NTNU – Trondheim
Norwegian University of
Science and Technology

Target Depth Estimation Using Hull Mounted Active Sonar

Torbjørn Ringholm

Master of Science in Physics and Mathematics
Submission date: June 2014
Supervisor: Einar Rønquist, MATH
Co-supervisor: Karl-Thomas Hjelmervik, Forsvarets Forskningsinstitutt

Norwegian University of Science and Technology
Department of Mathematical Sciences

Preface

This thesis was produced during the final 20 weeks of my master's degree programme - Applied Physics and Mathematics with specialization in Industrial Mathematics - at the Norwegian University of Science and Technology (NTNU). The work has been carried out under the supervision of both the Department of Mathematical Sciences at NTNU and the Norwegian Defense Research Establishment (FFI).

I would like to thank my supervisor Karl Thomas Hjelmervik at FFI for giving me the opportunity to work with him, teaching me underwater acoustics and providing valuable insights. Thanks are also given to professor Einar Rønquist at NTNU for his supervision and guidance through the course of this work. The code I have written utilises LYBIN, an acoustical platform developed by the Norwegian Defence Logistics Organization (NDLO) and maintained by FFI, and I would like to extend my gratitude to all the people who have worked on and are working on LYBIN.

Of course, the past five years would not have been quite as enjoyable were it not for my friends, whom I would like to thank for making the days brighter. Finally, I would like to thank my family for their support through the years and my partner Vanje for her love and encouragement.

Torbjørn Ringholm
Trondheim, June 2014

Abstract

High false alarm rates are a problem in anti-submarine warfare in littoral waters using active broadband sonar. Automatic classification algorithms may help combat this problem by filtering out detections due to non-threatening targets. An important feature for classification purposes is knowledge of the target's depth. Using active sonar with vertical beamforming capabilities, the received signal from a target can be used to find an estimate of the target's depth given an initial guess of the target's horizontal distance from the ship, the bottom profile and the sound speed profile.

The estimation is done by an optimization algorithm. The algorithm varies relevant parameters and models signals based on these parameters, comparing the modelled signals with the received signal until parameters providing an optimal fit are found. The modelling is based on using a ray tracing procedure to find eigenrays for a candidate target depth, finding vertical arrival angles and arrival times by use of these eigenrays, and synthesizing a signal based on the arrival angles and arrival times. The ray tracing procedure is done numerically using LYBIN, a platform developed by the Norwegian Defence Logistics Organization (NDLO). Three candidate objective functions for comparing recorded signals to modelled signals are presented.

The validity of the eigenray finding procedure is confirmed, and results from testing the optimization procedure on synthetic data when applying the different objective functions are presented. The results show that the method produces target depth estimates which are suitable for classification purposes.

Sammendrag (abstract in Norwegian)

Høye falsk alarm-rater er et problem ved bruk av aktiv bredbåndssonar under antiubåtkrigføring i farvann med mange fremmedelementer. Automatiske klassifikasjonsalgoritmer kan bidra til å bekjempe dette problemet ved å filtrere ut deteksjoner fra ufarlige mål. Kunnskap om målets dyp er viktig for å klassifisere mål. Ved bruk av aktiv sonar med mulighet for vertikal stråleforming kan det mottatte signalet fra et mål brukes til å finne et estimat for målets dyp, gitt en startgjetning for målets horisontale avstand fra skipet, bunnprofilen og lydhastighetsprofilen.

Estimeringen gjøres ved bruk av en optimeringsalgoritme. Algoritmen varierer relevante parametere og modellerer signaler basert på disse parameterene for så å sammenligne de modellerte signalene med det mottatte signalet. Dette gjøres inntil det blir funnet et sett med parametere som gir best mulig samsvar mellom modellert og mottatt signal. Modelleringen går ut på å anvende strålesporing for å finne egenstråler for et antatt måldyp og finne ankomstvinkler og ankomsttider basert på egenstrålene, for deretter å syntetisere et signal basert på ankomstvinklene og -tidene. Strålegangen beregnes numerisk ved bruk av LYBIN, et verktøy utviklet av Forsvarets logistikkorganisasjon (FLO). Det presenteres tre målfunksjonskandidater som brukes til å sammenligne mottatte signaler med modellerte signaler.

Validiteten til prosedyren som brukes for å finne egenstråler blir bekreftet, og resultater fra bruk av optimeringsalgoritmen på syntetisk data med de tre målfunksjonene blir presentert. Resultatene viser at metoden produserer måldypsestimater som er gode nok til bruk i klassifikasjonsøyemed.

Contents

Chapter 1

Introduction

Making observations under water is a complex task. Seawater is a highly absorbing medium when considering microwave transmission, making observation methods based on radar severely limited even at small ranges and rendering them useless for long distance underwater observations. However, as pressure waves are fairly well-behaved in water, acoustic methods using sonar (SOund Navigation And Ranging) equipment are suited for investigating underwater objects.

Two main kinds of sonar systems are in use - *passive* sonars record environmental sound only, without disturbing the surroundings. *Active* sonars work by emitting a sound pulse with a predetermined amplitude and frequency, called a *ping*, into the ocean and recording the resulting environmental sound [1]. If one considers the ocean as a system, the two approaches are equivalent to either observing the system passively, or by actively exciting the system with input in the shape of a ping and observing the recorded echoes as the response. Active sonar systems are able to obtain different information about their surroundings than passive systems, especially with their ability to measure *arrival times*, the time from a ping is emitted until an echo is heard. If the propagation speed of the ping is known, the distance traveled by it can then be estimated using its arrival time, providing an estimate of the distance between the sonar and the target from which the ping was reflected. Due to the availability of this additional information, active sonar is often used in anti-submarine warfare.

When using active sonar in anti-submarine warfare, classification of targets is a key issue - if the ping elicits a response from the surroundings in the form of an echo from a target, is the target a shoal of fish, an oil pipeline, a submarine, rock formations or something entirely different? Such questions arise especially often when using active sonar in littoral waters - oceanic regions with a high density of disturbing elements. Alarms from non-threatening objects are a problem since they complicate the tactical picture, and automatic classification schemes that can identify the source of an echo

and filter away such false alarms are needed in order to simplify the work of the operator [2]. Automatic classification schemes may also prove useful in the development of navigational systems for autonomous underwater vehicles, helping them navigate littoral waters successfully. An important feature in automatic classification algorithms is knowledge of the target's depth.

Regular sonar processing focuses mainly on estimating the range (horizontal distance from sonar), bearing and speed of a target. Knowledge of these features allows us to distinguish between stationary and moving targets in most cases. However, when encountering long objects such as oil pipelines, they may be erroneously classified as moving targets. A ship moving along a pipeline emitting pings at different points in time would place the pipeline at different points in space, thus creating the illusion of a moving target and causing a false alarm. This problem may be remedied by estimating the target's depth, as knowing the depth of the target could eliminate some options. If the target is located on the sea bottom, chances are that it is, in fact, a pipeline or some other large bottom litter object and thus not hostile, such that it may be deprioritized in favor of targets located closer to the surface. Conversely, and perhaps more importantly, if the target is *not* located at the sea bottom, it almost certainly is not an oil pipeline, shipwreck or some feature of the bottom topography, suggesting a more interesting target. Hence, knowledge of a target's depth is clearly important in automatic classication schemes. The purpose of this work is therefore to develop an algorithm capable of estimating a target's depth with sufficient accuracy to, at the very least, determine whether the target is located at the sea bottom or not, through the analysis of acoustic data collected by use of active sonar.

Active sonar systems work by having a transmitter emit a ping of known frequency and amplitude into the ocean, then recording acoustic data by use of a receiver consisting of an array of hydrophones, essentially listening for echoes of the emitted ping. The transmitter and the receiver are often assumed to be located at the same position. Once recorded, the acoustic data is processed to determine where the sound came from and at what times echoes of the ping return to the ship through the use of beamforming, matched filtering and CA-CFAR averaging [3][4].

The result of this processing is a data set containing digital acoustic data sampled over a period of time and distributed over several channels corresponding to the angles used in beamforming. The problem of depth estimation of a target now becomes part of the inverse problem of determining the properties of the ship's surroundings from the recorded acoustic data [5]. If no other information about the ocean were given, this would be quite a hard problem, as solving the inverse problem would encompass ab initio estimation of the shape and properties of the ocean floor, the target's range, the sound speed dependency on depth, et cetera, in addition

to the estimation of the target depth. However, assumptions can be made about these additional parameters from previous measurements, providing an initial guess for the properties of the surroundings.

The method proposed here for solving the inverse problem is one that utilizes modelling and optimization. If a sufficiently accurate mathematical model is available, one may simulate the propagation of sound in the ocean and use this to synthesize signals. Working with the assumption that a modelled signal resembles the recorded signal if the model parameters resemble the real world parameters, one could try to fit the model parameters by use of an optimization algorithm to create a modelled signal that resembles the recorded signal as closely as possible. Due to the nonlinear nature of sound wave propagation in water, some care must be taken in creating such a mathematical model. Here, we shall use a scheme similar to ray backpropagation, which has been used earlier for solving similar problems with passive sonar [6][7][8]. The scheme entails using *ray tracing* to obtain *eigenrays* [9]. Ray tracing is a procedure in which the path of sound rays are calculated from an approximation of the nonlinear wave equation, similar to the tracing of light rays done in optics, to determine the different paths taken by the emitted ping and its echoes. Eigenrays are the sound rays that reach a certain depth at a certain range, and are pivotal to determining the times at which and directions in which the echoes from a ping are recorded.

Calculating the path of sound rays in sea water is a non-trivial task. Due to differences in salinity and temperature at different ocean depths, the speed of sound will vary accordingly, producing refraction effects [9]. Matters are further complicated due to reflection of rays at the sea bottom and the surface. Obtaining an analytical solution for the paths taken by a ping will therefore be a near impossible problem in all but the simplest cases, introducing the need for numerical methods for obtaining these paths. This problem has been thoroughly analyzed, leading to tools such as LYBIN, a platform developed by the Norwegian Defence Logistics Organization (NDLO) which, among other things, can compute ray paths accurately and effectively [10]. We will use LYBIN extensively for this purpose.

Another point of interest is finding a way of comparing the modelled signal to the recorded signal. The way in which this is done leads to the formulation of the objective function used in the optimization procedure. The inverse problem should not be ill-posed, that is, there should exist a unique, stable solution to the problem. The choice of objective function will influence the problem's properties in this respect. In addition, the optimization procedure should not be too costly in terms of computational power, meaning the objective function should not be too time consuming to evaluate. Three objective functions will be presented which try to address these issues. The first of these is the *full comparison function*, wherein a full signal is modelled and compared entrywise with the recorded signal. The second, named the *simplified comparison function*, is an approximation

to the full comparison function. The third method is based on Bayesian inference, and therefore named the *Bayesian objective function*, wherein the recorded signal is further processed to extract discrete arrival times and arrival angles for comparison with modelled arrival times and angles, avoiding the time consuming signal modelling.

The optimization algorithm itself should be chosen so as to effectively and reliably produce satisfactory results. The objective function will have local minima which should be avoided. This leaves two options for the type of optimization algorithm - either to use a global optimization algorithm, capable of avoiding local minima by itself, or to first use an initialization procedure to provide a suitable starting point close to the global minimum, from which a local optimization algorithm is used. Here, we will employ the latter strategy, using an exhaustive search method to obtain a rough estimate of optimal model parameters, before refining the choice of parameters by use of the Nelder-Mead optimization algorithm [11].

Finally, the method's accuracy and stability in different situations and in the presence of noise must be tested. Unfortunately, obtaining real life acoustic test data is hard. This research is carried out in collaboration with the Norwegian Defense Research Establishment (FFI), which has extensive amounts of acoustic data to test the procedure on. However, due to the classified nature of this data, results obtained when using it cannot be presented here. Instead, we shall rely on the use of *synthesized* data - signals created by means of the aforementioned acoustic model - for testing the procedure. The synthesized signals are assumed to be modelled with sufficient fidelity so as to be interchangeable with real signals when it comes to testing.

Chapter 2

Theory

This chapter will provide a theoretical background for the problem, first formulating the problem in a general manner and giving an overview of the parameters involved and the proposed solution method. We will then expand on some key subjects, providing a derivation of the mathematical model for wave propagation, some considerations on factors influencing the strength of signals and a presentation of how to obtain eigenrays using the mathematical model. Continuing, we present how to model signals through the use of eigenrays, and some considerations about the preprocessing of recorded signals. Lastly, we will discuss how to represent the sound speed profile in a manner that is susceptible to ordinary optimization methods and present three choices of objective functions and the optimization algorithm used.

Some topics that are vital to the method will not be explained in much detail. Beamforming and matched filtering will not be explored in depth; as we assume that the data obtained is already processed using these techniques, a thorough explanation of them would be superfluous in the context of this thesis. Technical details regarding the sonar equipment are not discussed, and neither are environmental effects such as absorption and reflection coefficients. This is because neither of these topics are directly connected to the numerical experiments carried out later, in contrast with topics such as the analytical method for finding eigenrays, which is later used to verify the numerical eigenray finding scheme and therefore explored in greater detail.

2.1 Problem formulation and overview

Assume that a set of digital acoustic data is given, and that it is sampled at times $\{t_j\}_{j=1}^{N_t}$, beamformed vertically in the directions $\{\theta_i\}_{i=1}^{N_\theta}$ and matched filtered to yield a set of measurements $\{S_{ij}\} = \{S(\theta_i, t_j)\}$ for the acoustic intensity S at the receiver. This set of measurements is what we consider

as the *recorded signal*. Thus, we have a signal recorded in the directions θ_i and at the times t_j. Given that the signal contains echoes from a ping, our task is to solve the inverse problem of finding the target depth z_t the ping was reflected from.

However, the recorded intensity is dependent upon many other parameters in addition to the target depth, such as the geometry of the surroundings and the sound speed, and therefore the problem of target depth estimation is only part of the larger inverse problem of determining all properties of the surroundings from the recorded signal. We therefore need to determine certain parameters simultaneously, especially those which are connected to the geometry of the surroundings and the sound speed. This is similar to the problem encountered in matched field processing, which is often used for passive, low-frequency sonar [12] [13].

The parameters describing the geometry of the problem are the target range r_t, the depth of the sonar system, z_s, the bottom depth z_b and the target depth z_t. The parameters r_t, z_b and z_t are illustrated in figure 2.1. Note that while the bottom depth is in reality a function, $z_b(r)$, varying with the distance from the ship, we shall limit ourselves to considering flat bottom profiles and therefore denote the bottom depth as z_b. Another important factor, the sound speed profile $c(z)$, will generally be a function varying with depth.

Some parameters pertaining to the surroundings are not included in the inverse problem as they have less influence on the signal than the aforementioned parameters. Instead, these parameters are considered known, using standard values obtained empirically when needed. Examples of such parameters are reflection coefficients of the bottom and surface interfaces, surface wave heights, and temperature and salinity levels in the ocean (although, since they influence the sound speed, they are implicitly included through the sound speed profile) [14]. Also, the geometry of the target, such as its height and length, are disregarded.

Other parameters of importance are associated with the sonar system itself and the characteristics of the emitted ping, for example the sonar's bandwidth, B, and the sonar's vertical beamwidth, θ_{BW}. These parameters are controllable and therefore considered as known, so they are not part of the inverse problem. Their importance lies in determining statistical quantities, such as the uncertainty in measurements of arrival angles and arrival times, which are used in the modelling of signals.

In summary, there are five parameters, the estimation of which constitutes the inverse problem at hand: z_t, r_t, z_s, z_b and $c(z)$. Whereas z_t is unknown beforehand, some assumptions may be made about the four remaining parameters to provide a starting point for the solution method:

- r_t can be estimated by regular horizontal beamforming and processing, providing an initial guess. If an echo is detected at a time t, one can

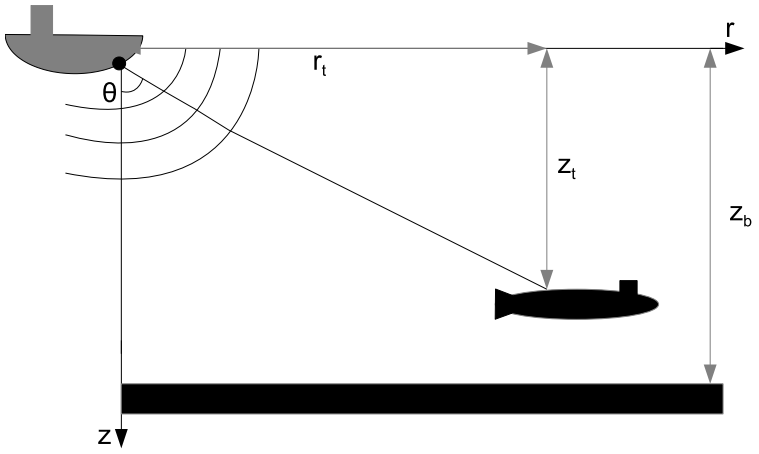

Figure 2.1: Simplified overview of geometric parameters. The line from the ship to the submarine shows the direct path taken by sound waves as a ray perpendicular to the wave fronts.

use a mean sound speed $\bar{c}$ to approximate the distance to the target by $r_t \approx \bar{c}t$, disregarding the vertical travel distance of the ping.

- z_b can be estimated if bathymetric observations for the area in which the signal is recorded are available.
- z_s will vary slightly in rough seas as the vessel follows the waves, yet one may initially assume that the sonar system is at the depth where it would be located in calm seas.
- As is the subject of section 2.7, $c(z)$ can be estimated from historical sound speed observations, and by representing it as a piecewise linear function, we can optimize with respect to $c(z)$ by use of Empirical Orthogonal Functions (EOFs) [6].

Note that these four parameters could, as a further approximation, be considered as known and thus excluded from the solution procedure. Although a solution could be obtained faster in this way, it would put the solution procedure at considerable risk of mismatching, an undesirable situation in which reliable solutions cannot be obtained due to discrepancies between the modelled situation and the real physical situation [6].

2.2 Solution approach

We shall determine z_t and the other parameters by a method based on comparisons of the recorded signal and signals modelled through a mathematical model; if the modelled signal resembles the recorded signal closely enough, the parameters used in the modelling should be close to the real parameters,

thereby constituting a good estimate of the real parameters. It is therefore important to obtain an accurate mathematical model of sound propagation and a good method of modelling signals based on this, so that the modelled signals closely resemble real signals. The sound propagation model used in this work is presented in section 2.3.

The signal modelling is based on finding *arrival times*, the time delays from the ping is emitted until the echoes are recorded at the receiver, and *arrival angles*, the directions in which the echoes are recorded. With knowledge of these, a signal can be constructed. The arrival times and angles are calculated by use of eigenrays, paths which the ping follows to reach a specified depth at a specified distance from the ship [9]. For a set of candidate parameters, including z_t and r_t, we want to find eigenrays for z_t at r_t; assuming that both the sonar system and the target are stationary, the sound must follow the eigenray paths to reach the target, and once reflected from it, must follow the same eigenray paths in reverse back toward the receiver. Hence, all possible paths for the ping are given by combinations of eigenrays. Both analytical and numerical methods for obtaining eigenrays are presented in section 2.4.

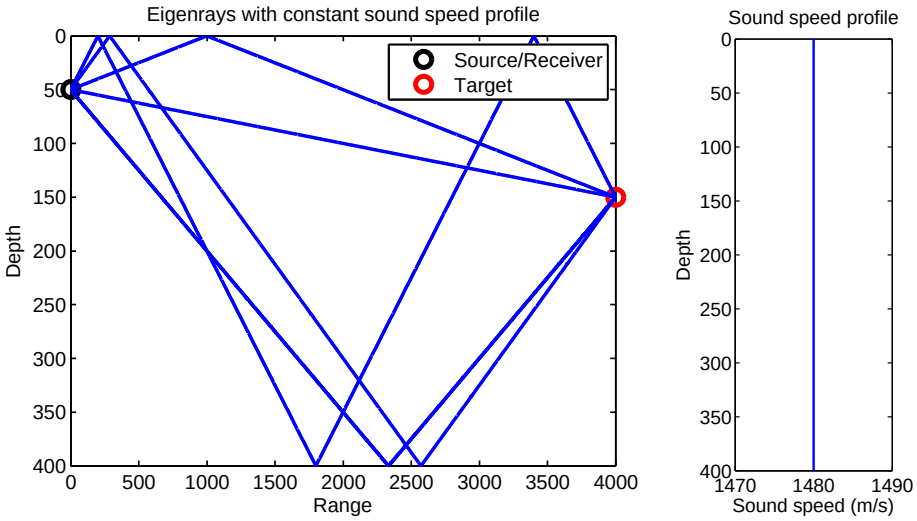

Figure 2.2: Left: Five eigenrays with constant sound speed profile. Right: Sound speed profile used. $z_s = 50$ m, $z_t = 150$ m, $z_b = 400$ m and $r_t = 4000$ m.

Figure 2.2 shows an example of five eigenrays. Consider the ping travelling along any of the five rays from the source to the target. Upon reflection, it may follow either of the five eigenrays back toward the source again. In total, this gives 25 possible combinations of eigenray paths for the ping to follow. The eigenrays' exit angles at the source then become the arrival angles for the backward propagated ping, and the arrival times become the sum of the travel times along the eigenrays chosen forward and backward.

Using the arrival angles and travel times obtained by use of eigenrays, we can synthesize signals by assuming that each arrival results in a Gaussian shaped signal centered at the arrival time and angle, and superpositioning these signals. Details about this procedure is given in section 2.5. The result of this is a synthesized signal as shown in figure 2.3, with intensities recorded as a function of time and vertical angle.

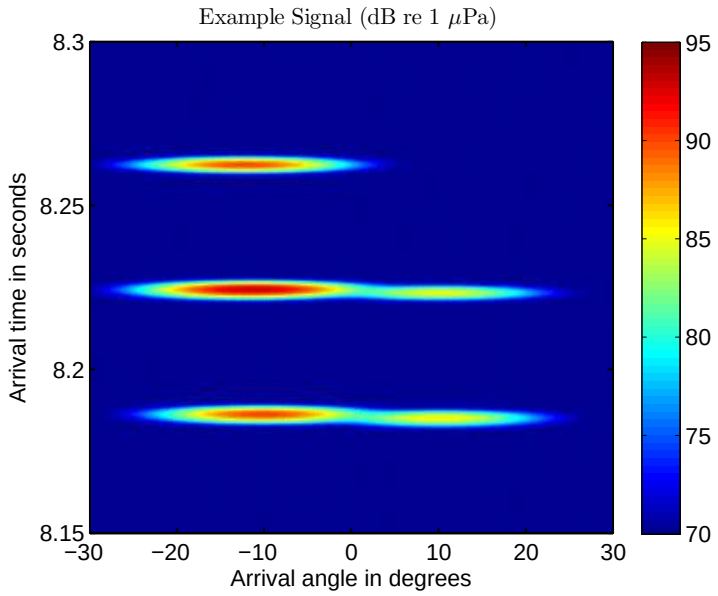

Figure 2.3: Example of a synthesized signal.

The modelled signal is then compared to the recorded signal through an objective function as explained in section 2.8, and by changing the model parameters through the use of the optimization algorithm presented in section 2.9, a set of optimal model parameters is obtained. The solution procedure is summarized in the flowchart shown in figure 2.4.

2.3 Mathematical model of sound propagation

Preliminary assumptions

Some assumptions must be made in order to obtain a mathematical model for the underwater propagation of sound. First, the speed of sound in the ocean varies in the range 1460-1500 m/s, greatly exceeding the maximum attainable relative speed between the target and the vessel, such that the target and the vessel can be considered stationary. Next, for a target located 10 km away, the ping will have to travel for 13-15 s, depending on the target's depth, which is such a small time scale that we may assume that temporal variations in ocean conditions are negligible, and thus that

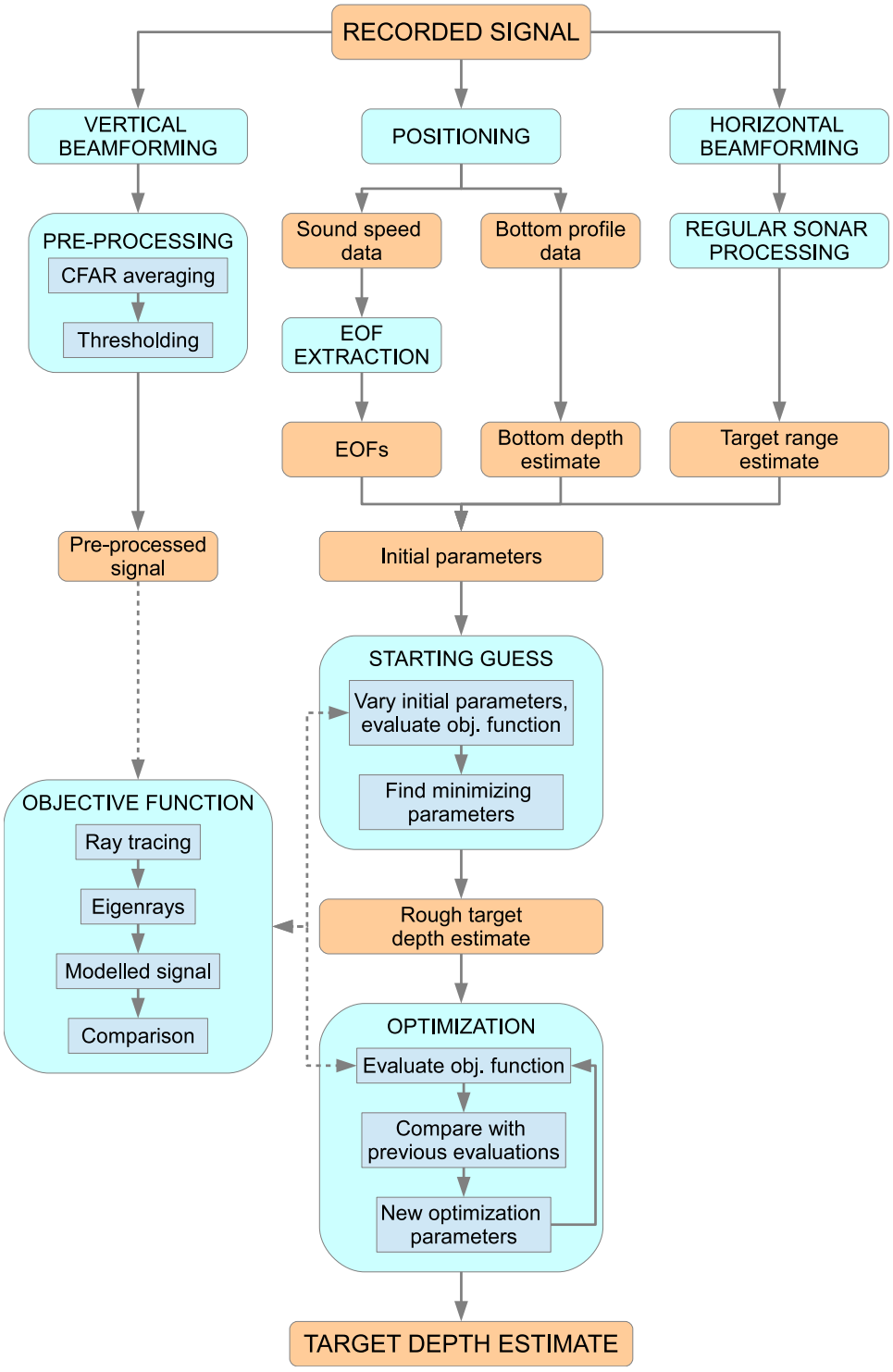

Figure 2.4: Flowchart of the complete solution process. Red boxes indicate data acquisition, green boxes indicate routines and blue boxes indicate subroutines.

sound speed is independent of time. Furthermore, we assume that both source and target can be considered as point masses, disregarding their geometrical shape. This assumption is valid for purposes of calculating the propagation of sound, but corrections must be made to account for the surface area of the target when the strength of the reflected signal is taken into consideration, as will be discussed in section 2.3.2. Next, we assume that the ping's source and the receiver are located at the same position, an assumption that is reasonable for hull mounted sonar systems since they are relatively compact. We also assume that signals are emitted omnidirectionally - that is, sound is emitted spherically from the transmitter. Finally, as the matched filter used in preprocessing the data compresses pulses in time to a length of $1/B$, where B is the signal's bandwidth, a ping will be of millisecond duration in the preprocessed signal [3]. For propagation modelling purposes, we therefore assume that the ping's duration is so short as to be considered instantaneous, although this assumption must be corrected when using the propagation model to model full signals, as explained in section 2.5. Building on these assumptions, we can derive a mathematically tractable yet physically feasible model for the propagation of the ping.

