
Room Boundary Estimation

Master Thesis Report
Group 1060

Aalborg University
Acoustics and Audio technology



Electronics and IT
Aalborg University
<http://www.aau.dk>

AALBORG UNIVERSITY
STUDENT REPORT

Title:

Room Boundary Estimation

Theme:

Master Thesis

Project Period:

Spring Semester 2016

Project Group:

1060

Participant(s):

Domínguez Simón, José Ignacio
Suhr Jøns, Christian

Supervisor(s):

Sejer Pedersen, Christian

Copies: 0

Page Numbers: 141

Date of Completion:

June 2, 2016

Abstract:

This thesis was made in order to investigate the possibility of estimating the geometry of a room and absorption of walls based on an acoustic measurement. Due to the fact that sound pressure builds up along walls and other obstacles, investigations were made in terms of how the sound field in a room is defined and how such a sound field could be reconstructed based on measurements. What defines a sound field in a room is the direct sound and the reflections. The reflections were modelled as virtual sources and if these are placed at great accuracy, in terms of distance and direction, a good estimation of the sound field can be made. The simulated measurement used a circular microphone array in order to use array signal processing to estimate direction of arrival of the virtual sources. The stochastic maximum likelihood estimator was used for the DOA estimation. The room can then be geometrically estimated if the power of the sound field is plotted. For the absorption of the walls, the spectrum from the first order virtual sources are compared to the spectrum of the direct sound with promising results.

The content of this report is freely available, but publication (with reference) may only be pursued due to agreement with the author.

Contents

Preface	ix
I Intro	1
1 Introduction	3
1.1 Initial Problem Statement	3
1.2 Naive solution to initial problem statement	4
II Problem analysis	5
2 The problem analysis	7
2.1 Sound in enclosed spaces	7
2.2 Direct sound	9
2.3 Reflected sound	10
2.4 Virtual sources	10
2.5 Problem statement	11
2.6 Main methods of solving the problem statement	12
2.7 Existing solutions	13

2.8	Structure of report	13
III	System design	15
3	System overview	17
4	Acoustic measurement simulation	19
4.1	Desired characteristics	19
4.2	Minimum technical capabilities	19
4.3	Accuracy obtainable	20
4.4	Designed simulation procedure	26
4.5	Designed simulation assumptions	29
4.6	Designed simulation limitations	30
5	Ray-tracing and synthetic IR generation	33
5.1	Basic idea explanation	33
5.2	virtual source formation	33
5.3	Parts of the program	35
5.4	Conducted example simulations	37
5.5	Implementation of air absorption	45
5.6	Implementation of wall absorption	49
6	Virtual sources cloud	51
6.1	Introduction to finding virtual sources	52
6.2	IR usage	53
6.3	Narrowband vs wideband DOA estimation	54

Contents	vii
6.4 DOA estimation	54
6.5 Determining the amount of sources present	73
6.6 Array geometry	79
6.7 Analysis procedure of time windows from the IRs	88
7 Results processing	95
7.1 Geometry estimation procedure	95
7.2 Geometry estimation accuracy	100
7.3 Finding the absorption characteristics of the boundaries	104
7.4 Estimation of sound field	111
IV Closing	115
8 Conclusion	117
9 Further Research	121
Bibliography	123
V Appendix	127
A Implementation of the DOA estimation	129
A.1 The steering matrix	129
A.2 The SML cost function	130
A.3 The genetic algorithm	130
B Implementation of the ray tracing	135

C Implementation of the detection

141

Preface

This report is written by Group 1060, José Ignacio Domínguez Simón and Christian Suhr Jøns, studying their Master Thesis in Acoustics and Audio technology at Aalborg University. The report analyses the ability to determine certain characteristics about a room from an acoustical measurement. This Master Thesis is produced with the supervision of Christian Sejer Pedersen.

There is attached a .zip file that contains the following:

- A copy of the report,
- a folder containing animations related to this study,
- the code to execute the full simulation,
- the code to run the DOA estimator alone and
- code for various demonstrations.

Aalborg University, June 2, 2016

Domínguez Simón, José Ignacio
<jdomin14@student.aau.dk>

Suhr Jøns, Christian
<cjans11@student.aau.dk>

Part I

Intro

1 | Introduction

When new buildings are made or old ones are restored acoustics is often not something that is thought of at first. So in the case of bad acoustic environments some kind of rebuilding of a room has to be done to secure a better acoustic environment. Normally one would have to go in the room and note down the material of all surfaces along with measurements hereof and afterwards feed them into software which then will estimate what for example causes bad acoustics. In this situation it could be time saving to be able to for example identify highly reflective surfaces with the aid of some microphones, a loudspeaker and related software. Another use case is that a surround sound system can benefit from knowing certain characteristics about rooms and take advantage of this to make a better listening experience for the user.

1.1 Initial Problem Statement

Based on this example a initial problem can be stated:

How can wall characteristics of rooms be evaluated in terms of geometry and sound absorption of reflective surfaces?

On behalf of the initial problem statement there will be made a problem analysis which will be based upon some clarifying points derived from the initial problem statement.

These points are:

- Description of sound in a room,
- existing solutions,
- chosen solution and
- elements in this solution.

The first point will address the physical properties of sound in a room and from which of these properties a system will be able to come up with answers to the initial problem. A literature search is made in order to evaluate what existing solutions there might be. Based upon the existing solutions and the knowledge gained from the first point it should be possible to form the basis of a system that can answer the initial problem statement.

1.2 Naive solution to initial problem statement

One way of determining the shape of a room can be done by physically measuring the the room dimensions. This is an accurate way of doing it, but for larger or complicated shape rooms this would be a time consuming method. For determining the absorption of the wall of a room one could go in the room and identify the wall materials. This can be tricky since walls are usually painted to cover up the actual materials they are made of. Also, it can be hard to find acoustical information about specific materials, mostly when they are old.

Another way of estimating the room could be as done in [1]. Here the author uses a Kinect camera to render a precise 3D model of a room. Based on this model a room impulse response can be estimated that can be used for auralization. Based on images wall materials can be estimated to a certain precision but often raw materials are covered by paint, plaster or other kinds of decorations making classification of materials hard.

Therefore an acoustic method of answering the initial problem statement will be looked into in the following parts.

Part II

Problem analysis

2 | The problem analysis

This part will hold explanations of the details about estimating the room that are constant regardless of the chosen solution. That is the sound field.

Introduction to the problem analysis

Since the problem is now limited to estimating room characteristics based on the sound inside the room there will be made a analysis of what the sound field might look like and what elements in the sound field that will hold information about the room.

2.1 Sound in enclosed spaces

This section will describe how a sound field propagates from a point source in free field conditions and in a closed environment. Producing these sound fields might reveal certain characteristics of the sound field that is closely connected to the room.

Sound propagating from an omnidirectional point source in free field will propagate spherically and concentric around the source point ([2], page 12). Attenuation of the sound waves will be inversely proportional to the distance from the sound source and a phase angle that aligns with velocity as the distance to sound source increases.

When the same sound source is placed within an enclosure a fraction of the sound waves' energy will be reflected of the boundaries and therefore create a much more complicated sound field ([3], page 55). Therefore the sound field is no longer a spherical wave propagating through air.

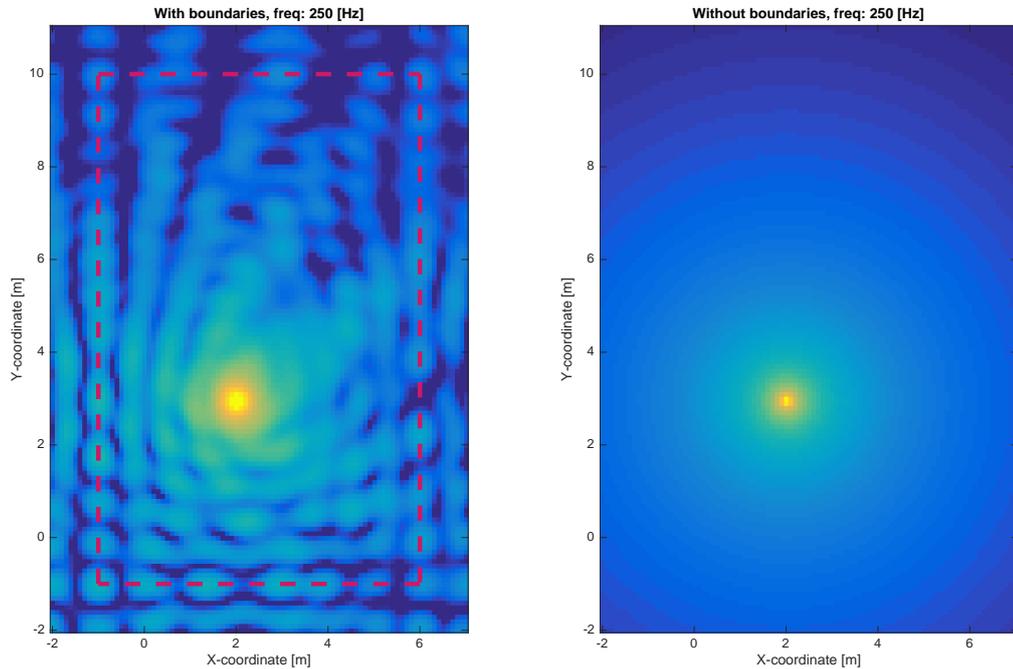


Figure 2.1: Comparison of power map for a point source radiating on an anechoic space and within an enclosure. The reflection from the enclosure boundaries produces distinctive wave patterns.

The sound field in enclosed environments can therefore be considered as combined contributions of the point source radiation plus all new waves originated on the boundaries. The result is, in general, an irregular distribution of pressures across the enclosed space.

As a consequence of this the sound field in any enclosure is defined¹, univocally, by the only two elements that are involved:

- The source characteristics (mainly in terms of directivity, frequency and level).
- The enclosure characteristics, the most important being:
 - Shape of boundaries and
 - absorption of boundaries (frequency dependent).

From a pure physical mode, the enclosure can be understood as a series of infinitesimal particles forming a closed surface. Those, when excited by the source, vibrate and produce new waves that are in turn propagated inside the enclosure ([2], page 36). These new waves

¹The medium is considered to be homogenous and isotropic, thus the sound propagation speed does not change in all the enclosure (homogeneous) and does not depend on the displacement direction (isotropic). If this does not hold through the enclosed space, the medium uniformity will be a third parameter.

act in the same manner as the one produced by the original source, and the whole process continues over time until the acoustic energy is dissipated due to the viscosity of the media (which means that the propagation occurs with a thermal consumption of energy).

The relation that exists between the vibration of enclosure boundaries and the sound field produced inside, when the enclosure is free of sources, is given by the Kirchhoff-Helmholtz integral. It states that the sound pressure is fully determined within the space contained when it is known across all points in its surface.

$$P(\vec{r}) = \frac{1}{4\pi} \int_S \left[P \frac{\partial}{\partial \hat{n}} \left(\frac{e^{jkd}}{d} \right) - \frac{e^{jkd}}{d} \left(\frac{P \partial}{\partial \hat{n}} \right) \right] \partial S \quad (2.1)$$

Kirchhoff-Helmholtz integral (2.1) for a monochromatic wave. It defines the sound pressure on a point r within a closed surface S , with d being the distance from the point to the surface element and k the wave number. The differentiation is done respect to the surface normal, that is considered to point inside the enclosed volume.

In the case of study, the enclosure is not free of sources, but it is equivalent to say that, the excitation introduced by the original source is instead produced by a combination of all boundaries vibrating in such manner that the same sound field exists. This principle is used in the technique of wave-field-synthesis [4], and it leads to a new model where the original source is removed and the boundaries are now active elements able to produce waves on its own. Since the sound fields in both situations are the same, they are said to be acoustically equivalent scenarios.

It is therefore shown that there exists a unique and well defined relationship between the enclosure characteristics and the sound field present inside. It can be concluded that, if the sound field is known, information about the boundaries can be obtained, at least indirectly.

The sound field present in a room could be seen as composed of two parts: the direct sound and the reflected sound.

2.2 Direct sound

The direct sound is, as the name states, the sound that is not reflected of any surface or boundary. So in order for sound to be direct sound there must be a line of sight between the transmitter and receiver. Typically the direct sound will be more prominent compared to reflections up until a certain time. In anechoic spaces there will only be the direct sound.

2.3 Reflected sound

Reflections occur when sound is reflected off some surface. Any sound coming from a direction where there is no source placed can be seen as a reflection. These are attenuated due to absorption of the reflective surfaces. This absorption depends on the material, angle of incidence and frequency of the signal ([2], page 106).

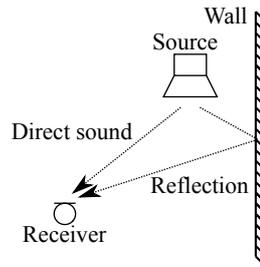


Figure 2.2: The principle of reflections are shown in this figure.

In Figure 2.2 it is seen that the reflection is visualised as a specular reflected ray (bounces off the wall with the same angle respect to the surface normal) ([2], page 11). This assumes that sound waves behave like rays, which only holds for high enough frequencies. It is more accurate to consider diffuse reflections (the reflection occurs on an area, not on a single point and the outcome is composed of a range of angles and not a single one). The validity of these assumptions will be shown further.

2.4 Virtual sources

A sound field in a room consists of the direct sound and the reflections from the surfaces. Reflections can be considered as displaced and time delayed versions of the source if the reflecting surface is ideal and no attenuation happens due to travel distance. Each of these displaced and time delayed versions of the source are called virtual sources. For non ideal and more real life like situations the reflective surfaces will attenuate the reflections depending on the frequency and angle of incidence ([2], page 105).

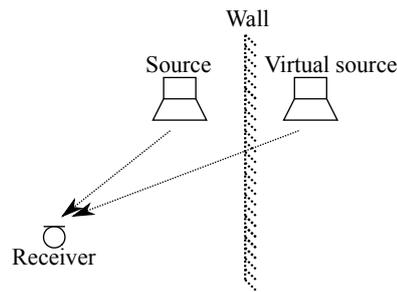


Figure 2.3: This figure shows the idea of virtual sources.

The sound field within a room can therefore be reconstructed if all virtual sources are known ([2], page 136 and 138). It is impractical to consider all of them, but even in the case of a limited amount of virtual sources a good estimate of the sound field can still be made [5].

2.5 Problem statement

Based on the problem analysis a problem statement is made.

How can wall characteristics of rooms be evaluated based on acoustic measurements in said room?

So on the basis of the problem statement it is possible to visualise the outlines of the system to be developed. This can be illustrated in Figure 2.4:

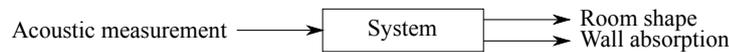


Figure 2.4: This figure show the very basic of any system that will solve the problem statement.

It is seen that a system that uses an acoustic measurement in order to estimate the room shape and wall absorption is wanted. So in other words the system must analyse the acoustic measurement, since it holds this information, and be able to extract this information based here on.

The acoustic measurement is expected to be a series of samples of the sound field at given positions in the room.

On behalf of the problem statement a literature search is made to discover what possible ways there already exist. Note that these solutions might not answer the problem statement to full extend but possibly only a part of it.

2.6 Main methods of solving the problem statement

Before investigating the already existing solutions the problem will be showed from a practical angle that is main methods of solving it. Here the two methods that are thought of is *direct* and *indirect* ways of finding the characteristics of a room based on acoustic measurements. As explained in Chapter 2 the sound field inside a room holds the acoustic information and spatial information about the room which is wanted. An acoustic measurement which can estimate the sound field is therefore wanted since it is practically impossible to measure the complete sound field within a room.

The following sections will give some information on how the needed characteristics from the sound field can be estimated given some acoustical measurements.

Direct method of sound field estimation

The direct way of obtaining information about the sound field in a room would be to record the sound field using a large grid of microphones that should cover the whole room. The pros of doing this is that a high accuracy can be achieved if enough microphones are used. The cons of doing it this way is that a high number of microphones will require a large amount of acquisition hardware and software which in many cases are expensive and not very handy. And most likely the most expensive part is the time it will consume.

Indirect methods of sound field acquisition

Indirect methods are thought of as methods that is not directly measuring the sound field.

Sound field reconstruction based on virtual sources

As explained in section 2.4, if a fair amount of the virtual sources can be estimated in terms of position and spectrum then the sound field within the room can be estimated to a fair degree. A fair degree means that surfaces, and characteristics hereof, inside the room can be estimated. This will only require an array of microphones placed in the room during recording of the acoustic measurements. Then it should be possible to extract the wanted information about the room by processing the obtained data.

2.7 Existing solutions

In [5] the authors are estimating virtual sources based on a room impulse responses so they can convolve those virtual sources with anechoic sound later on to auralize the room. This is somewhat similar to what this project tries to, but they are not determining reflective surfaces or absorption.

In [6] the author estimates room size and shape based on single directional microphone room impulse response measurements that is made with a directional speaker. The speaker and the microphone are facing each other. For every 20° in 360° an impulse response is made and based on this the room is estimated. This is a manual way of estimating a room where a lot of measurements are to be repeated with high accuracy for a good estimation.

A thesis was made at Delft University where a student tries to estimate the room shape based on a room impulse response [7]. It is done by capturing room impulse responses with a microphone array and then determining the virtual sources from it. It then uses the virtual sources, along with the real source, to place the walls of the room. He is limiting the thesis to only consider rectangular shoe box shaped rooms and he uses only the first order reflections ([7], page 17) which will limit this method to simple shapes only.

2.8 Structure of report

In this section there will be an explanation to the general structure of the report and what to expect from each chapter.

As for the problem solving parts of this report those chapters will start with a block diagram that shows the main modules of the system. The idea is that these modules will be explained each in their individual section.

These sections will contain:

- An introduction to the module,
- possible ways of implementing the module,
- a theoretical part on the chosen way,
- a part on how it is simulated or implemented and
- validation of implemented methods.

This should bring a simple and easy to follow structure throughout the report.

Part III

System design

3 | System overview

The intention of this chapter is to provide the reader with an overview of the modules that form the system in this project. In order to do that a figure is made and shown in Figure 3.1.

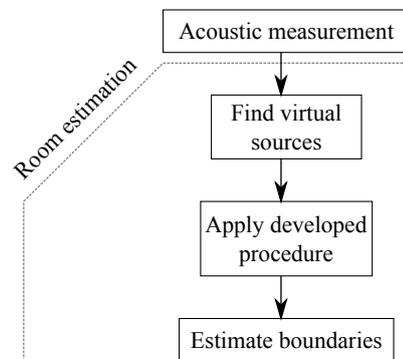


Figure 3.1: This figure shows what modules the simulation consists of.

It is seen that the system will consist of four major modules. These modules will be explained throughout the part in each their chapter. They will follow the structure dictated in Section 2.8. Since the system will be simulated at first, the chapters describing the modules will focus on the simulation of the system.

4 | Acoustic measurement simulation

A system that given an enclosure geometry and its materials' characteristics (in terms of acoustic energy absorption) can produce an estimation of the sound field in all points contained. For simplicity of implementation, it will only be realized in 2D, but the limitations and assumptions that apply are the same for a 3D case.

4.1 Desired characteristics

The simulation system must produce accurate enough data that can be used to:

- Show how the sound field changes when a source is placed on an enclosure as opposed to on free space (the effect of reflections).
- Test the effect of different geometries and materials, and combinations of both.
- Help design algorithms that try to estimate the enclosure geometry and materials' characteristics from a known sound field.

4.2 Minimum technical capabilities

Minimum requirements that must be fulfilled, such that the main sound field characteristics are properly described, are:

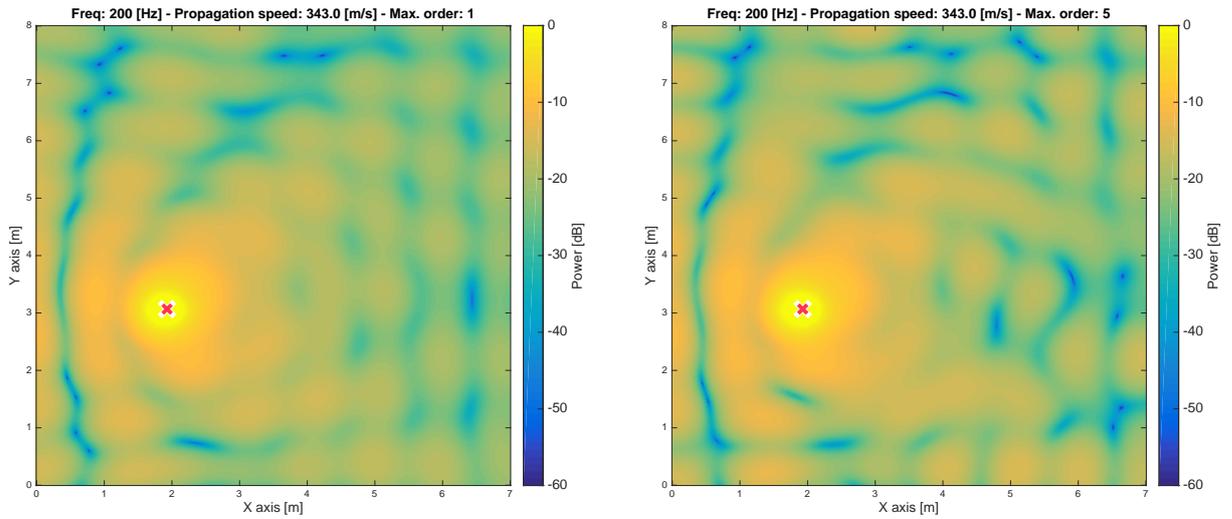
- The sound field estimation, provided as a set of discrete samples, must fulfill the Nyquist criteria to avoid aliasing.
- Peaks and valleys of the interference patterns caused by the reflections must be distinguishable (for example, if the output data is presented as an image this means that the color contrast needs to be properly scaled).

4.3 Accuracy obtainable

The sound field estimation accuracy is limited by two main factors:

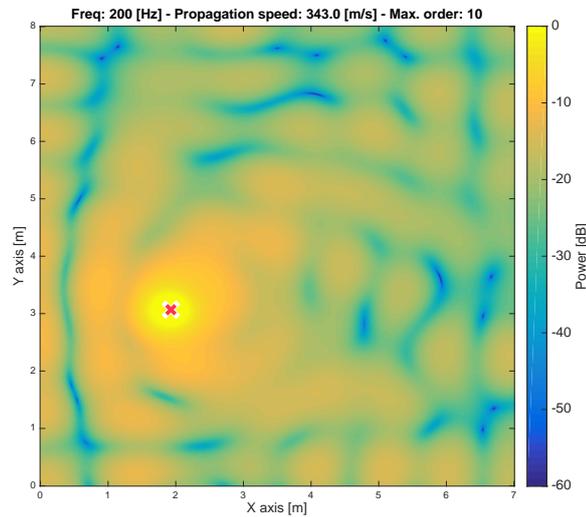
- The amount of virtual sources used
- The precision available on the virtual sources' characteristics:
 - Associated signal spectrum
 - Position

Thus, an ideal case would contemplate an infinite amount of virtual sources and would have a perfect description of the spectrum of the perturbation produced by each of them.



(a) Power map with a maximum reflection order of 1.

(b) Power map with a maximum reflection order of 5.



(c) Power map with a maximum reflection order of 10.

Figure 4.1: Comparison of power maps obtained when different maximum order virtual sources are considered.

Although the power map obtained resembles better the reality when higher order sources are included, the improvement gets more difficult as the order increases. That is, the accuracy of the power map does not increase linearly with the maximum virtual source order.

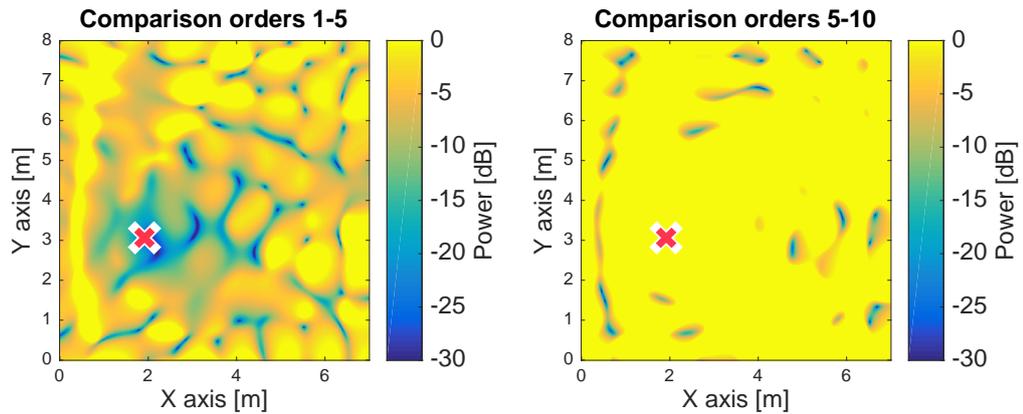


Figure 4.2: Difference in power maps for two consecutive increases of maximum order of 5.

As Figure 4.2 shows, the change in the power map is noticeable from order one to five, but a further increase of the same size (five to ten) does not reveal much changes. Therefore, there is no need in most cases to run simulations up to very high orders, since it does not add a significant amount of extra information.

The first factor is not attainable on a simulation but, as shown on Figure 4.1c, for practical cases there is no need for using an extremely large amount of virtual sources to achieve a good representation of the sound field.

The second factor, the spectrum of each virtual source, is mostly related to the frequency dependent behavior of the materials forming the enclosure and, to a lesser extent, to the air absorption phenomena. A poor precision on this data will reflect negatively on the estimated characteristics of the enclosure's materials.

Lastly, the correctness on positioning virtual sources defines the degree attained in the interference patterns, as the error in the position will produce a misalignment of the individual fields. The combined effect is an offset result, since the obtained interferences are not precise in space. See Figure 4.3.

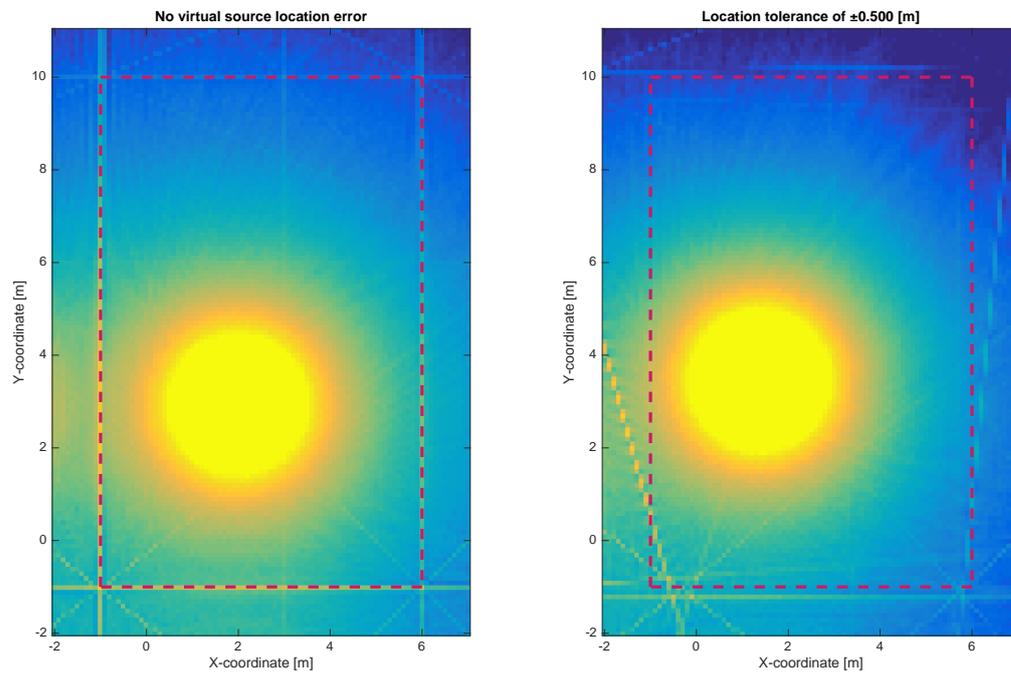


Figure 4.3: Power map averaged over frequency. Very large positioning error.

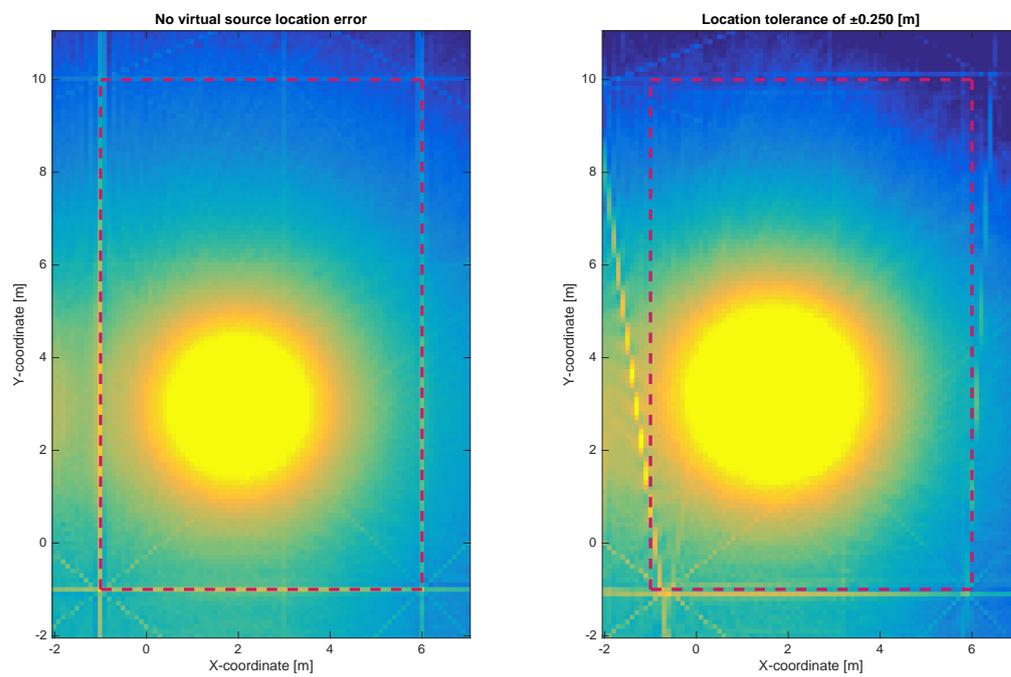


Figure 4.4: Power map averaged over frequency. Large positioning error.

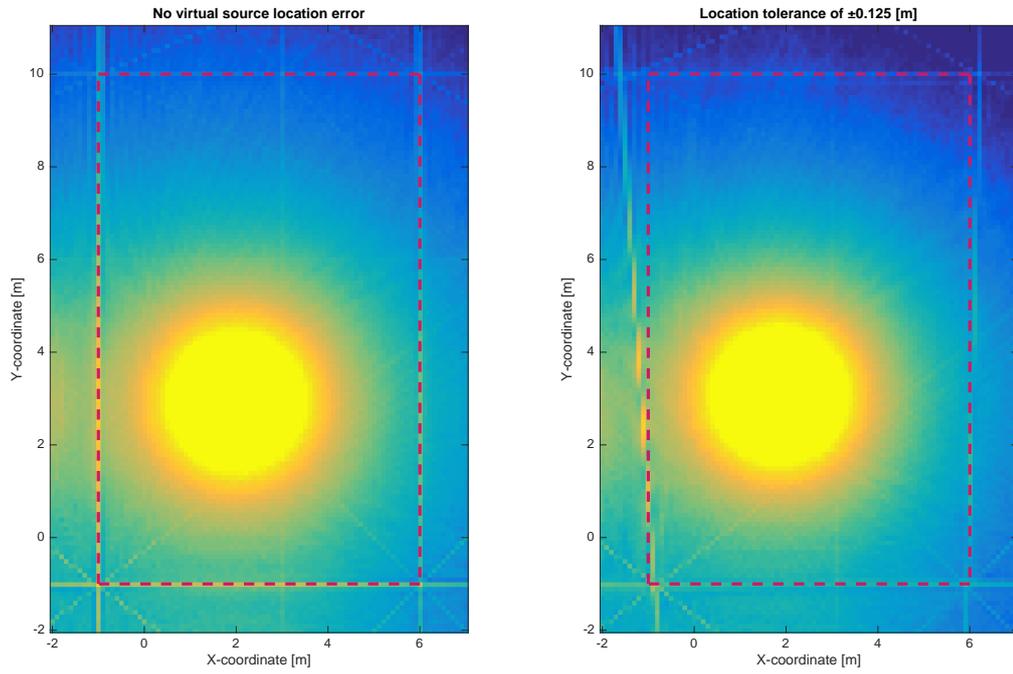


Figure 4.5: Power map averaged over frequency. Small positioning error.

