículos autónomos neste ambiente é muito desafiante, uma vez que sistemas como o GPS não podem ser usados. De qualquer forma, para um veículo autónomo ser capaz de navegar debaixo de água, precisa de se localizar. Assim, ao longo dos anos têm sido desenvolvidos sistemas que, em vez de utilizarem ondas eletromagnéticas, utilizam som para transmitir, no geral, sinais de baixo de água. Estes sistemas, quer sejam utilizados para localização ou navegação, sofrem quase sempre de múltiplos ecos, um fenómeno geralmente considerado mau. Um exemplo de vida real deste fenómeno são os ecos que nos fazem ouvir a mesma voz mais do que uma vez. Debaixo de água, quando um som é transmitido, navega para o destino por mais do que um caminho. Existe um caminho directo e existem caminhos reflectidos na superfície, no fundo, e em qualquer outro objecto que encontre. O resultado final é que, no recetor, temos várias cópias do mesmo sinal transmitido, todas atrasadas de um determinado tempo. O objetivo desta tese é o de perceber como, se for possível, é que se pode tirar partido destes múltiplos caminhos para realizar localização. Mais concretamente, será possível, utilizando apenas um transmissor, localizar o recetor em relação a ele? Se eliminarmos os múltiplos caminhos, a resposta é obviamente não. Com um só caminho, a única informação que se pode retirar é a distância entre transmissor e receptor. No entanto, se se conseguir identificar vários caminhos que a onda sonora tenha percorrido, talvez seja possível. A abordagem tomada começou por avaliar vários tipos de sinais e modulações para perceber quais as características que um sinal transmitido deve ter para facilitar esta tarefa. A seguir, foi construído um sistema de transmissão que permite, de forma fácil e flexível, recolher dados debaixo de água. De seguida, desenvolveu-se um simulador de acústica subaquática sobre o qual se implementou um algoritmo de localização relativa. O simulador foi usado para avaliar estatisticamente o algoritmo, para poder posteriormente testá-lo em dados reais. Concluí-se com uma avaliação do trabalho e considerações sobre trabalho futuro.

Abstract

Underwater, electromagnetic waves do not propagate. This facts leads to a serious problem in underwater robotics: localising robots underwater is very challenging, since systems like GPS are not possible tu use. Nevertheless, for a robot to be able to successfully navigate underwater, it still needs to locate itself. With this in mind, over the years the tendency has been to shift from electromagnetic waves to sound waves when trying to, in general, transmit signals underwater. These systems, whether intended for localisation or communication, typically suffer from multipath, a phenomenon usually regarded as bad. A typical real life multipath example is an echo, that makes us listen to the same voice more than once. Underwater, when sound is transmitted, it travels to the target by more than one path. There is a direct path, there are paths that have been reflected on the surface or on the bottom or on any other object. The end result is that at the receiver we have not one, but several copies of the same signal transmitted, all of them delayed by a different amount of time. The purpose of this thesis is to understand how, if it is possible, can we take advantage of these multiple paths to perform localisation. Specifically, is it possible, using only one transmitter, to localise the receiver with respect to it? If we eliminate multipath, the answer is obviously no. With one path alone, the only information possible to get is the distance between transmitter and receiver, at best. However, if we can identify the several paths that the sound wave has travelled, it might be possible. The approach taken is to firstly evaluate several types of signals and modulations to determine the best characteristics a signal should have to help us with this task. We then build an underwater transmission system to easily and flexibly collect data, and we develop an underwater acoustic simulator, on top of which a localisation algorithm is implemented. We use this simulator to statistically evaluate the algorithm, and proceed to test the algorithm on real data. We end by evaluating the work done, and give considerations on what should follow after.

Acknowledgements

I'd like to start by thanking my advisor, Professor Aníbal Castilho Matos, for his work with me during this thesis. His insights were critical to some of the successful results achieved. I'd also like to thank Professor José Carlos Alves, for challenging me with new and interesting projects since my first year in Electrical Engineering. These projects made me want to push myself into learning more each day, and I am sure that they are among the most valuable learning experiences I've had. I want to thank my friends for their support. In particular, I'd like to thank my colleague, Henrique Cabral, with whom I've been working together since our first year. Working continuously with a friend helped keeping me motivated and striving for the best results I could get. Finally, A big thank you note to my family, specially my mother, my father and my brother. Their support was always tremendous, no matter the choices I did throughout my life. I'm very lucky to say I couldn't wish for a more supportive family to have grown up with, and everything I am and will ever accomplish is due to them.

Diogo Minhava Lopes

"The first principle is that you must not fool yourself,
and you are the easiest person to fool."

Richard P. Feynman

Contents

List of Figures

List of Tables

List of Acronyms

AUV	Autonomous Underwater Vehicle
BFSK	Binary Frequency Shift Keying
BPSK	Binary Phase Shift Keying
BSD	Berkeley Software Distribution
CSS	Complimentary Sets of Sequences
DFT	Discrete Fourier Transform
DVL	Doppler Velocity Log
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
GPS	Global Positioning System
HDL	Hardware Description Language
INS	Inertial Navigation System
LBL	Long Baseline
MLS	Maximum Length Sequence
PRN	Pseudorandom Noise
RTC	Real Time Clock
SBL	Short Baseline
USBL	Ultra Short Baseline

Chapter 1

Introduction

Underwater acoustics is an extensively studied subject when dealing with underwater robotics. Since electromagnetic waves do not propagate in this medium, the alternative researched and implemented over the last decades has been to use sound waves to replace electromagnetic ones. There are two big areas for which this is required: communication, that is, when we want two AUVs, for example, to exchange information, and localisation, that is, when we need to have an underwater vehicle estimate its position with respect to some transmitter placed at known coordinates. Whether we're interested in communicating or localising, however, there is a phenomenon that typically affects results in a very negative way: multipath. When we transmit a signal, whether acoustic or electromagnetic, whether underwater or in a street with lots of buildings, it will travel in every direction and reflect off surfaces it finds in the way. The end result at the receiver is that it gets not only the signal that travelled directly to it, but every reflection that finds its way to the place where it is located. This is known to affect GPS performance severely in big cities, for example, where large structures exist. Underwater, multipath is one of the major unwanted issues that cripple acoustic systems. These systems typically use signal processing techniques to eliminate the effects that the multiple paths cause. The purpose of this thesis is to do precisely the opposite. We're interesting in studying the usefulness of multipath. Typical localisation systems for AUVs include the LBL. In these systems, several transmitters are placed at known positions and transmit different signals each, at different, previously established, times. Upon receiving each one of them, the AUV calculates the distance to the respective transmitter, and knowing the distance to every transmitter it estimates its own position. We're looking for something different, however. If the signal sent by a signal transmitter arrives at several different times at the AUV, it might be possible to figure out the paths that each reflected signal took to arrive. If this is, in fact, possible, then it is possible to localise the AUV with respect to the sender, using only one signal and one fixed position transmitter, and this is precisely what we propose to do.

The rest of this document is organised in 8 chapters. First, in the state of the art chapter, we present some of the work that is known on underwater acoustics and localisation. We then proceed to describe our problem in more detail, and give some ideas on how to approach it. The rest of the thesis is developed upon almost entirely on these two chapters. We begin by describing the

acoustic transmission system used to perform underwater tests. We follow with a brief study of different types of signals and modulations to determine the best options for our transmitted signals of choice. The goal is to find a signal that, when received several times at a target with different delays, makes it easy to identify these delays individually. The chapter following describes an underwater simulator built, with the purpose of analysing the algorithm detailed in the chapter 7. We then show results of real tests performed and finally discuss our work and present some considerations on what should follow.

Chapter 2

State of the Art

In this chapter an overview of the current state of the art related to the subject of study is presented. In the first section, the basic acoustic properties of the ocean are discussed. The purpose is to show that, even at a glance, sound transmission in this medium is far from being a trivial matter. The first object of discussion is sound speed and how it changes with respect to, fundamentally, three quantities: temperature, salinity and pressure. While salinity can usually be considered constant in a given area, the same is not valid for temperature and pressure. An overview of how the first changes with depth is given. As for pressure, it can be related to depth using the Leroy-Parthiot equation, for example [3].

Transmission loss and Doppler effect both play important roles in understanding this subject, so they are discussed separately.

With the basics gone through, some model is required to fully understand underwater acoustic wave propagation. Due to it's intuitive nature, it's simplicity and it's natural adequacy to localisation problems, ray tracing is the model we choose to focus on. Other models are briefly mentioned but with no intent of providing much detail.

After going through ocean acoustics, it is important to discuss localisation and navigation, which is the main purpose of this work. Extensive research has been done on these matters, and while a complete discussion is by no means be presented, the fundamental ideas for a basic understanding of what follows are discussed.

2.1 Ocean Acoustics

The ocean can be thought of as a waveguide limited by the sea surface and the seafloor [2]. In a homogeneous medium, waves propagate in straight lines. However, as in the case of light in a medium where the refractive index is not the same in every point, sound waves are bent if the velocity of propagation is not constant. This suggests that an analogy can be established between sound propagation and optics, which is true for sound of very high frequency when certain phenomena like diffraction and absorption can be ignored [4]. When considering sound refraction in the sea, it is usually assumed that temperature varies only vertically, which is the same as considering the ocean as consisting of strata, in any one of which the temperature is constant along a large horizontal distance [4]. The speed of sound in the ocean is a function of temperature, salinity and pressure, which is a function of depth. This relation can be expressed by the following simplified expression [2]

$$c = 1449.2 + 4.6T - 0.055T^2 + 0.00029T^3 + (1 - 34 - 0.01T)(S - 35) + 0.016z\,. \tag{2.1}$$

Usually, however, when deriving distances from time-of-flight measurements, more accurate formulas are used. Such an example is the equation by Chen and Millero [5], used in [6]. Looking at (2.1), it is easy to understand that seasonal and diurnal changes have an effect on sound speed in the upper part of the ocean. In non-polar regions, near-surface oceanographic properties result from wind and wave activity near the boundary between air and sea. In this layer, water temperature is constant and, thus, sound speed increases in it in an approximately linear fashion with depth. The more agitated the upper layer is, the deeper this region, known as the surface duct, is. Below the surface duct, temperature decreases with depth and, consequently, so does sound speed. This region is known as the thermocline. Below it, the temperature is constant and sound speed increases only due to increasing pressure [2]. In shallow waters (a few hundred meters deep), only the upper part of the described profile is relevant, which is affected mainly by season and time of day. Having discussed the basic principles of sound propagation in the ocean, it is necessary to have a propagation model in order to have a good understanding of the medium in different scenarios so it is possible to successfully deploy a localisation system. The rest of this section is dedicated to discussing several issues related with ocean acoustics as well as different models for sound propagation in this medium, emphasising ray tracing.

2.1.1 Ray

Throughout the rest of this document, the definition of ray will be important. A ray is a normal to a wavefront. Rays can be used to trace the paths that waves travel through and separate computations can be made on individual rays. Furthermore, an eigenray is defined as a ray that travels precisely from a source to a destination of interest. These are the rays that define the acoustic wave that actually reaches the destination, and are thus the rays of interest.

2.1.2 Transmission Loss

As sound propagates through water, it suffers attenuation due to several different factors. The standard unit of measure of underwater acoustic propagation is acoustic intensity (I) [7], which is defined as

$$I = \frac{p}{\rho c}, \tag{2.2}$$

where p is the pressure amplitude of the plane wave, ρ is the water density and c the speed of sound in the water. Transmission loss is usually defined as

$$TL = 10\log_{10}\frac{I_{ref}}{I(r,z)}, \tag{2.3}$$

where I_{ref} is the intensity measured at 1 m from the source and $I(r,z)$ the intensity measured at an arbitrary point at range r and depth z.

There are three main factors contributing to transmission loss: geometrical spreading, absorption and rebounds both at the surface and the bottom [6], each one of which is now described.

2.1.2.1 Geometrical Spreading

Geometrical spreading is caused by the expansion of the acoustic wave through the medium [6]. In other words, it's a measure of the signal weakening as it propagates outward from the source [2]. Two types of spreading are considered: spherical spreading and cylindrical spreading. The first one occurs when we consider the field near a source point, while the second occurs when we consider larger ranges. Transmission loss for both of these cases can be computed as, respectively, $20\log r$ and $10\log r$ [2] where r is the distance travelled by the ray considered.

2.1.2.2 Absorption

Part of the acoustic energy of a travelling sound wave in water is transformed into heat continuously. This absorption effect is partly due to the viscosity of the liquid and by scattering of sound waves by different inhomogeneities [8]. The combined effect of absorption and scattering is called sound attenuation and is typically only possible to measure their combined effect. Absorption increases with frequency and is dependent on temperature, salinity, depth and the pH value of the water [9]. The transmission loss due to absorption is a function of the distance r travelled by the

ray being analysed and is calculated as $\alpha r 10^{-3}$, where α is the absorption coefficient. As for the coefficient α, there are empirical formulas to compute it, for example the Francois-Garrisson equation [10] and the one by Marsh and Schulkin [11].

