UNIVERSITÀ DEGLI STUDI DI PADOVA

Master Degree Thesis in
Computer Engineering

A comparison of two auditory front-end models for horizontal localization of concurrent speakers in adverse acoustic scenarios

Author:

Andrea Almenari

Supervisor:

Prof. Giorgio Maria Di Nunzio

Co-Supervisor:

Dott. Roberto Barumerli

A.Y. 2018/2019

CONTENTS

SECTION 1

INTRODUCTION

Ears are complex and extraordinary instruments which help humans understand what is happening around them. By using two ears (binaural hearing), a person can localize multiple sound sources and focus his attention to a specific acoustic source. This is very important because, if someone cannot understand the location of the sound source, it becomes difficult to perceive and interact with the environment, especially in the presence of noise. Since the second half of the past century, several studies have been done to explore how mammal's auditory processing encodes acoustic information. The first auditory models appeared in literature in the previous century [16], [33], [40], [32], [52]; nowadays, new approaches extend previous findings [54], [63], [4], [12], [38]. An extensive research has been carried out through the years, but many details of the auditory processing remain unclear: auditory modelling tries to replicate ears' functionality succeeding only partially or with some caveats (i.e. high level analysis performed by the brain are not exploited at the time of writing [64]). In this document, two of these auditory models will be analyzed and extended. Such models were proposed in [13] and in [38].

The thesis is organized as follows: the next section will provide to the reader an introduction about auditory processing, than an overview of binaural listening for Direction of Arrival (DoA) estimation and, finally, the machine learning tools employed. In the second chapter, the software and auditory models used are described and explained, while in the third chapter experimental setups for paper's replicas and new experiments are provided. Finally, the last sections cover the results and their discussion.

1.1 Research Questions

This manuscript wants to deal with the following open questions:

- What are the actual performances of the evaluated models for binaural azimuth estimation? Can different model's assumptions lead to similar results?
- Can a model simulate a real user only relying on his acoustic description?
- Which metric can better describe the evaluation of an auditory model?
- Is it possible to improve the auditory models by using different classification methods?

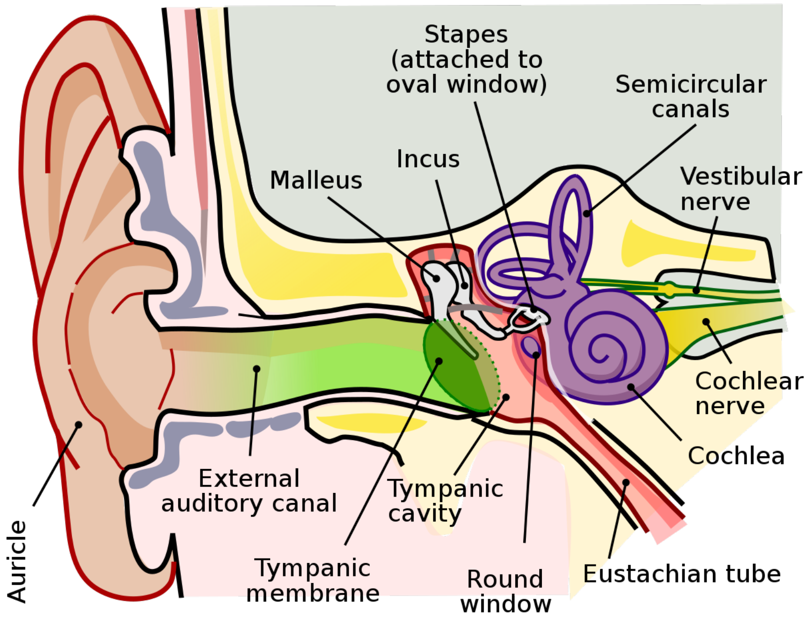

Figure 1: Ear anatomy - picture from Wikipedia[1]

1.2 Fundamentals of Auditory Processing

1.2.1 The human auditory system

Before starting to write about computational auditory models, we proceed to describe the components which compose the human auditory system. This is important because knowing the elaboration of the acoustic wave-field can allow more reliable and precise replication of the real human auditory processing.

The human auditory system includes both sensory organs and the auditory parts of the sensory system. A representation is available in Fig. 1. It is formed by three main parts:

- **Outer ear** is composed by auricles, cartilages which surround the ear canal. Pinna is made by folds of cartilage into the auricle, which reflects or attenuates sound waves as visible in Fig. 2. The ear canal amplifies sound frequencies between 3 and 12 kHz until the *tympanic membrane*. It can be noticed that outer ear can boost the sound pressure from 30 to 100 times near 3 kHz frequency [8].

- **Middle ear** starts with the *tympanic membrane*, where sound waves arrives and making it vibrate. The tympanic membrane transfer its movement to three specific bones, the *malleus*, the *incus* and *stapes*, which amplify the sound pressure at the *oval window*, approximately with a gain of at least 18:1 and with different lever arm factors for different frequencies (2 from 0.1 to 1 kHz, 5 at 2 kHz and decreasing fast

[1] https://commons.wikimedia.org/wiki/File:Anatomy_of_the_Human_Ear.svg, downloaded on May 10th, 2019

[2] http://www.cochlea.eu/en/ear/external-ear, downloaded on May 13th, 2019

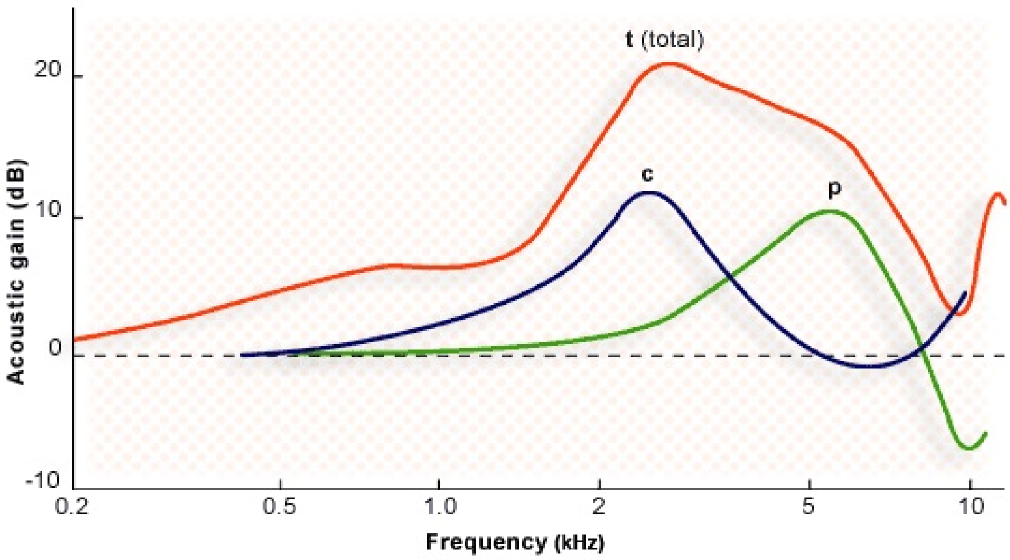

Figure 2: Outer ear frequency amplification. (c) refers to the ear canal while (p) refers to the pinna and (t) is the total amplification - picture from cochlea.eu[2]

above this last frequency) [29]. This increase of pressure in vibrations is necessary because, from the *oval window*, vibrations travel through a liquid instead of air. This allows impedance matching of sound travelling through a liquid in the inner ear instead of air. In some particular cases, sound pressure can also be damped through dedicated muscles (i.e. *stapedius muscle, tensor tympany muscle*) [29].

- **Inner ear**: in this part of the auditory system the acoustic vibrations are transduced in nerve impulses by the cochlea. Moreover the cochlea is also connected with the vestibular system. Cochlea is formed by three fluid-filled sections, *scala media, tympani and vestibuli* [67]. Vibrations, which arrive from the oval window, make the round window move, and the liquid, named *endolymph*, with it. The acoustic waves can arrive through air conduction (from the *tympani*) or bone conduction (from the skull) and they are threaten in the same way. The *basilar membrane* is the place where sound vibrations are captured in function of their frequency [55]. As a rule of thumb, the frequency is an exponential function of the length of the *cochlea* within the *organ of Corti*, as shown in Fig. 3. This last organ, positioned into the *scala media*, transforms sound vibrations into nerve signals using specific hair cells. These cells are separated into *inner* or *outer* and these are displaced as in Fig. 4. Inner cells actively convert vibrations into nerve impulses while outer cells amplify vibrations in a specific way and act as a motor structure. Inner cells' output, in contrast with neurons which show a spike response, exhibit a graduated response. In this point of the auditory system, mechanical signal is transformed into an electric signal which will be sent to the middle brain for high-level elaborations [24].

[3] https://www.britannica.com/science/basilar-membrane, downloaded in May 13th, 2019

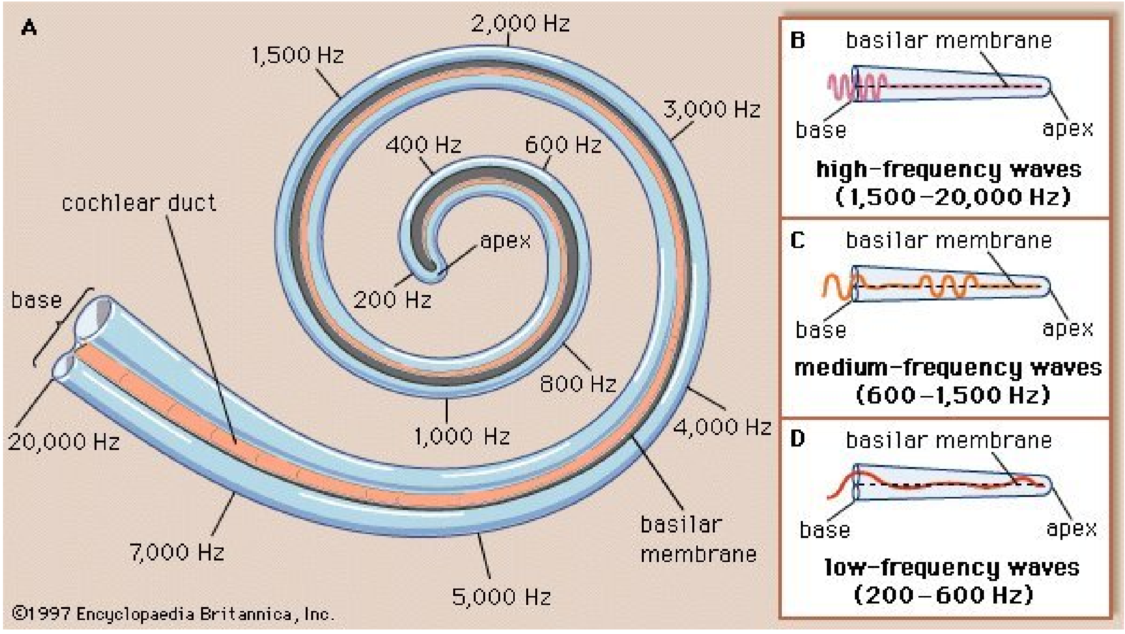

Figure 3: Relationship between basilar membrane and sound frequency - picture from Encyclopædia Britannica[3]

The **mid-brain organization** of the human auditory system is then formed of several parts:

- *trapezoid body*, which carries information used in binaural computations into the brain and helps sound localization [41];
- *superior olivary complex*, which detects interaural level and time differences (ITD and ILD) [43];
- *inferior colliculi*, which integrates localization information found by the *superior olivary complex* and *dorsal cochlear nucleus* before sending it to *thalamus* and *auditory cortex* [46] [47];
- *primary auditory cortex*, the first region of the external cortex which receives auditory inputs. Here there's the pitch, rhythm and speech perception. Neurons of this auditory cortex are selectively perceptive based on frequency [48] [66];
- *ventral and dorsal streams*, which are two different pathways for neural transmission of a sound. The ventral stream is responsible for sound recognition and meaning extraction from sentences, while the dorsal stream helps sound localization, articulation, phonological encoding and verbal working memory [23].

1.2.2 Binaural audio

Humans rely on two separate ears to analyze the acoustic environment. This ability let them to enhance hearing capabilities for instance: ease of listening and speech recognition, as reported in [15]. In addition, binaural hearing permits also to localize with a

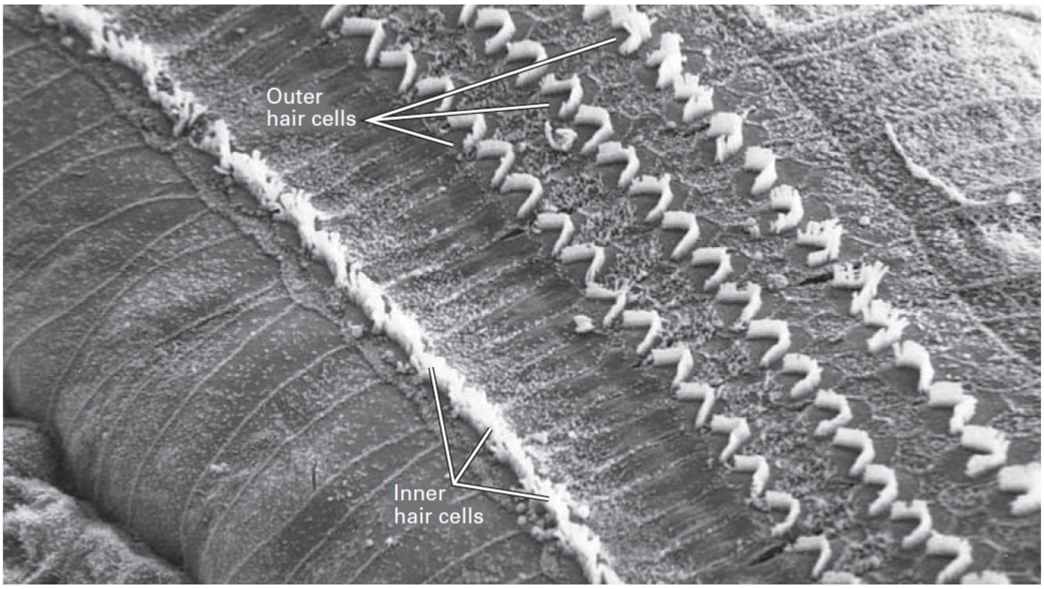

Figure 4: Inner cells and outer cells - picture from [27]

good accuracy the position of a sound source. The human brain has the extraordinary capacity to isolate a sound coming from a specific position, focusing only on what is the subject evaluates as interesting (cocktail party effect) [22]. An important challenge, which aggregates different research fields, is the computational auditory modeling which aims to develop a digital systems for enhancing speech intelligibility using spatial filters or localizing a sound from a specific position from a binaural source. These technologies can become very important for different type of applications: electronic hearing aids, cochlear implants and virtual or augmented reality. In this manuscript, the focus will be on sound localization; different models about this topic have been developed, both for horizontal and vertical planes. Horizontal-plane models as [38] or [13] are focused on the azimuth perceived by the listener, while the vertical-plane ones as [4] on the elevation.

Here the features for audio localization will be presented, together with methods to generate spatial audio used in experiments described in Section 3.

