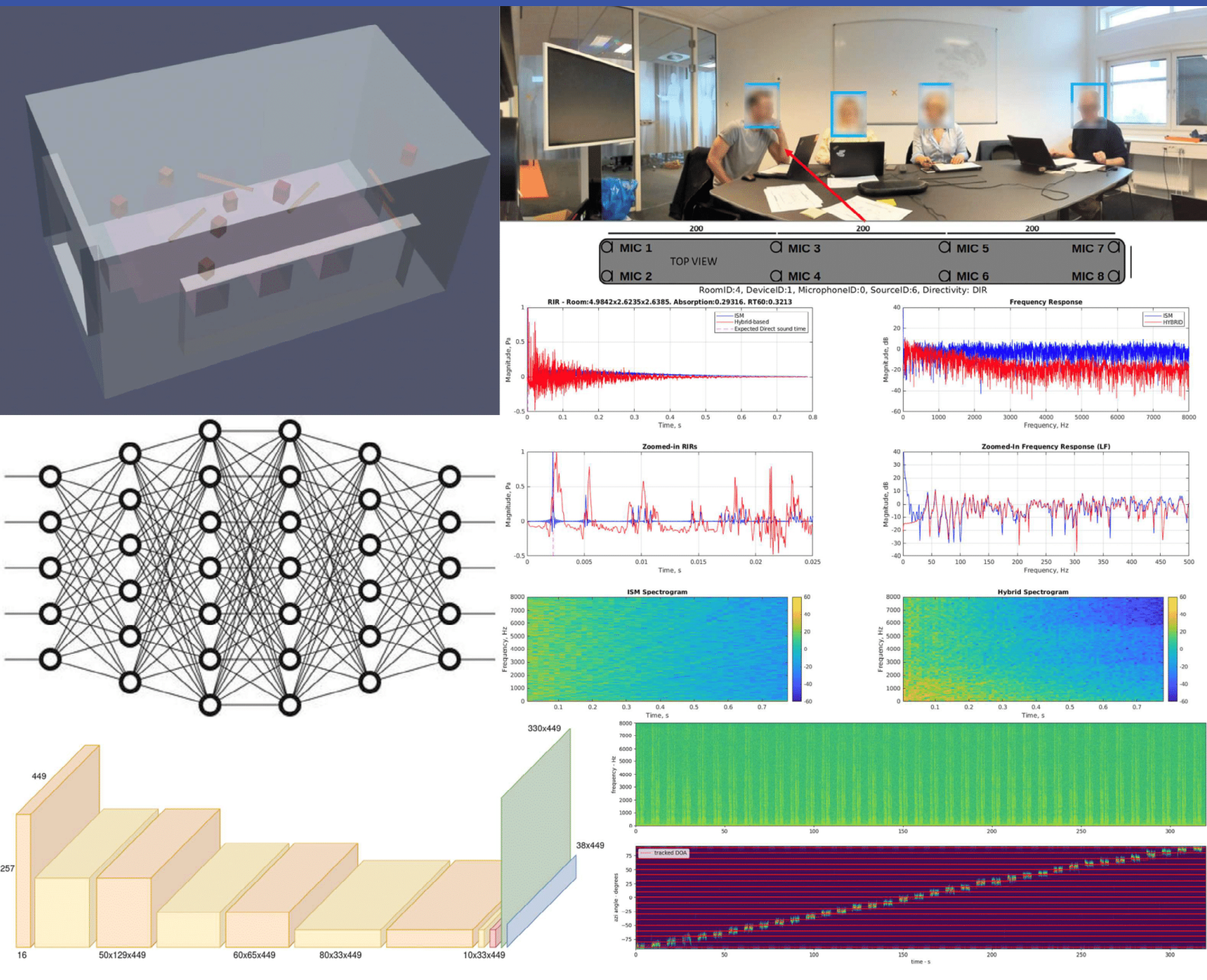
DTU Engineering Acoustics
Department of DTU Electro

Investigating the influence of RIR database and simulated training data on the performance of machine learned DOA methods

Enrico Leonardi

Master Thesis



Investigating the influence of RIR database and simulated training data on the performance of machine learned DOA methods
Enrico Leonardi

Master Thesis
July, 2024

By
Enrico Leonardi

Published by: DTU, Department of DTU Electro, Brovej, Building 118, 2800 Kgs. Lyngby Denmark
www.byg.dtu.dk
ISSN: [0000-0000] (electronic version)
ISBN: [000-00-0000-000-0] (electronic version)

ISSN: [0000-0000] (printed version)
ISBN: [000-00-0000-000-0] (printed version)

Approval

This thesis is submitted to the Technical University of Denmark (DTU) in partial fulfillment of the requirements for the Master of Science in Engineering Acoustics (MSc Eng). It is the result of a master project conducted at GN Store Nord A/S headquarter in Ballerup, Denmark, where I also gained work experience as a student assistant. Specifically, the thesis was prepared over five months at the Audio Research Section, Department of GN Audio.

Enrico Leonardi - s222721

.....

Signature

.....

Date

Abstract

Direction of Arrival (DoA) estimation, also known as Sound Source Localization (SSL), is a well-known challenge in the acoustics community. This study utilizes simulated speech recordings obtained from two different databases of Room Impulse Responses (RIRs) representing various virtual spaces. The first database is based on the Image-Source Method, a Geometrical Acoustic (GA) algorithm, while the second employs a hybrid method combining Wave-Based (WB) methods for lower frequencies and GA methods for higher frequencies. GA methods are a class of algorithms where the acoustic wave is approximated as a bundle of rays propagated in the room following the laws of ray optics, whereas WB methods directly approximate solutions of the partial differential equations governing wave motion. Being the training data based on such simulations, to address DoA estimation we utilize a Deep Neural Network (DoANet), performing iterative per-audio frame classification across possible DoAs within the azimuthal space of $[-90,\ 90]$ degrees. Several preprocessing steps for the simulated training data will be compared by evaluating DoANet's performance on real-world recordings. These preprocessing steps, representing the project's experiments, involve signal processing, data selection, and data manipulation. The effects of these steps will be assessed and discussed based on the network's output.

My work aimed to better understand which parameters are most and least relevant in DoA estimation using mentioned machine learned methods. Additionally, I provide a comprehensive overview of acoustic simulation methods, machine learning techniques, and preprocessing steps. This project is a joint collaboration between GN and DTU, developed primarily at the company's facilities.

Acknowledgements

Supervisors:

Rasmus Kongsgaard Olsson, Principal Research Scientist, GN Audio

Cheol-Ho Jeong, Associate Professor, DTU Electro

I would like to express my sincere appreciation to my supervisors, Associate Professor Cheol-Ho Jeong and Principal Research Scientist Rasmus Kongsgaard Olsson. Your mentorship has been invaluable, providing me with the opportunity to expand my knowledge in the fields of audio simulations and deep learning. Thank you for your availability, willingness to share your expertise, support, patience, and excellent suggestions.

I extend my heartfelt thanks to GN Store Nord A/S for generously allowing me to develop this project with the support of the company's facilities. I am also grateful to the entire Audio Research department and the Acoustic Development department for their attention and valuable insights regarding the project. A special mention goes to Senior Research Scientist Peter Risby Andersen, who first introduced me to my supervisor Rasmus and kindly guided me in understanding wave-based simulation methods.

La mia gratitudine si estende anche nel privato, alla mia famiglia e agli amici che mi hanno supportato in questi mesi. I miei pensieri vanno innanzitutto ai miei genitori - Silvio e Anna Rita, che hanno lavorato cosí tanto per garantire a me e ai miei fratelli il miglior percorso formativo che potessimo ricevere, e per il loro costante supporto. Voglio ringraziare i miei nonni - Elsa, Gerardo, Elena e Tiziano; e Silvia - madre dei miei due fratelli - per essere sempre stati presenti e per esserci tutt'ora. La mia massima premura va inoltre ai miei fratelli - Irene, Ruben ed Ermes - che con me rappresentano le nuove generazioni, le quali si trovano in quel punto critico della vita dove, parallelamente allo stare al passo con un mondo sempre piú dinamico, é comunque necessario essere in grado di prendere le scelte migliori possibili. Tuttavia, fortunatamente, non siete da soli in questo.

I also want to thank the dearest people of my Danish life, with whom I shared a similar path. First, my partner Hanlu, who faced so many challenges and achievements while being a true reference for me, both within and beyond this project. Additionally, I want to thank my friends Akul and Prajakta, who contributed with bringing into my life the social connections essential to achieve any happy life. Finally, I extend my gratitude to my Italian childhood friends, who are still close and have accomplished so much in these last few years: Alessandro, Loris, Bianca, Fabio, Eugenio and Matteo; I am proud of you all.

I feel extremely lucky, thanks to all of you for enriching my life.

Enrico Leonardi,

MSc Engineering Acoustics, DTU Electro

July, 2024

Contents

1 Introduction - The problem of Sound Source Localization: DoA finding

Sound Source Localization (SSL) is the problem of estimating the position of one or several sound sources relative to the position of the recording microphone array, based on a recorded multichannel acoustic signals. Typically, SSL simplifies to estimating the DoA of the sources, focusing on azimuth and elevation angles rather than the distance to the microphone array. SSL finds practical use in various fields such as source separation, automatic speech recognition, speech enhancement, human-robot interaction, noise control, and room acoustic analysis. [1]

Despite being extensively studied (e.g. Bartlett, Capon and MUSIC algorithms [2]), SSL remains a persistently challenging issue. Traditional SSL approaches rely on signal/channel models and signal processing (SP) techniques, which, despite their historical advancements, often falter in complex real-world scenarios characterized by noise, reverberation, and multiple simultaneous sound sources. Over the past decade, the emergence of data-driven deep learning (DL) methods has garnered significant attention for tackling these challenging circumstances. Consequently, an increasing number of SSL systems leveraging deep neural networks (DNNs) have surfaced in recent years. The majority of these studies have highlighted the superiority of DNN-based SSL techniques compared to conventional SP-based methods. [1]

The general principle of DL-based SSL methods and systems can be schematized with a simple pipeline, as illustrated in Figure 1.1. A multichannel input signal recorded with a microphone array is processed by a feature extraction module to provide input features. These input features are fed into a DNN, which delivers an estimate of the source location or DoA. A recent trend - which corresponds to what happens in this project - is to skip the feature extraction module to directly feed the network with multichannel raw data. In any case, the two fundamental reasons behind the design of such SSL are the following. [1]

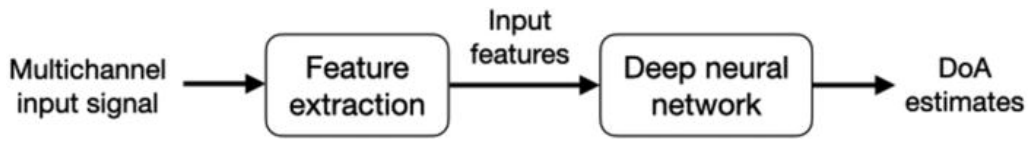


Figure 1.1: General pipeline of a DL-based SSL system

First, multichannel signals recorded with an array of microphones distributed in space contain information about the location of the source(s). Indeed, when the microphones are close to each other compared to their distance to the source(s), the microphone signal waveforms, although appearing similar from a distance, exhibit some amount of notable and complex differences in terms of delay and amplitude. These interchannel differences are due to distinct propagation paths from the source to the different microphones, for both the direct path and the numerous reflections that compose the reverberation in an indoor environment. In other words, a source signal is convolved with different room impulse responses (RIRs), which depend on the source position, microphone position and directivity (I denotes the microphone index in the array), and acoustic environment configuration (e.g., room shape). The microphone signals are often expressed in the

time-frequency (TF) domain, using the short-term Fourier transform (STFT), where the convolution in Equation 2.7 is assumed to transform into a product between the STFT of the source signal and the acoustic transfer function (ATF), which is the (discrete) Fourier transform of the corresponding RIR and is thus encoding the source spatial information. When several, sources are present, the recorded signal is the sum of their contribution (plus the noise). SSL then requires to proceed to some kind of source clustering, which is generally easier to proceed in the frequency or TF domain due to the natural sparsity of audio sources in that domain. [1]

The second reason for designing DNN-based SSL systems is that even if the relationship between the information contained in the multichannel signal and the location of the source(s) is generally complex, DNNs are powerful models that are able to automatically identify and exploit this relationship, given that they are provided with a sufficiently large number of representative training examples. While some conventional methods can adapt to the observed signals, they are all intrinsically based on certain (more or less plausible) modeling assumptions, which can limit their effectiveness when exposed to the complexity of real-world acoustics. Deep learning models do not explicitly impose any such assumptions, and instead they efficiently adapt to the presented training data. This is, however, also the major drawback of the DNN-based approaches, as they are less generic than traditional methods. A deep model designed for and trained in a given configuration (e.g., a given microphone array geometry) will not provide satisfying localization results if the setup changes, unless some relevant adaptation method can be used, which is still an open problem in DL in general. [1]

This study aims to use a Convolutional Neural Network (CNN) to compute the DoA of simulated recordings, expressed as azimuthal angle estimation. The simulated data is derived from anechoic recordings and large databases of two main types of provided RIRs: one developed solely by GN, and the other produced through a joint project between GN and the Icelandic company Treble Technologies. The research seeks to identify the most relevant data parameters for the DoA problem by considering the differences in simulation methods, and consequently refining the training data so that the CNN can achieve a desired performance over real-world scenarios. A holistic representation of the steps is displayed within Figure 3.1.

Simulation audio data training data is a practical alternative to real audio recordings for training machine learning applications such as speech enhancement, DoA finding, echo control, etc. Simulations are very flexible to form factor, microphone configuration and evolving sets of use cases. However, the conventional approaches of collecting such data present various limitations. Measurement based approaches are costly and time-consuming, and synthetic data generation using standard acoustics simulation methodology has been shown to generalize poorly to real world scenarios, due to limitations in capturing the intricacies of real world room acoustics [3]. The key challenge is to make the simulation realistic (and fast) enough so that the model generalizes well and performs well in all the desired scenarios. GN teamed up with Treble Technologies to explore advanced acoustic simulation techniques to create large scale audio training data for machine learning/AI product features.

2 Background Theory

The purpose of chapter 2 is to explain, in simple terms, the fields of room acoustic simulations (see section 2.1) and machine learning (see section 2.2), with particular emphasis on the methods and models relevant to this project. The chapter begins with an introduction to room acoustic simulations, followed by an overview of machine learning and specific DNN models. Finally, the chapter discussing the project's experiments will be presented (see chapter 3).

2.1 Room Acoustic Simulations

The field of room acoustic simulations dates back to the seminal work of Schroeder in the 1960's. The first computational implementation was that of Krokstad et al. in 1968, who implemented a ray tracing algorithm to compute the time-energy response of rooms. The first commercial tools began to appear in the 1990's, such as *Odeon*, *CATT* and *EASE*. As of 2020, the field is still an active field of research, with on-going work to improve the prediction methods themselves, their computational implementations, boundary modeling, sound source modeling and various other aspects. Also, in recent years, the use of room acoustic simulations has been extended beyond that of building design, e.g., to video games, and this has lead to new requirements on accuracy and performance [4].

Today, room acoustic simulations are a valuable tool for building designers to optimize the acoustic conditions of their designs prior to construction or renovation [4]. Simulations enable architects and acousticians to achieve a high level of acoustic detail in building design, enhancing project accuracy and reducing the risk of miscalculations. Acoustical properties are easier to validate through simulations, hence the advantages. However, the current state of computational resources presents challenges and limitations, particularly when aiming to make simulations as realistic as possible.

The numerous simulation methods proposed in the literature are typically divided into two main groups: **geometrical acoustics** (GA) methods and **wave-based** (WB) methods. GA methods are generally computationally efficient, but the accuracy can be low. They are based on the assumption that acoustic waves behave similarly to bundle of rays that are propagated in the room using the laws of ray optics. In WB methods, the governing differential equations of wave motion in an enclosure are solved directly instead, yielding highly accurate schemes, but hampered by excessive computation times [4].

In other words, the task can be challenging from a computational point of view as it involves simulating large and complex domains, over a broad frequency spectrum and long times. Room sizes range from $30\ m^3$ to $30.000\ m^3$, our auditory system can hear frequencies from $20\ Hz$ to $20\ kHz$, and sound typically lasts in rooms somewhere between $0.5\ s$ (living room) to $3.0\ s$ (concert hall). Historically, the prevailing approach has been to apply GA methods, such as the Image-Source (ISM) or Ray-Tracing (RT), where the acoustic wave is approximated to follow the laws of ray optics. This reduces the computational task considerably, but the approximation is only relatively appropriate for high frequencies, where the wavelengths are smaller than the dimensions of the room and the obstacles. At low-mid frequencies, wave phenomena such as diffraction and interference becomes more prominent and other simulation methods must be used [4].