2.1.2.3 Rebounds

The analysis of sound propagation is greatly simplified if no rays are traced into the bottom [9]. If the ocean is deep in the region of interest, bottom interaction will have little effect, so this can be safely assumed. In shallow waters, however, this is not true, as we will see.

When a ray hits the surface, it is reflected and suffers a transmission loss that can be computed by the simple equation given by the Beckmann-Spizzichino formula [6, 12].

2.1.3 Doppler Effect

This section follows the works of [2], [13] and [14]. Other references will be made in the text where appropriate.

Basic physics tells us that a moving source or receiver causes a frequency Doppler shift in the received signal whose magnitude is proportional do the ratio of the relative speed of the transmitter with respect to the receiver and the speed of sound: $a = v/c$. In the ocean, every source/receiver moves even if not intentionally due to currents. Furthermore, because sound speed is much smaller than the speed of light, Doppler shifts in acoustic transmissions in the water are proportionally much bigger than electromagnetic transmissions in free space. To complicate matters further, while in free space frequency shifts are expressed by a simple relation that makes a received signal be distorted in time having duration $T/(1+a)$ and a frequency offset $a f_c$ (where f_c is the frequency of the carrier), in a waveguide or stratified environment, such as the ocean, the Doppler structure found is much more complicated because of multipath phenomena. Each ray that arrives at the receiver has suffered a different Doppler shift, causing what's known as Doppler spread.

Another related and important effect, introduced as the dynamic effect of swell in [6] is caused by the random variability of the channel in time. Since the surface of the ocean is not perfectly flat, it's roughness introduces scattering and makes any assumption of specular reflections essentially wrong. What we have in the ocean surface are actually diffuse reflections. The vertical displacement of the surface can be modelled accurately as a zero-mean Gaussian random variable whose power spectrum is characterized by the wind speed [8]. This displacement that causes the scattering of reflected signal induces a very large Doppler spread that can easily dominate over the ones caused by other phenomena. The following equation can be used to calculate this spread [8]:

$$B = 0.0175 \left(\frac{f}{c}\right) w^{3/2} \cos\theta \,, \tag{2.4}$$

where c is taken to be $1500\,\mathrm{m\,s^{-1}}$, w is the wind speed and θ is the angle of incidence of the ray on the surface.

2.1.4 Multipath Propagation

Every wireless communication channel suffers from multipath effects. However, while an electromagnetic wave propagating travelling in free space will have this unwanted effect only due to reflections on obstacles, underwater there are mainly two components that contribute to the overall multipath: reflections on the surface, bottom and any object on the water and ray bending due to refraction. The intersymbol interference caused by this multipath in a single carrier communication system in the horizontal underwater acoustic channels can be of several hundred symbol intervals for moderate to high data rates as opposed to the typical several symbols for a common radio channel [14]. Although in theory there are infinitely many signal echoes from sender to receiver, the amount of rays that haven't lost most of their energy at the moment they reach the receiver, having gone through the significant paths, are finite in number [13]. If we consider the ocean surface to be flat, reflections on it are specular, meaning that the angle of incidence is equal to the angle of reflection with respect to the normal to the surface. As we have previously seen, though, scattering must be accounted for by introducing Doppler spread in the channel's transfer function. Bottom reflections are much more complicated, as they depend on the type of bottom and the grazing angle [13, 2].

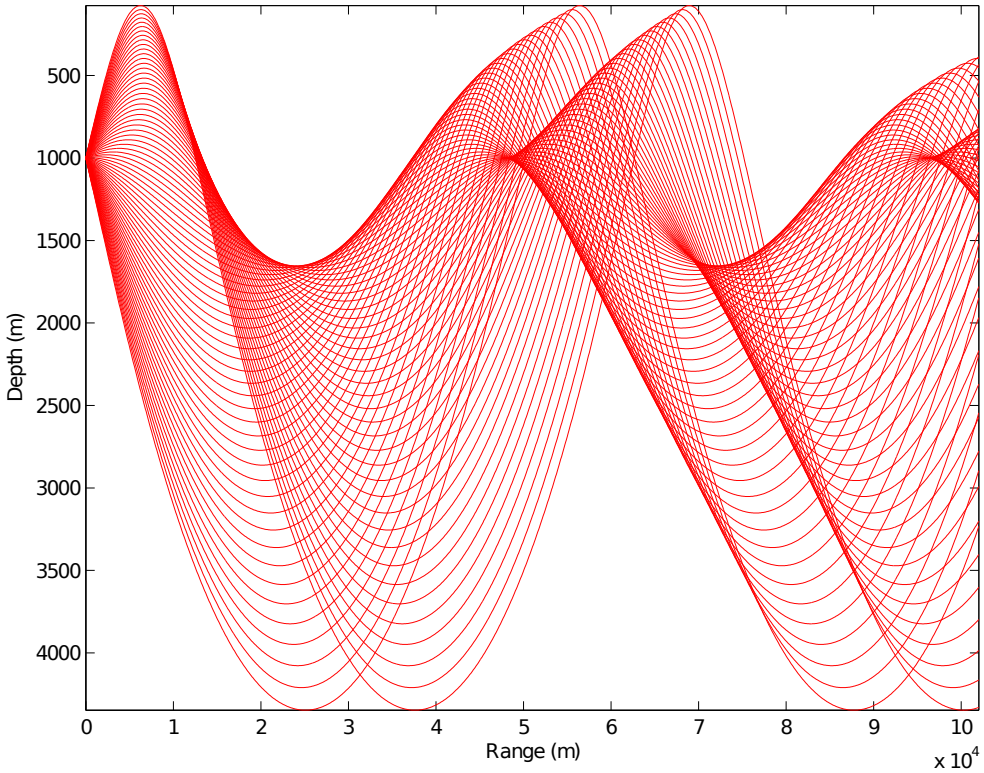

Figure 2.1: Rays calculated by **Bellhop** in a flat deep water waveguide. Taken from [1].

Fig 2.1 shows a typical ray trace diagram calculated by Bellhop [1] showing several different paths. The result of multipath is that, in any given point any given point in the ocean receives multiple copies of the same signal that have travelled through different paths and carry different

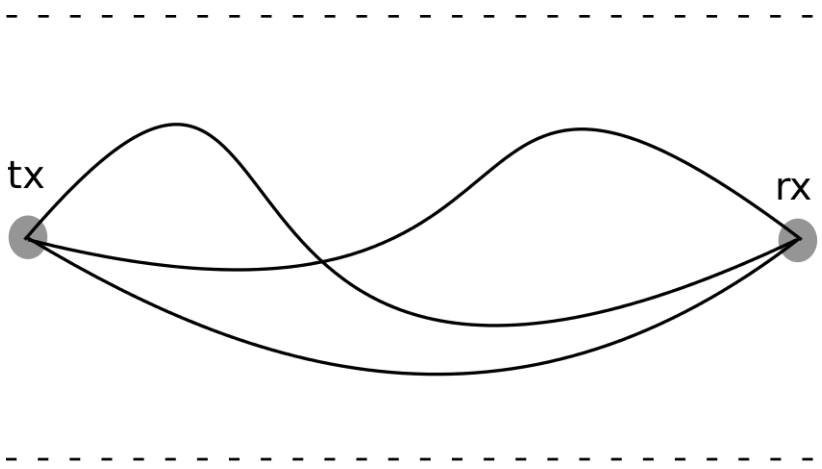

Figure 2.2: Multipath example in deep water.

amounts of energy, having also different delays. All of this causes several negative effects such as distortion and de-correlation between the same signal received at different sources [15]. Distortion has important consequences, as transmitted pulses are stretched at the receiver due to the several delayed arrivals. [15] shows an example where an explosive pulse with a duration of a few milliseconds is measured in seconds at the receiving end.

Sound propagates very differently in deep waters than it does in shallow waters. What accounts for these differences is mainly multipath differences, as we shall now see. The definition of shallow and deep is somewhat vague, but shallow usually refers to the region of continental shelves, with depths of a few hundred metres, and deep to the region past continental shelves [14].

2.1.4.1 Deep Water

In deep water, longer range propagations are possible without bottom interaction. The significant rays are typically either bent only due to refraction or refracted and reflected at the surface [2]. The situation will depend on the depth at which we are transmitting. Since rays bend towards layers where sound speed is lowest, it is possible to have a situation where the sound rays are confined in the region where their speed is minimum (axis of the deep sound channel) [14]. This property is used to communicate at several thousands of miles because the sound energy is essentially trapped in this channel. This situation is depicted in Fig. 2.2. In many situations, however, we will be interested in transmitting relatively close to the surface, so sound rays will still be reflect on the surface, but bottom interactions may safely be ignored, leading to a much simpler problem than that that will be described next.

2.1.4.2 Shallow Water

The situation is quite different when considering shallow water. In such cases, sound speed may be taken as approximately constant [13]. In these conditions, sound rays travel in straight lines and bounce both on the surface and bottom for as long as they have energy. A typical scenario is

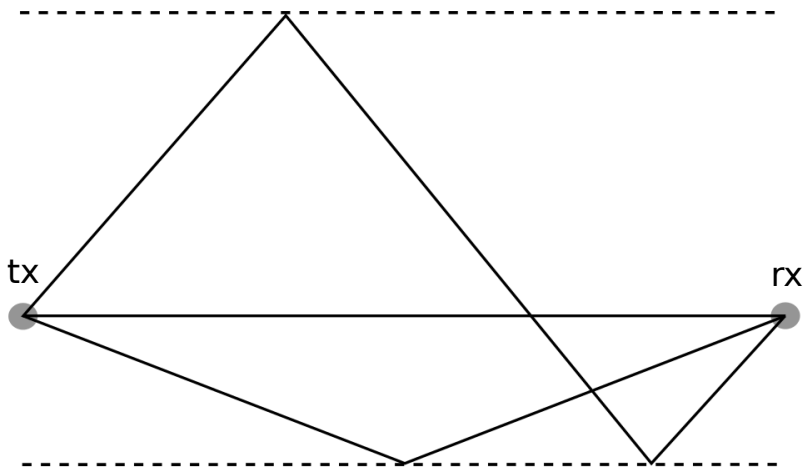

Figure 2.3: Multipath example in shallow water.

shown in Fig. 2.3. It is generally much harder to preform accurate ray tracing in shallow water. Even though we're assuming there is no refraction, bottom interactions can be much harder to deal with than this effect. Without knowing the type and shape of the bottom, little can be done to predict how rays will reflect on it. The two dimensional model generally used cannot even be relied upon in circumstances where rays diverge from the vertical planes they were launched in. Still, localisation in this scenario, however challenging, might be possible.

2.1.5 Ray Tracing

Ray tracing is about understanding how rays propagate in order to trace the propagation of wavefronts. The underlying assumption is that the energy of the wave is confined in different paths [6]. Ray theory can be derived from the wave equation taking advantage of some simplifications. The method essentially involves a high-frequency approximation, so it is sufficiently accurate for applications involving communication systems for short and medium distances. [9] shows, however, that ray theory can also be applied for low frequencies. One of the first descriptions of ray tracing in underwater acoustics is given by Lichte [16]. He presented an intuitive way of understanding how sound rays propagate in the ocean. The starting point is realising that the ocean is acoustically inhomogeneous in horizontal layers. As a result, sound rays are not straight lines, but rather curves. Turning back to our layered model in which sound travels at a constant speed horizontally, it is possible to use Snell's law to trace the entire path of a ray. Fig.2.4 shows such a scenario. The refraction law tells us that

$$k_1 \cos\theta_1 = k_2 \cos\theta_2 \,. \tag{2.5}$$

Using this for k_2 and k_3 , k_3 and k_4 , and so on, and allowing k to vary continuously as a function of the depth z, the above can be rewritten as

$$k(z_1)\cos\theta(z_1) = k(z_2)\cos\theta(z_2) \,. \tag{2.6}$$

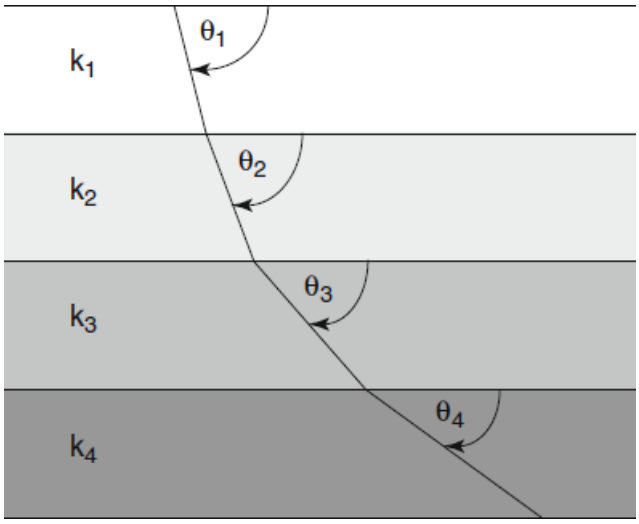

Figure 2.4: A ray refracting through a stack of layers. Taken from [2]