2.3.1 The ray equation

We follow [9] and start with the wave equation for pressure, which we assume the sound waves will act according to. From this, we shall describe the paths taken by the sound by rays perpendicular to the wave fronts of the pressure wave by means of characteristics of the eikonal equation. The wave equation for pressure is given by:

$$\Delta p - \frac{1}{c^2}\frac{\partial^2 p}{\partial^2 t} = 0,$$

where c is the (depth-dependent) speed of sound. Given proper boundary and initial values, this could be solved numerically by finite difference or finite elements methods, but such methods are ineffective at the scale of underwater acoustical problems; obtaining an accurate solution by such means would require a discretization with many nodes, resulting in a computationally expensive method [9]. Since our approach relies on determining the propagation of sound for many different configurations of the model parameters, we need a less computationally expensive method. To this effect, further assumptions and modifications can be made that yield simpler, more effective ways of solving the problem numerically. The first of these is using the one-dimensional Fourier transform as explained in [9] to eliminate time dependency and arrive at the Helmholtz equation:

$$\Delta p + \frac{\omega^2}{c^2} p = 0. \tag{2.1}$$

Here, ω is the angular frequency of the propagating wave, which is a known quantity since the ping is generated at a known frequency. We therefore have a problem in the spatial variables, represented in the vector $\mathbf{r}$. The next step is to seek a solution to the Helmholtz equation in the form of a ray series:

$$p(\mathbf{r}) = \mathrm{e}^{i\omega\tau(\mathbf{r})} \sum_{j=0}^{\infty} \frac{A_j(\mathbf{r})}{(i\omega)^j}, \tag{2.2}$$

yielding for the gradient:

$$\nabla p = \mathrm{e}^{i\omega\tau} \left[i\omega\nabla\tau \sum_{j=0}^{\infty} \frac{A_j}{(i\omega)^j} + \sum_{j=0}^{\infty} \frac{\nabla A_j}{(i\omega)^j} \right] \tag{2.3}$$

and for the Laplacian:

$$\Delta p = \mathrm{e}^{i\omega\tau} \left[(-\omega^2 |\nabla\tau|^2 + i\omega\Delta\tau) \sum_{j=0}^{\infty} \frac{A_j}{(i\omega)^j} + 2i\omega\nabla\tau^T \sum_{j=0}^{\infty} \frac{\nabla A_j}{(i\omega)^j} + \sum_{j=0}^{\infty} \frac{\Delta A_j}{(i\omega)^j} \right] \tag{2.4}$$

Inserting (2.2) and (2.4) into (2.1) now gives

$$(-\omega^2 |\nabla\tau|^2 + i\omega\Delta\tau) \sum_{j=0}^{\infty} \frac{A_j}{(i\omega)^j} + 2i\omega\nabla\tau^T \sum_{j=0}^{\infty} \frac{\nabla A_j}{(i\omega)^j} + \sum_{j=0}^{\infty} \frac{\Delta A_j}{(i\omega)^j} = -\frac{\omega^2}{c^2} \sum_{j=0}^{\infty} \frac{A_j}{(i\omega)^j},$$

and by equating terms of equal order in ω, we get:

$$\begin{aligned} \mathcal{O}(\omega^2) &: |\nabla\tau|^2 = \frac{1}{c^2} \\ \mathcal{O}(\omega) &: \Delta\tau A_0 + 2(\nabla\tau)^T \nabla A_0 = 0 \\ \mathcal{O}(\omega^{1-j}) &: \Delta\tau A_j + 2(\nabla\tau)^T \nabla A_j + \Delta A_{j-1} = 0. \qquad j = 1, 2... \end{aligned}$$

The first of these equations is called the *Eikonal* equation, while the rest are called the *transport* equations - the eikonal equation describes the propagation of the pressure wave, while solving the transport equations yields the amplitude of the pressure wave. Typically, as a high-frequency approximation ($\omega \gg 0$), all equations but the eikonal and the first transport are

neglected. In this application, as we are only interested in the paths taken by the wave, we disregard all transport equations and turn to the eikonal equation to obtain these paths. This equation may be solved by the method of characteristics. As can be seen from (2.3), ∇p is proportional to $\nabla\tau$ when disregarding terms of order less than ω, meaning $\nabla\tau$ is nearly perpendicular to the wave fronts. We therefore introduce the characteristic, or *ray trajectory*, $\mathbf{x}(s) = [x(s), y(s)]^T$, by

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}s} = c\nabla\tau \quad \Rightarrow \quad \left|\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}s}\right|^2 = c^2|\nabla\tau|^2 = 1, \tag{2.5}$$

where s is the arc length along the ray. Differentiating once more with respect to s and applying the chain rule now yields:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{1}{c}\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}s}\right) &= \begin{bmatrix} \dfrac{\partial^2\tau}{\partial x^2} & \dfrac{\partial^2\tau}{\partial x\partial y} \\ \dfrac{\partial^2\tau}{\partial x\partial y} & \dfrac{\partial^2\tau}{\partial y^2} \end{bmatrix} \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}s} = \begin{bmatrix} \dfrac{\partial^2\tau}{\partial x^2} & \dfrac{\partial^2\tau}{\partial x\partial y} \\ \dfrac{\partial^2\tau}{\partial x\partial y} & \dfrac{\partial^2\tau}{\partial y^2} \end{bmatrix} c\nabla\tau \\ &= \frac{c}{2}\frac{\mathrm{d}}{\mathrm{d}s}|\nabla\tau|^2 = \frac{c}{2}\frac{\mathrm{d}}{\mathrm{d}s}\frac{1}{c^2} = -\frac{1}{c^2}\nabla c\end{aligned}$$

Introducing cylindrical coordinates, $\mathbf{x}(s) = [r(s)\, z(s)]^T$, and defining

$$\xi = \frac{1}{c}\frac{\mathrm{d}r}{\mathrm{d}s}, \qquad \eta = \frac{1}{c}\frac{\mathrm{d}z}{\mathrm{d}s}$$

we get a system of ODEs:

$$\begin{aligned} \xi &= \frac{1}{c}\frac{\mathrm{d}r}{\mathrm{d}s} & \frac{\mathrm{d}\xi}{\mathrm{d}s} &= -\frac{1}{c^2}\frac{\partial c}{\partial r} \\ \eta &= \frac{1}{c}\frac{\mathrm{d}z}{\mathrm{d}s} & \frac{\mathrm{d}\eta}{\mathrm{d}s} &= -\frac{1}{c^2}\frac{\partial c}{\partial z}. \end{aligned} \tag{2.6}$$

This system is solvable, given the right initial values, but is not very descriptive. To get a solution of the more intuitive form $z(r)$ we may observe that

$$\frac{\mathrm{d}r}{\mathrm{d}s} = c\xi \Rightarrow \begin{cases} c\eta = \dfrac{\mathrm{d}z}{\mathrm{d}s} = \dfrac{\mathrm{d}z}{\mathrm{d}r}\dfrac{\mathrm{d}r}{\mathrm{d}s} = c\xi\dfrac{\mathrm{d}z}{\mathrm{d}r} & \Rightarrow \dfrac{\mathrm{d}z}{\mathrm{d}r} = \dfrac{\eta}{\xi} \\ \dfrac{\mathrm{d}\xi}{\mathrm{d}s} = \dfrac{\mathrm{d}\xi}{\mathrm{d}r}\dfrac{\mathrm{d}r}{\mathrm{d}s} = c\xi\dfrac{\mathrm{d}\xi}{\mathrm{d}r} = -\dfrac{1}{c^2}\dfrac{\partial c}{\partial r} & \Rightarrow \dfrac{\mathrm{d}\xi}{\mathrm{d}r} = -\dfrac{1}{\xi c^3}\dfrac{\partial c}{\partial r} \\ \dfrac{\mathrm{d}\eta}{\mathrm{d}s} = \dfrac{\mathrm{d}\eta}{\mathrm{d}r}\dfrac{\mathrm{d}r}{\mathrm{d}s} = c\xi\dfrac{\mathrm{d}\eta}{\mathrm{d}r} = -\dfrac{1}{c^2}\dfrac{\partial c}{\partial z} & \Rightarrow \dfrac{\mathrm{d}\eta}{\mathrm{d}r} = -\dfrac{1}{\xi c^3}\dfrac{\partial c}{\partial z}. \end{cases}$$

These three equations can be further manipulated to form one second-order ODE by differentiating the first equation with respect to r and substituting the remaining two:

$$\frac{\mathrm{d}^2 z}{\mathrm{d}r^2} = \frac{\dfrac{\mathrm{d}\eta}{\mathrm{d}r}\xi - \eta\dfrac{\mathrm{d}\xi}{\mathrm{d}r}}{\xi^2} = \frac{-\dfrac{\partial c}{\partial z} + \dfrac{\eta}{\xi}\dfrac{\partial c}{\partial r}}{\xi^2 c^3} = \frac{-\dfrac{\partial c}{\partial z} + \dfrac{\mathrm{d}z}{\mathrm{d}r}\dfrac{\partial c}{\partial r}}{\xi^2 c^3}. \tag{2.7}$$

Next, note that from (2.5) we have:

$$1 = \left(\frac{\mathrm{d}r}{\mathrm{d}s}\right)^2 + \left(\frac{\mathrm{d}z}{\mathrm{d}s}\right)^2 = \left(\frac{\mathrm{d}r}{\mathrm{d}s}\right)^2 + \left(\frac{\mathrm{d}z}{\mathrm{d}r}\frac{\mathrm{d}r}{\mathrm{d}s}\right)^2 = \left(\frac{\mathrm{d}r}{\mathrm{d}s}\right)^2 \left(1 + \left(\frac{\mathrm{d}z}{\mathrm{d}r}\right)^2\right)$$
$$\Rightarrow \left(\frac{\mathrm{d}r}{\mathrm{d}s}\right)^2 = \frac{1}{1 + \left(\frac{\mathrm{d}z}{\mathrm{d}r}\right)^2}.$$

Using this, we may rewrite the ξ^2 term in (2.7) by:

$$\xi^2 = \frac{1}{c^2}\left(\frac{\mathrm{d}r}{\mathrm{d}s}\right)^2 = \frac{1}{c^2}\frac{1}{1 + \left(\frac{\mathrm{d}z}{\mathrm{d}r}\right)^2}$$

and finally, by substituting this into (2.7) we arrive at the *ray equation*:

$$\frac{\mathrm{d}^2 z}{\mathrm{d}r^2} = \frac{1}{c}\left[1 + \left(\frac{\mathrm{d}z}{\mathrm{d}r}\right)^2\right]\left[-\frac{\partial c}{\partial z} + \frac{\mathrm{d}z}{\mathrm{d}r}\frac{\partial c}{\partial r}\right].$$

We can easily impose initial conditions for this. Any ray trajectory must start at the source depth, so $z(0) = z_s$. In addition, we require $z'(0) = \tan(\theta_0)$, where θ_0 is the exit angle of the ray [9]. Assuming the sound speed profile to be invariant within the geographical region surrounding the sonar, meaning that c is independent of r, the equation is simplified and our initial value problem becomes:

$$\frac{\mathrm{d}^2 z}{\mathrm{d}r^2} = -\frac{1}{c}\frac{\partial c}{\partial z}\left[1 + \left(\frac{\mathrm{d}z}{\mathrm{d}r}\right)^2\right], \qquad z(0) = z_s, \quad z'(0) = \tan(\theta_0). \tag{2.8}$$

Additionally, once the ray path is known, we can calculate the travel time along the ray by

$$t = \int_0^{r_t} \frac{\sqrt{1 + (z'(r))^2}}{c(z(r))}\mathrm{d}r. \tag{2.9}$$

Note that from (2.8), we can derive an important property of the sound rays. Since $c(z)$ is strictly positive, the sign of $z''(r)$ and thus the direction of curvature of $z(r)$ depends only on the sign of the $c'(z)$ term, with $\mathrm{sgn}(z''(r)) = -\mathrm{sgn}(c'(z))$. This means that if $c(z)$ increases with depth, such that $c'(z) > 0$, $z(r)$ will have negative curvature, toward shallower waters. Conversely, if $c(z)$ decreases with depth, such that $c'(z) < 0$, $z(r)$ will have positive curvature, toward deeper waters. This can be summarized in the rule that $z(r)$ will curve toward local minima in sound speed. Due to this, it is possible to find rays that follow paths that never collide with the surface or the bottom, thereby not experiencing the losses in signal strength that occur during such reflections.

2.3.2 Signal strength

A propagating ping is subject to several gain and loss mechanisms altering the ping's intensity upon arrival at the receiver. Knowledge of these mechanisms is needed both to synthesize a realistic signal and for estimating the probability of detection of a signal. In this section, some terminology and results pertaining to signal strength are presented. All quantities presented here are measured in the decibel scale. For a more detailed description of these, see [1].

The ping is emitted with a certain intensity called the *Source Level* (SL). The energy contained in the ping when emitted will be spread out spherically, meaning the intensity of the signal will be proportional to s^{-2}, where s is the distance travelled. In addition, events like bottom or surface reflections will cause intensity loss by means of absorption and scattering of the acoustic waves. These losses constitute the *Transmission Loss* (TL). We must also consider that the signal is recorded in the presence of noise, which has an intensity called the *Noise Level* (NL).

In formulating the mathematical model it was assumed that both the target and the source could be considered point masses. When considering reflection of the sound wave from the target, this assumption is no longer valid. A correction to the point mass assumption can be made by taking into account that the target reflects a part of the incoming ping proportional to its surface area, in effect acting as an amplifier for the reflected signal. This is represented by a gain called *Target Strength* (TS). Furthermore, during the beamforming done at the receiver, the signals received at several hydrophones are combined to yield one directed signal. This combination of information means there will be a gain in signal strength relative to the ambient noise. This gain is represented in the *Directivity Index* (DI).

The logarithmic sonar equation combines these five quantities to estimate the *Signal-to-Noise Ratio* (SNR) observed at the receiver:

$$\mathrm{SNR} = \mathrm{SL} - 2\mathrm{TL} + \mathrm{TS} - \mathrm{NL} + \mathrm{DI}, \tag{2.10}$$

which captures the effect of the different gain and loss mechanisms encountered by the signal when travelling from the sonar, via the target and back to the sonar [1]. Note that the transmission loss used here is defined for one-way transmissions, such that the transmission loss is doubled when considering two-way transmission of sound. If the sound follows a different path when travelling toward the target than upon returning to the sonar, different TL levels must be used for the forward and return paths, and (2.10) becomes:

$$\mathrm{SNR} = \mathrm{SL} - \mathrm{TL_F} - \mathrm{TL_R} + \mathrm{TS} - \mathrm{NL} + \mathrm{DI}, \tag{2.11}$$

where TL_F and TL_R are the transmission losses along the forward and return paths, respectively.

The SNR is used, among other things, for detection purposes. A method for discerning between noise and information in a signal is based on calculating the *Signal Excess* (SE), the strength of the recorded signal compared to some *Noise Threshold* (NT). The NT is a predetermined value, and if the SNR of some part of a signal is below this threshold, that part is considered noise. The SE is given by:

$$\mathrm{SE} = \mathrm{SNR} - \mathrm{NT} = \mathrm{SL} - 2\mathrm{TL} + \mathrm{TS} - \mathrm{NL} + \mathrm{DI} - \mathrm{NT}. \tag{2.12}$$

Thus, if $\mathrm{SE} < 0$ for some part of the signal, that part is considered noise. The SE is used to estimate the probability of detection of individual arrivals, and to remove noise from signals [1].

The choice of NT affects the probability of detection and the probability of false alarms, as low noise thresholds result in higher detection rates accompanied by higher false alarm rates. This is contrasted with the lower detection rates and lower probability of false alarms when using high noise thresholds. We will use LYBIN for the calculation of detection probabilities, which are used in the Bayesian objective function.

2.4 Finding eigenrays

The next problem at hand is using the ray equation to find eigenrays. Eigenrays can be found either analytically or numerically, depending on the complexity of the environment. For example, if the speed of sound is constant and the bottom perfectly flat, it is a simple geometric problem to find exit angles of the eigenrays given environmental parameters. However, if the bottom is irregular and the speed of sound varying with depth, as is the case in the ocean, analytical solutions are too hard to obtain, and numerical strategies must be employed. The forward problem of calculating the path of a sound ray in the ocean given initial angle, initial depth and sound speed profile, can be solved quite accurately by means of numerical ray tracing procedures [9]. However, the backward problem of calculating the initial angle from the target depth and range, i.e. finding which initial angles result in rays hitting the target, is a harder task. We here present first an analytical solution to the ray equation (2.8) for the linear sound speed profile case along with some considerations about the obtainment of eigenrays for the analytical situation, before introducing a numerical scheme for identifying eigenrays based on the forward propagation of large number of rays.

The entirety of the calculations for analytic ray tracing when considering linear sound speed profiles is presented here, as I have not been able to find a satisfactory derivation of them in the literature. For a given set of environmental parameters, eigenrays are fully characterised by their arrival angles, so the problem of finding eigenrays consists of finding θ such that for a solution of the initial value problem (2.8) with $z'(0) = \tan(\theta)$, we

have $z(r_t) = z_t$. Another quantity of interest is the travel time of sound following a given eigenray, in addition to the transmission loss occurring when following an eigenray. These parameters can be calculated for a given set of environmental parameters once the exit angle of the eigenray is known, and are used in the synthetization of signals.

2.4.1 Analytical solutions

In most cases, $c(z)$ is of such a form that finding an analytical solution to (2.8) is virtually impossible. However, analytical solutions can be found for some choices of $c(z)$, providing a useful basis for testing the accuracy and stability of the numerical eigenray schemes. One such choice is the linear sound speed profile. In this case, we have

$$c(z) = \frac{c_b - c_0}{z_b} z + c_0 \quad \Rightarrow \quad \frac{\partial c}{\partial z} = \frac{c_b - c_0}{z_b},$$

where c_b is the sound speed at the bottom. Thus, equation (2.8) becomes

$$\frac{\mathrm{d}^2 z}{\mathrm{d}r^2} = -\frac{1}{z+\gamma}\left[1 + \left(\frac{\mathrm{d}z}{\mathrm{d}r}\right)^2\right], \quad z(0) = z_s, \quad z'(0) = \tan(\theta_0), \quad \gamma = \frac{c_0 z_b}{c_b - c_0}.$$

This can be rearranged to find:

$$\begin{aligned} &\frac{\mathrm{d}^2 z}{\mathrm{d}r^2}(z+\gamma) + \left(\frac{\mathrm{d}z}{\mathrm{d}r}\right)^2 = \frac{\mathrm{d}^2}{\mathrm{d}r^2}\left[\frac{z^2}{2} + \gamma z\right] = -1 \\ &\Rightarrow \frac{z^2}{2} + \gamma z = -\frac{r^2}{2} + Ar + B. \end{aligned} \tag{2.13}$$

By imposing initial conditions, we find

$$\begin{aligned} A &= \tan(\theta_0)(z_s + \gamma), \\ B &= z_s(\gamma + \frac{z_s}{2}). \end{aligned}$$

Substituting this into (2.13), rearranging and solving a quadratic equation with respect to z now yields

$$z(r) = -\gamma \pm \sqrt{(z_s+\gamma)^2 + 2\tan(\theta_0)(z_s+\gamma)r - r^2},$$

where the choice of sign depends on which convention for the sign of z is used, and the sign of γ. We shall adopt the convention that $z \geq 0$, such that the positive solution is used if $\gamma > 0$ and the negative if $\gamma < 0$, i.e.

$$z(r) = -\gamma + \mathrm{sgn}(\gamma)\sqrt{(z_s+\gamma)^2 + 2\tan(\theta_0)(z_s+\gamma)r - r^2},$$

where sgn is the sign function.

Dealing with reflections is somewhat intricate. We first need to find the ranges $\{r_n\}_{n=1}^N$ at which reflections occur, determine whether the reflection is from the surface or the sea bottom, and then find reflection angles before determining the further propagation of the ray. It is now suitable to generalize somewhat. We look for a solution of the ray equation on several intervals $\{I_n\}_{n=0}^N$, where $I_n = [r_n, r_{n+1}]$ and $r_0 = 0$. Let $\{z_n\}_{n=0}^N$ be the collection of the $z(r_n)$, and $\{\theta_n\}_{n=0}^N$ be the angles which the ray path forms with the horizontal at the points $\{(r_n, z_n)\}_{n=0}^N$. Repeating the preceding calculations on each interval and imposing the initial conditions $z(r_n) = z_n$, $z'(r_n) = \tan(\theta_n)$ gives us the path restricted to an interval I_n as

$$z(r)|_{I_n} = -\gamma + \operatorname{sgn}(\gamma)\sqrt{(z_n+\gamma)^2 + 2\tan(\theta_n)(z_n+\gamma)(r-r_n) - (r-r_n)^2}. \tag{2.14}$$

Assuming that z_{n+1} is known, we now find an expression for which range the reflection occurs at by solving $z(r_{n+1})|_{I_n} = z_{n+1}$ with respect to r_{n+1}, which gives:

$$r_{n+1} = r_n + \tan(\theta_n)(z_n+\gamma) \pm \sqrt{\tan^2(\theta_n)(z_n+\gamma)^2 + (z_n+\gamma)^2 - (z_{n+1}+\gamma)^2}.$$

Again, some care must be taken in order to choose the right sign. There are two possible values of z_{n+1}: $z_{n+1} = z_b$ and $z_{n+1} = 0$. Also note that we require $r_{n+1} > r_n$. If $z_{n+1} = 0$ and $\gamma > 0$, we have

$$(z_n+\gamma)^2 - (z_{n+1}+\gamma)^2 \geq 0,$$

such that the positive solution must be chosen to satisfy $r_{n+1} > r_n$. Conversely, if $z_{n+1} = 0$ and $\gamma < 0$, we have

$$(z_n+\gamma)^2 - (z_{n+1}+\gamma)^2 \leq 0,$$

such that $r_{n+1} > r_n$ for either solution. We then choose the negative solution, since we want the first point of contact with the surface. In summary, if $z_{n+1} = 0$, we have:

$$r_{n+1} = r_n + \tan(\theta_n)(z_n+\gamma) + \operatorname{sgn}(\gamma)\sqrt{\tan^2(\theta_n)(z_n+\gamma)^2 + (z_n+\gamma)^2 - \gamma^2}. \tag{2.15}$$

A similar calculation for the case $z_{n+1} = z_b$ yields

$$r_{n+1} = r_n + \tan(\theta_n)(z_n+\gamma) - \operatorname{sgn}(\gamma)\sqrt{\tan^2(\theta_n)(z_n+\gamma)^2 + (z_n+\gamma)^2 - (z_b+\gamma)^2}. \tag{2.16}$$

It is now possible to determine z_{n+1}. First, we determine under which conditions a bottom reflection will occur. From (2.16), we observe that given an exit angle θ_n such that

$$\tan^2(\theta_n) < \left(\frac{z_b+\gamma}{z_n+\gamma}\right)^2 - 1,$$

there can be no bottom reflection since r_{n+1} would be a complex number. Thereby, we require

$$\tan^2(\theta_n) \geq \left(\frac{z_b+\gamma}{z_n+\gamma}\right)^2 - 1$$

for a bottom reflection to occur next. In addition, from the ray equation we know $\text{sgn}(z''(r)) = -\text{sgn}(c'(z)) = -\text{sgn}(\gamma)$, so that if $\gamma > 0$, a bottom reflection will happen before a surface reflection if possible. Also note that if $\gamma > 0$ and $\tan(\theta_n) < 0$, equation (2.16) states that $r_{n+1} < r_n$, which is inadmissible. Thus, we know that if $\gamma > 0$ and

$$\tan(\theta_n) > \sqrt{\left(\frac{z_b+\gamma}{z_n+\gamma}\right)^2 - 1}, \tag{2.17}$$

we will have $z_{n+1} = z_b$. Otherwise, $z_{n+1} = 0$. A similar calculation can be done if $\gamma < 0$; we will have $z_{n+1} = 0$ if

$$\tan(\theta_n) < -\sqrt{\left(\frac{\gamma}{z_n+\gamma}\right)^2 - 1}, \tag{2.18}$$

otherwise, $z_{n+1} = z_b$.

To obtain the exit angles θ_{n+1} at each reflection point, we use the law of specular reflection, $\theta_{n+1} = -\theta_{\text{in}}$, where θ_{in} is the incoming angle from the left, given by

$$\tan(\theta_{\text{in}}) = \lim_{r \to r_{n+1}^-} z'(r)|_{I_n}.$$

This leads to the condition

$$\tan(\theta_{n+1}) = -\lim_{r \to r_{n+1}^-} z'(r)|_{I_n},$$

from which we find

$$\tan(\theta_{n+1}) = \sqrt{(\tan^2(\theta_n)+1)\left(\frac{z_n+\gamma}{\gamma}\right)^2 - 1}, \qquad z_{n+1} = 0 \tag{2.19}$$

$$\tan(\theta_{n+1}) = -\sqrt{(\tan^2(\theta_n)+1)\left(\frac{z_n+\gamma}{z_b+\gamma}\right)^2 - 1}, \qquad z_{n+1} = z_b. \tag{2.20}$$

We have now fully determined the propagation of the rays in the linear sound speed case - given an initial angle θ_0 along with initial depth $z_0 = z_s$, we can determine the first reflection depth z_1 by use of condition (2.17) or (2.18), then find the first reflection range range r_1 by use of (2.15) or (2.16) and the first reflection angle θ_1 by use of (2.19) or (2.20). These may again

be used to find z_2, r_2 and θ_2, and so forth. The path the ray will follow between these points is given by (2.14).

Using equations 2.14-2.16 and 2.19-2.20, it is now possible to solve for the initial angle θ_0, given target depth z_t, source depth z_0, and target range r_t. The angle at which the ray hits the target is irrelevant. The direct path should pose no problems – we have one equation and one unknown, so the solution should be readily available. For each additional reflection, we introduce three unknowns; an additional z_n, θ_n and r_n. We now look for rays with certain patterns in bottom and surface reflections. Assuming knowledge of this pattern for a ray, we know all the z_n. In addition, we have one equation for each of the θ_n and r_n, leaving us with as many equations as unknowns and the possibility of solving the set of equations recursively. However, although interesting, this is considered out of the scope of this work, and we shall not attempt to solve these equations here.

The travel time integral given by (2.9) may be split this into several integrals if the number of reflections, N, of the ray is known:

$$t = \int_0^{r_t} \frac{\sqrt{1 + (z'(r))^2}}{c(z(r))} \mathrm{d}r = \sum_{n=0}^{N-1} \int_{r_n}^{r_{n+1}} \frac{\sqrt{1 + (z'(r))^2}}{c(z(r))} \mathrm{d}r + \int_{r_N}^{r_t} \frac{\sqrt{1 + (z'(r))^2}}{c(z(r))} \mathrm{d}r,$$

yet once again, attempting to obtain an analytical expression for these integrals is considered outside the scope of this work.