This motivates the use of Wave-Based (WB) methods, where the governing partial differential equations that describe wave motion are solved numerically. Given the right

input data, these methods can be very accurate, since no approximation to the wave propagation is made and all wave phenomena are inherently accounted for. Several different numerical techniques have been developed, such as the finite-difference time-domain method, the pseudospectral time-domain method, the finite volume method, the boundary element method, and the Finite Element Method (FEM). Recently, nodal high-order-accurate variants of the FEM, such as the spectral element method (SEM) and the discontinuous Galerkin Finite Element Method (**DGFEM**), have been applied to simulate room acoustics [5].

A last note on the simulation methods is the notion of method *hybridization*, i.e., to combine different simulation methods with the purpose of improving speed and/or accuracy. A classic approach is to combine the image source method and the ray tracing method. The ISM is more accurate than ray tracing, but its computational cost grows exponentially with the reflection order. Knowing the perceptual importance of early reflections, a natural approach arises where the image source method is used for simulating the early portion of the impulse response and the ray tracing method is used for the late reverb tail. Another natural hybridization approach is to combine wave-based and geometrical acoustics simulations, making the wave-based methods cover the lower portion of the spectrum where wave phenomena is prominent and wave-based methods are efficient, and the geometrical methods cover the high frequency portion, where the geometrical acoustics assumption is reasonable [4].

The most relevant simulation methods for the scope of this project are described in the following chapters.

2.1.1 Wave-Based: Finite Element Method

In WB methods, numerical techniques are applied to directly solve the governing partial differential equations (PDEs) that describe wave motion in an enclosure.

$$\nabla^2 p - \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2} = 0 \tag{2.1}$$

where p represents the acoustic pressure, c the speed of sound in air and t the time [4]. These methods can be highly accurate, but the major drawback is the associated high computational cost. Simulating large spaces (larger than, say, $1.000\ m^3$) into the mid-frequency range (up to and beyond $1\ kHz$) typically requires tens or hundreds of millions of degrees of freedom (DoF). For this reasons, and despite the progress in numerical methodology, WB methods are mostly applied to small rooms at very low frequencies.[5] One of the most commonly used numerical method for WB room acoustic simulations is the **Finite Element Method** (FEM). The FEM is a volumetric discretization method where the solution is represented in terms of globally continuous, piece-wise basis functions, which are defined by patching together local basis functions defined on each mesh element. The partial differential equations are satisfied by using variational methods to minimize an associated error residual by fitting weighing coefficients of test functions [4]. To be more precise, within the room acoustics community, the expression *finite element method* is traditionally associated with the second-order continuous Galerkin FEM (using linear or quadratic basis functions) [4]. Given the right input data, these methods are very accurate, since no approximation to the wave propagation is made and all wave phenomena are inherently accounted for, although assuming it as a linear and adiabatic process. Recently, nodal high-order accurate variants of the FEM, such as the spectral element method (SEM) and the discontinuous Galerkin finite element method (DGFEM), have been applied to simulate room acoustics [5].

Discontinuous Galerkin Finite Element Method

The classic second order FEM is a continuous formulation. However, it can also be formulated in a discontinuous and local manner. DGFEM relies on local weak formulations defined for elements rather than for the full domain. The nodal DGFEM is particularly well suited for room acoustic simulations, because it combines the attractive features of geometric flexibility, high-order accuracy, suitability for parallel computing, and lean memory usage. In the DGFEM schemes, the solution is computed locally for each element, with communication between elements only at the element boundaries. This data locality can be utilized for parallelization of the solver, which addresses the typical high computational cost problem of WB methods [5]. Being DGFEM a variant of the continuous Galerkin FEM, it is reasonable to understand the idea behind the latter first.

As in every wave-based simulation method, the task boils down to solving (approximately) the wave equation 2.1 [4]. By integrating by parts and applying the test function ϕ, the shape function N, the normal vector n, and a discretized pressure p, we can derive a weaker formulation:

$$-\int_{\Omega} \mathbf{p}\nabla\phi \cdot \nabla N \; d\Omega + \frac{1}{c^2}\int_{\Omega} \ddot{\mathbf{p}}\phi N \; d\Omega = -\int_{\Gamma} \phi \left(\nabla p \cdot \mathbf{n}\right) \; d\Gamma \; ^* \tag{2.2}$$

Here, ϕ is the test function useful to solve the resulting system of equations, and designed to minimize the numerical dispersion error across subsequent steps of the approximation.
* This formulation has yet to be discretized.

The specificity of Galerkin models (and generally of FEM), is considering $\phi = N^T$, where N is the matrix responsible for interpolating the discretized **p** approximating it to its continuous-valued version p (see Equation 2.3).

$$p \approx N\mathbf{p} \tag{2.3}$$

The formulation in Equation 2.2, can be traced back to the Newmark implicit time-stepping method:

$$\mathbf{Kp} + \mathbf{M}\ddot{\mathbf{p}} = \mathbf{f} \tag{2.4}$$

This can be used to solve the differential equation, as being performed in [6]. However, the key difference from its continuous counterpart, is that in the DGFEM (see [5]) not only no matrix inversion is being performed - which would be computationally expensive - but also the spatial discretization differs, as sketched in Figure 2.1.

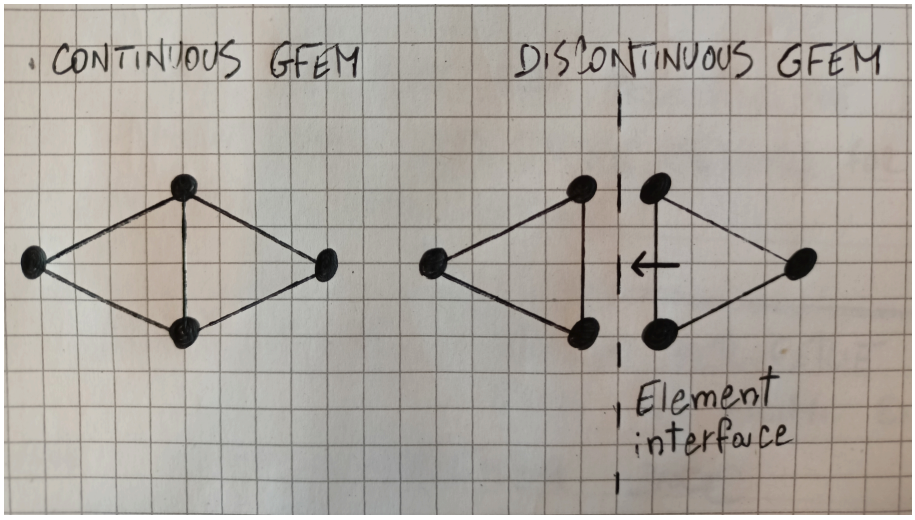


Figure 2.1: Qualitative sketch of spatial discretization: (CG)FEM vs. DGFEM

In DGFEM, instead of implementing a spatial discretization with common nodes, each node is described locally with its estimations of pressure and velocity, hence the term *discontinuous*. This increases the DoF of the shape function but enforces equal flux across the element interfaces, significantly reducing accumulated dispersion error and allowing for explicit time-stepping computation (so with $\phi = N$, without matrix inversion). Finally, this description facilitates parallelization of the computation [5].

This method has been used to approximate the behavior of acoustic waves in simulations adhering to the hybrid method below the threshold frequency of $1.6\ kHz$. Physically, this threshold can be considered an arbitrarily set Schroeder frequency:

$$f_{Schroeder} = 2000\sqrt{\frac{T_{60}}{V}} \tag{2.5}$$

Below this frequency, the wavelengths are large compared to room sizes and obstacles typically found in rooms - causing wave phenomena such as diffraction and interference to dominate the acoustics. Above this threshold, thousands of modes overlap and sound can be approximated as particles exhibiting ray behavior. In Equation 2.5 T_{60} stands for reverberation time and V for room volume [4].

2.1.2 Geometrical: Image-Source Method

The ISM is a GA method which is considered as a universal tool in many fields of acoustical and engineering research; it has been implemented as a basic principle for a wide range of purposes including: prediction of sound propagation in indoor environments, sound rendering in virtual environments, binaural auralization, noise control in closed environments, virtual reality applications, etc. [4]. Originally described in the 1979 Allen et al. [7], the ISM is based on the idea that when a sound ray collides with a plane wall, the reflected sound ray can be imagined as originating from an *image source*, which is the mirror image of the original sound source created by the wall. In other words, for every surface in the model and for every real sound source in the space, ISM creates virtual sources that simulate the reflections of the original sound waves. These virtual sources are located symmetrically with respect to the reflective surfaces and they are considered ”images” of the original source (see sketch in Figure 2.2).

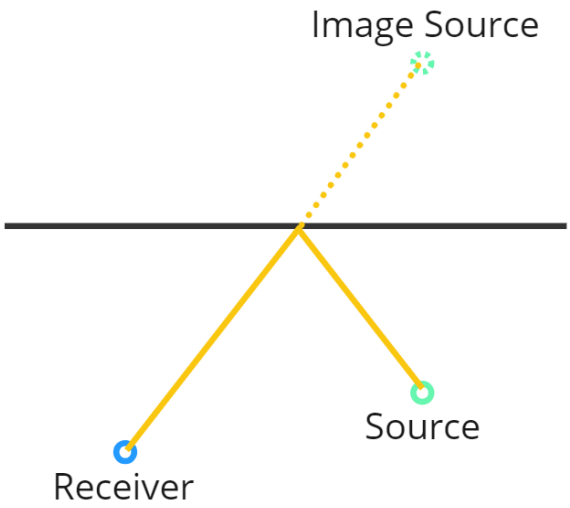


Figure 2.2: Visualisation of ISM principle

The image sources are then projected towards the receiver in order to compute the rays' path validity and the sound pressure at the receiver. The method is particularly useful in scenarios where the geometry of the space is relatively simple and can be approximated by a set of reflective surfaces, such as walls, floors, and ceilings. However, its accuracy depends on several factors, including the complexity of the geometry, the absorption properties of surfaces, and the frequency range of interest. To model the higher order reflections, all of the first order image sources needs to be recursively mirrored across the planes in the same manner as with the original source, forming a tree structure of image sources. After the tree is populated, the image sources are then used to compute the sound pressure of valid reflections projected directly at the receiver. The valid reflection paths are found by backtracing all of the tree branches from the receiver, through the image sources and back to the original source. Once all the valid reflection paths have been found, a Pressure-Based (PB) solver uses the reflection paths and the impedance and scattering boundary data to model how sound propagates and reflects to reconstruct a response for each reflection. All of these responses are then summed to create the resulting impulse response. The effect of the boundary at each reflection takes into account the complex impedance and thereby making it possible to model phase-changes introduced by the boundary. Using the impedance also has the added bonus that it inherently models the local reaction, and thereby the angle dependence of the reflection. Since the image source method only models the specular part of the response, the scattered energy is subtracted at each reflection [8].

This method has been used to model the behavior of acoustic waves in simulations adhering to the ISM across the whole spectrum. Additionally, it describes the direct path and first reflections in the hybrid method for frequencies above the threshold of $1.6\ kHz$.

2.1.3 Geometrical: The Ray-Radiosity Method

The traditional method of acoustic ray tracing is to trace rays around the space while looking for intersections between the rays and a small sphere. To limit the amount of rays needed, the solver uses a Ray-Radiosity (RR) approach to ray tracing the scattered and the late part of the response. This method generally assumes that the sound field is diffuse, which is considered fair for the scattered and the late part of the response [9].

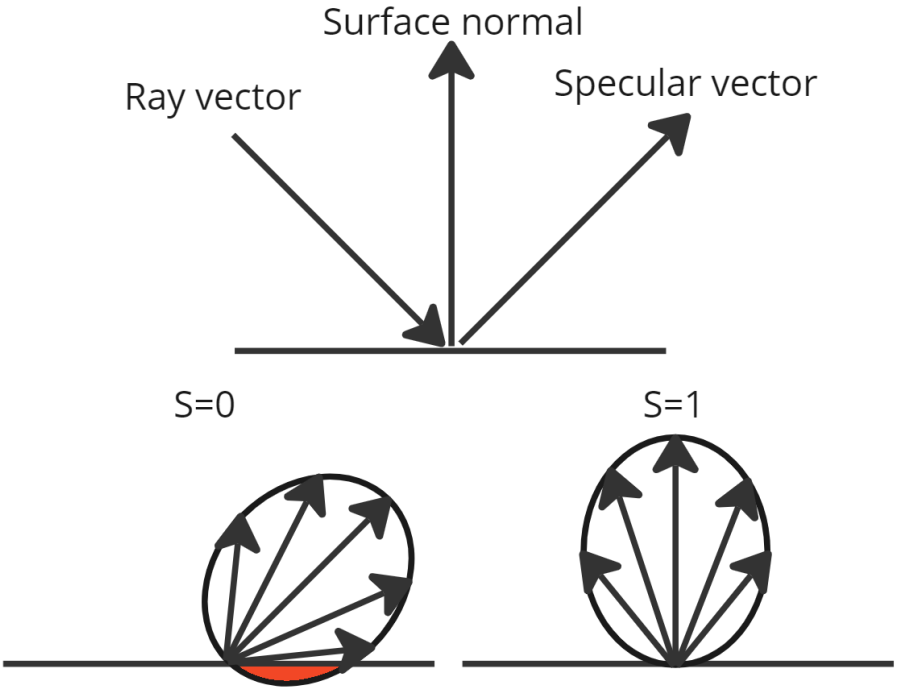


Figure 2.3: The Ray-Radiosity method parameters

In the RR solver, a ray is traced from the source like in traditional acoustic RT, but at every reflection a small part of the energy is traced back to each receiver under the assumption that the sound field and the reflection is diffuse. The energy that the ray carries is only decrease when a ray is reflected off an absorbing surface. Here the random-incidence absorption coefficient is used. The decay over distance is handled by the fact that the rays spread out as they propagate though the space. When the ray is reflected off a surface there are multiple factors that decide in which direction the new ray will travel. The specular vector is computed along with a scattered vector. The resulting vector will be a new vector based on a weighted average between the two vectors [9].

$$\hat{v}_{resulting} = (1 - s)\,\hat{v}_{specular} + s\hat{v}_{scattered} \tag{2.6}$$

The energy at a receiver is calculated by estimating how much energy would be probably to hit receiver in the case of a complete diffusion of the incoming wave. The scattering coefficient is taken into account in this step as well. Here the scattering coefficient is used to scale the amount of energy being registered. This works by angling the lambert distribution towards either the surface normal (scattering=1) and towards the specular vector (scattering=0) [9].

As introduced in the hybridization paragraph in section 2.1, the aforementioned method has been used to model the reverberation tail of simulations adhering to the hybrid method for frequencies above the threshold of $1.6\ kHz$. To summarize the characteristics of the hybrid model, Figure 2.4 is presented. Different parts of a spectrogram of a RIR can be attributed to the different components of the hybrid method.

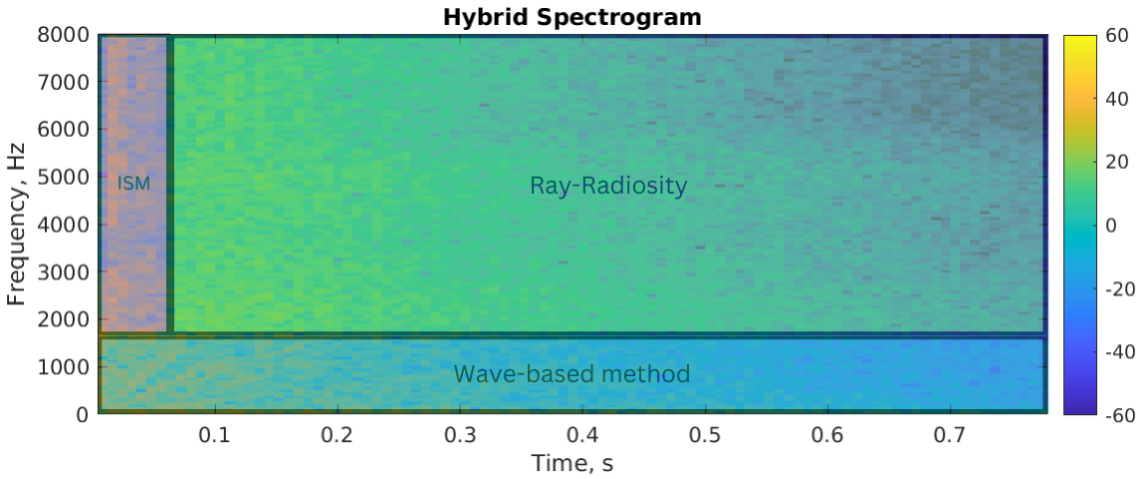


Figure 2.4: An overview of hybrid method's structure. Lower frequencies components have been chosen to be approximated by WB DGFEM, whereas the higher frequencies by ISM (early reflections) and RR (late reverb tail)

Further description and comparison of the RIRs generated by the employed methods are provided in Appendix A.