For an origin depth z_0 and corresponding declination angle θ_0 , the ray angle at a specified receiver depth z_r is given by

$$\theta(z_r) = \arccos\left[\frac{k(z_0)}{k(z_r)}\cos\theta(z_0)\right]. \tag{2.7}$$

Considering Fig.2.4 again and letting r be the horizontal axis and z the vertical one (depth), and considering the limit case where the number of layers is infinite, we can write the following differential equation

$$\frac{dr}{dz} = \cot\theta(z), \tag{2.8}$$

which can be integrated to yield

$$r(z) = \int_{z_0}^{z_r} \cot\theta(z)\,dz. \tag{2.9}$$

When integrating the ray path, care should be taken so that when the ray becomes horizontal, the integration step dz becomes negative. While the geometrical approach just described has been used with some success [6], finding eigenrays by interpolating the properties of two adjacent rays that suffer the same number of rebounds such that the receiver is in the middle of them (see Fig. 2.5), the classical, more formal approach starts with the Helmholtz equation in Cartesian coordinates $\mathbf{x} = (x, y, z)$:

$$\nabla^2 p + \frac{\omega^2}{c^2(\mathbf{x})} = -\delta(\mathbf{x} - \mathbf{x_0}), \tag{2.10}$$

where $c(\mathbf{x})$ is the speed of sound and ω is the angular frequency of the source located at $\mathbf{x_0}$. A full derivation of the ray equations from this starting point can be found in [2]. These are typically

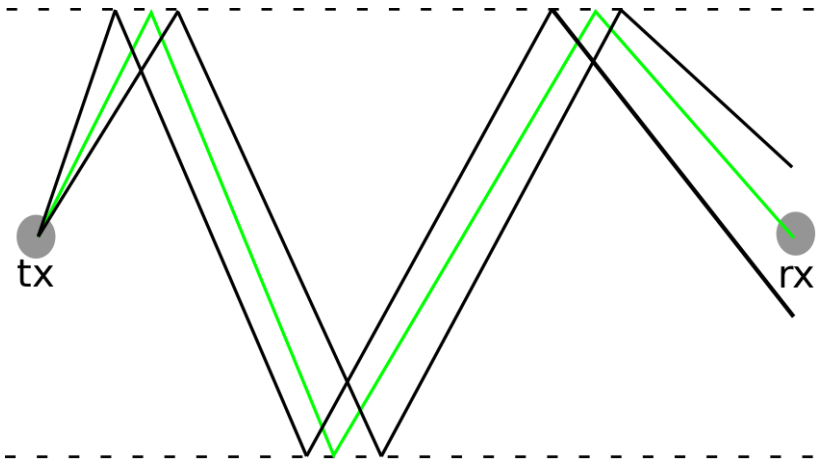

Figure 2.5: Geometrical approach to obtain an eigenray interpolating the properties of two adjacent rays.

written in cylindrical coordinates:

$$\frac{dr}{ds} = c\,\xi(s), \qquad \frac{d\xi}{ds} = -\frac{1}{c^2}\frac{\partial c}{\partial r}, \tag{2.11}$$

$$\frac{dz}{ds} = c\,\zeta(s), \qquad \frac{d\zeta}{ds} = -\frac{1}{c^2}\frac{\partial c}{\partial z}, \tag{2.12}$$

where $[r(s), z(s)]$ is the ray trajectory, s is the arc length along the ray and, since the tangent vector to such a curve is given by $[dr/ds, dz/ds]$, the pair $c[\xi(s), \zeta(s)]$ represents the tangent vector to the ray. The most common way of solving these equations is by using standard numerical integrators. One of the simplest of such methods is the Euler's method. Though it's a very simple first order method, it is very inefficient and yields an error per step proportional to the square of the step size. A second order Runge Kutta method might be more appropriate when accuracy is needed. In the case of the Bellhop program [1], a two step polygon method is used.

Finally, it is worth noting that, although standard codes exist to solve the ray equations, the sound speed profile of a particular section of the ocean is not typically known. A further issue has to do with the reflections on the surface and the bottom of our waveguide. When this happens, the integration must then be restarted. While this might be straightforward for surface reflections, it is far from being trivial when bottom reflections must be considered, since the bathymetry of a given area it not likely to be available. It is also worth noting that currently almost all methods for modelling the acoustic underwater channel are two dimensional, considering only range and depth. The assumption made is that a sound ray launched into a particular vertical plane remains in that plane over the entire transmission path [17]. This assumption is reasonable if we take into account the fact that horizontal sound speed variations are negligible. In shallow waters, however, this assumption may not be accurate due to bottom interactions, since the acoustic wavelength might be comparable to the ocean depth. Under these conditions, a reflection in the bottom of the ocean (which will not always be flat) will cause rays to reflect to different vertical planes.

2.1.5.1 Problems with Ray Theory

Although ray theory is widely used and provides good results under certain conditions, it suffers from some problems that make it diverge from experimental results. [4] provides a good summary on the efforts developed to understand these divergences, the first of which is *diffraction*. If we consider sound propagating from a point below which sound speed decreases, a shadow zone into which no rays penetrate appears [18]. The problem with this is that, such as light, shadow boundaries are not sharp. Light is diffracted around obstacles and so is sound. However, diffraction effects become bigger with increasing wavelengths (and consequently so does ray theory), and since for localisation applications we're mostly interested in high frequency signals, this effect is bounded.

Scattering effects are also not predicted by ray theory, but they may be responsible for the discrepancies shown with respect to experimental results. In the same fashion that sunlight scatters with raindrops to create rainbows, sound waves are known to suffer from scattering in the sea.

Finally, even though the horizontal temperature gradient in the ocean is much smaller than the vertical one, it is not zero. Although this is mentioned by [4], more recent works such as [2] assume a zero gradient. The results achieved in, for example, [6], show that this is a reasonable assumption to make in most conditions.

A more serious issue found in ray theory is the existence of singularities known as caustics [9]. According to ray theory, at these points ray intensity goes to infinity, which is obviously false. A workaround for this problem is presented in [2].

2.1.6 Other Models

Although ray tracing is a reasonable model that is still widely used, it is important to recognise that other models to describe sound waves underwater exist. Although different, they all have the same thing in common: their starting point. Every model naturally starts from the wave equation, and they differ essentially in the approach chosen to solve it. They furthermore assume harmonic solutions to this equation to obtain the Helmholtz equation, which is the one actually solved. There are fundamentally five methods to solve this equation [7, 2], which are: **ray methods**, **wavenumber integration** techniques, **normal modes**, **parabolic equations** and **finite differences and finite elements** approaches. [2] gives a very detailed analysis of each one of these methods.

As argued before, ray theory was the approach chosen for this work due to it's adequacy to localisation problems, where the frequencies used are typically high.

2.2 Localisation and Navigation

Localisation, in terms of mobile robots, is about the robot being able to determine where it is with respect to some reference. It is key for successful navigation and several challenges exist in this area. When considering underwater localisation, knowing where the robot is is critical to ensure that correct and repeatable measurements are being taken for, for example, reef surveying [19]. Good information on localisation is also essential for safe operation and recovery of an AUV [20]. The three primary methods for AUV navigation are dead-reckoning and inertial navigation, acoustic navigation and geophysical navigation [20]. Our focus is on acoustic navigation. In the end a brief description of sensor fusion and the Kalman filter is given. Before going into detail about these techniques, though, some challenges of localisation in general are first discussed.

2.2.1 Sensor Noise and Aliasing

This section is mostly based on [21] who presents a very good discussion on this subject. The first thing to notice in our particular environment is that there is no access to a GPS system, since water is essentially opaque to electromagnetic waves. Even if we had access to GPS positioning, what we're interested in most of the times is actually the relative position with respect to some object (a dam, for example) rather than the absolute position with respect to the Earth's reference frame.

Sensors are how a robot perceives it's environment, so the correctness of their readings must be understood. Every sensor is affected by noise that limits the consistency of their readings, and the source of this noise cannot usually be compensated for. A typical case is the one of a sonar. When a sonar emits a sound towards a surface, it's reflection will not be perfect, and some of the energy will not return. The distortion that the echoed sound suffers depends on a number of things, such as the roughness of the material it reflected on, for example. Multipath can also interfere with sonar readings. Other factors, such as variations on the speed of sound, introduce noise that cannot be factored out. The consequence of this is that noise reduces the amount of useful information that we can obtain from sensors. The solution is generally always taking multiple readings and combine information from several sensors while statistically the error of the readings.

Sensor aliasing is a different but also important phenomenon that occurs frequently with sensor readings. Even without the presence of noise, aliasing can be a limiting factor when trying to localise a robot. The problem is that sensors will naturally give the same reading for different situations, so usually several readings from different sensors are required to uniquely determine a robot's position.

2.2.2 Inertial Navigation

One of the most common localisation and navigation techniques is based on integrating acceleration and velocity to obtain position. This is known as dead-reckoning. Common sensors merged to do this include accelerometers, gyroscopes and Doppler velocity logs [20, 22]. Integrating inertial information with information from different sensors using a Kalman filter is a very common

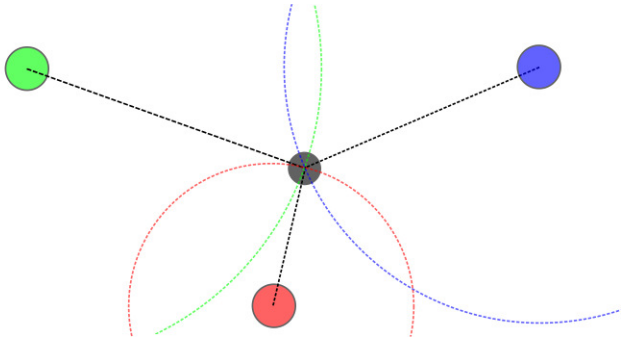

Figure 2.6: An example of 2D localisation using LBL.

approach which largely improves performance over dead-reckoning alone. Two problems with relying on inertial navigation are:

- Initialization of the system;
- Position drift.

The first problem has to do with the fact that, if we rely exclusively on an inertial navigation system (INS), we must be able to somehow determine where we're starting, otherwise integrating velocity doesn't help in determining the robots position. The second problem is related to the integration that has to be performed constantly. Since inertial sensors suffer from noise as any other sensor, when integrating their readings to obtain position we're actually integrating error as well, which accumulates and translates into a position drift that grows bigger with time and must somehow be bounded. A common way of bounding this error is to have to vehicle come near the surface to get a position fix from, for example, a GPS receiver [20].

2.2.3 Acoustic Navigation

There are typically three types of systems that have been used for acoustic positioning: long baseline (LBL) or short baseline (SBL) and ultra-short baseline (USBL) [20]. They provide low frequency measurements that can be used to update predictions obtained by an INS using, typically, a Kalman filter. [22] employs this strategy. In LBL systems, an array of transponders with known locations is deployed. Periodically, the AUV send a ping and the transponders respond to this ping. From this response, the AUV can calculate the time of flight to every transponder and either calculate it's position by intersecting spheres (or circumferences in a 2D scenario shown in Fig. 2.6) or use the raw time of flight measurements directly in a Kalman filter [20, 23]. SBL systems similarly calculate the target's distance to several transducers. USBL systems work in a different way: they have receiver arrays that can measure the angle and the range to an acoustic beacon [20].

A key issue in these systems is the problem of multipath interference, which may lead to an incorrect time of flight calculation.

2.2.4 Geophysical Navigation

[20] provides a brief discussion on geophysical navigation. This type of navigation requires a known map of the environment including elements such as bathymetry and magnetic field. Matching sensor data with known data it is possible to develop navigation systems. However, no details will be given here about these techniques since they are out of the scope of our work.

2.2.5 Time of Flight calculation

An important issue when using acoustic systems like LBL and USBL is how to calculate times of flight of received signals. A first approach is to actually measure the time from the moment the signal is sent to the moment it is received. This is a very poor approach, however, since received signals are very prone to be affected by noise, so there is no reliable way of determining when we actually started receiving it. The typical approach is to, instead, keep a local copy of the signal that's received and cross correlate it with the one actually received. It is then an easy matter to get the time of flight from the correlation peak. Such an approach is taken in, for example, [6], and is the one typically used. Signals should, however, be chosen with some care. A receiver will get the following signal:

$$r(t) = \alpha_0 b_0(t-t_0) + \alpha_1 b_1(t-t_1) + ... + \alpha_n b_n(t-t_n)\,, \tag{2.13}$$

where α_i is some attenuation. We must then guarantee that the signals used exhibit good cross correlation properties. This means that the cross correlation between the sent and received signal should have very distinct peaks that tell us the times different rays have travelled. [6] uses what is known as Complementary Sets of Sequences, presented in [24], to code the signals used. Other options include PRN sequences, most notably Gold and Kasami codes.

2.2.6 Sensor Fusion and the Kalman Filter

Sensor fusion is the group of techniques that merge information from several different sensors in order to obtain a better estimate than that that would be obtained by considering only single independent measurements. [25] presented one of the most widely used algorithms ever for performing this task, famously known as the Kalman filter. A very brief description of what it does is now given, without going into much detail. A simple introduction that provides the material required for implementation is given in [26].