An example of eigenrays for the linear sound speed profile case is given in figure 2.5. These eigenrays were obtained by use of the numerical scheme presented in the next section. We can see that due to the constant positive sound speed gradient, the rays curve toward the surface as explained in section 2.3.1, allowing for rays that follow paths of surface reflections only. This is important since bottom reflections absorb more energy from the ping than surface reflections, and we can see that it is possible for a ping to follow a path of significantly less loss. In fact, this is possible for any sound speed profile in which the sound speed is increasing over a certain depth interval [1].

We may now introduce some terminology used in describing eigenrays. The transmission loss qualities of paths following eigenrays are influenced by the amount and type of reflections occurring; surface reflections cause losses in signal strength, bottom reflections even more so, and so it is important to know an eigenray's *history*. The history of an eigenray is given as a sequence of letters specifying the amount and order of reflections in the ray, for instance an eigenray with the history 'sbs' would be characterised by a surface reflection, followed by a bottom reflection and another surface reflection before reaching the target. The empty history ' ' signifies a direct eigenray. Note that although it is possible to have two subsequent surface reflections without a bottom reflection in between, there must occur a lower

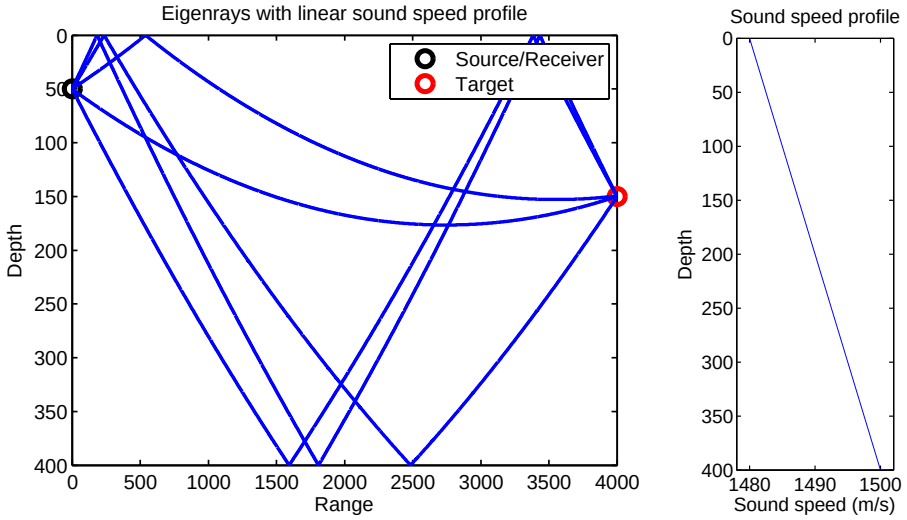

Figure 2.5: Left: Eigenrays with linear sound speed profile. Right: Sound speed profile used. $z_s = 50$ m, $z_t = 150$ m, $z_b = 400$ m, $r_t = 4000$ m.

turning point between the surface reflections. This information is important for numerical eigenray procedures, as it allows us to discern between different families of eigenrays, and is therefore included in the ray's history, signified by the letter 'l'. Also note that with negative sound speed gradients, it is possible to have two bottom reflections with an upper turning point between, which we shall signify by the letter 'u' in the history. Using this nomenclature, the eigenrays presented in figure 2.5 have the histories 'l', 'sl', 'sb', 'bs' and 'sbs'.

2.4.2 Numerical solutions

As is evident from the preceding section, numerical schemes are needed to find eigenrays due to the nonlinear nature of the ray equation making analytical solutions exceedingly hard to obtain for all but the simplest choices of sound speed profile and bottom profile.

We will employ a numerical eigenray finding scheme based on LYBIN, a platform developed by the Norwegian Defence Logistics Organization [10]. It works by calculating a large number of ray paths with varying exit angles, and selecting the rays that end up within a certain *target depth range* around the target depth. These rays are then grouped into families according to their history, and the mean of the exit angles for each family is taken as the exit angle of the eigenray belonging to that family. Similarly, travel times are calculated for each ray in a family, and the mean travel time of a family of rays is taken as the travel time of the eigenray belonging to that family. An example is shown in figure 2.6, where the rays belonging to nine

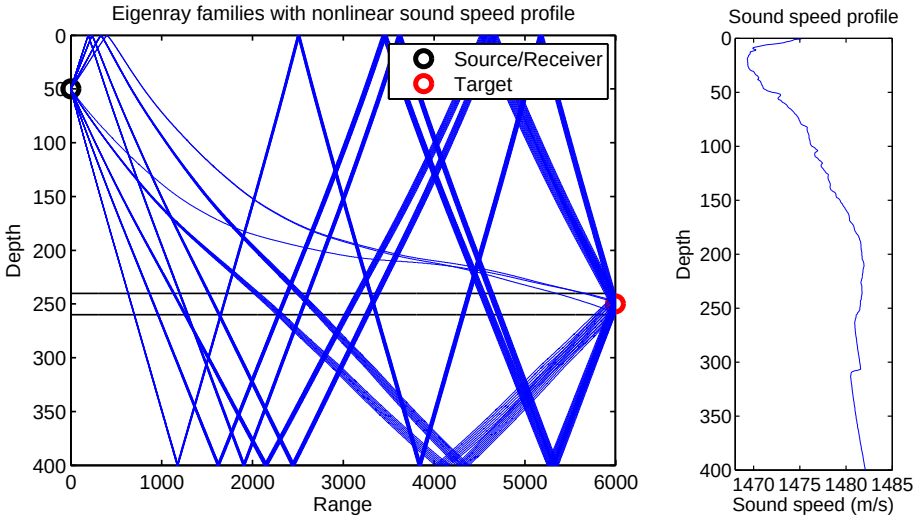

Figure 2.6: Left: Nine ray families, black horizontal lines denote edges of target depth range. Right: Sound speed profile used. $z_s = 50$ m, $z_t = 250$ m, $z_b = 400$ m, $r_t = 6000$ m. Target depth range: 10 m.

ray families are shown hitting a target depth range of extent 10 m in either direction around the target depth.

The numerical approach allows us to find approximate eigenrays for a wide range of sound speed profiles. An example is shown in figure 2.7, where eigenrays have been found when using a nonlinear sound speed profile. Note especially that the sound speed profile used gives rise to two nearly indistunguishable eigenrays with histories 'lulu' and 'lulul' following a near-sinusoidal trajectory with no bottom or surface reflections due to the tendency of refraction toward sound speed minima.

LYBIN is used here as a black box system, meaning its exact inner workings are not generally known to us. However, the accuracy of the eigenray estimates provided through the use of LYBIN can be checked. In chapter 4, this is done by finding eigenrays numerically using a linear sound speed profile, then using the numerical exit angles of the eigenrays in the linear sound speed model presented in section 2.4.1 to find the depth reached at the target range, and observing the error in eigenray depth at the target range. Note that the eigenray estimates are sensitive to the size of the target depth range and the number of rays traced. If too few rays are traced, some eigenray families may not be identified, as no rays from these families end within the desired depth range. Similarly, if the target depth range is too small, a large number of rays must be traced to ensure that all eigenray families are found, making the procedure computationally intensive. However, if the depth range is too large, there is a possibility that ray families incapable of reaching the target depth are considered as eigenray families.

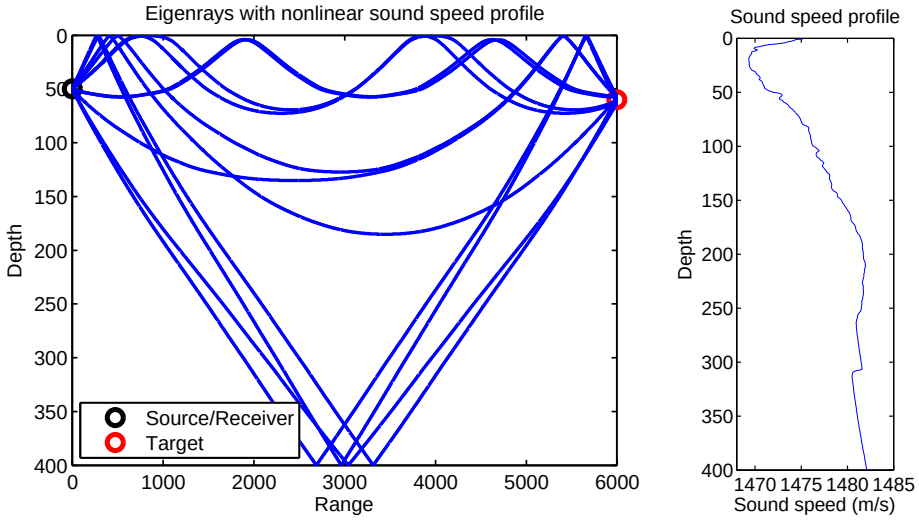

Figure 2.7: Left: Eigenrays for a nonlinear sound speed profile. Right: Sound speed profile used. $z_s = 50$ m, $z_t = 60$ m, $z_b = 400$ m, $r_t = 6000$ m.

It is therefore important to find a reasonable balance between the size of the depth range around the target depth and the number of rays traced. This issue is also addressed in chapter 4.

LYBIN is also capable of estimating the transmission loss along each ray, taking into account the distance travelled along a ray as well as losses occurring from absorption during bottom and surface reflections. It is possible to specify different sea floor types varying from from solid rock floors to sea floors consisting of deep sediment layers, each with different absorption properties. This allows us to model signals based on arrivals with different transmisson losses.

2.5 Modelling signals

Infinitely many eigenrays can be found, but most of these result in arrivals that are so weak as to be indiscernible from noise. From the large number of eigenrays that can be found numerically, only a few are needed to produce a realistic signal, namely those with least transmission loss. The optimum number of eigenrays to use in modelling signals must be decided as a trade-off between fidelity and speed; using more eigenrays in modelling will result in better accuracy of estimates, at the expense of execution time. This is the subject of one of the tests done, as presented in chapter 4.

Assume that N eigenrays are used. We now wish to construct a synthesized signal based on these, to use for comparison with the recorded signal. Recall that the signal consists of intensity measurements recorded in the

directions θ_i and at the times t_j. Each eigenray presents a path the sound will follow from the source to the target and, when assuming stationary conditions, a path the sound will follow from the target back to the source after reflection. Thus, the sound will follow all possible combinations of eigenrays forward and back again, resulting in N^2 distinct *arrivals* $(\hat{\theta}_m, \hat{t}_m)$, whose arrival times $\hat{t}_m$ are the travel times along the eigenrays followed toward the target in addition to the travel times along the eigenrays followed back toward the source. The arrival angle $\hat{\theta}_m$, the direction in which each arrival is recorded, is the exit angle of the eigenray followed back to the source.

Furthermore, a recorded signal will have certain uncertainties in the recordings; beamforming leaks information throughout the angle channels, such that an arrival will not be present in only one angle channel, but will be spread over several channels, the extent of which is dependent on the beamwidth of the receiver, θ_{BW}. Additionally, we can no longer assume that the ping is instantaneous, but must take into account that it has a certain duration. Due to this, after matched filtering, the signal will have a spread in time dependent on $1/B$, where B is the bandwidth of the sonar [3]. We will model the signal by disregarding phase information, and assuming that the m'th arrival results in a Gaussian signal given by

$$S_m(\theta_i, t_j) = A_m \exp\left(-\frac{1}{2}\left[\left(\frac{\theta_i - \hat{\theta}_m}{\sigma_\theta}\right)^2 + \left(\frac{t_j - \hat{t}_m}{\sigma_t}\right)^2\right]\right),$$

where σ_t is the ping's standard deviation in time and σ_θ its standard deviation in angle. We shall use $\sigma_t = 1/B$ and $\sigma_\theta = \theta_{BW}/2$ [2].

The amplitude, A_m, is chosen depending on the desired signal-to-noise ratio if noise is present, by letting

$$A_m = 10^{\frac{\mathrm{NL}}{10}} \cdot 10^{\frac{\mathrm{SNR}}{10}}, \tag{2.21}$$

where SNR is the logarithmic signal-to-noise ratio and NL is the noise level. The SNR can be estimated by use of equation (2.11), where we use estimates of transmission losses for the eigenrays obtained through LYBIN, and use empirical values for the signal level, directivity index, noise level and target strength. The signals resulting from the individual arrivals are superpositioned to yield the noiseless synthesized signal:

$$S = \sum_{m=1}^{N^2} S_m. \tag{2.22}$$

This generates the model signals which we try to fit to the recorded signal in the comparison based objective functions. If the modelled signal is used as a substitute for a real signal for testing the method, as is the case in chapter

4, it can be subjected to Gaussian noise to create a more realistic signal:

$$S = n(\mu_n, \sigma_n) + \sum_{m=1}^{N^2} S_m,$$

where $n(\mu_n, \sigma_n)$ is a random Gaussian process with expectation μ_n and standard deviation σ_n. Although it is possible to use a sophisticated acoustic model to calculate expected noise and reverberation levels for a given environment, we shall use the simplifying assumption that $\mu_n = 70$ dB re 1 μPa and $\sigma_n = 60$ dB re 1 μPa, respectively.

Note that when modelling signals for use in testing, the signal is given additional entries corresponding to 0.3 seconds of Gaussian noise before and after the first and last arrivals, respectively, to allow for use of the noise removal strategy described in the next section.

Example 2.5.1. Consider a situation in which the following parameters are used: $z_t = 150$ m, $z_s = 5$ m, $z_b = 500$ m, $r_t = 6000$ m and a linear sound speed profile $c(z) = 1480 + 0.04z$. The three strongest eigenrays are presented with their characteristics in table 2.1. Note the higher transmission loss in the third eigenray due to its bottom reflection.

No.	Exit angle	Travel time	TL	History
1	-5.9°	4.043 s	75.0 dB	'sl'
2	6.0°	4.042 s	76.5 dB	'l'
3	12.7°	4.101 s	82.8 dB	'bs'

Table 2.1: Eigenrays for the situation in question.

No.	$\hat{\theta}_i$	$\hat{t}_i$	TL	History
1	-5.9°	8.086 s	150.0 dB	'sl' + 'ls'
2	6.0°	8.085 s	151.5 dB	'sl' + 'l'
3	-5.9°	8.085 s	151.5 dB	'l' + 'ls'
4	6.0°	8.085 s	153.0 dB	'l' + 'l'
5	12.7°	8.144 s	157.8 dB	'sl' + 'sb'
6	12.7°	8.143 s	159.3 dB	'l' + 'sb'
7	-5.9°	8.144 s	157.8 dB	'bs' + 'ls'
8	6.0°	8.143 s	159.3 dB	'bs' + 'l'
9	12.7°	8.202 s	165.6 dB	'bs' + 'sb'

Table 2.2: Arrivals obtained from the eigenrays.

Combining these gives the nine arrivals presented in table 2.2. Using these arrivals, we synthesize a signal consisting of superpositioned Gaussian

shaped signals. The result is shown in figure 2.8. Notice how arrivals 1-4 and 5-8 superposition in clusters due to their similar arrival times, creating wide areas of intensity due to their spread in arrival angle. Also note how the cluster containing arrivals 1-4 has larger amplitude than the cluster containing arrivals 5-8 due to the difference in transmission loss between arrivals. Finally, we may note the weak, lone arrival stemming from taking the bottom reflected path both forwards and backwards.

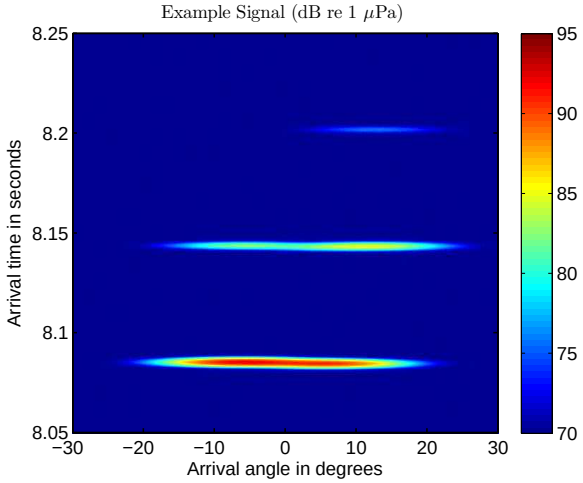

Figure 2.8: Signal based on the arrivals in table 2.2.

2.6 Preprocessing of signals

After being recorded at the receiver, the signal must be processed into a form suitable for our method. Recall that the receiver consists of a hydrophone array. In its rawest form, the signal is recorded as several time series, one for each hydrophone of the array. Determining the directionality of the signals is done by the *beamforming* technique; analogous to the human ear, the hydrophone array may be used to identify the direction from which sound originates by observing time delays and phase shifts between recordings from different hydrophones [3]. This is usually done in the horizontal plane to determine the North-South-East-West directionality of the signal, but although such horizontal beamforming is standard in most sonars, some sonars also have good vertical beamforming capabilities, meaning that one can obtain vertical directionality for the signal as well. After beamforming, the acoustic data is processed to minimize noise and thereby increase the Signal-to-Noise Ratio (SNR) of the signal. This is accomplished by first applying matched filtering, essentially looking for the known form of the emitted ping in the received signal by means of convolution [3]. The next

step is using an algorithm to discern signal from noise. Here, we shall apply CA-CFAR filtering to remove noise.

2.6.1 Noise removal by CA-CFAR filtering

After synthetization of a test signal, or after a real signal has been beamformed and matched filtered, one should try to remove ambient noise as the optimization procedure will become unstable in the presence of noise. Since noise levels may vary with time due to engine noise etc., one should first try to normalize noise levels by making a local in time estimate of noise levels. This is done by employing a normalisation scheme called cell-averaging constant false alarm rate (CA-CFAR) filtering [4]. The CA-CFAR filter normalizes the signal levels against a local estimate of noise levels; proceeding entrywise through the signal, each entry is, in turn, considered a Cell Under Test (CUT). A local estimate of the noise around the CUT is obtained as the average of the surrounding cells; the closest cells, or *guard cells*, are not included in this estimate as they may be corrupted by the signal contained in the CUT. Figure 2.9 gives an illustration; only the cells marked WINDOW are used in the noise level estimates.

The size of the guard band and the windows are chosen according to which type of signal the CA-CFAR filter is applied to. For signals with only one arrival, the guard band may be small enough to contain just one arrival. However, for signals with several arrivals with distinct arrival times one may risk corrupting the noise estimates with entries from other arrivals. Therefore, the guard band should be large enough to encompass all arrivals. In our case, we have used an ad hoc size of the guard band - the length of the signal minus 0.4 seconds worth of entries. Each window was given 40 entries. These sizes were chosen by trial and error, and are not necessarily optimal. To obtain a better size of the guard band, one may take into account that larger target ranges and bottom depths lead to a wider spread in arrival times, and adjust the guard cell number accordingly. However, the problem of finding an optimum size of guard bands is considered out of scope for this project, and the solution of this problem is left to later work.

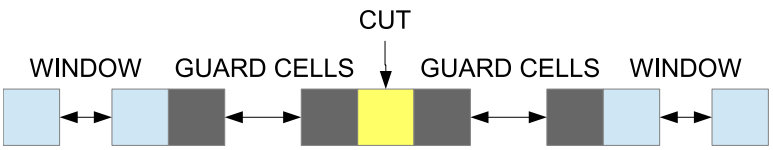

Figure 2.9: Illustration of CA-CFAR averaging

After a local noise level estimate is obtained, the value of the CUT is normalized by dividing the recorded intensity by the noise level estimate. This yields an estimate of the SNR levels in the signal, which can be used for estimating the signal excess as described in equation (2.12). The SE

estimate can in turn be used for *thresholding*, removing all signal entries with an SE estimate lower than 0. This removes most of the noise, and the remaining signal is well suited for optimization.

Also, after removing noise from the signal, it is possible to isolate arrivals by noting the times and angles (θ_i, t_i) for which local maxima appear in the signal. These arrivals are used in the objective function based on Bayesian inference presented in section 2.8.2.

2.7 Representing sound speed profiles by EOFs

Due to temperature and salinity fluctuations caused by seasonal variations and geographical differences such as ocean currents and sea depth, the sound speed profile, which dictates much of the propagation behaviour of sound, will vary between geographical regions and with the time of year. Also, ocean temperature and salinity is strongly dependent on depth, leading to depth dependent sound speed. This complicates the matter of choosing the correct sound speed profile $c(z)$ for a given situation. However, the reasonable assumption that sound speed profiles are range independent for the ranges considered here (0-10 km) simplifies the problem. Additionally, since the time scales used in this problem are small (0-20 seconds), we may assume that the sound speed profiles are time independent as well. This leaves us with the problem of choosing a sound speed profile that is representative of the time and geographical area considered, which may be solved in two ways. The first way is to make real-time observations of the sound speed profile and use these. The second is to use historical data for the sound speed profile, gathered in the same geographical area and at the same time of year as the signal is recorded.

There is some difficulty in optimizing with respect to the sound speed profile $c(z)$, the main challenge being that it is a function, implying the need for variational methods whereas standard numerical optimization methods optimize with respect to scalar quantities. We therefore want to represent the sound speed profile by means of scalars, and preferably as few as possible to simplify optimization. First, since we are reliant upon experimental data, where the sound speed is measured at certain depths, we begin by approximating $c(z)$ by a continuous, piecewise linear function:

$$c(z) = \begin{cases} c_0 \dfrac{z_1 - z}{z_1} + c_1 \dfrac{z}{z_1} & 0 \leq z < z_1 \\ c_1 \dfrac{z_2 - z}{z_2 - z_1} + c_2 \dfrac{z - z_1}{z_2 - z_1} & z_1 \leq z < z_2 \\ \vdots & \\ c_{n-1} \dfrac{z_n - z}{z_n - z_{n-1}} + c_n \dfrac{z - z_{n-1}}{z_n - z_{n-1}} & z_{n-1} \leq z \leq z_n \end{cases}$$

where $\{z_j\}_{j=1}^n$ is some partition of the ocean depth and $\{c_j\}_{j=1}^n$ the sound speed at these depths. This function is completely determined by the depth partitioning $\{z_j\}_{j=1}^n$ and the coefficients $\{c_j\}_{j=1}^n$, such that we may represent $c(z)$ by a vector $\mathbf{c} = [c_0\, c_1\, \ldots\, c_n]^T$ and a vector $\mathbf{z} = [z_0\, z_1\, \ldots\, z_n]^T$.

Changing the sound speed profile now amounts to changing the c_i coefficients. However, there may be impractically many of these. In addition, an initial guess for the c_i is needed. Both these issues can be solved by the use of historical data and EOFs. Extracting EOFs from a data set is equivalent to using principal components analysis on the data set, in which the most vital characteristics of the data are isolated by means of a Singular Value Decomposition (SVD) [15]. Assume that m measurements of the sound speeds at the specified depths have been made and are recorded in the vectors $\mathbf{c_1}, \mathbf{c_2} ... \mathbf{c_m}$. We may calculate the mean of these recordings, $\bar{\mathbf{c}}$, and use an SVD on the matrix formed by zero-mean column vectors, $[\mathbf{c_1} - \bar{\mathbf{c}}, \mathbf{c_2} - \bar{\mathbf{c}}, ..., \mathbf{c_m} - \bar{\mathbf{c}}]$ to obtain the EOFs $\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_m}$ and their associated variances $\sigma_1, \sigma_2, ..., \sigma_m$ as described in [6]. Using these, we can represent the sound speed profile as

$$\mathbf{c} = \bar{\mathbf{c}} + \sum_{k=1}^{m} \gamma_k \mathbf{v_k}$$

where the γ_j are the weighting coefficients for the EOFs. If the data is well correlated, the first few EOFs will account for most of the variation in the data, and we may therefore truncate the expansion of $\mathbf{c}$ after the first few EOFs; typically, three or four EOFs will account for >95% of the total variation in the data [2]. This is an acceptable error, and we therefore let

$$\mathbf{c} = \bar{\mathbf{c}} + \gamma_1 \mathbf{v_1} + \gamma_2 \mathbf{v_2} + \gamma_3 \mathbf{v_3}.$$

Now, by varying γ_1, γ_2 and γ_3, we also vary $c(z)$ in an efficient manner which is susceptible to ordinary optimization methods. The variances $\sigma_1, \sigma_2, ..., \sigma_m$ are useful for the optimization in that they provide reasonable limits within which the values of $\gamma_1, \gamma_2, ..., \gamma_m$ can be varied.

2.8 Objective function

In order to use optimization methods to find the best fitting model parameters, we need an objective function that compares model parameters to the recorded signal. The objective function should obtain a global minimum when optimal parameters are reached. Here, three candidates for objective function are presented. The first two are based on modelling signals from the model parameters for direct comparison with the recorded signal. The

last is based on the method of Bayesian inference, and differs from the first two in that arrival angles and times are extracted from the recorded signal and compared to modelled arrival angles and times, thus requiring further processing of the recorded signal but less modelling.

2.8.1 Direct comparison

Full comparison function

The first two objective functions make use of signal modelling as presented in section 2.5. For a given set of model parameters, eigenrays are found and used to model a signal. Since the recorded signal is sampled at N_t discrete points in time and beamformed in N_θ discrete angles, it can be represented in a matrix:

$$S = \begin{bmatrix} S_{11} & S_{12} & \dots & S_{1N_\theta} \\ S_{21} & S_{22} & & \vdots \\ \vdots & & \ddots & \\ S_{N_t1} & \dots & & S_{N_tN_\theta} \end{bmatrix}$$

where each $S_{ij} = S(\theta_i, t_j)$ is the intensity sampled in direction θ_i and at the time t_j. Similarly, the modelled signal is given by

$$M = \begin{bmatrix} M_{11} & M_{12} & \dots & M_{1N_\theta} \\ M_{21} & M_{22} & & \vdots \\ \vdots & & \ddots & \\ M_{N_t1} & \dots & & M_{N_tN_\theta} \end{bmatrix}.$$

The most obvious objective function for comparing the two signals, which we will name the *full comparison function*, is now given by

$$f(M; S) = ||S - M||^2, \tag{2.23}$$

for some matrix norm $||\cdot||$. We shall use the Frobenius norm,

$$||S - M||_F = \sqrt{\sum_{i=0}^{N_t} \sum_{j=0}^{N_\theta} |S_{ij} - M_{ij}|^2},$$

effectively finding the root-square distance between the two signals. If the recorded signal can be exactly reproduced, the function will have a unique minimum when $S = M$, a desirable property. However, when dealing with real world signals and in the presence of random noise it is almost certainly impossible to obtain a perfect reproduction of the recorded signal by modelling, and so the unique minimum will most likely not be attained. Still, the full comparison function is a viable option.

Simplified comparison function

A problem with the full comparison function is that it is computationally expensive, since each evaluation requires the formation of a full model signal M, which is a time-consuming task. An approximation can be done by considering the signal in vector form. Let

$$\mathbf{s} = \begin{bmatrix} S_{11} \\ \vdots \\ S_{N_t 1} \\ S_{12} \\ \vdots \\ S_{1N\theta} \\ \vdots \\ S_{N_t N_\theta} \end{bmatrix} \quad \text{and} \quad \mathbf{m} = \begin{bmatrix} M_{11} \\ \vdots \\ M_{N_t 1} \\ M_{12} \\ \vdots \\ M_{1N_\theta} \\ \vdots \\ M_{N_t N_\theta} \end{bmatrix}.$$

We now have

$$f(M;S) = ||\mathbf{s}-\mathbf{m}||_2^2 = \mathbf{s}^T\mathbf{s} - 2\mathbf{s}^T\mathbf{m} + \mathbf{m}^T\mathbf{m}.$$

Since the $\mathbf{s}^T\mathbf{s}$ term is independent of $\mathbf{m}$, and therefore independent of the model parameters, it is invariant under optimization with respect to the model parameters and can be disregarded. Moreover, the modelled signal M is a superposition of signals from the arrivals, as explained in (2.22), so we may write

$$\mathbf{m} = \sum_{k=1}^{N^2} \mathbf{m_k},$$

where each $\mathbf{m_k}$ corresponds to the partial signal resulting from the k'th arrival. From this, we see that by disregarding the $\mathbf{s}^T\mathbf{s}$ term, we can form the equivalent objective function

$$\bar{f}(M;S) = -2\mathbf{s}^T\mathbf{m} + \mathbf{m}^T\mathbf{m} = \mathbf{m}^T\mathbf{m} - 2\sum_{k=1}^{N^2} \mathbf{m_k}^T\mathbf{s}.$$

If the $\mathbf{m}^T\mathbf{m}$ term could now be disregarded, we would arrive at a much more computationally efficient objective function; since each of the $\mathbf{m_k}$, due to the Gaussian shape of the signal they contain, are mostly zeroes, we can compute the sum term very quickly by simply truncating the $\mathbf{m_k}$ to a smaller size containing only nonzero entries and computing the inner product of the truncated vector with the corresponding entries in the recorded signal, essentially exploiting the sparsity of the $\mathbf{m_k}$ signals.