2.2 Basics of Machine Learning

Machine learning is essentially a form of applied statistics with increased emphasis on the use of computers to statistically estimate complicated functions and a decreased emphasis on proving confidence intervals around these functions [10].

A machine learning algorithm is an algorithm that is able to *learn* from data. Learning is meant as in the definition provided by Mitchell (1997): "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E". In other words, it's an algorithm that is capable of improving a computer program's performance at some task via experience. Many kinds of tasks can be solved with machine learning. Some of the most common machine learning tasks include the **classification**: in this type of task, the computer program is asked to specify which of k categories some input belongs to. To solve this task, the learning algorithm is usually asked to produce a function $f : \Re^n \rightarrow \{1, ..., k\}$. When $y = f(x)$, the model assigns an input described by vector x to a category identified by numeric code y [10].

To evaluate the abilities of a machine learning algorithm, we must design a quantitative measure of its performance. Usually this performance measure P is specific to the task T being carried out by the system. For tasks such as classification, we often measure the **accuracy** of the model.[10] However, in our experiments we chose to use the **E_DoA** average error parameter, which is a regression loss metric, based on the raw DNN output classes. E_DoA computation will be further explained in a later chapter (see section 3.2).

Usually we are interested in how well the machine learning algorithm performs on data that it has not seen before, since this determines how well it will work when deployed in the real world. We therefore evaluate these performance measures using a test set of data that is separate from the data used for training the machine learning system.[10] In our experiments, the **training data** consist of simulated audio recordings, whereas the **test data** consist of real-world recordings.

Machine learning algorithms can be broadly categorized as unsupervised or supervised by what kind of experience they are allowed to have during the learning process. Most of the learning algorithms can be understood as being allowed to experience an entire dataset. A dataset is a collection of many examples, which sometimes we call data points. Supervised learning algorithms experience a dataset containing features, but each example is also associated with a label or target [10]. A supervised learning approach is utilized in this work, where the network during training is provided with both simulated recordings and per-frame target bell-curves of DoA (see Figure 3.5).

2.2.1 Machine Learning for Audio

The entire research domain on Audio and Acoustic Signal Processing (AASP) is recently witnessing a paradigm shift towards data-driven methods based on machine learning and especially deep learning. In many applications, such data-driven models obtain state-of-the-art results if appropriate data is available to train the models [11].

The simple machine learning algorithms work well on a wide variety of important problems. They have not succeeded, however, in solving some of the central problems in AI, such as recognizing speech or recognizing objects. The development of deep learning was motivated in part by the failure of traditional algorithms to generalize well on such AI tasks. The challenge of generalizing to new examples becomes exponentially more difficult when working with high-dimensional data, and the mechanisms used to achieve generalization in traditional machine learning are insufficient to learn complicated functions in high-dimensional spaces (curse of dimensionality). Such spaces also often impose high computational costs. Deep learning was designed to overcome these and other obstacles [10]. Indeed, the goal of this project is to further explore methods for addressing the DoA problem using DNNs instead of traditional algorithms.

2.2.2 Deep Learning and convolutional neural networks

As previously mentioned, DL is a specific kind of ML, where the field of DNNs has a long history and many aspirations. In order to properly understand the transition between ML and DL, we have to start by explaining Deep feedforward networks, the quintessential deep learning models - also called feedforward neural networks [10].

The goal of a feedforward network is to approximate some function f^*. For example, for a classifier, $y = f^*(\mathbf{x})$ maps an input **x** to a category y. A feedforward network defines a mapping $\mathbf{y} = f(\mathbf{x}; \theta)$ and learns the value of the parameters θ that result in the best function approximation. These models are called feedforward because information flows through the function being evaluated from **x**, through the intermediate computations used to define f, and finally to the output **y**. Feedforward neural networks are called **networks** because they are typically represented by composing together many different functions. For example, we might have three functions $f^{(1)}$, $f^{(2)}$, and $f^{(3)}$ connected in a chain, to form $f(\mathbf{x}) = f^{(3)}(f^{(2)}(f^{(1)}(\mathbf{x})))$. These chain structures are the most commonly used structures of neural networks. In this case, $f^{(1)}$ is called the **first layer** of the network, $f^{(2)}$ is called the **second layer**, and so on. The overall length of the chain gives the **depth** of the model. The name *deep learning* arose from this terminology. During neural network training, we drive $f(\mathbf{x})$ to match $f^*(\mathbf{x})$. The training data provides us with noisy, approximate examples of $f^*(\mathbf{x})$ evaluated at different training points. Each example **x** is accompanied by a label $y \approx f^*(x)$. The training examples specify directly what the output layer must do at each point **x**; it must produce a value that is close to y. The behavior of the other layers is not directly specified by the training data. Instead, the learning algorithm must decide how to use these layers to best implement an approximation of f^*. Because the training data does not show the desired output for each of these layers, they are

called hidden layers. Finally, these networks are called *neural* because they are loosely inspired by neuroscience. Each element of the vector may be interpreted as playing a role analogous to a neuron [10].

Convolutional networks, also known as convolutional neural networks (CNNs) are a specialized kind of neural network for processing data consisting of convolutional blocks. The name *convolutional neural network* indicates that the network employs the mathematical operation of convolution, which is a specialized kind of linear operation. Convolutional networks are simply neural networks that use **convolution** in place of general matrix multiplication in at least one of their layers. Usually, the operation used in a convolutional neural network does not correspond precisely to the definition of convolution as used in other fields, such as engineering or pure mathematics. In its most general form, convolution is an operation on two functions x and w of a real-valued argument.

$$s(t) = (x * w)(t) = \int x(a)\, w(t-a)\, da \tag{2.7}$$

However, Equation 2.7 does not apply in our case because we work with data on a computer, so time is discretized. Moreover, in machine learning applications the x **input** is usually a multidimensional array of data, and the w **kernel** (argument of the convolution) is usually a multidimensional array of parameters that are adapted by the learning algorithm. Furthermore, we often use convolutions over more than one axis at a time. Indeed, 2D convolution is employed on the CNN of this project, being 2D the nature of images. Therefore, if we use a two-dimensional image I as our input, and we employ a two-dimensional kernel K:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)\, K(i-m, j-n) \tag{2.8}$$

we obtain the actual definition of convolution being employed in such NNs [10].

Convolution leverages three important ideas that can help improve a machine learning system: sparse interactions, parameter sharing and equivariant representations. Moreover, convolution provides a means for working with inputs of variable size.

Traditional neural network layers use matrix multiplication by a matrix of parameters with a separate parameter describing the interaction between each input unit and each output unit. This means that every output unit interacts with every input unit. Convolutional networks, however, typically have **sparse interactions** (also referred to as sparse connectivity or sparse weights). This is accomplished by making the kernel smaller than the input. This means that we need to store fewer parameters, which both reduces the memory requirements of the model and improves its statistical efficiency. It also means that computing the output requires fewer operations. **Parameter sharing** refers to using the same parameter for more than one function in a model. In a traditional NN, each element of the weight matrix is used exactly once when computing the output of a layer. In a CNN, each member of the kernel is used at every position of the input. Convolution is thus dramatically more efficient than dense matrix multiplication in terms of the memory requirements and statistical efficiency.

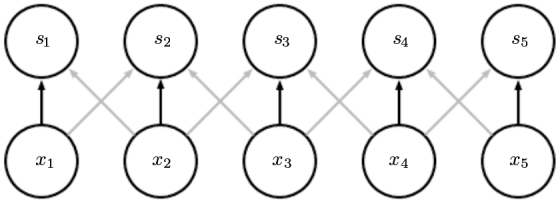


Figure 2.5: **Grey** sparse connectivity: when s is formed by convolution with a kernel of width 3, only three inputs affect s_3. **Black** parameter sharing: the arrows indicate uses of the central element of a 3-element kernel in a convolutional model. Because of parameter sharing, this single parameter is used at all input locations [10].

In the case of convolution, the particular form of parameter sharing causes the layer to have a property called **equivariance** to translation. To say a function is equivariant means that if the input changes, the output changes in the same way [10].

This concludes both the overview of the section related to machine learning and the entire introductory chapter.

3 Experimental setup

To evaluate the 'quality' of the simulations in terms of DoA estimation and error calculation, the following experimental setup has been designed. This setup chain consisted of five consequential main steps:

Step 1 Large RIRs and anechoic recordings databases.

Step 2 Simulated recordings.

Step 3 Individual experiments.

Step 4 CNN.

Step 5 DoA estimation.

whose visual overview is presented by Figure 3.1.

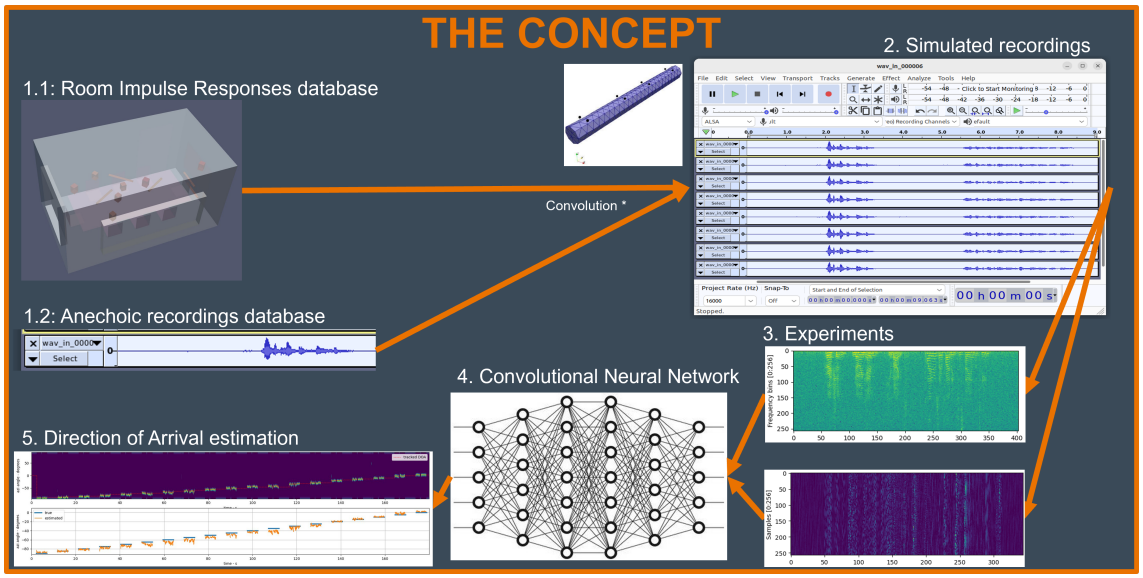


Figure 3.1: Diagrammatic representation of the experimental setup sequence

A description of each step lies within the following chapters.

3.1 Simulated Data Generation

Step 1
Large databases of RIRs and anechoic recordings have been provided. The methods used to obtain the RIRs are described in section 2.1. The simulated rooms vary in size - from approximately $2 \times 2 \times 2\ m$ to $6 \times 6 \times 6\ m$ - and in complexity, from empty shoebox-type rooms to those with a small number of objects and obstacles (e.g. tables, chairs, different wall properties for TV set, door, etc.) (see Figure 3.2).

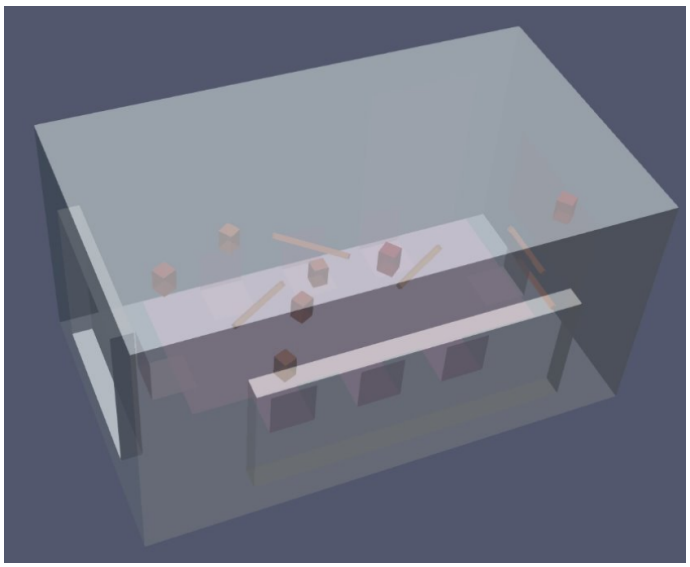


Figure 3.2: Example of modelled virtual space

The 1000 rooms were populated by 14 different sound sources (which can be either directional or omnidirectional) and 5 different receivers. All the receivers were modeled as 8-channel arrays to emulate the information retrieval performed by the *Panacast P50* GN meeting microphones. The 8-channel microphone array consists of a planar distribution of 4 pairs of isotropic sensors, which spatially sample the acoustic field at 8 different locations. The inter-pair distance is approximately $5\ cm$, while the intra-pair distance is approximately $20\ cm$ (see Figure 3.3). The hybrid simulation method modeled the receiver as shown in Figure 3.3, whereas the ISM modeled the array without support, with the individual microphones floating in air.

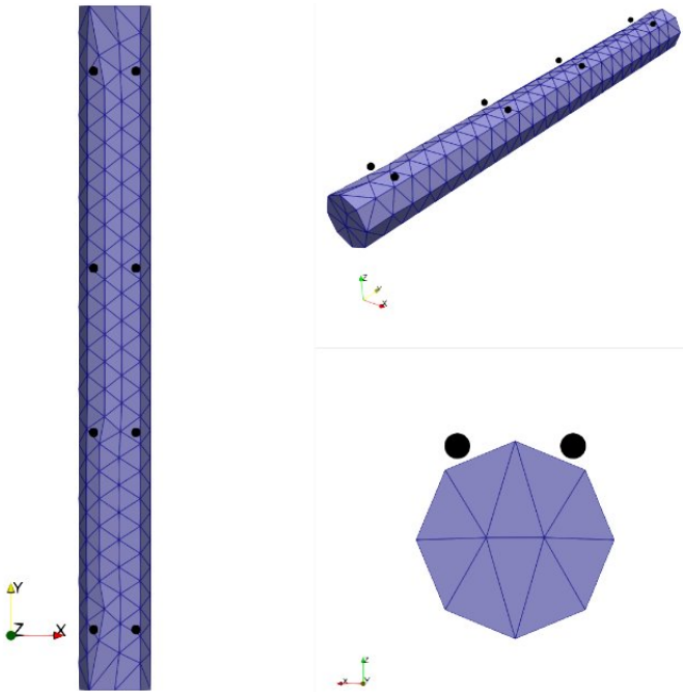


Figure 3.3: Modelled microphone array

The transition frequency parameter, which determines the frequency below which the hybrid method uses a wave-based approach and above which it uses a geometrical acoustics approach, was set to $1.6\ kHz$ (see Equation 2.5). Additionally, each object in the simulated rooms was assigned an absorption value to approximate its absorption coefficient, with values ranging from 0.05 to 0.8.

As previsouly mentioned, the simulated recordings were obtained by convolving RIRs with anechoic recordings. The anechoic recordings database was partially generated by GN, and partially retrieved online [12]. The CSTR's VCTK Corpus (Centre for Speech Technology Voice Cloning Toolkit) includes speech data uttered by 109 native speakers of English with various accents. Each speaker reads out about 400 sentences, most of which were selected from a newspaper plus the Rainbow Passage and an elicitation paragraph intended to identify the speaker's accent [12]. GN recordings include both speech recordings and various impulse noises sources.

The goal of these settings is to simulate 'normal use' scenarios, such as rooms where meetings take place with many participants, and sound is captured with a single microphone array in potentially noisy environments.

Step 2
It's possible to retrieve simulated recordings by performing a **linear convolution** operation $*$ between RIRs and anechoic recordings, described by the discrete-time Equation 3.1. In doing so, the resulting recordings will achieve the effect of sound as if they had been physically played and recorded in real spaces.

$$y[n] = x[n] * h[n] = \sum_{m=0}^{n} x[m]\, h[n-m] \;\; for\ n = 0, 1, ..., L-1 \tag{3.1}$$

Being y the $L = (N + M - 1)$-long outcome of the convolution, x the speech recording, h the RIR, N the length of x and M the length of h [13].