First of all, we're trying to estimate a certain state vector $\mathbf{x} = [x_1, x_2, ..., x_n]$. A covariance matrix representing the error associated with this state vector is also maintained. The state vector typically includes the robot's position but it may also include any other state variable of interest, such as a map representation. It is important to be aware that the Kalman filter assumes that all noise is Gaussian. In practice, however, it is used whether or not this is exactly true. A typical case where noise is not Gaussian is the GPS. The algorithm is recursive and consists of essentially two steps: a prediction and a correction. It is particularly suited to the case where we have high

frequency sensors such as accelerometers that can be integrated to **predict** a position and low frequency sensors like GPS or LBL to **correct** the prediction made. A simple proof that the resulting error of combining these two steps is better than using either one of them separately is given in [21]. One of the most interesting aspects of the algorithm is that it keeps track of the errors of the state estimations (by means of the covariance matrix). This can be very useful in decision making by an AUV: it can decide whether or not it is safe to perform a certain action based on how certain it is that its position is the one on the current state vector, for example.

Chapter 3

The Problem

The problem we're interested in is the one of performing the relative localisation of an underwater receiver (an AUV, for example) with respect to some transmitter. As we've seen, multipath makes underwater communication challenging. The goal of this work is, however, understanding how we can use these multiple paths sound takes to help us localise the target and ultimately develop a system that performs this task. This section is dedicated to explaining how this might be possible and discuss some issues that may appear from trying to perform such a task.

The approach taken here is to consider a single transmitter and a single receiver and explore possible solutions to the problem by firstly considering only a direct sound path, then a surface reflection and finally a surface and a bottom reflection. We assume that there is a way of calculating the time of flight for each ray. Considerations on how to do this are given in section 2.2. We also assume for now that accurate ray tracing has been performed so we have information on every eigenray, and that we can distinguish these eigenrays from each other at the receiver.

Direct Path

Considering only the direct path from sender to receiver is the simplest case, and it is shown in Fig. 3.1. In this case, the only information we have is the ray length and time of flight. It is possible to obtain the distance between the transmitter and the receiver but nothing else, as shown by the dashed line. In reality, the transmitter can be anywhere on the boundaries of a sphere centred in the source with radius equal to this distance measured. In this two dimensional problem considered, however, we consider that it lays somewhere on the circumference of the same radius.

Surface Reflection

When we consider one surface reflection added to the direct path, and if we add the assumption that sound propagates in straight lines (which is approximately true for shallow water), we have the situation illustrated in Fig. 3.2. If we assume that we have information on both rays that hit the

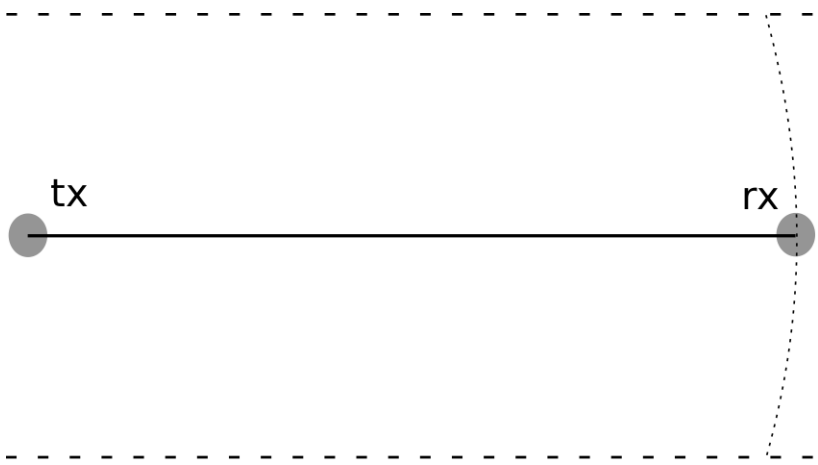

Figure 3.1: Direct path only.

target, estimating its position is a trivial matter. However, scenarios are possible where two rays alone yield more than valid solution, so more information might be required.

Surface and Bottom Reflections

The final scenario considered is shown in Fig. 3.3. We again assume we know the distance between the reflection point on the bottom of the ocean and the transmitter. Having this information the problem is straightforwardly solved.

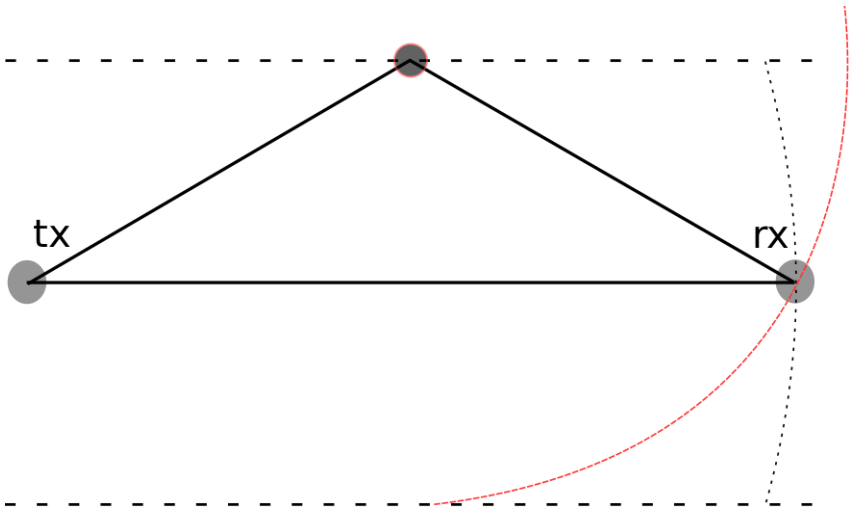

Figure 3.2: Direct path and surface reflection.

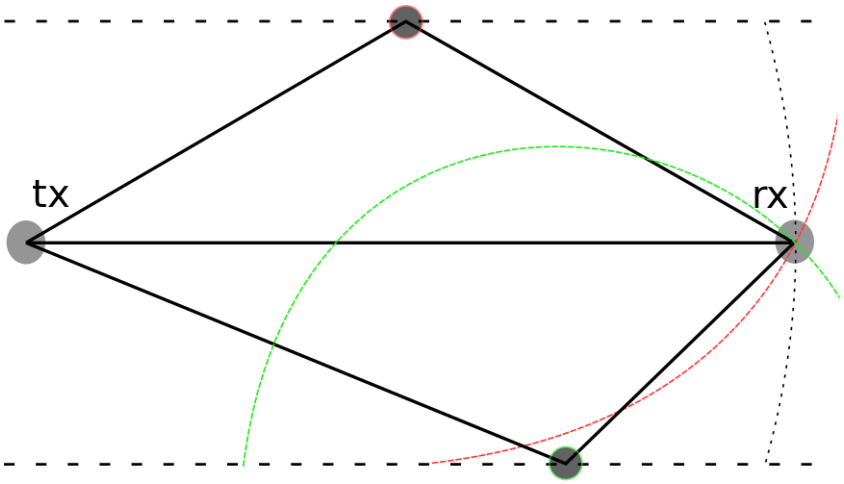

Figure 3.3: Direct path, surface reflection and bottom reflection.

Chapter 4

Transmission System

This chapter explains in detail the system developed to transmit acoustic signals underwater. The goal was to have a system capable of transmitting several different kinds of signals with different modulations. Specifically, it should be able to generate signals ranging from simple square waves to more interesting pseudorandom noise sequences. It should also allow for modulation in either BFSK or BPSK. It also had to, ideally, be easily and quickly configurable by an interface in a computer. Three essential parts make up the final transmitter: the signal generators, the modulators and the configuration interface. The rest of this chapter starts by reasoning the choice of platform and then describes each of the three parts separately.

4.1 Platform

Two platforms were considered to develop the transmitter: either an FPGA or a microcontroller. A combination of both was also considered as a possibility. FPGAs offer something that is hard to achieve in conventional microcontrollers: the ability to perform multiple tasks concurrently. On the other hand, all modules created have to be intensively tested, whereas microcontroller functionalities are ready to use, which leads to shorter development times. FPGAs clearly win when it comes to flexibility: adding or removing hardware modules is very simple, while microcontrollers, especially when acquiring large quantities, are very strict with respect to their existing peripherals. Although it was not a factor for this comparison, cost is usually much lower for typical microcontrollers than it is for FPGAs. This comparison is illustrated in Table 4.1.

Due to the fact that we were interested in a very application specific system, the flexibility of an FPGA made it the obvious choice. The specific FPGA used was the low-power iCE40HX1K from Lattice. The choice was made purely based on immediate availability, since we had an evaluation kit for that FPGA (the iCEblink40HX1K) ready to use. This board comes with a microcontroller to provide a USB interface that allows for easy flashing and communication with a computer. This is used to configure the transmitter modules.

The HDL of choice was Verilog.

Table 4.1: Comparison between FPGAs and Microcontrollers

	FPGA	Microcontroller
Concurrency	Easier	Harder
Development Time	Higher	Lower
Flexibility	Higher	Lower
Cost	Higher	Lower

4.2 Modulators

Both modulators are implemented in similar ways. The first step is to divide the global clock, which is of 3 MHz (measured in an oscilloscope, since the rated clock of the crystal in the board is of 3.3 MHz) in two other clocks: one of 20 kHz and another of 25 kHz. Taking advantage of our analog transmission chain's frequency response described 4.4, the BFSK modulator is simply a multiplexer controlled by the input signal. When we want to transmit a zero, the multiplexer selects the 20 kHz clock and when we want to transmit a one, it selects the other frequency.

The BPSK modulator acts in a very similar fashion. The multiplexer logic is precisely the same, except its inputs are now just a 25 kHz clock and the same clock after going through an inverter. This is effectively the same as shifting the clock's phase by 180°.

Block diagrams for both modulators are shown in Figs. 4.1 and 4.2.

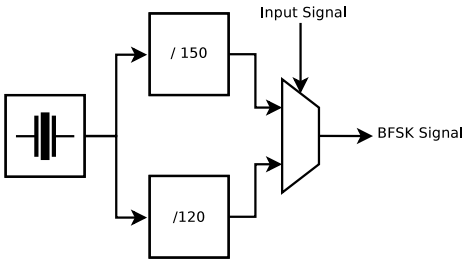

Figure 4.1: BFSK modulator block diagram.

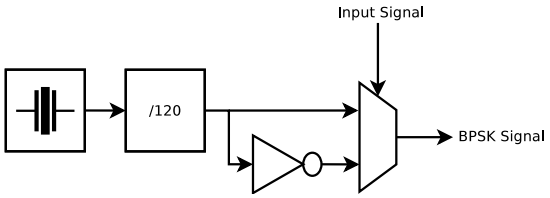

Figure 4.2: BPSK modulator block diagram.

4.3 Signal Generators

The FPGA generates three types of signals: a square wave, an MLS and a Gold sequence. The square wave generator is a simple counting module that outputs a one for a configurable amount

of time. As for the other two, they are direct implementations of the shift registers shown in Figs. 4.3 and 4.4. The polynomial for the first sequence is, thus:

$$x^4 + x + 1\,. \tag{4.1}$$

The initial state used in the shift register is $a_3a_2a_1a_0 = 1011$. The polynomials used to generate the Gold sequence are:

$$x^5 + x^2 + 1, \qquad x^5 + x^4 + x + 1\,. \tag{4.2}$$

Initial states used are $b_3b_2b_1b_0 = 01110$ and $c_3c_2c_1c_0 = 10110$;

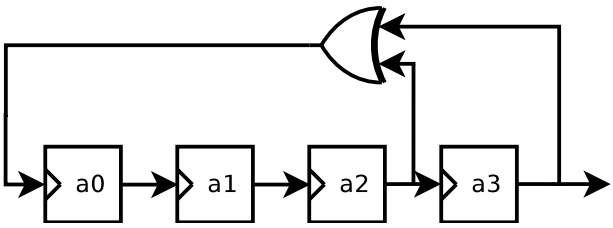

Figure 4.3: MLS generator block diagram.

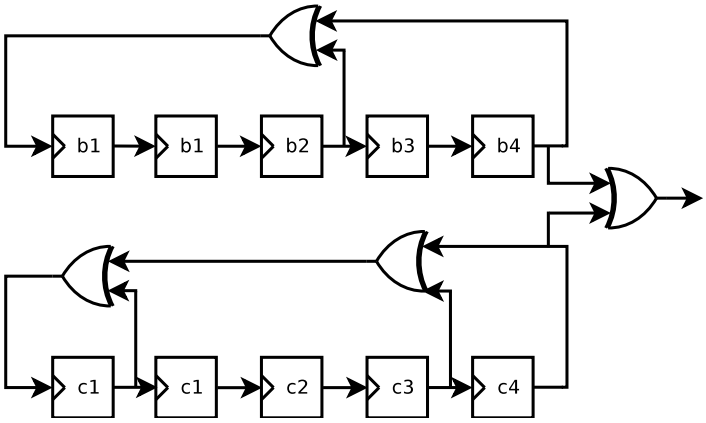

Figure 4.4: Gold sequence generator block diagram.