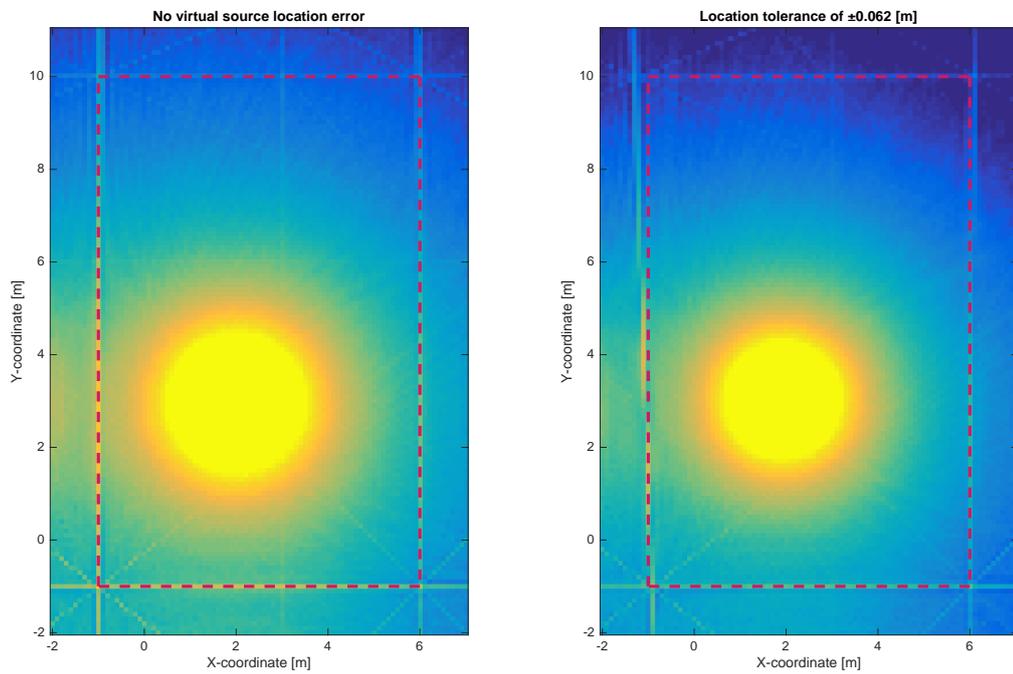


Figure 4.6: Power map averaged over frequency. Very small positioning error.

4.3.1 Resulting necessary virtual source positioning resolution

From previous plots, it is seen that a tolerance in the virtual source positioning of approximately ± 0.125 m should render results that are acceptably close to the case of having the perfect data. A smaller error (± 0.062 [m]) would give almost perfect results, but it may require a too high degree of precision. On the other hand, if the error approaches ± 0.251 m, the results will be considerably off. For this reason, the intermediate value of ± 0.125 m is selected. Based on this, the precision on placing the virtual sources can be separated in two values, when expressed in spherical coordinates: radius error and angular error.

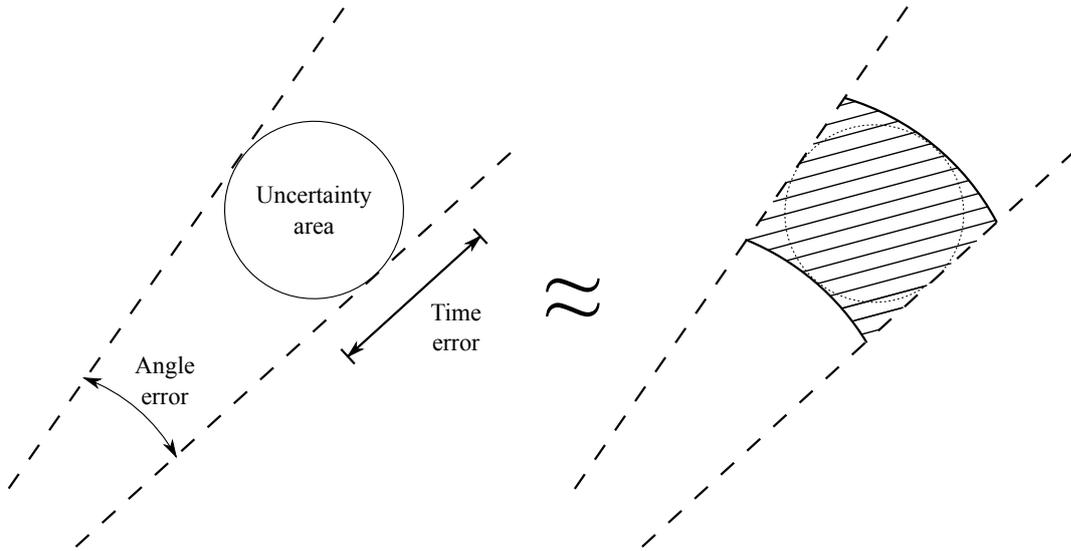


Figure 4.7: Approximation of uncertainty area as a ring segment with associated angular and radial errors.

As seen in the previous figure, the uncertainty area is approximated by the ring segment. The error angle is then defined as:

$$\theta_E = 2 \cdot \text{asin} \left(\frac{2 \cdot 0.125}{2 \cdot d} \right) \quad [\text{degrees}] \quad (4.1)$$

Being d the central radius of the ring. Thus, for example, for a virtual source placed at a distance of 20 m, the angular error could be as big as:

$$\theta_E = 2 \cdot \text{asin} \left(\frac{2 \cdot 0.125}{2 \cdot 20} \right) = 0.72 \quad [\text{degrees}] \quad (4.2)$$

Subsequently, the ring width is defined as:

$$r_E = 0.125 \cdot 2 \quad [\text{m}] \quad (4.3)$$

The accepted tolerance in positioning is now declared as an angular and radial errors.

4.4 Designed simulation procedure

A simulation system has been designed that tries to fulfill the requirements and capabilities listed previously. It comprises the following steps:

4.4.1 Define the source characteristics

In terms of spectrum and directivity. For simplicity, in all conducted simulations, the source is considered isotropic, that is, it radiates equally in all directions. This is a reasonable assumption for most sources in low frequency. For field testing, it is recommended that an omnidirectional speaker is used, such as [8].

4.4.2 Define the enclosure geometry

Real surfaces of enclosures may have a very complicated shape (for example, the rugosity of a plaster wall), but they can be approximated by a flat surface with the average frequency-dependent absorption coefficient. In the conducted simulations, being the space considered a 2D plane, each boundary is represented by a line (see Figure 4.9). Therefore the geometry of the enclosure will be composed of a series of lines that conform a closed polygon.

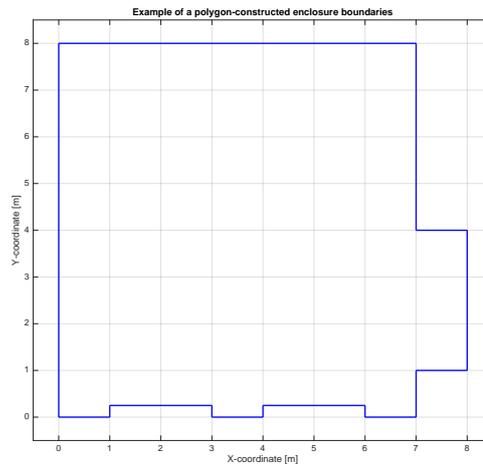


Figure 4.8: Example of room geometry (top view) constructed with two-point lines.

4.4.3 Perform computer based acoustic model analysis

The geometric model previously generated, needs to be analysed using any of the available techniques to produce some data that defines the enclosure and that can be later used for simulating the behaviour of sources and receivers within the room. This will allow us to recreate simulations without having to actually place a source and receivers within such room. Hence, it is required that the method used for obtaining such data about the enclosure can be implemented without too demanding computational needs, since it will only be a necessary tool for the development of this study. Furthermore, it must be accurate enough such that the results can be extrapolated, within an acceptable margin, to those obtainable by real measurements. It is not needed that the simulated data perfectly fits reality, since the objective of this study is to demonstrate the working principle of a method to estimate the geometry and absorption characteristics of the enclosure.

Any enclosure with a radiating source inside can be accurately defined by a set of virtual sources that represent the direct and reflected signals by all boundaries up to a certain order. This equivalent set of virtual sources (that produce the same sound field of a single source in the enclosure) can be estimated using several methods:

Image sources method

It creates virtual sources that are placed as a “mirrored” copy of the original respect to each boundary. It is easy to implement for shoebox geometries, even by hand, but considerably more complex when the geometries are irregular. Its main disadvantage is that it produces pure specular reflections, that is, the diffusion phenomena that naturally occurs is not simu-

lated ([3], page 237). This introduces a limitation in how similar the results can be compared to the reality.

Ray tracing method

It “launches” a series of rays in all possible directions, and follows them as they bounce on the boundaries until some of them reach a given receiver position. The algorithm can be adjusted to produce results that approximate the effects of diffuse reflections ([3], page 235), for example by having a small area that is considered the receiver, instead of a single point.

Other methods

There exist methods to estimate the sound field based on solving the wave equation, such as the finite element method [9], the boundary element method [10], and the finite-difference in time-domain method [11]. Those will generate the correct pressure values, but their use is limited to low frequency because of their high computational requirements ([5], page 26). Another method exists [12] that can produce a higher degree of precision for simulating diffusion, based on a post-processing of a ray-tracing model that considers the roughness of the boundaries. As the authors state, it is only valid for low frequency and therefore is dismissed as insufficient since this project aims to provide information on the materials, typically of interest within the mid-range frequencies. These methods can be used to provide an extended application range in low frequency to the information that image sources and ray tracing can provide. For the purposes of demonstrating the working principle of this study, it is not considered necessary but it may be of interest for a possible future application.

Method of choice

For ease of implementation, the ray tracing method is chosen to generate the data needed for simulations. It can provide accurate results in the mid to high range of frequencies, where the absorption of materials is usually most important, and it can simulate the diffusion phenomena to a certain degree by considering that the receiver is not a point but a small area, so it provides better results than the basic image source method. Lastly, its computational requirements do not scale as much for irregular geometries, as in the image source method.

The main limitations of this method comes from the assumption of sound behaving like rays regardless of the frequency. Hence, it is expected that results in low frequency will not be accurate. The analysis parameters must be taken into consideration as well; such as: angular spacing between consecutive rays, area of receiver, angular dependency of absorption and maximum order of reflections. This and other characteristics will be analysed further.

4.4.4 Application of Green's function

Once the set of virtual sources has been obtained by either method, using the Green's function [13], any required information of the sound field within the enclosure can be determined. For example, to compute the IR from the source to a certain point or to estimate the map of the sound pressure created by the enclosed source. The joint sound field existent on a point will be the linear combination of all individual contributions [14].

4.5 Designed simulation assumptions

The described procedure involves a series of assumptions:

4.5.1 Sound waves behave like rays regardless of the frequency

This holds valid for frequencies above $4f_s$ (the Schroeder frequency), and is partially correct for those in the range of f_s to $4f_s$. The Schroeder frequency defines the beginning of the region where the waves start to behave like rays, and where the enclosure modes are no longer the main responsible for the sound field. It is an statistical value, based on the enclosure dimensions and average absorption.

$$f_s = 2000 \sqrt{\frac{T}{V}} \quad (4.4)$$

Being T the reverberation time in seconds [s] and V the volume in cubic meters [m^3].

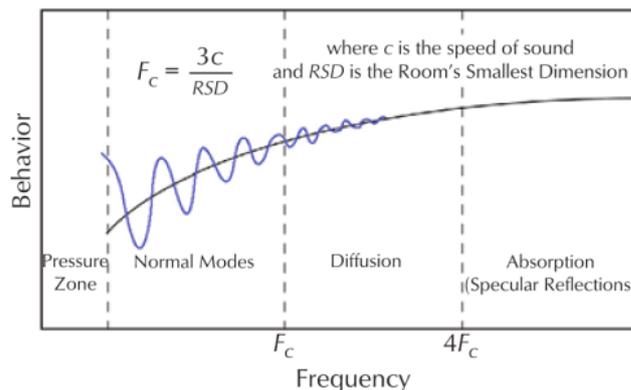


Figure 4.9: Critical frequency chart, showing a simplified equation. Plot from [15].

This imposes a lower limitation on the frequencies that can be included in the simulation. For practical cases, the octave band centered at 125 Hz is the lowest that can be used ([15], chapter 20.8). On the other hand, the analysis in very high frequency does not add much information since the absorption is usually high, so almost no energy is left after a few reflections.

The Schroeder frequency is still a subject of debate, and as such must be taken as general idea of where the simulation should provide reasonable results. Some recent investigations suggest that the modal behavior does not disappear above the Schroeder frequency, and that the transition zone may be measurable [16].

4.5.2 Boundaries are rigid but can absorb energy

The boundaries of the enclosure being rigid means that they cannot vibrate when excited by an incident sound wave ([17], page 213). This introduces the error of considering that materials' impedance are purely resistive (and never reactive), that is, the boundary does not introduce a phase offset in the reflected wave. This does not hold well in some cases, such as for soft materials and resonating boundaries (perforated panels).

4.5.3 There is no interaction between sources

The superposition principle holds on the assumption that the perturbation introduced by a source is not altered by the presence of others. In the case of study this means that the presence of some boundaries does not affect others. This is an approximation since boundaries are never perfectly rigid and can vibrate, in turn transmitting that vibration to the adjacent boundaries that may radiate consequently. That radiation is therefore not existent in this simulation procedure. It is considered to be a much smaller contribution to the total sound field compared to that produced by the reflecting waves. It can be important in the case of highly absorbent enclosures where the reflections do not pose such a greater role in the overall sound field.

4.6 Designed simulation limitations

In addition to the previously explained assumptions and capabilities, some other limitations exist.

4.6.1 Diffraction

Travelling sound waves that encounter obstacles of dimensions that are comparable to the wavelength have the ability to “skirt” around it and produce a new wavefront. This behaviour is not considered in this simulation, since the ray model is assumed to be valid through all frequency range. It is left to the user to properly set the frequency range of the analysis such that the diffusion phenomena occurrence is minimised for a specific enclosure geometry.

4.6.2 Angular dependency of materials’ absorption

This is not intrinsic to the method, but a design choice to simplify the simulation procedure. The materials are considered to have a constant absorption regardless of the incidence angle, being only a function of frequency. This does not hold well for incident angles close to being parallel to the surface plane, since most materials tend to be more reflective in such case ([3], page 105). It takes special importance for the coincidence effect, since at a certain angle, a boundary might provide almost total transmission, thus virtually eliminating that reflection. This comes from the fact that boundaries are not completely rigid and can hold waves that travel through its surface ([3], page 158).

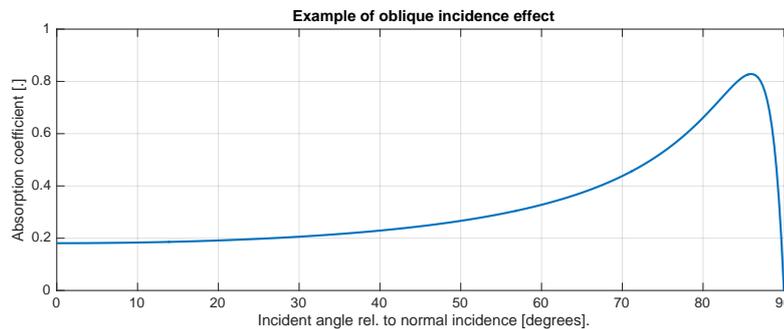


Figure 4.10: Oblique incidence effect example. The material shows an almost constant absorption coefficient within a wide margin of the normal angle, but becomes very absorbent at the coincidence frequency (most energy is transmitted, very little reflected).

5 | Ray-tracing and synthetic IR generation

In this section it is described how a ray-tracing system was implemented as a computer program (See Appendix B), and how it is used to generate synthetic IRs that allow to simulate point sources radiating inside an arbitrary enclosure.

5.1 Basic idea explanation

The goal of this program is to obtain a simulation of the impulse response corresponding to the input-output transfer function of a system formed by the elements: loudspeaker-room-microphone. This can be accomplished in several ways; the one chosen here corresponds to assuming that the sound propagates as rays, and that the objects (boundaries) produce pure specular reflections ([18], page 261).

These rays will bounce on the room's boundaries and may eventually reach the receiving microphone. In such case, they are said to correspond to virtual sources (in so far as they are not real sources) that are placed at a distance corresponding to the total travelled path, and located in the direction that they approach the microphone from.

Under these assumptions, the simulated IR is obtained by adding the contributions of all virtual sources, with their distance and amplitude, as if the whole environment was anechoic but containing many sources (instead of a single one).

5.2 virtual source formation

The obtained virtual sources produce a signal that is a copy of the original but delayed on time and filtered in frequency.

The reasons for the change in the spectrum of the signal on each virtual source come from:

- Spherical propagation law:

As the wave propagates, the energy is distributed on a larger area, therefore the intensity (watts per unit of area) gets lower. The reduction of the sound pressure amplitude is then inversely proportional to the distance ([19], page 160):

$$A \sim \frac{1}{r} \quad (5.1)$$

- Absorption on reflective boundaries:

Each time a ray bounces on a boundary, only a part of the energy is reflected. This effect is frequency dependent ([20], page 133). Therefore, depending on which boundaries the ray bounces on, the final result will vary, as it is formed from all partial contributions (all individual frequency dependent-reflections).

Different materials are commonly used to finish a room, such as painted plaster walls, carpets and wood. The presence of furniture or objects such as doors and windows also introduce an important contribution. The behaviour of some of these has been taken into account in the simulations conducted. The following plot offers a general idea of the absorption of some of them as a function of frequency.

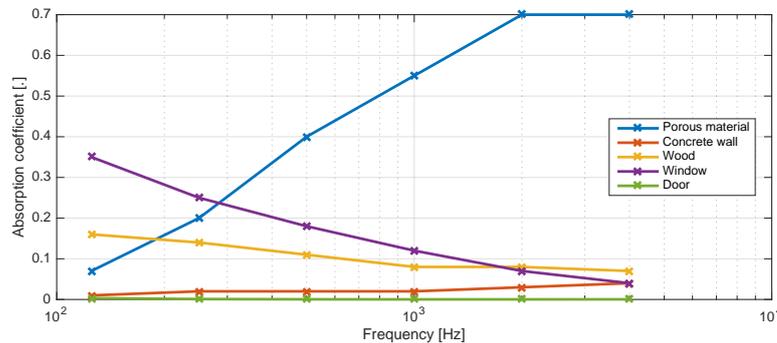


Figure 5.1: Frequency dependent absorption of some typical materials found on most rooms. Data extracted from [18], table A.2. Corresponds to diffuse incidence (average over all incident angles).

- Air absorption phenomena:

It has a considerably smaller contribution to the final result than the previous two effects, but it is also included in the program to allow for better results in big low-absorbent spaces. It mainly results in a loss of high frequency (specially above 4 kHz) [21]. Its magnitude depends on the traveled distance, and for a typical room such as the one studied here, its effect is mostly shadowed by the absorption of the boundaries. The implementation of this effect is treated later.

5.3 Parts of the program

- Knowing the geometry of the space (placement of boundaries and their frequency-dependent absorption), and the placement of source and receiver, a series of rays are “launched” from the source.
- They are launched in equally spaced angle intervals. A complete simulation system should work on 4π steradians. The current developed software considers only a plane, instead of a volume, therefore they cover 2π radians.
- Following the Snell’s law (impinging ray’s angle is equal to reflected ray’s angle respect to the surface’s normal direction), they bounce if a boundary is found. The absorption of the boundary is then applied to the ray.
- If a ray reaches the receiver before a maximum amount of reflections has been reached, it is saved as a “contributing ray”. The information that characterise each ray are the total traveled distance and the cumulative frequency response. This is handled by:
 - Producing a sinc function with a group delay corresponding to the total traveled distance.
 - Filtering the sinc function in frequency every time a boundary is reached.
 - Applying the correction for the air absorption on the total traveled distance.
- After all possible rays have been launched, a reflectogram is constructed by adding all contributing rays. This will represent the effect of the room alone. If the contribution of the frequency response of loudspeaker and microphone is wanted, the reflectogram can be convolved with the impulse responses of both transducers. As an example of this, figure 5.2 shows the comparison between the pure obtained reflectogram and its counterpart after being convolved with the impulse response of a typical studio monitor Genelec 1031A. Contrarily, the pure reflectogram can be considered as the effect of having ideal an loudspeaker and microphone.

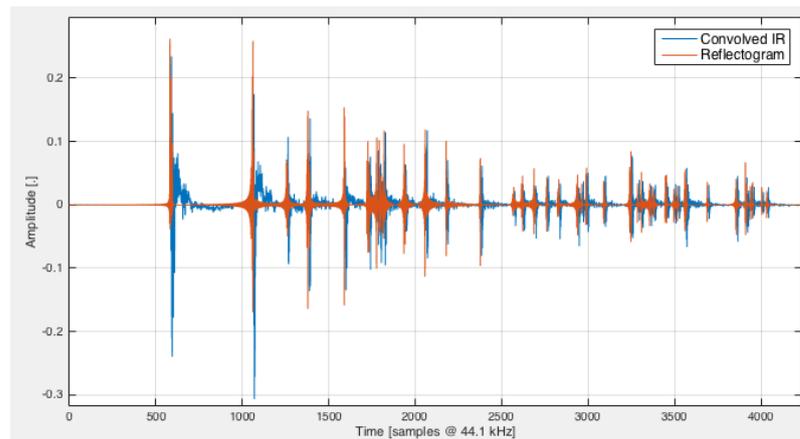


Figure 5.2: Comparison of obtained synthetic impulse response and its corresponding reflectogram.

Note:

For the last step it is assumed that the loudspeaker response is homogeneous across all radiating angles. For a better approximation, each ray should be processed with a different frequency response of the loudspeaker depending of the angle it was launched on. The main defect of the method used in this software is that it will include a higher amount of energy radiated in high frequency for all directions that are not the frontal.

5.4 Conducted example simulations

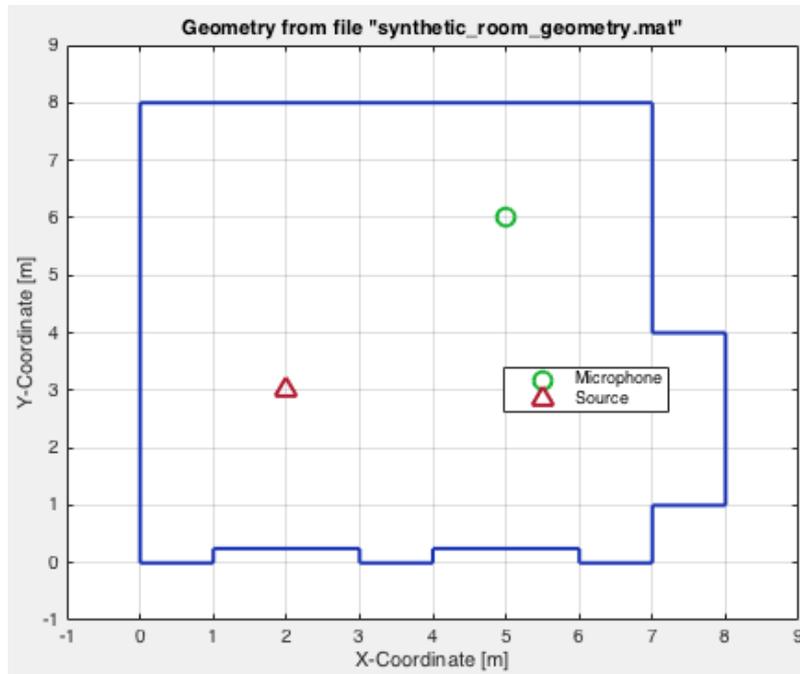


Figure 5.3: Designed room used for simulations in this section.

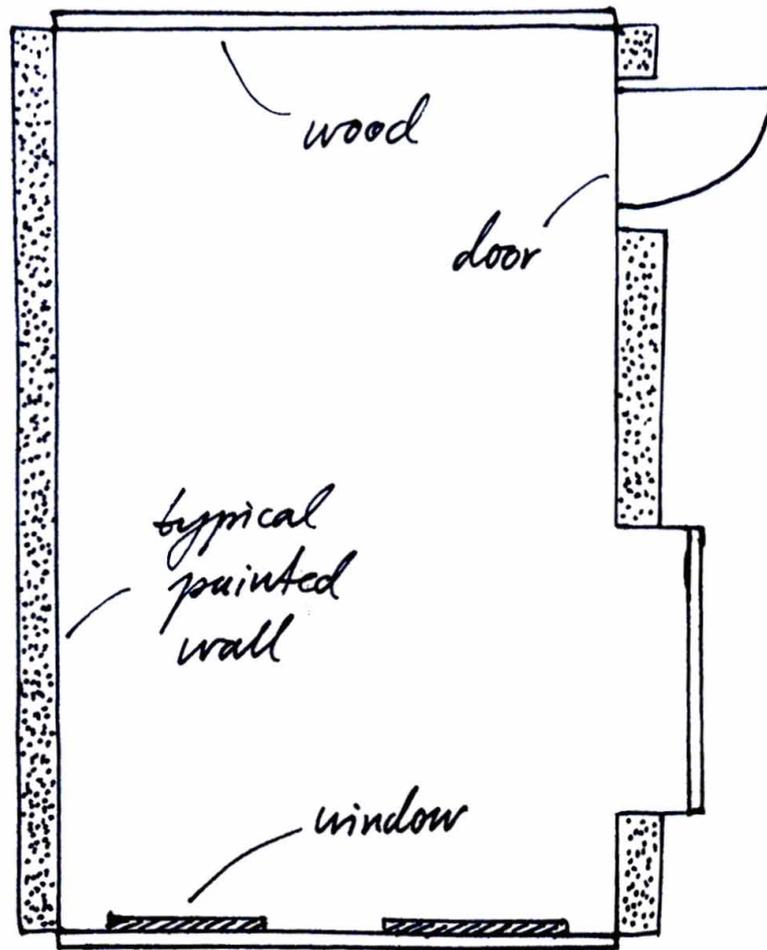


Figure 5.4: Hand drawn design, specifying materials used.

The results of this simulation come from a room with a geometry as given by figures 5.3 and 5.4. The simulation was conducted launching a ray every 0.5 degrees from the source position, and limiting the number of reflections to 25. The results are shown below in different formats. The value of 0.5 degrees was selected experimentally, and it seems clear that it has to be a function of the enclosure's dimensions, as small ones can be "scanned" with a larger angular spacing since to cover all boundaries with the same spacial resolution (longer distances at the same angular spacing will produce a more coarse scan of surfaces). Limiting the maximum reflection order to 25 is a decision based on:

- Keeping the computational demands reasonable to be able to perform simulations in a matter of minutes instead of hours. This criterion is obviously dependent on the machine used to run the simulations. In this study, most conducted ray-tracing

simulations were executed on a MacBook Air Intel i7 at 1,7 GHz, 8GB of DDR3 RAM at 1666MHz. Execution time varied from 2 to 20 minutes approximately, depending on the geometry and simulation parameters. For illustrative purposes, the ray tracing simulation of a shoebox room of 20 m^2 ($5 \times 4[m]$), with rays spaced 0.5 deg , required approximately 134 s for reflections up to order 10, 368 s for up to order 25 and 646 s for up to order 50.

- Reflections of very high order appear very late in the obtained IR, and are highly attenuated, mainly due to the spherical propagation and boundaries absorption. They can easily be close or below the noise floor, and they do not add much value to the whole IR in terms of estimation of geometry or absorption.

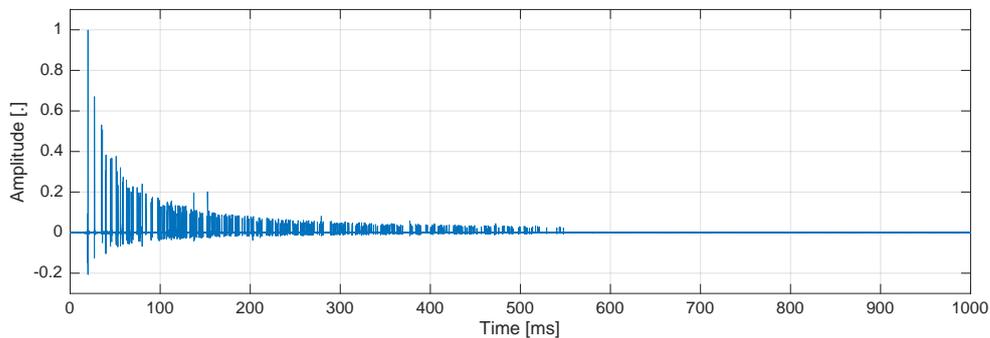


Figure 5.5: IR generated for a shoebox room of 20 m^2 . All walls completely reflective. Maximum reflection order of 25.

Figure 5.5 shows the IR obtained from simulating a shoebox room of 20 m^2 considering reflections up to order 25. The show the best case scenario, the SNR was set to the good value of 60 dB and materials are assumed completely reflective (absorption coefficient of 0.0). It can be observed that the tail of the IR has already reached a very low level and reflections density is high. This means it can be considered as within the reverberant field, and therefore not as useful for geometry or absorption estimation (since performing a time analysis of such part is very complicated having all reflections over imposed, and because every reflection has been deeply affected by air absorption phenomena).

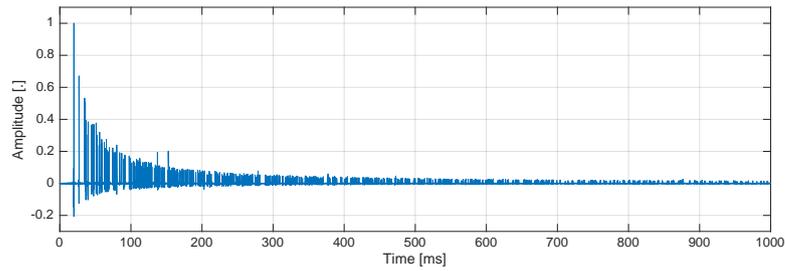


Figure 5.6: IR generated for a shoebox room of 20 m^2 . All walls completely reflective. Maximum reflection order of 50.

If the maximum order is increased to 50, the previous assumption of being within the reverberant is confirmed. This validates the hypothesis of not needing to compute ray tracing simulations for higher orders than 25 for such enclosure dimensions. It is important to remember that, for any real case, with worse SNR conditions and real materials that are not perfectly absorbent, the IR will fall into the background noise much sooner. An example of this is shown in figure 5.7.

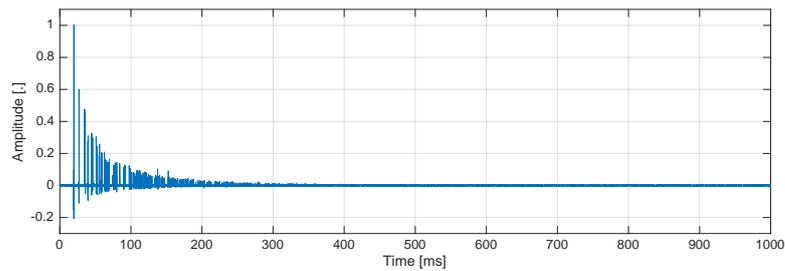


Figure 5.7: IR generated for a shoebox room of 20 m^2 . All walls with a reflection coefficient of 0.9. Maximum reflection order of 25.

From this short analysis, it can be concluded that any realistic scenario will not need to consider reflections higher than order 25, and probably that is already a very generous limit for some cases. Figure 5.7 can be compared to 5.5, where the reflection coefficient was decreased only to 0.9 and the SNR reduced in 5 dB from 60 dB to 55 dB, and the IR falls into the noise floor much before the ending.

Back to the description of the ray tracing procedure; all found virtual sources (346 in this example case, being the latest of order 25) are displayed in Figure 5.8 and 5.9.

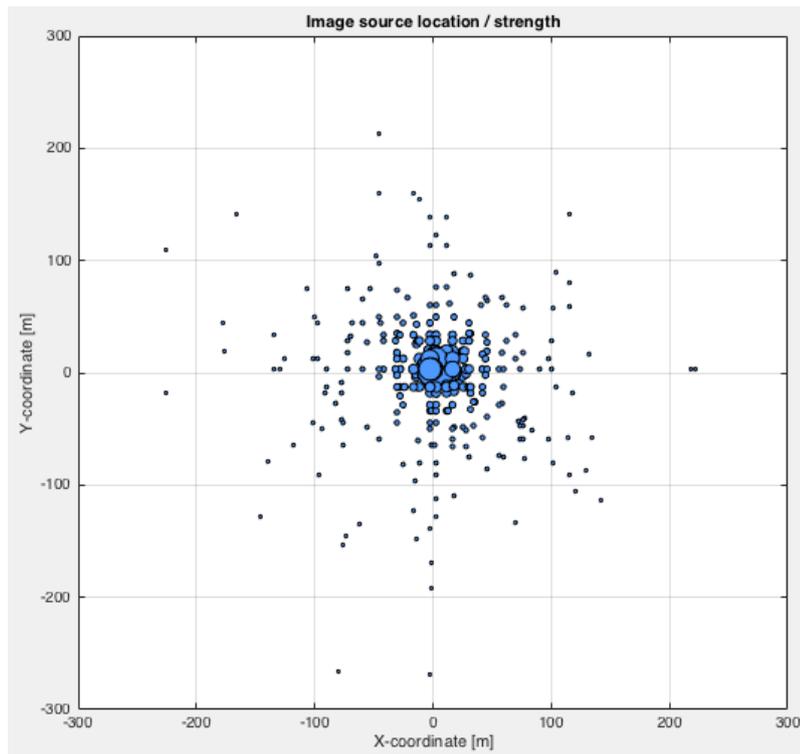


Figure 5.8: Complete set of virtual sources found.