Since most of the speech information lies within a lower-than $8\ kHz$ portion of the spectrum, which is also the reason why many consumer microphones' frequency responses are not required to operate across the entire $[20\ 20000]\ Hz$ audible range, simulations - and also the later processing stages - have been set to deal with frequencies up to $8\ kHz$ in order to save computational resources and reduce processing time.

The steps hitherto implemented have used a Matlab framework. However, all the following steps have been implemented using Python. Due to the signed non-disclosure agreement with GN, no specific lines of code will be attached to this thesis.

3.2 Training framework

As previously mentioned, the preprocessing steps represent the actual experiments of the project. In particular, the term 'experiment' refers to any decision involved in at least one of the following areas: choice of RIR databases, training data generation, training data selection/manipulation, or test data selection/manipulation.

Step 3
As previously mentioned (see chapter 1), the literature suggests that relying on spectrograms instead of raw audio data performs better when addressing the problem of DoA finding with a CNN [1]. Therefore, the first operation performed on the training data was to apply the **Short-Time Fourier Transform** (STFT) to compute their spectrograms (see Figure 3.11).

By definition (see Equation 3.3), the STFT outputs complex-valued numbers - composed by a $\Re$eal and an $\Im$maginary part - which are stored within 16 channels (8 real + 8 imaginary). The NN receives these data-blocks grouped into batches of 4 as inputs, each containing 16 channels. Each channel contains 2D matrices of values, where each value represents the intensity of a sound identified by a frequency bin (a total of 257, out of which the spectrum has been discretized) and by a frame number (from the entire audio sample, which is of non-fixed length). Given this data structure, the NN was designed to better match it. A **dilated Convolutional Neural Network** with bi-dimensional 3×3 kernel filters for feature map extraction has been employed. As a reference, Figure 3.4 displays the data flow shape from the beginning of the NN until its output.

Step 4

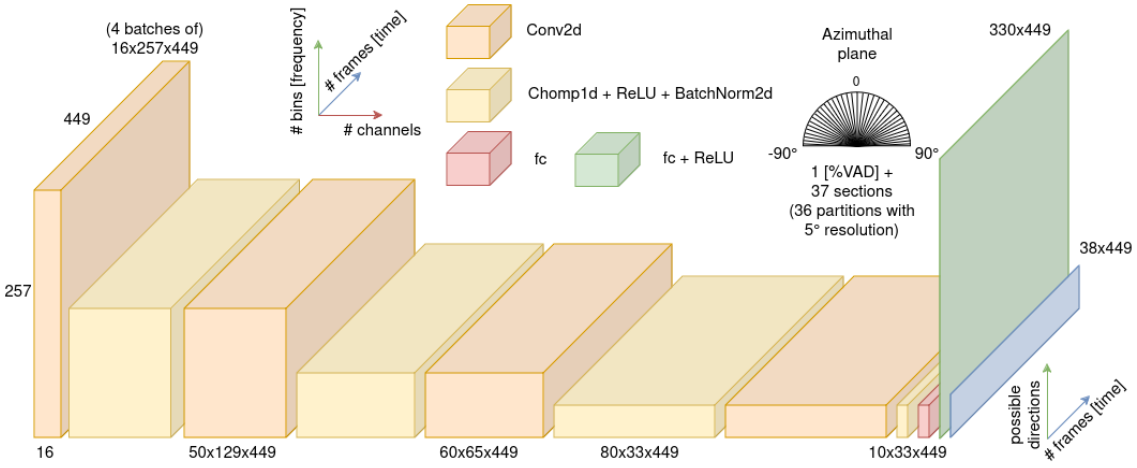


Figure 3.4: Diagram of data flow across the CNN

The NN processes the data through many 2D convolutional and fully connected layers, outputting a matrix that matches the length of the audio segment and the width of the number of classifications being performed. Specifically, the estimations are carried out across the $180°$ azimuthal plane, subdivided into 37 sections (yielding 36 partitions with $\frac{180}{36} = 5°$ resolution). The NN selects the section from which the sound is most likely arriving on a frame-by-frame basis. In other words, the NN acts as a **classifier** and performs a frame-size number of classifications for each frame. The value contained within the 37 rows represents the probability of the sound coming from a particular section of the azimuthal plane. An example of the interpolated distribution of probability across the 37 channels of a frame is provided by Figure 3.5. The 38th channel contains information about how much speech has been detected within each audio frame, known as Voice Activity Detection (VAD). This information is not considered for the scope of this project.

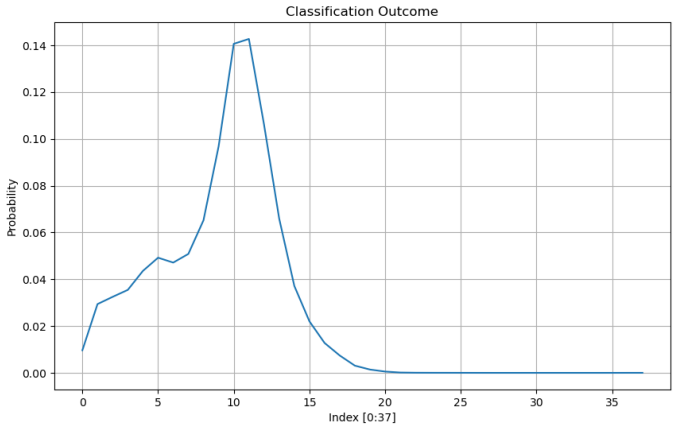


Figure 3.5: Classification outcome (one frame example)

3.3 The training setup

The training was conducted using the *PyTorch* framework [14] on a computer located at the GN headquarter in Ballerup, Denmark. The workstation operated on a Linux kernel with Ubuntu 22.04 as the operating system and utilized an NVIDIA GeForce GTX 1080 graphic processing unit (GPU).

The CNN processes 4 batches of $16 \times 257 \times$:-structured data blocks (refer to Figure 3.4). The network comprises several types of layers:

- $Conv2d$ "2D Convolutional" layer: Applies sliding convolutional filters (kernels) to the 2D input, resulting in a set of 2D feature maps. This operation depends on various parameters such as stride, padding or dilation;
- $Chomp1d$ layer: Removes excess padding from the convolutional output;
- $BatchNorm2d$ "2D Batch Normalization": Normalizes the inputs of each layer in a batch to have zero mean and unit variance;
- $ReLU$ "Rectified Linear Unit": A non-linear activation function that introduces non-linearity into the model by outputting the input directly if it is positive, and zero otherwise. This is crucial for enabling the network to approximate complex functions;
- fc "Fully Connected" layer: A layer where each neuron is connected to every neuron in the previous layer.

During training, the model iteratively processes batches of input data (simulated recordings spectrograms and target DoA bell curves, see the end of section 2.2) through the network. It applies sliding convolutional filters (kernels), each defined by a set of weights (parameters), which produce a same-amount of feature maps. The optimizer, Adaptive Moment Estimation (*ADAM*) [15], iteratively adjusts the nearly 500,000 parameters. It uses gradients of the loss function, a measure of how well the model's predictions match the actual target values (bell curves) during training, to minimize the loss for each parameter. This implementation employs the Binary Cross Entropy loss function combined with a sigmoid layer [16]. The optimizer utilizes hyperparameters such as the learning rate to scale the gradient steps it takes to update the model weights. The key parameters of the described architecture are summarized in Table 3.1.

Input	Output	# training data	Kernel size	Batch size	# parameters
$16 \times 257 \times :$	$38 \times :$	25754^*	3×3	4	478568

Iterations	Optimizer	Learning rate	Epochs	Loss function
$502k$	$ADAM$	0.001	≈ 80	$BCEWithLogitsLoss$

Table 3.1: Key parameters of the NN

*25754 is the number of training files generated using the ISM method and utilized to train the NN. The hybrid method, however, produced a larger dataset of 35001 files. To ensure consistent training, an identical number of training iterations (502000) was employed for both methods.

Given that the experiments involved various types of data manipulation, each experiment had its own training duration, as summarised in Table 3.2.

Spectrogram	LP-filter	HP-filter	Time domain	Inner Mics	Outer Mics	Mel
$\approx 29h$	$\approx 36h$	$\approx 21h$	$\approx 13h$	$\approx 33h$	$\approx 33h$	$\approx 46h$

Table 3.2: Approximate training time of the experiments

Step 5
The output of the NN (DoA bell curve, see Figure 3.5) is mapped into 180 azimuthal space partitions. To achieve a resolution finer than $5°$, the output is processed by subtracting each value from a series of 181 bell functions, each centered at one of the 181 angles defining the azimuthal space. The resulting values (see Figure 3.6) are then stored, and the value closest to 0 determines the estimated DoA for each classification.

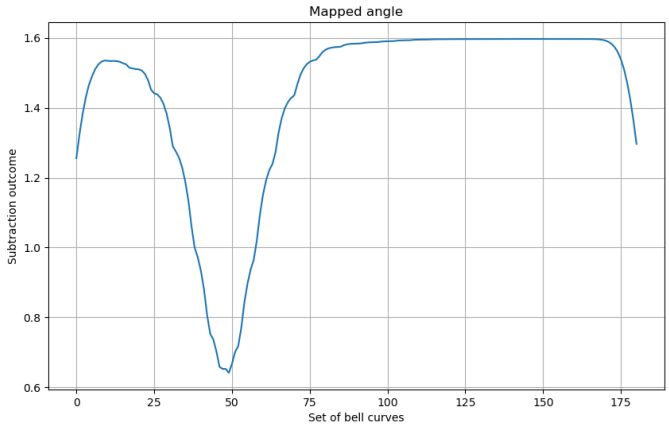


Figure 3.6: Azimuthal mapping

Finally, the array containing all the #-frames classifications is obtained, which will be crucial for the subsequent computation of the E_DoA score.[1]

[1]Many of the scripts used (e.g. Matlab scripts to compute the simulated recordings, or Python scripts to train the NN and apply it to real data) were provided by GN. My contributions primarily involved adjusting the provided base scripts/tools, designing and implementing the experiments, and creating the plots.

3.4 Test Data

As mentioned in section 3.1, the NN was trained using acoustic simulations of spaces. The simulation methods are detailed in section 2.1. The goal is to use the simulation-trained NN to estimate DoA with real-world recordings, which were obtained prior to this master's project. Specifically, three scenarios have been identified for testing the network: *'real_speakers_1'*, *'Python_FCS'*, and *'xeno_sweeps'*. The network has been tested with these scenarios, and the results have been computed accordingly.

1. **real_speakers_1** In this first scenario, recordings of real conversations between people were made in GN meeting rooms. The receiver, a cylindrical mock-up equipped with 8 DPA microphones, was tested in two conditions: close to a wall and on top of a table. The people acted as sound sources from different positions, and their location within the azimuthal space were measured with a digital angle finder protractor. During the sessions, the subjects were allowed to move only their head. Some footage of the measurements is shown in Figures 3.7 and 3.8.

Figure 3.7: Three people talking in GN meeting room

Figure 3.8: Receiver (cylinder mock-up) placed on wall

2. **Python_FCS** In this setting, a fixed Head-And-Torso Simulator (HATS) was used as the sound source, while a rotating P50 microphone array prototype served as the receiver. The HATS played the same speech 19 times per recording, and each time, the P50 was rotated on its axis by a known angle (typically $10°$). Both the source and receiver were placed in a IEC listening room.
3. **xeno_sweeps** In this scenario, both meeting rooms and anechoic rooms were used. A fixed cylinder prototype served as the receiver, and a loudspeaker acted as the sound source. The loudspeaker played the same speech 19 times, each time from a different known angle, typically $10°$. Some footage of these measurements has been reported in Figures 3.9 and 3.10.

Figure 3.9: Anechoic room, source on wall and loudspeaker at $0°$ position

Figure 3.10: Cylinder mock-up on wall

Test metric
During each of these experiments, the ground truth angle a was been measured. This information was then used to compute the DoA error, which is the absolute difference between the neural network's classification outcome and the ground truth angle across each frame n (see Equation 3.2).

$$E\_DoA = \frac{\sum_{m=1}^{L} \sum_{n=1}^{J} \mid a_{m,n} - \hat{a}_{m,n} \mid}{L \cdot T} \tag{3.2}$$

Being L the number of test files, J the number of frames* of file m, a the per-frame ground truth angle, $\hat{a}$ the per-frame DoA estimation and T the total number of frames*. In other words, **E_Doa** is defined as the mean of the computed DoA errors across each frame of every test file. This parameter is crucial for quantifying the network's success in the experiments discussed later in chapter 4.

*Only the frames with a detected voice activity ($VAD \neq 0$) have been included in the computation.

3.5 Experiments list

In the following subsections, a detailed list and description of each experiment are provided. The numbering of each experiment is summarized in Table 3.3.

Exp1	Exp2	Exp3	Exp4	Exp5	Exp6	Exp7
Spectrogram	LP-filter	HP-filter	Time domain	Inner Mics	Outer Mics	Mel

Table 3.3: The list of experiments

3.5.1 Experiment 1: Complex-valued spectrograms

As previously mentioned in section 3.2, spectrograms were the initial data representation used to train the network. Literature suggests that this representation is particularly effective for the task of DoA estimation [1]. Consequently, the neural network was designed to optimize feature extraction from the 2D representation of the data. This design leverages the 2D nature of the convolutional kernels, which process and analyze small, inherently correlated portions of the image iteratively.

The STFT of the recordings was computed by applying *Hann* windows of 512 samples to each time-defined audio channel. To minimize artifacts, each segment (or frame) overlaps

with adjacent frames by $\frac{512}{2} = 256$ samples. Each time-windowed discrete input frame is then transformed using its Discrete Fourier Transform. The mathematics behind the windowing process and the DFT are summarized in Equations 3.3 and 3.4. [17]

$$X\left[k, \lambda\right] = STFT\{x\left[n\right]\} = \sum_{n=0}^{M-1} \tilde{x}\left[n + \lambda R\right] e^{-jkn\frac{2\pi}{M}}, \tag{3.3}$$

with

$$\tilde{x}\left[n + \lambda R\right] = \begin{cases} x\left[n + \lambda R\right] w\left[n\right], \ n = 0, 1, ..., N-1 \\ 0, \ n = N, N+1, ..., M-1 \end{cases} \tag{3.4}$$

Being $X\left[k, \lambda\right]$ the STFT-ed frames (2D representation function of both time and frequency) with k as the frequency bin index ($k \in \{0, 1, ..., M-1\}$) and λ as the frame index ($\lambda \in \{0, 1, ..., L-1\}$); $x\left[n\right]$ a length-N discrete-time audio channel and $\tilde{x}\left[n + \lambda R\right]$ a discrete-time windowed frame being R the shift between adjacent windows, M the zero-padded DFT size, j the imaginary unit, L the number of frames and $w\left[n\right]$ the *Hann* window function (see Equation 3.5).

$$w\left[n\right] = \begin{cases} 0.5 - 0.5cos\left(\frac{2\pi n}{N-1}\right), \ 0 \leq n \leq N-1 \\ 0, \ otherwise \end{cases} \tag{3.5}$$

A spectrogram is visualized by displaying the absolute value of the STFT outcome (see Figure 3.11), which involves retaining only the real part and discarding the imaginary part.

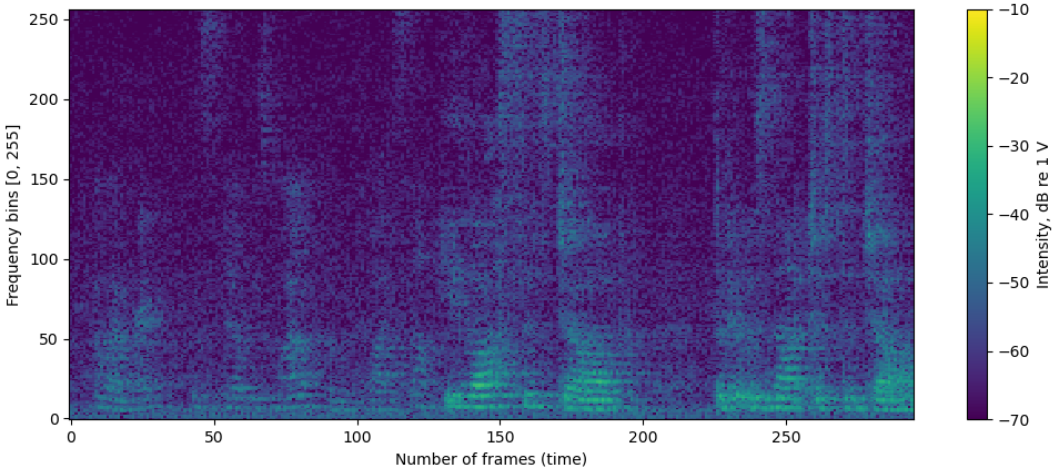


Figure 3.11: Spectrogram of a speech sample (one channel)

In this experiment, a complex-valued spectrogram was used. This means that both the real and imaginary parts of the STFT were preserved, allowing for the retrieval of phase differences across the microphone channels. Although the NN processes the information in its own way (see subsection 2.2.2), the phase information captured by the microphone array is crucial for computing cues such as Interaural Time Differences (ITD), which are essential for sound localization.