1.2.3 Computational Auditory Scene Analysis

Computational Auditory Scene Analysis (CASA) is the study of auditory scenarios by computational means, trying to replicate how the human listeners do when they listen. To achieve this purpose, auditory models which model outer, middle and inner ear are necessary. In particular, several types of filters are usually used in order to mimic the human auditory system behaviour, as lowpass, bandpass, highpass (especially for outer and middle ear) and Gammatone filters. This last type of filters have been used quite often in literature in filter-banks to model how basilar membrane responds to different frequencies because it has the potential to better resolve the harmonics of complex tones

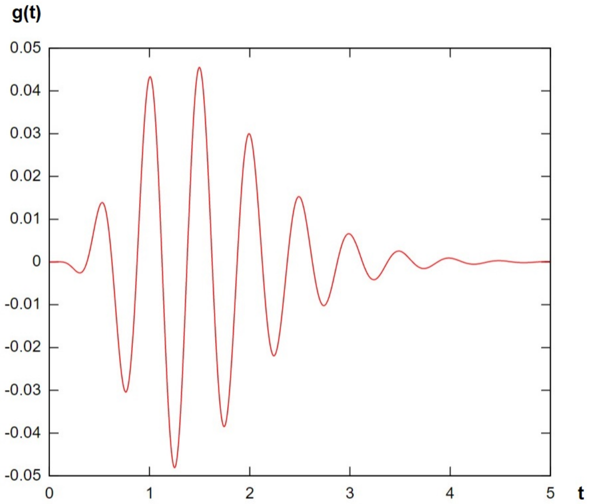

Figure 5: Example of Gammatone filter - picture from Wikipedia[4]

[61]. A Gammatone filter is essentially the product of a gamma function for a pure tone:

$$g(t) = at^{n-1}e^{-2\pi bt}\cos(2\pi ft + \phi) \tag{1}$$

where f is the center frequency, ϕ the phase of the carrier, a the amplitude, n the filter order, b the bandwidth and t the time. The typical shape of a Gammatone filter is in Fig. 5. Filter-bank structure is usually created by spacing each filter of a Equivalent Rectangular Bandwidth (ERB) measure unit, which gives an approximation of the bandwidth of human auditory filters. This permits to simplify that filters by modeling them as bandpass filters. ERBs have been defined by Moore and Glasberg using two different formulations, which are valid for moderate sounds and young listeners:

- polynomial approximation (dated 1983), valid from 0.1 to 6.5 kHz [42]:

$$ERB(f) = 6.23 \cdot f^2 + 93.39 \cdot f + 28.52 \tag{2}$$

- linear approximation (dated 1990), valid from 0.1 to 10 kHz [20]:

$$ERB(f) = 24.7 \cdot (4.37 \cdot f + 1) \tag{3}$$

In the above formulas, f is the center frequency of the filter in kHz and $ERB(f)$ the bandwidth of the filter in Hz.

The firing rate in the auditory nerve is quite-often used rather than spikes, modeling hair cells' behaviour with an half-wave rectification followed by a square-root compression.

[4] https://upload.wikimedia.org/wikipedia/commons/9/94/Sample_gammatone.svg, downloaded in May 13th, 2019

For pitch perception, instead, two different theories have been developed, *place theory* and *temporal theory*, emphasizing respectively resolved or unresolved harmonics. Using these two theories in the time domain and by auto-correlating the simulated auditory nerve activity to the output of each frequency channel, a *correlogram* can be computed. Using this information, and then, pooling the auto-correlation across frequency, pitch can be extracted from the *correlogram* as dominant peaks [65]. To determine the sound source, using the fact that ears receives sounds at different times, delays between ears can be exploited by using the cross-correlation of the signals at the left and right ears, or with alternative techniques [65]. The question about the use of cross-correlation for sound sources' extraction is still open and details can be found in next paragraphs.

1.2.4 Binaural features

The main important features used for audio Direction of Arrival (DOA) localization are:

- *Interaural Phase Difference* (IPD): the phase difference of a pure-tone signal when arriving before in a ear and after in the other.

- *Interaural Time Difference* (ITD): the time difference of an broadband audio signal to be heard from the other ear when arrived to the first ear.

- *Interaural Level Difference* (ILD): the acoustic source's difference of sound pressure between the outer ears.

These features can be adopted with the desired referral system (for example, ITD positive if the signal arrives before in the left ear and negative on the other side, or vice-versa). For frequencies below 1400 Hz, head's dimensions are smaller than the half of the wavelength of the sound waves and the auditory system can determine easily phase delays between ears. Level differences, instead, are difficult to estimate because of their low values. The situation becomes more critical below 200 Hz, because a precise localization is nearly impossible using level differences. Phase differences also become very low below 80 Hz. High frequencies also impose constraints on sound localization: if they're above 1.6 kHz head's dimensions are greater than the wavelength of sounds. Using only IPDs for DOA estimation is ambiguous, and here the use of ILDs can help to remove the ambiguity because they become strong enough to be used [55].

According to [11] and as written before, ITD and ILD are elaborated by the *superior olivary complex* in humans. Even if quite accurate artificial neural circuits have been developed above these features, how the human brain decodes these features it is still not clear. Such knowledge can improve the limits of cochlear implants for binaural hearing and threat with more precision hearing disorders. The first model to process the ITD feature was proposed in the Jeffress's work [25]: calculation relies on *delay lines*, where neurons on the *superior olivary complex* accept innervation from ears with axons (impulses' conductors of neurons) of different length. Delay elements compensate the ITD in order to detect the coincidence. This behaviour was considered in line with physiology outcomes because of the observation of these axonal delays and of an array of coincidence detectors in the barn owl and in other mammals spieces such as cats, and owls [68] [53]. This model can be considered as a physiological representation of the cross-correlation of the left and right signal perceived by ears. A lot of models today use cross-correlation to estimate the Direction of Arrival of a sound source (DOA) with very good performance (see [38], its derivatives [36], [35] and [31]), but the Jeffress's model cannot explain the precedence effect of sound event separation when two sounds are emitted with a sufficient delay time [11]. In addition, studies on guinea pigs do not support the existence of the previous cited physiological structure [39]. Finally, other computational models have been developed, but their results are not as good as the ones relying on cross-correlation. It is unclear if these worse performances are related to a fundamentally incorrect formulation or to lack of development as in cross-correlation-based models over several decades [11]. Moreover, actually auditory models operates for very-specific problems, as horizontal or vertical localization, leading to the absence of a general model.

1.2.5 Open questions on human brain's binaural processing

The academic and industrial research on human auditory system has more than fifty years of development, but some open issues still remains on how human process the acoustic information.

One of the most singular facts is that neurons responds maximally with ITD different from zero [26]. The distribution of ITDs at which the response of neurons is maximum is referred, according to the work of Stern and Colburn [58], as the $p(\tau)$ function, where τ is the delay perceived. The debate occurs when investigating if the maximum of that distribution is in the zero delay or in a delay approximately equal to an IPD of 45°[39]. This maximum delay is also called *best delay*. The distribution of these *best delays* and the cause of the non-zero delay have not been explained yet.

Another open question is how the DOA estimation is performed by the brain. Four approaches are commonly proposed in literature, but no one claims that there is a perfect match with physiological correlates. An overview is available in [11]. To understand better the meaning of the following formulas, some variables have to be defined:

- n: the n-th neuron, where $1 \leq n \leq N$ and N is the total number of neurons;
- R_n: the response of the neuron n;
- α_0: the direction of arrival (degrees);

- BD_n: the *best delay* for the neuron n (milliseconds);
- dir: the final direction of arrival estimate (degrees).

The proposed approaches are:

- *Place code*: it is used by the majority of the delay-line based models. The ITD is compensated by the neuron's best delay, but in this way neurons' weak responses are ignored [25].
$$dir = \arg\max_n (R_n(a_0)) \tag{4}$$
- *Hemispheric rate difference*: this technique involves all neurons and compares the neural activity of the two hemispheres or compares the sum of the activity of neurons with positive best delay with ones with negative, even for neurons with atypical behaviour [62].
$$dir \propto \sum_n R_{n,Right}(\alpha_0) - \sum_n R_{n,Left}(\alpha_0) \tag{5}$$
- *Centroid*: the total activity of coincidence-counting neurons is considered and the centroid along the best-delay axis is computed. This technique can give similar results of *place code* method if signal have reinforcing ITDs of small magnitude [58].
$$dir \propto \frac{\sum_n BD_n \times R_n(\alpha_0)}{\sum_n \alpha_0} \tag{6}$$
- *Pattern Matching and Maximum Likelihood Estimation*: detection is done using a model which learns the response of all neurons in function of their direction. For each stimulus, the direction will be the one which agrees better with the learned response [9] [21].

$$dir = \arg\max_T \frac{\sum_n R_n(\alpha_0) R_n(\alpha_T)}{\sqrt{\sum_n R_n(\alpha_0)^2}\sqrt{\sum_n R_n(\alpha_T)^2}} \quad \text{for} \quad \text{PM} \tag{7}$$

$$dir = \arg\max_\alpha (P(R_n(\alpha_0)|\alpha)) \quad \text{for} \quad \text{MLE} \tag{8}$$

In the last years, cortical electrophysiologic data were also taken into account, but discovering of real mechanisms of human's DOA estimation remain one of the most important challenges in binaural audio research.

In addition to the four measurements presented into the previous section, there is another unclear feature, the Interaural Group Delay (IGD), which is the derivative of ITD in the frequency. IGD becomes important for complex stimuli, since in pure-tone sounds phase difference is constant. In order to exploit better ITD variation, some

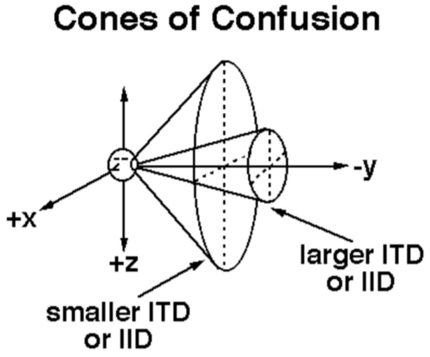

Figure 6: The confusion cone concept. Picture from [5]

.

models decompose the signal into different frequency bands and extract ITDs or cross-correlations from each one of them, but IGD calculation has not been used yet directly in any model.

1.2.6 The front-back confusion and frequency-related problems

Currently, the presented binaural features do not completely account an important issue: the *front-back confusion*. Front-back confusion is a phenomenon where, in the horizontal planes, a DOA can be misunderstood and perceived in the opposite hemifield. This depends on small differences of binaural features for directions affected by this issue. The problem, indeed, can be threaten also considering elevation in addition to azimuth, bringing the ambiguity not only to two ambiguous positions, but to infinite positions mapped by a *cone of confusion*. This ambiguity derives from the inability of common-used binaural features (ITD, ILD and IPD), with today's techniques, to be distinguished in these ambiguous situations. So, other techniques need to be exploited to remove ambiguity, as for example, head rotation [36].

1.2.7 Head Related Transfer Functions

As described in the section 1.2.1, the pinna can be modelled as an spatial-dependent set of acoustic filters. They account the pinna's shape, which lets to generate a particular frequency spectrum derived from direct and reflected sounds thanks to its asymmetries and complex folds. The obtained spectrum is associated to specific DOAs [6]. This pinna-filtering can be modeled with the so-called *Head-Related Transfer Functions* (HRTF), whose temporal representations are the *Head-Related Impulse Responses* (HRIR). HRTFs are usually recognized as Linear Time-Invariant (LTI) systems and can

[5] https://humansystems.arc.nasa.gov/groups/ACD/images/The.Role.of.D.Fig2.gif, downloaded in May 13th, 2019

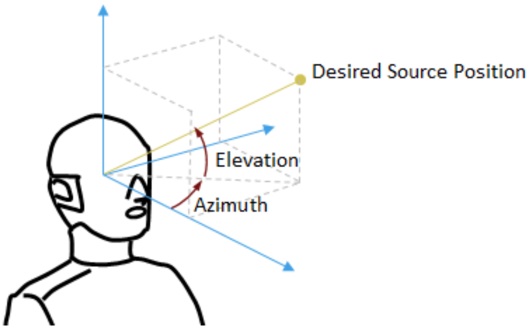

Figure 7: Measures involved in HRTFs - Picture from [6]

.

be written as:

$$H_d = H_d(r, \Theta, \phi, \omega, \alpha) = P_d(r, \Theta, \phi, \omega, \alpha)/P_0(r, \omega) \tag{9}$$

where d is the direction (left or right), P_d is the amplitude of the sound pressure in the direction d, P_0 is the amplitude of sound pressure at the center of the head, while r is the distance center-of-head - source, Θ is the azimuth, ϕ is the elevation, ω the angular velocity and α the equivalent dimension of the head.

HRTF acquisition is not a simple process, which requires measuring them on a spatial grid with discrete frequency samples in anechoic chambers. The complexity of the whole thing depends on the fact that each acquisition is a very delicate process, where many accidental variables can change the final result. Several studies on HRTF measurement repeatability have shown different sources of variability on the acquired quantities [2].

1.2.8 Binaural audio synthesis using HRIRs

Using HRIRs, a monophonic sound can be transformed and spatialized by headphones listening. The procedure involves the convolution of the monophonic signal with left and right HRIRs for specific azimuth, elevation and distance. Since a HRIR database contains a finite number of HRIRs, an interpolation technique is required. The convolution with left and right HRIRs permits to obtain a stereophonic signal which contains the spatial information. The listener have to listen with headphones in order to simulate the ears' filtering to perceive spatial audio.

A simple schema of the procedure is in the Fig. 8.

HRTF datasets HRTF datasets are composed by HRTF recorded from real subjects or dummy head (i.e. KEMAR mannequin). Some of the most famous datasets available in literature are the following ones:

[6] https://it.mathworks.com/help/audio/ref/interpolatehrtf.html, downloaded in May 13th, 2019

[7] https://it.wikipedia.org/wiki/File:Hrir_binaural_synthesis.png, downloaded in May 14th, 2019

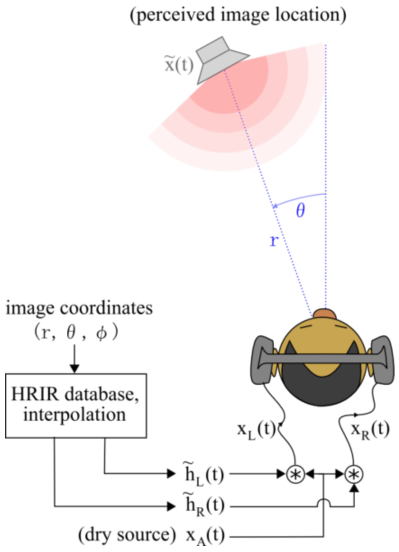

Figure 8: Binaural synthesis schema - picture from Wikipedia[7]

Figure 9: A KEMAR mannequin - picture from [8]

- *MIT*: recorded using a KEMAR mannequin, available in full and compact versions, often used in many publications;
- *CIPIC*: recorded using 45 real subjects and KEMAR mannequin;
- *ARI*: recorded using over 170 listeners and two KEMAR mannequins;
- *TU-Berlin*: recorded using a KEMAR and a FABIAN dummy heads.

A KEMAR (Knowles Electronics Manikin for Acoustic Research, in Fig. 9) is the first head and torso simulator designed for acoustic research. It has been designed with median human adult dimensions and the ear mounted on it can be replaced with different types of pinnaes and shapes. Its main goal is to make audiology measurements as repeatable as possible, simulating how a real human listens.

[8] https://www.gras.dk/products/head-torso-simulators-kemar/kemar-for-hearing-aid-test-1-ch/product/499-45bb-1, downloaded in May 14th, 2019

HRTF limitations Since pinna's shapes are different from person to person, the consequent spatial filtering varies. This problem introduces the need to personalize the spatialization techniques when the acoustic environment is rendered on headphones. Some ways to approach this problem would be HRTF personalization or best-HRTF selection.

- *HRTF personalization* consists on recording the HRIRs and to use them (also with interpolation) to generate spatial audio.

- *Best-HRTF matching*: by using a set of pre-recorded HRIR dataset it is possible to build some perceptual metrics to select the user nearest HRIR database.

The main disadvantage of these methods is the difficulty of training, which requires the listener to spend a significant quantity of time in order to obtain a binaural audio system that fits its needs. Other approaches were also tried, for example HRTF's synthesis based on ears' shape, but these techniques require a complex mathematical tools although leading to a limited spatialization accuracy.