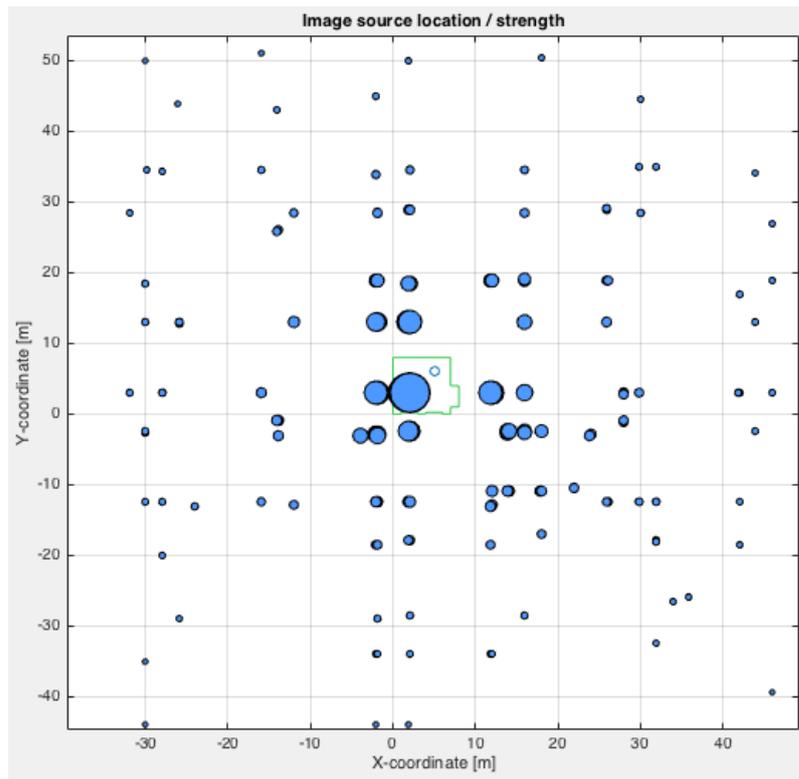


Figure 5.9: virtual sources found, zoom.

All virtual sources are used to generate a reflectogram, according to their amplitude and frequency response, as shown in Figure 5.10.

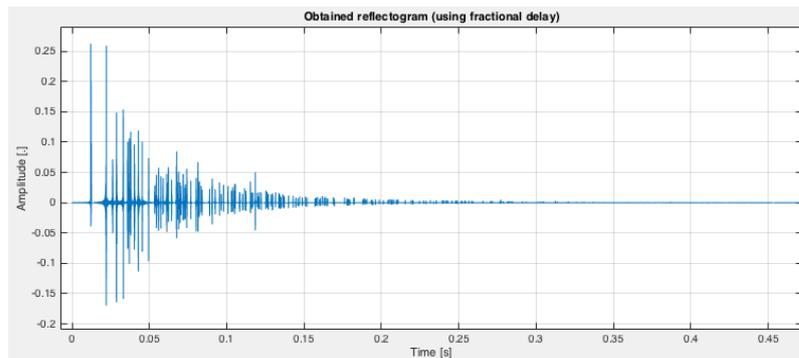


Figure 5.10: Reflectogram composed from all found virtual sources. Each of them produces a “peak” of different amplitude (due to the traveled distance and bounces absorption) and shape (the frequency response is modified as the wave travels and bounces).

As mentioned, if the IR is needed to include the effect of having a more realistic "speaker in

room" appearance, it can be convolved with the IR of the desired speaker. As an example, convolving the reflectogram with the impulse response of a typical studio monitor Genelec 1031A (Figure 5.11, frontal radiation, anechoic conditions) leads to the final synthetic IR, shown in Figure 5.12.

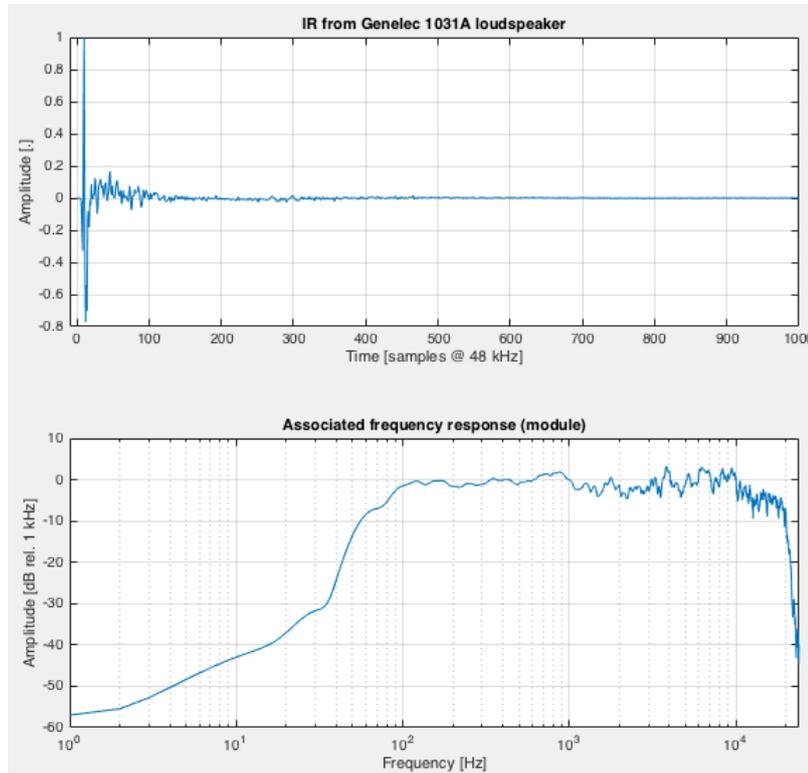


Figure 5.11: Measured response of a Genelec 1031A. Anechoic conditions, frontal radiation.

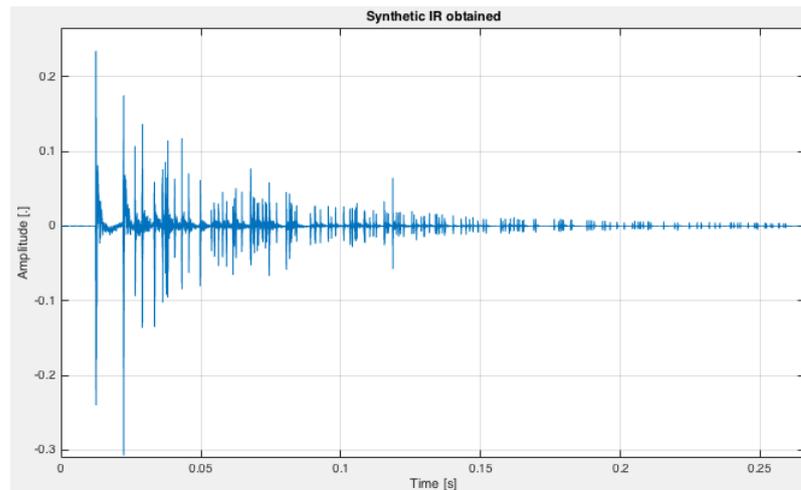


Figure 5.12: Final IR obtained.

The spectrum of the final impulse response was also computed, and is displayed on figure 5.13.

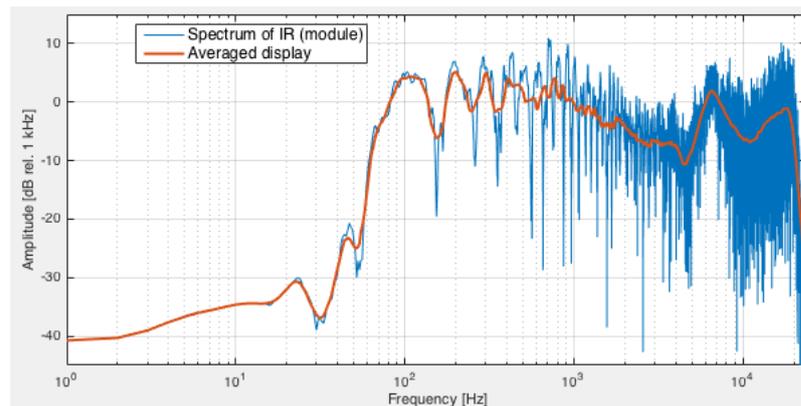


Figure 5.13: Spectrum of the obtained synthetic IR. Some clear comb filter effect appears, as a result of the addition of the strong early reflections. Some loss is also observed in high frequency, as some materials are more absorptive in that range. An averaged version is also plotted to help the reader visualise the behavior.

Comparing the final spectrum on Figure 5.13 to one obtained in a room of similar dimensions using the same Genelec loudspeaker (Figure 5.14) it can be seen that they have a resemblance in terms of the exponential decay and the pre-delay. The real one shows a much higher reflection density as well as worse SNR. This is expected since it comes from a real measurement of a room with reflections happening in three dimensions while the simulation is reduced to a 2D top view of the room. Also, the diffusion phenomenon in the simulation is somehow covered by the means of considering the receiver as a surface instead of a point, but this provides only a crude approximation. Finally, by looking at the time when

the tail fades into the noise floor (before 500 ms) it seems that the previous assumption for performing the simulation up to order 25 is appropriate.

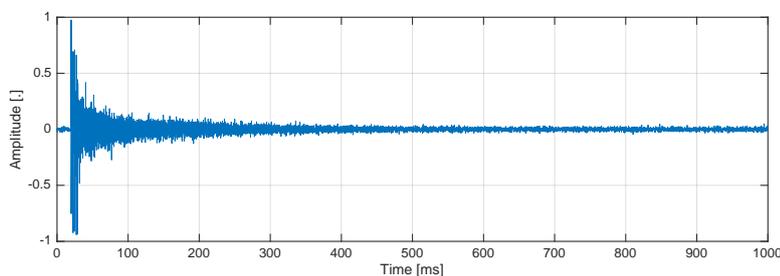


Figure 5.14: Real IR of a room of similar size, showing a much higher reflection density compared to the simulation results.

Lastly, the obtained IR can be used to process a dry audio sample and “listen” to the room. This is a typical application for modelling the reverberation of spaces digitally. Listening to such audios proved them to be not very realistic, but rather "robotic" sounding. This is expected since diffusion is only approximated and the density of reflections is low compared to a real case (only a 2D projection is simulated). It seems, although, that it has potential for being a good reverberation technique if such limitations are removed. Both the original and processed audios can be downloaded from Github: [original] [processed].

5.5 Implementation of air absorption

This section will explain the procedure followed to implement the air absorption phenomena as a computer program, in the form of a digital FIR filter.

5.5.1 Basis

The propagation of sound through air carries some energy dissipation, function of the frequency. Some of the factors [22] that influence the magnitude of that dissipation are:

- The non-uniformity of the propagation medium due to meteorological (refraction and turbulence). This will not be a main issue unless the volume of the enclosure is big enough to allow for considerable temperature changes and / or wind.
- Ground effect (it is an elastic medium capable of transmitting dilation and shear waves). This can be adequately modeled assuming that it is “locally reacting” [23],

so the waves within are negligible and the ray-tracing will be enough to describe its contribution.

- Absorption of sound in air.

The last addend is of special interest when simulating room impulse responses, since the traveled distance after a few reflections will effectively introduce a considerable attenuation. The lower bands of the audio frequencies are left almost untouched, while the higher ones (specially above 2 – 4 kHz) are highly attenuated. This attenuation is not linear, but exponential with the distance, so it can be described with a logarithmic unit such as [dB / km]. This parameter is denominated “atmosphere attenuation coefficient”, and is a function of the frequency of the propagated signal.

The mechanism by which the acoustical energy is absorbed in the air are mainly two [24]:

- Classical absorption: viscous losses due to friction between air molecules which results in heat generation.
- Relaxational processes: sound energy is momentarily absorbed in the air molecules that can then re-radiate sound at a later instant and partially interfere with the incoming sound.

The values of the attenuation coefficient can be calculated theoretically [25] and obtained empirically [26]. A set of values assumed to be applicable for most circumstances can be found in the standard ISO9613-2 [21]. The latter come as a set of discrete values corresponding to octave bands from 63 to 8000 Hz.

5.5.2 Implementation procedure

Based on the octave band values provided by the ISO9613-2, a digital filter is generated that can be used to process signals.

The desired impulse response of a FIR filter that emulates the behavior of air absorption is constructed from the frequency response using the inverse Fourier transform in its discrete form.

$$h[n] = F^{-1}H(f) \quad (5.2)$$

The frequency response used must be uniformly sampled in frequency. As the data provided by [ISO9613-2] is not, it needs to be resampled.

Using the data from table 2 on the ISO9613-2 and performing a cubic interpolation [27], the attenuation coefficient can be resampled. The result of doing so, with uniformly distributed frequency bins from 20 to 20 000 Hz is shown in Figure 5.15.

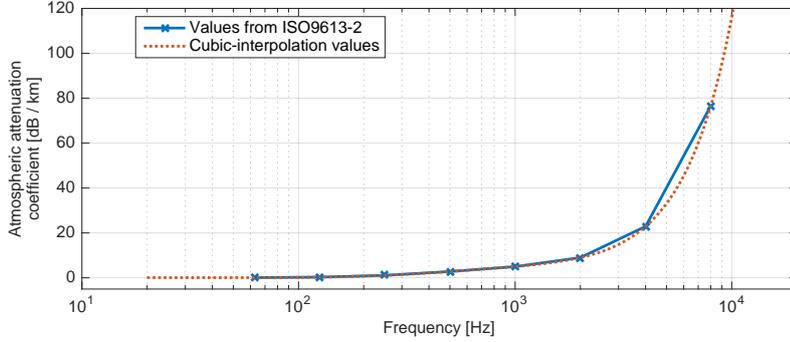


Figure 5.15: Air absorption as a function of the frequency. Original values from the ISO standard and interpolated trace.

This frequency response can now be converted into an impulse response by using the inverse Fourier transform. Previously, the attenuation coefficients are converted to a linear scale by doing:

$$a_{abs}(f_k) = 10^{\frac{-A_{abs}(f_k)}{20} \frac{d}{1000}} \quad (5.3)$$

Being A_{abs} the attenuation coefficient in [dB / km], a_{abs} its counterpart on a linear scale [.] and d the distance to the source in kilometers. This dimensionless values a_{abs} will be considered the module of the frequency response of the desired filter. The phase will be linear with non-zero slope, to produce a causal impulse response [28]. The slope of the phase will be set to a value that provides the lowest group delay, yet assures the obtained IR to be causal and stable (non oscillating).

$$|H(f_k)| = a_{abs}(f_k) \quad (5.4)$$

$$\angle H(f_k) = -\frac{f_k}{g\pi f_S} \quad (5.5)$$

Being g the group delay of the desired impulse response and f_S the sampling frequency. Its value is selected such that the obtained impulse response fulfils previous conditions, while keeping it as low as possible. The minimum group delay of the filter will depend on the amount of attenuation the filter has to provide, consequently on the maximum distance to the source that has to be emulated. For example, in case the maximum distance is set to

500 meters (sampling frequency of 48000 Hz), giving the group delay a value of 8 samples will produce an impulse response that has “lost” some information at the beginning (Figure 5.16).

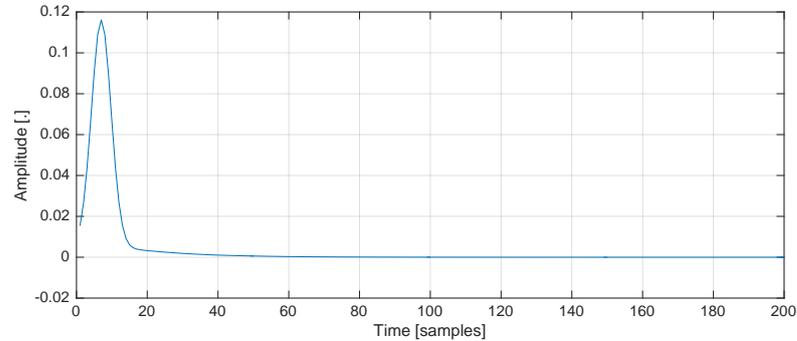


Figure 5.16: Impulse response obtained for air absorption filter. The group delay was selected too small; the beginning of the IR has lost some information.

If the group delay is set to 40 samples, instead (Figure 5.16), the obtained IR seems not to be incorrectly trimmed.

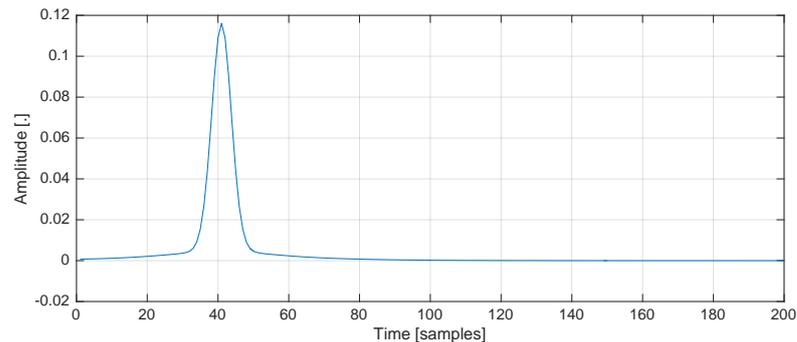


Figure 5.17: Impulse response obtained for air absorption filter. The group delay was selected properly; the beginning of the IR seems adequate.

The consequence of having a non-zero group delay is the introduction of a small delay in the processed signal. This is not avoidable (for the filter to be causal) and has to be considered when evaluating the feasibility of the system. Since it is a known value, the obtained data can be delay-compensated at the end.

5.5.3 Conclusion

The obtained impulse response has a Lorentzian shape [29] and necessarily introduces a non-zero group delay. The shape approximates a delta as the distance to the source tends to zero.

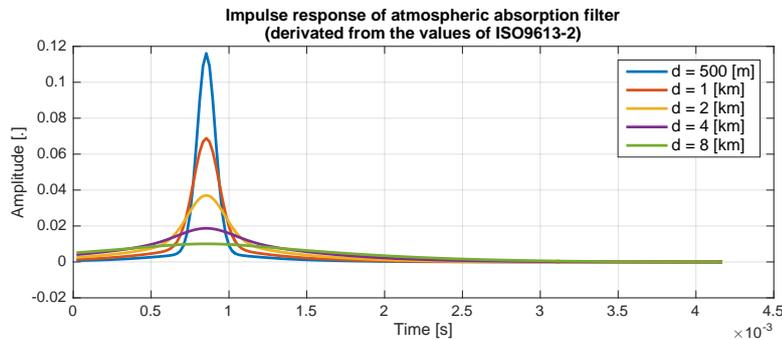


Figure 5.18: IRs obtained for air-absorption filters of different traveling lengths.

As seen on the Figure 5.18, when the distance grows, not only the peak gets smaller, the shape of the bell changes to a more flat one. This translates into having a variable frequency response, function of the distance to the source.

5.6 Implementation of wall absorption

Following the same procedure described for the air absorption phenomena to construct an FIR filter that implements the necessary frequency response shape, any boundary absorption can be simulated.

Typically, materials' absorption is defined as a set of values for octave bands from 125 to 4000 Hz. For a generic material of porous type, a typical absorption characteristics could be such as shown in Figure 5.19.

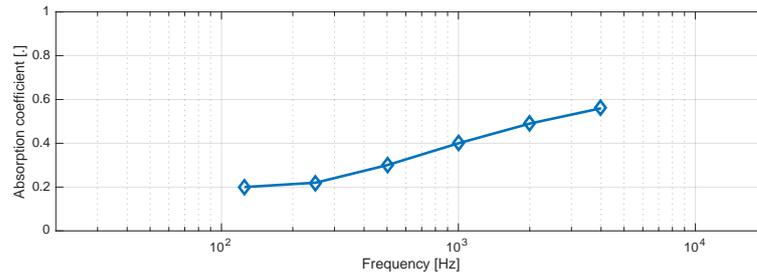


Figure 5.19: Absorption characteristics of a fictitious absorbent material of porous type. Absorbent coefficients are typically specified in octave bands for diffuse incidence.

With this data and using the procedure explained in the air-absorption phenomena implementation, the IR of an FIR filter can be calculated (see figure 5.20).

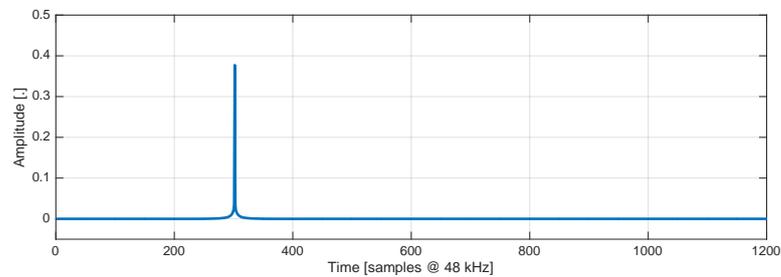


Figure 5.20: Calculated IR of an FIR filter that produces a similar effect to that of the absorbent material.

Computing the FFT of such IR (Figure 5.21), it can be observed that it corresponds to a low pass filter (as high frequencies are more absorbed than low frequencies) and that it follows the characteristics given by the material definition.

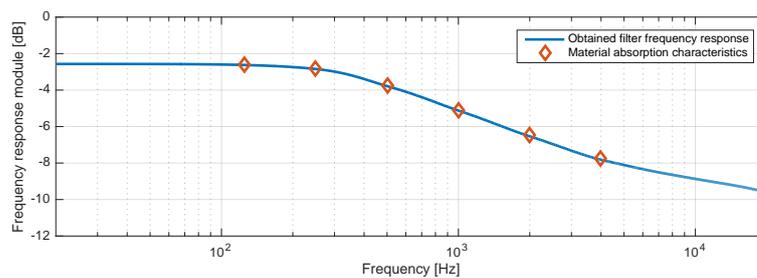


Figure 5.21: Frequency response (module) of calculated IR for the FIR filter. It follows the attenuation marked by the materials' absorption characteristics.

6 | Virtual sources cloud

This chapter will contain an explanation of the second module where methods for finding the cloud of virtual sources that represent a certain acoustical enclosure is analysed. It is divided in the following parts:

- Introduction to chapter
- IR usage
- Narrowband vs wideband DOA estimation
- DOA estimation
- Signal detection
- Signal estimation
- Array geometry
 - Array microphone count
 - Array size
- Time window IR analysis
 - Estimation of energy concentration areas through the IR
 - Estimation of the virtual source amplitude
 - Convolution of window signals
 - Estimated cloud of virtual sources

The introduction to the chapter will have content that helps clarifying the upcoming sections.

6.1 Introduction to finding virtual sources

As it is explained in Chapter 2, essential information about the room can be extracted if the sound field within is known. Such can be estimated from the cloud of virtual sources that represent the enclosure's acoustics, that in term are calculated from the direct and reflected signals recorded by acoustical measurements. These virtual sources can be placed in space if their distance and direction of arrival to the measurement microphone(s) can be found.

In order to obtain knowledge of the direction of arrival (DOA onwards) directional microphones [6] or arrays of microphones can be used along with some appropriate signal processing. The latter option is a typical choice and can yield a high accuracy and robustness. Having an array of microphones is more versatile (since multiple techniques exist to achieve any desired directivity pattern) and is a good step towards a more automated procedure in terms of room estimation (it can be built into a cheap system).

A simplified diagram of the problem is showed below.

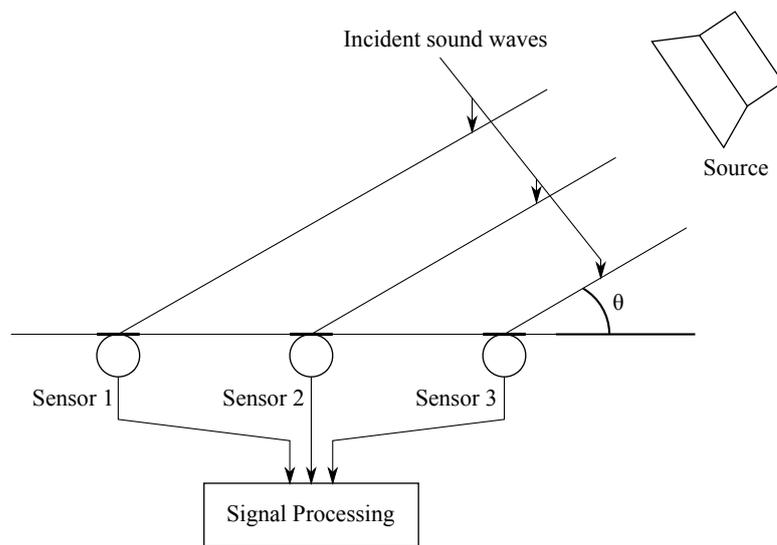


Figure 6.1: This figure shows the common case of study of estimating the DOA of a source. If far field conditions are assumed, the waves reaching all sensors of uniform linear array (ULA) travel in parallel directions. This is only presented for illustrative purposes.

It is seen in Figure 6.1 that some signal processing is needed that can give an estimate of the incident sounds wave angle to the sensor array. The captured signals are used to obtain an impulse response between the source and each of the microphones on the array, as described in section 6.2. There are several different well known ways of estimating the DOA of an IR. A discussion of possible candidates is presented after some practical assumptions are listed.

6.1.1 Assumptions and limitations

Here are the assumptions that are made prior to the signal processing. These assumptions cover the physical and practical aspects of the problem.

- It is assumed that the sensors and sources are omnidirectional in the frequency range of interest. A directivity function can be included, at a cost of a more complex simulation procedure. Omnidirectional microphones and speakers are available and should be used for real life implementation. A quasi omnidirectional speaker could be the OmniPower Sound Source Type 4292-L from Brüel and Kjær. When simulating, a fully omnidirectional source is used.
- As explained in the epigraph *Designed simulation assumptions*, the transmission medium, which in this case is air, is assumed to be isotropic and linear. This is to ensure the same propagation property is valid for every DOA [30].
- The signals that are transmitted from the source and virtual sources are highly correlated and in many cases even also coherent. This is because the reflections mostly are delayed, filtered and attenuated versions of the same source. This is a highly limiting factor for many DOA estimation algorithms.

6.2 IR usage

When studying the acoustical characteristics of an enclosure (a room under testing, for instance), it is generally considered a linear system, and as such, perfectly represented by its associated impulse response. This is interpreted as that the acoustic energy produced by a source, on a certain position within the enclosure, is processed by a linear filter and the outcome is observed on another position. The response will, therefore, depend on the position of both parts (transmitter and receiver) and the enclosure itself.

There are multitude of methods to measure impulse responses, that need not to be covered in this study as there is abundant literature in that regard. The data contained in this report corresponds to simulations where the necessary impulse responses have been synthesised, based on the spherical propagation model and the ray tracing method, as described.

In the case of a real implementation, an evaluation of the different available methods should be conducted. Without detriment to that, the authors would like to point to the two main techniques in use today; the maximum length sequence (MLS) and the logarithmic sine sweep. The first being recommended for those cases where the measurement needs to be performed on noise environments (such as occupied spaces) since it is very robust against such conditions [31]. The second being the best for achieving high signal to noise ratios

when a quieter environment is available. These and others are described and compared in [31].

6.3 Narrowband vs wideband DOA estimation

Narrowband beamforming techniques have been studied for decades [32], since they are essential for systems such as radars and sonars. They rely on the assumption that the bandwidth is small relative to the center frequency. There exist a vast bibliography to this respect. More recently, the acoustics beamforming have seen an increase in interest, due mainly to the computational capabilities reached by modern computers. Typically, for the acoustical cases, the signals are considered broadband by nature, which limits the direct use of most of the many algorithms that exist. Some of them have been generalised for the broadband case [33], [34] and others have been specifically designed for the application ([35], page 49), [36].

But because of the large amount of literature available for narrowband estimators and the fact that it is very typical to use a narrowband procedure to resolve the DOA of a wideband signal simply by using an appropriate narrow filter ([37], page 83) a narrowband estimator is chosen for this study.

6.4 DOA estimation

Now that the assumed conditions are noted, it is time to explore what options exist for such case. This will take basis in the summary of estimators from [37]. The summary divides all the estimators into 2 main groups. The first group consists of estimators that will work with any kind of sensor array. The second group consists of estimators that only works for uniform linear arrays. A uniform linear array is not optimal for this system 6.6 which means that only the estimators from the first group are wanted. That means that 7 estimators are to chosen between. It should be noted that the total amount of estimators available is far more than the 7 listed in Table 6.1. However these are in many cases variants of the listed estimators. Many of the variants tries to deal with some of the limitations affecting the different estimators.

Method	Consistency	Coherent signals	Statistical performance	Computations
Bartlett	N=1	-	-	1-D search
Capon	No	No	Poor	1-D search
MUSIC	Yes	No	Good	EVD, 1-D search
Min-Norm	Yes	No	Good	EVD, 1-D search
DML	Yes	Yes	Good	M-D search
SML	Yes	Yes	Efficient	M-D search
WSF	Yes	Yes	Efficient	EVD, M-D search

Table 6.1: This table holds the DOA estimators that are applicable to arbitrary arrays.

The first 4 estimators will fail if the signals are coherent which makes them unusable for this project. They rely on the covariance matrix having full rank and that is not the case for coherent signals. There exists methods that will restore the full rank property in the covariance matrix such as spatial smoothing [38]. This is an advanced task to perform on circular arrays and it is often not very robust in low SNR surroundings.

Therefore only 3 possible estimators remain from the table. A Deterministic Maximum Likelihood estimator (DML), a Stochastic Maximum Likelihood estimator (SML) and a Weighted Subspace Fitting estimator (WSF). These three estimators are all applicable to arbitrary sensor arrays and will work for coherent signals.

The DML estimator requires knowledge of the statistics of the signal. Since the acoustical environment is unknown to the system, it cannot be guaranteed that the transmitted signal resembles the received signal. On top of that the DML shows worse performance for large samples accuracy than the SML method [37]. Therefore the DML is not chosen.

The WSF and SML estimators does not need a-priori information about the impinging signals and they are very similar in both statistical performance and computational complexity. The WSF is less computationally intensive than the SML if a fast method of computing the eigenvalue decomposition is made [37]. Since there are no time constraints as is on this thesis this possible advantage is not considered.

Since the choice is between the WSF and the SML and since no significant difference can be found between them, the SML is chosen as the DOA estimator for this project because it has been most extensively researched.

6.4.1 Data model

The following parts will take basis in [39].

In order to form the maximum likelihood function a data model must be made. Here the standard data model is used. The model assumes d narrowband waves impinging on a M -sensor array.

$$\mathbf{x}(t) = \mathbf{A}(\boldsymbol{\theta})\mathbf{s}(t) + \mathbf{n}(t), \quad t = 1, 2, \dots, N. \quad (6.1)$$

Here $\mathbf{x}(t)$ is the array output, $\mathbf{s}(t)$ is the unknown vector source signal waveform, $\mathbf{n}(t)$ is a noise process. The number of samples is denoted by N . The noise process, $\mathbf{n}(t)$, is assumed to be white and distributed across all sensors in the array equally. The matrix $\mathbf{A}(\boldsymbol{\theta})$ is sometimes referred to as the steering matrix and it has the form.

$$\mathbf{A}(\boldsymbol{\theta}) = [\mathbf{a}(\theta_1) \dots \mathbf{a}(\theta_N)], \quad (6.2)$$

where $\mathbf{a}(\theta)$ is called the steering vector. $\boldsymbol{\theta} = [\theta_1, \dots, \theta_N]$ are the DOA of the corresponding impinging waves. The steering vector, $\mathbf{a}(\theta)$ is $M \times 1$, has a form that is array dependent. For the circular array that is used in this study (Explained in Section 6.6, the steering vector has the following form ([40], page 282):

$$\mathbf{a}(\theta) = [H_1(\omega_c)e^{-j\omega_c\tau_1} \dots H_M(\omega_c)e^{-j\omega_c\tau_M}]^T, \quad (6.3)$$

Where $\omega_c = 2\pi f$ is the angular frequency, τ_M is the time delay from source to sensor M and $H_M(\omega_c)$ is the transfer function for the M -th sensor.

It is seen that the number of sources is required. This is often not known beforehand, but a method of estimating the number of sources is presented in Section 6.5. It is assumed that the number of sources contained in a signal is less than the number of sensors in the array used. This is to ensure the uniqueness of the DOA estimation [41]. This means that in the proximity of the real DOA there exists an infinite amount of solutions since the number of unknowns exceeds the number of independent equations in the DOA estimator.

6.4.2 The maximum likelihood estimator

Here the maximum likelihood estimator is presented. The one chosen is the SML which assumes the signals to be random. Before stating the SML estimator, some of the parts hereof needs explanation.