Based on the outcomes of this experiment (see section 4.1), subsequent experiments - except for experiment 4 - were designed using this approach. In other words, the training data for these experiments was manipulated in the form of complex-valued spectrograms.

3.5.2 Experiments 2 and 3: LP- and HP- filtered databases

After the first experiment, the following research question emerged: how relevant are the low and high frequency components for the DoA computation? Experiments 2 and 3 were designed to address this question. Specifically, these experiments involved filtering the simulated training data using Low-Pass (LP) and High-Pass (HP) filters, both with cutoff frequencies set to approximately $1.6\ kHz$. The LP and HP filters were implemented by multiplying the relevant frequency bands in the STFT by zero. Two visual representations of the preprocessed data can be seen in Figures 3.12 and 3.13.

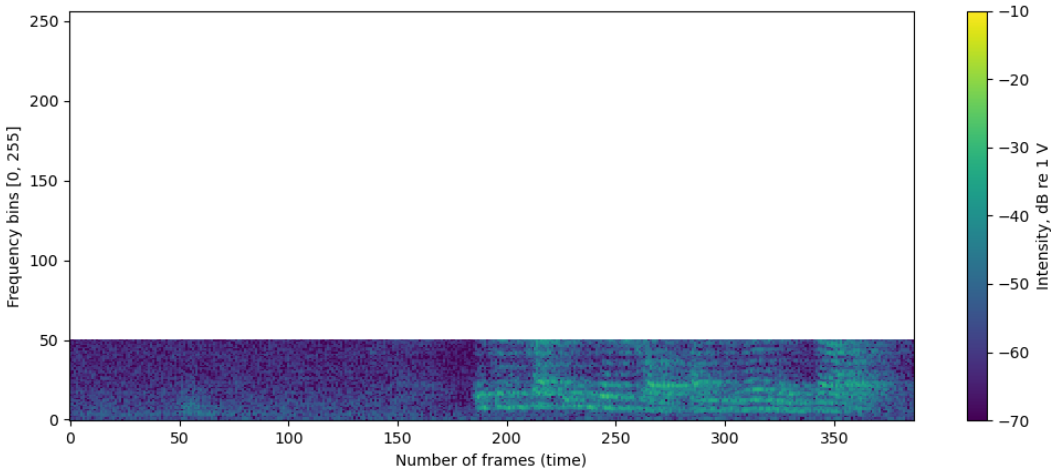


Figure 3.12: LP-filtered spectrogram

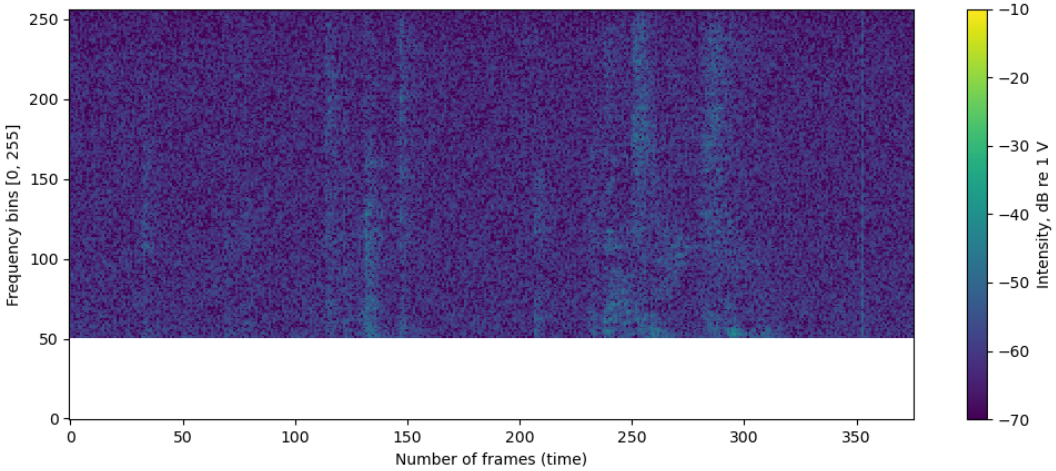


Figure 3.13: HP-filtered spectrogram

The choice of the cutoff frequency at $1.6\ kHz$ is primarily based on the design of the simulation methods. This frequency allows for the study of the NN's response to rooms exhibit-

ing either low-frequency or high-frequency behaviors exclusively. Specifically, $1.6\ kHz$ is significant because it marks the threshold in the spectrum where the hybrid method transitions from being modeled with WB to GA methods (subsection 2.1.1). In essence, the selected cutoff frequency serves as a powerful tool to analyze different components of simulated the data. For example, it facilitates comparison of the performance of the different GA methods utilized by the two simulation methods (subsection 2.1.2 for ISM and subsection 2.1.3 for RR) above this frequency threshold.

3.5.3 Experiment 4: Time domain input

After analyzing the results from the first three experiments, it was deemed necessary to revisit the assumption of disregarding time-domain analysis. As usual, the results will be presented in the following chapters (see section 4.3). In this experiment, the preprocessed data consisted of raw *.wav* audio recordings, rather than STFT representations. To align with the input shape expected by the neural network, the time-domain signal was windowed. Each frame was thus 257 samples long with a hop size of 256 samples, resulting in an overlap of only $257 - 256 = 1$ sample between adjacent frames. A visual representation of this preprocessed data is shown in Figure 3.14.

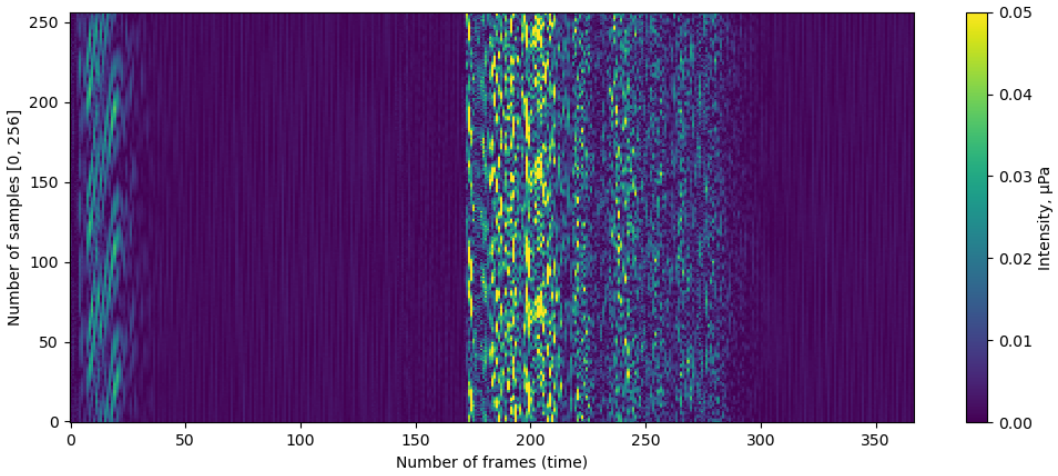


Figure 3.14: Time domain representation

In other words, the data structure provided to the NN consisted of 4 batches of 3-D data chunks. These chunks were defined by the frame number of each framed recording and by the values contained within each audio sample in the frame. The third dimension of the data represented the number of channels, with 8 channels per recording. To match the network's input shape requirements (see section 3.2), this was doubled to 16 channels. Consequently, the input data was entirely real-valued and not complex. While this approach might not be the most efficient, it effectively achieved the intended purpose.

3.5.4 Experiments 5 and 6: Microphone subsets

Using an 8-channel microphone array is indeed a high-end setup, and it's important to assess how well the system performs with fewer microphones, as such an array is not always available. The goal of this experiment was to challenge the initial assumption of having all microphones available and to understand the impact of microphone configuration on the data. In particular, the approach was to halve the number of microphones by multiplying the channels of the discarded microphones by zero. Experiment 5 involved

discarding the information from the outer four microphones of the array, while Experiment 6 involved discarding the information from the inner four microphones. Visualizations of the adopted arrays are provided in Figures 3.15 and 3.16.

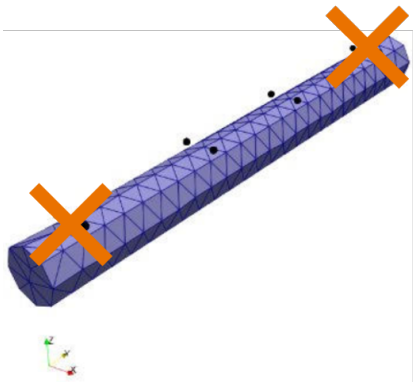


Figure 3.15: (5) Zeroed outer microphones

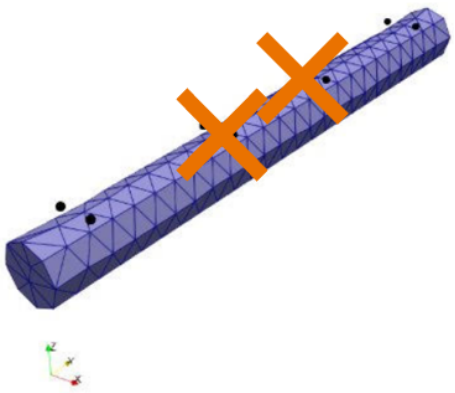


Figure 3.16: (6) Zeroed inner microphones

In other words, the input data comprised 16-channel complex-valued spectrograms (see subsection 3.5.1), but only half of these channels contained useful signal information.

3.5.5 Experiment 7: Mel Spectrograms

At this stage of the project, with many results already available, an intuition emerged to create a version of the spectrogram that emphasizes low-frequency components while reducing the importance of high-frequency components. To achieve this, a mel-scaled spectrogram solution was implemented. A mel spectrogram is a variant of the traditional spectrogram that uses a perceptually relevant frequency scale, known as *mel* scale, which better aligns with human auditory perception of frequency.

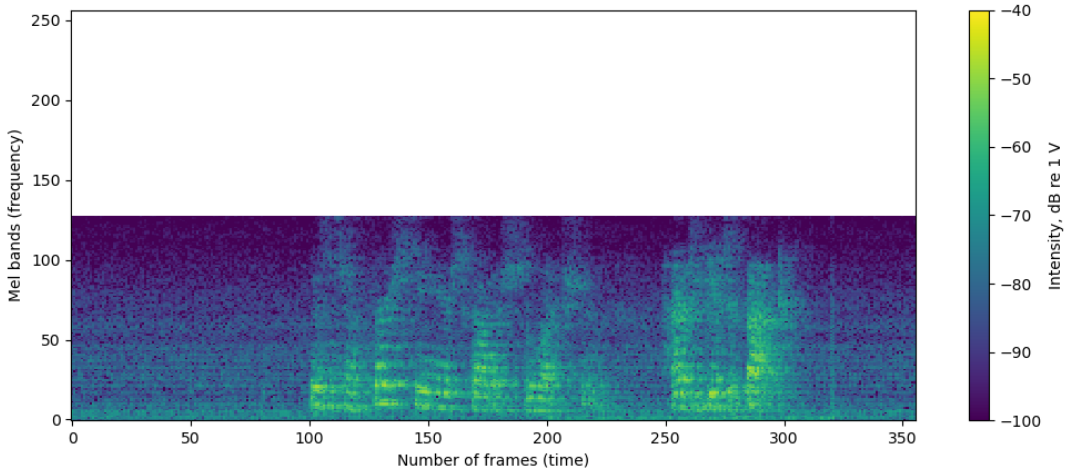


Figure 3.17: Input mel spectrograms

The computation of the mel spectrogram involves a matrix multiplication between each complex-valued spectrogram and a mel filterbank [18]. Given that 257 mel bands cover a broad portion of the spectrum, the filterbank was configured with 128 mel bands. To match the NN's input structure requirements, an additional $257 - 128 = 129$ zeroed bins were

appended to the array. This adjustment was made to maintain the original dimension of the neural network. Again, although this solution might not be the most efficient, it effectively serves its purpose and its results are visualized in Figure 3.17.

4 Results and first comments

As introduced at the end of section 3.4, the primary metric used to measure the network's performance in the DoA task was the mean of the **angle deviation errors** computed across all classifications. Specifically, by feeding the NN with real-world recordings, numerous per-frame classifications are obtained. These classifications are then compared to the actual measured DoA by calculating the absolute difference between the two values (with smaller differences indicating better performance). The results are subsequently presented in a bar plot, categorized by experiment and database.

In the following sections an overview of the results and an initial interpretation are provided. A thorough discussion is then presented in chapter 5.

4.1 Experiment 1: complex-valued spectrograms

Following the methods explained in subsection 3.5.1, the NN was trained using complex-valued spectrograms inputs. The results, averaged across all test files, are shown in Figure 4.1.

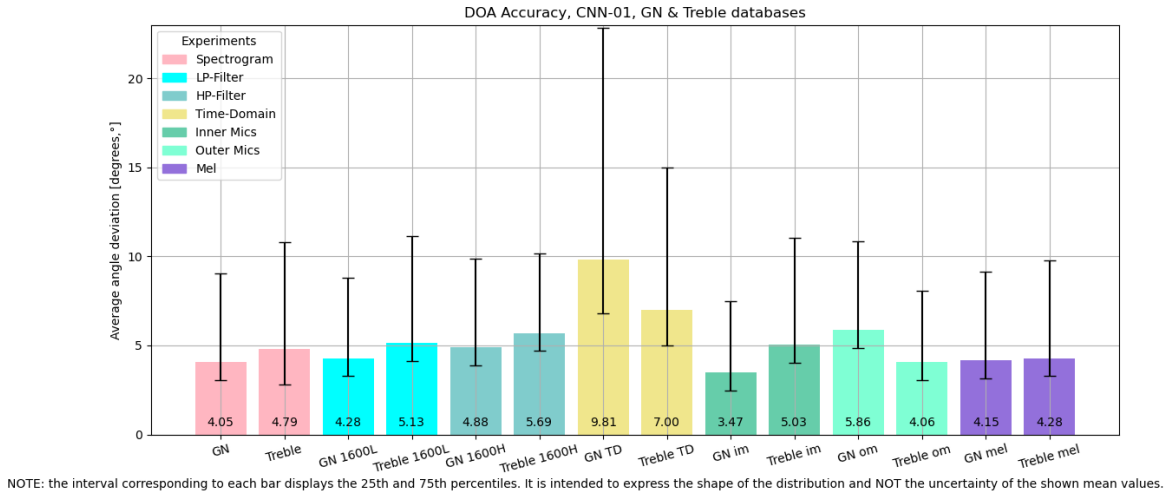


Figure 4.1: Outcome of the experiments - average angle deviation

Several methods to express the uncertainty of the distributions have been tested (see visualization in Figure 5.1). Given the wide distribution of the experiments and their probability density function resembling a power-law trend, the standard deviation was too high to be meaningfully represented in the bar plot. Conversely, due to the high number of data points (approximately 7 millions), the standard error was too small ($< 1°$). Therefore, the bar plot displays the first and the third quartiles on top of the means of the distribution, representing the interval within which the central 50% of the data falls. This approach effectively captures the power-law nature of the distributions.

The NNs successfully performed the DoA estimations in all test cases with an average error of less than $5°$. Results are stratified by database: the same neural network was separately trained using either ISM- or hybrid-simulated training data and then tested with

the same population of test data to provide qualitative results for the two different methods. The plot indicates that the mean value of the ISM-trained NN is slightly smaller than that of the hybrid-trained NN.

4.2 Experiments 2 and 3: LP- and HP- filtered databases

Following the methods explained in subsection 3.5.2, complex-valued spectrograms of (2) low-pass filtered and (3) high-pass filtered simulated recordings were used as the NN's training data. The results, averaged across all frames of the test files, are shown within Figure 4.1.

As introduced in subsection 3.5.2, the cutoff frequency for both filters was chosen to be $1.6\ kHz$, the threshold frequency where the hybrid method changes its behavior. Consequently, for both experiments and both databases, the reduced amount of information in the training data led to a slight overall increase in E_DoA. A further discussion is provided within chapter 5.