1.2.9 Room Acoustic Simulation

A room can be seen as a Linear-Time-Invariant (LTI) system if source and receiver are fixed in the room. For this reason, it is possible to define the room with an impulse response. The impulse response of a room is usually shortened with the acronym RIR. A RIR defines exactly a source-receiver combination inside a room; as an example, if a RIR has been recorded from a seat in a theatre, and if that RIR is convoluted with a monophonic sound and reproduced on headphones, that sound will be perceived as if the user is listening to it from that seat.

There are two ways to obtain a RIR of a room:

- *Direct recording*: position a dummy head/mannequin in the desired receiver position, a source in its position and then the RIR is recorded. This approach permits to obtain a very precise impulse response, but it is poor in terms of flexibility because a RIR needs to be recorded for each source - receiver positions;

- *Room simulation*: use a software to simulate a room and different source-receiver configurations. Room simulation leads to worse results with respect to direct RIR recording but it allows more flexibility in positioning sources and receivers.

The complexity of the problem depends on room's size and structure, wall materials and physical conditions, as for instance temperature and humidity. The common simulation approach is ray-tracing technique and many commercial solutions relies on this [9] [10] [11]

[9] Odeon, available at https://odeon.dk/

[10] CATT Acoustic, available at http://www.catt.se/

[11] EASE, available at https://ease.afmg.eu/

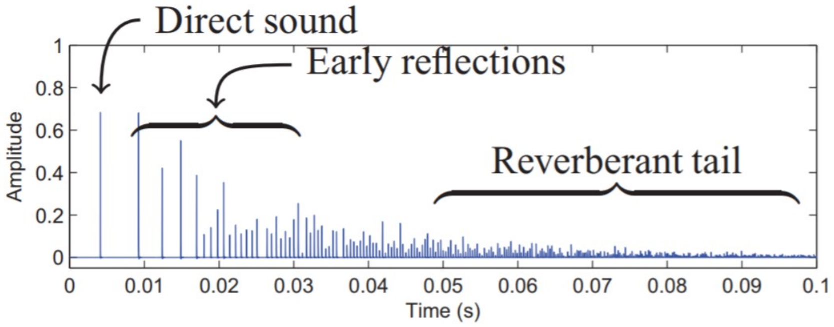

Figure 10: An overview of a channel of a RIR. Picture from [34]

.

[57]. Ray-tracing uses the hypothesis that sound wave propagates with specular reflections on the walls inside the room. The more are the rays, the more precise will be the synthetic RIR. Sound rays are emitted by a sound source in several directions according to the source emitting pattern and the reflections are captured by a receiver, both localized in a specific point of the room. The sum of rays captured by the receiver contains reverberation pattern which are related to the geometry and the acoustic parameters of the simulated room and can be divided into several parts:

- *Direct sound*: it is the first impulse in the RIR waveform and is generated by direct rays which hit the receiver.
- *Early reflections*: they can be seen in the waveform as the first part of the reverberant tail. They have quite large peaks and generated by first reflections of sound rays.
- *Late reverations*: they form the last part of the reverberant tail and they are generated by sound rays which are reflected several times.

Sound rays' intensity in time tends to fade out because of absorption and diffusion phenomena: the walls absorb some quantity of sound energy and reflect the remaining sound. After a number of iterations each rays will be completely lose its energy.

An example of the method is illustrated in the Fig. 11.

An implementation used to synthesize the RIR is the *image-source method* [34]. In this case (Fig. 12), the sound source is mirrored in the room's walls to create virtual sources corresponding to the real one and the RIR can be computed considering all the straight lines starting from all sources (real and virtual) and arriving to the receiver. This method tends to be usually more efficient than a general ray-tracing method but it remains inaccurate for high-order reflections computation. To overcame this limitation the diffuse rain algorithm can be used [34].

To generate a sound from a position not recorded by RIRs, instead, is necessary to elaborate impulse responses in a such way that accounts positions not recorded by the

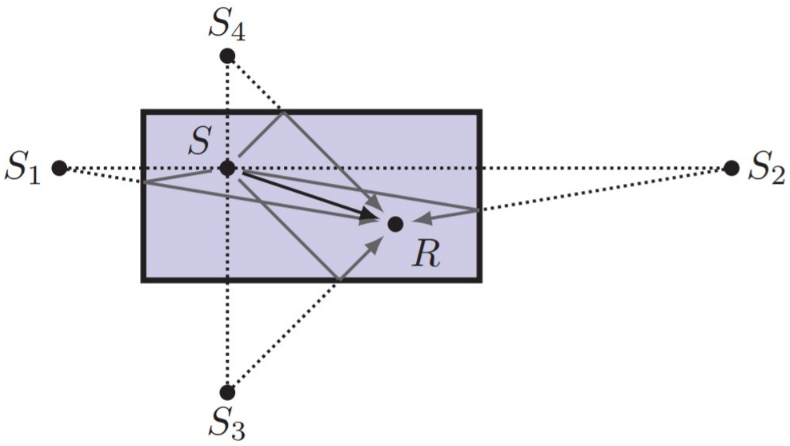

Figure 11: Ray-tracking for room simulation. Picture from [34]

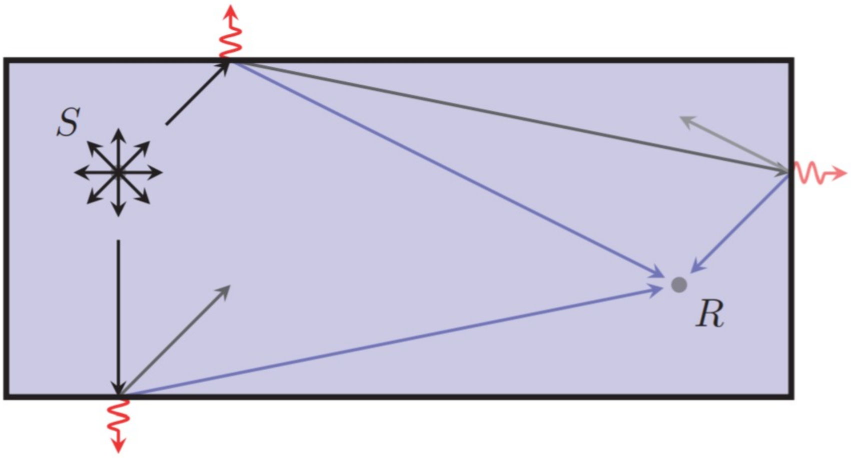

Figure 12: Image-sources for room simulation. Picture from [34]

.

responses' dataset. Two important techniques are used to overcome this issue: one is interpolation, which let to estimate a RIR which is between two different responses (using for example bilinear interpolation); the second one is related to the distance. This last problem can be solved accounting the attenuation due to the ray's path length. Distance attenuation works on the principle that the sound energy decreases proportional to the square of the distance.

1.2.10 The Sabine formula

This formula, used to estimate reverberation times in a simple room, is based on three important assumptions:

- *persistence*: two different sounds with a time distance of less than 0.1 s cannot be distinguished;

- *sound speed*: it is of 340 m/s in a room with temperature of 20° .
- *position*: the sound source and the listener must be in the same axis with respect to the obstacle.

The main Sabine formula formulation is the following:

$$T_{60,f} = 0.16 \cdot \frac{V}{\sum_i \alpha_{i,f} A_i} \tag{10}$$

where V is the volume of the room, $\alpha_{i,f}$ is the absorption coefficient of the room's wall i at a certain frequency f and A_i is the area of the wall i.

With the assumptions written above, the formula can be inverted and $\alpha_{i,f}$ for every frequency f can be found.

1.3 Computational tools

In this section the computational tools used in this work are illustrated.

1.3.1 Machine Learning: a brief introduction

Machine Learning (ML) is a branch of artificial intelligence (AI) which is focused on making a computer capable of learning information and predict results. Research in this field started from the '50s, when some researchers (Minsky, Samuel and Rosenblatt) tried to understand if computers can learn from data [49]. The first time the expression "neural network" has been used was in the late 50s [1]. The architecture of the network at that time was very different from the one which is intended now: it was formed by a single perceptron, a binary classifier which used linear models to predict a boolean value [45]. Some first probabilistic models were also created, but they were plagued by problems [45]. Research continued slowly in particular in the Information Retrieval (IR) and the pattern matching fields because of domination of *expert systems* while, in the '80s, back propagation applied to neural networks has been discovered ([45]) and, from the '90s, researchers started to solve practical problems instead of "obtaining" the artificial intelligence [30]. ML also started to separate from symbolic approaches given by AI and began to be coupled with statistical models and methods [30].

Since ML procedures are data-centric, it is essential to have enough data related to reveal the mathematical model behind the studied problem. Data in ML can be used for training, validation and testing purposes [5].

ML is implemented with different assumptions, depending on the type of the problem:

- *Supervised Learning*, where each given training input has also the desired output (usually called label)

- *Unsupervised Learning*, where only training inputs are given without labels and the model has to find a structure inside data
- *Reinforcement Learning*, where the model tries to reach a goal with a "teacher" which tells the system if it succeeded or not.

For each one of the methods above, there can be three different categories of tasks:

- *Classification*: predicted labels are finite and cannot be different from the ones used for the training set. It will be the type of problem which will be analyzed in this document.
- *Regression*: predicted labels can be different from the ones used in the training set. Usually they're continuous real values.
- *Clustering*: labels do not come from the training set, data is divided into groups by an unsupervised learning technique.

ML can be done using different types techniques, each one with a strong mathematical background behind, for example: Neural Networks (NN), Linear prediction, Logistic Regression, Support Vector Machines (SVM), Gaussian Mixture Models (GMM), Hidden Markov Models (HMM), Bayesian Networks.

In this manuscript GMMs, NNs and HMMs will be used and explained.

1.3.2 Datasets in ML

As mentioned above, ML requires data in order to train a specific model. Data can be grouped to three different sets:

- *Training set*: used to train the model
- *Validation set*: used to verify if the trained model is good enough and optimize hyper-parameters if necessary (usually with model-selection)
- *Test set*: used to test performances of the chosen model.

In supervised learning, every set is accompanied by a specific error metric, which denotes in classification the probability of a prediction mismatch with respect to initially assigned samples' values. In addition, training error can not be representative of the real performance of the model. As an example (Fig. 13), consider a 10 points training set brought from points of a curve. Since points are 10, it can be seen that the polynomial which fit better them is a 10-degree polynomial, with a null training error. But, if other points of the curve outside the training set are chosen, predictions will be not so good. For this

[12] https://medium.com/@lotass/machine-learning-what-you-need-to-know-about-model-selection-and-evaluation-8b641fd37fd5, downloaded in May 16th, 2019

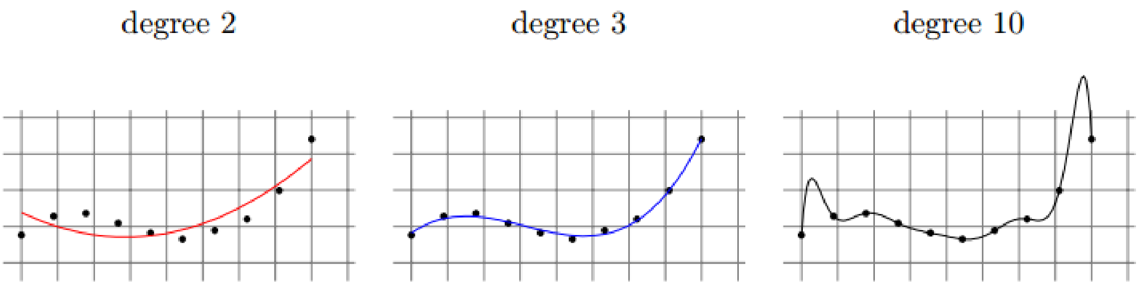

Figure 13: The polynomial example - picture from [12]

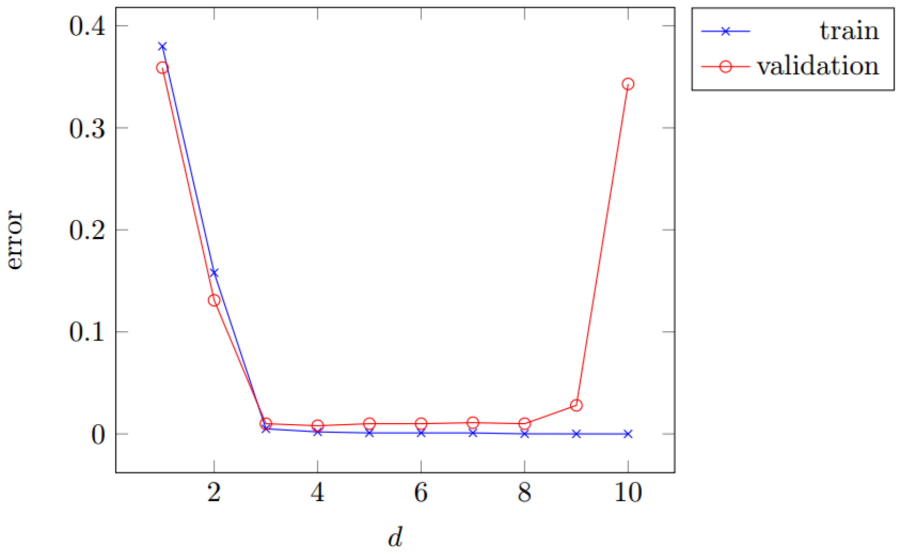

Figure 14: Model selection curve for the polynomial example - picture from [12]

reason, *model selection* and *validation* have been introduced: to train using different models with the same training set and to verify their performances with another set of samples, i.e. the *validation set*. Looking for the trend of training and validation errors (*model-selection curve*) in Fig. 14, it can be seen the model which does fit better the initial curve. It can also be seen that there are models which are too simple to fit the problem (causing *underfitting*) and others which captures a lot of information from the training set, but do not generalize well (causing *overfitting*).

To test final performances of the selected model the Test set is used, since it has not been touched by the previous operations.

In the following sections the ML techniques used in this work are explained in detail.

1.3.3 Used Machine Learning techniques

Gaussian Mixture Model (GMM) Suppose data have been generated according to a mix of different probabilistic distributions (mixtures), but after generation data can be only identified ad produced by a unique multi-modal distribution. The main purpose is to try to identify the single distributions' parameters inside the multi-modal distribution, acting as a sort of clustering algorithm. It is assumed for simplicity that, at the beginning, every distribution has the same mathematical description. A common case of this scenario is the one where these distributions are multi-normal (Gaussian) and it is needed to estimate their parameters (mean (μ) vectors, covariance (Σ) matrices and α coefficients) [50].

In this situation, the probability of the event x knowing Θ, which defines the multi-normal distribution, would be the following:

$$p(x|\Theta) = \sum_{i=1\ldots s} \alpha_i \cdot p_i(x|\Theta_i) \quad \text{where} \quad \Theta_i = \{\mu_i, \Sigma_i\} \tag{11}$$

Parameters estimation is done through the general criterion of maximum likelihood; the likelihood is defined as

$$L(\Theta|X) = p(X|\Theta) = \prod_{j=1\ldots n} \left(\sum_{i=1\ldots s} \alpha_i \cdot p_i(x_j|\Theta_i) \right) \tag{12}$$

where Θ defines the parameters and X the given patterns. For simplicity the logarithm of the likelihood is maximized instead of the direct likelihood measure; indeed, the previous formula becomes

$$\log L(\Theta|X) = \log \prod_{j=1\ldots n} \left(\sum_{i=1\ldots s} \alpha_i \cdot p_i(x_j|\Theta_i) \right) = \sum_{j=1\ldots n} \log \left(\sum_{i=1\ldots s} \alpha_i \cdot p_i(x_j|\Theta_i) \right) \tag{13}$$

Now, the main difficulty is the sum inside the logarithm, which makes the maximization procedure heavy to calculate. In order to remove that sum, it is necessary to know what mixture component $p_i(\cdot)$ generates every pattern x_j. Here comes in aid the Expectation-Maximization (EM), an iterative procedure which let the calculation of the maximum likelihood when some data in X are missing (Y).