In fact, as we shall see in chapter 4, this approach works well for a range of problems, although not in all cases. We therefore introduce the objective function, which we will name the *simplified comparison function*, given by

$$g(M;S) = -\mathbf{s}^T\mathbf{m} = -\sum_{k=1}^{N^2} \mathbf{m_k}^T\mathbf{s} \tag{2.24}$$

as an inferior, yet more efficient alternative to the full comparison function.

The problem with the simplified comparison function is that if the recorded signal has an area in which arrivals are clustered (that is, several arrivals that are so close in time and angle that they superposition on top of each other), thereby having a stronger signal in this area, the simplified comparison function will value model parameters which gather all arrivals in this cluster area higher than those where arrivals are more spread out. This can be seen from the terms

$$\mathbf{m_k}^T\mathbf{s} = \sum_{i=1}^{N_t}\sum_{j=1}^{N_\theta} S_{ij}M_{k,ij},$$

which are largest when the maxima of the $M_{k,ij}$ are close to the maximum of S_{ij}. In the full comparison function, this is corrected by the term

$$\mathbf{m}^T\mathbf{m} = \sum_{k=1}^{N^2}\sum_{l=1}^{N^2} \mathbf{m_k}^T\mathbf{m_l},$$

which punishes model parameters that assign too many arrivals close to each other. Since this term is discarded in the simplified comparison function, no such punishment exists, and a minimum occurs when all modeled arrivals are centered near the strongest point of the recorded signal, meaning the global optimum is no longer obtained when $S = M$.

2.8.2 Bayesian inference

An alternative objective function can be derived by use of Bayesian inference. Methods based on Bayesian inference have earlier been used successfully for estimating environmental properties based on acoustic data [16][17][18]. Bayesian inference is a method of model inference based on Bayes' rule where the probability of a hypothesis being correct is estimated, taking into regard experimental evidence. Bayes' rule states that [19]:

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)},$$

with H denoting the event that the hypothesis is correct and E the event that the evidence is observed. The above terms can be described further.

$P(H)$ is the prior probability, an estimate of the probability that the hypothesis is correct, calculated before reviewing the evidence, while $P(H|E)$ is the posterior probability, the probability that the hypothesis is correct when taking the evidence into account. The $P(E|H)$ terms represents the probability of observing the experimental evidence given that the hypothesis is correct, and the $P(E)$ term, called the model evidence, is the probability of observing the experimental evidence when viewed independently of the hypothesis.

We seek to maximise $P(H|E)$ by varying H, that is, to find the hypothesis most likely to be true when comparing it to the experimental evidence. Note that the $P(E)$ term is independent of the hypotheses being considered, and can therefore be disregarded in the optimization. We now wish to present our problem in the framework of Bayesian inference. The hypothesis H will be that a given set of model parameters are correct, and the experimental evidence considered will be a set of arrivals (θ_i, t_i) extracted from the recorded signal as described in section 2.6.1.

Under the assumption that the sound propagation model is accurate enough as to be considered exact, H, the event that the model parameters are correct, is equivalent to M, the event that the m modeled arrivals $\{m_j\}_{j=1}^m = \{(\hat{\theta}_j, \hat{t}_j)\}_{j=1}^m$ are correct. This assumption is not entirely true; in general, we can be quite sure that the model is qualitatively correct, but with minor numerical errors in eigenray paths and errors in sound speed causing slightly incorrect modeled arrivals even with correct model parameters. This issue is addressed later in this section.

We define the event R as the event that the set of n recorded arrivals $\{r_i\}_{i=1}^n = \{(\theta_i, t_i)\}_{i=1}^n$ is observed, and take this as the evidence event E. In the framework of Bayesian inference, we thereby wish to maximize

$$P(M|R) = P(R|M)P(M),$$

and so our objective function is $P(M|R)$. Here, the $P(M)$ term is the probability that the modeled arrivals are correct when not taking the recorded arrivals into account. Note that since M and H are equivalent events, we may take $P(M) = P(H)$. The $P(M|R)$ term is the probability that the model is correct when taking the recorded arrivals into account. The $P(R|M)$ term denotes the probability of observing the recorded arrivals when assuming that the model is correct. The next step is to explore the two right hand terms more closely, starting with the $P(R|M)$ term.

First, we need to arrive at a working definition of the event $[R|M]$, which can be described as the event:

The n recorded arrivals r_i are observed, given that the m modeled arrivals m_j are correct.

If all possible arrivals are modeled and all modeled arrivals are correct, each recorded arrival must be predicted by the model unless it is a false alarm. Stated more rigorously, this means that unless r_i is a false alarm, there must exist an m_j such that $r_i = m_j$. On the other hand, if a modeled arrival has no corresponding recorded arrival, that modeled arrival must have remained undetected. Therefore, we may refine our definition of $[R|M]$ more rigorously as the event:

All r_i are either matched or false alarms, all matched m_j are detected and all unmatched m_j are not detected.

By labelling an r_i as *matched*, we mean that there exists an m_j such that $r_i = m_j$, and vice versa for labelling an m_j as matched.

Next, using the law of total probability we condition on F, the number of false alarms among the n recorded arrivals, and find that [19]

$$P(R|M) = \sum_{k=0}^{n} P(R|M, F = k)P(F = k). \tag{2.25}$$

Let us now assume that k recorded arrivals are false alarms, and that $m \geq n$, that is, there are at least as many modeled arrivals as recorded arrivals. This implies that there must be $n - k$ matchings between recorded and modeled arrivals, and hence $m - n + k$ undetected modeled arrivals. For convenience, let $I = \{1, .., n\}$ and $J = \{1, .., m\}$. We can now define $[R|M, F = k]$ as the event:

There exist sequences $\{i_l\}_{l=1}^{n-k}, i_l \in I$ and $\{j_l\}_{l=1}^{n-k}, j_l \in J$ such that $r_{i_l} = m_{j_l}$ and m_{j_l} are detected for all l, and such that all $m_j, j \notin \{j_l\}_{l=1}^{n-k}$, are undetected.

For ease of notation, denote by $\tilde{j}$ any number in J that is not in a given sequence $\{j_l\}_{l=1}^{n-k}$, and write $\{\tilde{j}_l\}_{l=1}^{m-n+k}$ for a sequence consisting of all such numbers. The last condition in the event can then be rewritten as:

...all $m_{\tilde{j}_l}, j \in \{\tilde{j}_l\}_{l=1}^{m-n+k}$ are undetected.

All arrivals are unique unless two eigenrays have the same travel time. This occurs very rarely, and so we may assume that all arrivals are unique. Thus, there can only be one pair of sequences that yields a correct combination of recorded and modeled arrivals, excepting sequence pairs that yield the same matchings. For example, the pair $\{1, 2\}, \{1, 2\}$ and the pair $\{2, 1\}, \{2, 1\}$ are different pairs of sequences, but yield the same matchings $d_1 = m_1$ and $d_2 = m_2$. To ensure uniqueness of the different pairs made such that we avoid redundancy, we impose the condition that the sequences $\{i_l\}_{l=1}^{n-k}$ must

be ordered, that is, $i_1 < i_2 < ... < i_{n-k}$, while the $\{j_l\}_{l=1}^{n-k}$ remain unordered. The reasoning behind this is demonstrated in the following example:

Example 2.8.1. With two recorded arrivals, three modeled arrivals and no false alarms, we have $I = \{1, 2\}$, $J = \{1, 2, 3\}$, $n = 2$, $m = 3$ and $k = 0$. Allowing non-ordered subsets from both I and J, we can make twelve pairs of sequences of length $n - k = 2$:

a)	$\{1,2\},\{1,2\}$	b)	$\{1,2\},\{1,3\}$	c)	$\{1,2\},\{2,1\}$
d)	$\{1,2\},\{2,3\}$	e)	$\{1,2\},\{3,1\}$	f)	$\{1,2\},\{3,2\}$
g)	$\{2,1\},\{1,2\}$	h)	$\{2,1\},\{1,3\}$	i)	$\{2,1\},\{2,1\}$
j)	$\{2,1\},\{2,3\}$	k)	$\{2,1\},\{3,1\}$	l)	$\{2,1\},\{3,2\}.$

Note that pairs a) and i) are equivalent in that they both state that $r_1 = m_1, r_2 = m_2$. The pairs b) and k), c) and g), d) and l), e) and h), and f) and j) are also equivalent in the same respect. However, requiring that the sequences from I be ordered leaves us with the six unique pairs:

a)	$\{1,2\},\{1,2\}$	b)	$\{1,2\},\{1,3\}$	c)	$\{1,2\},\{2,1\}$
d)	$\{1,2\},\{2,3\}$	e)	$\{1,2\},\{3,1\}$	f)	$\{1,2\},\{3,2\},$

thus eliminating the problem of redundant pairs.

Next, we define I_p to be the set of all ordered sequences from I of length p, and J_p to be the set of all unordered sequences from J of length p. From this, we define

$$\bigcup_{i \in I_p} \bigcup_{j \in J_p}$$

to be the union over all sequences $i = \{i_l\}_{l=1}^{p} \in I_p$ and $j = \{j_l\}_{l=1}^{p} \in J_p$, allowing us to write $[R|M, F = k]$ as a union of events:

$$\bigcup_{i \in I_{n-k}} \bigcup_{j \in J_{n-k}} \{\text{All } r_{i_l} = m_{j_l}, \text{ all } m_{j_l} \text{ detected, all } m_{\tilde{j}_l} \text{ undetected}\}.$$

Note that the events $\{\text{All } r_{i_l} = m_{j_l}, \text{ all } m_{j_l} \text{ detected, all } m_{\tilde{j}_l} \text{ undetected}\}$ are disjoint as the pair of sequences providing the correct matching is unique, meaning no two such events can occur simultaneously unless the underlying sequences i and j are equal. This allows us to write $P(R|M, F = k)$ as

$$\begin{aligned} &P\left(\bigcup_{i \in I_{n-k}} \bigcup_{j \in J_{n-k}} \{\text{All } r_{i_l} = m_{j_l}, \text{ all } m_{j_l} \text{ detected, all } m_{\tilde{j}_l} \text{ undetected}\}\right) \\ =&\sum_{i \in I_{n-k}} \sum_{j \in J_{n-k}} P(\text{All } r_{i_l} = m_{j_l}, \text{ all } m_{j_l} \text{ detected, all } m_{\tilde{j}_l} \text{ undetected}). \quad (2.26) \end{aligned}$$

To progress further, we assume that all matchings, detections and non-detections are independent and introduce the notations $p_{ij} = P(r_i = m_j)$ and $p_{dj} = P(m_j \text{ detected})$. Using this, we find for some arbitrary $i \in I_{n-k}$, $j \in J_{n-k}$:

$$\begin{aligned} &P(\text{All } r_{i_l} = m_{j_l}, \text{ all } m_{j_l} \text{ detected, all } m_{\tilde{j}_l} \text{ undetected}) \\ =&P(\text{All } r_{i_l} = m_{j_l})P(\text{All } m_{j_l} \text{ detected})P(\text{All } m_{\tilde{j}_l} \text{ undetected}). \end{aligned} \tag{2.27}$$

Continuing, due to independence of matchings we have:

$$\begin{aligned} &P(\text{All } r_{i_l} = m_{j_l}) \\ =&P(r_{i_1} = m_{j_1}, r_{i_2} = m_{j_2}, ..., r_{i_{n-k}} = m_{j_{n-k}}) \\ =&P(r_{i_1} = m_{j_1})P(r_{i_2} = m_{j_2})...P(r_{i_{n-k}} = m_{j_{n-k}}) \\ =&\prod_{l=1}^{n-k} p_{i_l j_l}, \end{aligned} \tag{2.28}$$

and by the same argument for the remaining terms:

$$P(\text{All } m_{j_l} \text{ detected}) = \prod_{l=1}^{n-k} p_{dj_l} \tag{2.29}$$

$$P(\text{All } m_{\tilde{j}_l} \text{ undetected}) = \prod_{l=1}^{m-n+k} (1 - p_{d\tilde{j}_l}). \tag{2.30}$$

Inserting (2.28), (2.29) and (2.30) into (2.27), and inserting the result into (2.26) now yields:

$$P(R|M, F = k) = \sum_{i \in I_{n-k}} \sum_{j \in J_{n-k}} \left(\prod_{l=1}^{n-k} p_{i_l j_l} p_{dj_l} \right) \left(\prod_{l=1}^{m-n+k} (1 - p_{d\tilde{j}_l}) \right), \tag{2.31}$$

and finally, by inserting (2.31) into (2.25) we arrive at the expression:

$$P(R|M) = \sum_{k=0}^{n} P(F = k) \sum_{i \in I_{n-k}} \sum_{j \in J_{n-k}} \left(\prod_{l=1}^{n-k} p_{i_l j_l} p_{dj_l} \right) \left(\prod_{l=1}^{m-n+k} (1 - p_{d\tilde{j}_l}) \right). \tag{2.32}$$

Next, we assume that false alarms occur independently, and with a fixed probability p_{FA} for a given noise threshold [4]. Thus, the number of false alarms is binomially distributed and we have

$$P(F = k) = \binom{n}{k} p_{FA}^k (1 - p_{FA})^{n-k}. \tag{2.33}$$

We now need an expression for p_{ij} and p_{dj} that depends on the $r_i = (\theta_i, t_i)$ and the $m_j = (\hat{\theta}_j, \hat{t}_j)$. First, since arrival angles and times are measured discretely, we must define matching using the measured angles and times. We

now say that r_i matches m_j if $\theta_i \in (\lfloor\hat{\theta_j}\rfloor, \lceil\hat{\theta_j}\rceil)$ and $t_i \in (\lfloor\hat{t_j}\rfloor, \lceil\hat{t_j}\rceil)$, where $\lfloor\cdot\rfloor$ and $\lceil\cdot\rceil$ are the floor and ceiling operators rounding the quantities down and up to the closest units of precision, respectively. For arrival angles, the unit of precision is $\Delta\theta$, the angular resolution used in the beamforming, while for arrival times the unit of precision is Δt, the sampling period. Furthermore, we assume that r_i is bivariate normally distributed with expected value (θ_i, t_i), no covariance between θ_i and t_i, and standard deviations [20]:

$$\sigma_\theta = \frac{\theta_{BW}}{\sqrt{s}} \tag{2.34}$$

$$\sigma_t = \frac{1}{B\sqrt{s}}, \tag{2.35}$$

where s is the linear SNR of the signal, θ_{BW} is the vertical beamwidth of the sonar and B is the bandwidth of the sonar. As an approximation we now find:

$$\begin{aligned} p_{ij} &= P(r_i \text{ matches } m_j) = P(\lfloor\hat{\theta_j}\rfloor \leq \theta_i \leq \lceil\hat{\theta_j}\rceil \cap \lfloor\hat{t_j}\rfloor \leq t_i \leq \lceil\hat{t_j}\rceil) \\ &= \int_{\lfloor\hat{\theta_j}\rfloor}^{\lceil\hat{\theta_j}\rceil}\int_{\lfloor\hat{t_j}\rfloor}^{\lceil\hat{t_j}\rceil} C\exp\left[-\frac{(\theta-\theta_i)^2}{2\sigma_\theta^2} - \frac{(t-t_i)^2}{2\sigma_t^2}\right] \mathrm{d}t\mathrm{d}\theta \\ &\approx \Delta\theta\Delta t C\exp\left[-\frac{(\hat{\theta_j}-\theta_i)^2}{2\sigma_\theta^2} - \frac{(\hat{t_j}-t_i)^2}{2\sigma_t^2}\right], \end{aligned}$$

where C is a normalization constant dependent on $\Delta\theta$ and Δt, both of which are constants. In fact, we may take $K = \Delta\theta\Delta t C$, such that

$$p_{ij} \approx K\exp\left[-\frac{(\hat{\theta_j}-\theta_i)^2}{2\sigma_\theta^2} - \frac{(\hat{t_j}-t_i)^2}{2\sigma_t^2}\right].$$

As mentioned earlier, our model will not be entirely exact, and we need take this into account. A correction can be made through moderation of the choice of σ_θ and σ_t which, due to the inexactness of the modeled arrivals, will be larger than proposed in equations (2.34) and (2.35). Especially noteworthy is the error in modeled arrival times arising from inexact sound speed profiles. A rough approximation to the travel time along a path is $t = s/\bar{c}$, where s is the arc length of the path and $\bar{c}$ the mean sound speed. The error in travel time estimation arising from sound speed uncertainty is then

$$\sigma_t^c = \frac{s}{\bar{c}}\frac{\Delta\bar{c}}{\bar{c}},$$

where $\Delta\bar{c}/\bar{c}$ is the relative uncertainty in mean sound speed. For a target located 2000 m from the sonar, a minimum esimate of s is 4000 m, and using a mean sound speed of 1480 m/s with a relative uncertainty of 0.2%, we have

$\sigma_t^c = 5 \cdot 10^{-3}$ s. Compared to the σ_t presented above, which with a sonar bandwidth of 1 000 Hz and an SNR of 14 dB will give $\sigma_t = 2 \cdot 10^{-4}$ s, we see that the uncertainty in arrival time is grossly underestimated. Therefore, in applications, care should be taken to choose a reasonable value for σ_θ and σ_t.

The p_{dj} term for a modeled arrival is dependent on the noise threshold level used in thresholding when pre-processing the signal, in addition to the estimated signal excess level of the arrival. LYBIN provides a way of calculating p_{dj} which takes into account the distribution of environmental noise. However, as LYBIN is a black box system, the method by which it calculates p_{dj} is unknown to us, and we shall have to rely on the values provided. This concludes our derivation of $P(R|M)$. The next step is to obtain an expression for the prior probability $P(M)$.

The prior probability $P(M)$ can be estimated through an analysis of all model parameters and their uncertainties. Recall that the event M is equivalent to the event H that the model parameters are correct. The model parameters considered here are r_t, z_t, z_b, z_s, and $c(z)$, represented by the EOF coefficients γ_1, γ_2 and γ_3. Regarding them as random variables and assuming independence of their correctness, we find

$$\begin{aligned} &P(M) = P(H) \\ =&P(r_t = \hat{r_t}, z_t = \hat{z_t}, z_b = \hat{z_b}, z_s = \hat{z_s}, \gamma_1 = \hat{\gamma_1}, \gamma_2 = \hat{\gamma_2}, \gamma_3 = \hat{\gamma_3}) \\ =&P(r_t = \hat{r_t})P(z_t = \hat{z_t})P(z_b = \hat{z_b})P(z_s = \hat{z_s})P(\gamma_1 = \hat{\gamma_1})P(\gamma_2 = \hat{\gamma_2})P(\gamma_3 = \hat{\gamma_3}), \end{aligned}$$

where $\hat{p}$ denotes the estimated quantity of the parameter p. The problem now becomes assigning probability distributions to each random variable. If all variables are considered uniformly distributed, it is easily seen that $P(M)$ becomes constant and thereby irrelevant to the optimization. This is in fact what has been done in this implementation.

On the other hand, a more sophisticated model may view the random variables as having distinct distributions. We are given initial estimates r_{t0} and z_{b0} of r_t and z_b, and uncertainties in these estimates can be used as standard deviations, allowing us to view r_t and z_b as normally distributed. The sonar depth z_s will vary with surface wave height, and assuming a model for the surface waves is obtainable, for instance assuming they will follow a sinusoidal shape with a given frequency and amplitude, a probability distribution for z_s can be obtained. The EOF coefficients can also be assumed to be normally distributed with zero mean and variances $\sigma_1, ..., \sigma_m$. The target depth z_t is tricky. If there exists an estimate for the target depth from a previously recorded ping, the target depth may follow a Gaussian distribution with this previous estimate as expected value and a certain standard deviation which should be based on worst-case-errors using σ_θ. The use of non-uniform distributions in calculating $P(M)$ is an interesting topic for further work.

Approximate expression

Evaluation of the objective function for Bayesian inference is rather computationally expensive; the number of unordered sequences of length $n-k$ from $\{1, ..., m\}$ is $m!/(m-n+k)!$, where $p!$ denotes the factorial of p. The number of ordered sequences of length $n-k$ comprised of numbers from $\{1, ..., n\}$ is $n!/(k!\,(n-k)!)$, such that the number of pairs of sequences made with k false alarms is:

$$\frac{m!n!}{(m-n+k)!\,k!\,(n-k)!},$$

a number which grows prohibitively large as n and m grows. This means that the number of terms in the double sum involved in evaluating equation (2.32) will be very large, and we will need to find an approximation to equation (2.32) for the objective function to work in practice.

First, we may note that for a noise threshold level of 13 dB, p_{FA} is of the order of magnitude 10^{-6}, and so by equation (2.33), $p(F=0) \approx 1$ and $p(F=k) \approx 0, k = 1, ..., n$ if n is not too large [4]. The choice of NT used in the tests in chapter 4 is 13 dB, and so this is a reasonable simplification. This gives us the approximate expression

$$P(R|M) \approx \sum_{i \in I_n} \sum_{j \in J_n} \left(\prod_{l=1}^{n} p_{i_l j_l} p_{d j_l} \right) \left(\prod_{l=1}^{m-n} (1 - p_{d \tilde{j}_l}) \right). \tag{2.36}$$

This reduces the amount of additions to $m!/(m-n)!$. In fact, when using a uniform distribution on the model parameters as described above, we can use this $P(R|M)$ as the objective function, which is what is done in the implementation used for testing the Bayesian objective function in chapter 4. Note that using the same amount of modeled arrivals as recorded arrivals, letting $m = n$, gives the even more simplified expression

$$P(R|M) \approx \sum_{i \in I_n} \sum_{j \in J_n} \left(\prod_{l=1}^{n} p_{i_l j_l} p_{d j_l} \right).$$

2.9 Optimization

The final part of the solution procedure, optimization, is done by comparing the recorded signal to a modelled signal, then attempting to modify the optimization parameters in the modelled signal in order to obtain a better fit. The choice of which parameters to optimize with respect to is a matter of complexity and accuracy. By optimizing with respect to too few or inconsequential parameters, we risk obtaining a sub-optimal fit, and as a result, an incorrect target depth estimate. On the other hand, if too many parameters are included, the computational complexity of the problem may

become insurmountable. The parameters to optimize with respect to should therefore be chosen carefully.

Of course, z_t should be among the optimization parameters. Other suitable candidates for optimization parameters are r_t, z_b, z_s and c, as these parameters influence the eigenray paths used in modelling signals. As explained in section 2.7, optimization with respect to c entails optimizing with respect to the EOF weighting coefficients $\gamma_1, \gamma_2, ... \ \gamma_m$.

2.9.1 Optimization algorithm

The black box nature of LYBIN makes partial derivatives of any objective function with respect to the problem parameters impossible to obtain, leaving us with the choice of either a derivative-free optimization algorithm or using numerical gradients in a more sophisticated algorithm. As the objective functions are generally computationally expensive to evaluate, we would like to limit the amount of evaluations needed. Calculating numerical gradients calls for several evaluations per approximation, thereby favouring derivative-free algorithms. Due to its robustness and ease of implementation, the algorithm chosen here is the popular derivative-free Nelder-Mead algorithm for unconstrained nonlinear optimization problems [11] .

Algorithm 2.9.1. Nelder-Mead

Initialization: Starting with the initial guess $\mathbf{x}_0 \in \mathbb{R}^n$, generate n additional points $\mathbf{x}_i$ by increasing the i'th component of $\mathbf{x}_0$ by 5 %. If the i'th component of $\mathbf{x}_0$ is 0, instead change that component to 0.00025 in $\mathbf{x}_i$.

1) Sort the $\mathbf{x}_i$ according to their objective function values $f(\mathbf{x}_i)$, and denote by $\mathbf{x}^{(1)}, ..., \mathbf{x}^{(n+1)}$ the sequence of points sorted from lowest to highest objective function value. Compute the centroid of the n first points, $\mathbf{x}_m = \frac{1}{n}\sum_{k=1}^{n} \mathbf{x}^{(k)}$.

2) Find $\mathbf{x}_r = \mathbf{x}_m + (\mathbf{x}_m - \mathbf{x}^{(n+1)})$, the reflection point of $\mathbf{x}^{(n+1)}$ through the centroid.

3) Calculate $f(\mathbf{x}_r)$. If $f(\mathbf{x}^{(1)}) \leq f(\mathbf{x}_r) < f(\mathbf{x}^{(n)})$, exchange $\mathbf{x}^{(n+1)}$ with $\mathbf{x}_r$ and return to step 1. If $f(\mathbf{x}_r) < f(\mathbf{x}^{(1)})$, go to step 4. If $f(\mathbf{x}_r) \geq f(\mathbf{x}^{(n)})$, go to step 5.

4) Find $\mathbf{x}_e = \mathbf{x}_r + (\mathbf{x}_m - \mathbf{x}^{(n+1)})$, expanding on the reflection point. Calculate $f(\mathbf{x}_e)$. If $f(\mathbf{x}_e) < f(\mathbf{x}_r)$, exchange $\mathbf{x}^{(n+1)}$ with $\mathbf{x}_e$ and return to step 1. Otherwise exchange $\mathbf{x}^{(n+1)}$ with $\mathbf{x}_r$ and return to step 1.

5) If $f(\mathbf{x}_r) < f(\mathbf{x}^{(n+1)})$, let $\mathbf{x}_c = \mathbf{x}_m + (\mathbf{x}_r - \mathbf{x}_m)/2$. Otherwise, let

$\mathbf{x}_c = \mathbf{x}_m + (\mathbf{x}^{(n+1)} - \mathbf{x}_m)/2$. If $f(\mathbf{x}_c) < f(\mathbf{x}_r)$, exchange $\mathbf{x}^{(n+1)}$ with $\mathbf{x}_c$ and return to step 1. Otherwise, go to step 6.

6) Shrink the simplex by assigning $\mathbf{v}_i = \mathbf{x}_1 + (\mathbf{x}_i - \mathbf{x}_1)/2,\ i = 2, ..., n{+}1$ and exchanging all $\mathbf{x}_i$ for $\mathbf{v}_i$ except $\mathbf{x}_1$. Return to step 1.

Termination: The algorithm is terminated upon some reaching some stopping criterion, such as the volume of the simplex shrinking below some tolerance level, or the length of each edge of the simplex being smaller than the tolerance level.

The idea behind the algorithm is quite simple. Given n optimization variables, we create a simplex of dimension n defined by $n+1$ corner points distributed around the starting guess. An n-dimensional simplex is the n-dimensional equivalent of a triangle; in two dimensions, it is a triangle, in three dimensions a tetrahedron, et cetera. The shape and position of this simplex is then altered by exchanging the corner point with the highest function value for some new point located outside or inside the simplex with a lower function value. If no point with a lower function value is found with which to exchange the worst corner point, the simplex is shrunk. Thus, the simplex will move toward areas of lower function values, eventually shrinking as it approaches a local minimum, enclosing the minimum. The algorithm is terminated upon reaching some stopping criterion - for example if the volume of the simplex becomes smaller than a given tolerance.