4.3 Experiment 4: Time domain input

Following the methods explained in subsection 3.5.3, data blocks of time-domain audio information were fed into the NN. The results, averaged across all test files, are shown within Figure 4.1.

By training the NN with time-domain data only, an overall increase in the E_DoA metric was observed, suggesting that the current NN may not receive sufficient training with this type of data alone. Further comments are provided in chapter 5.

4.4 Experiments 5 and 6: Microphone subsets

Following the methods explained in subsection 3.5.4, complex-valued spectrograms were fed into the NN for training. However, only half of the channels contained useful signals, while the other half contained zeros. First (5), the four central microphones of the array have been preserved, and second (6), only the four external microphones were considered. The NN trained with this data was then tested with real-world recordings, partially zeroed in the same way. The results, averaged across all frames of the test files, are shown in Figure 4.1.

Interesting patterns emerged from these experiments: the ISM-based training resulted in a relatively low angle deviation for experiment 5, despite a relatively high angle deviation for experiment 6; the hybrid-trained NN showed an inverted trend. Possible interpretations will be discussed in chapter 5.

4.5 Experiment 7: Mel Spectrograms

Following the methods explained in subsection 3.5.5, mel-scaled spectrograms were fed into the NN. The results, averaged across all frames of the test files, are shown in Figure 4.1. Although the average angle deviation is generally low ($< 4.3°$), it is not the lowest observed. However, this experiment shows a flattened intra-database difference in the metric. Additional comments will be provided in chapter 5.

5 Overall discussion

To perform meaningful analysis based on Figure 4.1, a statistical test was conducted across the two distributions resulting from the seven experiments. Given the frame-level errors consist of approximately 7 million data points per experiment, forming a non-Gaussian distribution, the *Mann-Whitney U* test was employed to ensure that the distributions are actually different. The *Mann-Whitney U* test is a nonparametric test of the null hypothesis that the distribution underlying sample *x* is the same as the distribution underlying sample *y* [19]. In essence, this test helps reject that any observed difference between the distributions is merely due to noise. To account for the lack of matching pairs and the fact that data points originate from the same population, the test was repeated 10 times per experiment. Each iteration was conducted on randomly [20] selected subsets of 50000 data points from the original distributions. The 10 resulting *p-values* were then averaged, and the results are presented in Table 5.1.

	Exp1	Exp2	Exp3	Exp4	Exp5	Exp6	Exp7
p-value	≈ 0.0	≈ 0.0	≈ 0.0	≈ 0.0	≈ 0.0	≈ 0.0	≈ 0.0

Table 5.1: Outcome of *Mann-Whitney U* test

By accepting the assumptions of the *Mann-Whitney U* test [21], we can confidently reject the null hypothesis regarding the two distributions (ISM-based and hybrid-based). This allows us to assert that any differences between the pairs of distributions in Figure 4.1 are statistically significant.

The following sections will provide a detailed discussion of the results presented in chapter 4.

5.1 Spectrogram experiment

Based on Figure 4.1, it appears that the ISM-based simulation method, despite being entirely a GA method, is not outperformed by the hybrid-based method in terms of DoA estimation. The ISM-based method has a slightly lower average angle deviation error (by $4.79 - 4.05 = 0.74$) and more robust classifications, indicated by the smaller percentile intervals. This warrants further investigation into which particular scenarios yielded the highest errors, prompting the discussion within subsection 5.8.1.

In the spectrogram experiment alone, the ISM-trained NN estimated the DoA with an average error of almost $4°$. While this is a promising result, it is important to note that speakers may sometimes be positioned within a smaller azimuthal angle relative to the microphone, necessitating an even smaller error margin for precise localization. The quantile intervals indicate that the hybrid-based NN produced more diverse results, whereas the ISM-trained NN demonstrated greater robustness in terms of classification variety.

5.2 Low-Pass filter experiment

Before conducting this and the subsequent experiment, we posed several questions: how relevant are different portions of the spectrum for DoA estimation? Additionally, are the low and/or high-frequency components of the simulations accurate, or do they contain

any flaws? Our expectation was that by removing some amount of information, performance would decrease. If so, by how much? Alternatively, could performance improve, suggesting that certain parts of the spectrum might be misleading for the task?

In this experiment, filtering out higher frequencies slightly increased the average angle deviation compared to the first experiment. However, this increase was minimal ($\approx 0.3°$ on average), indicating that frequencies above $1.6\ kHz$ play a minor role in the task. Even so, for the hybrid-trained NN, high frequencies seem more significant than for the ISM-trained NN. After filtering, the score for the hybrid-trained NN increased by $0.34°$ (from $4.79°$ to $5.13°$), compared to an increase of $0.23°$ (from $4.05°$ to $4.28°$) for the ISM-trained NN. The percentile intervals of the distributions are consistent with those from the previous experiment.

5.3 High-Pass filter experiment

Similarly to the previous case, filtering out frequencies below $1.6\ kHz$ results in a further overall increase in the average angle deviation. This metric is higher than in the LP experiment, suggesting that higher frequencies generally have less impact on DoA estimation compared to lower frequencies. Compared to the first experiment, the increase in average angle deviation is slightly greater for the hybrid-based NN ($0.9°$, from $4.79°$ to $5.69°$) than for the ISM-based NN ($0.83°$, from $4.05°$ to $4.88°$). This implies that ISM-based training may be more robust, as filtering out different frequencies causes a smaller increase in error compared to the hybrid-trained case. Furthermore, the shrinkage of both percentile intervals may suggest that lower frequencies are more likely to estimate DoAs with the highest deviation from the mean value. However, this is not confirmed by the percentile intervals from the previous experiment. Interestingly, the percentile interval of the hybrid-trained NN is the smallest observed so far, even smaller than that of the ISM-trained NN. This indicates that, at least in the hybrid-based scenario, lower frequencies are indeed responsible for a wider distribution of classification errors.

5.4 Time domain experiment

As introduced in subsection 3.5.1, the literature suggests that a time-frequency representation could be the most effective for DoA estimation. In the current experiment, we are exploring the time-domain case to verify the solidity of our conclusions and to investigate the results obtained from purely time-defined input data.
The results (Figure 4.1) confirmed that using a purely time-domain trained CNN for DoA estimation is likely to fail, yielding an average angle deviation of approximately $8°$. However, in this scenario, the hybrid-based database seems to perform better, indicating that ISM-based training is less robust to time-only input data. The hybrid-based method may provide more accurate phase estimation, resulting in better scores. This is also suggested by the smaller percentile intervals for the hybrid-trained NN. Furthermore, the overall worsening of the scores may confirm that the designed CNN, which employs a 2D 3×3 kernel filters for features extraction, is better suited for 2D representation of the data (such as the time-frequency domain). Thus, further experimentation in this domain could be relevant, utilizing a DNN architecture fully adapted to process sequential data (instead of a grid of values) and better suited for time-domain input. This could be achieved by employing Recurrent Neural Networks, specifically Gated Recurrent Units [10]).

5.5 Inner Microphones experiment

As illustrated in subsection 3.5.4, only the central microphones of the microphone array were used, while the signals from the remaining channels were zeroed out both during

training and testing.
While the hybrid-trained NN showed a slight worsening in performance compared to the first experiment (an increase of $0.24°$ from $4.79°$ to $5.03°$), this decrease was unexpectedly small. Surprisingly, the ISM-based training produced the best results so far in terms of average angle deviation. Specifically, the NN estimated the DoA with an average angle error of $3.47°$ across the test data, and the percentile interval suggests that these estimations are also more robust. Contrary to the expectation that removing more information would result in higher error, the ISM-based NN did not follow this trend. One possible explanation is that ISM does not accurately estimate conditions at the boundaries of the microphone array, leading to potentially misleading data for DoA estimation. This discrepancy is likely due to a modeling mismatch between the two simulation methods and reality. The hybrid simulation models the microphones as part of a polygonal cylinder (see Figure 3.3), whereas the ISM approximates the microphones as suspended points, neglecting complex acoustic artifacts caused by the microphone array's structure. Further experimentation could explore this issue in greater detail. For example, creating artificial test sets with varying levels of mismatch could help increase our understanding of the problem.

5.6 Outer Microphones experiment

Similarly to the previous case, as illustrated in subsection 3.5.4, only the external microphones of the microphone array were used, while the signals from the four central microphones were zeroed out both during training and testing.
The ISM-trained NN returned a relatively high average angle deviation E_DoA of $5.86°$, which may confirm the previous conclusion: boundary conditions at the edges of the microphone array can introduce artifacts into the sound recorded by the outer microphones, leading to lower DoA accuracy when relying solely on them. In contrast, the hybrid-trained NN performed the best so far, with an average angle deviation of $4.06°$. This experiment also gave the NN some robustness in its classifications, as indicated by the small percentile interval. The interpretation of these results is not straightforward. However, excluding information from the four central microphones seems to make DoA finding easier for the hybrid-trained NN. One possible explanation could be that the outer microphones were more robustly simulated compared to the central ones in the hybrid model. Additional experimentation could be directed to explore this further.

5.7 Mel Spectrograms experiment

As illustrated in subsection 3.5.5, complex spectrograms were used to compute mel-scaled spectrograms. Compared to the first experiment, the performance differences between the ISM and hybrid-trained NNs have nearly flattened, with both averaging around $4.2°$ in angle deviation. By emphasizing lower frequencies and filtering out some higher frequencies, the hybrid-trained NN showed a slight improvement of $0.51°$ in average angle deviation, whereas the ISM-trained NN exhibited a slight worsening of $0.1°$. These results support the interpretations from the filtering experiments (2 and 3): not all frequency components are necessary for achieving reasonably good DoA estimation results, and lower frequencies appear to be the most relevant. The improvement in the hybrid-trained NN suggests that some very high frequencies may introduce issues in DoA estimation. However, some frequency components above $1.6\ kHz$ are still useful for the task and should not be entirely filtered out, as seen in experiment 2. This experiment demonstrated that some spectrum information is either unnecessary or actually misleading for DoA estimation. Further filtering experiments could help identify the optimal cutoff frequencies. Although converting to a mel-scaled frequency representation may not significantly impact

the metric, it provides a basis for further experimentation (e.g., Mel-frequency cepstral coefficients), which could yield more interesting results. Finally, the percentile intervals are consistent with those from the first experiment, indicating that they are not significantly affected by this approach.

5.8 Plotting distribution of uncertainty

The same data shown in the bar plot in Figure 4.1 is displayed with a raincloud plot in Figure 5.1. Each experiment's plot combines the probability density function and the box plot representation of all data points (i.e. DoA estimation errors). This type of plot more clearly shows the distribution of errors. The box plot indicates the median and the 25th and 75th percentiles of the distributions, while also providing information on the sparsity of the data points and the extremely high standard deviation. As previously mentioned, the distribution resembles a power-law, with most data points lying at low values and a very long tail towards higher values.

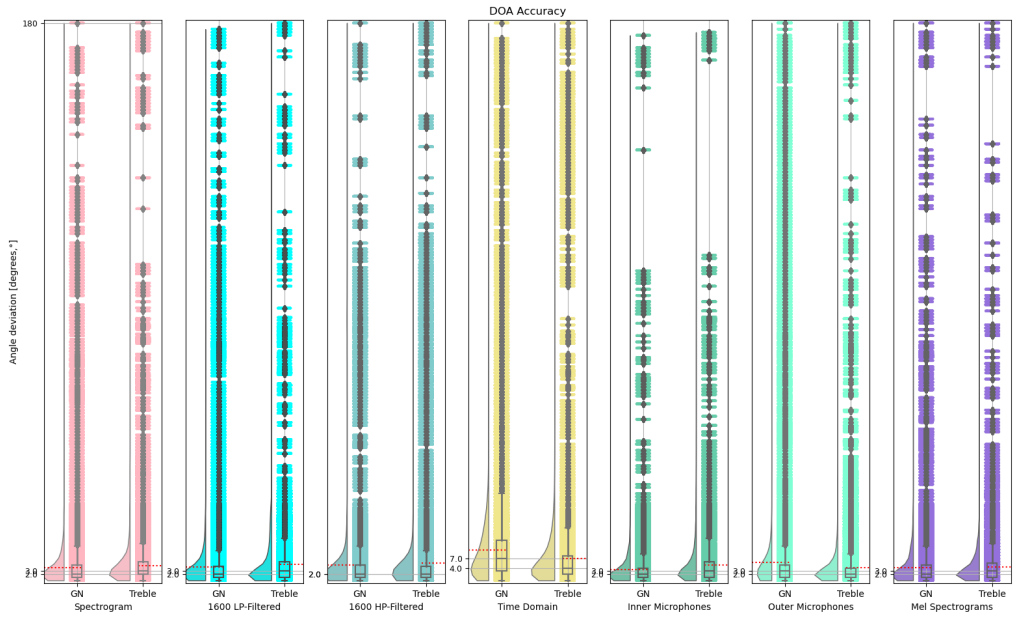


Figure 5.1: Reinterpretation of results of Figure 4.1

All the plots in Figure 5.1 have the same linear y-axis range between $[0\ 180]$, where the distributions lie. As a reference, the average angle deviation described in Figure 4.1 is also shown with a red dashed line.
It is interesting to observe that, for some experiments, there are no data points within certain angle deviations. For example, in experiment 5, the NN did not return any estimated DoA differing from the real DoA by around $120°$.

5.8.1 Stratified results across different test databases

As previously mentioned (see section 3.4), the final E_DoA parameter results from applying the trained NN to real-world recordings. The closer the match between training and test data conditions, the better the network performs. In this project, we have three scenarios: *python_FCS*, *real_speakers*, and *xeno_sweeps*; which are described in section 3.4. The E_DoA score for each real-world database is shown in Figure 5.2. Their

mean, represented in Figure 4.1, is displayed as the *Overall E_DoA* column. Note that this is not a mean of the three databases, but an average across all corresponding frames (since different test scenarios consist of different numbers of frames).

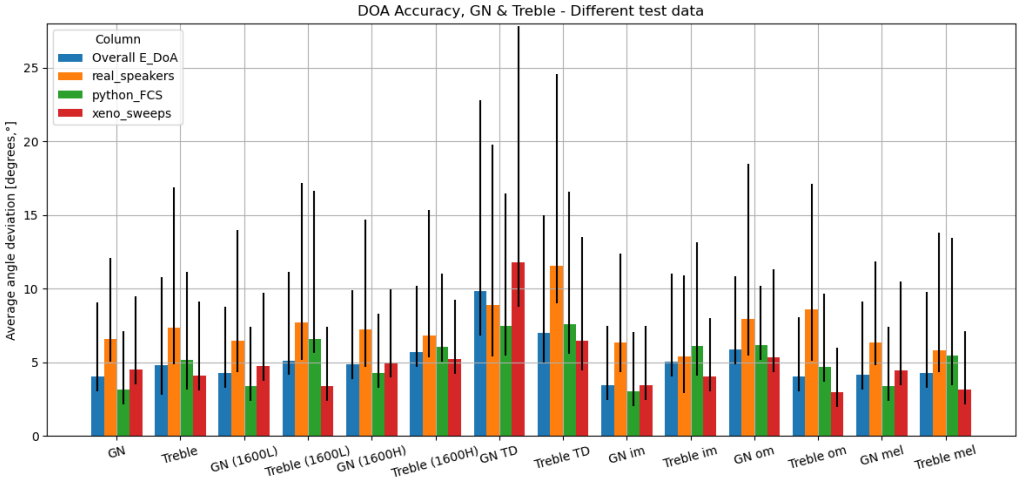


Figure 5.2: Different performance across different test databases

Visualizing the data this way provides various insights.
Generally, it appears that the *python_FCS* database is responsible for the overall worsening of the hybrid-trained NN performance compared to the ISM-trained NN. This was discovered because the database contains many recordings where the microphone array was placed close to a wall — a realistic situation. However, the hybrid simulations never placed any receiver closer than $20\ cm$ to a wall, while the ISM simulations placed receivers as close as $5\ mm$ from the walls. These close reflections are expected to significantly influence intra-microphone differences, illustrating a train/test mismatch. Additional steps could be taken later in the project to prove this point, and if confirmed, further steps may be taken to improve the hybrid-based simulations' settings. Furthermore, it seems that the *real_speakers* database generally results in worse performance compared to the others, likely due to it being the most "uncontrolled" setting, but also the most realistic. The *xeno_sweeps* database, being the largest, has the most influence on the overall result.