EM is designed to maximize the log-likelihood of the full data (*complete log-likelihood*):

$$\log L(\Theta|Z) = \log L(\Theta|X, Y) = p(X, Y|\Theta) \tag{14}$$

This can be done using two procedures until convergence:

- *Expectation*: the expectation of the *complete log-likelihood* is computed, given the training set X and the parameters Θ^g calculated in the previous iteration. Following that, the expectation (average) is computed with respect to the random variable Y, governed by the distribution $f(y|X, \Theta^g)$.

$$Q(\Theta|\Theta^g) = E\left(\log p(X, Y|\Theta)|X, \Theta^g\right) = \int_{y \in \Psi} \log p(X, Y|\Theta) \cdot f(y|X, \Theta^g) dy \tag{15}$$

- *Maximization*: the maximum value of Θ is then computed from the previous

Expectation step.

$$\Theta^{g+1} = \arg\max_{\Theta} Q(\Theta|\Theta^g) \tag{16}$$

In this case, however, X is complete, but EM is used to make the internal sum's calculation easier. Y will indicate the unknown components which have generated every single pattern. Each pattern has a hidden component y_j which indicates what Gaussian distribution has generated the pattern x_j. Now, the *complete log-likelihood* can be written in the following form:

$$\log L(\Theta|X,Y) = \sum_{j=1\ldots n} \log\left(\alpha_{y_j} \cdot p_{y_j}(x_j|\Theta_{y_j})\right) \tag{17}$$

An estimation of various y_j can be derived by Θ_g available at the current iteration g. Since, for a generic observation vector y:

$$P(y|X,\Theta^g) = \prod_{j=1\ldots n} P(y_j|x_j,\Theta^g) \tag{18}$$

the expected value Q can be rewritten as:

$$Q(\Theta|\Theta^g) = \sum_{y\in\Psi} \log L(\Theta|X,Y) \cdot P(y|X,\Theta^g) =$$

$$\sum_{y\in\Psi}\left(\sum_{j=1\ldots n} \log\left(\alpha_{y_j} \cdot p_{y_j}(x_j|\Theta_{y_j})\right) \cdot \prod_{j=1\ldots n} P(y_j|x_j,\Theta^g)\right) \tag{19}$$

An important feature about GMMs is that can capture clusters of ellissoidal form instead of only spheric form. Medium values μ are often called centroids.

An example of clustering with GMM can be seen in the Fig. 15

Neural Networks [10] Another method used to predict values from given data consists on Neural Networks (NNs). They can be considered as a simplified model of human brain, where the single units are called *neurons*. Each neuron receives data from input or other neurons and elaborates a response through an *activation function*. Formally, if each neuron input is x_i, each input has a weight w_i, and the activation function is σ, the output response of the neuron will be:

$$out = \sigma(\sum_i (w_i \cdot x_i)) \tag{20}$$

where $\sigma(\cdot)$ can be for example a sign, $\mathrm{sigmoid}$ or $\tanh$ function. In this section feedforward NNs will be discussed, but other types of NNs exist, such as Recurrent NNs and Convolutional NNs, used a lot for image processing problems. A feedforward NN (Fig.

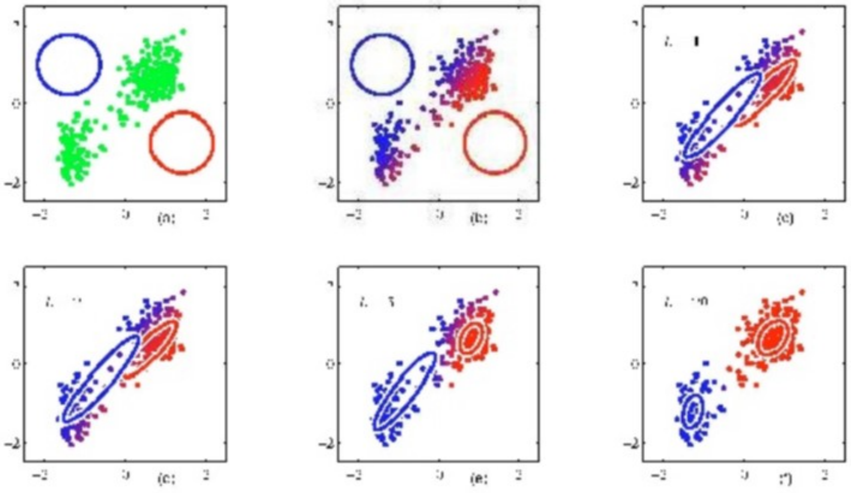

Figure 15: An example of GMM fitting - picture from [5]

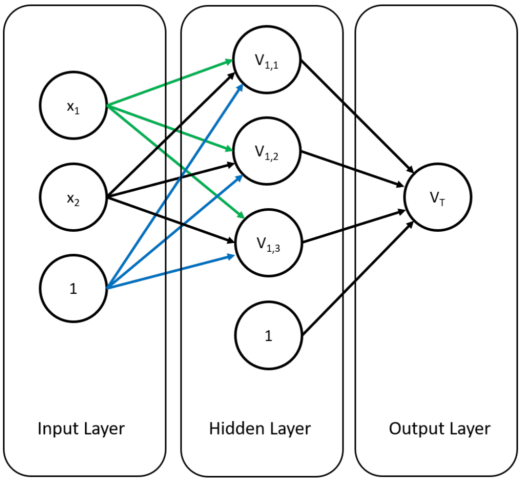

Figure 16: An example of feed-forward NN

16) can be seen as an oriented graph $G =< V, E >$ where V is the set of vertices of the neurons and E the set of edges, each one with a weight $w(e)$, where $w : E \rightarrow \Re$. Vertices are arranged into layers, where the first layer is the input layer, the latter is the output layer and remaining are called hidden layers. It can be seen the presence of the bias for every layer through the vertex indicated as "1", which will be multiplied for the bias weight.

Each NN has some properties associated:

- A node can belong only to a layer
- An edge cannot go backwards, but only from the layer t to the layer $t + 1$
- *Depth*: the number of layers less the input layer
- *Size*: the number of nodes
- *Width*: the maximum number of nodes in a layer

For example, the network of the picture above has Depth=2, Size=8 and Width=4.

Let's see more in detail the point of view of a single node of the network and consider the node $v_{t+1,j}$, where $t+1$ is the number of the layer and j the number of the node inside the layer $t+1$. The input of the node $a_{t+1,j}(x)$ when x is fed to the NN can be written as

$$a_{t+1,j}(x) = \sum_{r:(v_{t_r}, v_{t+1,j}) \in E} w((v_{t,r}, v_{t+1,j}))o_{t,r}(x) \tag{21}$$

and the output of the node $o_{t+1,j}(x)$ at this point will be

$$o_{t+1,j}(x) = \sigma(a_{t+1,j}(x)) \tag{22}$$

Neural networks have two important features which make them very suitable for ML: they can implement every binary operation and can be used as universal approximators. Despite this, training time using a naive strategy of minimization of the training error will be exponential. So, an alternative approach needs to be found for training and it was discovered that heuristic with Stochastic Gradient Descent gives good results in practice [5]. Gradient Descent (GD) is a technique used to find the minimum of a certain cost function using the fact that the direction where the cost function decreases faster in a point x is the one opposed to the gradient of x. This method starts from a initial solution x_0 chosen randomly and updates it using the formula:

$$x_{k+1} = x_k + \alpha_k \cdot p_k \tag{23}$$

where p_k is the opposite of the gradient of x and α is the learning factor, i.e. the size of the step of descent. If α is too large, the method will jump too much from a value to another making difficult to reach the minimum, while a α too small makes the algorithm converge very slowly. The size of α can be optimized using functions which makes it a function of the previous updating steps. GD is a reliable procedure, but it tends to be slow when data size is huge: it needs to check each training sample for every iteration. In order to solve this problem, another variant of GD has been made, the Stochastic Gradient Descent (SGD).

SGD works by considering only a sample or a small subset of data (*mini-batch*) for each iteration, calculating the gradient only for those data, making the procedure way faster than vanilla GD. Comparing the plot of the solution during various iterations, it can be seen that both SGD and GD arrive to a minimum, but SGD has a more "noisy" path. This because of the size of the set used for solution's updating.

In the NN training main steps are the *forward propagation* and the *backward propagation*, which form the *backpropagation algorithm*. The basis steps of this algorithm are the following:

Initialize NN weights at random and do until convergence:

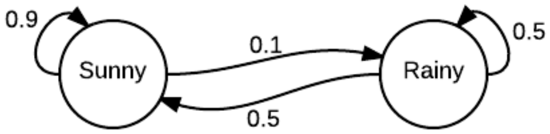

Figure 17: The Markov chain of the example - picture from Wikipedia

- pick a sample (x_k, y_k) from training data at random;
- *forward propagation*: compute values $v_{t,j}$ of all nodes of the NN with current weights;
- *backward propagation*: compute sensitivities of every node and layer (δ_j^t) and update all weights $w_{i,j}^{(t)}$ with SGD;
- return weights if converged, otherwise repeat.

Training usually needs to be repeated several times in order to verify if SGD algorithm stops in a local minimum instead of a global minimum.

Markov Models Before entering into details of Markov Models, a simple Markov chain is defined.

Markov chains[13] Intuitively, a Markov chain gives a model to describe the behaviour of a discrete system which can be in a certain state in a specific time. The system changes state in every time instant according to a probabilistic law. The most important fact of a Markov chain is that, if the system is in the state s at a time t, it can pass to another state v in the following time instant with a probability $P_{s,v}$ independent from the previous evolution of the chain. These probabilities $P_{i,j}$ are defined in a transition matrix which accounts all possibilities to pass from a state to another state in the next time instant.

As an example, consider a simple weather model. The situation to be be modeled will be the following: in a sunny day, the next day will be 90% sunny and 10% rainy, while in a rainy day the next day will be sunny with 50% of probability and 50% rainy. The Markov chain which represents this system is in Fig. 17

and the transition matrix T will be

$$T = \begin{pmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{pmatrix}$$

[13] Contents from http://www.dei.unipd.it/~fornasini/10Bis_Catene%20di%20Markov.pdf and Wikipedia at https://en.wikipedia.org/wiki/Examples_of_Markov_chains

where first row/column represent the state "sunny" and the other the state "rainy". The probability for the following day if today is sunny (i.e. $x_0 = \begin{pmatrix}1 & 0\end{pmatrix}$) is be given by

$$x_1 = x_0 T = \begin{pmatrix}0.9 & 0.1\end{pmatrix}$$

which corresponds to the first row of the matrix T.

For the day 2 probabilities can be computed as:

$$x_2 = x_1 T = \begin{pmatrix}0.86 & 0.14\end{pmatrix}$$

but also as:

$$x_2 = x_0 T^2 = \begin{pmatrix}1 & 0\end{pmatrix} \begin{pmatrix}0.9 & 0.1 \\ 0.5 & 0.5\end{pmatrix}^2 = \begin{pmatrix}0.86 & 0.14\end{pmatrix}$$

which gives the final rule for the prediction for the n-th following day:

$$x_n = x_0 T^n = x_0 \begin{pmatrix}0.9 & 0.1 \\ 0.5 & 0.5\end{pmatrix}^n$$

From a Markov chain steady-state probability can be also calculated. Steady-state probability is defined as $\lim_{n\to\infty} x_n$. It can be different from 0 only if there is at least one T^n with all non-zero entries. The steady-state probability for all states of a Markov chain can be calculated knowing that, if $q_i = P_i^{steady}$:

$$\begin{pmatrix}q_1 & ... & q_n\end{pmatrix} T = \begin{pmatrix}q_1 & ... & q_n\end{pmatrix}$$

since q_is are independent from initial conditions. Using the fact that $q_1 + ... + q_n = 1$, a linear system can be obtained and q_is can be obtained.

Hidden Markov Models[14] In previous Markov models, states are explicit (each one with a name) and observable because there are some observations which identify singularly each state. As an example, a semaphore can be modeled as a Markov process and states are explicit (light lamps) and observable (through a camera). In these conditions, if noise is not present, a state is always observable through observation, but with noise the prediction for the following state can become unfeasible. Hidden Markov Models are based on an evolving system where some *hidden states* are defined (hidden because of the noise). These states cannot be directly seen, but only another phenomenon related to them (how cars in the street behaves). If the relation between observations and hidden states can be found, an HMM can model with probabilities the dynamic of that system.

[14] Contents from Wikipedia at https://it.wikipedia.org/wiki/Modello_di_Markov_nascosto (in Italian)

A Hidden Markov Model (HMM) is a special type of Markov chain where:

- states cannot be directly observed;
- the chain has a certain number of states;
- states evolve according to a Markov chain;
- every state generates an event with a probability distribution which depends only on that state;
- the event can be observed but the state can not.

Analysis of HMMs tries to recover the sequence of states from observed data. Data have to be generated through a generative process in the form of state sequences.

An HMM is formed by these basic components:

- Hidden states $\mathbf{S} = \{S_i\}$ for $1 < i < N$: the states which cannot be seen directly. In some cases, even they are hidden, a physical meaning can be inferred.
- A probability of initial states $\pi = \{\pi_i\}$, where $\pi_i = P(Q_1 = S_i)$ for $1 < i < N$
- A transaction probability between states $\mathbf{A} = \{a_{i,j}\}$, where $a_{i,j} = P(Q_t = S_j | Q_{t-1} = S_i)$ for $1 < i, j < N$
- An emission probability of symbols $\mathbf{B} = \{b_j(v)\}$, where

  $$b_j(v) = P_{v \in V} \left(v \, \text{is emitted at time} \, t | Q_t = S_j \right)$$

  and V is the set of symbols observable of the system.

An HMM can be used for the following base problems:

- *Evaluation*: given a sequence s and a model m, find $P(s|m)$. It can be done using the *forward-backward* procedure.
- *Decoding*: given a sequence s and a model m, find the optimal sequence of states which generates the sequence s. It can be done with the Viterbi aglorithm.
- *Training*: given a set of sequences S, find the model m such that $P(S|m)$ is maximum. It can be done with the *Baum-Welch* algorithm, which uses the *Expectation-Maximization* procedure seen before for GMMs, or with the *Viterbi training* algorithm. In reality, HMMs can be seen as a generalization of GMMs.

In this document, training and evaluation of an HMM is discussed. An important thing about training is that the *Baum-Welch* algorithm implements a sort of gradient-descent algorithm, which is a local optimizer because of the multi-modality of the log-likelihood function. For this reason, it is important to initialize well transaction and emission probabilities in order to not stop to a local maximum.

The *Viterbi algorithm* [19], instead of using the E-M algorithm, has the main purpose of discovering the sequence of HMM sates which makes the joint probability of observation maximum. Since all possible paths of an HMM must be evaluated, the problem to solve becomes extremely heavy also for small HMMs. To overcome this issue, the algorithm uses dynamic programming to keep track of consequent updates of score values for each path. Precisely, the procedure creates a cost grid (*trellis diagram*) where HMM states are vertically-placed and the sequence of observations is horizontal-placed. In each cell (i,t) an accumulated probabilistic score $\alpha t(i)$ is calculated, which is the probability to reach state i through an optimal sequence of states after the first t observations. The *Viterbi* algorithm computes $\alpha t(i)$ for each node starting from $t = 1$ (first column). The maximum value of the last column is the probability to emit emit the complete sequence of states through an optimal sequence of states, extracted by a *backtracking* operation. *Viterbi training* segments data and then applies the Viterbi algorithm to get the most likely state sequence in the segment, then uses that most likely state sequence to re-estimate the hidden parameters. This procedure doesn't give the full conditional likelihood of the hidden parameters, as instead *Baum-Welch* algorithm does, but is significantly faster than the latter one.

SECTION 2

MATERIALS

In this section the material needed for analysis and further elaborations of the auditory models will be presented. Auditory models used for the study will be analyzed, compared and tested with different machine learning methods: Gaussian Mixture Models, Neural Networks and Hidden Markov Models.