To avoid local minima, the Nelder-Mead algorithm requires that the optimization start reasonably close to the global minimum. To find such an initial guess, an exhaustive search method is employed, computing the objective function values with different problem parameters and choosing the parameters that yield the lowest objective function value. This approach quickly runs into the curse of dimensionality, as an increasing number of optimization parameters necessitates a large number of evaluations to obtain a reasonable initial guess. For example, with two parameters one may wish to check the objective function values for five choices of each parameter, necessitating 25 evaluations of the objective function. If a third parameter were introduced, and one wants to check the objective function values for five choices of this parameter as well, 125 evaluations of the cost function are needed, resulting in a large increase in function evaluations. Due to this, a global optimization algorithm able to obtain the global minimum without relying on the expensive initialization procedure would be preferable. One such algorithm is the simulated annealing algorithm, a derivative-free, stochastic global optimization method, which has been applied with success to geoacoustical parameter estimation using matched field processing [21]. The application of simulated annealing to this optimization problem may be an interesting task in the future. On a side note, we may mention that

the exhaustive search is very parallelizable, and a parallel implementation should be considered in the future if the simulated annealing algorithm does not provide significant improvements.

Example 2.9.1. With a geometry of $z_s = 50$ m, $z_t = 350$ m, $z_b = 400$ m and $r_t = 8000$ m, and a given sound speed profile, the signal displayed in figure 2.10 is synthesized and used as a recorded signal.

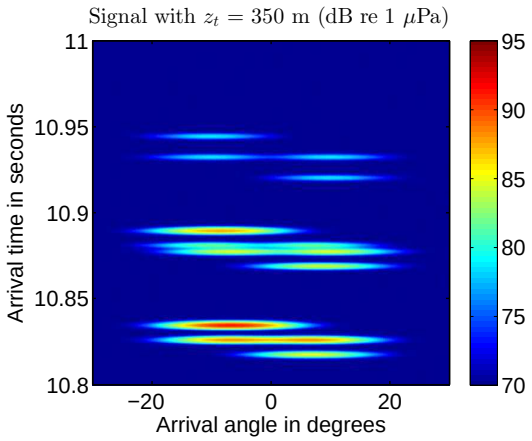

Figure 2.10: Signal when $z_s = 50$ m, $z_t = 350$ m, $z_b = 400$ m, $r_t = 8000$ m.

The method is applied, optimizing with respect to bottom depth and target depth using the simplified objective function, while considering $z_s = 50$ m and $r_t = 8000$ m as known. The initialization procedure considers bottom depths from 350 to 450 m in steps of 10 m, and target depths from surface to bottom in steps of 10 m, obtaining the objective function values displayed in figure 2.11.

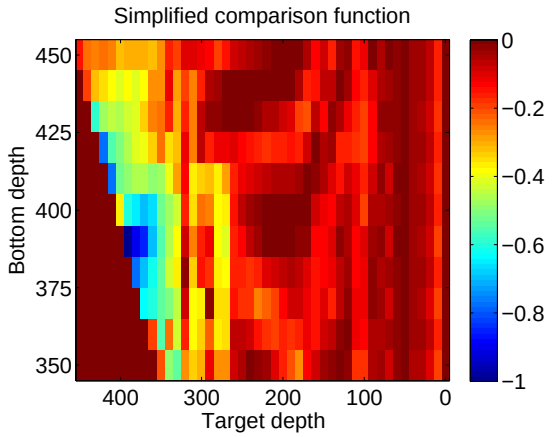

Figure 2.11: Simplified comparison function for various model bottom depths and target depths. Correct model parameters: $z_t = 350$ m, $z_b = 400$ m.

The minimum of these is found at $z_b = 390$ m, $z_t = 390$ m, which is used as an initial guess in the Nelder-Mead algorithm. After applying the Nelder-Mead algorithm, the optimal parameters are found to be $z_b = 386$ m, $z_t = 372$ m. The signal produced with these parameters is shown in figure 2.12, resembling that of figure 2.10.

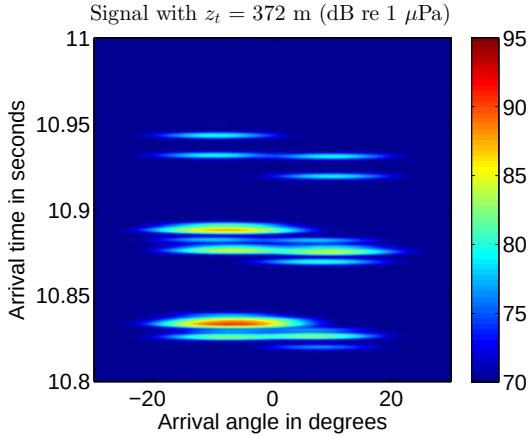

Figure 2.12: Signal when $z_s = 50$ m, $z_t = 372$ m, $z_b = 386$ m, $r_t = 8000$ m.

Chapter 3

Implementation and test setup

3.1 Implementation

The solution method was implemented in MATLAB. Due to the modular structure of the program, as can be observed from the flowchart in figure 2.4, implementation of the method is done by use of many separate functions. Use of LYBIN is facilitated through the binary interface LybinCom 6.2 and incorporated in the MATLAB code through the use of a COM server [22]. Since LYBIN is currently a 32-bit-only program, a 32-bit version of MATLAB had to be used in order for the LybinCom interface to function. The Nelder-Mead optimization algorithm described in section 2.9 is already available as the built-in MATLAB function `fminsearch.m`, and is the only non-basic built-in MATLAB function used in the code.

In a previous implementation, LYBIN version 6.1 was used. This version had an issue with the inability to modify the target depth range used for identifying eigenrays as explained in section 2.4.2. The range was set to 80 m with no possibility of change, causing errors in eigenray estimation which led to inaccurate target depth estimates [23]. With the release of the unofficial update version 6.2 in February 2014, this is no longer an issue as the target depth range is modifiable. This is demonstrated in section 4.1.

When implementing the Bayesian objective function, the approximate expression given in equation (2.36) was used. However, there was an unfortunate oversight; the p_{ij} were calculated using the incorrect formula

$$p_{ij} \approx K \exp\left[-\frac{(\hat{\theta}_j - \theta_i)^2}{2\sigma_\theta^2} - \frac{(\hat{t}_j - t_i)^2}{2\sigma_t}\right]$$

and not the the correct version:

$$p_{ij} \approx K \exp\left[-\frac{(\hat{\theta}_j - \theta_i)^2}{2\sigma_\theta^2} - \frac{(\hat{t}_j - t_i)^2}{2\sigma_t^2}\right],$$

meaning that the standard deviation in time was effectively $\sqrt{\sigma_t}$ instead of the correct σ_t. Luckily, this error turns out to be less significant than expected; the method was tested using

$$\sigma_t = \frac{1}{B\sqrt{s}},$$

instead of the corrected version

$$\sigma_t^c = \frac{s}{\bar{c}}\frac{\Delta\bar{c}}{\bar{c}}$$

proposed in chapter 2.8.2. During testing, the SNR ranged from 10 dB to 30 dB and B was set to 1000 Hz, and so the effective standard deviation in time used in the implementation varied from $\sqrt{\sigma_t} = 0.017$ s for an SNR of 10 dB to $\sqrt{\sigma_t} = 0.0056$ s for an SNR of 30 dB. Had the correct expression for p_{ij} and the corrected version of σ_t been used, the standard deviation in time would have varied from $\sigma_t^c = 0.027$ s for a target range of 10 000 m to $\sigma_t^c = 0.005$ s for a target range of 2 000 m, when using a relative sound speed error of 0.2% and a mean sound speed of 1480 m/s. Therefore, we can consider the results obtained here by use of the Bayesian objective function as comparable to results obtained with the corrected σ_t^c. Since the incorrect implementation was used during all tests where the Bayesian objective function was applied, the long time needed for carrying out the tests (2-3 weeks of runtime using a computer cluster) made redoing the tests impractical. It was therefore decided to keep the results obtained with the implementation error, and prioritize a new run of these tests during future work.

LYBIN allows the user to assign parameters pertaining to the sonar system and the environment, all of which have default values unless otherwise specified [22]. Some of these parameters were varied during the testing, and the choice of these parameters is described in the next section. The parameters which were set to a non-default level but otherwise kept constant throughout the testing is described below:

- Wind speed was set to 0 m/s, meaning no waves were assumed.
- A silt bottom was assumed, which corresponds to bottom type 4 in LYBIN [22].
- The bandwidth of the sonar, B, was set to 1 kHz.
- The vertical beamwidth of the signal, θ_{BW}, was set to 10 degrees.
- The beamforming resolution, $\Delta\theta$, was 0.1 degrees. If this had been a real recorded signal, this would imply extensive oversampling as the vertical beamwidth is 10 degrees, but since we are working with synthesized data, no actual beamforming is done, and so this only determines the angular resolution of the synthesized signals.

- The frequency of the emitted ping, f, was set to 6.5 kHz.
- The sampling frequency, f_s, was set to 2 kHz. Note that if this were a real signal, the signal would be undersampled. The sampling frequency would then need to be higher to satisfy the Nyquist sampling criterion unless we were considering baseband converted, complex sampled data, but once again, since we are working with synthesized data, this only determines the time resolution of the signal.
- The signal level used was 218 dB re 1 μPa, the target strength 10 dB, the noise level 70 dB re 1 μPa and the directivity 20 dB.
- The noise threshold value used was 13 dB. Note that this value may need to be adjusted for use with real data, as it is quite restrictive.

3.2 Test setup

When testing the method, we are interested in whether the eigenray estimates produced are correct, how accurate the target depth estimates are, which objective functions perform best under different circumstances, and how sensitive the method is to noise and disturbances. We therefore wish to test the method on a wide range of situations, meaning there is a time constraint on how long the optimization procedure is allowed to run. As explained in section 2.9, the initialization procedure is subject to the curse of dimensionality, meaning that the execution time grows exponentially in the number of optimization variables. Therefore, we shall instead opt for the use of several tests, while limiting ourselves to a maximum of three optimization variables in each test. All tests of the target depth estimate accuracy are done on synthesized data.

In all, five tests have been carried out. The first test is designed to check the accuracy of the numerical eigenray estimates. The second, third and fourth tests investigate the accuracy of the target depth estimation method when using the different objective functions, in the presence of noise and under differing environmental conditions. In the second test, we optimize with respect to bottom depth and target depth in different problem geometries using a fixed sound speed profile. In the third test, we optimize with respect to the two most significant EOF coefficients and the target depth in different problem geometries. In the fourth test, we again optimize with respect to bottom depth and target depth, but this time with several qualitatively different sound speed profiles. The fifth and last test considers the execution time of the method when using different objective functions for a smaller set of problem geometries.

Target range and source depth are not considered as optimization variables in any test; it is assumed that optimizing with respect to bottom depth

will be sufficient to investigate the performance of the method when using geometric parameters as optimization variables, and so it would be unnecessarily time consuming to run tests with other geometric parameters as optimization variables.

3.2.1 Accuracy of eigenray estimation

A test was done to check the accuracy of the eigenray estimates, in which cases the eigenray estimates might fail, and to demonstrate the improvements in eigenray accuracy from LYBIN version 6.1 to version 6.2. For each set of environmental parameters (r_t, z_b and z_t), five eigenrays were calculated by the numerical method presented in section 2.4.2. The range of r_t considered was 1000 m to 10 000 m in steps of 100 m, with z_b from 100 m to 1000 m in steps of 10 m, and z_t from 50 m to 650 m in steps of 200 m. The source depth z_s was kept constant at 50 m throughout the test. A linear sound speed profile was used, in which sound speed varied from 1480 m/s at the surface to 1500 m/s at the bottom. During this test, 5000 rays were traced per eigenray estimate.

The numerical eigenray method was set to produce five eigenrays, represented by their exit angles $\{\theta_i\}_{i=1}^5$, for each set of parameters. These exit angles were used as initial angles in an analytical ray tracing, as described in section 2.4.1. The resulting analytical depth at target range given the numerical exit angles, $z(r_t; \theta_i)$, was compared with the desired target depth z_t, giving the mean error in eigenray depth at target range, which is used as the benchmark for checking eigenray accuracy:

$$E = \frac{1}{5} \sum_{i=1}^{5} |z_t - z(r_t; \theta_i)|. \tag{3.1}$$

Note that this is a test of eigenray accuracy with linear sound speed profiles only, since we do not have an analytical solution for a general sound speed profile. The idea is that accuracy with a linear sound speed profile will imply accuracy when using a nonlinear sound speed profile.

Sensitivity to number of rays used and target depth range size

The eigenray estimates are sensitive to the size of the target depth range used for identifying ray families for eigenrays, as explained in section 2.4.2. With LYBIN version 6.2, the size of the target depth range can be varied, allowing for more accurate eigenray estimates. Another important detail in the numerical eigenray estimation is the number of rays traced; with too few rays traced, some ray families may be overlooked, causing the procedure to fail in identifying certain eigenrays. On the other hand, more rays in the eigenray finding procedure will lead to longer execution time. We therefore

want to find a trade-off between the size of the target depth range and the number of rays traced in the eigenray finding procedure.

This issue has been investigated by varying the target depth range from 15 m to 55 m in steps of 10 m and the number of rays used from 0 to 3000 in steps of 500, measuring the mean errors in eigenray estimates using equation (3.1). The bottom depth was set to 200 m, the target depth 50 m, the source depth 50 m and the target range to 6000 m during this test, representing a typical scenario. Once again, as we are reliant on analytical solutions to test the eigenray estimates against, a linear sound speed profile was used.

3.2.2 Tests on synthesized data

We have not been able to obtain real acoustic data to test the method on. FFI has several sets of recorded data which are suited for testing, yet all of these are classified such that results found based on this data cannot published. However, it is possible to test the procedure using synthesized data. When using the full comparison and simplified comparison functions, this entails using a signal modelled as proposed in section 2.5 with added Gaussian noise as the recorded signal. When using the Bayesian objective function it is unnecessary to model signals, and only arrival angles and arrival times are synthesized.

To obtain a more realistic situation and to test the method's sensitivity to inexact angle and arrival time measurements, the modeled arrival angles and arrival times were treated as Gaussian distributed random variables, as proposed in [2]:

$$\theta_i = n\left(\hat{\theta}_i, \frac{\theta_{BW}}{\sqrt{s}}\right), \quad t_i = n\left(\hat{t}_i, \frac{1}{B\sqrt{s}}\right).$$

Here, $n(\mu, \sigma)$ specifies a Gaussian process with expected value μ and standard deviation σ. The $\hat{\theta}_i$ and $\hat{t}_i$ are the arrival angles and arrival times as found by the numerical eigenray scheme, θ_{BW} is the vertical beamwidth of the sonar, B is the bandwidth of the sonar and $s = 10^{\mathrm{SNR}/10}$ is the linear signal-to-noise ratio. The θ_i and t_i were obtained by first calculating $\hat{\theta}_i$ and $\hat{t}_i$ numerically and then adding Gaussian noise to these values.

In addition to the Gaussian noise added to signals and the arrivals, we want to be able to test the method's accuracy with differing SNR levels. This has been done by modifying the signal strengths of the different arrivals. The SNR level for each arrival is first calculated as proposed in section 2.3.2, and we denote by SNR_m the calculated SNR value of the m'th arrival. Then, the SNR levels are adjusted by assigning

$$\mathrm{SNR}_m \leftarrow \mathrm{SNR}_m - \max_k\{\mathrm{SNR}_k\} + \mathrm{SNR}_{\mathrm{model}}.$$

This alters the strongest arrival's SNR level to $\mathrm{SNR}_{\mathrm{model}}$, and the other arrivals' SNR values are modified relative to this level. The new SNR levels

are then used for modelling signals as described in section 2.5, yielding a modeled signal with an adjusted SNR level for use with the comparison objective functions. Note that by varying the SNR values, s is also varied, allowing us to test the method's stability in the presence of different levels of noise in the signal and inaccuracies in measurements.

In all tests, unless otherwise specified, five eigenrays were used for synthetization of recorded signals and modelling of signals for use in the two comparison functions. When testing the Bayesian objective function, two eigenrays were used for synthetization of recorded arrivals and two eigenrays were used for modelling arrivals to compare with the recorded arrivals.

Optimizing bottom depth and target depth

In this test, bottom depth and target depth were chosen as the optimization parameters, with the purpose of investigating the performance of the method with varying geometric parameters and SNR levels. The geometric parameters used in the test were r_t from 2000 m to 10 000 m in steps of 2000 m, z_b from 200 m to 1000 m in steps of 200 m, and z_t from 50 m to 50 m above bottom depth in steps of 100 m. SNR levels were varied from 10 dB to 30 dB in steps of 2 dB. The source depth was held constant at 50 m. The nonlinear sound speed profile shown in figure 3.1 was used. The test was done using both three and five eigenrays, translating to 9 and 25 arrivals, for modelling when applying the comparison function. The eigenray finding procedure traced 3000 rays per run. The size of the target depth range was set to 15 m.

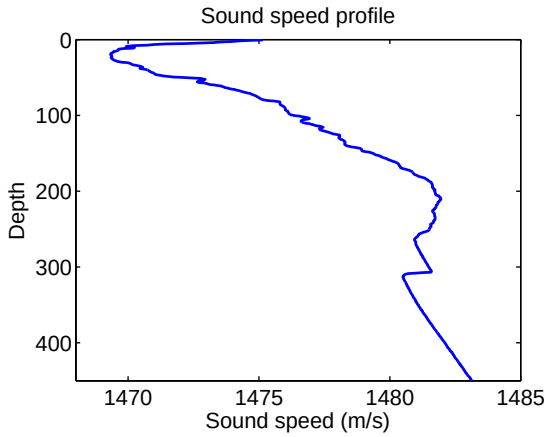

Figure 3.1: Sound speed profile used when optimizing with respect to bottom depth and target depth.

All three objective functions presented in section 2.8 were used in the test, to explore the difference in estimates provided by these. For each set of geometric parameters and each SNR level, ten iterations were made in which

a signal was synthesized by the method described in the preceding section, and the optimization procedure using the simplified comparison function was applied to this signal in order to estimate the target depth. The mean absolute error of these target depths estimates were then calculated. Also, for each set of geometric parameters and each SNR level, ten iterations were made in which arrivals were calculated and subjected to noise as described in the preceding section, before the optimization procedure using the Bayesian objective function was applied to these arrivals to estimate the target depth. The mean absolute error of these estimates was calculated. Due to the long evaluation time of the full comparison function, this objective function was tested for a smaller selection of parameters; r_t was varied from 2000 m to 10 000 m in steps of 4000 m, z_b from 200 m to 1000 m in steps of 400 m, and z_t from 50 m to 150 m above bottom depth in steps of 200 m. The SNR was still varied from 10 dB to 30 dB in steps of 2 dB. Five iterations equivalent to those made with the simplified comparison function were made with the full comparison function.

The results of this test are analyzed in four ways in section 4.2.1:

- The first analysis determines whether three eigenrays are sufficient in modelling signals for comparison with recorded signals or if five eigenrays should be used, based on results for the simplified comparison function.

- The second analysis looks at the error in target depth estimation as a function of SNR and target range, to see how sensitive the method is to increasing noise levels, and how sensitive it is to increasing target range. It also looks at the differences between the comparison and Bayesian objective functions.

- The third analysis is similar to the second, as it looks at the error in target depth estimation as a function of SNR and bottom depth to further determine the sensitivity of the method to noise, and to investigate the sensitivity to bottom depths. A comparison between the two objective function types is also given.

- The fourth analysis compares the results obtained by use of the full comparison function with the results obtained by use of the simplified comparison function and the Bayesian objective function, to see whether the simplified objective function is a reasonable approximation to the full comparison function, and to further explore the differences between the the comparison functions and the Bayesian objective function.

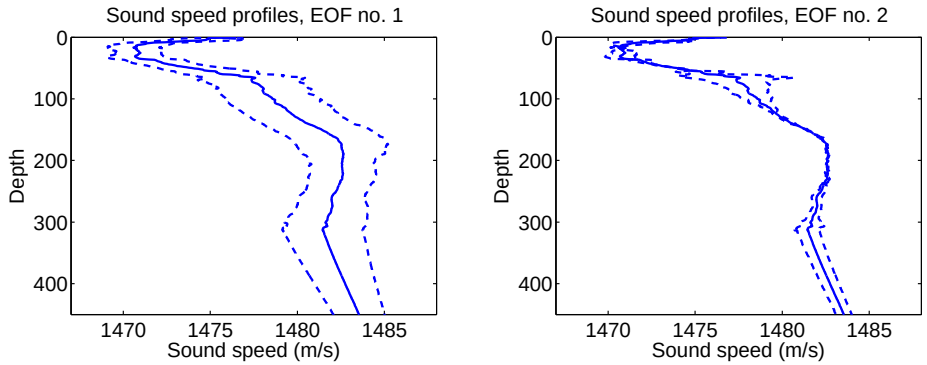

Figure 3.2: Maximal and minimal extent of sound speed profiles when varying EOF coefficient γ_i from $-\sqrt{\sigma_i}$ to $\sqrt{\sigma_i}$. Left: Varying γ_1. Right: Varying γ_2. Full lines indicate mean sound speed. Dashed lines indicate maximum and minimum extent of sound speed profiles considered.

Optimizing sound speed and target depth

In the next test, we look at optimization with respect to the EOF coefficients of the two most significant EOFs and the target depth. A data set of six sound speed profiles was used. From the sound speed data set, EOFs were extracted, with $\sigma_1 = 3.2296 \cdot 10^3$, $\sigma_2 = 0.2973 \cdot 10^3$, and $\sum \sigma_i = 3.6172 \cdot 10^3$. From this we see that the first EOF accounts for 89.28% of the variation within the data set, while the second accounts for 8.22% of the variation; in total, the two EOFs account for 97.5% of the variation, meaning they are well suited for approximating sound speed profiles. Had the data set been larger, this number may have been lower. The mean sound speed profile is shown in figure 3.2, along with the maximum and minimum extent of the variations made when varying the EOF coefficients.

During the test, r_t was varied from 2000 m to 10000 m in steps of 4000 m, and z_t from 60 m to 300 m in steps of 60 m. The EOF coefficients γ_1 and γ_2 were varied from -100% to 100% of $\sqrt{\sigma_1}$ in steps of 40% of $\sqrt{\sigma_1}$ and -100% to 100% of $\sqrt{\sigma_2}$ in steps of 40% of $\sqrt{\sigma_2}$, respectively. The bottom depth was kept constant at 450 m. The source depth was at 50 m. The SNR was kept constant at 25 dB. Only the simplified comparison function and the Bayesian objective function were used in this test, since using three optimization variables increased the computational complexity of the initialization procedure so much that optimization using the full comparison function used too much time.

For each set of parameters, ten iterations were made where a signal was synthesized with added Gaussian noise for use in the simplified comparison function and a set of arrivals for use in the Bayesian objective function, and the target depth estimated subsequently, before calculating the mean absolute error in estimates. In addition to the arrival noise and signal noise, an

element of uncertainty was added to the sound speed profiles used for synthesizing recorded signals by adding Gaussian noise distributed as $n(0, \sqrt{\sigma_i})$ to the EOF coefficients before synthesizing.

Sensitivity to sound speed profile

The different sound speed profiles obtained when varying the EOF coefficients in the preceding test are qualitatively similar, as seen in figure 3.2, yet it is also of interest to see how the method performs when qualitatively different sound speed profiles are being used, as is the case for sound speed profiles measured during different months of the year at a given geographic location. Figure 3.3 shows the seasonal variation of the sound speed profile at a location in the North Sea. Note the increasing sound speed near the surface due to increasing temperature during the summer months and the decrease during the winter months.

In this test, as the sound speed profiles are measured for depths up to 300 m, the bottom depth was held at a constant 300 m. The target range was varied from 2000 to 10000 m in steps of 2000 m, the target depth from 10 to 290 m in steps of 70 m, and the source depth was kept constant at 5 m to simulate a more realistic propagation situation, where the sonar system is located close to the surface. All different sound speed profiles shown in figure 3.3 were used for testing. The SNR was set to 25 dB. For each parameter setup and sound speed profile, when using the simplified comparison function and the Bayesian objective function, ten iterations were in which recorded signals or arrivals were modelled before applying the optimization procedure. When using the full comparison function, three iterations of modelling signals and applying the optimization procedure were done. The optimization parameters were target depth and bottom depth. Mean absolute errors in target depth estimate were noted.

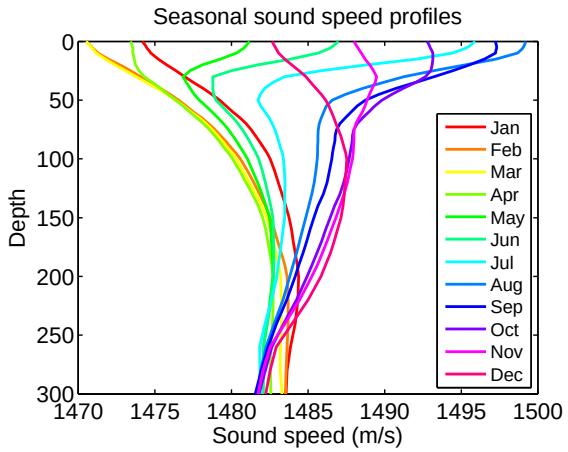

Figure 3.3: Sound speed profile exhibiting seasonal variations

3.2.3 Execution time

Finally, a test was done to investigate the execution time of the method using the different objective functions in different geometries, and to see the difference in time usage with an increasing number of rays traced in the eigenray finding procedure.

The test was performed by varying r_t from 2000 m to 10000 m in steps of 2000 m, and by varying z_b from 250 m to 650 m in steps of 200 m. The target depth was kept constantly at 200 m, and the source depth at 5 m. The test was first done with 3000 rays traced in the eigenray procedure, then redone with 5000 eigenrays. When testing the comparison objective functions, recorded signals were synthesized with a sampling frequency of 8000 Hz, in contrast with earlier tests which used a sampling frequency of 2000 Hz, meaning the time resolution of the signal was greater during this test. This did not make a difference in the accuracy of the method, but as real signals are sampled with similar frequencies, it is reasonable to test with realistic values here. In this spirit, the beamforming resolution was increased from 0.1 degrees to 1 degree, a more usual choice in real situations. When testing the Bayesian objective function, two eigenrays were used for synthesizing recorded arrivals, while both two and three eigenrays were used for modelling arrivals for comparison with the recorded arrivals.

For each choice of geometric parameters, the execution time of the method using the different objective functions was measured. This was done by first synthesizing a recorded signal (or recorded arrivals, in the case of the Bayesian objective function), then applying the method to obtain an estimate of the target depth by optimizing with respect to target depth and bottom depth. Only the time used when applying the method was measured, by means of the built-in MATLAB functions `tic` and `toc`. The execution time of the simplified comparison function was measured by carrying out ten such iterations per parameter setup and calculating the mean time usage. Similarly, the Bayesian objective function was tested with ten iterations. The full comparison function was tested using three iterations due to its long execution time.

The computer used for testing was a TOSHIBA Satellite 830W laptop, with an Intel i5-2467 CPU running at a clock rate of 1.6 GHz, 4 GB of RAM and Windows 7 64-bit version. It is worth noting that a more powerful computer may have achieved faster execution times, but as the relative time difference between objective functions is most of interest here, testing on this computer should be sufficient.

Chapter 4

Results and discussion

In this chapter, we present and discuss the results of the various tests described in section 3.2. We will start by presenting the results of the tests eigenray estimation procedure, before continuing with the results of the various tests on synthesized data and finishing with the results pertaining to execution time. All results are discussed as soon as they are presented.

4.1 Accuracy of eigenray estimation

Figure 4.1 shows the errors in eigenray depth at target range as calculated by use of equation (3.1) for all choices of r_t, z_b and z_t and a linear sound speed profile. Note the large improvement in accuracy from LYBIN version 6.1 to 6.2. Also note that both versions show diminishing accuracy as r_t increases, and as z_b decreases. This is to be expected, as ray tracing at large ranges requires more steps with the underlying numerical scheme than ray tracing at close range in deep waters, thus accumulating a larger numerical error, and since ray tracing in shallow waters is more sensitive to errors due to a higher number of bottom reflections. Still, in general, the eigenray estimation procedure is more than accurate enough when using LYBIN version 6.2; an error of 2 m at 8000 m translates to an error of less than 0.01 degrees in exit angle, which is less than the angular resolution used in most beamforming schemes.