4.4 Analog Chain

The analog transmission system consists of a low pass filter, followed by a TDA2003 [27] car audio amplifier and finally a piezoelectric transducer. The overall frequency response is situated between 20 kHz and 30 kHz, making in appropriate to use square waves of fundamental frequencies situated in this range as carriers.

Chapter 5

Studied Signals

Good signal processing is one of the most determinant aspects for achieving good results when trying to localise a target using a transmitted signal and its multiple paths. For this matter, properly selecting a signal to use properly is of critical importance. Both the type of signal used along with the modulation chosen will greatly affect processing at the receiver. For instance, signals with bad cross correlation properties should be avoided. In this section we show a comparative study between three types of signals with two different modulations: a square wave (a pulse, actually), an MLS and a Gold sequence, modulated with either BFSK or BPSK. For this purpose a Simulink model was built to generate the signals of interest and modulate them. We then add four delayed copies of the signal together: one without delay, one with a 2 ms delay, one with a 3 ms delay and another with a 15 ms delay. We are using a sampling time of 1 µs for our signals. Since we'll be using 10 ms long signals, we expect correlation peaks at 10^3, 12^3, 13^3 and 25^3 sample times. To compute the delay, we simply subtract the duration of the signal (in samples) to the number of the sample where a peak occurred and multiply the result by the sampling time. It is also worth noting that the carrier frequencies were, for BFSK, 20 kHz for the symbol 0 and 25 kHz for the symbol 1. BPSK modulation uses a 25 kHz carrier.

5.1 Square Wave

The signal used was a pulse of 10 ms duration. The correlation result, using both BPSK and BFSK is very bad. Although some edges on the correct delays are identified, automatic detection, particularly in the presence of noise, would be extremely difficult. This result is not unexpected, since square pulses don't have good correlation properties. If we imagine the sum of two square pulses delayed with respect to each other but that overlap, it is easy to understand that the correlation between that sum and the original signal will not yield a good result. The only possibility of this working in a slightly acceptable way would be if the delay between pulses would be larger in time than the duration of the signal, which is not the case for all the three delays we are considering.

5.2 MLS

The sequence used has the generator polynomial $x^4 + x + 1$ with initial states of the shift register equal to 1011. The signal is 10 ms long with symbol time of 100 µs. Results for correlation using BFSK and BPSK are shown in Figs. 5.1 and 5.2. Results are, as expected, much better. Nevertheless, while we should see 4 distinct peaks, we see many more. In fact, some peaks that are unwanted are bigger that the ones we're looking for. This alone would cause large errors in detecting the arrival times of different rays. The reason is that, even though we're using a MLS sequence, its cross correlation properties are still not as good as we need them to be.

5.3 Gold Sequence

The polynomials used to generate the gold sequence were $x^5 + x^2 + 1$ and $x^5 + x^4 + x + 1$ with initial states 01110 and 10110, respectively. The signal length and symbol time are the same as above. Results are shown in Figs. 5.3 and 5.4. Gold sequences perform better than the previous maximum length sequence due to their very good cross correlation properties. With either modulation, the peaks at the expected sample times are clear. Using BPSK yields better results than using BFSK, which might be justified by the fact that BPSK is a linear modulation (the bandwidth of the modulated signal is equal to that of the baseband signal) and BFSK is not.

5.4 Conclusion

The best option, from the simple analysis presented, is clearly to use Gold sequences with BPSK modulation. While other sequences known for their good cross correlation properties (CSS [24] or Kasami, for example) could be interesting to study, the gain in using them would not be significant considering that Gold sequences are extremely easy to implement in an FPGA, our platform of choice. Hence, from now on, we assume that the transmitted signal of choice will always be a Gold sequence, modulated using BPSK.

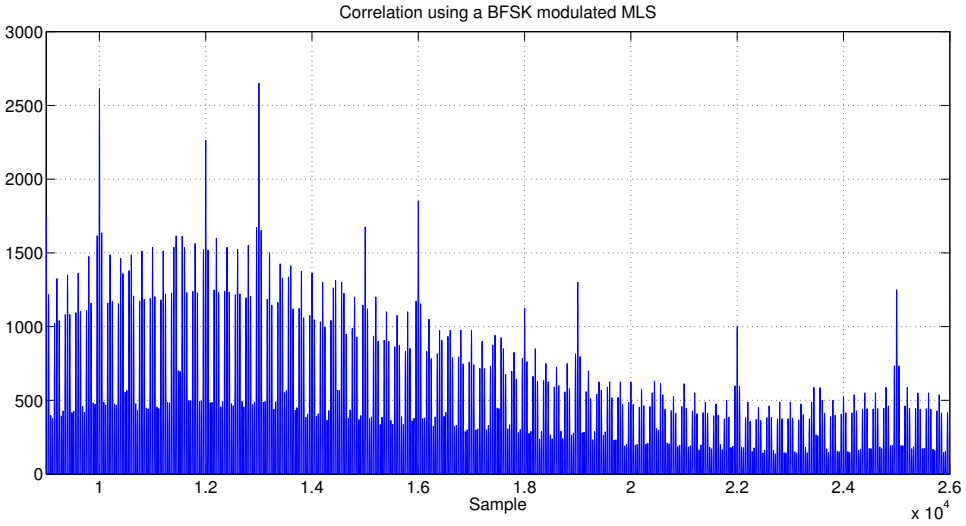

Figure 5.1: Correlation using a MLS with BFSK modulation.

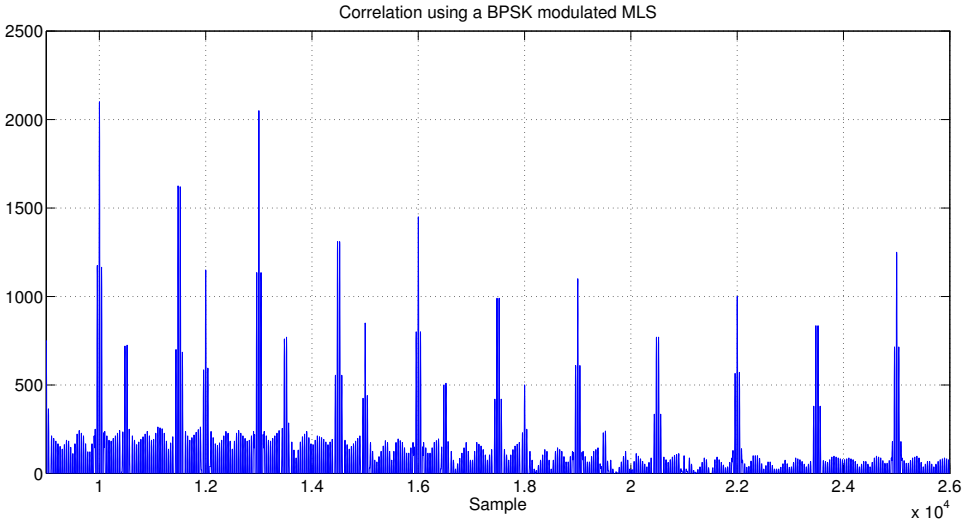

Figure 5.2: Correlation using a MLS with BPSK modulation.

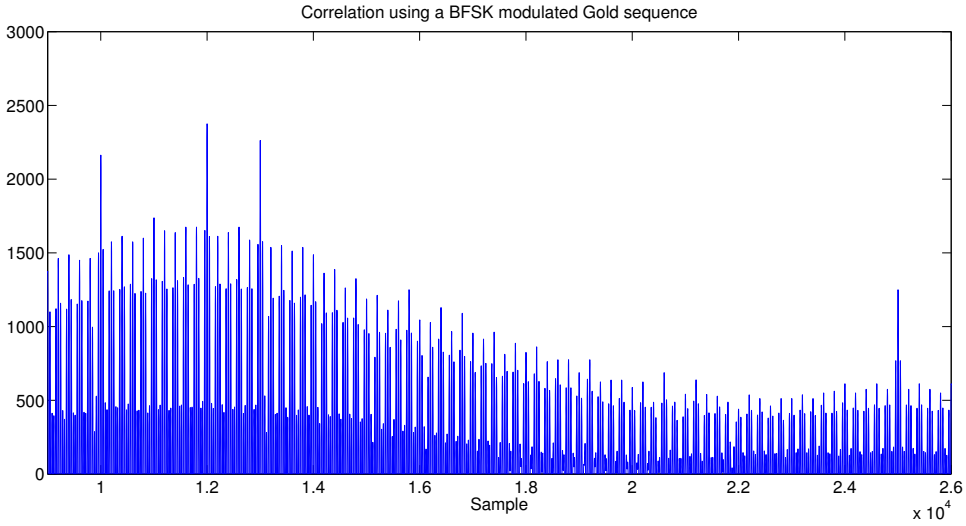

Figure 5.3: Correlation using a Gold sequence with BFSK modulation.

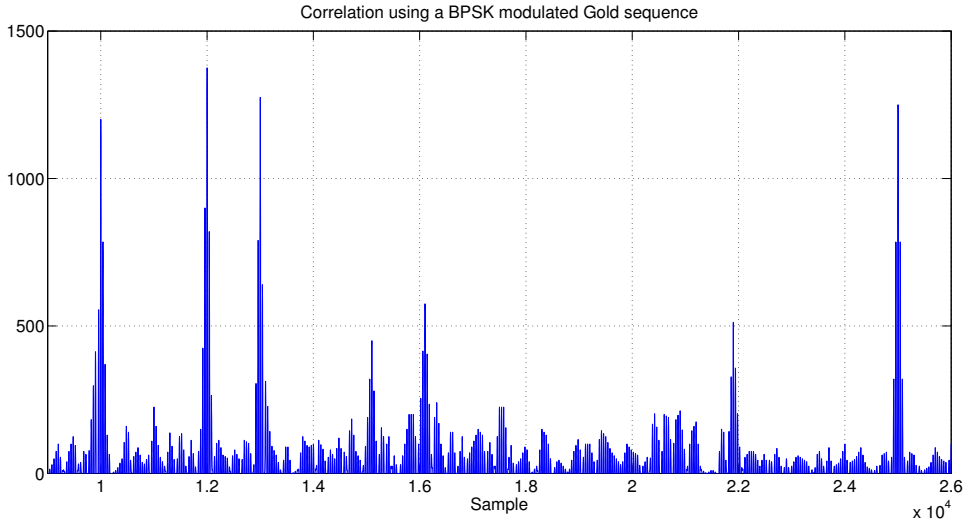

Figure 5.4: Correlation using a Gold sequence with BPSK modulation.

Chapter 6

Simulator

Evaluating potential localisation algorithms and signal conditioning techniques by going out to the ocean with a real AUV is clearly not a viable option. With this in mind, one of the main goals of this work was the development of a simulator. The purpose of this simulator was to, given a signal to be transmitted, get an estimation of what the signal would be at the receiver. This is accomplished by finding eigenrays through ray tracing. The delays of each eigenray are individually computed as well as the estimated losses suffered by the signal, so that the result is the following function:

$$y(t) = \alpha_0 x(t-t_0) + \alpha_1 x(t-t_1) + \ldots + \alpha_n x(t-t_n)\,, \tag{6.1}$$

where t_n is the delay of ray n and α_n the attenuation suffered by ray n. The inputs to the simulator would then be environmental parameters such as sound speed profile, water temperature and wind speed, and the outputs would be the attenuations and delays of the eigenrays found.

6.1 Architecture

The simulator is constituted by two distributed components: an underwater acoustic simulator, which acts as a transmitter, and an AUV simulator. Both components are written exclusively in C and communicate through BSD sockets. The reason for distributing is that the simulations can get very heavy on the transmitter side, so this allows for running this component on a remote server.

The AUV simulator keeps track of its real position and the one of the transmitter. These are assumed constant during one simulation. The AUV communicates with the transmitter through a set of commands, the most relevant of which sends its real position to the transmitter. The acoustic simulator then performs intensive ray tracing to find the eigenrays. When finished, it sends the eigenray information (delay and attenuation) back to the AUV. With this information plus an input signal, the AUV simulator builds the received signal that can then be processed. This sequence is depicted in Fig. 6.1.

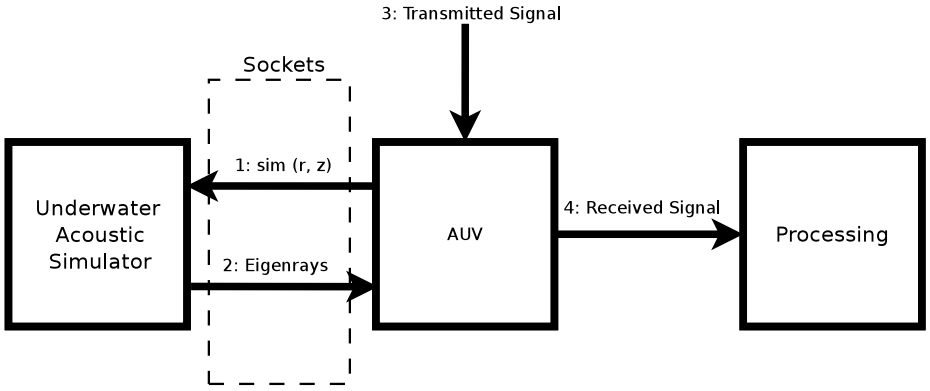

Figure 6.1: Simulation process.