Fig. 18 shows how every tool described before is combined with the others to perform the simulation tasks.

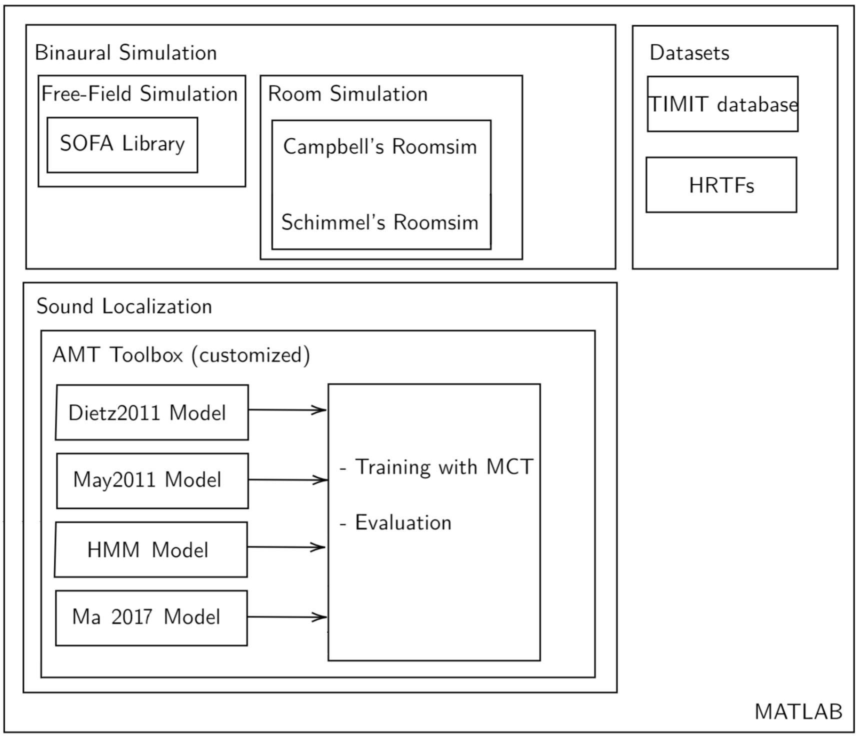

Figure 18: Schema with tools used for tasks in this manuscript

2.1 Software, libraries and databases

This work is based on the following software:

2.1.1 Software

- MATLAB[15] from MathWorks as development environment for audio modeling, processing and simulation.

[15] https://www.mathworks.com/products/matlab.html

- *Room acoustic simulators*: Roomsim in the two versions designed by Schimmel [34] and Campbell [7].

2.1.2 Libraries

- The *Auditory Modeling Toolbox*[59], a toolbox which contains several implementations of different auditory models.

- The SOFA (*Spatially Oriented Format for Acoustics*) Library [37], used to load HRTFs wrapped in a public format.

- The NETLAB library for MATLAB, used to train GMM models [44].

2.1.3 Datasets

- The *TIMIT database*, a corpus of recorded speech samples for acoustic and phonetic studies; it has been used as source to train and evaluate models discussed in this document [18].

- MIT's *KEMAR HRTFs* to generate binaural signals from voices of the TIMIT database [17]. This dataset has been used in the compact version, with 72 azimuth of the round corner spaced by 5° at elevation 0° .

2.2 Auditory Models

The following models have been analyzed and extended:

- *"A Probabilistic Model for Robust Localization Based on a Binaural Auditory Front-End"*, May *et al.*, 2011 [38]

- *"Auditory model based direction estimation of concurrent speakers from binaural signals"*, Dietz *et al.*, 2011 [13]

Variants of the first model have been also tested:

- *"Exploiting Deep Neural Networks and Head Movements for Robust Binaural Localization of Multiple Sources in Reverberant Environments"*, Ma, May et al., 2017 [36]

- A simple Hidden Markov Model for DOA estimation

2.2.1 May's 2011 auditory model

The auditory model used in May's paper uses ITD (Interaural Time Difference) and ILD (Interaural Level Difference) as base features for the direction-of-arrival estimation, constrained to the frontal horizontal plane at zero elevation.

The model mimics the processing of the peripheral human auditory system by using a Gammatone filterbank with hair cell processing. Input sound is decomposed into 32 auditory channels thanks to a 4th-order Gammatone filterbank, with center frequencies distributed between 80 Hz and 5 kHz. Later on, a halfwa1ve-rectification with square-root compression is used to simulate neural transduction. Binaural cues are calculated using a 20 ms window with a sampling frequency f_s=44.1 kHz and a frame overlapping of 50% to follow rapid changes in multi-source scenarios. Each auditory channel is processed to find ILD by computing the ratio between the energy integrated in the time interval between left and right ears. ITD, instead, is extracted for every frame as $(\tau + \delta)/f_s$ where τ is the time lag which maximizes the normalized cross-correlation and δ is the peak position relative to τ.

Formulas for normalized cross-correlation $C_i(t, \tau)$ and peak position $\hat{\delta}_i(t)$ are reported in Equations 24 and 25, where t is the time in milliseconds, τ is the delay in milliseconds, l_i, r_i and C_i are the left, right and cross-correlation input functions, W the window size and $\overline{l_i}$ and $\overline{r_i}$ the average values of input left and right signals.

Finally, azimuth estimation is done through a probabilistic model based on GMMs (Gaussian Mixture Models). The ML model has been applied to a features' space created by combining computed ITD and ILD from each frame, for each azimuth and Gammatone filter.

The estimated azimuth is then computed by taking the direction which maximizes the log-likelihood of the single observation across all Gammatone channels.

A more detailed schema about May's 2011 model is shown in Fig. 20.

The training with GMMs for the original auditory model has been done using a simulated room with [7] as described in Section 3 for fixed positions. Source signals were mono voices from TIMIT database, convolved with BRIRs obtained by the room simulator, where the KEMAR dummy head receiver was located at 1.5 m radial distance from the sources. Some criteria have been introduced to remove noisy samples in order to ensure a good training:

$$C_i(t,\tau) = \frac{\sum_{n=0}^{W-1}(l_i(t \cdot W/2 - n) - \bar{l}_i)(r_i(t \cdot W/2 - n - \tau) - \bar{r}_i)}{\sqrt{\sum_{n=0}^{W-1}(l_i(t \cdot W/2 - n) - \bar{l}_i)^2}\sqrt{\sum_{n=0}^{W-1}(r_i(t \cdot W/2 - n - \tau) - \bar{r}_i)^2}} \tag{24}$$

$$\hat{\delta}_i(t) = \frac{\log C_i(t, \hat{\tau}_i(t) + 1) - \log C_i(t, \hat{\tau}_i(t) - 1)}{4 \log C_i(t, \hat{\tau}_i(t)) - 2 \log C_i(t, \hat{\tau}_i(t) - 1) - 2 \log C_i(t, \hat{\tau}_i(t) + 1)} \tag{25}$$

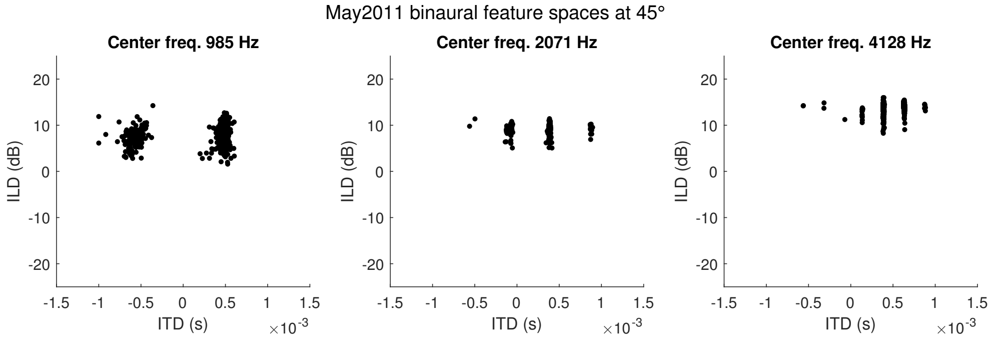

Figure 19: Examples of binaural features' space for azimuth = 45°

- remove frames whose energy level dropped by more than 40 dB with respect to the samples' global maximum;
- keep frames where the target source was stronger than the interfering source;
- remove frames whose amplitude of cross-correlation was below a threshold θ_c set by inspection, to consider only sounds not dominated by room reflections;
- remove frames where maximum value of cross-correlation corresponded to one of the most lateral time lags of ± 1 ms.

This model has been used several times as base point for other more complex models, as [36] and [56]. The model described in [36] will be further discussed.

2.2.2 Dietz's 2011 auditory model

Dietz's 2011 azimuth estimation model works on the horizontal plane at zero elevation as the May's one. It is an improved version of the model presented in [12], adding some extra processing of the binaural features for the extraction of the DOA. First of all, interaural parameters are extracted by implementing a processing pipeline inspired on the audiology literature:

- the middle ear has been modeled with a 500-2000 Hz first-order band-pass filter;
- the basilar membrane has been represented by a band-pass filterbank with a 4th-order Gammatone filterbank of 23 filterbands spaced of 1 ERB in 200-5000 Hz range;
- cochlea compression using instantaneous compression with power 0.4 after the filterbank;
- neural trasduction in inner hair-cells with half-wave rectification and a 770 Hz fifth-order low-pass filter;
- temporal disparities with a second-order complex Gammatone filter [14].

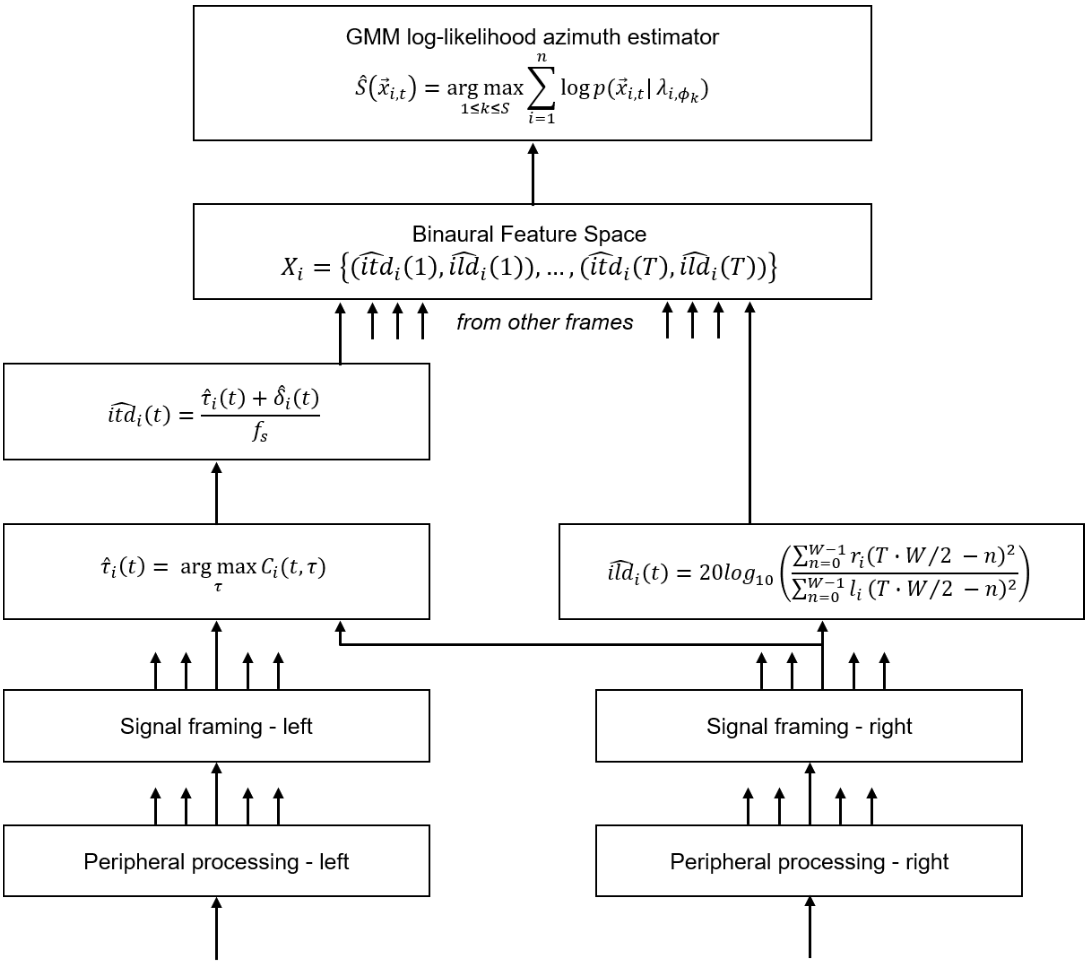
GMM log-likelihood azimuth estimator
$\hat{S}(\vec{x}_{i,t}) = \arg\max_{1\le k\le S} \sum_{i=1}^{n} \log p(\vec{x}_{i,t} | \lambda_{i,\phi_k})$
Binaural Feature Space
$X_i = \{(\widehat{itd}_i(1), \widehat{ild}_i(1)), \ldots, (\widehat{itd}_i(T), \widehat{ild}_i(T))\}$
from other frames
$\widehat{itd}_i(t) = \frac{\hat{\tau}_i(t) + \hat{\delta}_i(t)}{f_s}$
$\hat{\tau}_i(t) = \arg\max_{\tau} C_i(t,\tau)$
$\widehat{ild}_i(t) = 20log_{10}\left(\frac{\sum_{n=0}^{W-1} r_i(T \cdot W/2\ - n)^2}{\sum_{n=0}^{W-1} l_i\,(T \cdot W/2\ - n)^2}\right)$
Signal framing - left
Signal framing - right
Peripheral processing - left
Peripheral processing - right

Figure 20: May's 2011 model schema

After these procedures, complex signals $g(t)_l$ and $g(t)_r$ for both ears are obtained, each one with amplitude $a(t)$ and phase $\phi(t)$. The output of the peripheral processing for each ear is processed separately by a fine-structure filter (Q=3) and modulation filters (Q=8). These filters were used in order to have a better temporal resolution, since fine-structure information is important for frequencies lower than 1.4 kHz, and modulation becomes considerable above that frequency. In each separate process, $ITF(t)$ (Interaural Transfer Function) is calculated from the two $g(t)$ and, from this last function, $IPD(t)$ can be extracted as the argument of $ITF(t)$ and a specific low-pass filter. ITD can be also extracted dividing IPD by the mean instantaneous frequency of left and right signals. To derive ILD, instead, a second-order modulation low-pass filter with 30 Hz cut-off frequency has been used both for left and right signals, obtaining an energy ratio between right and left signal in dB. A schema describing the model is shown in Fig. 21.