Firstly the sample covariance matrix is stated,

$$\hat{\mathbf{R}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}(t_i)\mathbf{x}^H(t_i),$$

the signal covariance matrix estimate is,

$$\hat{\mathbf{S}}(\boldsymbol{\theta}) = \mathbf{A}^\dagger(\boldsymbol{\theta})(\hat{\mathbf{R}} - \hat{\sigma}^2(\boldsymbol{\theta})\mathbf{I})\mathbf{A}^\dagger(\boldsymbol{\theta}),$$

and the noise power is given by

$$\hat{\sigma}^2(\boldsymbol{\theta}) = \frac{1}{M-d} \text{Tr}\{\mathbf{P}_\mathbf{A}^\perp(\boldsymbol{\theta})\hat{\mathbf{R}}\}.$$

Here \mathbf{A}^\dagger is the pseudo inverse of \mathbf{A} and $\mathbf{P}_\mathbf{A}^\perp$ is the orthogonal projector onto the null space of \mathbf{A}^H , i.e.

$$\mathbf{A}^\dagger = (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H$$

$$\mathbf{P}_\mathbf{A} = \mathbf{A} \mathbf{A}^\dagger$$

$$\mathbf{P}_\mathbf{A}^\perp = \mathbf{I} - \mathbf{P}_\mathbf{A}$$

The above can be combined into a minimization problem,

$$V_{SML}(\boldsymbol{\theta}) = \log \det[\mathbf{A}(\boldsymbol{\theta})\hat{\mathbf{S}}\mathbf{A}^H(\boldsymbol{\theta}) + \hat{\sigma}^2(\boldsymbol{\theta})\mathbf{I}] \quad (6.4)$$

Where $\boldsymbol{\theta}$ is the angle(s) that minimizes the function. If a source is placed at a 0 degree angle of the array and the cost function, equation 6.4, is calculated for the angles from -180 to 180 then the following plot can be made.

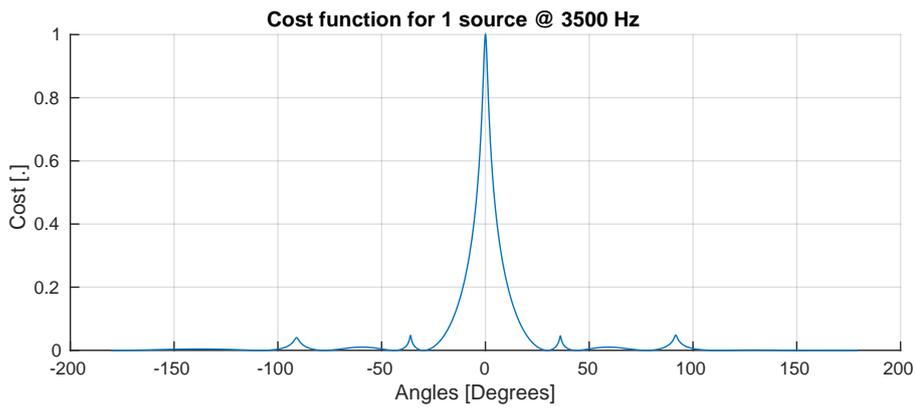


Figure 6.2: This plot shows the values of equation 6.4 for a source placed at 0 degrees. Here a full 360 degrees scan was made. The plot is normalized.

The Hilbert transform

The signals that are received by the microphones will be Hilbert transformed ([28], page 987). The Hilbert transformation is used here to remove the negative part of the signal spectrum. The Hilbert transform is often used to transform signals to complex baseband representation. If not the cost function will always have a mirrored source.

Now the minimization problem is stated and it should therefore be minimised.

6.4.3 Minimizing the SML

The procedure of minimizing $V_{SML}(\theta)$ will be explained in the following sections. Since the $V_{SML}(\theta)$ is non linear and multimodal, not every traditional minimization method is applicable or optimal ([42], page 52), [43]. Nonlinear optimization problems can be solved using a Newton based search method [37]. A Newton based search method will have to search through many of the local minima of the cost function. The same goes for the well known method, Hill climbing ([42], chapter 6). It is often not very efficient and without precaution it might get trapped inside a local minima. This is not the only way of solving the nonlinear optimization problems. Evolutionary algorithms are able to minimize multimodal optimization problems. These algorithms tries to mimic certain patterns seen in nature such as natural selection and mutation. In [44] the authors presents a genetic algorithm (GA) that minimizes the maximum likelihood estimator to estimate the DOA at great speed and accuracy. Genetic algorithms are much more efficient when it comes to searching a complicated solution space. Therefore this is the method that is used in order to estimate the DOA.

As stated, the SML will be minimized using a genetic algorithm. The GA is a combination of several steps which are shown in Figure 6.3.

Before going into details about the developed genetic algorithm, there should be provided some general information about genetic algorithms and the principles of these.

Chromosome: In this study a chromosome is thought of as a possible solution, or minimizer, to the cost function. That means that a chromosome will consist of a number of angles, set by the amount of DOAs to be estimated. An illustration hereof is shown in Figure 6.4.

Population: The population is a collection of possible solution to the cost function. Therefore the population is all the chromosomes.

Gene: A gene, in this study, relates to an angle in a chromosome.

Parent: A parent here, is one of two chromosomes that is selected for reproduction. Reproduction happens based on schemes and it is referred to as crossover, since the genes (Angles) are crossed over (See Figure 6.8).

Child: A child is the result of two parents reproducing. A child is therefore a chromosome that shares genes/angles from two other chromosomes.

Mutation: Mutation is a tool that is used to make angular changes to random angles (Genes) in random solutions (Chromosomes). It resembles the natural mutation, where small changes sometimes makes for a better solution.

Fitness: The fitness is another expression for the cost function value. The solution that lowers the cost function value the most, is thought to have the best fitness as it is the most fit solution.

With this information the following figure is presented:

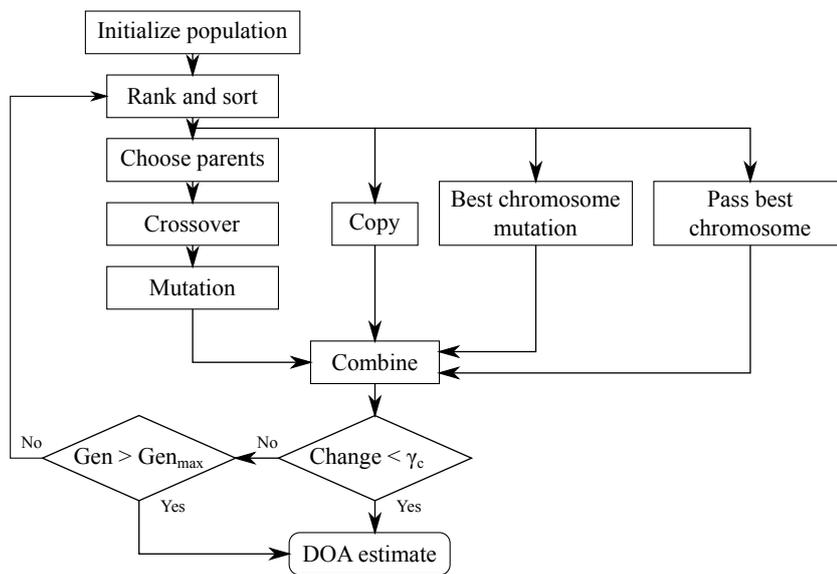


Figure 6.3: This figure shows how the genetic algorithm minimizes the ML optimization problem over Gen generations or until the change is smaller than γ_c .

This figure is made with inspiration from [45], [42] and [44]. But since most scenarios and use cases are unique it is only the conceptual idea and structure of the individual blocks that is based upon the literature.

In order to fully understand how the genetic algorithm there will be an explanation of each step in the following sections.

The initial population

The initial population is a randomly generated population that consists of $iPop$ chromosomes. Each chromosome is made of N -genes. N is the amount of sources that is contained within the signal evaluated. The population is generated within a certain range. In this case it is $[-180, 180]$ degrees.

The initial population is visualized in the figure below:

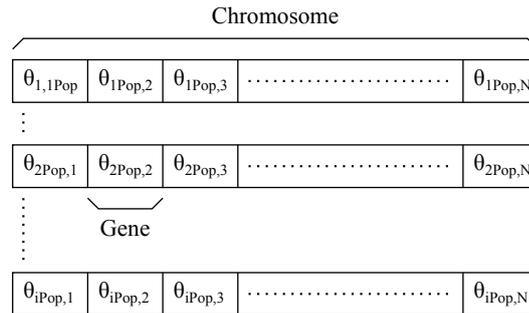


Figure 6.4: This figure illustrates how the initial pool of chromosomes is structured. Here, $iPop$, is the number of chromosomes and N is the number of sources expected to be contained within the signal evaluated. The θ 's are the angles.

As seen the initial population can be represented as a matrix of size $iPop \times N$. According to the authors of [44], the typical population size 30-80 and having a size of this ensures a good precision in DOA estimation of this type. Another 20 chromosomes are added to the population to ensure the variance is preserved for more generations. The initial population can be generated randomly or in smart way.

The smart way consists of using a Bartlett beamforming algorithm (Or any other sub optimal, but fast DOA estimator) to first scan the complete circle around the array [44], [39]. The output of the Bartlett beamformer is then evaluated and a linear initialization is done in the proximity of the detected coarse DOAs. This greatly improves the computational time because it creates a better starting point than the random initialization. An example of 100 chromosomes generated ± 15 degrees around the peak is shown in Figure 6.5

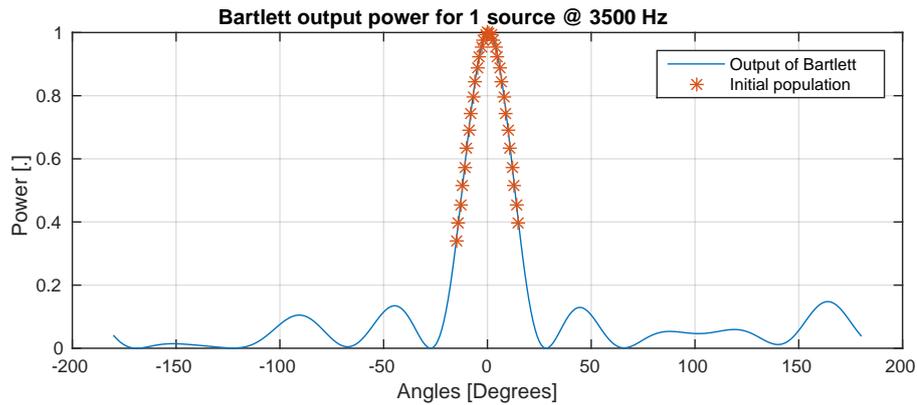


Figure 6.5: Here the output power of a Bartlett beamformer is computed for a circular scan with a source illuminating the array from 0 degrees. The initial population of chromosomes are created ± 15 degrees from the peak in the output power of the Bartlett beamformer.

Ranking and sorting

After the population is initialised it is used as input to equation (6.4). A fitness is then returned for each chromosome. The chromosome that minimizes the cost function is considered the best possible solution at the given generation and therefore it is ranked as number one. This is done for each generation iteration to ensure that the most fit chromosomes are preserved and passed through generations.

Parent Selection

Based on the sorted population some parents must be chosen.

It is illustrated in Figure 6.6 that the two fittest chromosomes always are passed on to the parent pool. From the third chromosome and on to the last chromosome of the population there is a probability of P_c of choosing the first over the second in every adjacent pair.

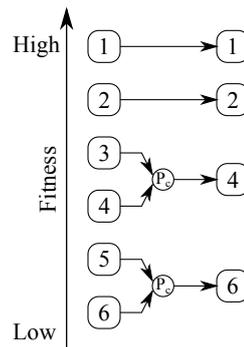


Figure 6.6: This figure shows how the parent selection is made. The two fittest chromosomes are passed to the parent pool and the remaining, less fit chromosomes are then chosen by random ($P_c = 0.5$).

It is seen that the number of parents created is equal to half the population size plus one. This ensures that the population size stays the same during generations.

Crossover

Now that the parents are chosen a mating scheme must be chosen. Different mating schemes have different characteristics. The main characteristics are: Convergence properties and maintaining the genetic variation¹. Two different schemes for this are the Emperor-Selective(EMS) method and the Best-Mate-Worst(BMW) method [45].

EMS is good for fast convergence since it allows for the most fit chromosome to procreate freely with all other chosen parents. This means that the good genetic material from the best chromosome is mixed with all other parents at each generation.

BMW is less good for converging but it allows the worst chromosomes to still get a chance. The worst chromosomes might in some cases have some valuable traits that otherwise would be lost. In other words, the BMW maintains the genetic variety for more generations which allows for a broader solution space search ([46], page 124).

In the case of this study there is not much need for a fast convergence. Robustness is more valuable and since the EMS method might cause the algorithm to converge towards a local minimum rather than a global the BMW method is chosen.

The BMW scheme is illustrated in Figure 6.7.

¹Genetic variation means that the chromosomes do not converge too quickly to a minimum, but that the search space is properly explored.

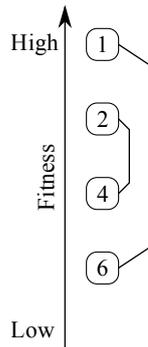


Figure 6.7: The figure shows the basic principle of BMW.

Each mating pair from the parent pool forms two new children. The children are made using linear cross over. Linear cross over means that two parents produce two children. The cross over point is chosen to be the halfway point in the parent chromosomes. At this point the genes are switched to generate two offsprings.

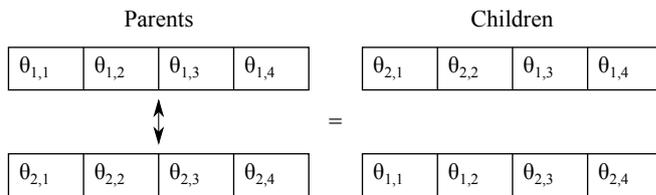


Figure 6.8: The figure shows how the cross over is computed.

Mixing the genes from different parents results in children with new attributes as to them of their parents. This is one of the basic method in evolutionary algorithms that makes them search throughout the solution space.

Mutation

If only cross over was performed on the giving population no new genes will be introduced and the algorithm might terminate trapped inside a local minimum which is in many cases undesirable. Therefore mutation should also happen to the genes. In this study there is a probability of mutation of $P_m = 0.05$. That means that there is a 5% chance that a gene will be changed for each child produced by the parents. The amount of change that a gene will be mutated is $\pm 0.5^\circ$. These values are found by trial and error. The reason for this is that nearly no scenarios are the same and therefore it might be that what some authors finds suitable for their scenario will not work for another scenario.

Although there are some guidelines for choosing the chance of mutation and size hereof.

Too high chance of mutation might constantly change the best chromosome and therefore limit the chances of convergence. Too small chance of mutation works the opposite way. May cause the algorithm to never escape a potential local minimum. The size of the change works in the same manner. A too small change might have no effect and a too big change might ruin some of the best chromosomes.

Copy

This module makes a copy of the current population and mutates every chromosome at a high chance of mutation. The probability of mutation is $P_{cm} = 0.5$. The reason for mutating this copied population is to keep the exploration of the solution space during generations. The amount of change is not kept constant in this module. It is changing based on the inverse of the variance of the fitness of the current generation.

$$\alpha_c = \pm \frac{1}{1 + \sigma_{Pop}} \cdot \pi$$

When the variance becomes smaller it is a sign of the algorithm beginning to converge. In this way the genetic variance is kept high in the current population.

Best chromosome mutation

This module mutates the best chromosome of the current generation by a very small amount. By changing the best chromosome by a small amount it is made sure that the best chromosome is at the bottom of a minimum. If it is not at the bottom of a minimum the mutated best chromosome will be a fitter version of the current best chromosome and therefore take its place as the best chromosome.

Pass best chromosome

This module simply passes the best fit chromosome to the next generation without any changes made to it. This is to ensure that the best fit is not lost due to mutation or cross over.

Termination criteria

The genetic algorithm must be terminated at some point. The termination occurs when one of two possible criteria are fulfilled. The first one is that if no change bigger than 0.1 degrees has happened over the last 15 generations it will terminate. If the first criteria is not met in time, the algorithm will terminate after a maximum amount of generations.

DOA estimate

When the genetic algorithm terminates one way or the other, chromosome that minimized the cost function the most is considered the final best solution and the angles of this chromosome are considered the estimated θ .

6.4.4 Validation of DOA estimation

In order to make sure that the DOA estimation fulfills the requirements stated in section 6.6.3, it must be tested. The tests made are as follows:

- Angular error for a single source present,
- angular error for multiple sources present and
- minimum resolvable angle difference.

These tests are conducted in Matlab and the essential code is listed in Appendix A. The conditions for these test are as follows:

Test parameter	Value	Unit
Number of sources	Varying	N/A
Distance from source to array	20	m
Radius of array	7.5	cm
Number of microphones	20	N/A
SNR	80	dB

Table 6.2: This table holds the information that makes the basis of this test.

The genetic algorithm is programmed to run for 150 generations or until the change over the last 15 generations is below 0.1 degrees. The population is made of 100 chromosomes. It sounds a little too high, but when there are no obvious time constraints so therefore the population will be set a little higher than recommended for genetic algorithms of this type. This is to ensure that the genetic variety is maintained. In the following tests, $\hat{\theta}$ will be the estimated angle and θ will be the actual angle.

6.4.5 Angular error for increasing amount of sources

These tests will show how much the DOA estimator deviates from an increasing amount of sources that is rotating 360 degrees around the sensor array in 1 degree steps. For the multiple sources test, the spatial spacing between the sources is always 60 degrees.

An illustration of how the simulation was done 1 source is shown in Figure 6.9.

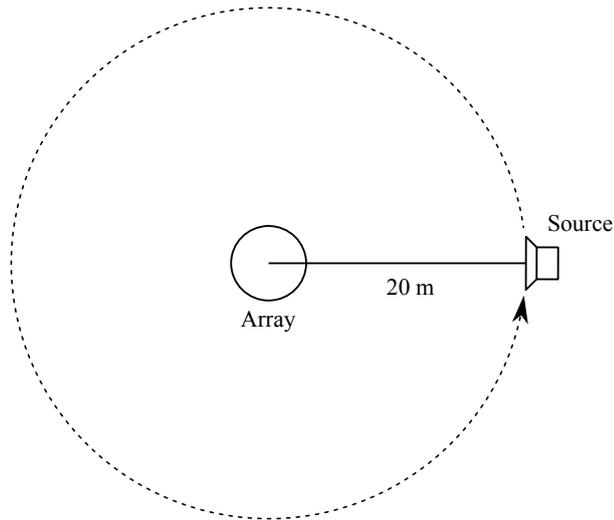


Figure 6.9: This figure is illustrating how 1 source is rotating around the array at a distance of 20 m.

Based on Figure 6.9 along with the implementation of section 6.4.2 and section 6.4.3 produces the following plot.

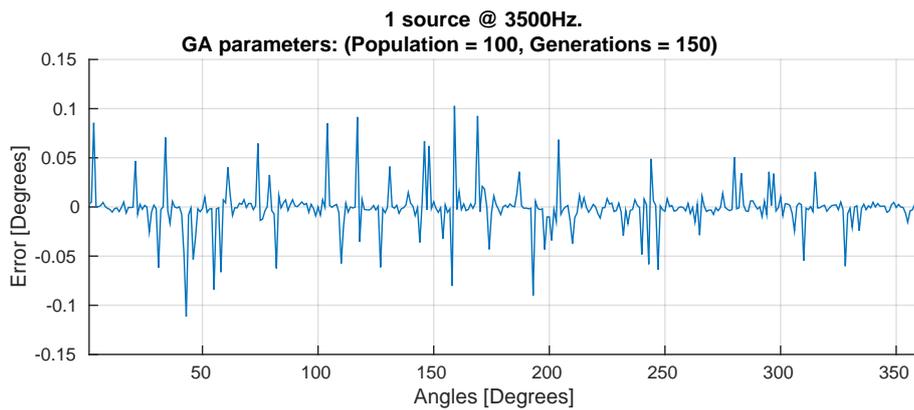


Figure 6.10: The figures shows how $\hat{\theta}$ deviates from θ when 1 source is exciting the array. This is data for one 360 degrees sweep of source angles.

It is seen in Figure 6.10, that the error when estimating the DOA of 1 source is well below the acceptable limit for all estimates of the DOA. Details about the performance with one source are listed in Table 6.3.

The same test is conducted for 2 sources rotating around the sensor array. The spatial spacing between the sources remain the same during the full rotation. A figure hereof is shown in Figure 6.11

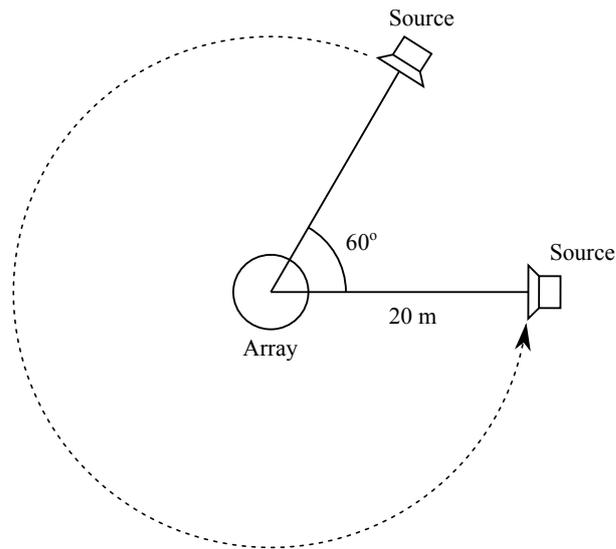


Figure 6.11: This figure is illustrating how 2 sources is rotating around the array at a distance of 20 m. The 2 sources are rotating in the same direction and they are rotating at the same speed. Note that it is no longer a complete 360 degrees rotation but a 300 degrees rotation.

Following Figure 6.11 and the implementation of section 6.4.2 and 6.4.3 the results is plotted below.

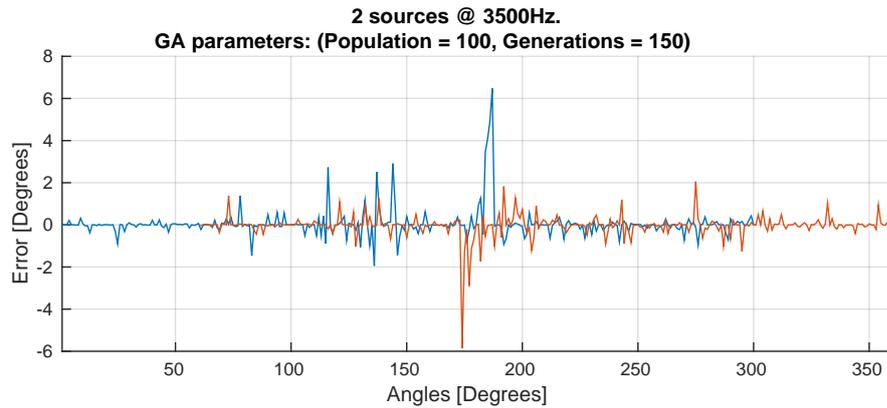


Figure 6.12: This plot shows how $\hat{\theta}$ deviates from θ when 2 sources are exciting the sensor array. This is data for one 300 degrees sweep of source angles

Figure 6.12 shows that error has increased as the number of sources went from one to two. There are peaks in the error plot that reveals violations regarding the minimum acceptable error. Closer inspections are made in Table 6.3. The last test made is for 3 sources. The setup is shown in Figure 6.13.

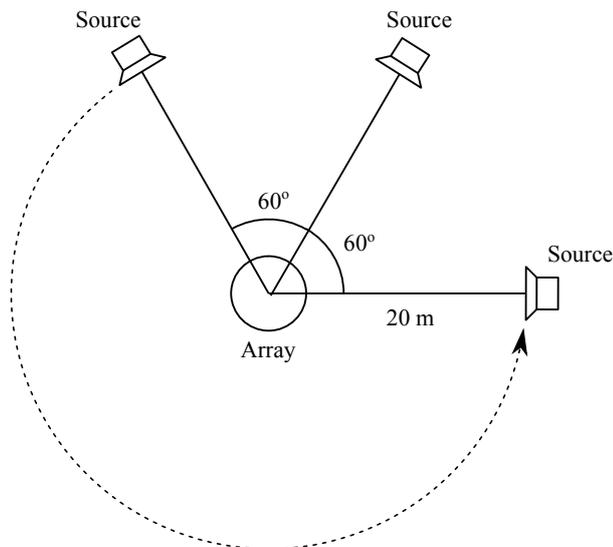


Figure 6.13: This figure is illustrating how 3 sources is rotating around the array at a distance of 20 m. All 3 sources are rotating in the same direction and they are rotating at the same speed. Note that it is no longer a complete 360 degrees rotation but a 240 degrees rotation.

Based on Figure 6.13 and implementation of sections 6.4.2 and 6.4.3 the following results were obtained.

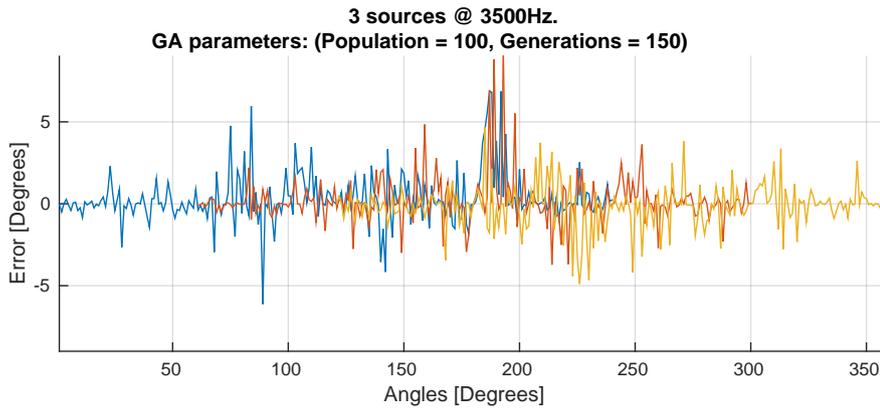


Figure 6.14: This plot shows how $\hat{\theta}$ deviates from θ when 3 sources are exciting the sensor array. This is data for one 240 degrees sweep of source angles

What is evident in Figure 6.14 is that the error has increased, and there are prominent peaks in the angular error. The upcoming Table puts some numbers to these errors.

Some brief statistics are made in Table 6.3.

Sources	μ [Deg]	σ [Deg]	Max [Deg]	Min [Deg]	$\hat{\theta} \geq 0.5 $	μ time [Sec]
1	0.0109	0.0224	0.1115	0	0 (100%)	5.8
2	0.2456	0.6248	6.4765	0	77 (87.2%)	21.1
3	0.8451	1.4297	9.1601	0	324 (55%)	53.7
3 (high perf.)	0.1321	0.3296	2.9292	0	52 (92.7%)	117.8

Table 6.3: This table holds the statistics for the angular error tests.

Based on Table 6.3 it can be concluded that the mean, variance, maximum error and time grows as the amount of sources grows. The time grows as the dimension of the cost function increases due to the increased amount of sources. The reason for the mean, variance and maximum error to increase is a result of keeping the "performance parameters" of the GA constant throughout the tests. Keeping these constant for an increase in complexity of the cost function leads to gradual increase in error. The tests were computed on a Sony Vaio VGN-NW21SF with an Intel Core 2 Duo CPU (P7450) @ 2.13GHz (dual core) running Ubuntu 14.04.

The following test is identical to the three source error test, only this time the performance of the GA is turned up at the expense of computational time.

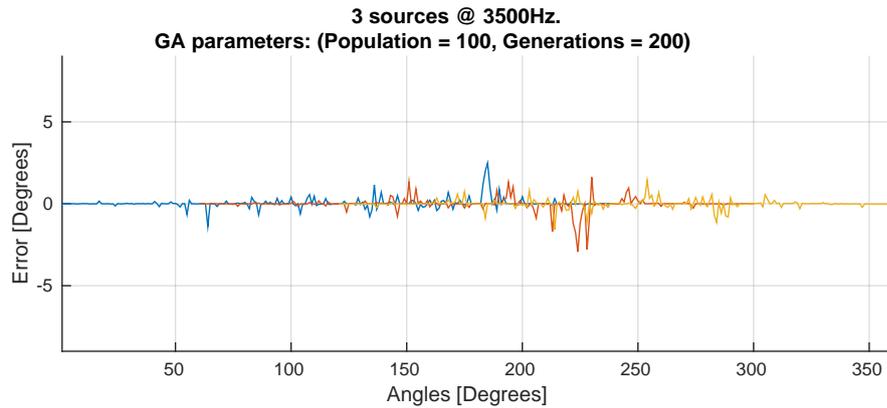


Figure 6.15: This plot shows how $\hat{\theta}$ deviates from θ when 3 sources are exciting the sensor array. This is data for one 240 degree sweep of source angles. Here there are 200 generations and no termination criteria.

It is seen that the error is greatly reduced. The statistics of this test is the fourth row of Table 6.3. If the SML minimizer is given the time needed it will minimize the errors significantly. Additionally, the SML estimator can attain the CRLB, therefore is an optimal estimator [47].

6.4.6 Minimum resolvable angle difference

In this test, the algorithms minimum resolvable angle is examined in order to see what can be expected. The closer to sources are to each other the harder it gets to distinguish them from one another. In this study it is not of greatest importance since it often is very unlikely that two sources that are placed close together in DOA are also placed close together in time.

The upcoming plots are only showing two sources, because the dimension of the minimization problem scales with the amount of sources and as a results of that it is very hard to visualize it.

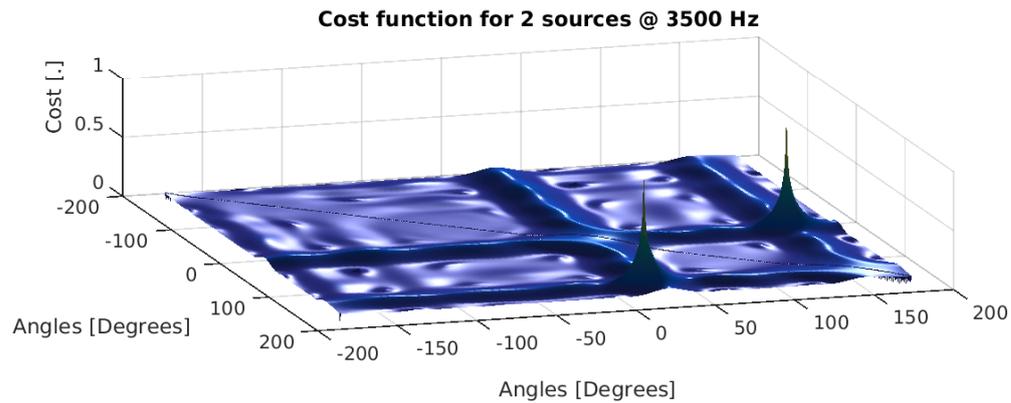


Figure 6.16: Here the normalized cost function (equation 6.4) is plotted in two dimension for two sources. The sources are $\theta = \{20, 150\}$.

The visualization in two dimensions reveals the accurate peaks in the normalized cost function. It is seen that the peaks corresponds to the actual source DOA. The following plots will show what happens when the sources are moved closer and closer together in DOA.

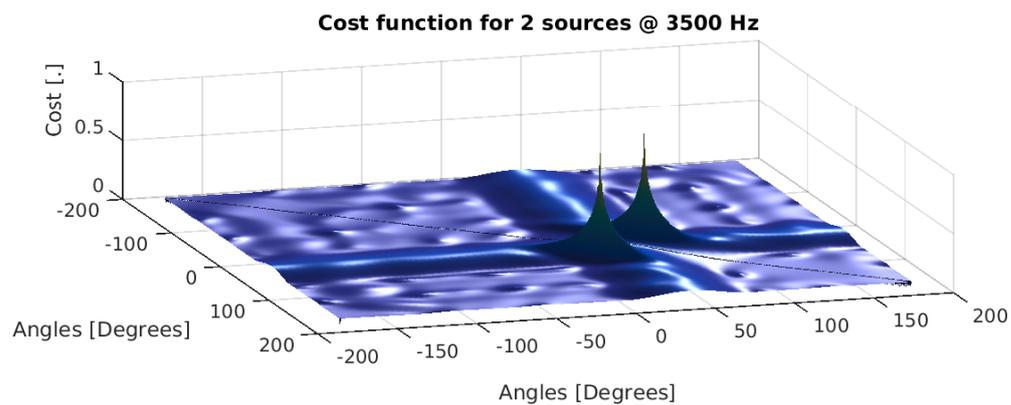


Figure 6.17: Here the normalized cost function (equation 6.4) is plotted in two dimension for two sources. The sources are $\theta = \{20, 60\}$.