6.1.1 Acoustic Simulator

The underwater acoustic simulator acts as a server and takes commands from incoming clients. When it receives a position from the client, it performs ray tracing starting at wherever we decided to place the transmitter. The equations implemented are the ones shown in (2.11) and (2.12). We rewrite them here in a discrete form:

$$r_{i+1} = r_i + c(z_i)\xi_i ds, \qquad \xi_{i+1} = \xi_i - \frac{1}{c(z_i)^2}\alpha_i, \tag{6.2}$$

$$z_{i+1} = z_i + c(z_i)\zeta_i ds, \qquad \zeta_{i+1} = \zeta_i - \frac{1}{c(z_i)^2}\beta_i, \tag{6.3}$$

where α_i and β_i are the partial derivatives of c with respect to r and z, respectively. It should be noted that the derivative of c with respect to r is, according to our assumption that the speed of sound is horizontally constant, always zero. The direct implication is that $\xi_{i+1} = \xi_i$ for every i. We further more keep the time each ray has travelled, integrating it at each step according to the following relation:

$$t_{i+1} = t_i + \frac{ds}{c(z_i)} \tag{6.4}$$

Every ray is traced until either:

- It has horizontally passed the AUV;
- It is within the AUV's receive radius (explained shortly).

The simulator uses the following parameters: **vertical step, arclength step, receive radius, transmitter position, angle span, angle spacing**. A short explanation of each one of them follows. It should be noted that there are actually more parameters we need to define in (such as environmental parameters), but we're considering only the ones that directly affect performance here.

Arc Length Step

This parameter is the step in arc length to use in the ray tracing equations.

Vertical Step

This defines the vertical step in the water. In other words, the depth divided by this step equals to the number of water rows we consider. It is used solely for calculating the derivative of the sound speed profile.

Receive Radius

The receive radius is the radius that defines the circumference centered at the receiver position. It is used to know when to stop ray tracing, since rays will rarely exactly hit the receiver. When any ray enters this circumference, we decide it is an eigenray.

Angle Span

This parameter, defines the angle span for which we want to simulate ray tracing. This concept is better illustrated in the Fig. 6.3.

Angle Spacing

The angle spacing is the step in angle between consecutive rays launched. Another way to look at it is that the total number of rays simulated is equal to the angle span divided by the angle spacing.

These parameters should be carefully chosen. Although the concern here is not to have an extremely fast simulator, poorly chosen parameters can lead to extremely heavy simulations without any justifiable accuracy gain. An example of good parameters for shallow water, according to several simulations performed, is an arc length step of 0.1 metres, a vertical step of 0.1 metres and a receive radius of 2 metres. The only parameter that was found to lead to very poor ray tracing results when increased was the arc length step. Bigger steps obviously lead to much faster simulations, but they degrade in precision at a rate that might not be acceptable. This precision loss gets bigger the longer the rays are, as shown in Fig. 6.2. The blue ray is traced with a step of 0.1 metres and the red ray is traced with a step of 10 metres. At the third surface reflection, the rays are around 20 m apart.

6.1.2 AUV Simulator

The AUV simulator is built as a simple shell that takes commands from the user. It fundamentally allows a user to define its real (r,z) coordinates and then connects to the server (the acoustic simulator described above) to get the eigenrays it would receive in a real situation. From this information plus the signal to be transmitted it generates the received signal. This is the end of

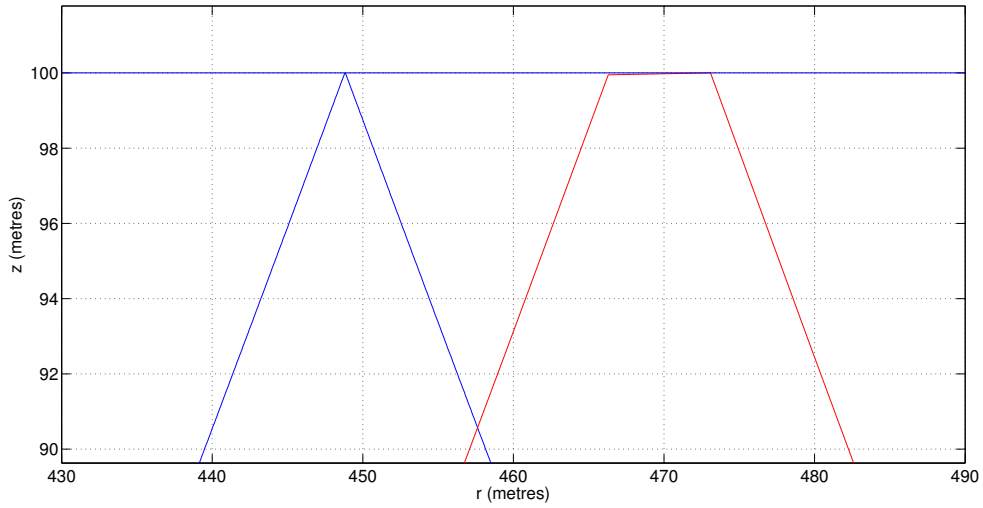

Figure 6.2: In blue we see a ray traced with a step of 0.1 metres. The same ray is traced in red with a step of 10 metres.

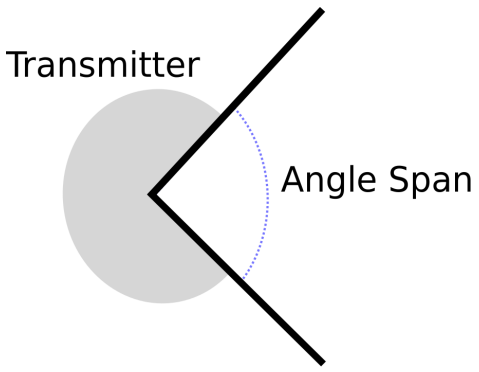

Figure 6.3: Illustrating the Angle span concept. Rays are traced only inside the depicted angle.

the simulation part, and after this there is a section in the code that allows a developer to analyse the received signal. The intent is that whatever code goes in here to try and estimate the AUV's location can be almost directly ported into a real system. A set of commands implemented allows to set the AUVs position, perform remote simulations on the server and estimate position, among other auxiliary functions.

The simulation parameters of this part of the simulator directly affect the performance of the localisation algorithm implemented on top of it. Since this algorithm is to run on an embedded system, preferably online, it is important to understand how we can modulate these parameters according to our need for precision and how we wish to balance it with runtime. The parameters used to configure the AUV simulator are essentially the ones described before for the other part of the simulator. In the next chapter we will see how a set of parameters affects localisation performance.

Chapter 7

Localisation Algorithm

This chapter presents the relative localisation algorithm developed. It was built on top of the simulator with the goal of having it ready to deploy in a real platform very quickly with very few alterations. The inputs of the algorithm are the transmitted and received signals along with the position of the transmitter and the environmental parameters. Its output is the estimated (r,z) coordinates of the AUV.

We first explain the fundamental idea behind the algorithm, after which we describe its implementation in detail. It is important to understand how the algorithm parameters affect performance and accuracy, so we dedicate a section to this topic.

One of the main goals of having a simulator is to be able to test an algorithm such as this one intensively and get an idea of what kind of errors we mights expect from using it. Section 7.3 shows results for simulations performed and analyses them statistically.

Finally, we end with a discussion of how this algorithm may fail and how it could be improved.

7.1 The Algorithm

The basic idea behind the localisation algorithm is that if, upon receiving a certain signal we can determine how many different eigenrays we got and their travel times, we might be able to find a new set of rays that is very similar to the one we received. Our approach is the following: we take the received signal and cross correlate it with the transmitted. This should preferably be done using FFTs for efficiency (we have done so in this simulator using the library described in [28]).

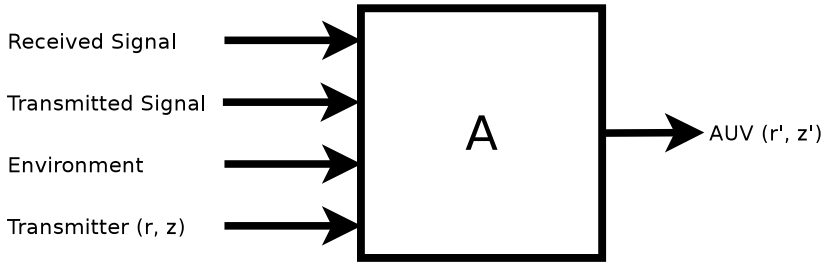

Figure 7.1: Localisation algorithm.

Having the cross correlation, the next step is to identify its peaks. These correspond to the time the eigenrays have travelled, which is all the information we need to proceed.

Let $\mathscr{E}$ be the set of eigenrays identified, for which we know only the time travelled. We should also be able to retrieve information about the power received, but we ignore that for now. The goal is to try and create a new set $\mathscr{E}'$, knowing only the transmitter position, such that:

- For every ray in $\mathscr{E}$ with delay d, there is a ray in $\mathscr{E}'$ with delay d' such that $|d - d'| < d_t$, where d_t is a configurable threshold;

- The coordinates $\mathbf{x_i} = (r_i, z_i)$ and $\mathbf{x_j} = (r_j, z_j)$ found for each ray i and j in $\mathscr{E}'$ should be such that $\mathbf{x_i} - \mathbf{x_j} < \mathbf{x_t}$ for all i and j , where $\mathbf{x_t}$ is a configurable threshold.

It should be noted here that the set $\mathscr{E}'$ is built by the receiver, so the coordinates of each ray in it, whether it is an eigenray or not, are known. When these conditions are met, we assume we found a set that very closely resembles $\mathscr{E}$, which is the same as saying we found the eigenrays received, and so the problem is solved. What we're ultimately looking for is, thus, an algorithm to find $\mathscr{E}'$, if there is one. Having it, we simply average the position of every ray in it and that is our estimate.

The proposed algorithm to do this is as follows, where d_n is the time travelled by ray_n:

Parameters: d_t, $\mathbf{x_t}$, *ray_amount*
Inputs: $\mathscr{E}$, *no_rays*
Initialisation: $n = 0$, $i = 0$, $cur_d = 0$

for $n = 0$ until *ray_amount* :
  trace ray_n until $|d_n - d_{cur_d}| < d_t$
  $cur_d = 1$;
  $visited_i = 0$ **for** all i
  for $i = 0$ until *ray_amount*:
    if $i \neq j$ **and** $visited_i = 0$
      trace ray_i until $|d_i - d_{cur_d}| < d_t$
      if $\mathbf{x_i} - \mathbf{x_n} < \mathbf{x_t}$
        if $cur_d = (no\_rays - 1)$
          $solution = 1$
          end
        else
          $visited_i \leftarrow 1$
          $cur_d = cur_d + 1$
          $i = 0$
          reset_rays
        endif
      endif
    endif

To be clear, *ray_amount* is a parameter defining how many rays we want to test inside the *angle_span* (defined in the previous chapter), whereas *no_rays* is the number of rays in $\mathscr{E}$. The same algorithm is shown in a flowchart in Fig. 7.2. The averaging part is not shown in either description, though the rays marked as visited are the ones that we use to calculate it.

7.2 Parameters

Although some parameters were discussed on the previous chapter, we now go over the fundamental ones we can control directly in the algorithm.

- Arc Length Step;
- Ray Amount;
- Intersection Threshold.