The computed features were filtered by calculating the IVS (Interaural Vector Strength), which captures the IPD fluctuation and can be used as an equivalent alternative of IC (Interaural Coherence) which cannot be calculated because this model doesn't rely on cross-correlation:

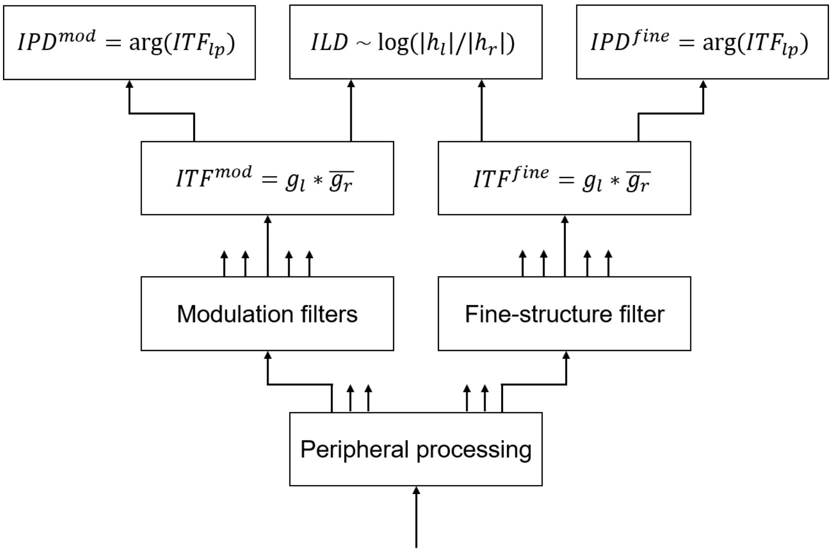
$IPD^{mod} = \arg(ITF_{lp})$
$ILD \sim \log(|h_l|/|h_r|)$
$IPD^{fine} = \arg(ITF_{lp})$
$ITF^{mod} = g_l * \overline{g_r}$
$ITF^{fine} = g_l * \overline{g_r}$
Modulation filters
Fine-structure filter
Peripheral processing

Figure 21: Dietz 2011 model schema - adapted from [13]

$$IVS_G(t) = \frac{1}{\tau_s} \cdot \left| \int_0^{\infty} \mathrm{d}\tau e^{i \cdot IPD(t-\tau)} e^{-\tau/\tau_s} \right| \tag{26}$$

where t is sample time and τ_s a time constant for temporal integration. A filter mask can be derived from IVS in order to extract reliable segments, based on two binary weights. The first weight discards every sample at time t which has $IVS \geqslant IVS_0$, where IVS_0 is an arbitrary-defined threshold. The second one excludes samples where the derivative of $IVS(t)$ is less than 0. This in order to remove corrupted samples which need an infinite time to drop below threshold IVS_0. After reliable segments extraction, IPD and ITD can be used in order to extract the correct azimuth. But there is an ambiguity problem because ILD reaches its maximum at about 60°; for example, angles $>$ 60° can be ambiguous with angles $<$ 60°, as described in Fig. 22.

To solve this issue, IPD has been used despite being ambiguous at frequencies above 700 Hz. But, after evaluation of some IPD studies, as reported in [13], it was reasonable to define that the absolute value of IPD allowed to localize two possible directions: $\alpha_1 = p_f(| \ IPD \ |)$ and $\alpha_2 = p_f(| \ 2\pi - IPD \ |)$, where p_f is a IPD-to-azimuth mapping function. By using this approach, IPD can be used to solve ILD ambiguities. Function p_f is a 9th-order polynomial function for continuous azimuth deriving. Its parameters have been found by training the model with a short set of 10 speech segments convolved with each one of 37 HRIRs from anechoic chamber. A histogram is then computed based on the azimuth estimations for each frame involved in the process. At the end, a sum of seven Gaussian functions is fitted on the histogram and the main peak is used to derive the speakers' direction. The Gaussian fit of the Dietz's model is not currently implemented in the toolbox [59] and it has been added using MATLAB built-in functions.

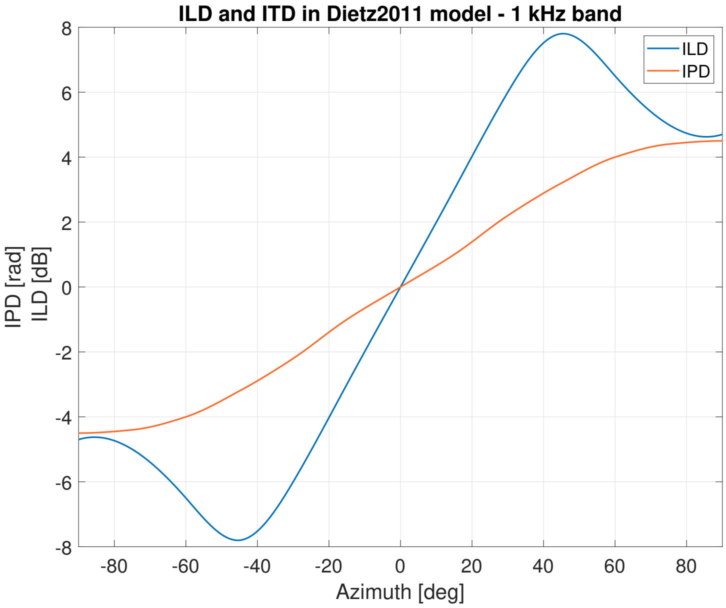

Figure 22: Dietz 2011 ILD and IPD behaviour - single sound source in free-field environment at 1kHz band

2.2.3 Ma's 2017 auditory model

The Ma's 2017 auditory model is based, differently from the two models above, on Deep Neural Networks (DNN). This model uses a set of specifically designed DNNs for each Gammatone filter, where the Gammatone filter-bank has been designed as [38]. Each DNN receives in input, for every frame, a number of features which increases with the sampling frequency of the signal, where all the features minus one are the frame's cross-correlations between left and right signals in an interval of ± 1 ms. The last feature in input to each DNN is the ILD of the frame, computed as described in [38]. For example, a binaural signal with sampling frequency $f_s = 16000$ Hz has 34 features, while a signal with $f_s = 44100$ Hz has 90 features. The features are Gaussian-normalized before DNN processing. Each DNN has been designed with an increasing number of hidden layers (limited to 2 in [36]) containing 128 hidden nodes plus a softmax output layer, which returns the probability of the frame of belonging to one of the azimuths trained for the model (angles from -180° to 175° every 5°). Outputs of the neural network are considered as $P(k|t,f)$, i.e. the probability of the azimuth k given the frame t and the filter f. These values are then integrated across frequency to obtain $P(k|t)$, the probability of a certain azimuth k given a specific frame at time t, with the assumption of no prior knowledge of the azimuth and equal probability for every source direction.

$$P(k|t) = \sum_{f} P(k|t,f) \tag{27}$$

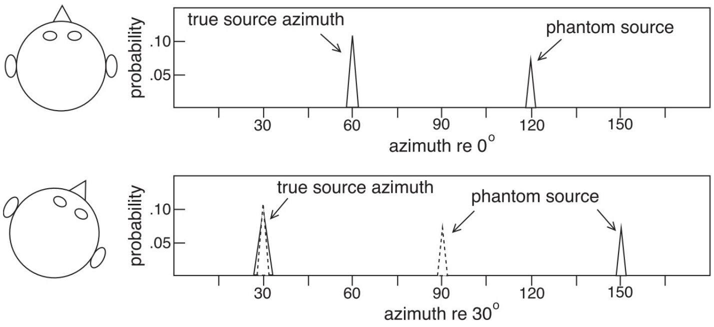

Figure 23: The head movement strategy described above - picture from [36]

Marginal probabilities $P(k)$ can be calculated averaging the probabilities obtained for every frame in the previous step.

$$P(k) = \frac{1}{T} \sum_{t}^{t+T-1} P(k|t) \tag{28}$$

This step is equivalent to considering a uniform probability distribution on t.

The best azimuth found by the procedure is, at this point, simply the one which has the higher probability in the whole signal.

$$\hat{k} = \arg\max_{k} P(k) \tag{29}$$

In addition to the evaluation using a single sound source to detect the DoA, a technique which uses a random head rotation in the range of [-30° to +30°] was also tested in the paper, increasing the accuracy of the model of about a 5-10 percent. This technique allowed to detect the phantom peak related to front-back confusion, as described in the Fig. 23. This part of the model has not been replicated in this manuscript. The training set used in [36] to train DNNs consisted in 30 voices for each of the 72 azimuths trained at a SNR of 20, 10 and 0 dB. White Gaussian noise with mean 0 and variance 0.4 has been added to each sample of the training set in order to avoid over-fitting. This model was then compared to GMM models to underline performance's differences: the DNN showed the best results, especially with more than one simultaneous voices. A diagram of the system proposed in this paragraph is in the Fig. 24. The Ma's 2017 model has been implemented from scratch both for training and for evaluation, trying to be more compliant possible with informations reported inside [36]. Instead of Stochastic Gradient Descent (SGD) with mini-batch size of 128 samples, Gradient Descent has been used because MATLAB does not support SGD for custom neural networks.

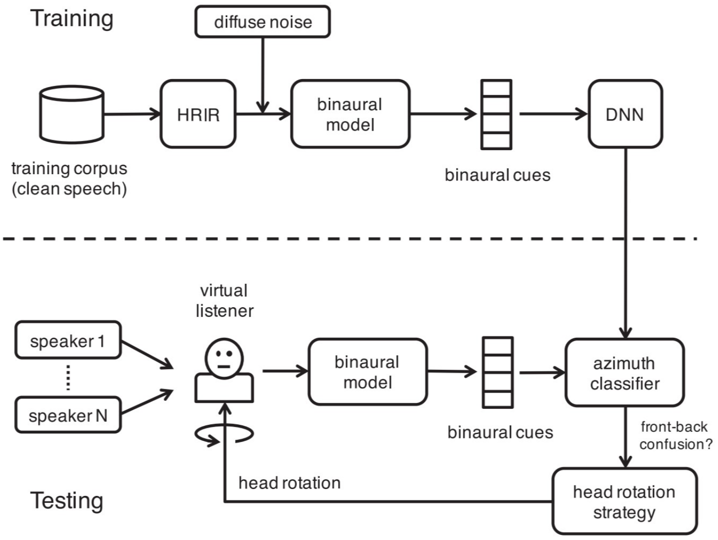

Figure 24: Ma's 2017 model schema - adapted from [36]

2.2.4 A simple HMM for DOA estimation

In addition to these documented models, a new model with HMMs has been created based on May's 2011 binaural features ITD and ILD. This model uses simple binary Markov chains for each trained azimuth and a Gammatone filterbank (32 filters for [38]), where hidden states are identified as values of every combination of feature and azimuth in the same Gammatone filter. The two states of every chain have been called "YES" and "NO"; the first one identifies a situation where a "valid" state for that azimuth has been found and the second one the opposite situation. Since MATLAB r2019a does not support HMMs with continuous hidden states, these have been calculated, for each Gammatone and feature, by computing the median of every value set for each azimuth and quantizing the features to train as the ones corresponding to the nearest median related to a certain azimuth. Median has been used as a simple outliers' removal technique. After some experimentation, the Viterbi training performed better than Baum-Welch, not only in speed, but also in overall DOA estimation precision. Initial guesses for each HMM has been done using the following setup:

- *Transition matrix guess*: YES-YES transition with probability equal to the sum of the probabilities of the n most-frequent hidden states in the training set related to a certain filter and azimuth until at least 80%, YES-NO with the complementary-to-1 value; for the NO-YES and NO-NO combinations, the same as above, but the opposite values. This approach has been used to lower probability of "YES" when an anomalous sample is found.

- *Emission matrix guess*: the first row (state "YES") has the probabilities of each state in the training set, while the second row (in the state "NO") tries to invert the behaviour with respect the row above, making a subtraction (one minus each value of the row above) and then normalizing to obtain a sum-to-one for each value of the row. This approach tries to model the behaviour that, if the chain is in the "NO" state, it is more probable that the following hidden state will be "not allowed" for that feature, Gammatone and azimuth. The opposite, instead, for the state "YES".

A separate training has been done for each Gammatone, azimuth and feature, giving a total of $|azimuths| \times |Gammatones| \times |features|$ (hidden) Markov chains.

Evaluation follows a procedure similar to the training one, but the discrete hidden states (and median values) are the same as in the training step for conformity. The sequence of states is then given to an evaluation *forward-backward* algorithm which computes the probability, for a certain Gammatone, that ITD or ILD is of a certain azimuth. The "YES" probability is accounted for each feature and threatening each feature as independent, the product between probability values related to each Gammatone and azimuth is done.

$$P^{MIX}_{f,a} = P^{ITD}_{f,a} * P^{ILD}_{f,a} \, where \, 1 \leq f \leq |Gammatones| \, and \, 1 \leq a \leq |azimuths| \quad (30)$$

Then, the probability of every azimuth is the product of the probability of the azimuth for all Gammatone filters (integration on all frequencies).

$$P_a = \prod_f P^{MIX}_{f,a} \quad (31)$$

The final estimated azimuth is computed as the one which occurs with higher frequency.

$$azimuth_{final} = \max_a P_a \quad (32)$$

A simplified schema of the processing both for training and evaluation procedures is in the Fig. 25.

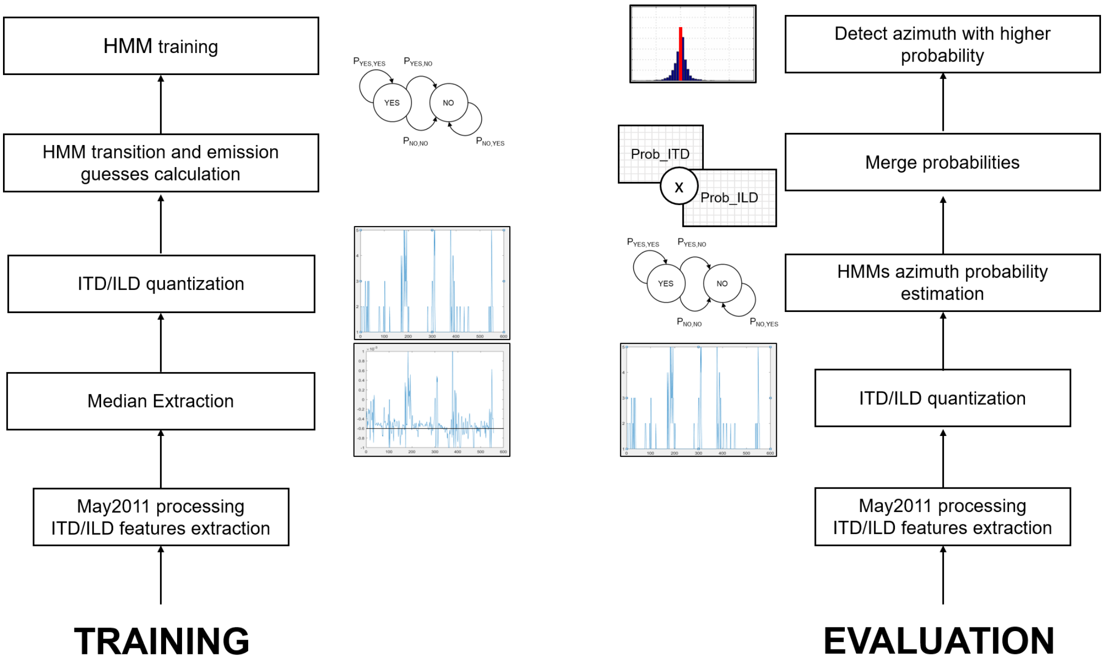
HMM training
$P_{YES,YES}$
$P_{YES,NO}$
YES
NO
$P_{NO,NO}$
$P_{NO,YES}$
HMM transition and emission guesses calculation
ITD/ILD quantization
Median Extraction
May2011 processing ITD/ILD features extraction
TRAINING
Detect azimuth with higher probability
Prob_ITD
X
Prob_ILD
Merge probabilities
HMMs azimuth probability estimation
ITD/ILD quantization
May2011 processing ITD/ILD features extraction
EVALUATION

Figure 25: HMM model schema

SECTION 3

SIMULATIONS

In this section, experiments performed for performance evaluation of models discussed before will be presented and explained. In Section 4, results about what is discussed here will be shown.

3.1 Metrics

Before listing the simulations, this section describes the metrics introduced to compare the results. For the experiments, the following metrics have been used:

- **Accuracy**: number of correctly-classified audio samples with an error less than 5° over the total number of samples, expressed in percentage, inspired by metrics in [38] and [36].
- **Correctness**: as Accuracy, but also considering as accurate a front-back estimation if the error is, as before, less than 5°. This metric was used for extended horizontal plane experiments.
- **Front-back confusion**: number of audio samples where the estimated DOA is the complementary angle with respect to the real one. *Example: real DOA of 20°, estimated DOA of 160°.*

In addition, additional data to compute confusion matrices were collected and other metrics from Information Retrieval have been tested. Confusion matrices are particular types of matrices where classification results are arranged counting how much data with a certain label x are classified with a label y. From these matrices, four types of values for every class/label can be extracted:

- *True Positives* (TP): the number of elements labeled x which are classified as x;
- *True Negatives* (TN): the number of elements not labeled x which are not classified as x;
- *False Positives* (FP): the number of elements not labeled x which are classified as x;
- *False Negatives* (FN): the number of elements labeled x which are not classified as x;

For a multi-class classification problem, these values can be calculated according to Fig. 26.