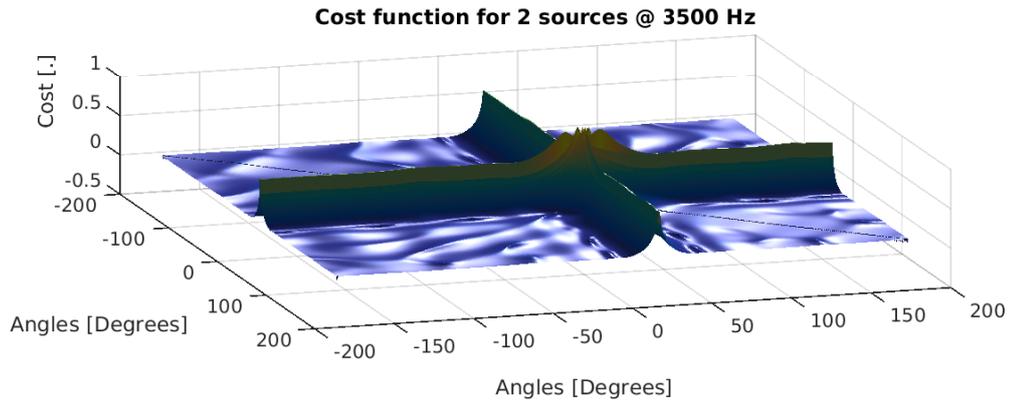


Figure 6.18: Here the normalized cost function (equation 6.4) is plotted in two dimension for two sources. The sources are $\theta = \{20, 25\}$.

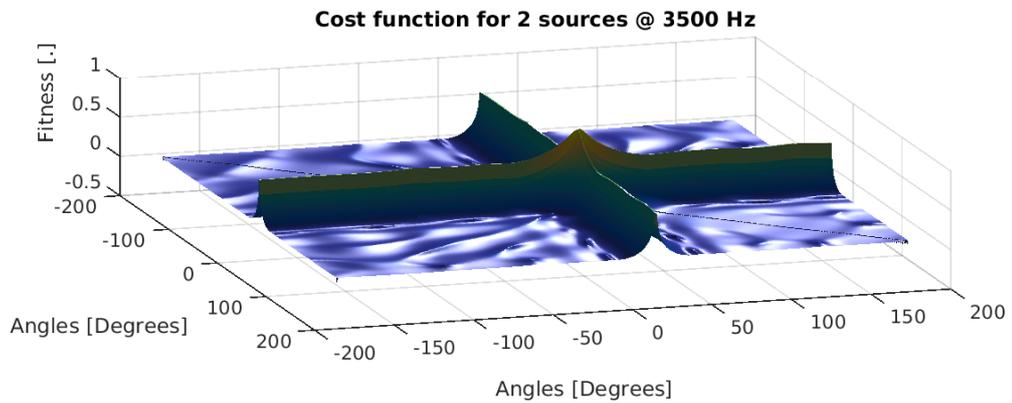


Figure 6.19: Here the normalized cost function (equation 6.4) is plotted in two dimension for two sources. The sources are $\theta = \{20, 21\}$.

It is seen that the solutions in the two dimensional cost function gets harder to resolve in terms of minimizing values. The following plot shows how the error in resolving two sources grows as the sources get closer and closer.

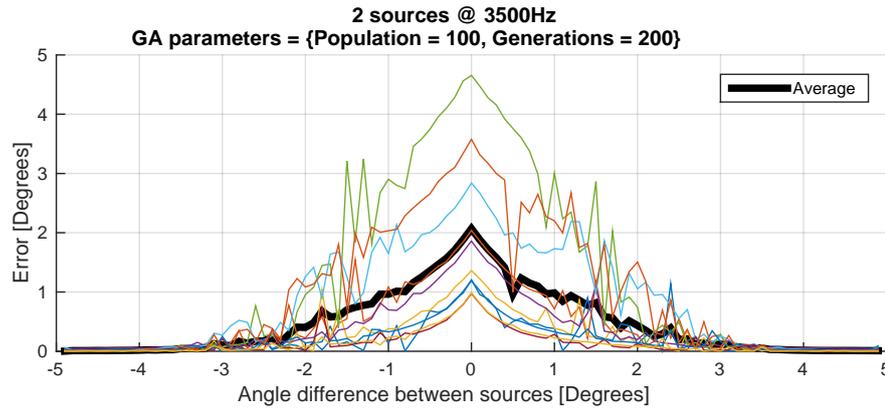


Figure 6.20: This plot shows the error in estimating two sources that moves closer. The average is based on 10 repetitions of the same test.

It is seen in Figure 6.20 that as the two sources are getting spatially closer they get harder to resolve for the DOA estimation algorithm. When sources are spaced more than 2 degrees in angle the average is below the needed precision for the system (see section 6.6.3).

6.5 Determining the amount of sources present

Up until now the $V_{SML}(\theta)$ has been tested with a known amount of sources. In [5] the authors try to choose a time window that theoretically only allows for 1 reflection, but that yields the detection of many virtual sources that is not there (Illustrated in Figure 6.36. This is a complex task since the time arrival of reflections are very dependent on room geometry and sensor array placement, but a method of properly making the time windows is made in Section 6.7. When the signals impinging on the sensor array are not correlated or coherent it is common practice to evaluate the eigenvalues and inspect which relates to the noise space and which relates to the signal space ([39], page 48). In this study the signals are indeed coherent and therefore the eigenvalue method renders useless. This is because the rank of the signal covariance matrix is full for non coherent signals and the method relies on the full rank property of the signal covariance matrix [39].

Since evaluation the structure of the eigenvalues of the signal covariance matrix another approach must be chosen. The generalized log likelihood ratio test (GLRT) and weighted subspace fitting for detecting the amount of sources in a given signal. Both methods are based on joint estimation of the number of sources along with their DOAs. In this study the GLRT is chosen since it can be used with the already described maximum likelihood estimator. The WSF would require for an additional DOA estimator to be implemented. It seems that GLRTs are widely used in scenarios like this study [48], [39], [37].

Before estimating the number of signals some notes are worthy of mentioning. When relating to the number of sources it is often denoted as the model order of the estimator. The reason for this is because the order of the V_{SML} increases with the amount of sources that it is trying to estimate. Therefore the model order will denote the amount of sources estimated to be in the signal.

A GLRT is used in order to test the following hypothesis'. Is the previous model order a good fit for the data or is the current model order a better fit?

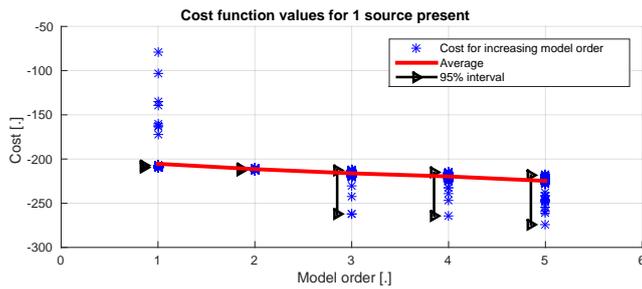
$$H_0 : k \text{ sources} \quad vs. \quad H_1 : (k + 1) \text{ sources}$$

According to [49] and [50] this can be stated as follows:

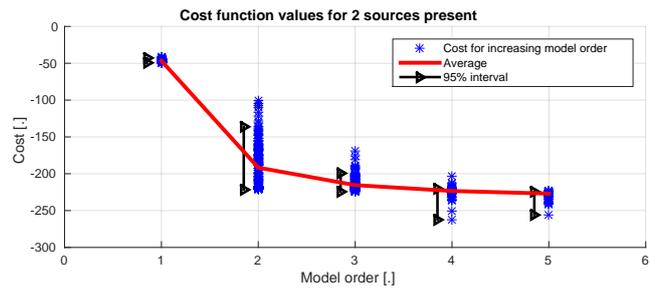
$$\frac{1}{N} \left(\frac{V_{SML}(\boldsymbol{\theta}^{(k+1)})}{V_{SML}(\boldsymbol{\theta}^{(k)})} \right) > \gamma \quad (6.5)$$

This is a well documented way of determining the model order by hypothesis testing. A scheme for testing these hypothesis is made in [39]. Since V_{SML} is already explained this section will try to determine a reasonable threshold, γ , for which the null hypothesis is accepted upon.

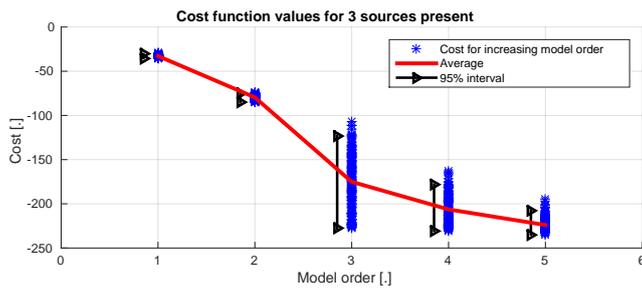
Multiple simulations were computed with 1, 2, 3 and 4 sources present in the signal received at the sensor array. Here the model order was increased after each estimation of the DOA. The results are shown below.



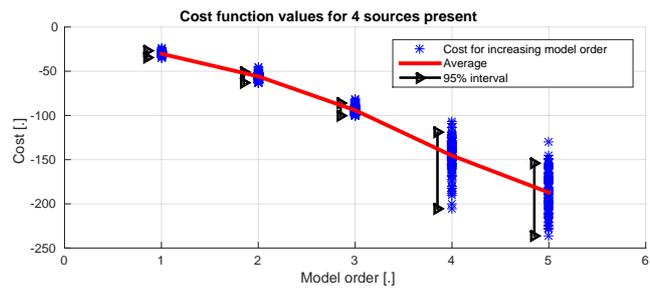
(a) Cost function values with 1 source.



(b) Cost function values with 2 sources.



(c) Cost function values with 3 sources.



(d) Cost function values with 4 sources.

Figure 6.21: These plots display the value of the cost function for increasing amount of sources exciting the sensor array.

Now the task is to determine a threshold based on the figures in Figure 6.21. In the simulation environment that gives the results in Figure 6.21 there is a clear trend in the cost function. The model order that yields a cost below -110 would provide the correct model order in most cases. However when the same test is run for the simulated IR (A much less controlled environment) it is seen that the threshold is much harder to determine.

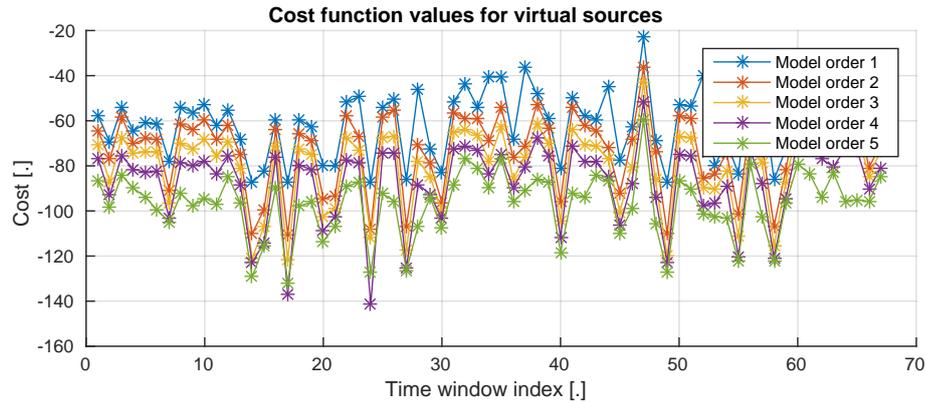


Figure 6.22: This figure shows the cost function for an increased model order when the input signal to the algorithm is the time windows from the simulated IR. Here the number of virtual sources contained in the evaluated signal is unknown and that is what should be estimated.

As it is seen in Figure 6.22 it is not easy to set a threshold or see a pattern when the number of virtual sources contained in the examined window is unknown. Therefore another method must be used.

6.5.1 New detection method

As shown previously, there exist several methods in the technical literature that can be used to estimate the amount of independent sources present on the captured signals by a microphone array. After some of the most typical have been studied, it can be concluded that, although they are valid and can be made to work for the case of application, it can not be done so without a great deal of fine tuning. Therefore a newer method is needed that can produce an appropriate estimation of the number of independent sources without requiring too much tuning for each execution case (for each scenario where the system may be needed to run).

Peak detection algorithm

Based on the assumption that the array is made sufficiently small, two or more independent sources reaching the array at the same time (or within the same time window) should produce separate peaks on most microphones, provided that they radiate from different enough angles.

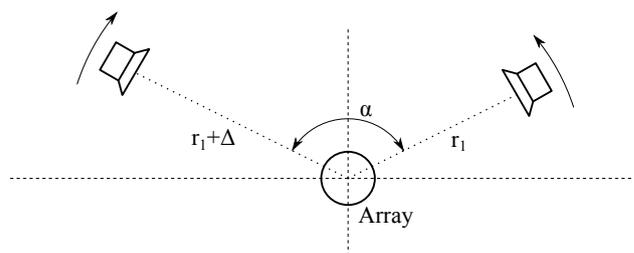


Figure 6.23: Setup of conducted test for determining the precision and working range of the new detection method. Two sources placed at nearly identical radii rotate getting closer to each other.

Following the setup used in previous sections, a circular array of 7.5 cm of diameter with 10 microphones was tested for different SNR values. The test consisted of two coherent sources, placed at approximately the same distance from the array (the first one at exactly 3 m and the second one at a slightly longer distance). Initially, they are located 180 degrees apart respect to the center of the array. Then, the new detection method is tested while they rotate in opposite directions (getting closer) until they can not be separated anymore (see Figure 6.26). This angle at which, for a certain radius difference, the two sources are detected as a single one, marks the limit of usage of the new detection method for each case.

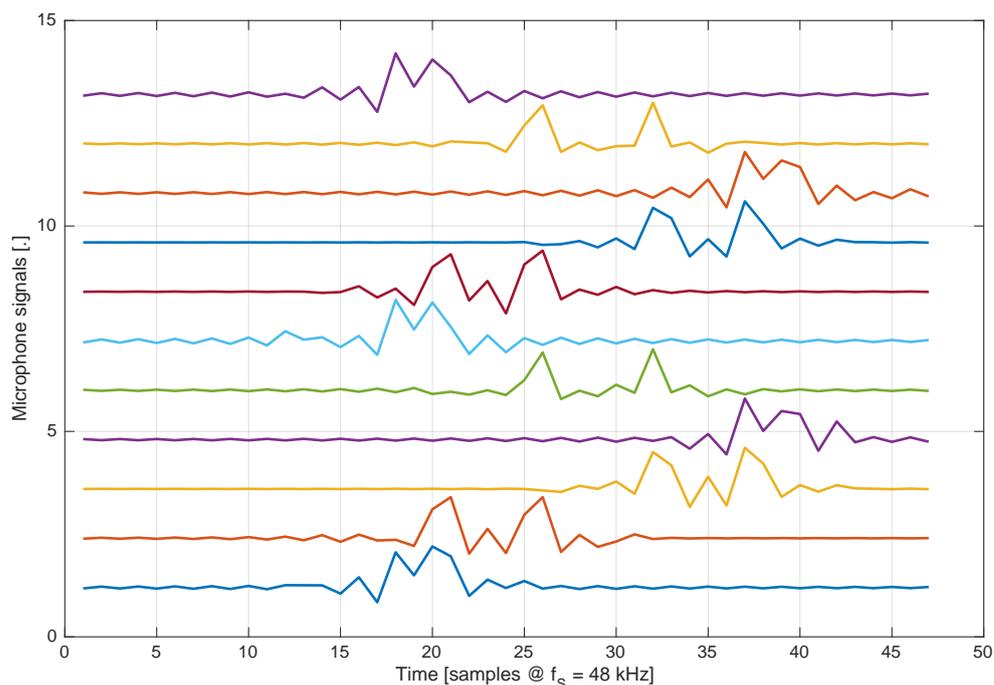


Figure 6.24: Example of time window signals captured by the microphone array when two sources placed at the same radius (3 m) radiate separated 35 degrees. Microphone signals vertically staged for visualisation purposes.

The captured signals (Figure 6.24), received as a time window, are rectified and low-pass filtered to obtain a slope for each microphone (Figure 6.25). As shown, each processed signal leads to a local maxima detection. The count of maxima detected over a certain threshold (defined as 2 dB above the background noise) is used to calculate the mode (value that is most repeated). The outcome is taken as the estimated amount of signals present. The value of 2 dB over the noise level as a threshold was determined experimentally by observing some windows at different SNR. The low pass filter is a simple moving average of 3 samples long, performed 3 consecutive times.

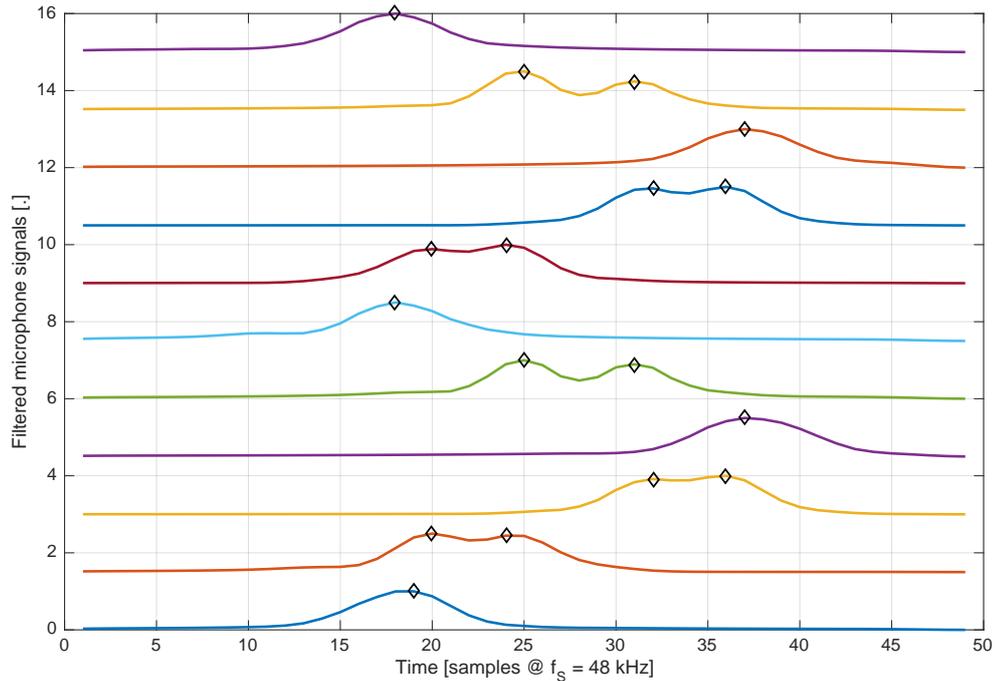


Figure 6.25: Example of low-pass filtered time window signals captured by the microphone array when two sources placed at the same radius (3 m) radiate separated 35 degrees. Showing detected peaks positions. Microphone signals vertically staged for visualisation purposes.

Figure 6.26 shows the results of the test conducted and how the SNR has an important role on how close in time the sources can appear (radius) in order to be still separable by the new detection method. It also shows that, provided the SNR is sufficient, the sources can be detected as independent regardless of the angle, if they are found at radii with a difference larger than 14 to 10 mm for SNR values in the range -20 to 40 dB. This is a positive result, since it will allow for detection of multiple sources within a time window for a broad range of SNR values.

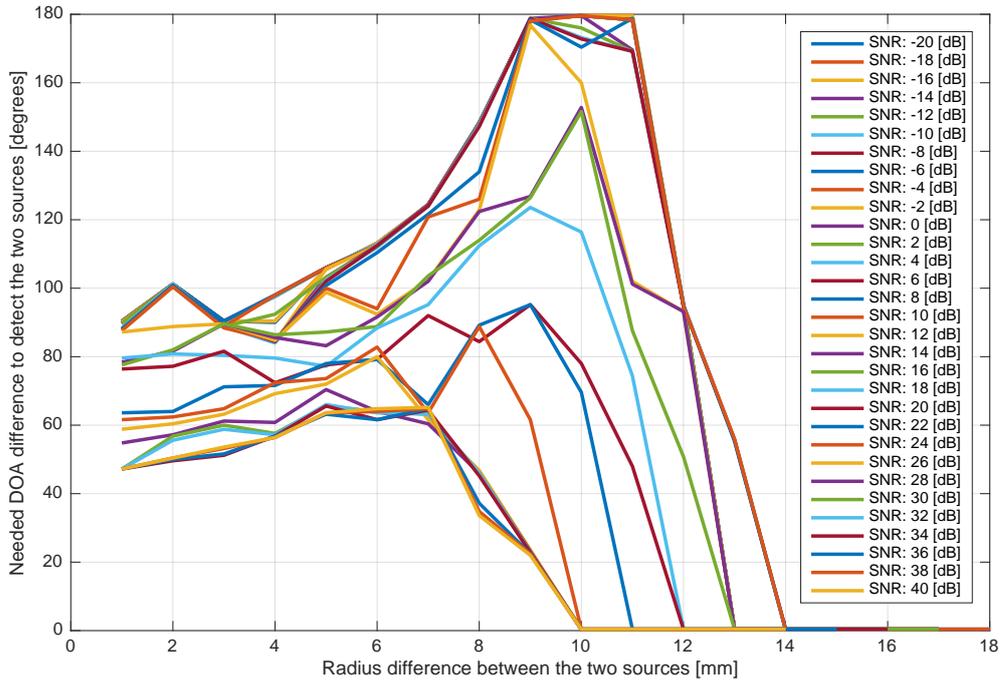


Figure 6.26: Results of the tests conducted. Necessary angle between coherent and closely placed sources in order to be separable by the new detection method. Tested for different SNR values.

For a window size of 1 ms, as discussed previously, the distance traveled by a sound wave will be of:

$$d = c \cdot t = 343 \cdot 0.001 = 0.343[m] \quad (6.6)$$

This means that, ideally, the new detection method should be able to identify up to 34 different reflections at arbitrary DOAs staged on time intervals of at least 1 cm at sound speed (29.1 μ s).

6.6 Array geometry

Given the characteristics of the problem under study, it is required that the array used to acquire information from the environment be capable of operating with the same ability in all directions. The problem, as mentioned before, is reduced to a 2D formulation in order to reduce the implementation complexity, knowing that the principles applied extend in the same manner to a 3D case. The uniform circular array (UCA) is the go-to option for these cases [51] [52] [53] [36] [54]. Other simpler geometries such as uniform linear arrays are not

suitable since they can not uniquely resolve the direction of arrival (given their construction symmetry).

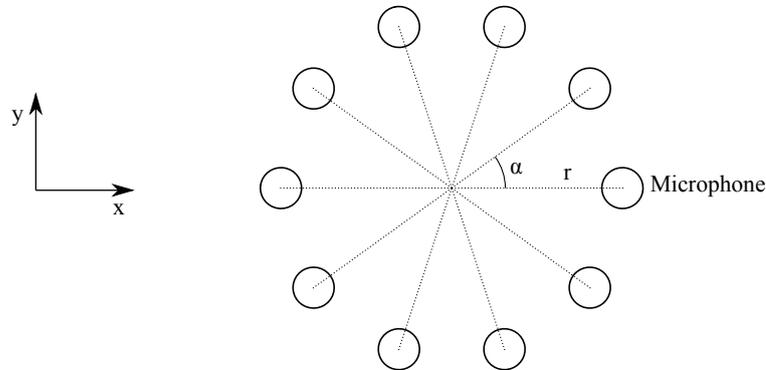


Figure 6.27: This figure shows a sketch of a 10 sensor circular array. The distance from the center of the array to each microphone is the same and so are the angles between each microphone.

More advanced geometries, such as concentric circular arrays, are also used in similar applications. They can usually provide a higher attenuation of side lobes and can incorporate additional processing techniques to increase the signal to noise ratio obtainable [55]. For the purpose of this study, a UCA provided accurate enough results.

6.6.1 Array microphone count

The two main factors when choosing the number of microphones are:

- the selectivity of the resulting array combined with the beamformer
- the required signal to noise ratio (SNR)

A poor SNR level will introduce additional errors (to those inherent by the precision of the system) and may even render the process unfeasible (the beamformer algorithm may not be able to properly estimate the DOA). As an initial, guideline value, the ISO-3382:1997 establishes that for a similar type of experiment (reverberation measurement) a SNR of 35 dB is adequate. The process described in this study also requires a source placed inside a room and several microphones to capture the direct and reflected signals from all surfaces, so this figure can be considered as a starting point when designing the SNR requirements of the new system. However, the characteristics of the array geometry combined with the beamformer of choice should allow the system to perform satisfactorily under such or better conditions.

Increasing the number of microphones will increase the maximum attainable SNR gain of

the system (SNR input-output relationship). With a set of 20 microphones, according to [56] page 26, the SNR gain will follow:

$$G(w) = \frac{oSNR(w)}{iSNR(w)} \leq M \cdot \text{tr}(\Gamma_v^{-1}(w)) \quad (6.7)$$

Where Γ_v is the noise covariance matrix and M is the amount of microphones. Assuming that all noise corresponds to a spatially and temporally white distribution, the covariance matrix will be:

$$\Gamma_v = I_M \quad (6.8)$$

And subsequently:

$$G_{WN}(w) \leq M \quad (6.9)$$

This is important to understand that the improvement of the system's performance in terms of DOA estimation will always be limited by the amount of microphones available. This provides a limit, but the actual SNR needed for the system to work will change according to the geometry and beamformer algorithm used. A series of simulations were conducted to evaluate the statistical ability of a UCA when paired with the SML beamformer combined with the SNR. The tests considered that the estimation was correct if it was within an error of 0.5 degrees (since it was calculated as the maximum admissible tolerance to assure good results, see section 6.6.3). The simulated system was presented with a rotating source (See Figure 6.9).

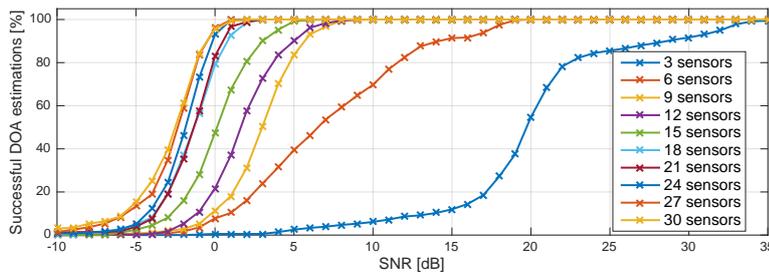


Figure 6.28: Probability of estimating correctly the DOA for different array element counts respect to the SNR.

The results show (Figure 6.28) that a low count of microphones would require a high SNR in order to assure a good performance. The initial criteria of 35 dB is clearly an unnecessarily good scenario for this combination of array geometry (UCA) and beamformer. This offers numerous possibilities when working in noisy environments. Choosing the number

of sensors is then a solution of compromise between requiring a high SNR and the need to incorporate many microphones and process their signals. If a central value is chosen (20 microphones), the system can be expected to work with a success rate of 95% for an input SNR (signals captured by the microphones) as low as 3 dB.

6.6.2 Array size

The aperture of the array, defined as the space occupied by the array measured in units of signal wavelength [40], page 276, has to be selected according to the highest signal frequency that will be received to prevent spatial aliasing, that is, to fulfil the Nyquist criteria applied to spatial sampling ([56], page 6). Hence, the interelement distance —separation between consecutive microphones— δ , must fulfil the following condition:

$$\delta \leq \frac{\lambda}{2} = \frac{c}{2f} \quad [m] \quad (6.10)$$

Being c the propagation speed of sound, λ the shortest wavelength and f the maximum frequency of the source. For the case of a UCA, the angular position for each element m is given by:

$$\varphi_m = \frac{2\pi(m-1)}{M} \quad [deg] \quad \text{for} \quad m = [1, \dots, M] \quad (6.11)$$

Being M the amount of sensors. Accordingly, the distance between adjacent microphones is:

$$\delta_{UCA} = 2r \cdot \sin\left(\frac{\pi}{M}\right) \quad [m] \quad (6.12)$$

Which can be approximated to:

$$\delta_{UCA} \approx \frac{2r\pi}{M} \quad [m] \quad (6.13)$$

If the microphones are very close (the arc is much smaller than the radius), meaning that M is large enough for the given radius.

Additionally, the use of a time window with a certain size is desired. To assure that all microphones are able to receive the signal within that time window, the maximum separation between all elements of the array (diameter) must be smaller than the distance traveled by a

sound wave for the duration of the time window. The optimal time window size is a matter of discussion [57] and different methods consider variable window size as a function of the frequency of the analysis and the time lag.

A common approach to deciding a time window size is derived from the statistical model of reverberation, a parameter known as echo density [58], page 29. It describes the average number of echoes in a room per time instant for any enclosure geometry:

$$\frac{N_r}{dt} = 4\pi \frac{c^3 t^2}{V} \quad [\text{amount of reflections per time unit}] \quad (6.14)$$

It requires the enclosure volume V , that can be approximated from an estimation of the reverberation time ([20], page 94). The previous expression allows to establish the time threshold for when more than N_r reflections are present in a time window of a given size:

$$\tau_1 = \sqrt{\frac{N_r V}{dt 4\pi c^3}} \quad [ms] \quad (6.15)$$

For the case of study, considered only in two dimensions, the expression would be adapted to match the units as:

$$\tau_{1,2D} = \frac{N_r S}{dt 4\pi c^2} \quad [ms] \quad (6.16)$$

Being S the surface in [m^2].

In [57] and [5] the time window used had a size of 1.4 and 1.33 ms respectively. Yet, both studies recommended using shorter windows in the conclusion to avoid multiple reflections within the same time window.

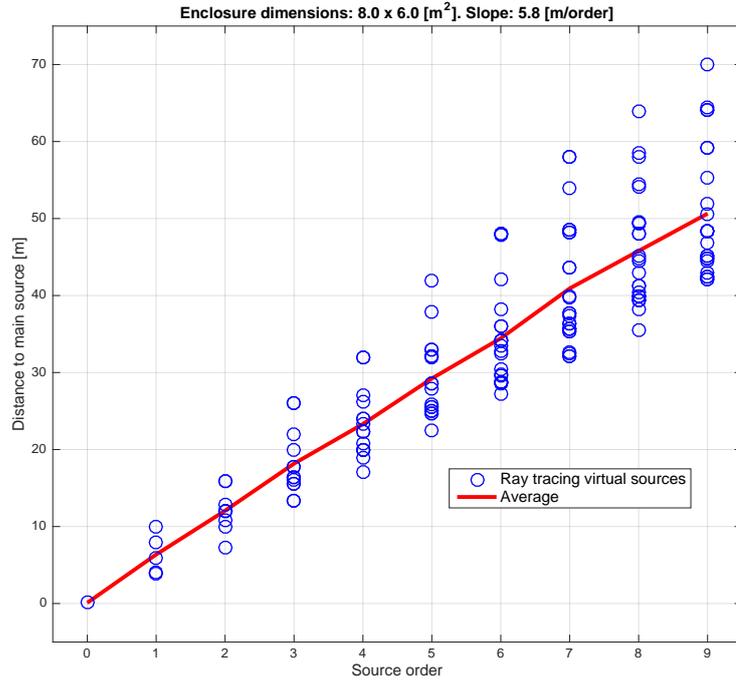


Figure 6.29: Mean distance to virtual source as a function of its order.

According to the tests conducted (shown in section 4.3), virtual sources of orders higher than 10 seem to have little contribution to the obtained power map. Thus, the analysis of the IR should focus on correctly estimating the virtual sources for orders below that threshold. In Figure 6.29, it is shown that the average distance of the virtual sources increases with the order in an approximately linear relationship. If the time $\tau_{1,2D}$ is set corresponding to the expected maximum distance of virtual sources of order 10 and lower, that is 70 m:

$$\tau_{1,2D,70m} = \frac{d}{c} = \frac{70}{343} = 204.1 \quad [ms] \quad (6.17)$$

Applying the previous value, for the example case of a small size room with top view dimensions of 8×6 [m], and expecting a single reflection per window for time lags before 204.1 ms, the time window should be of:

$$dt = \frac{N_r \cdot 8 \cdot 6}{\tau_{1,2D,70m} 4\pi c^2} = \frac{1 \cdot 8 \cdot 6}{204.1 \cdot 1 \cdot 4\pi c^2} = 0.16 \quad [ms] \quad (6.18)$$

This value, although statistically optimal for avoiding multiple reflections to fall into the same window, is impractical since a propagating wave would only be able to travel a distance of:

$$0.16 [ms] \cdot 343 [m/s] = 5.46 [cm] \quad (6.19)$$

That would force the array aperture to be extremely small in order for all microphones to receive the signal. Then, the frequency limitation criteria needs to be evaluated as well to find a more reasonable value. As explained before, the aperture must not exceed half the wavelength of the maximum frequency. The upper limit in frequency is given by the needs of estimating the absorption of materials as a function of frequency, and those are typically specified up to the 4 kHz octave band [3], Table A.2. There is no intrinsic lower limit in frequency, but the ability to beamform and estimate DOA will degrade as the wavelength gets farther from that corresponding to double the separation between elements, since the difference in captured signals between microphones is reduced (longer wavelengths). Using very closely placed microphones is another possibility, and in such case it is defined as a "differential microphone array", since the magnitude measured (sound pressure) changes almost differentially between adjacent elements. These techniques are fully described in [56], but are not considered in this study for simplicity of implementation. Finally, the interelement distance of the array should not exceed then:

$$\delta \leq \frac{c}{2f} = \frac{343}{2 \cdot 4000 \cdot 2^{1/2}} = 3.03 [cm] \quad (6.20)$$

And for a UCA, this would lead to a radius of:

$$r \approx \frac{\delta_{UCAM}}{2\pi} = \frac{3.03[cm] \cdot 20}{2\pi} = 9.64 \approx 10 [cm] \quad (6.21)$$

That a propagating wave would travel in:

$$\frac{d}{c} = \frac{2 \cdot 10[cm]}{343[m/s]} = 0.584 \approx 0.6 [ms] \quad (6.22)$$

This value, being larger than the statistically optimal, will make the windows to contain more than one reflection at earlier time instants. In those cases, the estimation of distance and DOA must be able to function with multiple coherent sources within the same time window. This value was tested as window size, and the data of multiple windows visualised to evaluate its suitability.