5.9 DoA error per angle

Additional visualizations displaying the error per angle have been provided within this section. Specifically, the idea is to quantify how likely the NN is to fail the DoA classification task per each azimuthal angle (with a $5°$ resolution). These plots aim to explore the data differently, potentially leading to new conclusions. Two plots per experiment are provided, each with the same y-axis for consistency.

Spectrogram experiment
The first scenario involves feeding the NN with complex-valued spectrograms of the training data (see subsection 3.5.1), which consists of simulated recordings based on two simulation methods described in section 2.1.

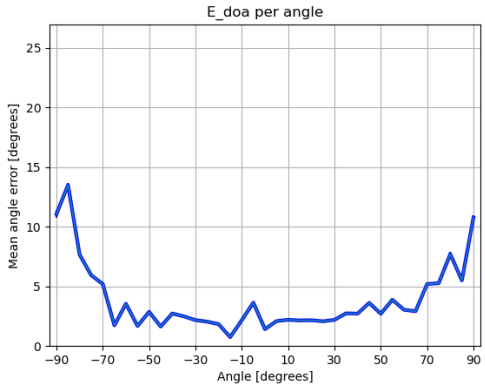


Figure 5.3: Exp1 - ISM-based training

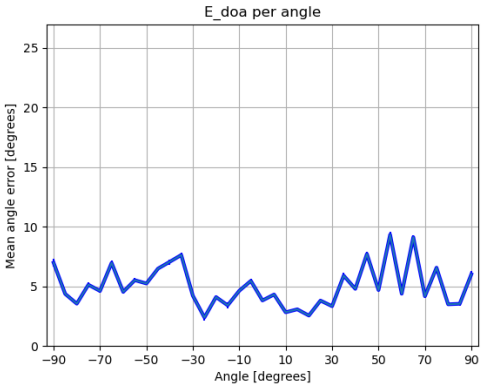


Figure 5.4: Exp1 - hybrid-based training

From Figures 5.3 and 5.4, it is evident that the ISM-trained NN performs poorly around $\pm 90°$, indicating potential artifacts when simulating sound arriving endfire (i.e., directions near the longitudinal axis of the microphone array). Conversely, both ISM and hybrid-trained NNs show no significant issues in capturing sound from broadside directions (i.e., when the maximum radiation is perpendicular to the plane of the array). These observations may not necessarily reflect a deficiency in the NN itself but rather highlight inherent characteristics of the array geometry. Arrays typically struggle with endfire directions due to challenges in accurately capturing phase and wavenumber differences in the signals [22]. However, the hybrid-trained NN appears generally more robust across azimuthal angles, although it does exhibit issues around $65°$ and $-35°$ directions of arrival.

Low-Pass filter experiment

The findings from the LP experiment (see subsection 3.5.2) reinforce the points made in the previous paragraph: for the ISM method, significant issues are observed around the longitudinal directions ($\pm 90°$ and $\pm 80°$), as shown in Figure 5.5. Conversely, the hybrid method encounters challenges around mid-plane directions ($\pm 65°$, $\pm 55°$, $\pm 45°$), depicted in Figure 5.6.

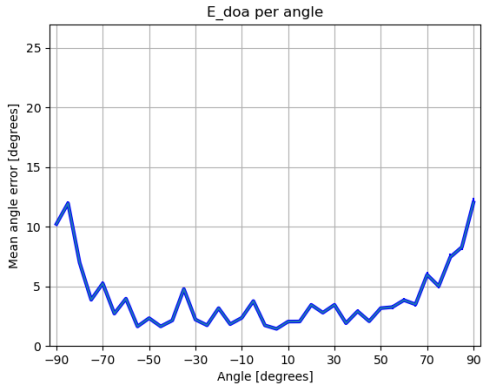


Figure 5.5: Exp2 - ISM-based training

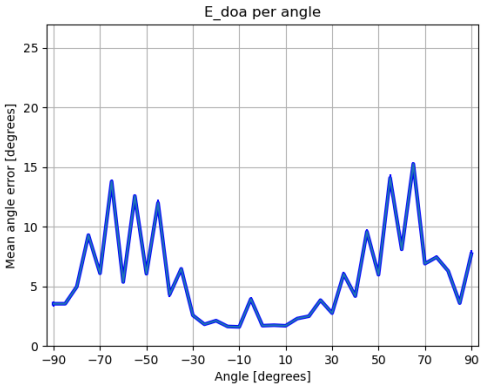


Figure 5.6: Exp2 - hybrid-based training

These results indicate persistent difficulties in accurately simulating sound arrival angles near these azimuthal angles using both simulation approaches.

High-Pass filter experiment
Of particular interest is the next experiment (subsection 3.5.2). Figures 5.7 and 5.8 highlight significant issues for the ISM-trained NN around angles like $-85°$, $-75°$, and $80°$, whereas the hybrid-trained NN shows notably high errors at angles such as $-75°$, $-65°$, $-45°$ and $70°$. However, it's important to note that the peaks with the highest errors also exhibit higher standard errors, indicating potential variability due to a smaller number of data points. Once again, the lowest errors are consistently observed around the normal direction ($0°$).

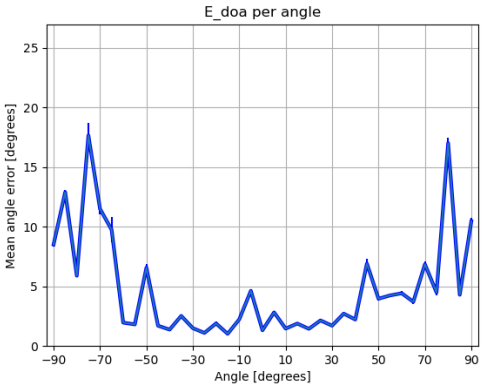


Figure 5.7: Exp3 - ISM-based training

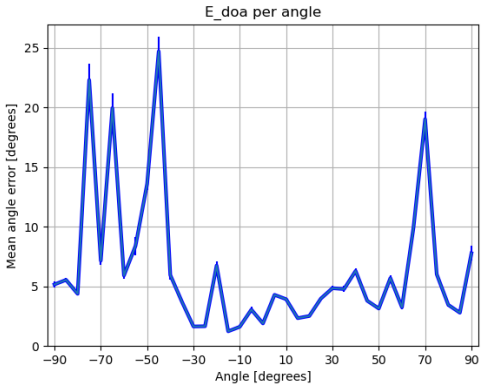


Figure 5.8: Exp3 - hybrid-based training

These results may partly stem from artifacts related to spatial aliasing. In order to prevent spatial aliasing, one should sample at half of the wavelength corresponding to the smallest wavelength (or highest temporal frequency) of interest [23]. Therefore, resulting arrays should be very limited in spatial extent - for example, a two-element array should be spaced no more than $d < \frac{343}{8000}/2 = 0.0214\ m = 2.14\ cm$ apart to prevent aliasing for up to $8\ kHz$. Spatial aliasing is a well known problem, especially given the width of the operating spectrum and the size of the microphone array of this scenario (see section 3.1). However, this primarily affects narrow-band signals, and its impact on broadband signals like speech is less straightforward due to a number of parameters modulating. Unless a wide-band signal possesses a strong harmonic component, spatial aliasing is not experienced with broadband signals, and one cannot simply "superimpose" narrow-band array results onto broad-band arrays without careful thought [23].

Time domain experiment
The time domain experiment (subsection 3.5.3) resulted in highest errors, consistent with the findings in Figure 4.1. Specifically, for the ISM-trained NN (see Figure 5.9), there is a gradual increase in average angle error as the direction moves away from the normal direction at $0°$. Similarly, for the hybrid-trained NN (see Figure 5.10), significant issues arise around $-55°$, and also at $55°$, $65°$, $80°$, and $90°$.

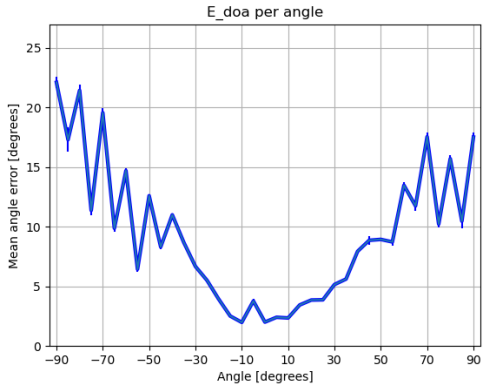


Figure 5.9: Exp4 - ISM-based training

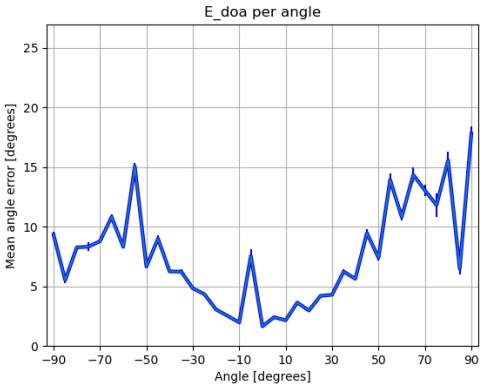


Figure 5.10: Exp4 - hybrid-based training

Inner Microphones experiment

The results from the inner microphones experiment (subsection 3.5.4) appear cleaner compared to the previous experiment. In Figure 5.11, we observe very low error values, with noticeable issues around $-85°$ and $90°$. This suggests that the outer microphones may be responsible for the high-frequency artifacts seen in Figure 5.7, which have now disappeared. Conversely, in Figure 5.12, most of the issues occur around $\pm 65°$, $\pm 55°$ and $\pm 45°$, consistent with the observed DoA errors for the hybrid-trained NN in the second experiment.

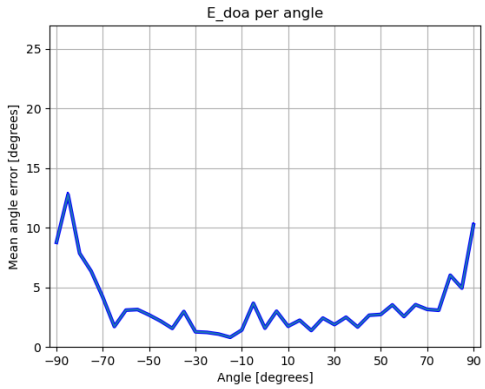


Figure 5.11: Exp5 - ISM-based training

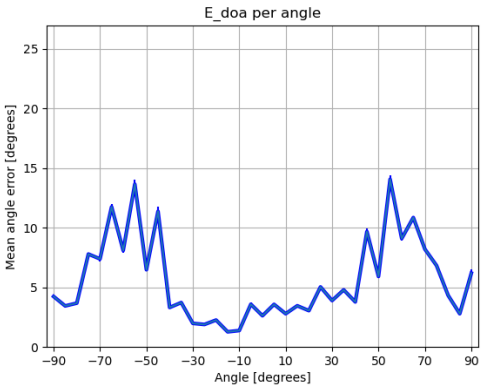


Figure 5.12: Exp5 - hybrid-based training

Outer Microphones experiment

The outer microphones experiment (subsection 3.5.4) reveals an inversion of trends compared to its counterpart, consistent with the results in subsection 3.5.5. Specifically, the acoustic field around the outer microphones appears to have been inaccurately estimated by the ISM-trained NN. In Figure 5.13, the lowest error peaks are again around the normal direction, while the highest errors occur at around $\pm 85°$. Additional inaccuracies are observed at $55°$ and $75°$, although their high standard deviation suggests they may be less reliable or have a lower impact on overall performance scores. In contrast, hybrid-trained NN (Figure 5.14) shows significantly lower errors compared to the complementary experiment. However, issues persist around the typical angles of $\pm 75°$, $65°$ and $\pm 55°$.

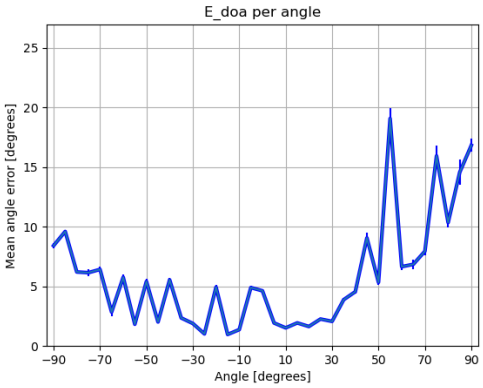


Figure 5.13: Exp6 - ISM-based training

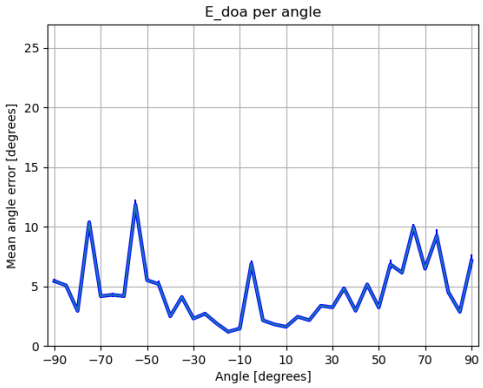


Figure 5.14: Exp6 - hybrid-based training

Mel Spectrograms experiment

The final experiment (subsection 3.5.5) yielded results very similar to those of experiment 2, which is consistent with the experimental setup. Generally, the E_DoA scores are low around the normal directions, but increase due to typical artifacts observed around $\pm 90°$ and $\pm 85°$ for the ISM-trained NN (see Figure 5.15). Similarly, hybrid-trained NN exhibits issues at azimuthal angles of $\pm 75°$, $\pm 65°$ and $\pm 55°$.

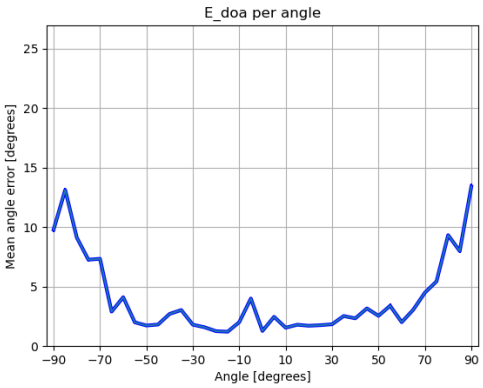


Figure 5.15: Exp7 - ISM-based training

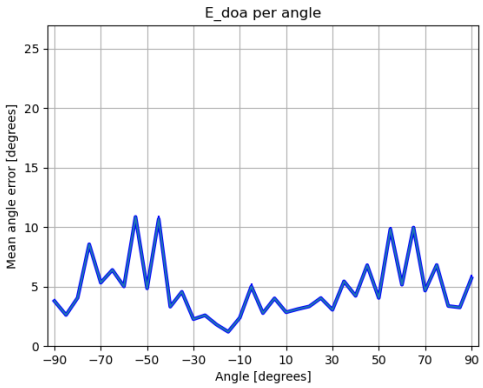


Figure 5.16: Exp7 - hybrid-based training

In conclusion, this type of visualization proves invaluable for identifying precise issues that might be overlooked with other types of analysis. The observations from these experiments highlight specific challenges: the ISM method introduces artifacts or inaccuracies in estimating the acoustic field, notably affecting DoA estimation for sounds arriving longitudinally to the microphone array ($\pm 85°$). Conversely, the hybrid-based simulation algorithm appears to struggle with simulating sounds arriving from azimuthal angles around $\pm 55°$. Moving forward, several steps can be considered for the next phase of this project to delve deeper into understanding and addressing these challenges. Targeted experiments aimed at debugging data simulations could provide insights into improving accuracy in these critical azimuthal spaces. Nevertheless, these experiments also underscore the reliability of these simulation methods across other portions of the azimuthal space, where performance remains robust.

6 Conclusion

Sound Source Localization (SSL) is the problem of estimating the position of one or several sound sources relative to the position of the recording microphone array, based on a recorded multichannel acoustic signals. Traditional SSL approaches rely on SP techniques, which, despite their historical advancements, often falter in complex real-world scenarios characterized by noise, reverberation, and multiple simultaneous sound sources. Over the past decade, the emergence of data-driven DL methods has garnered significant attention for tackling these challenging circumstances. This study, simplifying the SSL problem to estimating the DoA of the sources within the azimuthal plane, aims to employ a CNN to compute the DoA of simulated recordings.

Large databases of RIRs and anechoic recordings have provided the basis for generating simulated recordings. The project's objective was to train a CNN using these simulated recordings to enable it to estimate DoA in real-world scenarios. Two distinct simulation methods were employed to generate the training data. Numerous experiments were conducted and analyzed, focusing on identifying the most critical components of the simulated data to address the DoA problem. Key findings include:

- The CNN's performance with STFTed input recordings was initially assessed, demonstrating the feasibility of addressing the DoA problem with this setup. This served as a baseline for comparing subsequent experiments;
- Filtering operations were applied to the training data, revealing that low frequencies in simulated recordings alone were sufficient for effective DoA estimation, often more significant than high frequencies;
- A time domain experiment was conducted to validate the CNN's architecture for the DoA task;
- Halving the number of microphones in the array showed that not only using fewer (than 8) microphones may suffice for the task, but also highlighted potential improvements in receiver simulation methods;
- An experiment in the Mel-frequency domain reaffirmed the findings of earlier experiments, marking a foundation for future project developments in this domain.