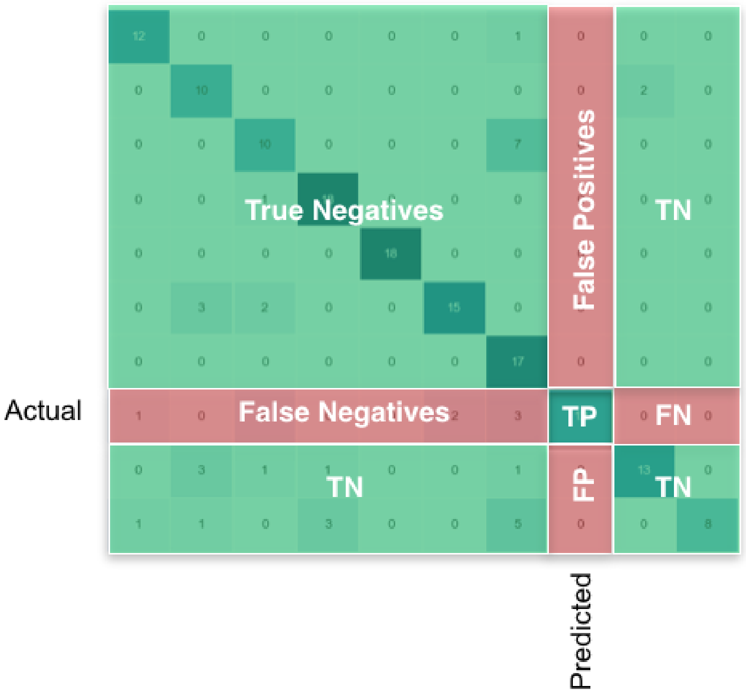

Figure 26: True and False Positives and Negatives on a confusion matrix. Picture from [16]

Now, the other metrics can be defined:

- **Precision**: $Pr = TP/(TP + FP)$, it is the number of correct predictions for a class x divided by the total number of prediction for the same class
- **Recall**: $Re = TP/(TP + FN)$, it is the number of correct predictions for a class x divided by correct predictions and predictions of values of other classes which are really belonging to x
- **F-measure**: the harmonic mean of Precision and Recall, $Fm = 2 \cdot (Pr \cdot Re)/(Pr + Re)$

3.2 Papers' models replica

In the first part of this work, some experiments found in the models' papers have been replicated in order to evaluate the performances of the models implemented in the toolbox [37]:

- May 2011 - Figure 4: anomalies with constant RT_{60} and different receiver positions.

[16] https://stackoverflow.com/questions/31324218/scikit-learn-how-to-obtain-true-positive-true-negative-false-positive-and-fal, downloaded on May 20th, 2019

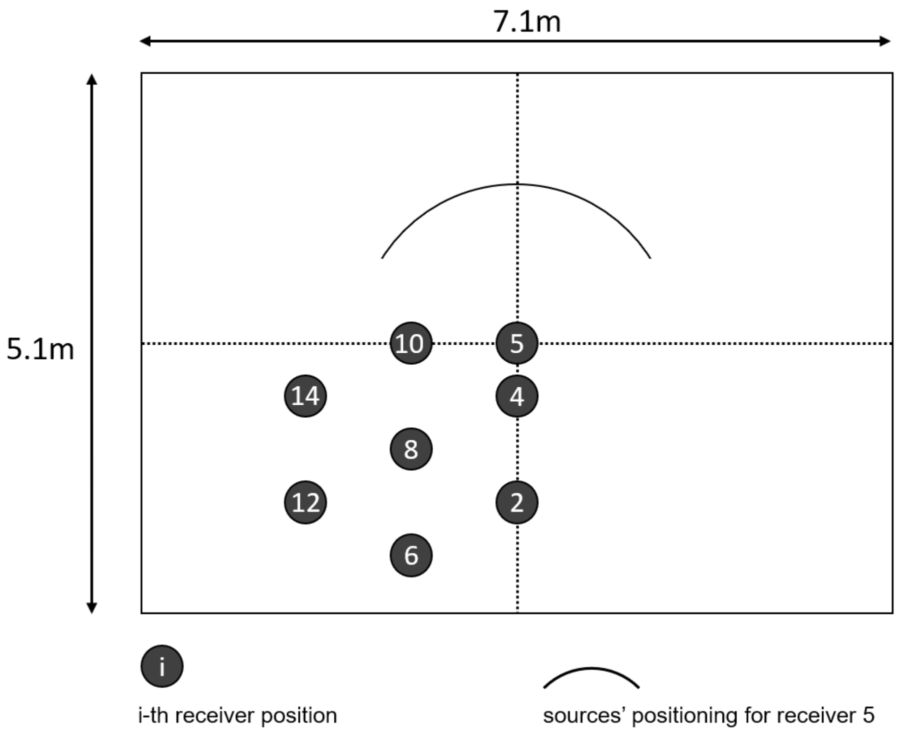

Figure 27: Room configuration for May's 2011 model simulations

- Dietz 2011 - Figure 5c: frequency charts of azimuth with one, two, three concurrent speakers.

For each experiment performances were tested for both models.

3.2.1 May's 2011 model experiments

May's model experiments were done in a reverberated room of $5.1 \times 7.1 \times 3$ m with different reverberation times. The general room configuration used for experiments in this manuscript is shown in the Fig. 27.

The evaluated positions for receivers are the ones marked withy, while the sources' positions are shown for the receiver number 5 as an example: 21 different azimuths from -50° to 50°, spaced by 5° and with a source-receiver distance of 1.5 m.

In the paper's experiment, Fig. 4 shows that was used an average reverberation time across the entire room of 0.69 s. The environments simulated in [38] did not reported the absorption coefficients used in the experiments. In order to find the correct ones, it was made an attempt to "reverse-engineer" the reverberation times and the estimation of absorption coefficients has been done using the RT_{60}'s values in [38] and the Sabine formula.

Under these conditions, the evaluation procedure has been made for every receiver and every source position with:

- 1: single source from -50° to 50°

Frequency [Hz]	125	250	500	1000	2000	4000	mean
RT_{60} [s]	1.26	1.03	0.69	0.48	0.37	0.29	0.69
α	0.096	0.118	0.176	0.253	0.325	0.406	

Table 1: Reverberation times used in [38] for the experiment in this manuscript and absorption coefficients from Sabine's formula used to obtain average RT_{60}=0.69s.

- 2, 3, 4: added other 1, 2, 3 random sources such that the distance between every source was at least 10°

The experiment tries to estimate anomalies in azimuth estimation, which are counted for every source in a specific scenario.

The experiment has been carried out by operating with with a Gaussian fitting with one Gaussian for May's model and 7 Gaussians for the Dietz's one.

Reverberation effects have been simulated using [34]. Monoaural voices from TIMIT database were normalized to 0 dB before processing, resampled from 16 kHz to 44.1 kHz to adhere to the one of RIRs, convolved with impulse responses and then resampled again to 16 kHz for memory usage reasons. Each experiment reported in this section has been replicated 3 times for each model in order to have a more robust result.

3.2.2 Dietz 2011 Experiments

Dietz's model experiments have been performed in a free-field environment, without reverberation effects, using the same room simulator as in the previous May's experiment, but setting reverberation parameters in order to obtain the most possible anechoic room. This was achieved by making the room very large and room's walls totally absorbing. Simulations have been done with three different scenarios: one, two and three competing talkers respectively at -30°, 0° and 30° inside a speech-shaped noise modeled from -90° to 90° such that the resulting SNRs were 0 dB and -6 dB and the source-receiver distance of 3 m. As for May's experiments, each scenario has been replicated for each model and number of concurrent voices 10 times.

Used azimuth estimation procedures have been the same of the previous experiment, but in this case it was not necessary to find the best-matching azimuth. The whole set of voices is from the TIMIT database, resampled to 44.1 kHz and normalized to 0 dB.

Results of May's model have been also calculated and compared to the Dietz's ones.

3.3 New experiments

In this section, an extension of the previous models will be described and will be the object of an additional performance analysis.

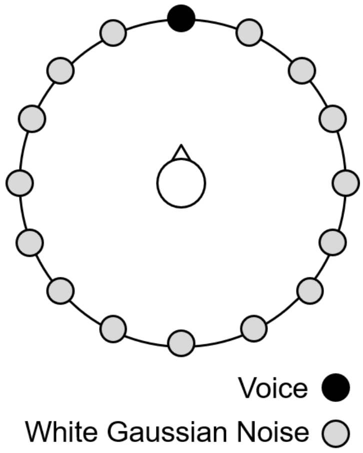

Figure 28: MCT single step for a voice at azimuth 0°

3.3.1 Models' extension

Models' extension has been made because previous models can actually estimate DOA from -90° to 90°. This modification allows to evaluate models' performances also for azimuths behind the listener and to understand how much front-back confusion determines ambiguity in DOA estimation.

It was necessary to adapt the May's 2011 model in order to extend their azimuth range through re-training, while the Dietz's model training with GMMs has been written from scratch using the same process flow of the May's one.

3.3.2 May's and Dietz's models re-training

Re-train has been made using MCT (Multi-Condition Training), inspired by [36]. Three different SNR values have been used (20, 10 and 0 dB), with round corner coverage from -180° to 175°, spaced by 5°. For every spaced azimuth and every SNR, 30 different voices from TIMIT database, normalized to 0 dB and resampled to 44.1 kHz, have been used, with a total number of 6480 combinations. Every single binaural voice sample is then resampled to 16 kHz in order to speed up computations. MCT works by putting a single voice source to a specified azimuth and White Gaussian Noise (WGN) to the other trained azimuths maintaining the desired SNR. A more-explaining picture of a single step of MCT is shown in Fig. 28.

The so-generated training set is given to two different training procedures called `may2011ttrain` and `dietz2011train`. The first function uses GMM as in the paper to train the azimuth estimation model, which works identically to the one included in the toolbox [37] used in this document. `dietz2011train`, instead, changes the ending behaviour of the relative model because it now does not rely anymore on a 9-degree polynomial for azimuth estimation, but on GMM as the May's one. This choice derived from the need to compare the models with the same identical ML procedures. For the first model, the features used are ITD and ILD for each Gammatone filter (and azimuth for training) and, for the second one, unwrapped ITD and ILD for each Gammatone (and also azimuth for training). In order to choose the number of Gaussians for GMMs which fits better data with less effort, a model-selection procedure has been implemented using the same dataset for both models.

May's paper in [38] reported that the number of Gaussians in GMMs for best performances is variable between 5 and 25, but the authors used the average value of 15 Gaussians as a sort of compromise between accuracy and computational costs. In addition, it was stated that increasing the number of Gaussians from a value of 11 did not improve significantly model's performances. Model selection has been introduced also to verify these facts.

For the model selection procedure, it was decided to train GMMs using 80% of voices inside the "TRAIN" folder of the TIMIT database, while the remaining voices have been used as the validation dataset, used to test performances of the trained models. The test set will be used only to verify performances of the best-selected model during final evaluations and not to choose the best number of Gaussians. Each training procedure has been done using the NETLAB library for MATLAB [44].

The SOFA dataset of MIT KEMAR with compact pinna for HRTFs has been used for dataset generation in all experiments.

It is important to notice that the maximum number of iterations of EM used to train both models was set to 100 because of long training times imposed by the Dietz's model, while the May's model in [38] has been trained with maximum 300 iterations of EM [38].

3.3.3 Ma's 2017 model training

The third model analyzed in this document does not have an implementation inside the Auditory Modeling Toolbox [37]. So, it was necessary to re-create it from scratch. To achieve this, a set of 32+32 Neural Networks (NNs) was created, trying to agree as much as possible to the specifications described into [36] considering also what can be done with MATLAB in training custom NNs. The NNs structure is the one shown in the Fig. 29. Training for each one of the 32 Gammatone filters used in [36] has been made using two NNs: first, the network called "NN1" was trained and, in the second step, weights of the only hidden layer of trained "NN1" have been used as the first hidden layer of "NN2" before training. The resulting NN for every Gammatone filter is then the trained "NN2" network. Every NN receives in input a set of 34 features (see Section 2 for details), while every set is about a single frame of every binaural audio file, obtained from the MCT procedure as in previous experiments. MATLAB has some limitations for training a custom NN. Therefore, the following values have been used for

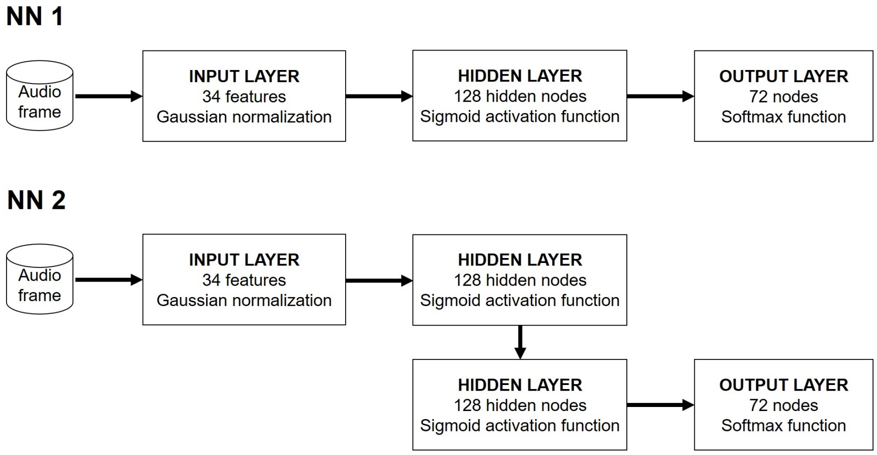

Figure 29: NNs structures used for [36]

each of the paramters described hereby: vanilla Gradient Descent with Momentum = 0.5 and Adaptive Learning rate have been used instead of Stochastic Gradient Descent with Momentum = 0.5 and mini-batch size of 128 samples. This could have made training longer and less accurate than the original. Despite this, in order to speed up the training, it has been done with the parallelization of the computation. To maintain the same training set size as in other two models analyzed while having a validation set, the complete dataset has been created in a such way that the number of voices for each azimuth was 36 for each one of the 3 SNRs, the 80% of features' data (randomly selected) has been used for training and the remaining 20% for validation. The training procedure for each NN was stopped after 20 iterations without a performance increase.

3.3.4 HMM model training

The HMM model has been trained using the same setup as in the GMMs models: 30 voices for each azimuth and SNR picked from the TIMIT database and auralized with MIT's KEMAR compact HRTFs given by a SOFA file with WGN (MCT procedure). The details on how features extracted using the May's auditory model are used in this model are in the previous Section 2.2.4.