However, we may also note that there appears to be band structures wherein estimate quality deteriorates. The estimated eigenrays for a case lying in one of these bands is shown in figure 4.2. It is obvious that the 'ls' and 'sl' eigenrays are incorrect. An explanation of this can be given based on figure 4.3. In this figure, we can see that one ray from each of the families 'ls' and 'sl' enters the target depth range, thereby causing these families to be counted among the eigenray families although the families do not include eigenrays. The problem in this case is alleviated by reducing size of the target depth range, such that the 'ls' and 'sl' families no longer enter the

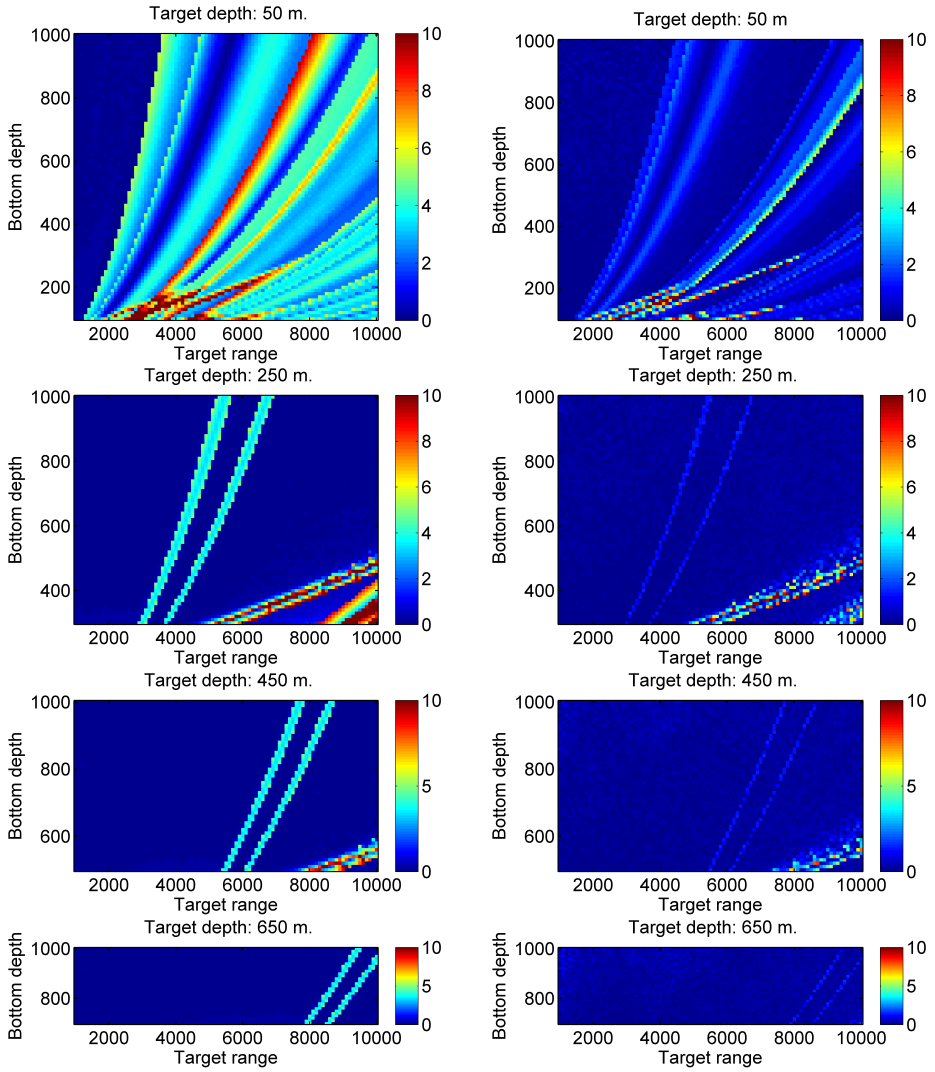

Figure 4.1: Mean error in eigenray depth at target range. Number of rays traced: 5000. Left: LYBIN version 6.1, target depth range: 80 m. Right: LYBIN version 6.2, target depth range: 15 m.

target depth range. Another way of dealing with this problem would be to require that ray families must enclose the target depth, that is, at least one ray from the family must end below the target depth and one above the target depth. Implementing this solution means changing the source code of LYBIN, which could not be done in the time frame of this work and is therefore considered a candidate for further work.

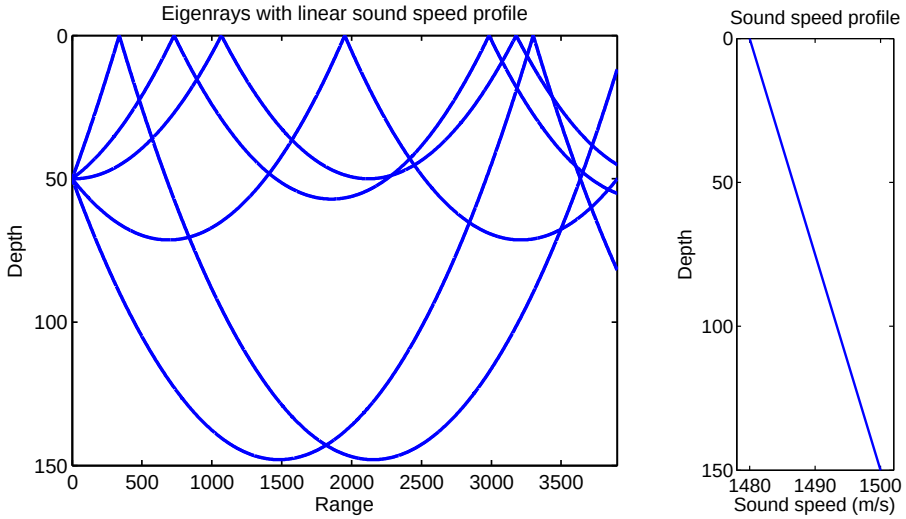

Figure 4.2: Numerical eigenrays. $z_s = 50$ m, $z_t = 50$ m, $z_b = 150$ m, $r_t = 3900$ m. Target depth range: 15 m.

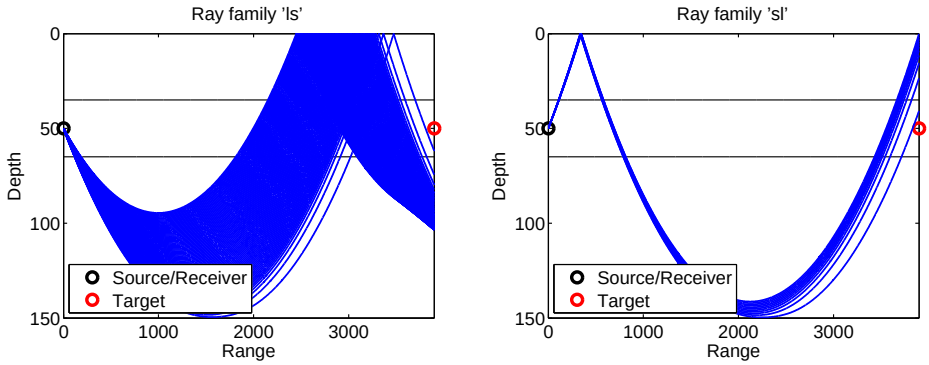

Figure 4.3: Ray families. Left: 'ls' family. Right: 'sl' family. Black horizontal lines denote edges of target depth range. $z_s = 50$ m, $z_t = 50$ m, $z_b = 150$ m, $r_t = 3900$ m. Target depth range: 15 m. Number of rays traced: 5000.

Sensitivity to target depth range size and number of rays used

Figure 4.4 shows the results of the test of eigenray sensitivity to the number of traced rays and the target depth range size for the case $z_b = 200$ m, $z_t = 50$ m, $z_s = 50$ m, $r_t = 6000$ m. We can see that reducing the target depth range size from 55 m to 15 m improves estimates, and that increasing the number of rays used increases accuracy, to a certain point. It would seem that there is little to gain from increasing the number of rays used beyond 1000, but figure 4.5 suggests otherwise. Here, we see a case where by increasing the number of rays from 2000 to 3000, a direct eigenray is identified. This shows that although accuracy does not improve noticeably beyond 1000 rays, tracing more rays allows for the discovery of new eigenray families with less transmission loss, which are of importance as they represent the paths most likely to result in a detectable signal. In the following sections, eigenray estimation is done with 15 m target depth range and 3000 rays traced.

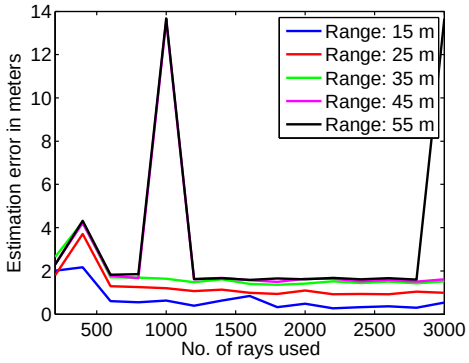

Figure 4.4: Mean error in eigenray depth at target range with varying number of rays and target depth range size. $z_s = 50$ m, $z_t = 50$ m, $z_b = 200$ m, $r_t = 6000$ m.

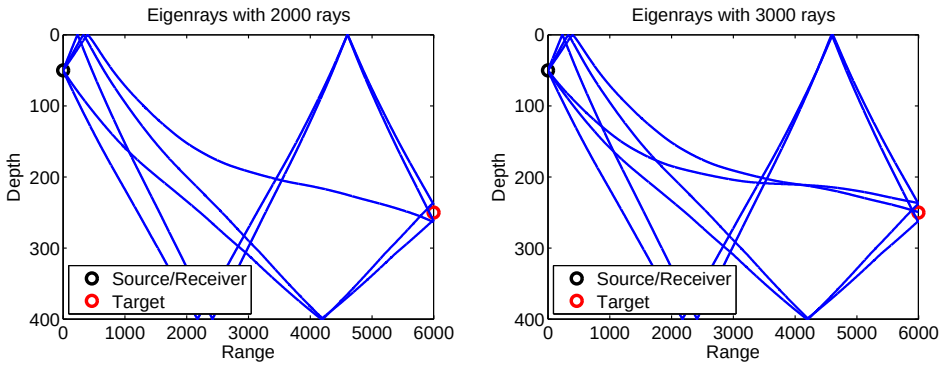

Figure 4.5: Eigenrays for a nonlinear sound speed profile using different numbers of eigenrays. Left: 2000 rays used. Right: 3000 rays used. $z_s = 50$ m, $z_t = 250$ m, $z_b = 400$ m, $r_t = 6000$ m.

4.2 Tests on synthesized data

Here, the results from the three tests of the target depth estimation method carried out on synthesized data, described in section 3.2.2, are presented and discussed. First, we consider the results of optimizing with respect to bottom depth and target depth. Next, we shall look at the results of optimizing with respect to two EOF coefficients and target depth. Finally, we present the results of the test with qualitatively different sound speed profiles.

4.2.1 Optimizing bottom depth and target depth

Number of eigenrays for signal modelling

Figure 4.6 shows the minimum, maximum and mean estimates of target depth obtained by use of the simplified comparison function on recorded signals synthesized with differing SNR levels for a set of geometric parameters. On the left side we see results when using three eigenrays for modelling signals to compare with the recorded signal, while on right side we see results using five eigenrays. We may observe that using five eigenrays provides more consistent and correct estimates for the target depth than using three eigenrays when SNR levels are low. That five eigenrays is superior to three is to be expected; using more eigenrays provides more information, which should lead to better estimates, especially when SNR is low. Low SNR levels makes it difficult to discern signal from noise during the thresholding described in chapter 2.6. Note that estimates improve as SNR increases, but that for low SNR values (10-14 dB), estimates fail since the SNR is close to or below the noise threshold value of 13 dB, such that arrivals risk being dismissed as noise. Also, the results are satisfying enough to discourage the use of more than five eigenrays. We will therefore look only at results obtained by use of five eigenrays for modelling in the following sections.

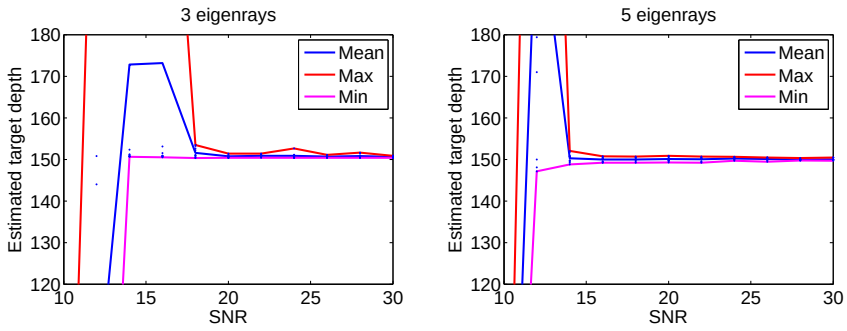

Figure 4.6: Estimates of target depth as a function of SNR. Dots represent individual estimates. Left: 3 eigenrays. Right: 5 eigenrays. $z_s = 50$ m, $z_t = 150$ m, $z_b = 600$ m, $r_t = 4000$ m.

Estimation error as a function of SNR and target range

The plots in figure 4.8 show the error in target depth estimation as a function of SNR and target range for four different target depths. The left column contains errors occurring when using the simplified comparison function, while the right column contains errors occurring when using the Bayesian objective function.

First, we may observe the positive result that the errors are mostly within acceptable range for classification purposes irrespective of objective function; we only need an approximate estimate for target depth to say whether it is close to the bottom or not, and estimates with errors of less than 50 m are considered good enough for this purpose. Second, we may note the failure of the simplified comparison function in the case where $z_t = 250$ m and $r_t = 8000$ m, as well as the case where $z_t = 350$ m and $r_t = 4000$ m. We will now attempt to explain this failure.

In the case with $z_t = 350$ m and $r_t = 4000$ m, the method estimates z_t to be 55 m. Figure 4.7 shows the recorded signal in this case, synthesized with $z_t = 350$ m, along with the signal modelled with $z_t = 55$ m. We see that the signal produced with $z_t = 55$ m has all arrivals clustered around 5.44 seconds and 2 degrees, whereas the recorded signal has its arrivals more spread out, although with a maximum strength near 5.44 seconds.

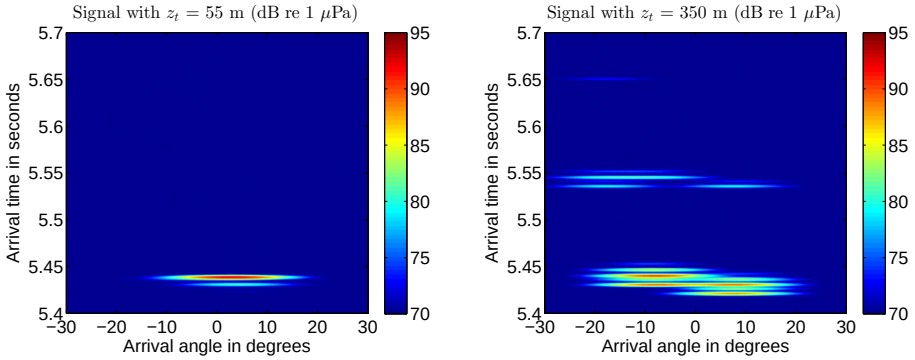

Figure 4.7: Synthesized signal. Left: Modelled signal, $z_t = 55$ m. Right: Recorded signal, $z_t = 350$ m. $z_s = 50$ m, $z_b = 400$ m, $r_t = 4000$ m.

Based on this observation, the failure may be attributed to the shortcomings of the simplified comparison function; as explained in chapter 2.8.1, the simplified comparison function will, if possible, prefer parameters producing arrivals clustered around the strongest part of the recorded signal. The objective function values for various choices of bottom depths and target depths is shown in figure 4.9. From this figure we can see that although there are local minima near a target depth of 350 m, there is also a minimum at a target depth of 55 m which is marginally stronger, thereby constituting the global minimum. Since the faulty estimates occur due to the problem

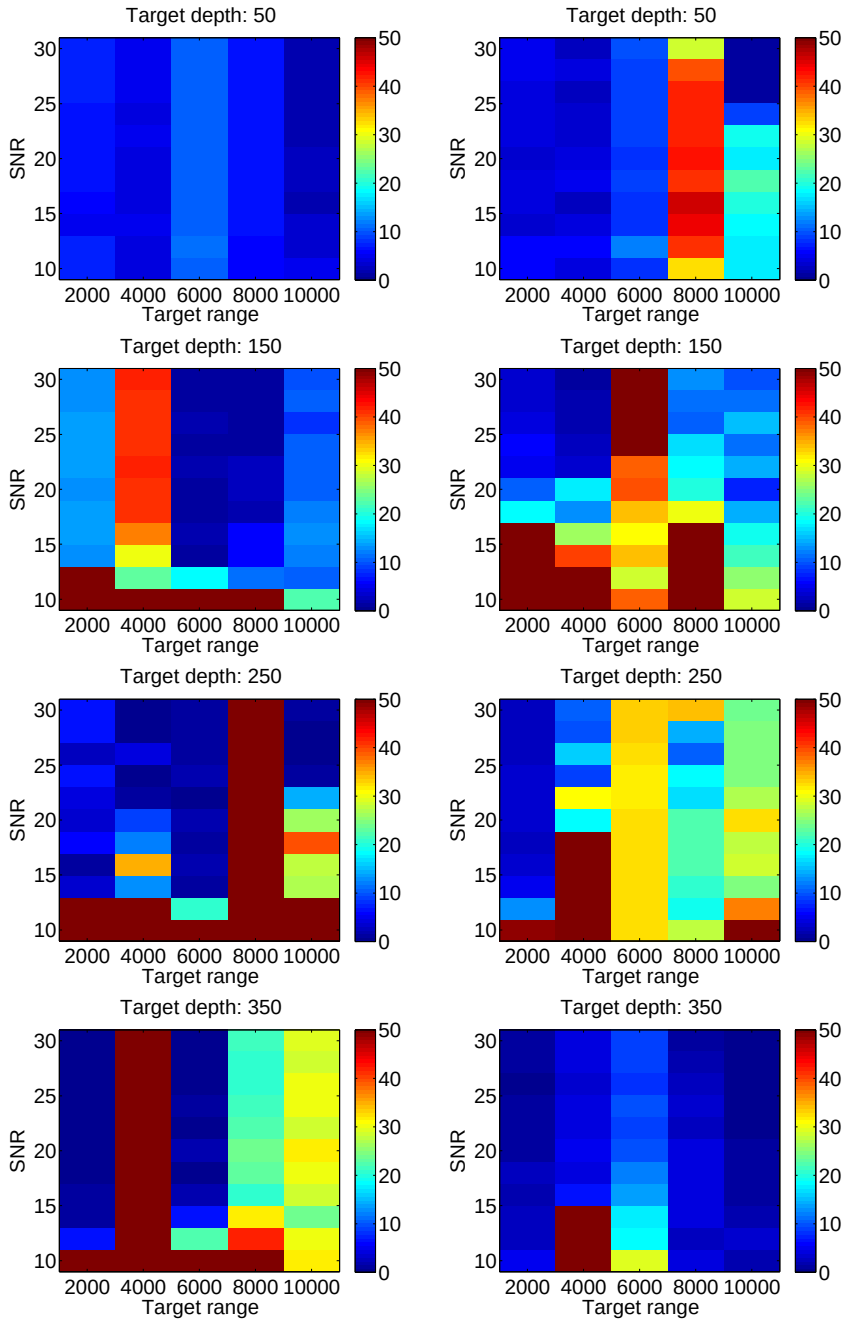

Figure 4.8: Mean error in estimates of target depth as a function of SNR and target range. Bottom depth: 400 m. Left column: Simplified comparison function. Right column: Bayesian objective function.

geometry, the error persists independent of SNR values, as seen in figure 4.8. Running the estimation with the same parameters but using the full comparison function yields correct estimates for the target depth. Since the full comparison function does not have the same affinity for clustering as the simplified comparison function, this should solidify the explanation that the incorrect target depth estimate is due to the flaws in the simplified comparison function and thereby is unavoidable. As a side note, we may observe that the minimum at $z_t = 55$ m observed in figure 4.9 is valid for all bottom depths; this is since the signal at $z_t = 55$ m is modelled by eigenrays with no bottom reflections, thus the modelled signal is the same for all bottom depths.

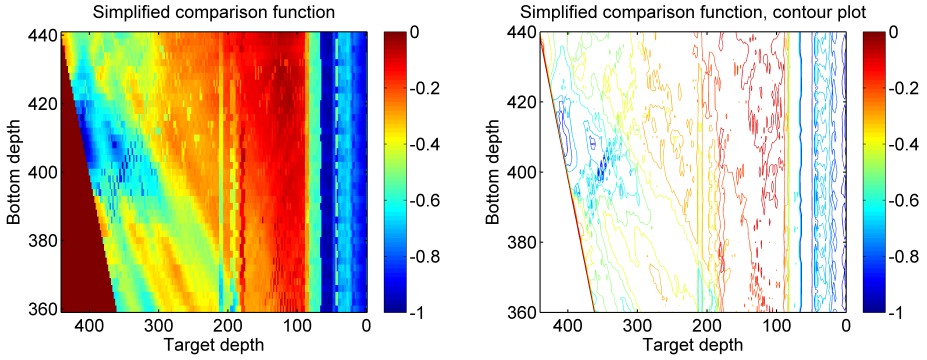

Figure 4.9: Simplified comparison function for various model bottom depths and target depths. Correct model parameters: $z_t = 350$ m, $z_b = 400$ m.

We now turn to the case where $z_t = 250$ m and $r_t = 8000$ m. Upon closer inspection of results here, it is revealed that in three out of ten iterations with SNR = 20, the target depth is estimated successfully to 250 m ± 1 m. However, in the remaining seven cases, the method was unsuccessful, yielding target depth estimates of 50 m. An analysis of this case shows the same situation as for $r_t = 4000$ m, $z_t = 350$ m, with arrivals clustering at the strongest part of the recorded signal. However, this is a borderline case; the random noise added to arrival angles and arrival times times as described in section 3.2 can decide whether the minimum of the simplified comparison function is located at a target depth of 250 m or 50 m, explaining why three of the ten iterations yield correct results for this parameter setup. This observation would suggest that in certain cases, the simplified comparison function is sensitive to disturbances in arrival times and arrival angles.

Next, we look at the performance of the Bayesian objective function. Figure 4.8 shows that the Bayesian objective function fails to provide estimates of sufficient accuracy for classification purposes for just one set of parameters, when $r_t = 6000$ m and $z_t = 150$ m. In this case, the estimate accuracy deteriorates with increasing SNR. The figure would imply mediocre performance with an SNR of 20; when looking at the results in detail, it

turns out that here, too, three out of ten iterations produced a target depth estimate of ~155 m, which is close enough to the true target depth. The seven other estimates place the z_t at 210 m with $z_b = 370$ m. This behaviour may be due to too few eigenrays being used in the optimization. As can be seen from table 4.1, for the true geometric parameters there are four eigenrays available with very similar transmission loss properties, meaning that small changes in bottom depth or target depth will influence which eigenrays are the strongest and thus used for modelling arrivals. Indeed, just by using 4000 rays in the eigenray procedure instead of 3000, we see from table 4.2 that the ordering of the eigenrays change.

No.	Exit angle	Travel time	TL	History
1	10.2°	4.103 s	85.8 dB	'bs'
2	-10.8°	4.116 s	86.0 dB	'sbs'
3	-8.9°	4.083 s	86.1 dB	'sb'
4	8.4°	4.073 s	86.4 dB	'b'

Table 4.1: Eigenrays with $z_s = 50$ m, $z_t = 150$ m, $z_b = 400$ m, $r_t = 6000$ m. Rays traced: 3000.

No.	Exit angle	Travel time	TL	History
1	10.2°	4.103 s	85.9 dB	'bs'
2	-8.9°	4.083 s	86.0 dB	'sb'
3	-10.8°	4.116 s	86.1 dB	'sbs'
4	8.4°	4.073 s	86.5 dB	'b'

Table 4.2: Eigenrays with $z_s = 50$ m, $z_t = 150$ m, $z_b = 400$ m, $r_t = 6000$ m. Rays traced: 4000.

The test was carried out while using only two eigenrays for the recorded arrivals and two eigenrays for modelled arrivals. Thus, while the recorded arrivals may be obtained using one pair of eigenrays, subsequently modelled arrivals using nearly the same parameters as the recorded arrivals may be modelled using a different eigenray pair, resulting in the modelled arrivals being non-matching when, in fact, the model parameters are nearly correct. This means that the global minimum may be missed. Additionally, with a target depth of 210 m and bottom depth of 370 m, the two strongest eigenrays have nearly identical exit angles and travel times to those of the eigenrays used in the recorded arrivals, meaning that when using two eigenrays, $z_t = 210$ m and $z_b = 370$ m looks like a good candidate since all arrivals match the recorded arrivals. However, if more eigenrays were used, both in

modelling the recorded arrivals and the arrivals used for matching, fewer arrivals would match for this erroneous choice of parameters.

Also, as the SNR increases, σ_t and σ_θ decreases, meaning the objective function becomes more sensitive to discrepancies in arrival time and arrival angle. Since random noise is added to the recorded arrivals, this means that arrivals modeled using the correct model parameters will not generally match the recorded arrivals, and even less so as the objective function becomes more sensitive. Thus, a starting point close to but not exactly at the correct model parameters will be preferred. However, due to the eigenray problems explained in the preceding paragraph, these close model parameters will produce non-matching arrivals when using two eigenrays, and so the arrivals with a target depth of 210 m and bottom depth of 370 m are increasingly chosen as the best match as SNR increases.

The problem may be corrected by increasing the number of eigenrays used for modelling arrivals. When increasing the number of eigenrays used for modelling arrivals for matching from two to three, the target depth estimate became 130 m for all iterations, a significant improvement. This would indicate that a more sophisticated method of determining which eigenrays to use may be beneficial, for example determining the number of eigenrays to use based on the transmission losses of the eigenrays instead of using a predetermined number of eigenrays, for example that all eigenrays with transmission losses within 5 dB of the strongest arrival should be used for modelling arrivals.

Otherwise, we can see from figure 4.8 that the Bayesian objective function seems to be more sensitive to the SNR values than the simplified comparison function. This can be attributed to the Bayesian objective function being more sensitive to the added noise than the simplified comparison function; whereas the simplified comparison function smears arrivals over a quite large area and therefore is more robust to perturbations in arrival angles and arrival times, the Bayesian objective function requires the modelled arrivals to be quite close to the recorded arrivals to make an impact on the value of the function. It may also be due to the implementation error mentioned in section 3.1; redoing the test with corrected values of σ_t proposed in section 2.8.2 may reveal a different behaviour, since σ_t^c is independent of SNR.

We may also observe that while both functions display a general trend of improved estimates for shorter ranges, the Bayesian objective function seems more sensitive to target range than the simplified objective function. Once again, this may be due to the Bayesian objective function being more sensitive to disturbances in arrival angles and arrival times than the simplified comparison function. At longer ranges, eigenray estimates are less accurate, as seen in section 4.1, which may result in larger variations in arrival angles and arrival times due to numerical errors. This would affect the Bayesian objective function negatively. However, if the test is redone with corrected values of σ_t, this behaviour may be subdued since the σ_t^c increases

with target range, making the Bayesian objective function less sensitive to disturbances in arrival angle and arrival times at larger ranges.

As a final observation, we may see that excepting cases with SNR = 10-12 dB, the two objective functions never produce faulty estimates simultaneously, a trend we shall see continuing in the following results.