These experiments and discussions have provided valuable insights and potential directions for future research in the field. Deep learning has demonstrated its effectiveness in tackling the DoA problem, suggesting further promising advancements ahead.

Bibliography

[1] Pierre-Amaury Grumiaux et al. *A survey of sound source localization with deep learning methods*. 2022. URL: https://pubs.aip.org/asa/jasa/article/152/1/107/2838290.

[2] Shari R and Sarah Jacob. "Comparative study of DOA estimation algorithms". In: *2022 IEEE 19th India Council International Conference (INDICON)*. 2022, pp. 1–4. DOI: 10.1109/INDICON56171.2022.10040076.

[3] Erin Driscoll et al. "Data generation with device-modeling using Treble's hybrid cloud-based system". In: *Audio Engineering Society* (2023).

[4] Finnur Pind. *Wave-Based Virtual Acoustics*. DTU, 2020.

[5] Anders Melander et al. "Massively Parallel Nodal Discontinous Galerkin Finite Element Method Simulator for Room Acoustics". In: *DTU* (2016).

[6] Markus Faustmann and Joachim Schöberl. "Numerics of PDEs - Lecture Notes". In: *Institute of Analysis und Scientific Computing TU Wien* (2022).

[7] J. B. Allen and D. A. Berkley. "Image method for efficiently simulating small□room acoustics". In: *The Journal of the Acoustical Society of America* (1979).

[8] Treble Technologies. *The Image-Source Method*. 2024. URL: https://docs.treble.tech/geometrical-solver/image-source-method.

[9] Treble Technologies. *The Ray-Radiosity Method*. 2024. URL: https://docs.treble.tech/geometrical-solver/ray-radiosity.

[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[11] Gaël Richard et al. "Audio Signal Processing in the 21st Century". In: *HAL Open Science, IEEE Signal Processing Magazine* (2023).

[12] The University of Edinburgh. *VCTK*. 2024. URL: https://datashare.ed.ac.uk/handle/10283/2950.

[13] Tobias May. "Course 22001: Acoustic signal processing. Lecture 03 : Linear system analysis". In: *DTU - Hearing Systems Group* (2022).

[14] The Linux Foundation. *PyTorch*. 2024. URL: https://pytorch.org/.

[15] The PyTorch Foundation. *Adam*. 2024. URL: https://pytorch.org/docs/stable/generated/torch.optim.Adam.html.

[16] The Linux Foundation. *BCEWithLogitsLoss*. 2024. URL: https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html.

[17] Tobias May. "Course 22001: Acoustic signal processing. Lecture 02 : Fourier analysis". In: *DTU - Hearing Systems Section* (2022).

[18] Librosa development team. *librosa.feature.melspectrogram*. 2023. URL: https://librosa.org/doc/main/generated/librosa.feature.melspectrogram.html.

[19] The SciPy community. *scipy.stats.mannwhitneyu*. 2024. URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.mannwhitneyu.html#scipy.stats.mannwhitneyu.

[20] NumPy Developers. *numpy.random.choice*. 2024. URL: https://numpy.org/doc/stable/reference/random/generated/numpy.random.choice.html.

[21] Laerd Statistics. *Assumptions of the Mann-Whitney U test*. 2018. URL: https://statistics.laerd.com/statistical-guides/mann-whitney-u-test-assumptions.php.

[22] Harry Van Trees. *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*. eng. WILEY Online Library, 2002, pp. 17–230. ISBN: 9780471093909. DOI: 10.1002/0471221104.

[23] Jacek Dmochowski, Jacob Benesty, and Sofiène Affès. “On Spatial Aliasing in Microphone Arrays”. In: *IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 57, NO. 4* (2009).

A Appendix

The purpose of this appendix is to delve deeper into how the simulation methods describe the rooms utilized in this project. In other words, it provides a complementary description of the simulation methods and their results, expanding on section 2.1.

A.1 A comparison between ISM and Hybrid-based methods

To compare the simulation methods used in the project, it is essential to examine pairs of RIRs under identical conditions. The following paragraphs include visualizations in the time domain (waveforms), frequency domain (spectra), and time-frequency domain (spectrograms). Each set of visuals compares two RIRs: one generated using the ISM, and the other using the hybrid method. Both RIRs are obtained under the same room dimensions, conditions, absorption coefficients, and receiver positions, differing only in the simulation method employed.

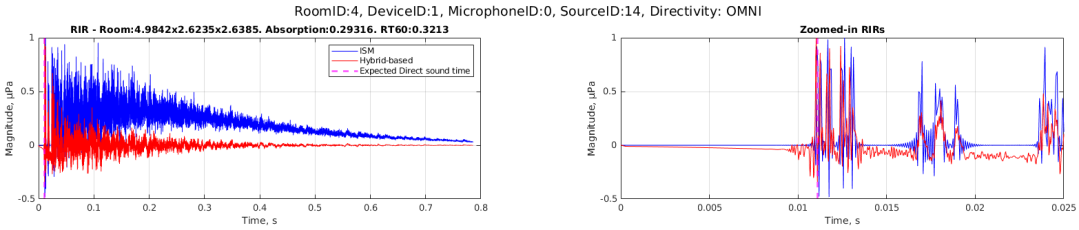


Figure A.1: RIR from omnidirectional source (ISM vs hybrid method, time domain)

Considering the RIRs in their time domain (see Figure A.1), several differences between the ISM and hybrid methods become apparent. Firstly, there are distinct decay times between the two methods, notably with the hybrid method exhibiting a longer decay. This difference can be attributed to the discrepancies between ISM and the RR method (used for approximating the late reverb tail in the hybrid method, see subsection 2.1.3). Additionally, a DC component is observable in the hybrid method (as seen in the zoomed-in RIR), which is absent in its ISM counterpart. Typically, the absence of a DC component suggests an ideal scenario, which is often distant from reality, while the presence of a DC component can effectively simulate the noise floor typical of real recordings. This aligns with the inherent differences in nature between the two simulation methods. However, despite these differences, both methods are designed to model rooms under identical conditions. This consistency is reflected in the timing alignment of the most significant pressure peaks in their impulse responses. This alignment is evident when comparing the timing of their first peaks with the Expected Direct Sound time, which, given the identical room conditions, marks the point in time when the first reflections of the responses are expected to occur.

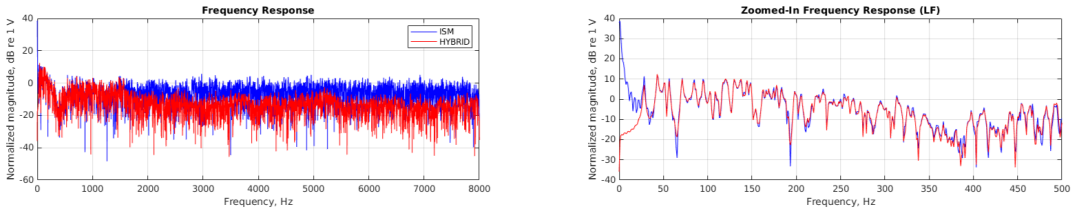


Figure A.2: RIR spectrum (ISM vs hybrid method, LF-zoom)

In the frequency domain analysis of the RIRs (see Figure A.2), several observations can be made. Notably, a decay is noticeable starting around the threshold frequency of $f = 1.6\ kHz$, particularly evident in the hybrid method. This decay suggests that the current implementation of the hybrid method may have issues related to magnitude matching.The absence of a similar decay in the ISM method across the spectrum indicates that the WB and GA components of the hybrid method may require adjustments in sensitivity or amplitude to achieve greater coherence between the outcomes of the two methods. Zooming into the lower part of the spectrum reveals a close similarity between the two methods (ISM vs DGFEM), as expected. This similarity, particularly noticeable below $500\ Hz$, aligns with the fact that both methods simulate the same virtual space, presumably characterized by identical room modes. In summary, while discrepancies are observed in the higher frequency range, the coherence in the lower frequencies reinforces the notion that both methods accurately simulate the same virtual acoustic environment.

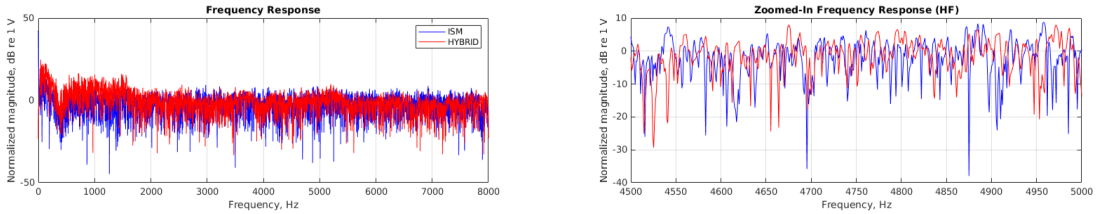


Figure A.3: RIR spectrum (ISM vs hybrid method, HF-zoom)

Additionally, it is valuable to investigate spectrum differences at higher frequencies (see Figure A.3). After compensating for the previously noted magnitude mismatch (by normalizing the entire hybrid spectrum using the average value of the ISM spectrum between frequencies $[4500\ 5000]\ Hz$), clear distinctions in spectral trends are apparent. This discrepancy suggests varying performance between the ISM and the GA components of the hybrid method, complicating a determination of which method might be more reliable. This higher frequency range typically involves overlapping of hundreds of modes, resulting in complex responses that pose challenges in drawing definitive conclusions.

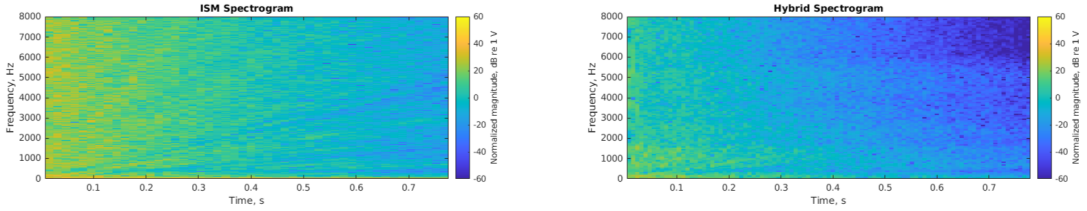


Figure A.4: Spectrograms for omnidirectional source (ISM vs hybrid method)

The spectrograms (see Figure A.4 visualize the previously discussed information across the time-frequency domain, corroborating the findings from Figures A.1 and A.2. Specifically, they illustrate the consistent time and frequency decay behaviors observed earlier, reaffirming the previous statements. One additional observation is the temporal resolution difference between the two methods. Notably, the hybrid method exhibits a higher sampling rate with double the frequency compared to its ISM counterpart.

To validate the previous hypotheses, it is valuable to compare RIRs that describe a new scenario under the same virtual conditions but with a different type of source (modeled loudspeaker). In this scenario, the modeled source is placed at a different position within the virtual space and is set to be directional instead of omnidirectional (see Figure A.5 and Figure A.7).

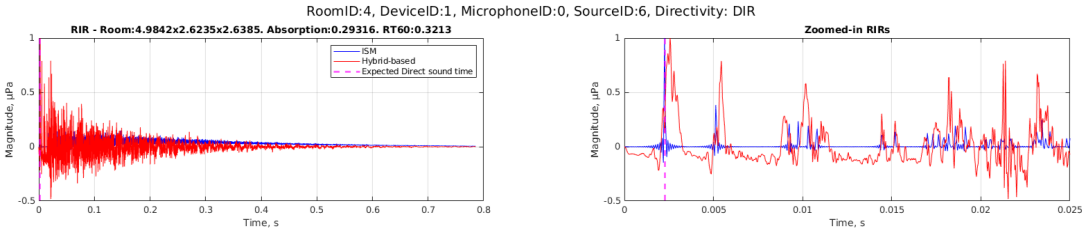


Figure A.5: RIR from directional source (ISM vs hybrid method, time domain)

Some of the previous observations regarding time characteristics find consistency with the new data (see Figure A.1). Specifically, a longer decay time is noticeable in the hybrid method. However, there is a discrepancy observed in the timing alignment of RIR peaks corresponding to the first reflections compared to the Expected Direct Sound time. This discrepancy may be attributed to the directivity properties inherent in the hybrid method (see subsection 2.1.3). While both methods assume a diffuse sound field in the higher frequencies, the ISM assumes all sources to be omnidirectional and does not account for directivity effects. On the other hand, the delayed peak of the first reflection in the hybrid method compared to the Expected Direct Sound time suggests potential issues in its directivity computation.

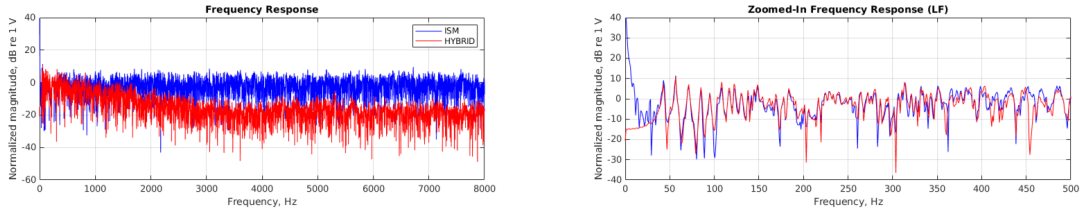


Figure A.6: RIR spectrum (ISM vs hybrid method, LF-zoom)

By examining the spectra of the RIRs and zooming into frequencies between $[0\ 500]\ Hz$ (see Figure A.6), the previous observations regarding magnitude mismatch within the hybrid method are consistent. Moreover above $400\ Hz$, the spectra of the two methods show a lack of overlap. This discrepancy can be attributed to the differences in directionality discussed earlier.

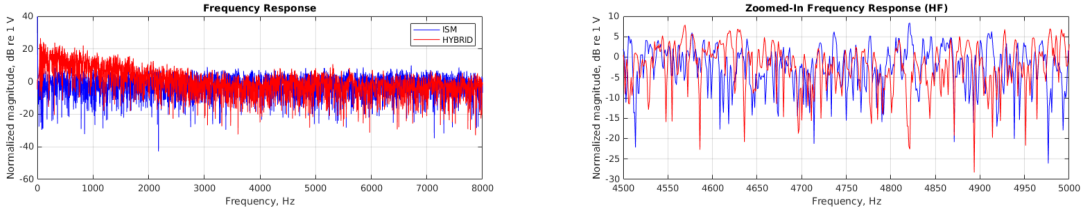


Figure A.7: RIR spectrum (ISM vs hybrid method, HF-zoom)

Analogously, after compensating for the magnitude mismatch in the spectrum and zooming into a higher portion of the spectrum ($[4500\ 5000]\ Hz$, same as before), the observations from the previous case are consistent (see Figure A.7). In this higher frequency range, the spectra of the two methods do not clearly overlap, highlighting the inherent differences between them. The lack of directionality in the ISM source may contribute to this mismatch. The ISM method assumes omnidirectional sources, which may result in a different spectral distribution compared to the hybrid method, where sources are modeled with directional properties.

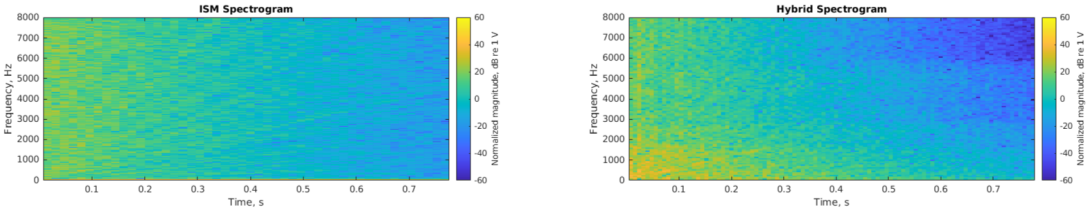


Figure A.8: Spectrograms for directional source (ISM vs hybrid method)

The previous statements regarding the RIR characteristics find consistency with the latest visualization of this last scenario (see Figure A.8). The information conveyed by these spectrograms aligns with what was discussed for Figures A.5, A.6, and A.4.

Technical
University of
Denmark

Brovej, Building 118
2800 Kgs. Lyngby
Tlf. 4525 1700

www.byg.dtu.dk