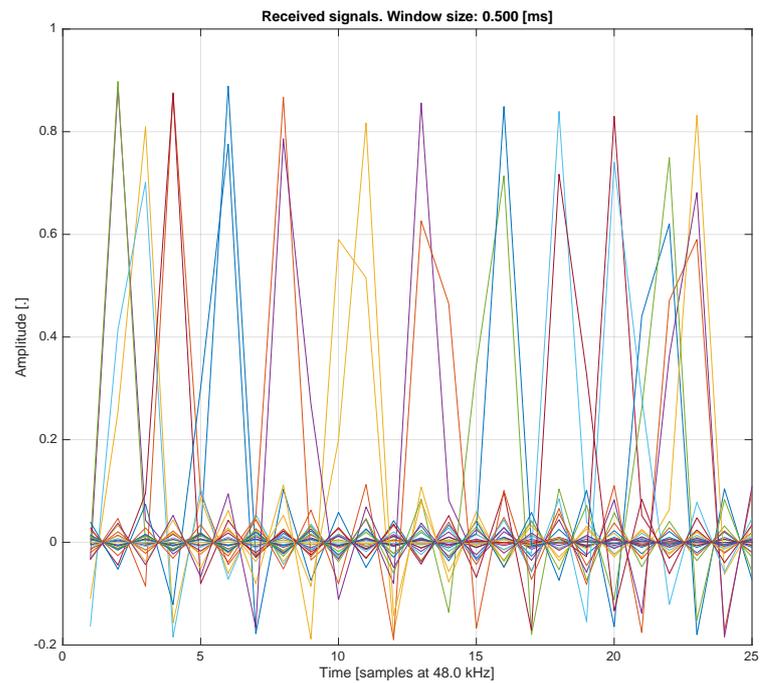


Figure 6.30: Captured signals on a window of 0.5 ms. The signal of each microphone is presented in a different color.

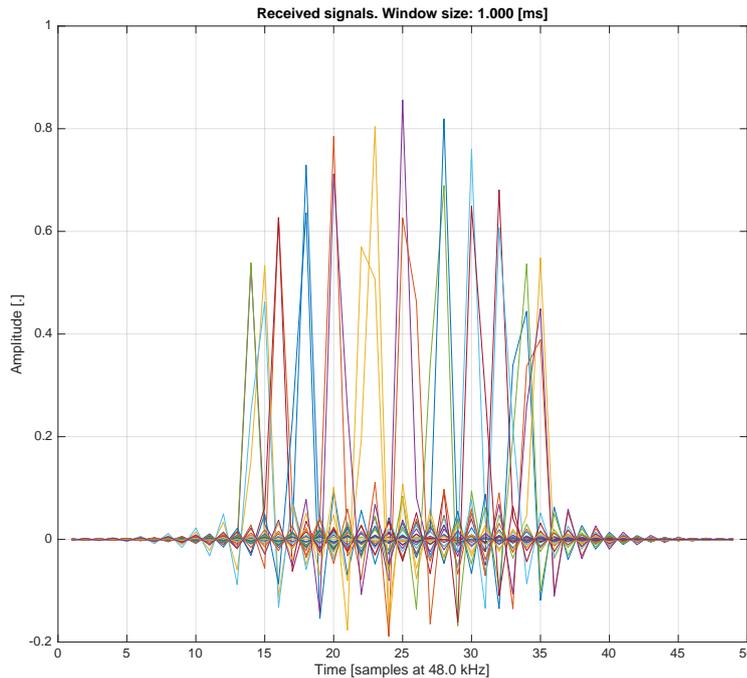


Figure 6.31: Captured signals on a window of 1 ms. The signal of each microphone is presented in a different color.

From the figures 6.30 and 6.31 it can be seen that, while a window of 0.5 ms can be perfectly good to capture the signal arriving from a source on all microphones, it relies on the ability to estimate the central time (where to place the window). In the case where the central time estimation contains some offset, the first or last microphones receiving the signal could be cropped. Using a slightly longer window can prevent this, with the downside of increasing the chances of containing multiple reflections. A Hamming window can be applied to try to minimise the effect of adjacent signals. The possible downside is attenuating some of the microphone signals that are the first and last to receive the impinging signal, so it has to be implemented with care.

From this, it can be concluded that a UCA formed by 20 microphones with a radius of 10 cm should be appropriate for receiving signals up to the 4 kHz octave band, and adequately analysed in windows ranging from 0.5 ms to 1 ms. It will require algorithms for DOA estimation capable of working with multiple coherent sources since several reflections will probably exist within the time span of interest.

6.6.3 Revisiting the needed angular precision

In section 4.3.1, it was determined that the required precision for locating virtual sources could be expressed in terms of a radial (that will correspond to an error in the arrival time measure) and an angular component (associated to the ability to discern the direction of arrival of the virtual source).

On the view of Figure 6.29, and based on the hypothesis of not considering virtual sources up to order 10 as the most important for the purposes of this study (see section 4.3), the mean distance to virtual sources will be 30 m and the required angular resolution can be resolved as:

$$\theta_E = 2 \cdot \text{asin} \left(\frac{2 \cdot 0.125}{2 \cdot 30} \right) = 0.48 \approx 0.5 \quad [\text{degrees}] \quad (6.23)$$

Having a system with a worse angular resolution than this value will not assure good results in the estimated geometries. This does not mean that the system can not function at all, but that the results need to be further evaluated to assess their fitness.

6.7 Analysis procedure of time windows from the IRs

In order to separate the contributions of different reflections (that can later be considered virtual sources), it is a known technique to analyse the impulse response data in windows [5]. Typical techniques are based on the use of a sliding window, that runs through the impulse response, each time estimating the probability of signals to be contained and, if any, the DOA of each of them. This method seems to work reasonably well, but it can lead to false detections, resulting in a partially incorrect estimated map of virtual sources [5], figure 3. A new method has been developed to minimise the false detections.

The developed procedure consists of

- estimation of energy concentration areas through
- the impulse response application of window on each detected point of evaluation
 - convolution of window signals with a sine wave
 - detection (estimation of amount of signals)
 - beamforming (estimation of DOA of each signal)
 - estimation of their amplitude

The estimated cloud of virtual sources (with their position and amplitude) has been found to be more accurate this way than when the sliding window method is used.

6.7.1 Estimation of energy concentration areas through the impulse response

Processing the signals received by the array can provide information about where signals may be present. This allows the system to avoid needing to perform a costly analysis such as that of a sliding window. It also reduces the possibility of false detection (mistaking noise for actual virtual sources).

A simple method is to calculate the sum of all microphone signals and low pass filter it to obtain a soft envelope curve where the peaks can be detected more easily.

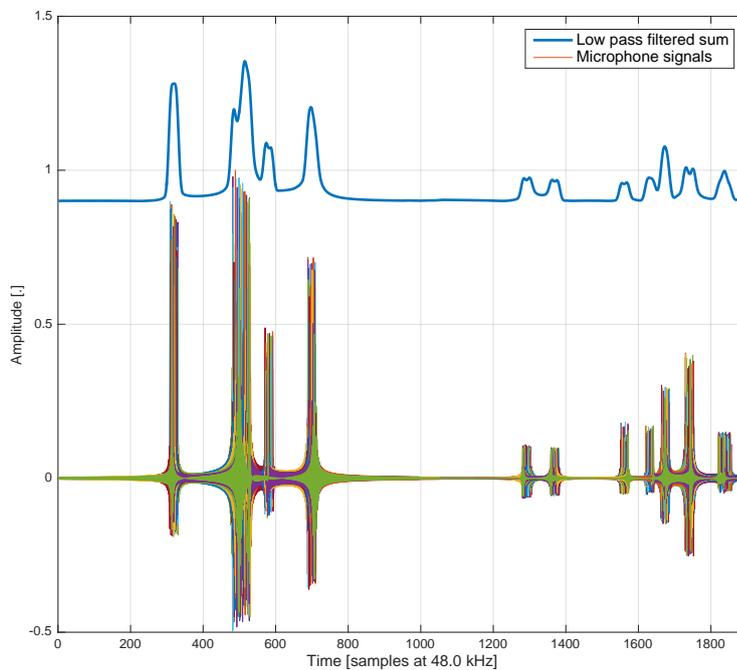


Figure 6.32: Impulse responses and estimated slope. Each microphone signal on the lower plot is presented with a different color.

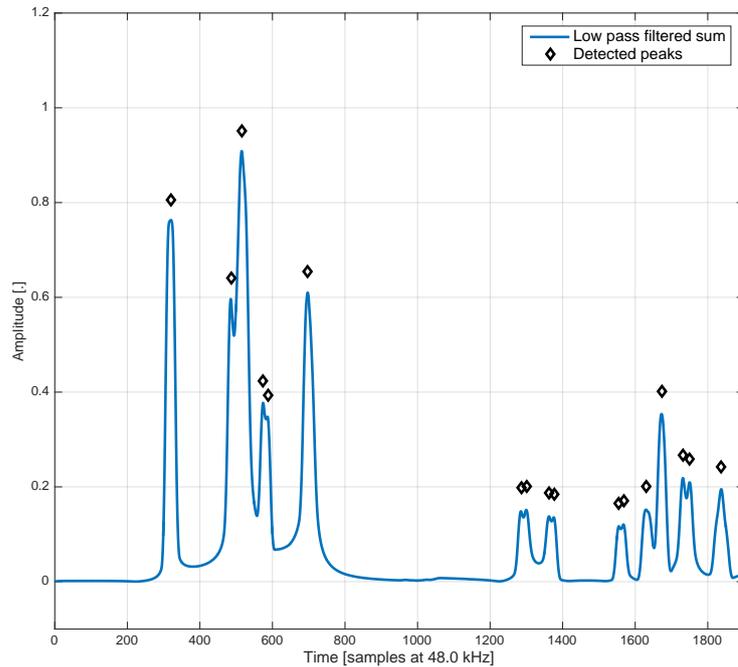


Figure 6.33: Detected peaks from the IR slope

Once the signals are summed, rectified and low pass filtered, the peaks can be found by detecting all local maxima above the noise threshold, which can be estimated from the pre-delay time (initial part of the IR before any peaks appear). A time window will be evaluated for each detected peak.

6.7.2 Estimation of the virtual source amplitude

The power of each found virtual source needs to be estimated in order to have sufficient data to calculate the frequency dependent boundaries' absorption. This problem can be approached in several manners. For the purposes of finding the enclosure boundaries, the power map will feature bolder traces when the boundaries are completely reflective. Hence, the power of each virtual source is established only as a function of the traveled distance (since completely reflective surfaces do not absorb any energy).

6.7.3 Convolution of window signals

The signals captured within a given time window contain peaks belonging to the reflections captured on that time interval and, as such, are broadband signals. In order to use a nar-

rowband algorithm for resolving the DOA, these signals are convolved with a sine. Since convolution is a linear operator, the correlation between signals remains unaltered, but the assumed signal conditions for narrowband algorithms are now satisfied. The frequency of the sine is selected to approximately reach the Nyquist limit: the separation between the closest elements on the array is equal to half the wavelength.

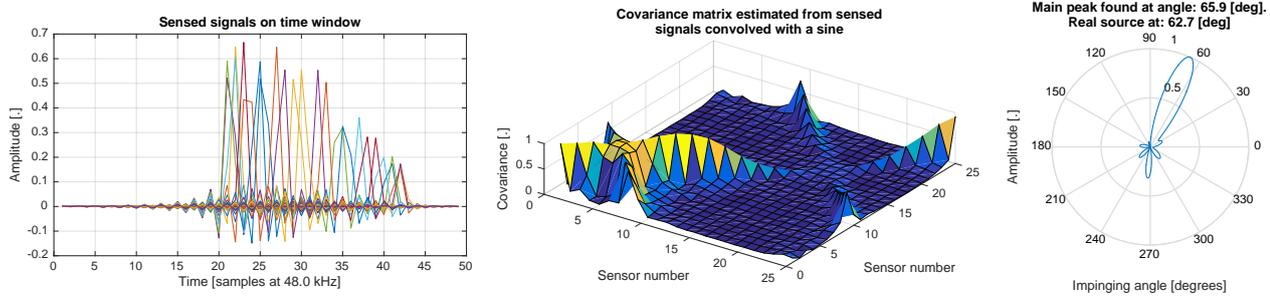


Figure 6.34: Estimating the DOA using the peaks contained on a time window of the IR.

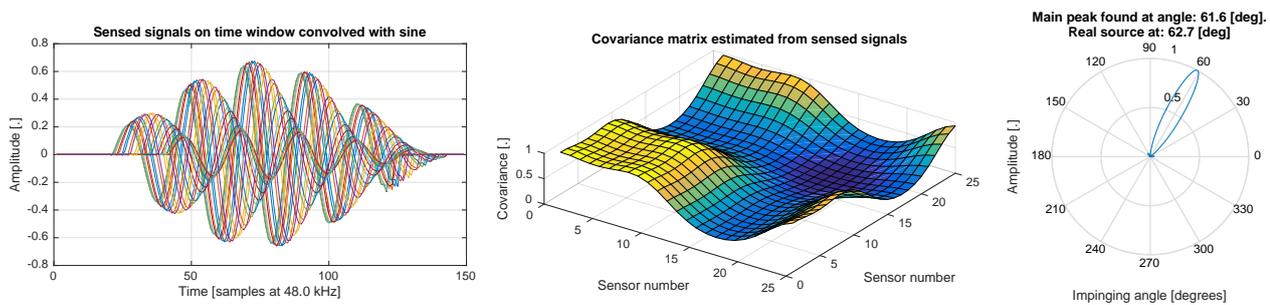


Figure 6.35: Estimating the DOA using the peaks contained on a time window of the IR after convolving them with a sine.

As the figures 6.34 and 6.35 show, the output power of the beamformer shows a reduction in the size of the side lobes, and the main peak is located closer to the actual value. The covariance matrix also changes and shows a softer grading than when using the sensed signals directly. The spectrum of the input signals change and most energy is now predominant on a single frequency below the Nyquist limit, as opposed to the very wide band of the original peaks.

6.7.4 Estimated cloud of virtual sources

Applying the procedure described to a set of IRs obtained through ray tracing, a cloud of virtual sources is obtained. A simple shoebox type room geometry was approximated by a two dimensional model of 8×6 [m].

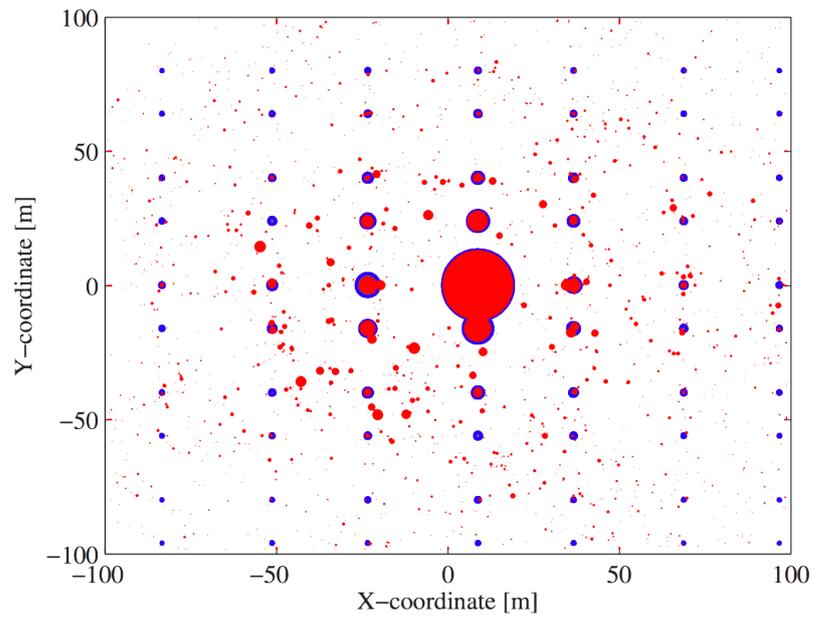


Figure 6.36: Comparison of correct and estimated clouds of virtual sources, from [5]. The authors used a sliding window with overlapping of 99% and a beamformer algorithm to decompose the IRs of a microphone array. This method seems very prone to false detections, as the cloud shows many incorrectly detected virtual sources (red dots) where should not exist (only blue dots are correct).

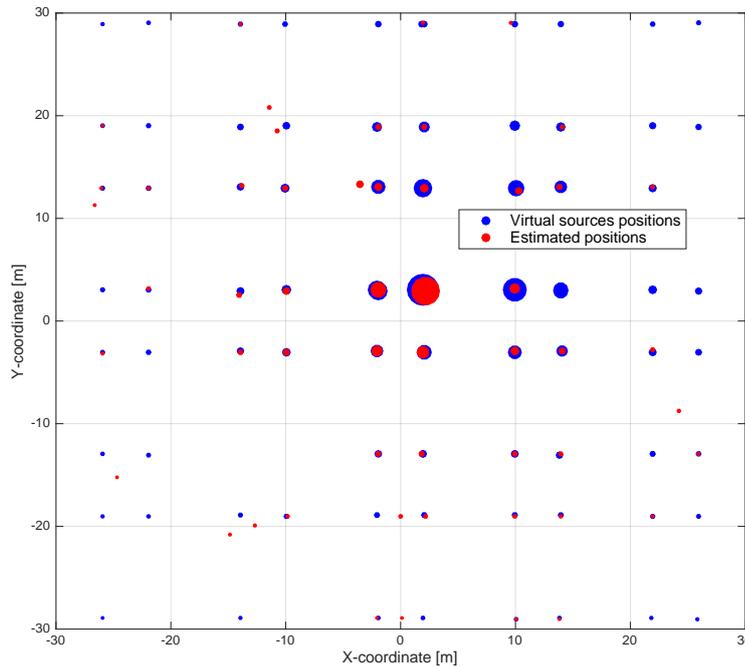


Figure 6.37: Comparison of correct and estimated clouds of virtual sources applying the new developed method. The false detection is virtually non-existent, while the precision remains unaltered. The new procedure also saves a lot of computations, since it only needs to be applied to the selected moments in time when an impinging wave is detected.

The plots 6.36 and 6.37 compare the cloud of virtual sources obtained by a state of the art method and the new developed method in this study. It shows that the new developed method performs significantly better, almost completely eliminating the false detections, since there is a previous condition for placing a new virtual source that does not exist in the other method (the detection of an impinging wave). Additionally, it greatly reduces the amount of time needed to evaluate the complete IRs, since the algorithms for estimating the virtual source position and spectrum / power only needs to be applied to those moments in time when it is deemed necessary (impinging signal is detected).

7 | Results processing

From the obtained cloud of virtual sources, following the previously described procedures, the geometry of the boundaries and their absorption characteristics can be estimated. In this chapter, the new developed techniques for these purposes are presented.

7.1 Geometry estimation procedure

From the estimated cloud of virtual sources, the geometry of the enclosure can be estimated. In this study a new procedure was developed that can obtain an image where the boundaries are shown.

From the general model of spherical propagation, it is known that, as the wave propagates, the pressure and velocity vary. In the case of a wave bouncing inside a space with limits, a stationary wave is formed—a limit being understood as a sudden change of acoustical impedance, such as a wall or a change in the section of a pipe—. When the boundary or limit is a rigid wall, the pressure will always have a maximum precisely at the wall, and the velocity a minimum. The pressure distribution in the rest of the enclosure will depend on the absorption of the boundaries and the frequency. Thus, difference frequencies will show different interference patterns within the enclosed space, but they will all have a pressure maximum right at the boundary. This property can be exploited to estimate the geometry of the boundaries if the pressure is known at all points for different frequencies. By averaging the pressure maps over many frequencies, the maxima and minima that appear inside the enclosure will smooth out, while the maxima at the boundaries will remain.



Figure 7.1: Simulation of sound pressure map for a simple source placed in front of a reflecting surface.

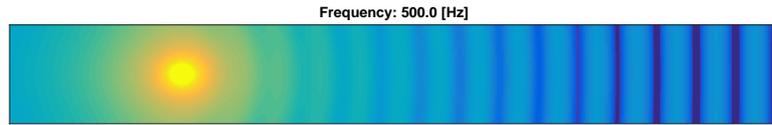


Figure 7.2: Simulation of sound pressure map for a simple source placed in front of a reflecting surface.

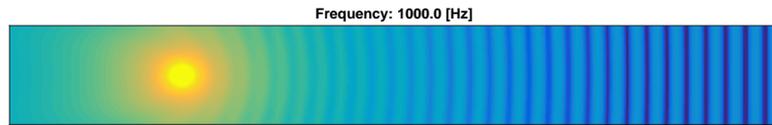


Figure 7.3: Simulation of sound pressure map for a simple source placed in front of a reflecting surface.



Figure 7.4: Simulation of sound pressure map for a simple source placed in front of a reflecting surface.

The figures 7.1, 7.2, 7.3 and 7.4 show the pressure distribution of a spherical wave that propagates and finds a perfectly reflective surface. The reflected wave interferes with the incident and the standing wave pattern appears with minima and maxima ([59], page 41). Those are found in different places depending on the frequency of excitation, but in all cases there exist a maxima where the reflective surface is located. Averaging over frequency, it is possible to obtain a geometrical place of maxima at that location.

When this procedure is applied to the cloud of virtual sources that emanates from a ray tracing analysis, the boundaries appear as brighter lines. Afterwards, the position of the boundaries could be detected, for example, using algorithms for detecting connected lines in images. This is out of the scope of this study, thus it is only suggested as a possible way.

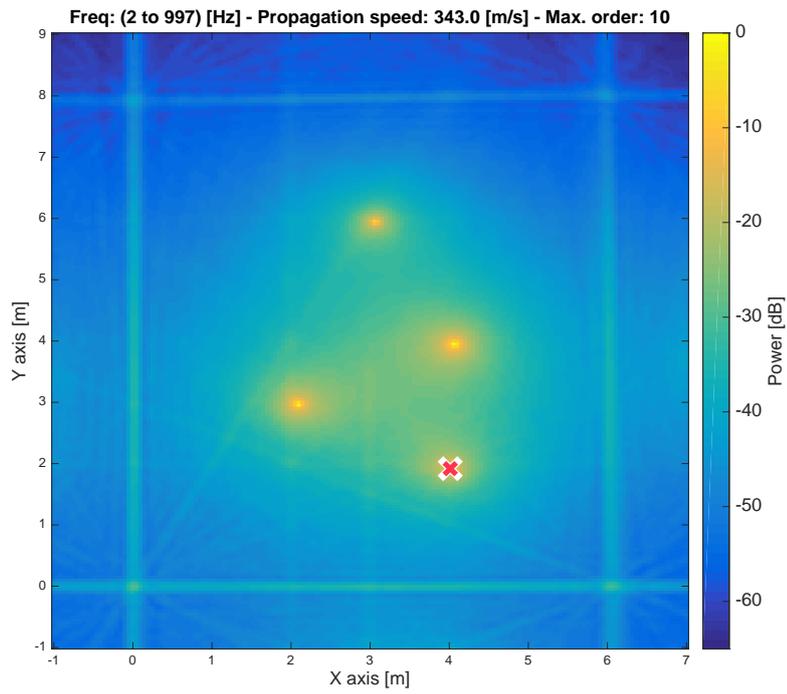


Figure 7.5: Reconstructed power map showing the boundaries location. Average over the range 2 to 997 Hz

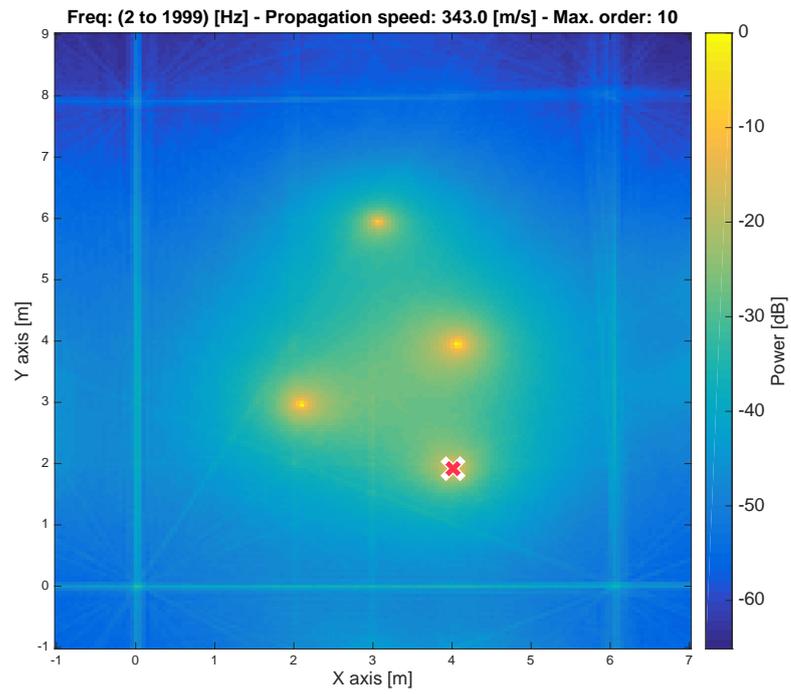


Figure 7.6: Reconstructed power map showing the boundaries location. Average over the range 2 to 1999 Hz

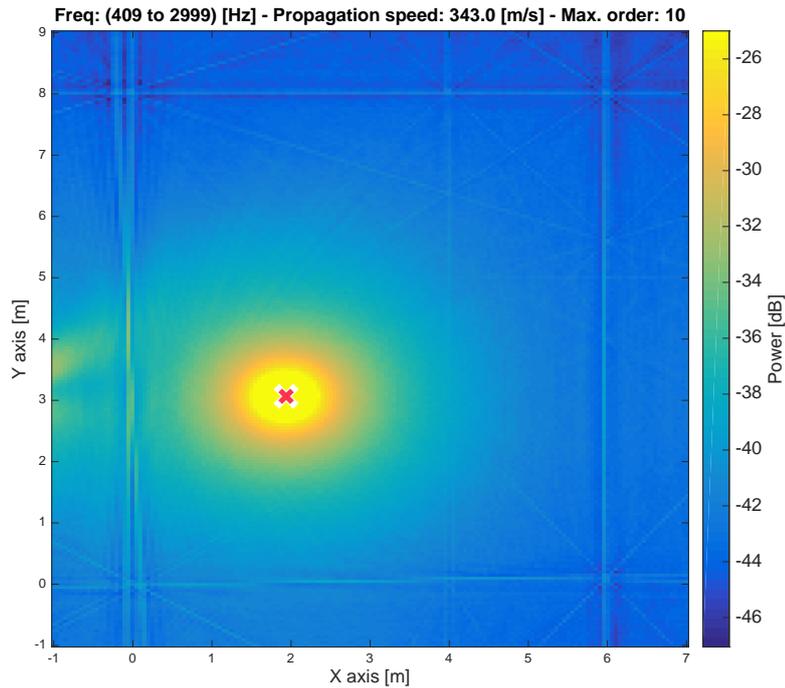


Figure 7.7: Reconstructed power map showing the boundaries location. Average over the range 409 to 2999 Hz

As figures 7.5, 7.6 and 7.7 show, the boundaries appear as brighter lines. Some artifacts also appear, that may increase the difficulty of estimating the geometry. As they are dependent on the source position (bright yellow spots), they smooth out if the same procedure is applied for different source positions and the results averaged.

The range of averaging frequencies also alters the resulting boundary lines obtained, thickening as the frequency decreases. If the range of frequencies is selected in the low range (Figure 7.5), the lines appear thicker, so it may be more difficult to pinpoint the boundaries. If, contrarily, the frequencies covers a higher range (Figure 7.6), the lines are sharper and the tolerance when pointing the boundaries is smaller, but at the same time defects or "ghost walls" may be more visible. The combination of the images obtained both ways can be beneficial, as the lower range of frequencies can provide an easier approximation to the boundaries placement (and rejecting the "ghost" information) while the high range will produce additional information that can further improve the first estimation based on the low frequency.

When performing the frequency average, the selection of frequencies included in the analysis is not a closed parameter. Since the target is to smooth out the standing wave pattern (see Figure 7.4), the process can be optimised if the selected frequencies do not produce overlapping peaks and valleys (that is, if they are not multiple of each other so that the wavelengths are not easily a linear combination of each other). In this regards, choosing the frequen-

cies from the prime numbers contained in the range of interest produces good results (see figures 7.5 and 7.6) while avoids extending unnecessary the computation time by including frequencies that do not help defining the boundaries.

Lastly, it is shown that the boundaries show more clearly when the source was placed close to it. It is important, however, to avoid placing the source too close to any boundary, since any position estimation error would greatly affect the overall result (as the relative angle to that boundary has a greater error when the source is very close, hence the boundary line may be tilted). See Figure 7.7.

7.2 Geometry estimation accuracy

The developed method has been tested for different room geometries, where the boundaries have been estimated and compared to those known. This intends to offer a general idea of the precision that can be obtained with the new developed method. It is important to note that these results are orientating, since the quality of the global solution will depend on several factors such as:

- Geometry and dimensions of enclosure.
- Absorption of boundaries (more absorbent materials difficult the estimation since reflections are weaker).
- Positioning of array and source.
- Averaging, moving the source to different positions.

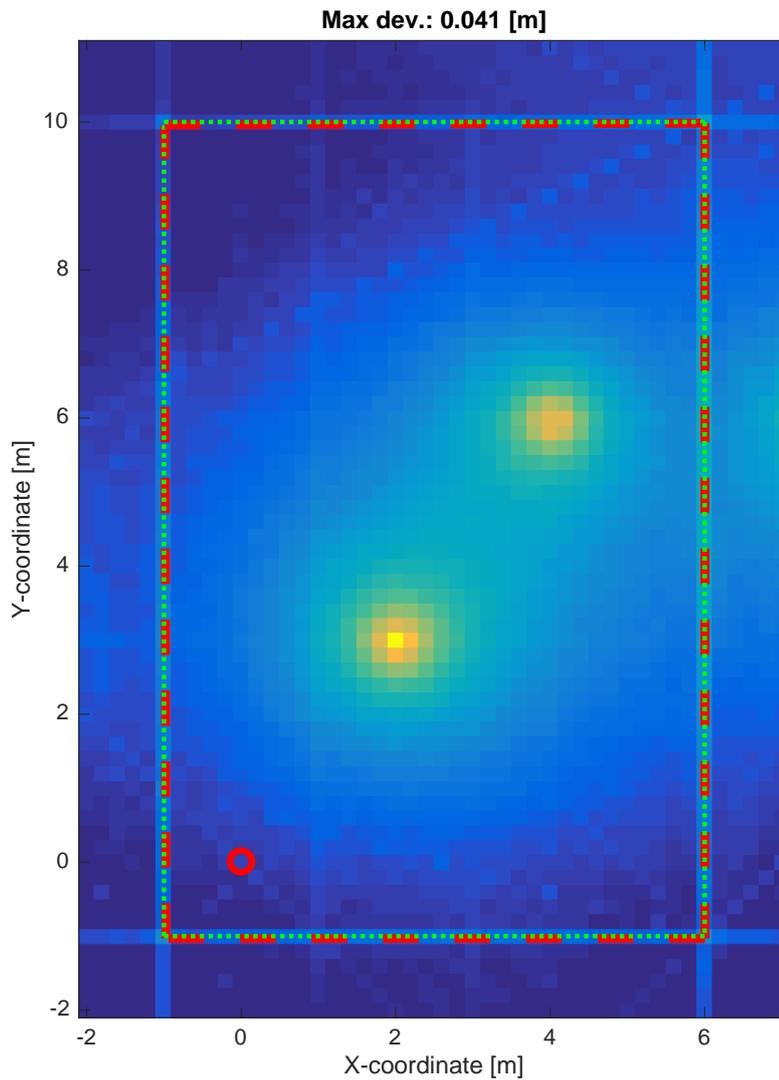


Figure 7.8: Example of estimated boundaries on a simple shoebox type room. The error in the estimated location of the boundaries is small (≈ 4 cm) compared to the room's dimensions. Red lines show the estimated boundaries' location, green dots the real ones.

As shown in Figure 7.8, using only two averages (two different source positions), the boundaries' geometry can be estimated with a maximum error of approximately 4 cm. If this value is compared to the dimensions of the room (6 x 10 m), it represents less than 1% of relative error. This can be considered a good precision in general terms, as it is usual within measurement equipment to consider tolerances of up to 1 or 2% in some cases.