Estimation error as a function of SNR and bottom depth

Figure 4.10 shows the error in target depth estimates as a function of SNR and bottom depth for four different target ranges. The left column contains errors occurring with use of the simplified comparison function, while the right column contains errors when using of the Bayesian objective function.

Again, we see that most estimates using both methods are suitable for classification purposes, yet we may observe that the simplified objective function breaks down in some cases. Inspection of the results reveals that the analysis from the preceding section applies here as well; clustering of arrivals lead to an incorrect global minimum in the objective function. We also see once more that the Bayesian objective function, while somewhat less accurate, still reliably produces acceptable estimates in all cases but one, with $z_b = 400$ m, $r_t = 6000$ m and $z_t = 150$. This is the same parameter setup that caused the trouble seen in the previous section, and the analysis from the previous section still applies here. Disregarding the failed cases, there seems to be a slight but not definite trend toward better estimates for larger bottom depths when viewing the results from the simplified comparison function. Errors in the Bayesian objective function seem to be largely independent of bottom depth. The trend of more SNR dependency for the Bayesian objective function than the simplified comparison function seems to persist in these results. We may also note that here, too, erroneous estimates in the Bayesian objective function and the simplified comparison function do not occur simultaneously except for at low SNR values, where poor estimates are to be expected.

Comparison of objective functions

Figures 4.11 and 4.12 show the error in target depth estimation as a function of SNR and bottom depth, and as a function of SNR and target range, respectively. In both figures, the leftmost column contains estimation errors occurring after applying the full comparison function, the middle column contains estimation errors occurring after use of the simplified comparison function, and the rightmost column contains estimation errors occurring after use of the Bayesian objective function. In figure 4.11 three target ranges are used while z_t is kept at 50 m, and in figure 4.12, five target depths are used while z_b is kept at 1000 m.

In both figures, we can see that the full comparison function is slightly

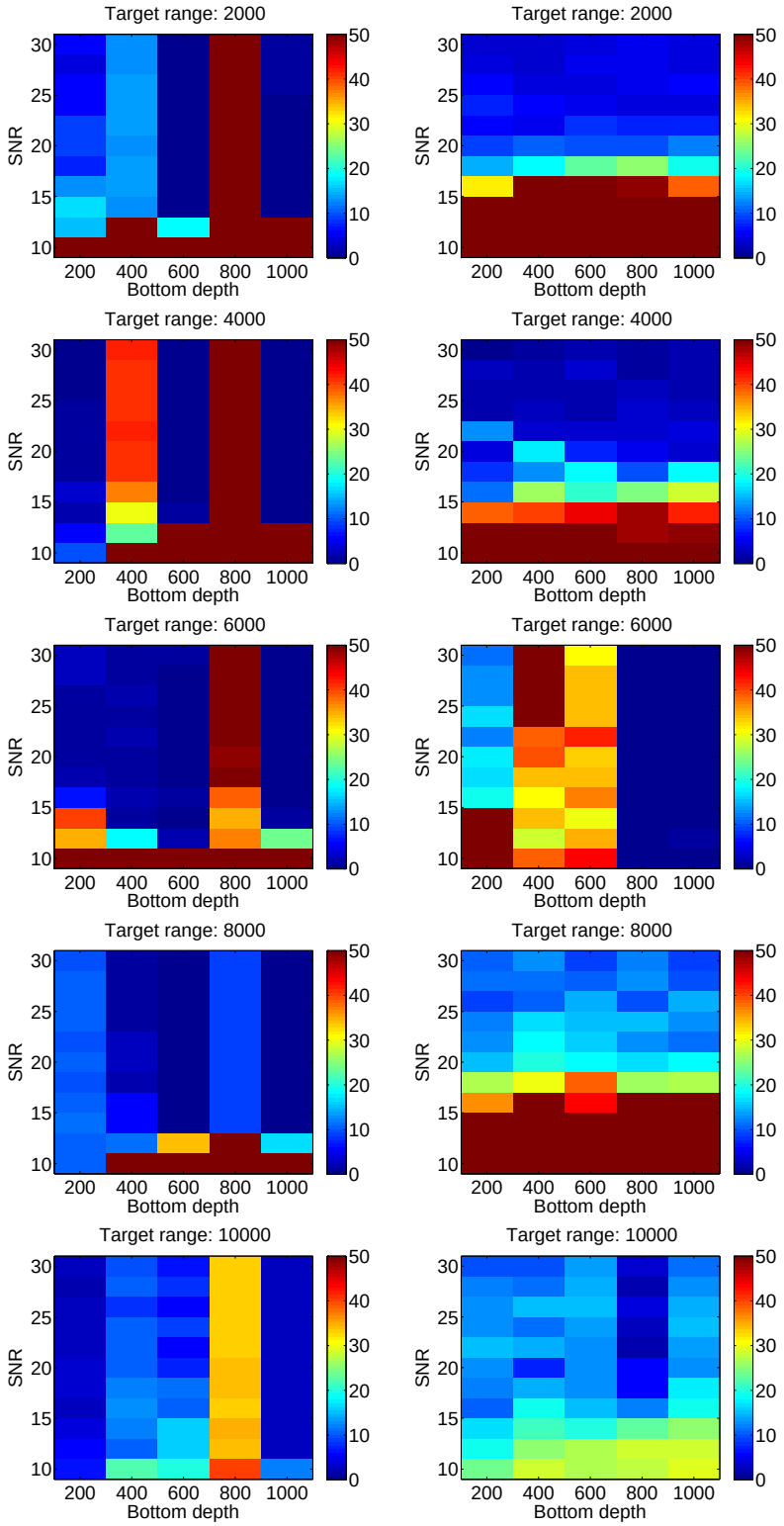

Figure 4.10: Mean error in estimates of target depth as a function of SNR and bottom depth. Target depth: 150 m. Left column: Simplified comparison function. Right column: Bayesian objective function.

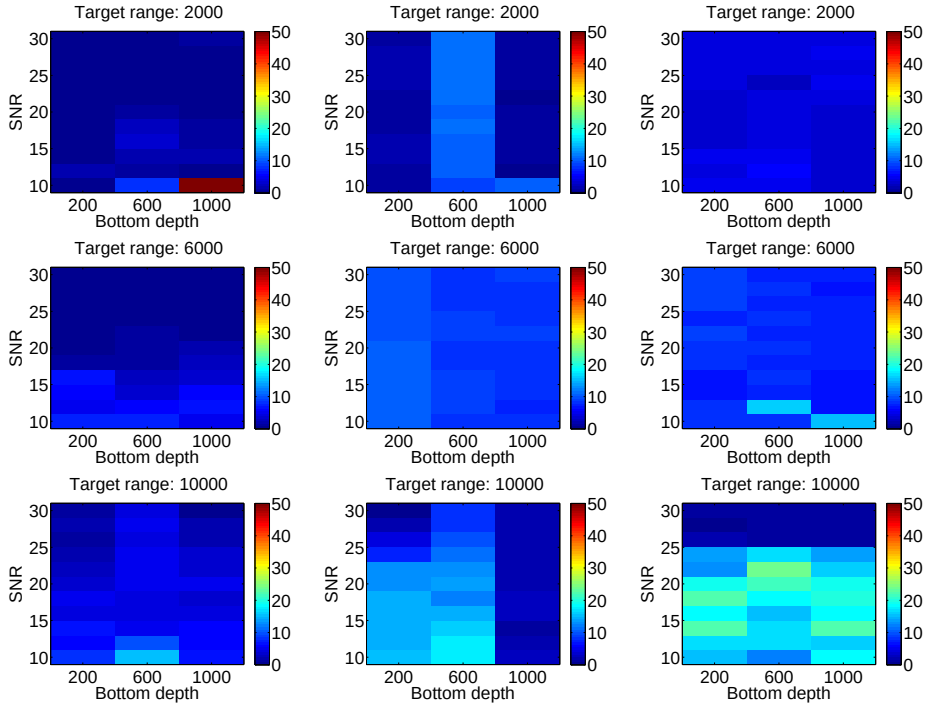

Figure 4.11: Mean error in estimates of target depth as a function of SNR and bottom depth. Target depth: 50 m. Left: Full comparison function. Middle: Simplified comparison function. Right: Bayesian objective function.

better than the simplified comparison function, but not decisively so. It should therefore be reasonable to use the simplified comparison function instead of the full comparison function as long as arrival clustering is not a possibility. We may also notice from figure 4.12 that the Bayesian objective function still performs worse at long ranges than the comparison functions. Again, it would be interesting to see whether the use of more than two eigenrays and use of the corrected σ_t would improve this aspect of the Bayesian objective function, and such a test should be carried out in the future.

Once again, we see that the Bayesian objective function and the simplified comparison function seldom produce faulty estimates at the same time, which would imply that this behaviour is consistent throughout all tests. This may be useful; due to its large computational cost, the full objective function may not be viable for use in a real-time system. However, a strategy for estimating target depths may be formulated based on first using both the Bayesian objective function and the simplified comparison function to provide estimates. Then, if the two estimates differ by a distance less than some predetermined value, for example 50 m, one may be fairly certain that the estimates are good. If the estimates do not match, then the full objective function may be applied in an attempt to ascertain the target depth.

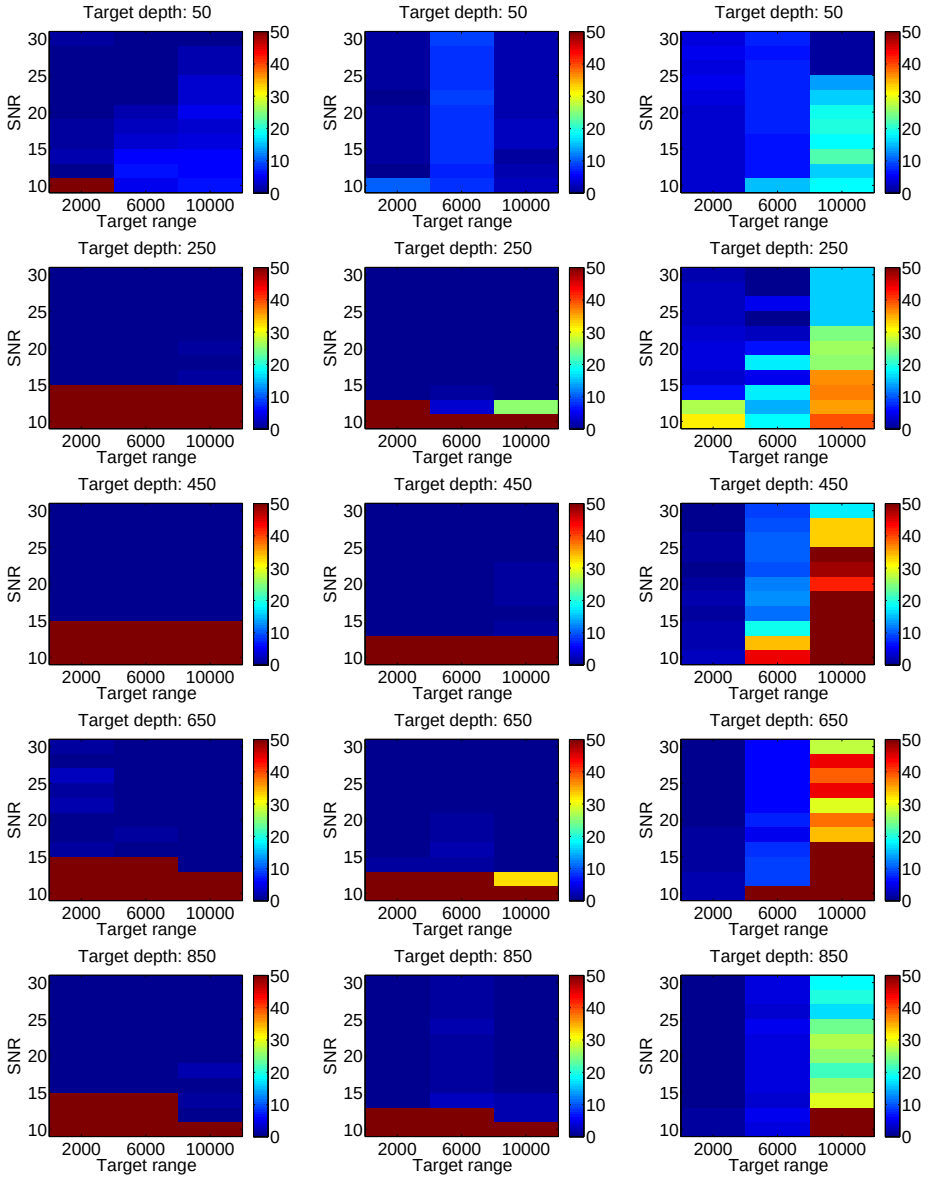

Figure 4.12: Mean error in estimates of target depth as a function of SNR and target range. Bottom depth: 1000 m. Left: Full comparison function. Middle: Simplified comparison function. Right: Bayesian objective function.

4.2.2 Optimizing sound speed and target depth

Figures 4.14 and 4.17 show the error in target depth estimation when optimizing with respect to the two leading EOF coefficients γ_1 and γ_2, and target depth. Figure 4.14 shows the error as a function of γ_1 and target range, and figure 4.17 shows the error as a function of γ_2 and target range. To clarify, the coefficient values in the figures signify the EOF coefficients used when synthesizing the recorded signal. In both figures, results for different target depths are displayed, with the left column containing estimation errors occurring after applying the simplified comparison function and the right column containing estimation errors occurring after use of the Bayesian objective function. Also common to both figures is the bottom depth, which is kept constantly at 450 m. In figure 4.14, the γ_1 used when synthesizing is varied between -100% and 100% of $\sqrt{\sigma_1}$ while the γ_2 used when synthesizing is kept at -20% of $\sqrt{\sigma_2}$. In figure 4.17, the γ_2 used when synthesizing is varied between -100% and 100% of $\sqrt{\sigma_2}$ while the γ_1 used when synthesizing is kept at 20% of $\sqrt{\sigma_1}$. To illustrate the space of sound speed profiles considered when optimizing with respect to γ_1 and γ_2, figure 4.13 shows the sound speed profiles generated when using several different values of γ_1 and γ_2, along with the mean sound speed profile of the data set from which the EOFs originate.

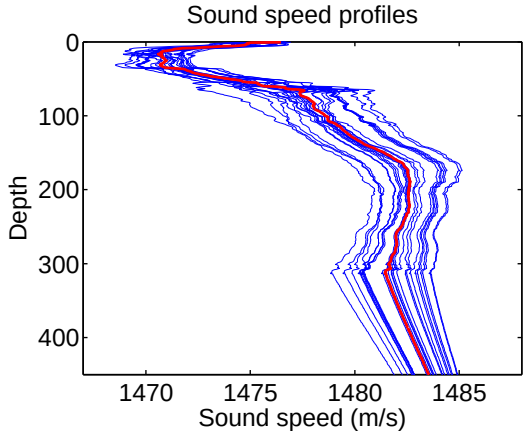

Figure 4.13: The space of sound speed profiles considered when optimizing with respect to EOF coefficients. Red: Mean sound speed profile. Blue: sound speed profiles generated by use of EOFs.

Estimation error as a function of γ_1 and target range

Looking at figure 4.14, we first consider the performance of the simplified comparison function. It generally provides accurate estimates, failing in 18 out of 90 cases but otherwise working well, even at large ranges. The failed

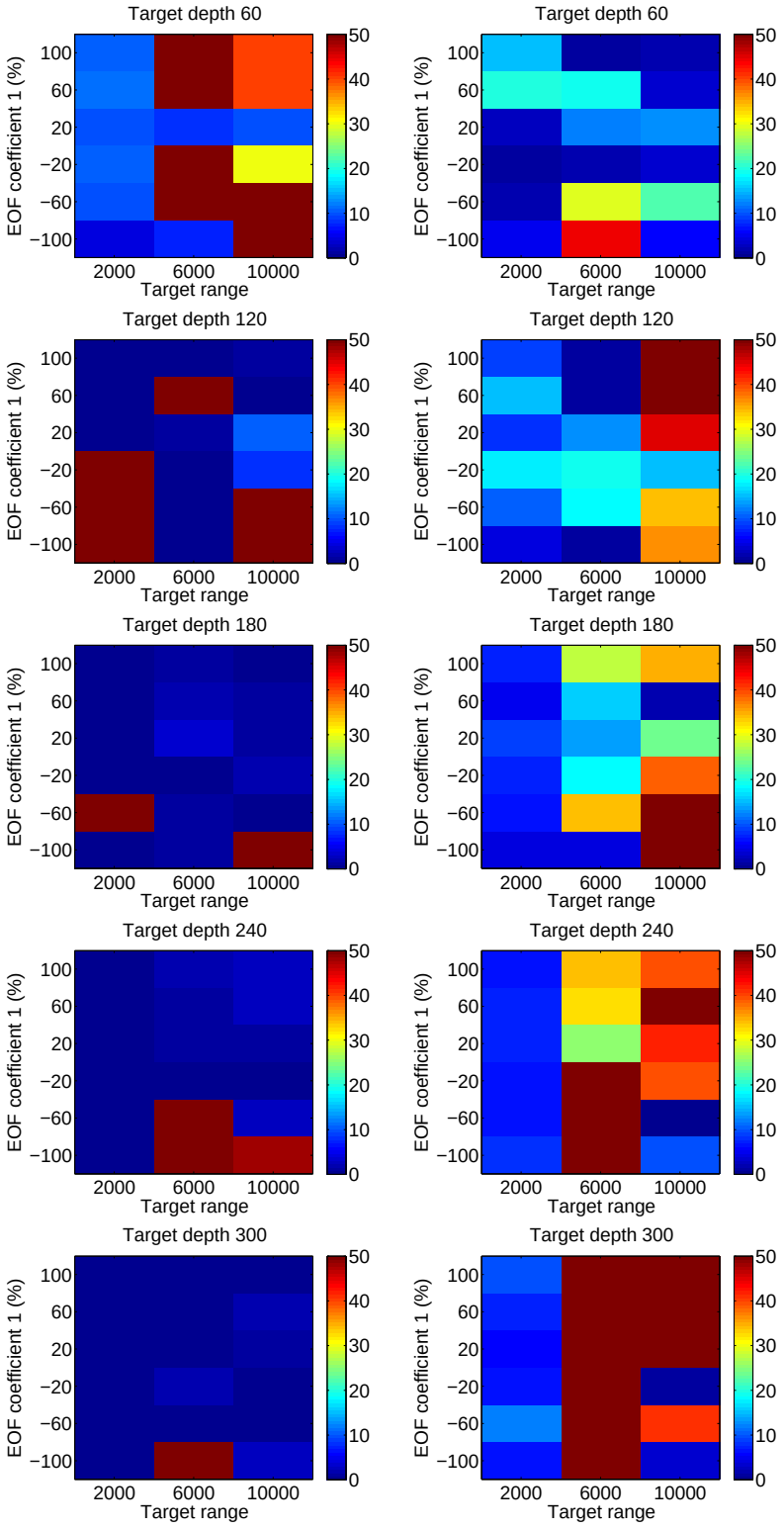

Figure 4.14: Mean error in estimates of target depth as a function of target range and EOF coefficient 1. EOF coefficient 1 measured in percentage of $\sqrt{\sigma_1}$. Bottom depth: 450 m. SNR: 25 dB. EOF coefficient 2: -20% of $\sqrt{\sigma_2}$. Left column: Simplified comparison function. Right column: Bayesian objective function.

cases will now be investigated further. We may look at the case when $z_t = 60$ m, $r_t = 6000$ m and $\gamma_1 = -0.2\sqrt{\sigma_1}$. Table 4.3 displays the eigenrays used to synthesize the recorded signal in this case. The optimization procedure suggests $z_t = 10$ m, $\gamma_1 = 0.4\sqrt{\sigma_1}$ and $\gamma_2 = 0.8\sqrt{\sigma_2}$ as the optimal parameters. The sound speed profile this produces is contrasted with the correct sound speed profile in figure 4.15. We see that the suggested sound speed profile has a sharper increase in sound speed near the surface and in the 75-150 m depth zone, which traps more eigenrays in the upper water body due to more acute refraction of rays. This allows for the eigenrays shown in table 4.4. Note that many of these eigenrays have no surface collisions, leading to lower transmission loss and thereby stronger arrivals.

No.	Exit angle	Travel time	TL	History
1	-4.7°	4.076 s	70.4 dB	'sls'
2	3.2°	4.077 s	74.6 dB	'lslsl'
3	-3.8°	4.076 s	75.3 dB	'slsl'
4	3.6°	4.077 s	76.8 dB	'lsls'
5	2.2°	4.078 s	85.2 dB	'lulu'

Table 4.3: Eigenrays used for synthesizing the recorded signal. $z_s = 50$ m, $z_t = 60$ m, $z_b = 450$ m, $r_t = 6000$ m. $\gamma_1 = -0.2\sqrt{\sigma_1}$, $\gamma_2 = -0.2\sqrt{\sigma_2}$.

No.	Exit angle	Travel time	TL	History
1	-2.9°	4.077 s	69.7 dB	'slsl'
2	1.0°	4.077 s	71.9 dB	'lululu'
3	-1.4°	4.077 s	74.1 dB	'ululu'
4	1.4°	4.077 s	75.3 dB	'lulul'
5	-1.6°	4.077 s	78.7 dB	'ulul'

Table 4.4: Eigenrays used for modelling the signal matched to the recorded signal. $z_s = 50$ m, $z_t = 10$ m, $z_b = 450$ m, $r_t = 6000$ m. $\gamma_1 = 0.4\sqrt{\sigma_1}$, $\gamma_2 = 0.8\sqrt{\sigma_2}$.

We may now look at the signals shown in figure 4.16. On the left side, we see the recorded signal, produced from the eigenrays in table 4.3. On the right side we see the signal obtained by the optimal model, produced from the eigenrays in table 4.4, which is nearly identical to the recorded signal but with greater signal strength. The extra signal strength causes the simplified objective function to choose $z_t = 10$ m, $\gamma_1 = 0.4\sqrt{\sigma_1}$, $\gamma_2 = 0.8\sqrt{\sigma_2}$ as a better match than the original model parameters; the stronger arrivals cause the products $\mathbf{m_k}^T\mathbf{s}$ used in equation (2.24) to be larger for the incorrect model parameters than the original model parameters. This can be compared to the analysis in section 4.2.1, where the recorded signal had several distinct arrivals, and a failure occurred due to the method producing a signal with

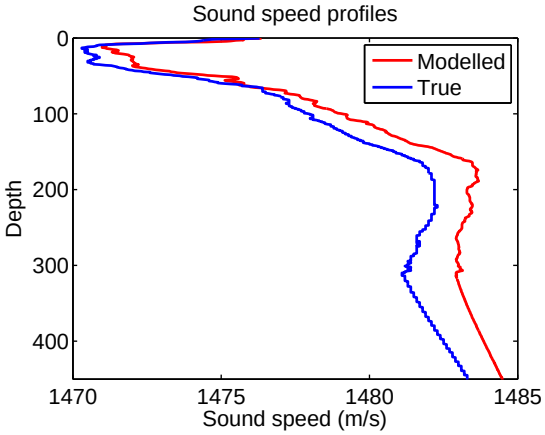

Figure 4.15: Sound speed profiles. Blue: Sound speed profile used when synthesizing recorded signal. Red: Sound speed profile suggested by the method.

a single strong cluster centered at the strongest arrival among the recorded arrivals. Here, the recorded signal consists of such a single strong cluster, but the method produces parameters resulting in an even stronger signal which otherwise resembles the recorded signal, causing a failed estimate.

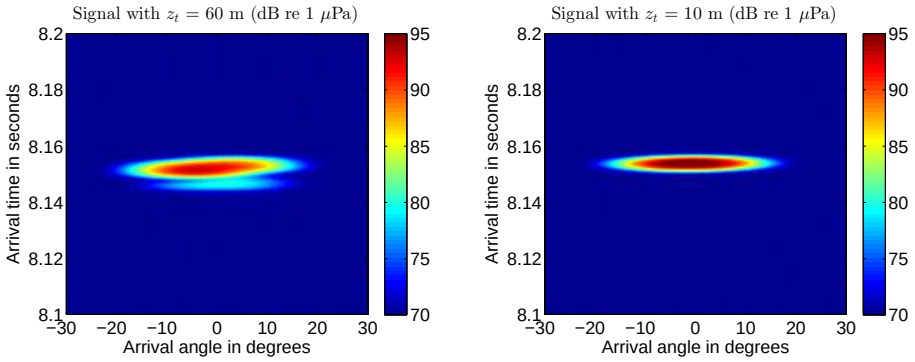

Figure 4.16: Left: Recorded signal, with $z_t = 60$ m, $\gamma_1 = 0.2\sqrt{\sigma_1}$, $\gamma_2 = -0.2\sqrt{\sigma_2}$. Right: Optimal signal, with $z_t = 10$ m, $\gamma_1 = 0.4\sqrt{\sigma_1}$, $\gamma_2 = 0.8\sqrt{\sigma_2}$. Both: $z_s = 50$ m, $z_b = 450$ m, $r_t = 6000$ m.

Analysis of the other failed cases for the simplified comparison function reveals the cause of all failures to be the same; the sound speed profile is changed to focus eigenrays to yield a single, strong arrival that corresponds to the strongest arrival in the recorded signal. We may note that there is a large number of errors when $z_t = 60$ m and $r_t = 6000$ m, which would imply that this particular geometry provides ideal conditions for the focusing described earlier. This is reasonable; the target is situated in the sound channel, and the recorded signal will resemble the signal in figure 4.16 for most sound speed profiles. The method will then alter the sound speed

profile to focus rays into even stronger arrivals, mimicking the situation analysed in the preceding paragraph. A significant observation is that all the false predictions made in this way placed the target depth closer to the surface than the correct target depth. This may be explained in light of the earlier analysis; the distinct minimum in sound speed at 25-30 m depth will cause many rays to refract toward this area, creating many strong eigenrays for depths here, and an ideal situation for focusing arrivals into one strong cluster. If the clustered arrivals happen to match a strong arrival in the recorded signal, the simplified comparison function will value this strongly, as seen earlier, and therefore the method will predict a target depth close to the surface. As the failures are due to a flaw in the objective function, it does not seem likely that they can be remedied unless changes are made to the simplified comparison function.

The Bayesian objective function exhibits some of the behaviour seen from earlier results, with estimate accuracy deteriorating as the target range increases. The most glaring errors are seen for a target range of 6000 m and target depth of 300 m. What happens in this geometry is similar to the situation where the Bayesian objective function fails in the results from section 4.2.1; when estimating the eigenrays used for generating the recorded signals, there are several eigenrays with very similar transmission losses, and small perturbations in model parameters alter the choice of eigenrays due to fluctuations in transmission loss, leading to mismatching of arrivals. Additionally, it is possible to alter the sound speed profile to find two eigenrays with similar arrival times and angles to the two eigenrays used for synthesizing the recorded arrivals. If three eigenrays were used, this may have been prevented, as not all arrivals produced with the erroneous model parameters match with the recorded arrivals when three eigenrays are used. Inspection of results shows that the Bayesian objective function exhibits no general trend in whether erroneous estimates are over- or underestimated.

Failed cases aside, we may note that the method largely predicts the target depth with sufficient accuracy for classification purposes, and that there is no clear dependency on the value of γ_1 used. It is worth noting that the Bayesian objective function performs best at shorter ranges. Also, we can once again observe that the two objective functions are seldom simultaneously wrong in their target depth estimates.