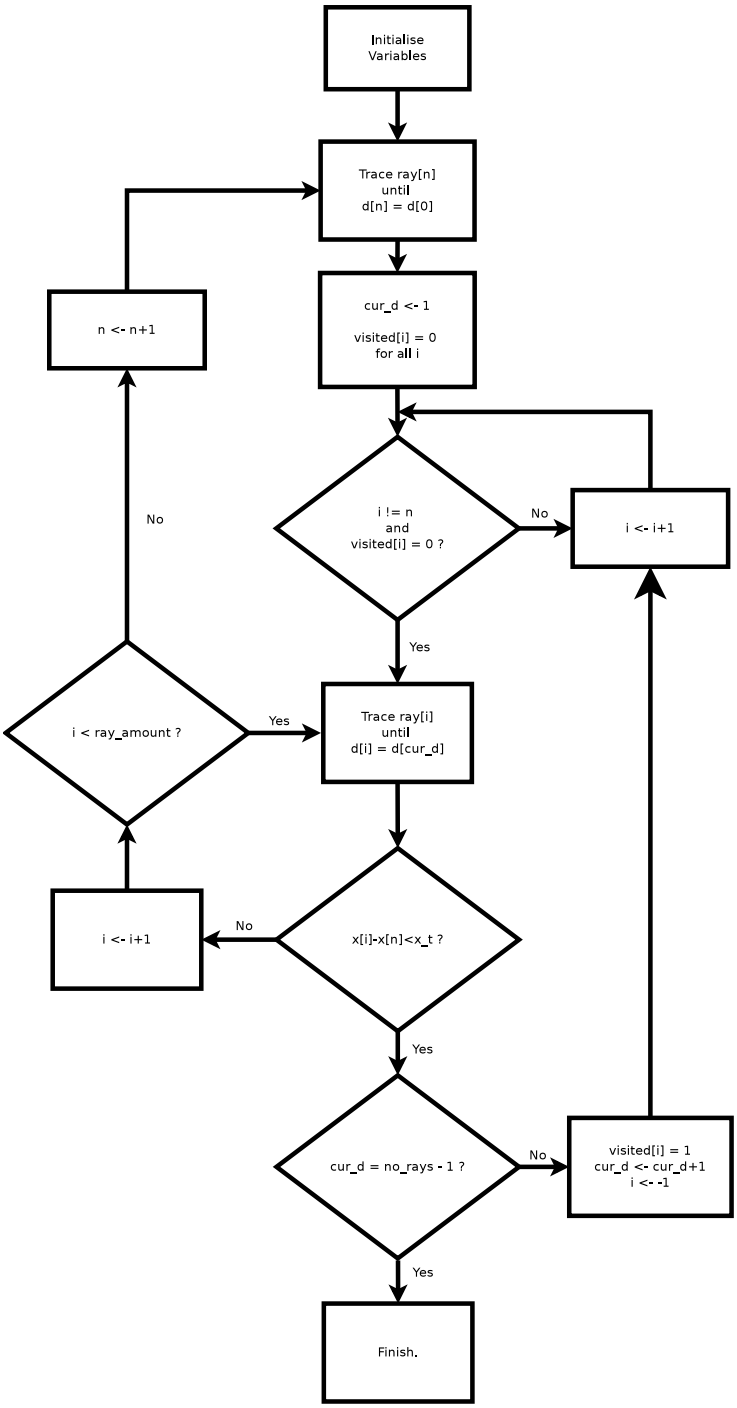

Figure 7.2: Relative localisation algorithm.

The first two on the list affect mostly performance, while the last one will affect mostly accuracy. Note, however, that changing the intersection threshold expecting accuracy won't work unless the other parameters are adjusted. In other words, it is not possible to keep runtime and improve accuracy. The arc length step is the *ds* step in our ray equations shown in (6.2) and (6.3). Is corresponds to how far we trace the ray each time we take a new step. The intersection threshold determines how close together two rays must be so we consider them to be in the same place. It is the x_t in our algorithm description above. It is easy to see how these two parameters correlate: If we have a very big arc length step, in one step we might trace two rays over each other without detecting it. The consequence is that if we want to have small intersection thresholds (better accuracy), we need to use small arc length steps (higher runtime). The ray amount parameter also has a role in this. It tells us how many rays we want to trace in our angle span. If it is low, we have fewer rays to trace, so performance is better. However, this also means that the launch angle between two consecutive rays is large. This means that, the farther the rays travel, the farther apart they will be. This leads to the same issue as using large arc lengths steps: we might be losing ray intersections, which will ultimately result in inability of reaching a solution. The problem grows with distance: the farther apart the receiver is from the transmitter, the more rays we need to trace in order to get a solution.

7.3 Results

In this section we present the results obtained with our algorithm running on top of the simulator described on the previous chapter. Different combinations of the parameters were used for several different AUV positions to get relevant information. Shallow water, of 100 metres of depth, was assumed, so the sound speed profile was constant throughout the entire simulation. The transmitter was placed at $(r,z) = (0,65)$. A box with diagonal defined by the vertices at $(50,10)$ and $(200,90)$ was defined and, for each combination of parameters, we ran a simulation for every point inside this box (with a step of 2 metres) and tried to estimate the position with our algorithm. These numbers add up to 3116 AUV coordinates simulated. We simulated for every combination of $ray\_amount = [30,100,200]$ and $ds = [10,1]$, while keeping the intersection threshold constant at 3 m, so in total 18696 simulations were performed, taking approximately 7 hours. Table 7.1, in which we included information about mean and maximum errors as well as mean runtime, summarises the results.

This table shows some interesting results. First of all, the column Max (r,z) shows that maximum coordinates for which the algorithm found a solution (the third column shows the percentage of points for which the algorithm failed to find a solution). For only 30 rays, it stopped being able to find solutions quite a bit before the end of our box. This is coherent with our discussion in the previous section. An apparently weird result is that mean algorithm runtime was shorter for 100 rays than it was for 30 rays. This is not particularly significant since the percentage of fails is much bigger for the first, so a comparison makes little sense, since we have a large difference in amount of information. For all other comparable cases, runtime increases with either decreasing

Table 7.1: Simulation results.

ds	**Ray Amount**	Fail %	Max (r,z)	Mean e_r	Mean e_z	Max e_r	Max e_z	Mean t
10	30	83%	$(132,10)$	5.3	2.5	27.1	37.6	1.3018
10	100	55%	$(200,90)$	6.0	2.5	71	26.4	1.2837
10	200	46%	$(200,90)$	6.0	2.5	28.1	78.9	1.3119
1	30	79%	$(187,84)$	9.6	0.5	28.3	29.9	1.2419
1	100	21%	$(200,90)$	9.4	1.0	30.9	73.0	1.3705
1	200	3%	$(200,90)$	9.9	1.8	30.9	73.4	1.6407

ds or increasing ray amount. Another fact that appears troubling is the maximum errors. However, comparing with the mean ones, these are always clear outliers, which we disregard in Figs. 7.3 and 7.4, where we show the histograms for the errors on r and z of the last row. We show the histograms of this row (using 100 bins) because it is the one with smallest percentage of fails. Both distributions are similar and resemble normal distributions, which makes it easy for integration in a Kalman filter. However, and this is also noticeable in Table 7.1, the distribution for e_r is not centered on 0, as opposed to the one for e_z. This can be explained by a combination of several facts. Notice again in our result table that, when we decrease ds, the mean e_r **increases** while the mean e_z **decreases**. It is interesting to realise that, from our algorithm construction, a negative error in r (considering we're measuring error as subtracting the estimated position from the real one), corresponding to the majority of rays in our set $\mathscr{E}'$ being beyond the real position, is very difficult to happen. The same is not true for z. As we increase ds, however, this situation becomes more likely to reverse itself. This is easier to understand if we remember that we are considering shallow water, so sound rays travel in straight lines. This hypothesis is difficult to prove by simulation, though, since a ds much bigger than 10 will lead to larger fail rates. A smaller simulation with $ds = 15$ and ray amount of 100 was performed, however. The algorithm found 96 valid solutions, and the average error on r was 3.3 m, against the 3.8 m found in z, which is in agreement with our idea.

7.4 Shortcomings and Optimisations

Although the algorithm performed in a satisfactory way in simulations, it has some issues and some details that could be changed in order to increase performance. We go through these details in this section.

The first problem encountered has to do with a symmetry that may happen. The scenario is depicted in Fig. 7.5. As an example, imagine that the water depth is 100 metres, and that the transmitter is placed at around 50 metres of depth. Say, for instance, that the AUV is located at $(100,70)$. If we are dealing with **shallow water**, where the sound speed is approximately constant, rays are straight lines. The fact that the transmitter is placed at approximately half the depth will lead to the existence of two sets $\mathscr{E}'$ that fulfill the conditions stated previously. The direction in

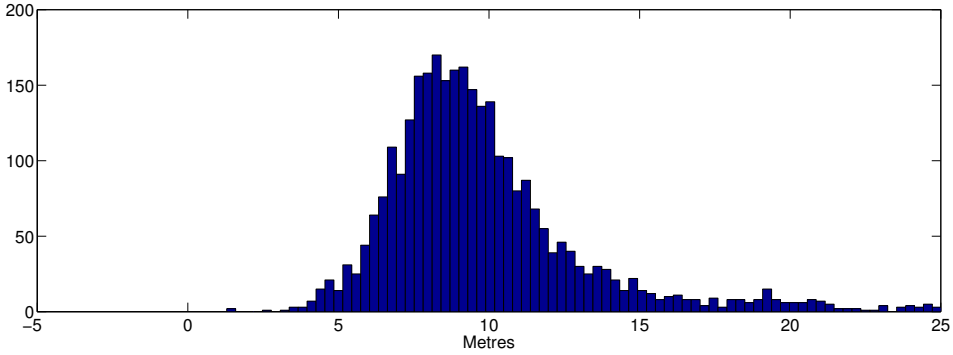

Figure 7.3: Histogram for e_r.

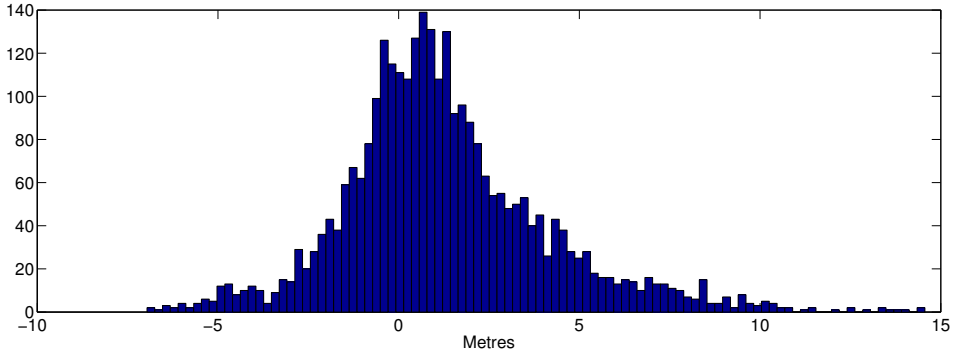

Figure 7.4: Histogram for e_z.

which we choose to trace rays (clockwise or counterclockwise) will lead to different estimations. In this scenario, the algorithm can output a solution close to $(100, 30)$. In reality, this might be seen as not a shortcoming of the algorithm, but rather an effect that there's no way around of. To see this, consider that a receiver at $(100, 70)$ would indeed receive the exact same eigenrays as a receiver at $(100, 30)$. A workaround for this problem is to simply avoid the symmetry by not placing the transmitter in the middle of the water column.

Another issue with the algorithm, and the simulator itself, is the fact that it assumes a flat bottom, which will seldom be the case. Applying this algorithm directly without account for this fact will certainly lead to poor results. The hard part, however, is not accounting for irregular bottoms, but rather having some sort of map of the bottom of the region. Even so, since the deflection angle of a ray will directly depend on the shape of what it hits, if the bottom is very irregular, very small steps need to be used to make sure that we're not reflecting rays off the bottom in a completely wrong way. At this point, we're talking about keeping very complex maps and using very small steps. These two factors combined would lead to an algorithm that would clearly be too expensive computationally to be valuable. In deep water, this problem can be partially avoided by placing the transmitter relatively close to the surface to guarantee that, even if any ray reflects of the bottom (which may not happen), it will reach the receiver with very little power. As for shallow water, the only way option is to keep a map of the bottom and allow for accuracy loss.

A third issue with the algorithm is the fact that, at times, it does not find a solution. This has to do mostly with the fact that it tries to find a complete set $\mathscr{E}'$. Many times, however, it finds a partially full set but ignores it. This is obviously easy to deal with, but accuracy will be lost by allowing smaller sets. The fact that the algorithm does not always find a complete set is related to the following two issues:

- Our model is not perfect;
- Our parameters have to be rough for runtime to be acceptable.

To finalise, we note that there is one optimisation that would possibly lead to much faster results and help us use much tighter parameters. In our algorithm, we're starting ray tracing from scratch **everytime**. In a realistic scenario, however, an AUV will perform two consecutive localisations in two not very different coordinates. This leads to the idea that when we finish estimating its position once, we should keep the information about the rays obtained. When running the algorithm again, ray tracing can just pick up where it was left off, tracing forward or backwards depending on the direction the AUV is moving in (we assume this is known by, say, a DVL that estimates velocity). If the flat is bottom or there are no bottom reflections, this will be valid whether or not the AUV has left the vertical plane we're tracing in, since in this circumstance rays are traces equally for every vertical plane.

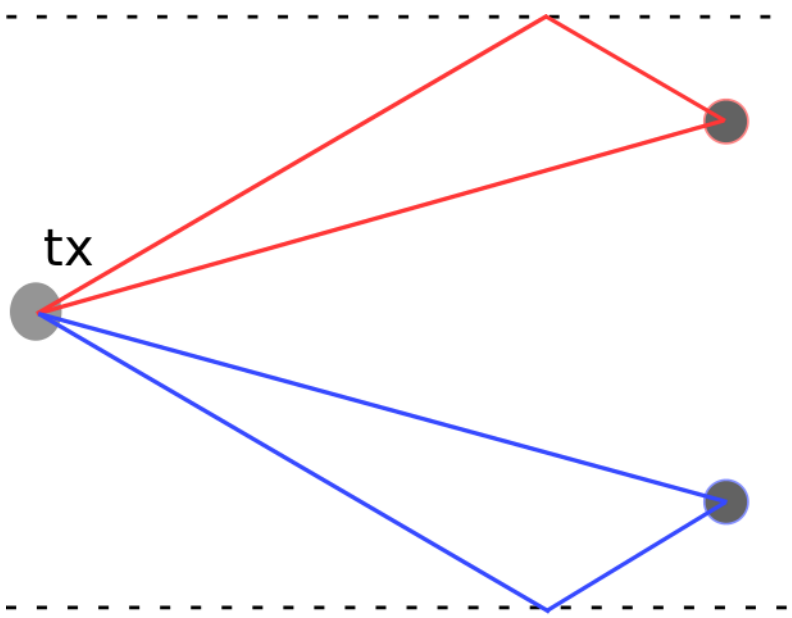

Figure 7.5: Symmetry issue. The blue rays appear to their receiver exactly the same as the red rays to theirs.