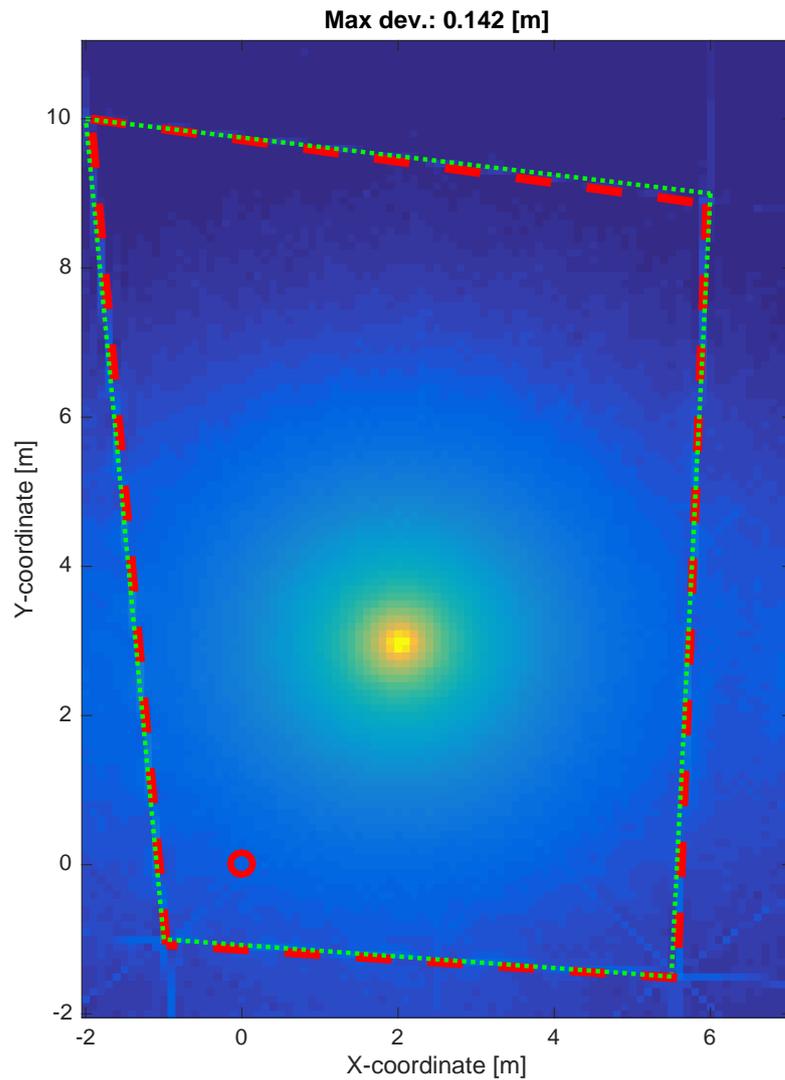


Figure 7.9: Example of estimated boundaries on a four walls room with an arbitrary geometry. The error grows in the estimated location of the boundaries. Red lines show the estimated boundaries' location, green dots the real ones.

To show a worse application case, the room geometry was made arbitrary (no walls are parallel this time) and only one source position was used (Figure 7.9). The estimated boundaries are more off the real value than before, having a maximum error of 14.2 cm, much larger than before, but still under 2% relative to the enclosure's dimensions.

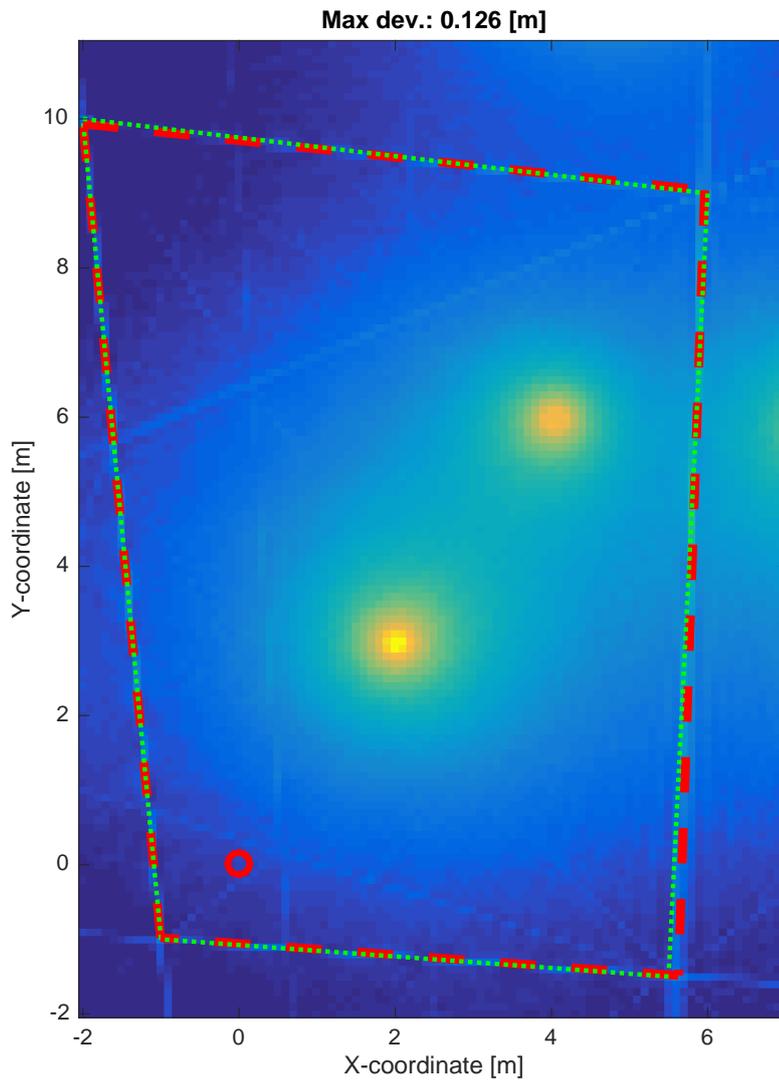


Figure 7.10: Example of estimated boundaries on a four walls room with an arbitrary geometry, using two source positions. The error is still bigger than for a simple shoebox geometry, but it is reduced due to the averaging. Red lines show the estimated boundaries' location, green dots the real ones.

If two positions are used in the irregular geometry (Figure 7.10), the error is decreased on 11.3% compared to when no averaging is used.

These results show that the developed procedure can be used satisfactorily to estimate an enclosure's geometry, with a precision that depends on several environment factors but that can be improved, at least, by means of averaging.

7.3 Finding the absorption characteristics of the boundaries

Estimating the absorption of the boundaries is not a trivial task, and many different ways of approaching the problem can be found. All will require to know the power spectral distribution of each virtual source. For this purpose, two possible ways are presented here:

- Compute the frequency content of the time windows when performing the IR analysis
- Steer the array using a beamforming technique and “scan” the placement of all found virtual sources

If the spectrum of all virtual sources is known, the boundaries’ frequency dependant absorption can be calculated, for instance by comparing the first order reflection with the direct signal.

The first method is easier to apply and requires much less computations. It simply requires a FFT per detected virtual source. It has the inconvenience that, when several sources are contained within the same time window, it may be very difficult to estimate both with low crosstalk. This will present a problem as the analysis approaches the end of the IR, since the chances of getting multiple reflections in a single window increase. It may still be valid for the early reflections, that are usually easier to separate on individual windows. These early reflections, as has been shown, contribute the most to the calculated power map within the enclosure, so the method can be considered appropriate.

The second method may be more precise and possibly capable of reducing the crosstalk between close sources, given that the beamformer has a good side lobe attenuation and no grating lobes exist (Nyquist criteria is satisfied). On the other hand it will have, presumably, a higher computational cost. The second method is described in Section 7.3.2

7.3.1 Boundary absorption estimation example

Following the first procedure, a simulation has been conducted with the purpose of demonstrating the feasibility of the method. As shown in Figure 7.11, a source is placed near a partially absorbent boundary, such that the microphone array is excited by the direct and reflected signals. The first is a delayed and attenuated copy of the source signal, due to the spherical propagation law (amplitude decreases proportionally with the distance). The second is a delayed, attenuated and frequency content modified copy, since the boundary introduces a frequency dependent absorption.

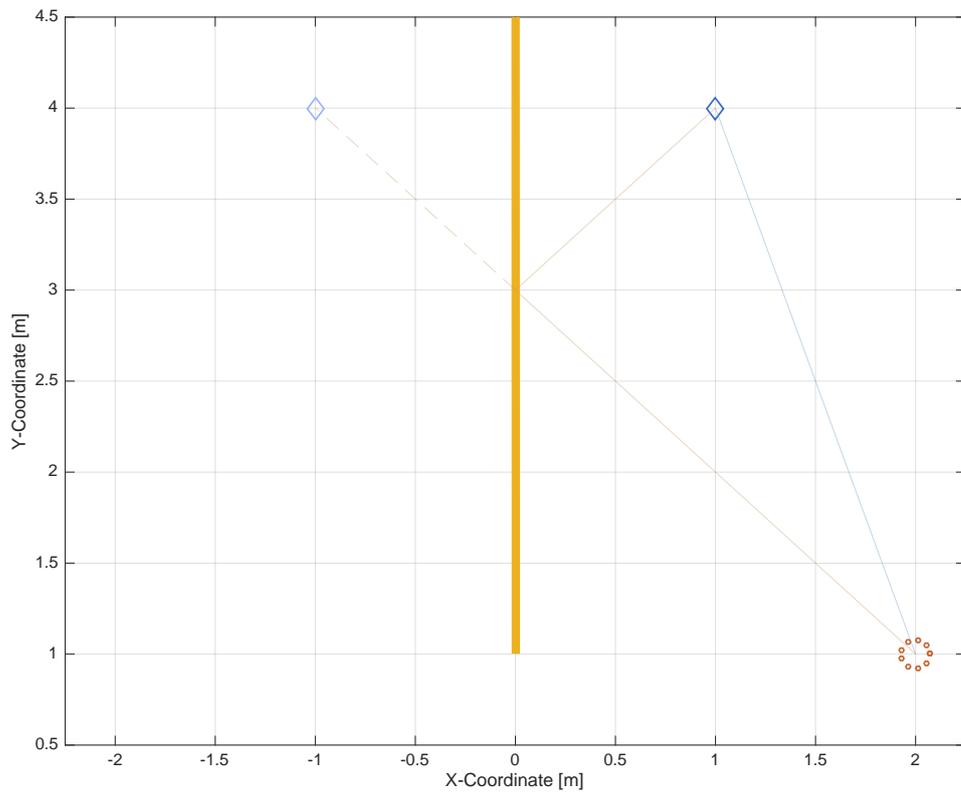


Figure 7.11: Setup of simulation for testing the absorption estimation method.

The mentioned setup provided a set of IRs, shown in Figure 7.12.

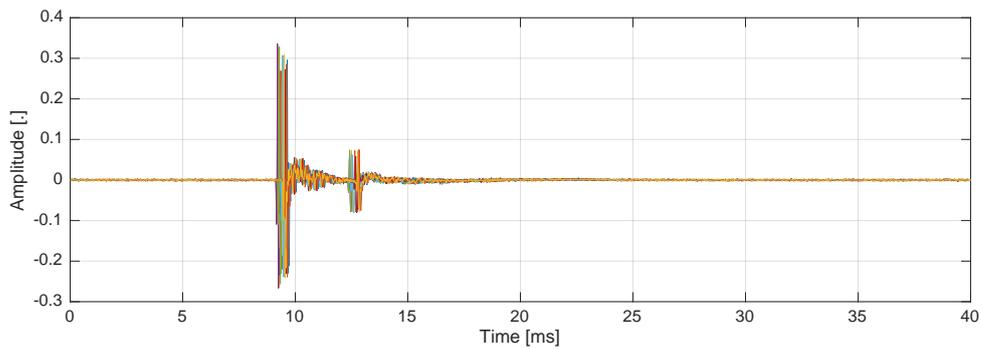


Figure 7.12: IRs obtained in simulation with a source radiating close to a boundary. The array captures two events very close in time. Each microphone signal is plotted in a different color.

Each event was detected with the developed procedure and a standard window of 1 ms was

applied on them. The content of the first window corresponds to the direct signal from the source, and the second to the reflected on the boundary (see Figure 7.13).

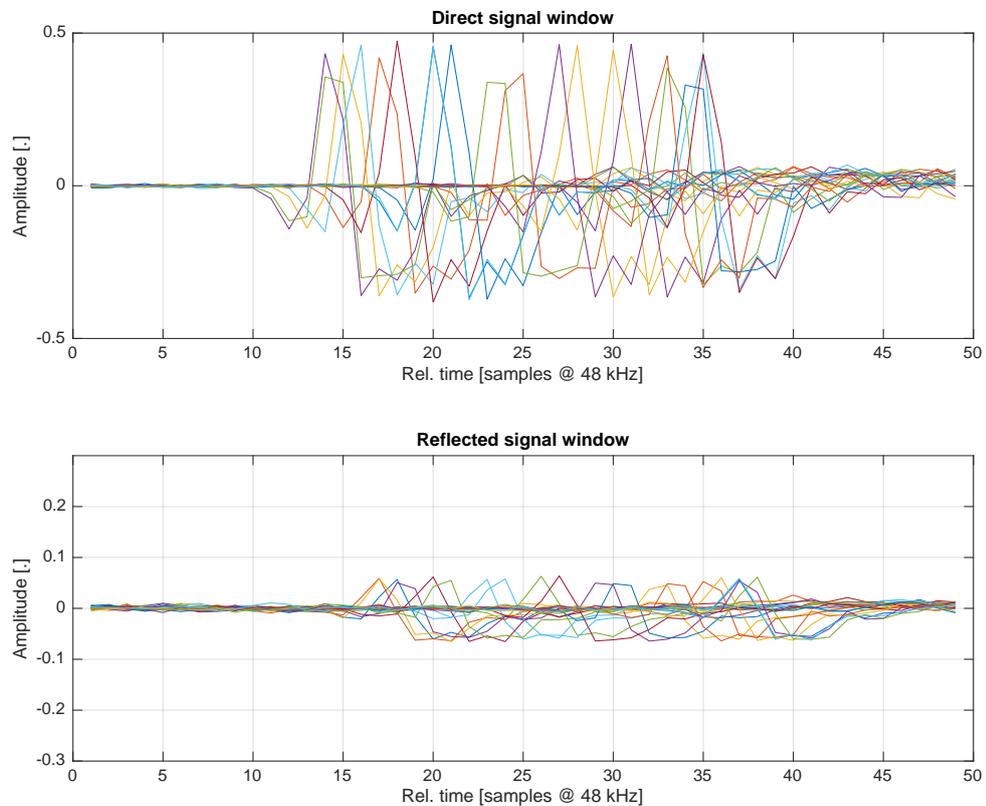


Figure 7.13: Content of time windows for direct and reflected windows. Windows of 1 ms.

Analysing each window in frequency, an estimation of the spectral distribution of power can be made. The spectrum from all microphones can be averaged to provide a better estimate (see Figure 7.14).

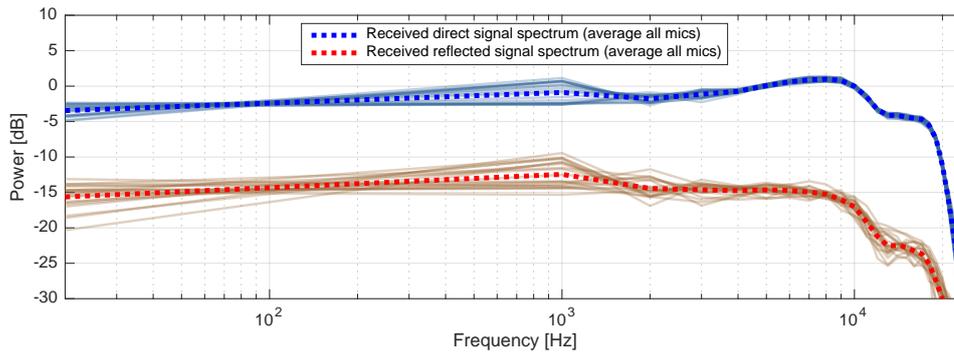


Figure 7.14: Spectra of received signals in direct / reflected time windows. Average trace for all microphones shown in bold. Time window of 1 ms.

These estimated spectra (direct and reflected) can then be compared to those known (in this case only, since it is a pure simulation) to assess the quality of this method. Figure 7.15 shows that the estimated spectra resembles well the original for mid and high frequency, but not for low frequency. This is due to the window being very short, as low frequency information from an impulse is usually found later in time (the tail for an IR).

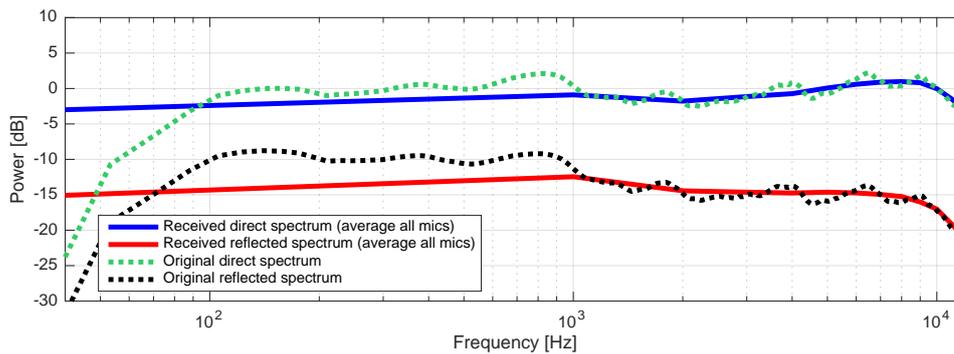


Figure 7.15: Comparison of known vs estimated spectra, corresponding to the direct and reflected signals respectively. Time window of 1 ms.

By subtracting the spectra, an estimation of the boundary absorption can be obtained. Compared to the known absorption characteristics of the boundary (figure 7.16), the estimation is valid for frequencies higher than 1 kHz. As mentioned before, for lower frequencies, the low frequency information is missing on such a very short time window. If the window size is increased, the boundary absorption characteristics are better estimated. This has the downside of possibly allowing other reflections to be also contained in the same time window. Therefore, it is a technique that may require a dynamic window size.

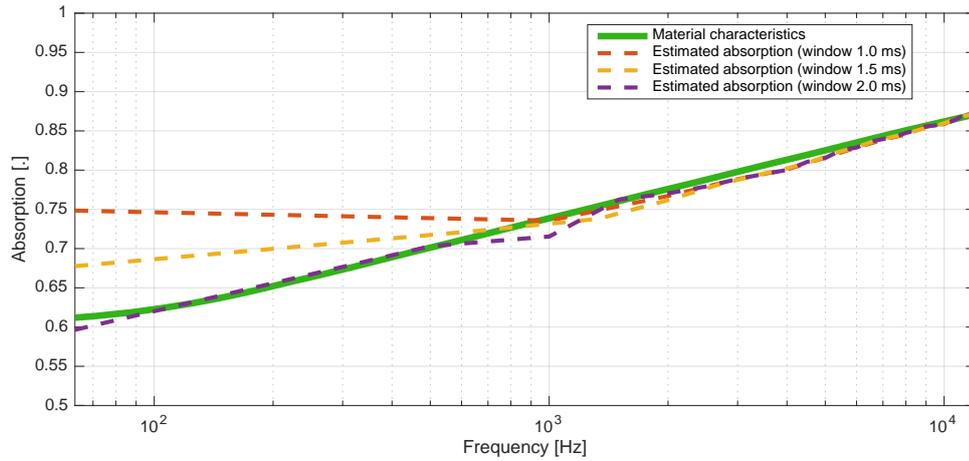


Figure 7.16: Comparison of material's absorption characteristics; known vs estimated from captured signals. Results for different window sizes.

7.3.2 Multiple sources spectrum estimation

Now that it is stated how it is done when only 1 source is present in the time window analysed, the following text will elaborate on how it is done when more than 1 source is present in the analyzed time window. Because the SML estimator is used, the spectrum estimation will also be based on the SML estimator. If the estimated DOA is sufficiently accurate, it is possible to obtain an estimate of the emitted signal ([30], page 52).

$$\hat{\mathbf{s}}(t) = \mathbf{A}^\dagger(\hat{\boldsymbol{\theta}})\mathbf{x}(t) \quad (7.1)$$

The literature does not state what a sufficiently accurate estimate is, so therefore a couple of tests are conducted. The tests will show the accuracy of estimating the signal(s) when the DOA estimation is worsened.

Test parameter	Value	Unit
Number of sources	2	N/A
Distance from source to array	20	m
Radius of array	7.5	cm
Number of microphones	20	N/A
SNR	80	dB

Table 7.1: This table holds the information that makes the basis of the following tests.

The setup stated in the table above is used throughout the following tests of the signal spectrum estimation. The signals emitted by the sources are not related to any room related scenario. They are used as an example to proof the concept.

For the case of more than 1 source being present there will be crosstalk present for inaccurate DOA estimations along with amplitude differences. Below are three figures where the source signals are a 5000 Hz and a 4000 Hz sine respectively. The distance from the sources to the array is 20 meters. The sources impinge on the array at 0 degrees and 30 degrees. The plots of the estimated signal spectra is compensated for the distance attenuation.

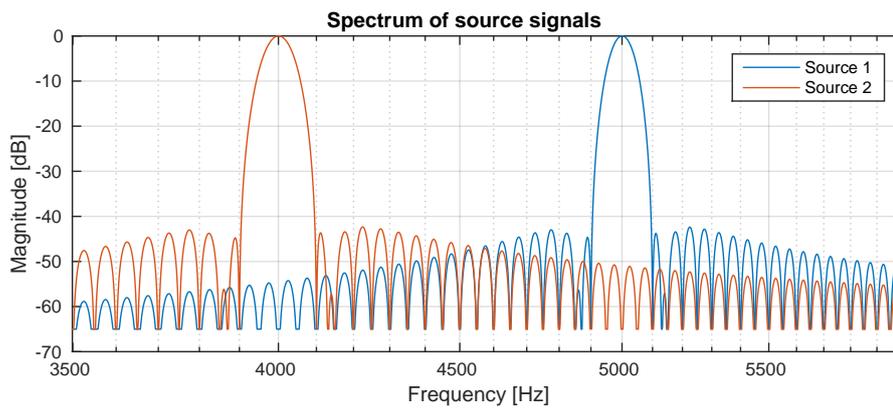


Figure 7.17: Here the spectrum for the 2 sources is shown. This is the spectrum that should be estimated.

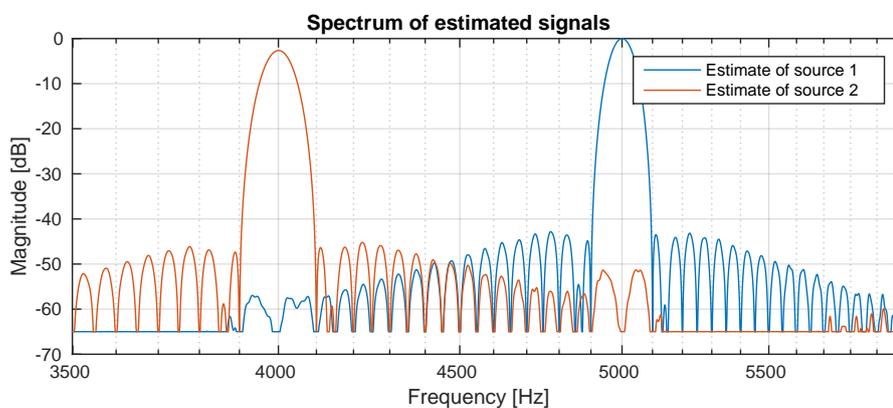


Figure 7.18: The figure shows the estimated spectrum when the DOA estimation error is set to zero.

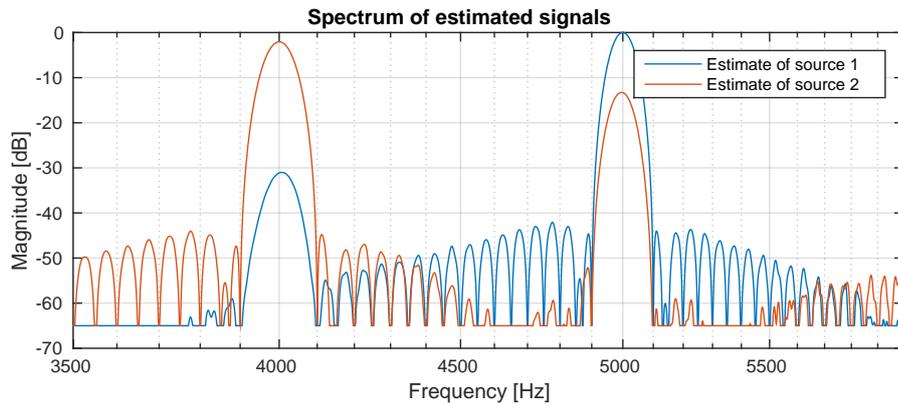


Figure 7.19: The figure shows the estimated spectrum when the DOA estimation error is set to 10 degrees.

It can be seen in Figure 7.19 that there is crosstalk present. The crosstalk is increased for worse DOA estimation and decreased for better DOA estimated. It is seen in Figure 7.18 that when the DOA estimate is correct then the crosstalk is decreased and the estimated signal spectrums are close estimates of the source signal spectrums.

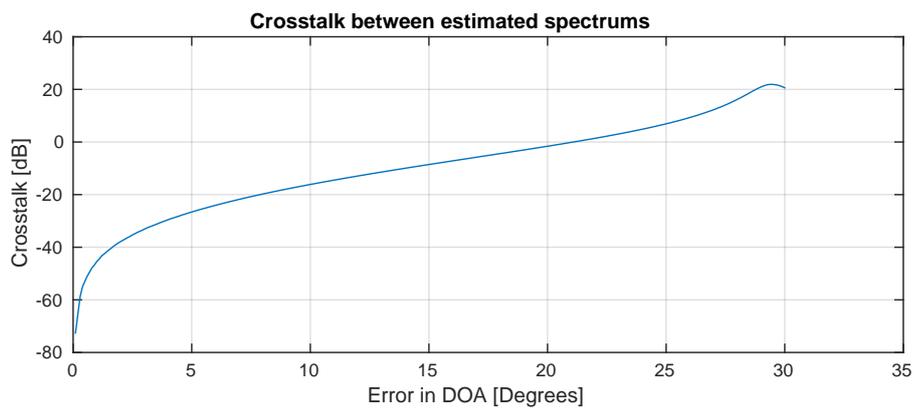


Figure 7.20: This plot shows that the crosstalk increases with the error in DOA estimation.

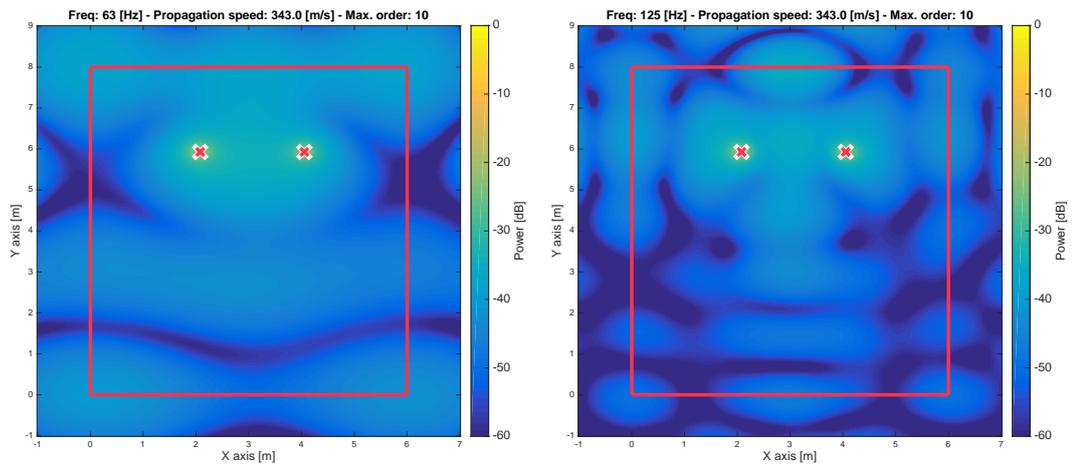
It can be seen in Figure 7.20 that the accuracy if the DOA estimate is indeed very important in order to ensure that the estimated signal resembles the source signal.

7.4 Estimation of sound field

Once the cloud of virtual sources has been estimated up to a certain order, the sound field within the enclosure in terms of sound pressure can be calculated on any point by using the omnidirectional point source model and adding the contribution of all virtual sources.

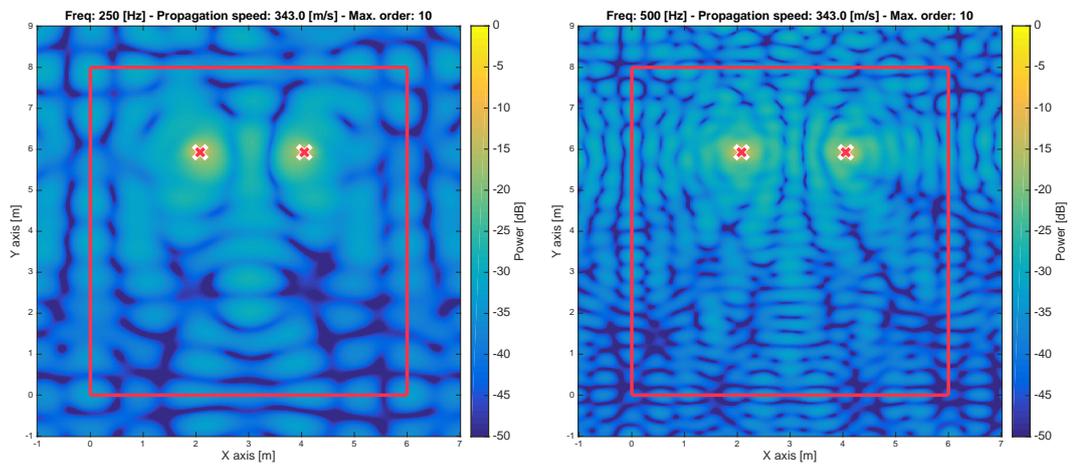
7.4.1 Example use cases

The cloud of virtual sources that represents the acoustics of an enclosure can be used for multiple purposes. For example, a simple and direct application would be for equalisation of the level on a listener position across bands of frequencies. This process typically involves placing a measurement microphone in the desired listening position to obtain the information used to introduce a proper compensation (whether it is an impulse response or the level in bands). Having the cloud of virtual sources, the room's behaviour can be predicted.



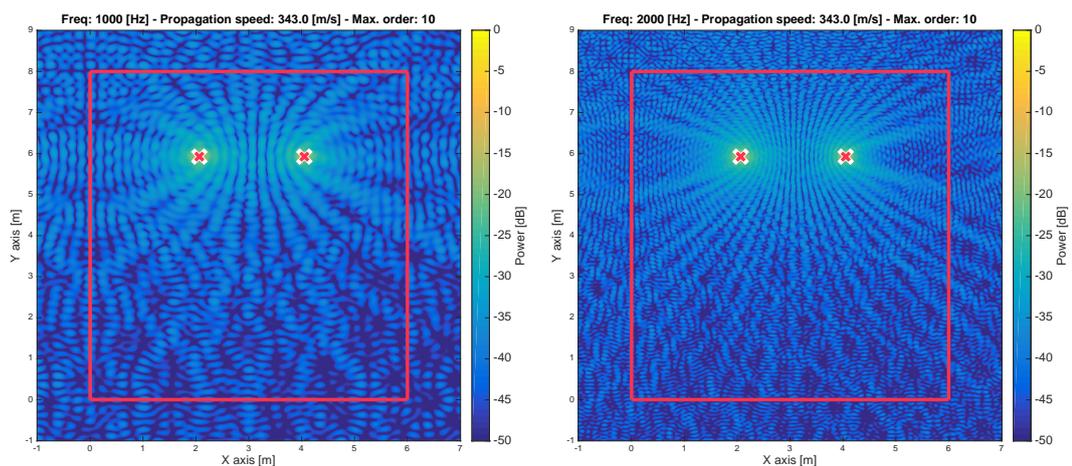
(a) Reconstructed sound pressure map for a stereo setup in a room. Frequency of 63 Hz.

(b) Reconstructed sound pressure map for a stereo setup in a room. Frequency of 125 Hz.



(c) Reconstructed sound pressure map for a stereo setup in a room. Frequency of 250 Hz.

(d) Reconstructed sound pressure map for a stereo setup in a room. Frequency of 500 Hz.



(e) Reconstructed sound pressure map for a stereo setup in a room. Frequency of 1000 Hz.

(f) Reconstructed sound pressure map for a stereo setup in a room. Frequency of 2000 Hz.

In the previous plots, a rectangular room was analysed by the complete system and the obtained cloud of virtual sources was used to recreate the sound pressure level in all points for some frequency bands.

Part IV

Closing

8 | Conclusion

The conclusion will summarize what was experienced during the development of this study. It will make a separate conclusion on each developed module and an overall conclusion.

This study tries to estimate the room geometry and absorption of the walls based on acoustic measurements. This is wanted i.e. a better and faster way of performing acoustic treatment to any room. The system is sought to tell the user about the geometry of the room and also about the absorption of the walls, making it easier to detect possible causes for a certain acoustic environment such as a reflective room or a dry room.