Estimation error as a function of γ_2 and target range

Looking at figure 4.17, we start by considering the performance of the simplified objective function. As for the preceding results, we see that the overall performance is good; the target depth was estimated correctly in all but 13 out of the 90 cases considered. Closer inspection reveals that the explanation for the failed estimates is the same as earlier; the sound speed profile is altered to focus eigenrays such that arrivals are clustered and hit a strong

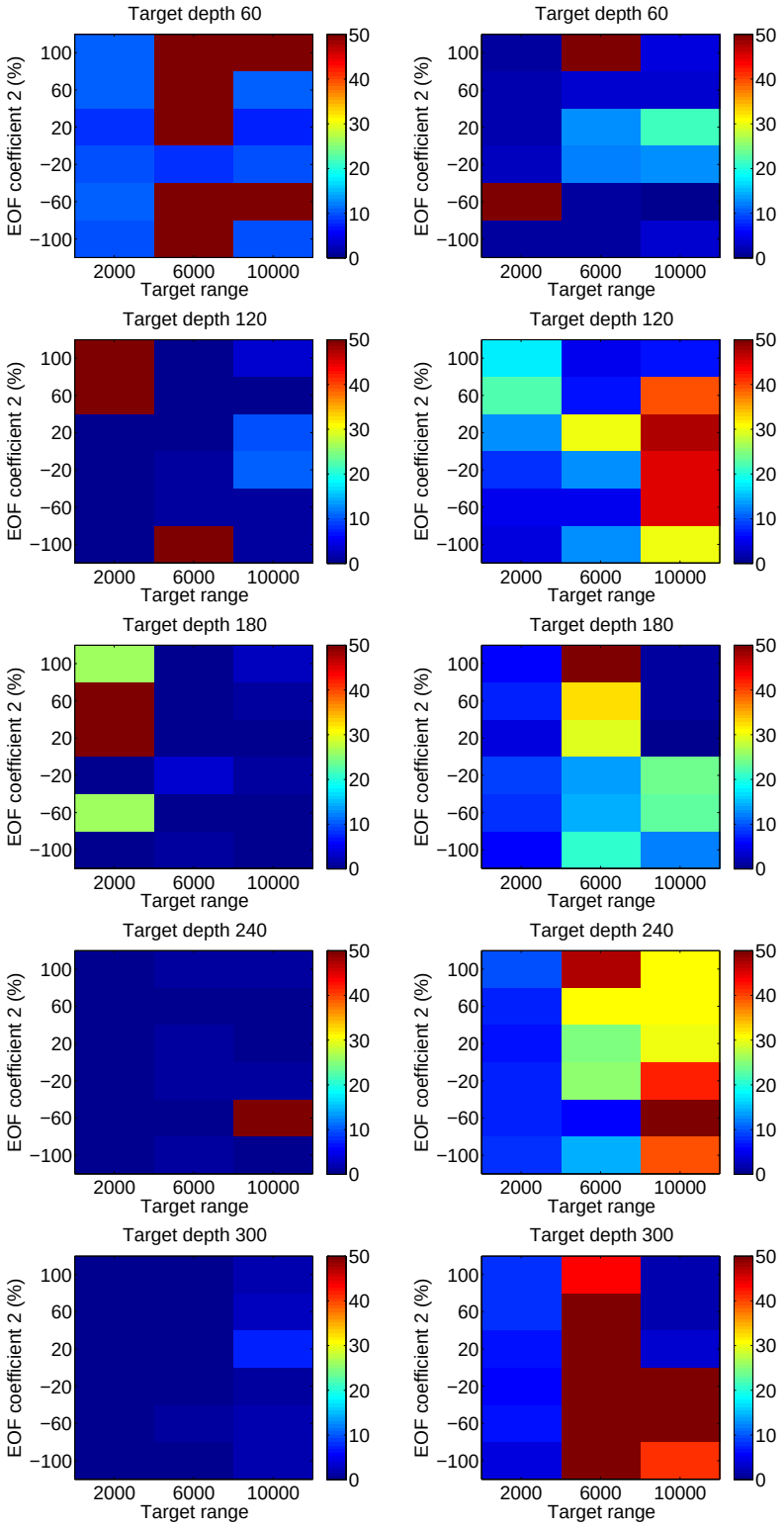

Figure 4.17: Mean error in estimates of target depth as a function of target range and EOF coefficient 2. EOF coefficient 2 measured in percentage of $\sqrt{\sigma_2}$. Bottom depth: 450 m. SNR: 25 dB. EOF coefficient 1: 20% of $\sqrt{\sigma_1}$. Left column: Simplified comparison function. Right column: Bayesian objective function.

arrival in the recorded signal or, when the recorded signal is a single strong cluster, focuses arrivals into an even stronger cluster. We may observe that as in the preceding section, there is a large number of errors with the simplified comparison function for $r_t = 6000$ m, $z_t = 60$ m, further cementing the assertion that the errors occur as a result of the geometry in these cases.

The Bayesian objective function performs on par with than the simplified comparison function; when disregarding the failure for the case with $r_t = 6000$ m and $z_t = 300$ m where we once again see failed estimates for five out of six choices of γ_2 which may be due to problematic geometry, most of the estimates provided are exact enough for classification purposes, although less accurate than those produced by the simplified comparison function. Neither of the two objective functions show clear patterns of dependency on γ_2. It is worth noting that in this case, too, the Bayesian objective function and the simplified objective function both produce faulty estimates in only two of the ninety cases considered, further strenghtening the estimation strategy proposed in section 4.2.1.

4.2.3 Sensitivity to sound speed profile

Figure 4.18 shows eigenray estimation errors for different target depths, target ranges and the different sound speed profiles presented in figure 3.3. The objective function used was the simplified comparison function. The bottom depth was fixed at 300 m. The source depth used was 5 m. The SNR used was 25 dB. Looking at the figure, it is apparent that when applying the simplified comparison function, the accuracy of the target depth estimates will vary according to which sound speed profile is being used. A closer look at the cases in which the method fails reveals that the cause of the erroneous estimates is the same as discussed in section 4.2.1; clustering of arrivals causes the simplified objective function to value wrong parameters higher than the correct parameters. We may observe that the frequency of faulty estimates seems to be dependent on the seasonal variations in the sound speed profile. The distribution of estimation error seems to be comparable for the winter months (January, February and March), the spring months (April, May and June), the summer months (July, August, September) and the autumn months (October, November, December). The largest number of faulty estimates is seen during the summer months, followed by the winter months. Incidentally, the sound speed profiles for these months contain the most extreme sound speed variations, indicating that the simplified comparison function performs better for sound speed profiles with less variation. This is reasonable, since sound speed profiles with more extreme variations allow for the clustering of arrivals in sound speed channels, which is known to put the simplified comparison function at risk of erroneous estimates.

Next, figure 4.19 shows the results obtained by use of the full comparison function, the simplified comparison function and the Bayesian objective

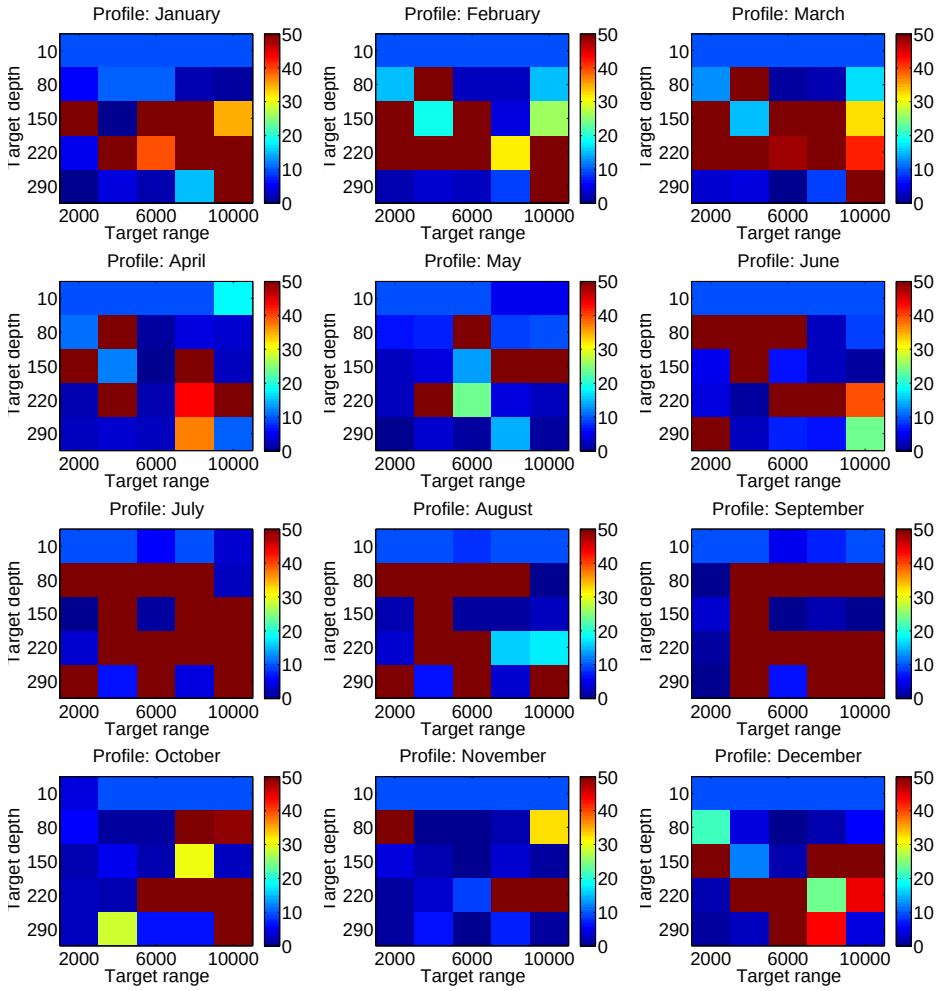

Figure 4.18: Mean error in estimates of target depth as a function of target range and target depth for different sound speed profiles. Bottom depth: 300 m.

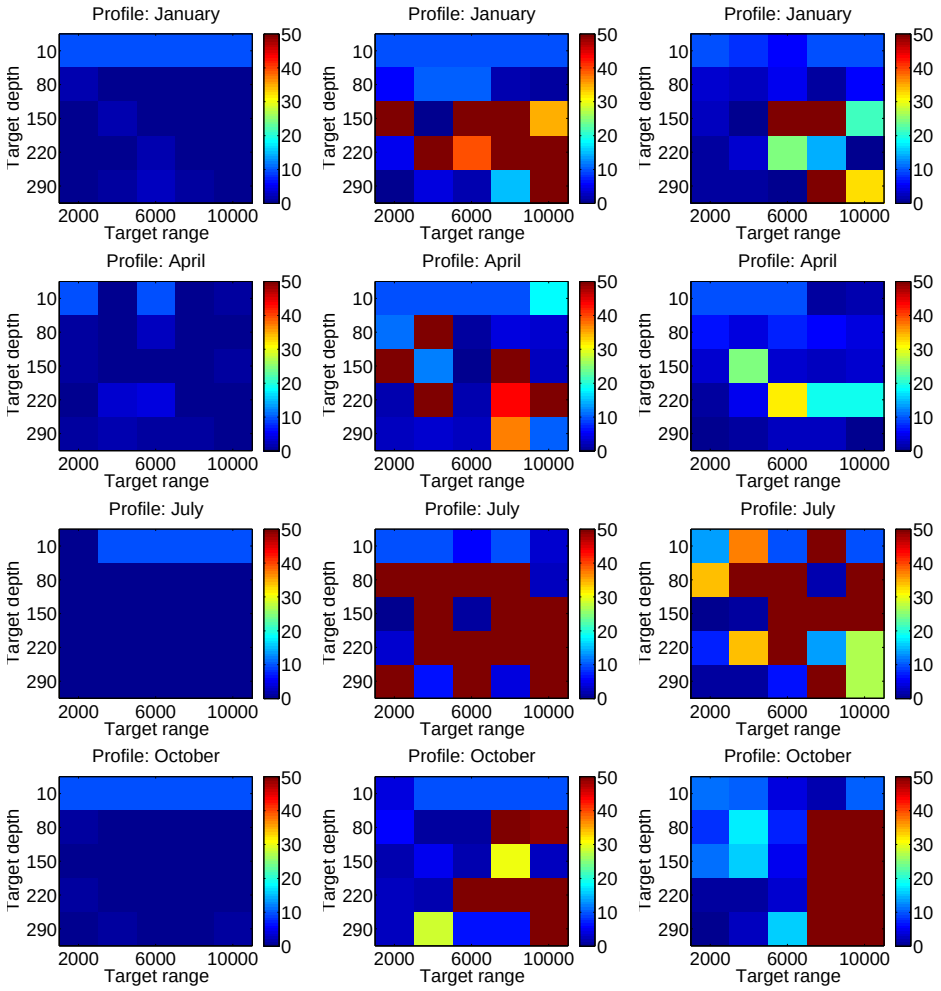

Figure 4.19: Mean error in estimates of target depth as a function of SNR and target range. Bottom depth: 1000 m. Left: Full comparison function. Middle: Simplified comparison function. Right: Bayesian objective function.

function when considering the four sound speed profiles presented in figure 4.20 which are deemed representative of each season. We see that the full comparison function performs flawlessly, while the Bayesian objective function in general performs better than the simplified comparison function although estimates get worse as the target range increases. We also see some seasonal variation in the performance of the Bayesian objective function, with the best estimates occurring for January and April, and worse estimates for July and October. Once again, it would be interesting to see whether the erroneous estimates persist with a correct implementation of the Bayesian objective function and when using more eigenrays for modelling arrivals.

We also see that at least one of the Bayesian objective function and the simplified comparison function produces erroneous estimates in 34 out of the 125 cases. In twelve out of these cases, both of them fail. In five of these twelve cases, the two objective functions produced target depth estimates differing by more than 50 m. Had the strategy proposed in section 4.2.1 been applied here, this means that in just 7 out of 125 cases, the target depth would be estimated incorrectly.

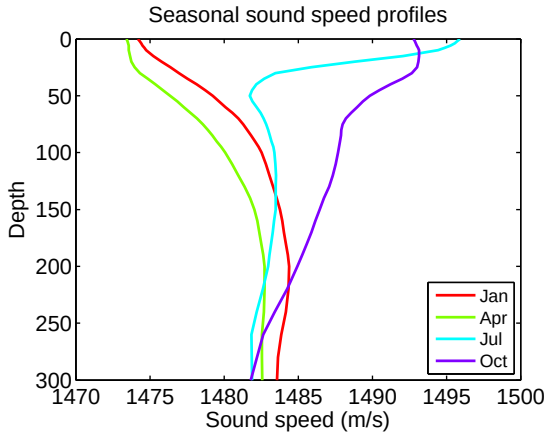

Figure 4.20: The four sound speed profiles deemed representative of their season.

4.3 Execution time

Figure 4.21 shows the mean execution time of the method as a function of target range for several bottom depths when using the simplified comparison function and the Bayesian objective function with two and three eigenrays for modelling. In the left column, we see execution times with 3000 rays traced in the eigenray estimation procedure. In the right column, we see execution times with 5000 rays. Execution times when using the full comparison function are omitted from the plots as they are very large compared

to those of the other objective functions.

The execution times are not surprising; we can see that execution times are close for all objective functions shown. This is because 70-80% of the computational work in all three objective functions is done in the eigenray estimation. We can see that the larger sums involved in evaluating the Bayesian objective function when going from two to three eigenrays yield a constant increase in computational time of ~30 seconds. We can also see that the execution time increases with target range; this is because the underlying numerical algorithm in LYBIN will need to take more steps for longer ranges. We can also see that the execution time increases with bottom depth. This is because of the additional target depths we need to consider in the initialization procedure. The spacing between candidate target depths in the exhaustive search was set to 10 m, meaning that with a bottom depth of 250 m, we would need to consider 26 target depths, while with a bottom depth of 450 m, this number would increase to 46, causing the execution time to grow. Furthermore, we see that when increasing the number traced rays from 3000 to 5000, the slope of the execution time graph increases since each run of the eigenray finding procedure, and thus each evaluation of the objective functions, becomes more costly.

The mean execution time of the full comparison function ranges from 1100 s for $z_b = 250$ m to 1800 s for $z_b = 650$ m, with variations of $\pm$ 100 s depending on target range and the number of rays used, and so the full comparison function is not as sensitive to target range and number of rays used. This is because most of the work done in evaluation the full comparison function comes from the arithmetic operations involved in modelling a full signal for comparison and the subsequent comparison; compared to this, the work done in the eigenray finding procedure is small. This information gives more credence to the strategy proposed in section 4.2.1, since it appears that doing one estimation with the simplified comparison function and one estimation with the Bayesian objective function will still be faster than a single estimation using the full objective function.

From these results, we can also glean some information about what should be done to speed up the method using the different objective functions. Using the simplified comparison function and the Bayesian objective function, most of the work is done in LYBIN during the ray tracing, and so decreasing the execution time of the method here would involve optimization of LYBIN. As we do not know the inner workings of LYBIN, no concrete measures can be presented here in this regard. Reductions in the execution time using the full comparison function could be obtained if the modelling and norm evaluation involved could be sped up. This should be possible using a compiled language such as C++ or Fortran instead of MATLAB. Another way of speeding up the method irrespective of objective function is using a more effective optimization algorithm which requires less function evaluations.

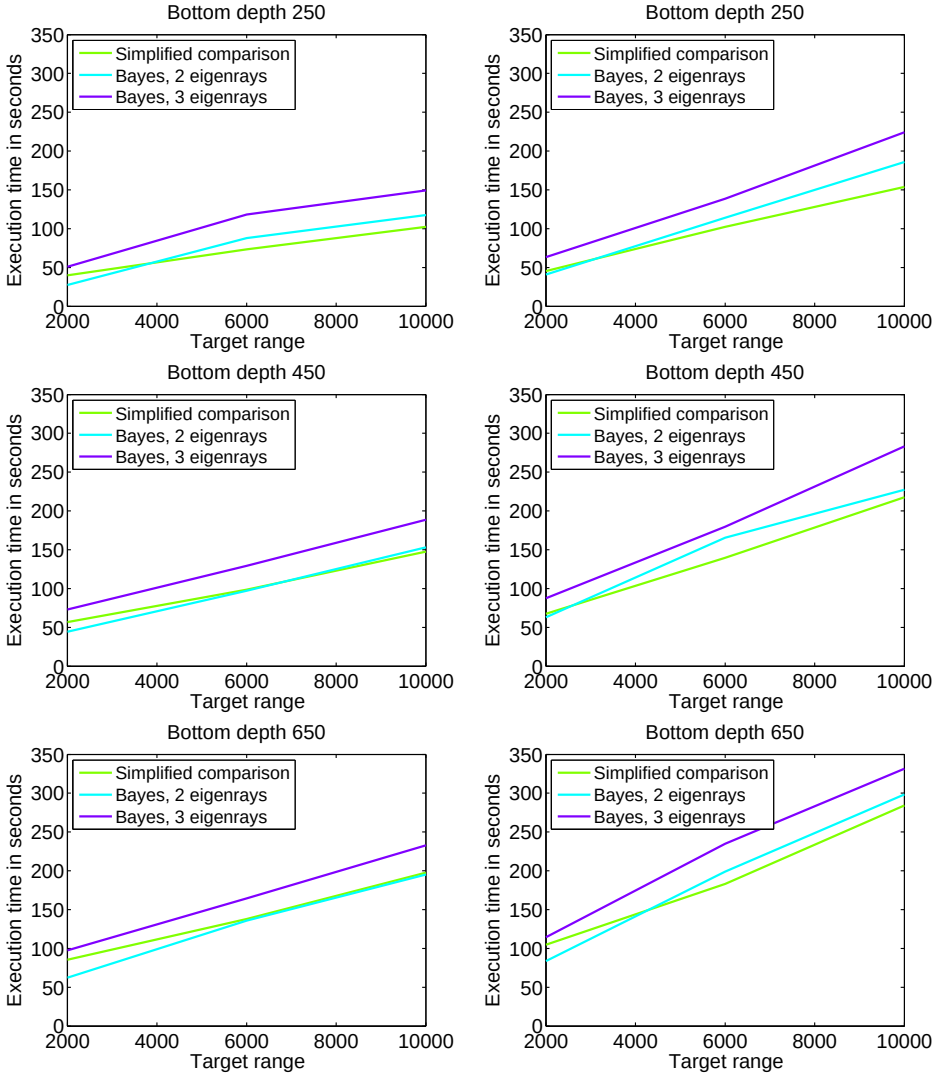

Figure 4.21: Execution times for the simplified comparison function and the Bayesian objective function. Left: Using 3000 rays in eigenray estimation. Right: Using 5000 rays in eigenray estimation.

Chapter 5

Future work and conclusion

5.1 Suggested directions for further work

Based on the preceding results and analyses, we now review some possible topics for future work.

As mentioned in section 4.1, the eigenray finding procedure can be improved by requiring that ray families enclose the target depth at target range. This would reduce the chance of obtaining spurious eigenrays, thus improving the accuracy of the target depth estimation algorithm. Applying this condition would entail editing the source code of LYBIN. Also, it may be possible to obtain initial angles of eigenrays analytically when using linear sound speed profiles, as outlined in section 2.4.1. If so, it may be possible to expand this result to analytical solutions for initial angles of eigenrays with piecewise linear sound speed profiles, eliminating the need for LYBIN when considering flat sea bottoms and speeding up the method significantly. However, further expanding this to encompass general bottom profiles seems improbable, so the use of numerical methods for obtaining eigenrays seems unavoidable. Therefore, improvements to the numerical method for obtaining eigenrays should take precedence over exploring analytical solutions.

As of now, the optimization algorithm being used is based on a brute-force search to obtain initial values for the Nelder-Mead algorithm. This is not necessarily the best solution, especially as the number of optimization parameters, and thereby the size of the parameter search space, grows, increasing the execution time exponentially. It is therefore desirable to use a more sophisticated optimization algorithm where the initial search is done more intelligently or is not needed at all. A possible candidate is adaptive simulated annealing, as mentioned in section 2.9. On the other hand, the initialization procedure is readily parallelizable, and a parallel implementation of the brute-force search may prove to be fast enough provided that it is run on a sufficiently fast computer. Also, to improve the execution time of the method, a C++ implementation should be considered.

It is of interest to test the method further. This would entail redoing the testing of the Bayesian objective function with correct implementation and when applying the correction to σ_t as suggested in section 2.8.2, due to the implementation error in the Bayesian objective function as described in section 3.1. It is also possible to implement the $P(M)$ term in the Bayesian objective function using non-uniform probability distributions for the model parameters, as suggested in section 2.8.2. Another improvement to the Bayesian objective function would be to implement an adaptive method of choosing the number of eigenrays modelled, as proposed in section 4.2.1, perhaps in combination with testing the Bayesian objective function when using more than two eigenrays for modelling arrivals. This may resolve the problem of ambiguity of eigenrays described in section 4.2.1.

It would also be of interest to test the method on real data. This would entail obtaining unclassified recorded acoustic data from a sonar system with good vertical beamforming capabilities, and with detections from a target situated at a known depth. From the preceding results and discussion, it would appear that the strategy suggested in section 4.2.1 could be used for this testing. The proposed strategy consists of first estimating target depth using both the simplified comparison function and the Bayesian objective function, then checking whether the estimates agree. If they do not agree, the full comparison function could be applied to determine the target depth. It may also be of interest to use bottom profiles instead of flat sea beds to investigate sensitivity to varying bathymetric conditions. As a side note, a better way of choosing the size of guard bands for CA-CFAR filtering should be found, as suggested in section 2.6. This is not strictly related to the estimation method, but may prove useful when encountering real signals.

5.2 Conclusion

Estimation of target depth has been carried out on synthesized acoustic data using three different objective functions. The results obtained are acceptable for classification purposes. The most accurate estimates were obtained when using the full objective function. The simplified comparison and Bayesian objective functions generally produce acceptable estimates, and are viable as more computationally efficient alternatives to the full objective function.

More testing needs to be done to ensure the stability and accuracy of the method using real acoustic data. Additionally, the method must be tested once more using the Bayesian objective function with a correct implementation to ensure the validity of the Bayesian objective function. Improvements can be made with regards to eigenray estimation accuracy and the choice of optimization algorithm. A promising strategy for target depth estimation can be formulated using simultaneous estimates by both the simplified comparison function and the Bayesian objective function.

Bibliography

[1] R. J. Urick, *Principles of Underwater Sound*, pp. 1–237,291–328,377–406. Westport, Connecticut: Peninsula Publishing, 1996.

[2] K. T. Hjelmervik, *Sonar false alarm rate suppression using classification methods based on acoustic modelling*. PhD thesis, NTNU, 2011.

[3] B. R. Mahafza, *Radar Systems Analysis and Design Using MATLAB*, pp. 143–165,607–631. Huntsville, Alabama: Chapman & Hall/CRC, 2000.

[4] M. A. Richards, *Fundamentals of Radar Signal Processing*, pp. 335–362. New York: McGraw-Hill, 2013.

[5] A. Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation*, pp. 1–41. Philadelphia: SIAM, 2005.

[6] M. Collins and W. Kuperman, "Environmental focusing and source localization," *Journal of the Acoustic Society of America*, vol. 90, no. 3, 1991.

[7] I.-T. Lu, "Simultaneous characterization of source, array and environment using a ray travel-time inversion approach," *Journal of Computational Acoustics*, vol. 5, no. 2, 1997.

[8] P. Voltz and I.-T. Lu, "A time-domain backpropagating ray technique for source localization," *Journal of the Acoustic Society of America*, vol. 95, no. 2, 1994.

[9] F. B. Jensen, W. Kuperman, M. Porter, and H. Schmidt, *Computational Ocean Acoustics*, pp. 65–230. New York: Springer-Verlag, 2000.

[10] E. Dombestein and T. Jenserud, "Improving underwater surveillance: Lybin sonar performance prediction," in *Maritime Systems and Technologies Conference*, 2010.

[11] J. Nocedal and S. J. Wright, *Numerical Optimization*, pp. 238–240. New York: Springer Verlag, 2006.

[12] H. Bucker, "Use of calculated sound fields and matched-field detection to locate sound sources in shallow water," *Journal of the Acoustic Society of America*, vol. 59, no. 2, 1976.

[13] A. Baggeroer, W. Kuperman, and P. Mikhalevsky, "An overview of matched field methods in ocean acoustics," *IEEE Journal of Ocean Engineering*, vol. 18, no. 4, 1993.

[14] G. Wong and S. Zhu, "Speed of sound in seawater as a function of salinity, temperature and pressure," *Journal of the Acoustic Society of America*, vol. 96, no. 3, 1995.

[15] H. Björnsson and S. Venegas, *A Manual for EOF and SVD Analyses of Climatic Data*. Montreal, Quebec, Canada: McGill University CCGCR Report 97-1, 1997.

[16] J. Dettmer, S. E. Dosso, and C. W. Holland, "Model selection and bayesian inference for high-resolution seabed reflection inversion," *Journal of the Acoustic Society of America*, vol. 125, no. 2, 2009.

[17] P. Gerstoft and C. F. Mecklenbräuker, "Ocean acoustic inversion with estimation of a posteriori probability distributions," *Journal of the Acoustic Society of America*, vol. 104, no. 2, 1998.

[18] S. E. Dosso, P. L. Nielsen, and C. H. Harrison, "Bayesian inversion of reverberation and propagation data for geoacoustic and scattering parameters," *Journal of the Acoustic Society of America*, vol. 125, no. 5, 2009.

[19] R. E. Walpole, R. H. Myers, S. L. Myers, and K. Ye, *Probability & Statistics for Engineers & Students*, pp. 58–73. Upper Saddle River, New Jersey: Pearson, 2007.

[20] A. Farina and E. Hanle, "Position accuracy in netted monostatic and bistatic radar," *IEEE transactions on aerospace and electronic systems*, vol. AES-19, no. 4, 1983.

[21] C. E. Lindsay and N. R. Chapman, "Matched field inversion for geoacoustic model parameters using adaptive simulated annealing," *IEEE Journal of Ocean Engineering*, vol. 18, no. 3, 1993.

[22] E. Dombestein, "Lybincom 6.1 - description of the binary interface," tech. rep., FFI, 2012/01453.

[23] T. Ringholm, "Target depth estimation using hull mounted active sonars," in *Proceedings of the 37th Scandinavian Symposium on Physical Acoustics*, 2014. To appear.