Chapter 8

Field Tests

This section describes some of the tests performed using the transmission system detailed in 4. One test was performed in a pool at FEUP and two others at the Leixões Marina in Porto, Portugal. We now proceed to explain the purpose and results of each one. For every test, the SR-1 hydrophone [29] was used. This hydrophone has a sampling frequency of 101 562 Hz.

8.1 22/May/2014

At this point the FPGA was programmed only to modulate square pulses or maximum length sequences in BFSK. The purpose of this test was to validate the functionality of the transmitter system. Note that, since a pool has a very small amount of water when compared to the ocean, multipath will be overwhelming. In fact, for a transmitted signal of 10 ms, a signal of around 70 ms was registered. Figure 8.1 shows the FFT (using 2048 points) of the recorded signal. Since we're using BFSK with 20 kHz and 25 kHz, and accounting for the sampling frequency of the hydrophone, the result is exactly what we expected, thus confirming that the system is working.

8.2 Second Test - 3/Jun/2014

This test was performed at the Marina, but before Gold sequences and BPSK were implemented. The only recorded signal was an MLS modulated with BFSK. Though the transmission worked, as again proven by the spectrum of the received signal, the conditions were very poor. Specifically, transmitter and receiver were placed around 10 m apart and the water depth at that location was, in this day, of around 3 m. Besides this, there were walls and other obstacles that caused too many reflections to use the collected data to effectively perform localisation. Furthermore, there is a big problem with this approach: there is no synchronization between transmitter and receiver. That is, after collecting the data, there is no way of knowing when the signal is transmitted, so it is very hard to determine the delays of the arrivals. Part of the correlation of the recorded signal with the transmitted one is reproduced in Fig. 8.2. As it is possible to notice, combining the poor signal

used for transmission with the very bad environmental conditions, the result is not adequate to input to our algorithm.

8.3 Third Test - 20/Jun/2014

This test was, as the previous, performed at the Leixões Marina. However, the tests failed and we didn't manage to record any signal. We tried to transmit both MLS and Gold sequences modulated with BFSK and BPSK. The recording showed lots of noise, but none of the frequencies we were looking for. The system was later tested once more in the pool and worked, but with a lot more noise than in the previous tests. The best hypothesis we currently have for this is that, since the filter used was changed between tests, some electrical issue such as a bad ground connection was the cause. The tests are to be repeated as soon as possible.

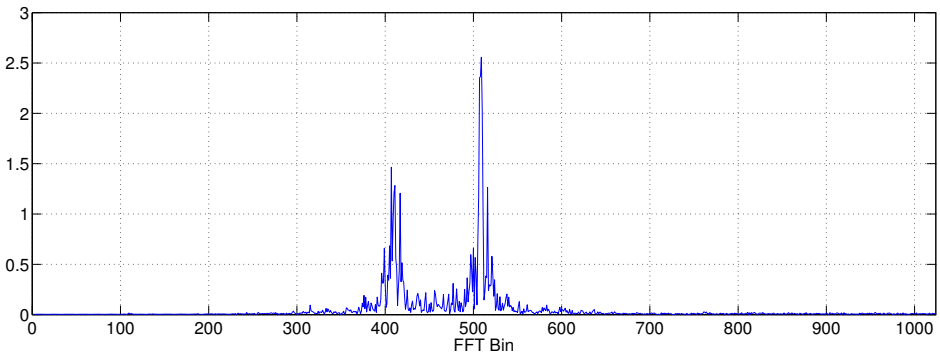

Figure 8.1: Spectrum of the received MLS sequence modulated in BPSK in the pool.

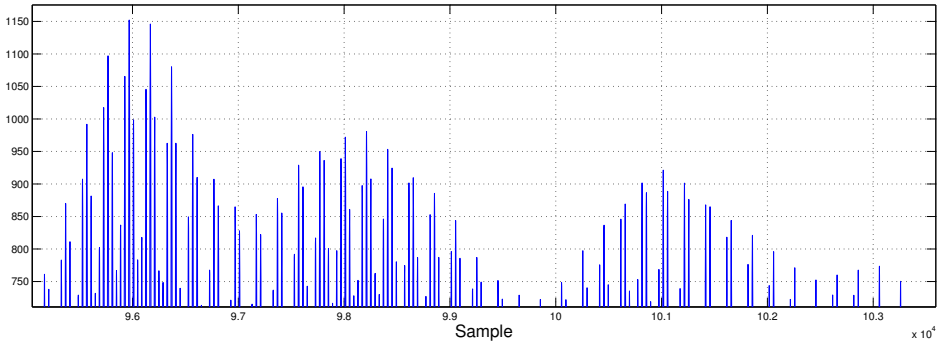

Figure 8.2: Section of the correlation for an MLS transmitted using BFSK at the Marina in Leixões.

Chapter 9

Conclusions

We now conclude this thesis by analysing how it went according to the goals we set in the beginning. We explore what is new in this work, what could be better and what was successfull. To finalise, we consider what should be done next by anyone who starts where we left off.

9.1 Final Assessment

Our ultimate goal when defining the topic for this thesis was clear: building a relative localisation system for underwater vehicles that was based on the multiple paths sound takes on this medium. To this end, we developed a transmission system, studied different types of s ignals evaluating their benefits in terms of signal processing, developed an underwater acoustic simulator and, finally, a localisation algorithm. This acoustic simulator is something new. There are codes for performing ray tracing (such as Bellhop [1]) as there are ways of detecting eigenrays (interpolation being the amongst the most common of them [2, 6]). However, no complete simulator consisting of the components presented that automates both tasks together to ease algorithm development was found to exist. This can be, after further development, very useful for future research in the area, as it makes algorithm testing very quick. The localisation algorithm presented was also completely developed from scratch, and is furthermore completely written in C to allow for almost immediate deployment to a real AUV. Although no actual complete system was put together and developed (since using/building a real AUV would not have been viable), almost every piece to do so was presented in this thesis. Furthermore, although no sensor fusion system was implemented, since there was not a chance to evaluate the algorithm presented in a real platform and, as such, we have no other sensor information, a statistical analysis of the algorithm was performed, showing that it is possible to fit our estimation error in a satisfactory way to a Gaussian distribution. This makes integration in a Kalman filter, which is likely to be present in any good navigation system, extremely easy.

As a negative point, the real data collected was not very good and did not allow for evaluating the algorithm in a real scenario. With experimental validation missing, everything else might be called into question. We need to realise, however, that this is a system intended to work in the open

ocean, a medium hard to access. There was not an opportunity, during the course of this work, to perform any tests in such an environment.

Despite the negative point, since interesting results were achieved and new ideas were presented, the overall balance is positive.

9.2 Future Work

To conclude, we now present some considerations on the steps to take next. The first task would be to collect real data on the ocean. The water should be deep enough to avoid bad bottom reflections, and, more importantly, a synchronization method between transmitter and receiver should be developed. A possible way of doing this with an AUV is to have it surface periodically, get a GPS signal to record the current time and afterwards synchronize receptions with an internal RTC. Following this idea, it would now be important to implement the algorithm in an AUV and test it for known coordinates. A way of doing this would be using a validated LBL system at the same time and compare results.

Another important aspect to improve upon is the models used by the simulator. Although it does account for losses, for example, it does so roughly, and the algorithm disregards this information. While geometrical spreading losses might be somewhat accurate, reflection and absorption losses are certainly not, since these losses depend on a very big number of factors difficult to estimate. Obviously, the better the model we use for the simulator, the more realistic the results any algorithm yields will be. Other things of interest are worth exploring in the simulator. One of them is the fact that it does an intensive ray search. A better approach would be the one of interpolation presented in [6], that would likely lead to largely reduced simulation times.

References

[1] Orlando Camargo Rodríguez. *General description of the BELLHOP ray tracing program (June 2008 release)*. Faculdade de Ciências e Tecnologia, Universidade do Algarve, June 2008.

[2] Finn B. Jensen, William A. K Kuperman, Michael B. Porter, and Henrik Schmidt. *Computational Ocean Acoustics*. Springer, 2nd edition, 2011.

[3] Claude C. Leroy and François Parthiot. Depth-pressure relationships in the oceans and seas. *Acoustical Society of America*, 1998.

[4] Robert J. Urick. *Principles of Underwater Sound*. McGraw-Hill, 3rd edition, 1983.

[5] C.-T. Chen and F. J. Millero. Speed of sound in seawater at high pressures. *Acoustical Society of America Journal*, 62:1129–1135, November 1977.

[6] Joaquín Aparicio, Ana Jiménez, Fernando J Alvarez, Jesús Ureña, Carlos De Marziani, and Cristina Diego. Modeling the behavior of an underwater acoustic relative positioning system based on complementary set of sequences. *Sensors (Basel, Switzerland)*, 11(12):11188–205, January 2011.

[7] Paul C. Etter. *Underwater Acoustic Modeling and Simulation*. CRC Press, 2013.

[8] L. M. Brekhovskikh and Yu. P. Lysanov. *Fundamentals of Ocean Acoustics*. Springer, 2003.

[9] Jens M. Hovem. *Modeling and Measurement Methods for Acoustic Waves and for Acoustic Microdevices*, chapter Ray Trace Modeling of Underwater Sound Propagation, pages 573–598. InTech, 2013.

[10] R. E. Francois and G. R. Garrison. Sound absorption based on ocean measurements. part ii: Boric acid contribution and equation for total absorption. *Acoustical Society of America*, 1982.

[11] H. W. Marsh and M. Schulkin. Absorption of sound in sea-water. *Radio and Electronic Engineer*, 1963.

[12] R. Coates. An empirical formula for computing the beckmann-spizzichino surface reflection loss coefficient. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, 1998.

[13] Milica Stojanovic. Underwater acoustic communications: Design considerations on the physical layer. *Proc. IEEE / IFIP Fifth Annual Conference on Wireless On demand Network Systems and Services (WONS 2008)*, 2008.

[14] M. Stojanovic. *Wiley Encyclopedia of Telecommunications*, pages 36–47. John Wiley & Sons, 2003.

[15] R. J. Urick. Multipath propagation and its effects on sonar design and performance in the real ocean. In *Aspects of Signal Processing - Part 1*, 1976.

[16] H. Lichte. On the influence of horizontal temperature layers in sea water on the range of underwater sound signals. *Tracor Sciences & Systems*, 1919.

[17] Michael J. Buckingham. Ocean-acoustic propagation models. *J. Acoustique*, 1992.

[18] C. L. Pekeris. Theory of propagation of sound in a half-space of variable sound velocity under conditions of formation of a shadow zone. *Acoustical Society of America*, 1946.

[19] P. Corke, C. Detweiler, M. Dunbabin, M. Hamilton, D. Rus, and I. Vasilescu. Experiments with underwater robot localization and tracking. In *Robotics and Automation, 2007 IEEE International Conference on*, 2007.

[20] John J. Leonard, Andrew A. Bennett, Christopher M. Smith, and Hans Jacob S. Feder. Autonomous underwater vehicle navigation. *MIT Marine Robotics Laboratory*, 1998.

[21] Roland Siegwart, Illah R. Nourbakhsh, and Davide Scaramuzza. *Introduction to Autonomous Mobile Robots*. MIT Press, 2011.

[22] Paul A. Miller, Jay A. Farrell, Yuanyuan Zhao, and Vladimir Djapic. Autonomous underwater vehicle navigation. *IEEE Journal of Oceanic Engineering*, 35, 2010.

[23] J. Vaganay, J.J. Leonard, and J. G. Bellingham. Outlier rejection for autonomous acoustic navigation. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, 1996.

[24] C. C. Tseng and Liu C. Complementary sets of sequences. *Information Theory, IEEE Transactions on*, 1972.

[25] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 1960.

[26] Greg Welch and Gary Bishop. An introduction to the kalman filter, 1995.

[27] ST Microelectronics. TDA2003 10W Car Audio Amplifier Datasheet. http://pdf.datasheetcatalog.com/datasheet/stmicroelectronics/1449.pdf. Accessed on 20/Jun/2014.

[28] mborg. Kiss FFT. http://sourceforge.net/projects/kissfft. Accessed on 15/Jun/2014.

[29] MarSensing. A Self-Recording Digital Hydrophone. http://www.marsensing.com/en/Products/digitalHyd_SR-1. Accessed on 15/May/2014.