During the preliminary literature search it was found that the sound field inside a room is closely connected to the boundaries of said room. The idea was to exploit the fact that sound pressure builds up in front of walls and to use this knowledge to estimate the room geometry as an accurate way of obtaining information about the walls. It was therefore natural to investigate what a sound field is composed of and how it can be estimated. Here it was found that if the virtual sources up until a certain order are known, it is possible to estimate the sound field at a satisfactory precision.

The first step in this study was to create the ability to recreate the acoustics of a room. The room acoustics were based on the rooms geometry and the materials of the walls. With this information at hand the developed program is able to generate an impulse response in every point in the room. This is very useful when testing different positions for the source in the room and different configurations of the microphone array. This was done using ray tracing. After placing the source and the array, acoustic rays are fired in every direction at a predefined angle interval. The rays are then captured by the array if they intersect a predefined area surrounding the array. By this method, the impulse responses are captured. Because the simulation knows that distance each ray has travelled and the amount of times it has bounced off a wall (the order), it is possible to terminate based on maximum order or distance desired. This is for used comparison when the system later has to locate the virtual sources based on the impulse response captured.

After the simulation of the captured microphone signals were completed, the next step was to figure out how locating the virtual sources could be made with enough precision so that

the reconstructed sound field is a close estimate of the theoretical sound field. A method of estimating the direction of arrival and the distance of each reflection was wanted. Sensor array processing was found to be able to provide the necessary precision under the conditions of this study. The captured impulse responses are divided into small windows, based on where peaks are detected. These windows are then passed to a stochastic maximum likelihood estimator in order to approximate the DOA of each reflection contained in the window. The SML estimator was minimized using a genetic algorithm. Depending on the desired precision of the DOA estimate, the genetic algorithm is capable of estimating closely spaced coherent sources at an acceptable speed and with consistent estimates. The SML estimator proved to work very well under most scenarios such as closely spaced coherent signals and bad SNR conditions. Knowing the time delay from the IR and the DOA from the SML estimator it was possible to position the virtual sources with a high accuracy up to the needed order. The accuracy has proven to be good enough for estimating arbitrary polygonal rooms of low order.

In order to know how many signals the SML estimator should approximate, a method of detecting the amount of signals was needed. Here there were multiple candidates, but the most used one, the GLRT, was the most fitting to this case. It was shown to work well under simple conditions, but as soon as the scenario started to change the threshold estimation, needed for the GLRT, became very complex. Therefore another method was developed. This method consists of low-pass filtering each time window and then counting the peaks of each microphone signal and taking the mode of the peaks detected. This method has proven decent as long as the sources causing the peaks are separated enough in time and/or DOA. In a room it is very uncommon for 2 reflections to have the same time of arrival and DOA. In adverse circumstances the method proved to underestimate the amount of sources rather than overestimating them. This is a beneficial characteristic since it is worse to misplace a source than to not place it at all.

For the absorption of the walls to be estimated the procedure was to find the first order reflections' spectrums and then compare them to the spectrum of the direct sound (the source). For the case of one reflection present, it is done by computing the FFT of the received signal. In the case of two or more reflections present the method used is based on estimating the signal through beamforming using the DOAs estimated with the SML. If the estimate is accurate enough, it is possible to extract the signals of the time window.

A complete system was implemented in Matlab. This system is made as a proof of concept and it takes the user through multiple steps. First the user must declare the dimensions of the room and the absorption of the different walls along with the position of the array and the source. The system then computes the IRs for each microphone and estimates the room geometry and the absorption of the walls. The system is able to accurately estimate the geometry of the room, but the estimation of the absorption of walls needs some more research to verify the accuracy and increase the working frequency range. Preliminary results are obtained and they are promising. For a shoebox room, the maximum wall positioning error was approximately four centimeters. If that is compared to the size of the room (6 x 10 m),

then it represents an error of less than 1. For a more irregular four walled room the estimated geometry had a maximum error of 12.6 centimeters, equivalent to less than 2% of the room size.

9 | Further Research

This study was made over a period of four months by two students. Some of the modules have not been fully researched because of the time constraints. For this reason there will be presented a text that describes what work should be continued in the future.

In the study, a narrow band DOA estimator was used, and it proved to be a fair choice for locating the virtual sources. With the chosen DOA estimator it is possible to extract the signals from each virtual source given that the DOA estimate for the virtual sources is accurate. This was tested for virtual sources emitting sines (narrow band) and the tests showed precise estimations. However the estimation of the absorption of the boundaries should be done when for sources (and therefore virtual sources) emitting a broadband signal. Therefore, in a future research it should be looked into transforming and testing the system with a broadband signal estimator instead of a narrow band.

As the system is now, the user is presented with an estimate of the sound field in which the boundaries of the room can be seen. The user has to mark the position of the walls so that the absorption of the materials can be estimated. It is sought for an automated geometry recognition method. Ideas are towards representing the sound field in a matrix form and then using an algorithm to estimate where in the matrix a wall appears.

As it is now, the absorption estimation is made purely based on the first order reflections. A method is wanted through which it is possible to determine the absorption in a finer detail i.e. based on higher order reflections. The method should therefore be able to integrate the information of the higher order reflections.

As the system is made now, it provides a coarse estimation of the room geometry. Further testing of the performance of the system is wanted in order to see how it behaves with smaller objects in the room, like a table or a couch. More precision is wanted to be able to detect smaller objects in the room and analyse how they affect the room acoustics. It is also wanted to fully test the precision of the system by estimating numerous different geometries and calculating the error of the estimation.

For a better, more realistic estimation of room acoustics, the third dimension should be

looked into. In our case it relates to including the floor and the ceiling. The floor and the ceiling, which also contribute to the room acoustics i.e. in some cases it might be the ceiling that is responsible for unwanted acoustical characteristics of a room.

The last aspect to look into is to make the system work in real time. As for now the system spends approximately 5 minutes for a typical measurement with averaging, which may be inconvenient if the user has a limited amount of time to identify the acoustics of a room. If the complete system were able finish within a reasonable time frame, this would allow for a more dynamic inspection of a room where, for example, several tests were made in order to improve the acoustics of the room.

Bibliography

- [1] D. H. Søren Krarup Olesen, Milos Markovic, “Fast rendering of scanned room geometries,” *Joint Baltic-Nordic Acoustics Meeting 2014*, 2014.
- [2] H. Kuttruff, *Room Acoustics*. Spon Press, fifth ed., 2009.
- [3] P. L. Z. Maekawa, J.H. Rindel, *Environmental and Architectural Acoustics*. Spon Press, second ed., 2011.
- [4] R. R. Spors, S. M. and J. Ahrens, “The theory of wave field synthesis,” *AES 124th convention, Amsterdam*, may 2008.
- [5] A. K. Sakari Tervo, Jukka Patynen, “Spatial decomposition method for room impulse responses,” *Audio Engineering Society*, vol. Volume 61, jan 2013.
- [6] B. Gunel, “Room shape and size estimation using directional impulse response measurements.” Article, mar 2016.
- [7] I. Jager, “Room shape estimation from acoustic echoes,” master thesis, Delft University of Technology, 2015.
- [8] B&K, “Product data for omnipower sound source type 4292-1.” <http://www.bksv.com/Products/transducers/acoustic/sound-sources/omni-power-light-4292.aspx>
- [9] S. S. G. Gioda, “Back analysis procedures for the interpretation of field measurements in geomechanics,” *International Journal for Numerical and Analytical Methods in Geomechanics*, 1987.
- [10] T. T. Y. Kawai, “A numerical method for the calculation of transient acoustic scattering from thin rigid plates,” *Journal of Sound and Vibration*, 1990.
- [11] T. O. T. Sakuma, S. Sakamoto, *Computational Simulation in Architectural and Environmental Acoustic*. Springer, 2008. Chapter 2.
- [12] S. T. S. Siltanen, T. Lokki and L. Savioja, “Modeling incoherent reflections from rough room surfaces with image sources,” *Journal of Acoustic Society of America*, vol. 132, p. 4604–4614, jun 2012.

- [13] K. Watanabe, *Integral transform techniques for Green's function*. Springer, 2014. page 42.
- [14] F. B. Jensen, *Computational ocean acoustics*. Springer, 2011. Chapter
- [15] P. B. Don Davis, Eugene Patronis, *Sound System Engineering*. Focal Press, fourth ed., 2013.
- [16] M. Skålevik, "Schroeder frequency revisited," *Forum Acusticum*, 2011.
- [17] D. R. D. Pijush K. Kundu, Ira M. Cohen, *Fluid Mechanics*. Elsevier, fifth ed., 2012.
- [18] P. L. Z. Maekawa, *Environmental and Architectural Acoustics*. Spon Press, first ed., 1994.
- [19] A. R. F. Lawrence E. Kinsler, *Fundamentals of Acoustics*. Wiley, first ed., 1962.
- [20] H. Kuttruff, *Room Acoustics*. Elsevier, third ed., 1991.
- [21] "Attenuation of sound during propagation outdoors iso-9613-2," 1996.
- [22] J. Lamancusa, *Noise control*. Pennsylvania State University, 2009.
- [23] C. M. Harris, "Absorption of sound in air versus humidity and temperature," *Department of Electrical Engineering, Columbia University, New York*, feb 1966.
- [24] L. C. S. J. E. Piercy, T. F. W. Embleton, *Review of noise propagation in the atmosphere*. John Wiley & Sons Ltd, 1986.
- [25] H. E. Bass, "Kinetic description of sound absorption in the atmosphere," *Acoustical Society of America*, nov 1974.
- [26] E. J. Rickley, *Computing the Absorption of Sound by the Atmosphere and its Applicability to Aircraft Noise Certification*. Office of Environment and Energy Washington, 1998.
- [27] R. E. Crochiere, *Multirate Digital Signal Processing*. Prentice-Hall, 1983.
- [28] R. W. S. Alan V. Oppenheim, *Discrete Time Signal Processing*. Pearson, third ed., 2009.
- [29] U. Schumann, *Atmospheric Physics: background, methods and trends*. Springer, 2012.
- [30] Y. Y. Zhizhang Chen, Gopal Gokeda, *Introduction to Direction-of-arrival Estimation*. Artech House, 2010.
- [31] A. D. Stan Guy-Bart, Embrechts Jean-Jacques, "Comparison of different impulse response measurement techniques," *Sound and Image Department, University of Liege*, dec 2002.

- [32] P. N. K. John E. Piper, *Beamforming Narrowband and Broadband Signals, Sonar Systems*. InTech, 2011.
- [33] R. L. K. A. Satish, *Direction of arrival estimation and tracking of narrowband and wideband signals*. <http://docs.lib.purdue.edu/ecetr/112>: ECE Technical Reports, 1995.
- [34] D. F. D. E. B. Z. Z. Cha Zhang, Member, “Maximum likelihood sound source localization and beamforming for directional microphone arrays in distributed meetings,” *JOURNAL OF LATEX CLASS FILES*, vol. 6, jan 2007.
- [35] S. C. Y. Yoon, L. M. Kaplan, *Estimation of Wideband Signals*. Norwood, 2006.
- [36] W. L. J. S. Yong Rui, Dinei Florêncio, “Sound source localization for circular arrays of directional microphones,” *One Microsoft Way*, 2005.
- [37] M. V. Hamid Krim, “Two decades of array signal processing research,” *IEEE Signal Processing Magazine*, 1996.
- [38] J. S. Mati Wax, “Direction finding of coherent signals via spatial smoothing for uniform circular arrays,” *IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION*, vol. 42, 1994.
- [39] P. S. A. N. B. Ottersten, M. Viberg, *Radar Array processing*. Springer, 1993.
- [40] R. M. Petre Stoica, *Spectral analysis of signals*. PHI Learning, 2011.
- [41] I. Z. Mati Wax, “On unique localization of multiple sources passive sensor arrays,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 7, 1989.
- [42] L. D. Chambers, *The Practical Handbook of Genetic Algorithms Applications*. Chapman and Hall/CRC, second ed., 2000.
- [43] T. V. Mathew, *Genetic Algorithm*. Indian Institute of Technology Bombay, Department of Civil Engineering, Unknown.
- [44] Y. L. Minghui Li, “A refined genetic algorithm for accurate and reliable doa estimation with a sensor array,” *Wireless Personal Communications*, p. 533–547, 2007.
- [45] Y. L. Beng-Kiong Yeo, “Array failure correction using genetic algorithm,” *IEEE Transactions on Antennas and Propagation*, vol. 47, no. 5, 1999.
- [46] M. Melanie, *An Introduction to Genetic Algorithms*. A Bradford Book, 1998.
- [47] B. H. Minghui Li, Yilong Lu, “Array signal processing for maximum likelihood direction-of-arrival estimation,” *Journal of Electrical and Electronic Systems*, 2013.
- [48] A. P. Sakari Tervo, “Direction of arrival estimation of reflections from room impulse responses using a spherical microphone array,” *IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, vol. 23, oct 2015.

- [49] B. N. Noam Arkind, "Parametric joint detection-estimation of the number of sources in array processing," *IEEE Sensor Array and Multichannel Signal Processing Workshop*, 2010.
- [50] S. M. Kay, *Fundamentals of Statistical Signal Processing - Detection theory*, vol. 2. Prentice Hall PTR, 1998.
- [51] M. K. Clemens Hage, Tim Habigt, "Sparse doa estimation of wideband sound sources using circular harmonics," *Technische Universität München Fakultät für Elektro- und Informationstechnik*, mar 2014.
- [52] X. P. Fangzhou Wang, "Acoustic sources localization in 3d using multiple spherical arrays," *Journal of Electric Engineering Technology*, 2016.
- [53] R. D. Dmitry N. Zotkin, "Plane-wave decomposition of a sound scene using a cylindrical microphone array," *Perceptual Interfaces and Reality Lab Institute for Advanced Computer Studies (UMIACS), University of Maryland*, 2008.
- [54] B. P. J. J. L. Ana M. Torres, Maximo Cobos, "Robust acoustic source localization based on modal beamforming and time–frequency processing using circular microphone arrays," *Acoustical Society of America*, jul 2012.
- [55] H. S. M. Dessouky and Y. Albagory, "Efficient sidelobe reduction technique for small-sized concentric circular arrays," *Progress In Electromagnetics Research*, vol. 65, pp. 187–200, 2006.
- [56] I. C. Jacob Benesty, Jingdong Chen, *Design of circular differential microphone arrays*. Springer, 2015.
- [57] J. M. Ville Pulkki, "Spatial impulse response rendering ii: Reproduction of diffuse sound and listening tests," *Journal of Audio Engineering Society*, vol. 54, 2006.
- [58] H. Kuttruff, *Room Acoustics*. CRC Press, fourth ed., 2000.
- [59] A. B. C. J. V. S. Lawrence E. Kinsler, Austin R. Frey, *Fundamentals of Acoustics*. Wiley, fourth ed., 2000.

Part V

Appendix

A | Implementation of the DOA estimation

This chapter will showcase details about the implementation of the DOA algorithm that was developed for this thesis in MATLAB.

A.1 The steering matrix

In this section it is seen how the generation of the steering matrix was made.

```
for sourceIndex = 1 : NUMBER_OF_SOURCES
    for sensorIndex = 1 : amountOfSensors
        currentAmplitude = 1;
        currentAngle = chromosomePool(currentChromosome, sourceIndex);

        % Position of the current sensor for the new steeringMatrix input
        currentSensorCoordinates = sensorPositions(sensorIndex, :);

        % Of the point we're beamforming at (the radius given by the time and the angle
        % respect to the 0,0)
        beamformingPosition = rms(searchDistance) * [cos(currentAngle) sin(currentAngle)];
        currentDistanceVector = beamformingPosition - currentSensorCoordinates;

        lambda = propagationSpeed / detectedPeakFrequency;
        xsi = norm(currentDistanceVector) / lambda;
        steeringMatrix(sensorIndex, sourceIndex) = currentAmplitude * exp(-1i * 2 * pi * xsi);
    end
end
```

A.2 The SML cost function

Here it is shown how the SML cost function is implemented.

```
% -- Form the UML cost function and calculate the fitness

pseudoInversSteeringMatrix = pinv(steeringMatrix' * steeringMatrix) * steeringMatrix';
projectorOnSteeringMatrix = steeringMatrix * pseudoInversSteeringMatrix;
orthogonalProjectorOnSteeringMatrix = idMatrix - projectorOnSteeringMatrix;

% -- Estimate noise power
noisePower = 1 / (amountOfSensors - NUMBER_OF_SOURCES) ...
    * trace(orthogonalProjectorOnSteeringMatrix * covarianceMatrix);

% -- Estimate the signal covariance matrix
estimatedSignalCovarianceMatrix = pseudoInversSteeringMatrix ...
    * (covarianceMatrix - noisePower * idMatrix) * pseudoInversSteeringMatrix';

% -- Calculate the fitness for given generation
currentFitness = log( det( steeringMatrix * estimatedSignalCovarianceMatrix ...
    * steeringMatrix' + noisePower * idMatrix));
```

A.3 The genetic algorithm

This section will contain the essential code for the genetic algorithm.

A.3.1 Initialization of the population

Here it is shown how the initialization is made:

```
% -- Intelligent initialization
searchAngle = [-180 : 1 : 180]/180*pi;
for angleIndex = 1 : length(searchAngle)
    for sensorIndex = 1 : amountOfSensors
        currentAmplitude = 1;
        currentAngle = chromosomePool(currentChromosome, sourceIndex);

        % Position of the current sensor for the new steeringMatrix input
        currentSensorCoordinates = sensorPositions(sensorIndex, :);

        % Of the point we're beamforming at (the radius given by the time and the angle
        % respect to the 0,0)
        beamformingPosition = rms(searchDistance) * ...
            [cos(currentAngle) sin(currentAngle)];
```

```

    currentDistanceVector = beamformingPosition - ...
    currentSensorCoordinates;

    lambda = propagationSpeed / detectedPeakFrequency;
    xsi = norm(currentDistanceVector) / lambda;
    steeringMatrix(sensorIndex, sourceIndex) = currentAmplitude ...
    * exp(-1i * 2 * pi * xsi);
end
bartlett(angleIndex,1) = abs(steeringMatrix' * covarianceMatrix ...
* steeringMatrix) / (norm(steeringMatrix)^2);
end

bartlett = bartlett/max(bartlett);

% Find peaks above 50% percent
peakAboveThresholdCounter = 1;
peakAngles = [];
[peakValue, peakPos] = findpeaks(bartlett, searchAngle);
for angleIndex = 1 : length(peakPos)
    if peakValue(angleIndex) > 0.5
        peakAngles(peakAboveThresholdCounter) = peakPos(angleIndex);
        peakAboveThresholdCounter = peakAboveThresholdCounter + 1;
    end
end

if isempty(peakAngles)
    peakAngles(peakAboveThresholdCounter) = pi + (-pi-pi).*rand(1,NUMBER_OF_SOURCES);
    disp('No peaks above 50%. Random start activated!')
end
disp(['Peaks detected at: ' num2str(peakAngles*180/pi)])

if length(peakAngles) > NUMBER_OF_SOURCES
    peakAngles = peakAngles(1:NUMBER_OF_SOURCES);
    disp('More peaks found than number of sources stated!')
end

% -- Create population in the proximity of the peaks
proximityRange = 15*pi/180;
chromosomePool = zeros(POPULATION, NUMBER_OF_SOURCES);
for geneIndex = 1 : length(peakAngles)
    chromosomePool(:,geneIndex) = linspace(peakAngles(geneIndex) ...
    - proximityRange, peakAngles(geneIndex) + proximityRange, POPULATION);
end

```

A.3.2 Rank and sort

Here it is shown how the sorting is done.

```
[sortedFitnessEvaluation, indexMatrix] = sort(fitnessEvaluation,1,'ascend');
```

```

%Now rearrange the chromosomePool
for i = 1 : size(fitnessEvaluation,1)
    sortedChromosomePool(i,:) = chromosomePool(indexMatrix(i),:);
end

```

A.3.3 Parent selection

```

if EMS
    FITTEST_PARENTS = 2;
    % Choose randomly between adjacent pairs
    k = 0;
    for i = 1 : size(parentPool,1) - 1
        choosePair = randi([1,2],1);
        parentPool(i + 1,:) = sortedChromosomePool(k + choosePair,:);
        k = k + 2;
    end
    % Make sure the #FITTEST_PARENTS is moved to parentPool
    parentPool(1:FITTEST_PARENTS,:) = sortedChromosomePool(1:FITTEST_PARENTS,:);
end

```

A.3.4 Copy pool

Here it is shown how a copy of the chromosome pool is made and mutated.

```

amountOfChange = zeros(1,size(sortedChromosomePool,2));
PROBABILITY_MUTATION = 0.6;
% -- Form the pools
pool = sortedChromosomePool;

% -- Mutate the pools
for geneIndex = 1 : size(pool,2)
    for chromosomeIndex = 1 : size(pool,1)
        if rand(1) < PROBABILITY_MUTATION
            for sourceIndex = 1 : size(sortedChromosomePool,2)
                amountOfChange(sourceIndex) = ...
                    1/(1 + var(sortedChromosomePool(:,sourceIndex))) * pi;
            end
            if randi([1 2]) == 1
                signOfChange = 1;
            else
                signOfChange = -1;
            end
            pool(chromosomeIndex, geneIndex) = ...
                sortedChromosomePool(chromosomeIndex, geneIndex) ...
                + amountOfChange(geneIndex) * signOfChange;
        else
            pool(chromosomeIndex, geneIndex) = pool(chromosomeIndex, geneIndex);
        end
    end
end

```

```

        end
    end
end

```

A.3.5 Best chromosome mutation

Here it is shown how the best chromosome is mutated.

```

amountOfChromosomes = 1;
bestFit = sortedChromosomePool(1:amountOfChromosomes,:);
for chromosomeIndex = 1 : amountOfChromosomes
    for geneIndex = 1 : size(sortedChromosomePool,2)
        if randi([1 2]) == 1
            signOfChange = 1;
        else
            signOfChange = -1;
        end

        amountOfChange = rand(1)/20 * pi / 180;

        bestFit(chromosomeIndex, geneIndex) = bestFit(chromosomeIndex, geneIndex) ...
            + amountOfChange * signOfChange;
    end
end
copiedPoolFull(4: 3 + amountOfChromosomes,:) = bestFit;
noisedCopy = copiedPoolFull;

```

A.3.6 Crossover

Here it is seen how the mating scheme is made. Two schemes can be chosen.

```

%Perform crossover
CROSS_OVER_POINT = ceil(size(parentPool,2)/2);
if EMS
    emperor = parentPool(1,:);
    k = 1;
    for i = 1 : size(parentPool,1)/2
        if size(parentPool,2) < 2
            child1 = emperor;
            child2 = parentPool(i+1);
        else
            child1 = [emperor(1:CROSS_OVER_POINT) ...
                parentPool(i + 1, CROSS_OVER_POINT + 1:size(parentPool,2))];
            child2 = [parentPool(i + 1, 1:CROSS_OVER_POINT) ...
                emperor(CROSS_OVER_POINT + 1:size(emperor,2))];
        end
    end
end

```

```

        childrenPool(k,:) = child1;
        childrenPool(k + 1,:) = child2;
        k = k + 2;
    end
end

if BMW
    k = 1;
    fatCounter = 0;
    for chromosomeIndex = 1 : size(parentPool,1)/2
        currentParentFit = parentPool(chromosomeIndex,:);
        currentParentFat = parentPool(size(parentPool,1) - fatCounter,:);
        if size(parentPool,2) < 2
            child1 = currentParentFit;
            child2 = currentParentFat;
        else
            child1 = [currentParentFit(1:CROSS_OVER_POINT) ...
                    currentParentFat(CROSS_OVER_POINT + 1 : size(currentParentFat,2))];
            child2 = [currentParentFat(1:CROSS_OVER_POINT) ...
                    currentParentFit(CROSS_OVER_POINT + 1 : size(currentParentFit,2))];
        end
        childrenPool(k,:) = child1;
        childrenPool(k + 1,:) = child2;
        k = k + 2;
        fatCounter = fatCounter + 1;
    end
end
end

```

A.3.7 Mutation

Here it is seen how the mutation is done.

```

PROBABILITY_MUTATION = 0.05;
amountOfChange = pi * 0.5/180;
childrenPoolMutated = childrenPool;
% -- Mutate the childrenPool
for i = 1 : size(childrenPool,1)
    if rand(1) < PROBABILITY_MUTATION
        chromosomeChooser = i;
        geneChooser = randi(size(childrenPool,2),1);
        childrenPoolMutated(chromosomeChooser,geneChooser) = ...
            childrenPool(chromosomeChooser,geneChooser) + ...
            (-amountOfChange + (amountOfChange+amountOfChange).*rand(1));
    else
        childrenPoolMutated = childrenPool;
    end
end
end

```



```

last_touched_boundary = [];
while 1

    % Check if we reached the microphone with this ray
    [mic_reached, collision_point, ray_points] = reached_position(ray_origin, ...
        ray_direction, mic_position, max_radius, mic_collision_error_margin);
    if mic_reached
        current_track_bounces_count = current_track_bounces_count + 1;
        current_track(current_track_bounces_count, 1) = sqrt((ray_origin(1) - ...
            collision_point(1))^2 + (ray_origin(2) - collision_point(2))^2);
        current_track(current_track_bounces_count, 2 : size(current_track, 2)) =
            ones(1, size(current_track, 2)-1);
    else
        % Otherwise detect if it collides with a wall (it should)
        closest_collision_point = [inf inf];
        for current_boundary_index = [1 : size(room_boundaries, 1)]
            current_boundary = room_boundaries(current_boundary_index, :);
            current_boundary = [current_boundary(1) current_boundary(2); ...
                current_boundary(3) current_boundary(4)];

            % Check that the reached boundary is not the one
            % it's actually on right now
            if ~isequal(last_touched_boundary, current_boundary)

                % Check if collision occurs with current boundary
                [current_boundary_collision, test_collision_point, ...
                    test_reflection_direction, test_trazed_ray] = ...
                    detect_collision(ray_origin, ...
                        ray_direction, current_boundary, ...
                        max_radius, ...
                        boundary_collision_error_margin);
                if current_boundary_collision

                    wall_collision = 1;
                    % This code is to make sure we pick up the
                    % shortest collision point (such that we dont
                    % cross walls)
                    last_dist = sqrt( (ray_origin(1) - ...
                        closest_collision_point(1))^2 + ...
                        (ray_origin(2) - ...
                        closest_collision_point(2))^2 );
                    curr_dist = sqrt( (ray_origin(1) - ...
                        test_collision_point(1))^2 + ...
                        (ray_origin(2) - ...
                        test_collision_point(2))^2 );
                    if curr_dist < last_dist
                        collision_point = ...
                            test_collision_point;
                        closest_collision_point = ...
                            test_collision_point;
                        reflection_direction = ...
                            test_reflection_direction;
                        closest_boundary = current_boundary;
                    end
                end
            end
        end
    end
end

```

```

        boundary_index_closest_reflection = ...
            current_boundary_index;

        % Used for debug plot:
        trazed_ray = test_trazed_ray;
    end
end
end
end
if wall_collision
    amount_of_collisions = amount_of_collisions + 1;
    last_touched_boundary = closest_boundary;

    current_track_bounces_count = current_track_bounces_count + 1;
    current_track(current_track_bounces_count, 1) = ...
        sqrt((ray_origin(1) - collision_point(1))^2 + ...
            (ray_origin(2) - collision_point(2))^2);
    current_track(current_track_bounces_count, 2 : size(current_track, 2)) = ...
        boundaries_reflection_coefficients(boundary_index_closest_reflection, :);
end

end

% Plot the new segment: from ray_origin -> collision_point
if (mic_reached || wall_collision) && show_animated_plot
    % Use this other plot insted for debug:
    plot([ray_origin(1) collision_point(1)], ...
        [ray_origin(2) collision_point(2)], '-');
    title(sprintf('Bounces: %d', current_track_bounces_count))
    pause(animated_plot_pause_duration)
end

if wall_collision
    % Update the ray_origin, ray_direction
    ray_origin = collision_point;
    ray_direction = reflection_direction;
end

% Finish current ray if we reached the mic
if mic_reached
    % Discard the real source if told so
    if include_real_source || (amount_of_collisions > 0)

        % Traveled distance for the delay on time
        traveled_distance = ...
            sum(current_track(1 : current_track_bounces_count, 1));
        propagation_delay = ...
            traveled_distance / propagation_speed;

        % Calculate position of current virtual source
        image_souce_position = (traveled_distance * ...
            [cos(ray_direction + pi) sin(ray_direction + pi)]) + ...

```

```

mic_position;
image_sources_count = image_sources_count + 1;

% Checks that the new virtual source is separated
% enough from all other virtual found sources
if (min(sqrt( (image_souce_position(1) - ...
image_sources_list(1 : image_sources_count, 1)).^2 + ...
(image_souce_position(2) - ...
image_sources_list(1 : image_sources_count, 2)).^2    )) ...
> sources_minimum_distance)
    % -- Add the new data to the reflectogram --

    % Attenuation from the spherical law
    attenuation_spherical_law = 1 / traveled_distance;

    % Generate delta on time
    fractional_delta = pad_with_zeros( ...
        place_fractional_delta(propagation_delay * sample_rate), ...
        length(reflectogram));

    % Correct amplitude with spherical law
    fractional_delta = fractional_delta * attenuation_spherical_law;

    % Attenuation (in 1/3 oct bands) from absorption on boundaries
    attenuation_boundaries_absorption = ...
    prod(current_track(1 : current_track_bounces_count, ...
    2 : size(current_track, 2)), 1);

    % Apply the corresponding filter to the generated delta
    [fractional_delta, filter_design_introduced_delay] = ...
    arbitrary_filter_design(fractional_delta, ...
    boundaries_reflection_frequencies, ...
    attenuation_boundaries_absorption);

    % Correct for the air absorption attenuation
    [fractional_delta, air_absorption_introduced_delay] = ...
    correct_air_absorption(fractional_delta, traveled_distance);
    fractional_delta = ...
    fractional_delta(air_absorption_introduced_delay + ...
    filter_design_introduced_delay : length(reflectogram) + ...
    air_absorption_introduced_delay - 1);

    if size(reflectogram) ~= size(fractional_delta)
        disp('ERROR: unexpected size unmatched!')
    end

    % Add the new ray to the reflectogram
    reflectogram = reflectogram + ...
    pad_with_zeros(fractional_delta, length(reflectogram));

    % Save new image source position and amplitude
    image_sources_list(image_sources_count, 1 : 4) = ...

```

```

[image_souce_position ...
attenuation_spherical_law ...
amount_of_collisions];

% Save the IR associated to the new virtual source
image_sources_list(image_sources_count, ...
5 : size(image_sources_list, 2)) = ...
pad_with_zeros(arbitrary_filter_design([1 ...
zeros(1, length(fractional_delta)-1)], ...
boundaries_reflection_frequencies, ...
attenuation_boundaries_absorption), ...
length(reflectogram));

if show_animated_plot
    plot(image_souce_position(1), image_souce_position(2), 'x')
    plot([image_souce_position(1) mic_position(1)], ...
        [image_souce_position(2) mic_position(2)]);
    pause
end
else
    % Skipping repeated virtual source
    image_sources_count = image_sources_count -1;
end
end

% Skip one degree after a collision
if skip_after_succesful_ray
    initial_ray_angle_index = initial_ray_angle_index + ...
        ceil(1 / scan_angle_resolution);
end

break
end

% Finish current ray because max order was reached
if amount_of_collisions >= max_reflection_order
    break
end

% Sanity check, boundaries should be closed ideally
if ~mic_reached && ~wall_collision
    disp('** Unexpected. No collision and didnt reach the mic ... It ...
        could mean that there''s a hole in the boundaries ...');
    disp(sprintf('Throwing ray on direction: -000.000 [deg] ...'));
    break
end
end
end
end

```


C | Implementation of the detection

This chapter will showcase details about the implementation of the detection algorithm that was developed for this thesis in MATLAB.

```
function detetedAmount = detectAmountOfSignals(inputSignals)

    % Compute a low pass average of all IRs
    windowPeaksCount = zeros(size(inputSignals, 1), 1);
    for index = 1 : size(inputSignals, 1)

        % Estimate the slope
        filteredIR = averageIRShortly(inputSignals(index, :));

        % Normalize and vertical stretch
        filteredIR = ...
            (filteredIR - min(filteredIR)) / ...
            max(filteredIR - min(filteredIR));

        % Find the peaks in the average
        [tempIRPeaksValues, tempIRPeaksPlaces] = findpeaks(filteredIR);
        tempIRPeaksValues = tempIRPeaksValues / max(tempIRPeaksValues);

        tempPeaksCount = 0;
        for index2 = 1 : length(tempIRPeaksPlaces)
            if tempIRPeaksValues(index2) > 0.02
                tempPeaksCount = tempPeaksCount + 1;
            end
        end
        windowPeaksCount(index) = tempPeaksCount;
    end

    % Return the value that appears most
    detetedAmount = mode(windowPeaksCount);
end
```