3.3.5 Extended models evaluation

In order to evaluate extended models' performances, two different experiments have been set up. In the first one, the environment is anechoic while in the second one sources and receivers are inside a reverberant room simulated with [34] with the following parameters:

- Size (x, y, z): $5.1 \times 7.1 \times 3m$;
- Materials: as in Table 2
- Temperature: 20° C;

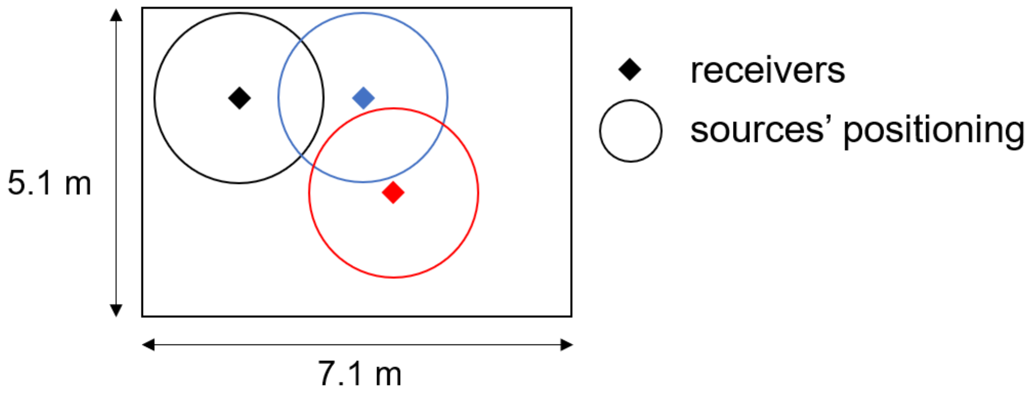

Figure 30: Configuration for sources and receivers in the reverberant room described above

- Humidity: 30%;
- Receivers' positions (x, y, z): (1.5, 1.5, 1.75) m, (3.55, 1.5, 1.75) m, (4.05, 3.05, 1.75) m;

							Reverb	small	medium	high
Frequency [Hz]	125	250	500	1000	2000	4000	Diffusion	Room1	Room2	Room3
Ceramic Tiles	0.01	0.01	0.01	0.02	0.02	0.02	0.3			floor
Wood panel	0.15	0.10	0.06	0.08	0.10	0.05	0.1		floor	walls
Carpet on concrete	0.02	0.06	0.14	0.37	0.60	0.65	0.1	walls	walls	
Seating people	0.55	0.86	0.83	0.87	0.90	0.87	0.5	floor		
							RT_{60}	0.37s	0.98s	1.88s

Table 2: Absorption coefficients and room configurations deployed for the experiment. RT_{60} was computed with the Sabine formula.

The absorption and scattering coefficients used for these simulations can also be found in [60]. The main motivation on using these coefficients was to make simulations with realistic materials and environments. The sources-receivers configurations for reverberant rooms can be found in Fig. 30 In these experiments, the execution will be similar: a number of one, two, three concurrent speakers at a certain distance (1.4m) for each azimuth will be proposed (one DOA every 5° to cover a round corner) to each receiver. Metrics used to compare results and the results of these experiments will be explained and shown in Section 4.

SECTION 4

RESULTS

4.1 Reproducibility of experiments

In this subsection, results obtained from replication of some experiments reported in [38] and [13] will be presented. Each experiment has been evaluated with the two original May's and Dietz's models.

4.1.1 May 2011 experiments

In Fig. 31, accuracy related to a single-maximum azimuth peak for each room position is shown for the two auditory models described in sections 2.2.1 and 2.2.2. The results are compared with those presented in [38]. Error bars report the standard deviation for each bar in three different runs. Each model uses a Gaussian fit in the last step to perform a more accurate prediction of the Direction of Arrival.

4.1.2 Dietz 2011 experiments

Fig. 32 shows the frequency of detected DOA for every azimuth spaced by 5° from -90° to 90° using both Dietz's 2011 with IVS mask and May's 2011 models. In addition, the results of the paper are reported to make a comparison with obtained data.

Note that Dietz and May's obtained results were normalized because of the different metric (and number of frames) used in histograms to show the probability of each azimuth.

4.2 Additional Experiments

In this section results about model's extension, training and evaluation will be presented.

4.2.1 Model selection

In Figs. 33 and 34 results about the model-selection for the GMM extended models are presented. The training for this task has been performed with a reduced dataset ten times smaller than the original one, and we used a number of Gaussian functions from 1 to 25. Plots indicate the trends of training and validation errors.

To extract the two best GMM fittings for each model from plots in Figs. 33 and 34, a compromise between computational cost and validation error has been considered. This compromise lead to choose 11 and 15 Gaussian functions for the May's model and 9 and 13 for the Dietz's model. These GMM models were then trained with the full training dataset and their training errors were used to choose the best number of Gaussian functions for each auditory model. The minimum training error was obtained for May's model with 11 Gaussians and the Dietz's with 9 Gaussians.

4.2.2 Models' extension experiments

The first part of the results will be about the free-field evaluation. The plots in Fig. 35 show a direct comparison between the various models in all scenarios, with a decreasing noise starting from a SNR of 0 dB until 20 dB.

Plots in Figs. 36 and 37 have been generated for the evaluations in reverberant rooms, showing a direct comparison of all models discussed in this manuscript considering different reverberation times and receivers' positions.

All plots have been reported only with relevant metrics cited in the section 3. The discussion of other metrics not used will be done in the next section 5.

4.2.3 Confusion matrices

In this section, the most relevant confusion matrices will be presented.

The first set of plots in Fig. 38 shows how well models analyzed in this manuscript perform in very noisy free-field environments, with a SNR of 0 dB.

The set of plots in Fig. 39 shows detailed performances of the models with a more attenuated noise at SNR of 20 dB. Fig. 40, instead, shows in detail the performances of every model in a central position (the third of ones indicated in section 3.3.5) with small reverberation times.

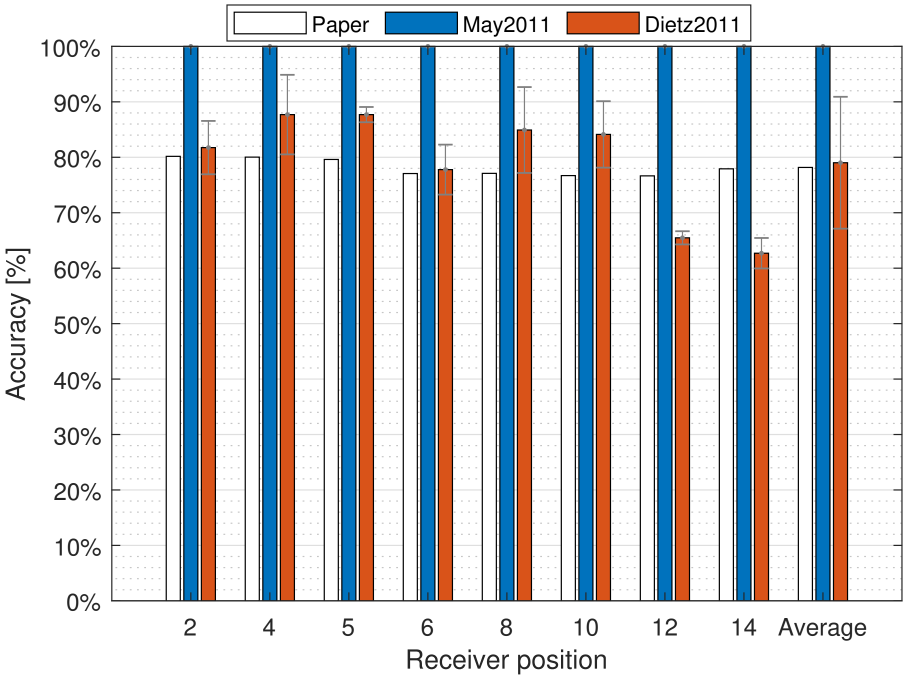

Figure 31: Accuracy for May's 2011 experiment - figure 4

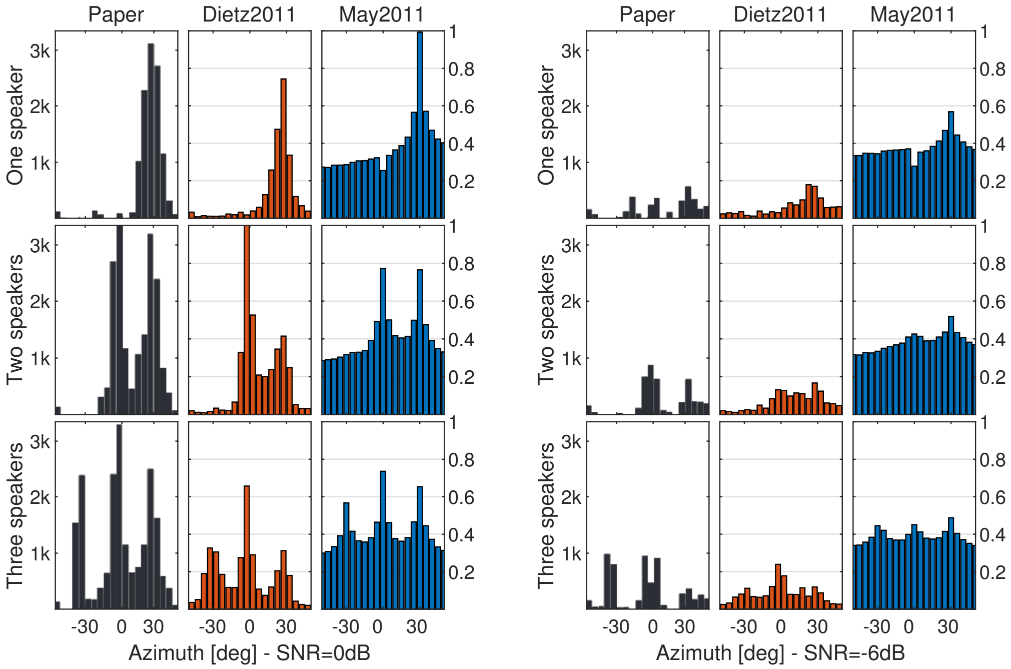

Figure 32: Histograms for Dietz 2011 experiments - SNR: 0 and -6 dB

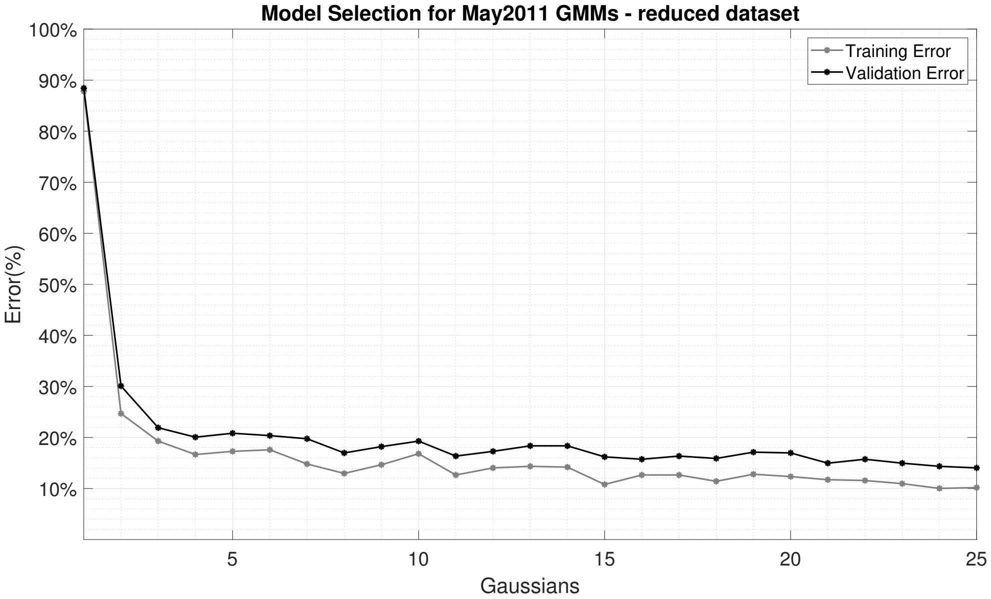

Figure 33: Model selection for May2011 model

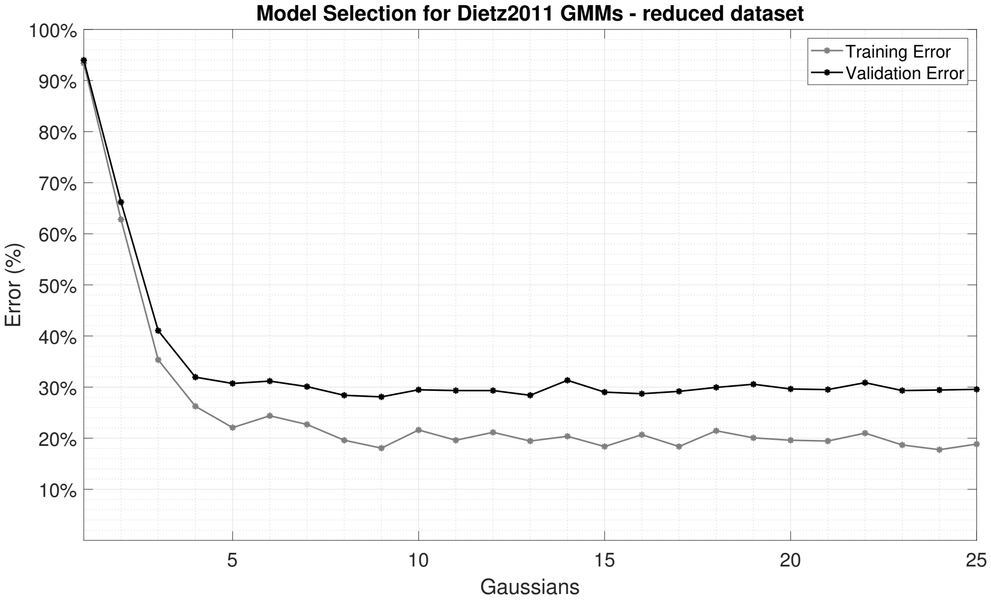

Figure 34: Model selection for Dietz2011 model

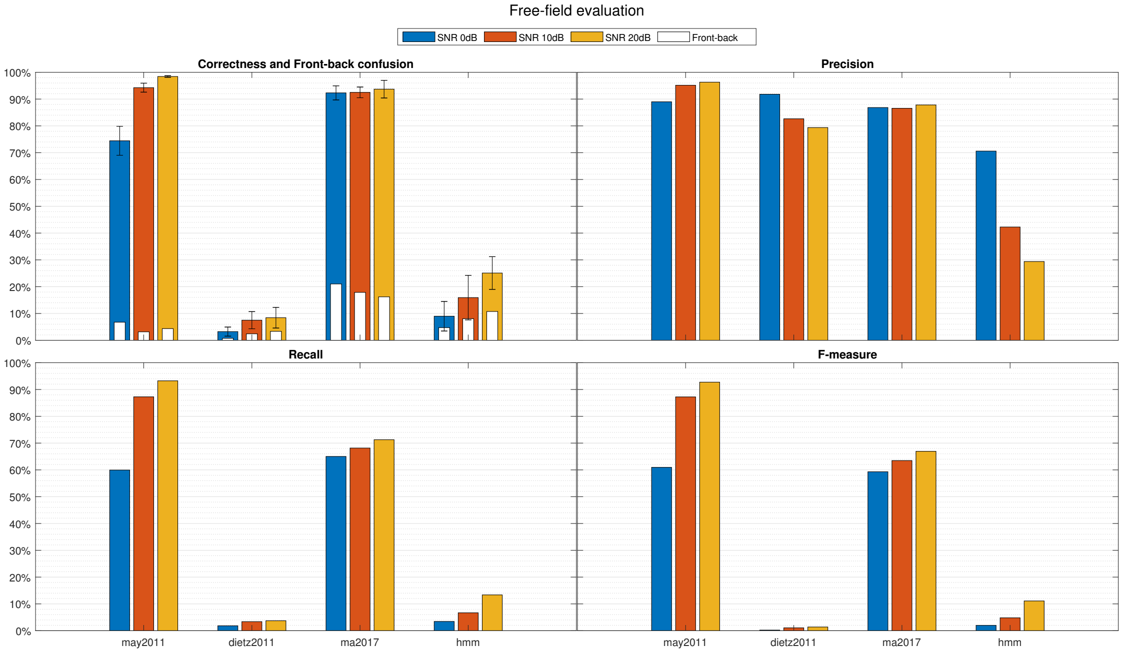

Figure 35: Average results for each model tested in the free-field scenario. Error bars report the standard deviation of results with different number of voices.

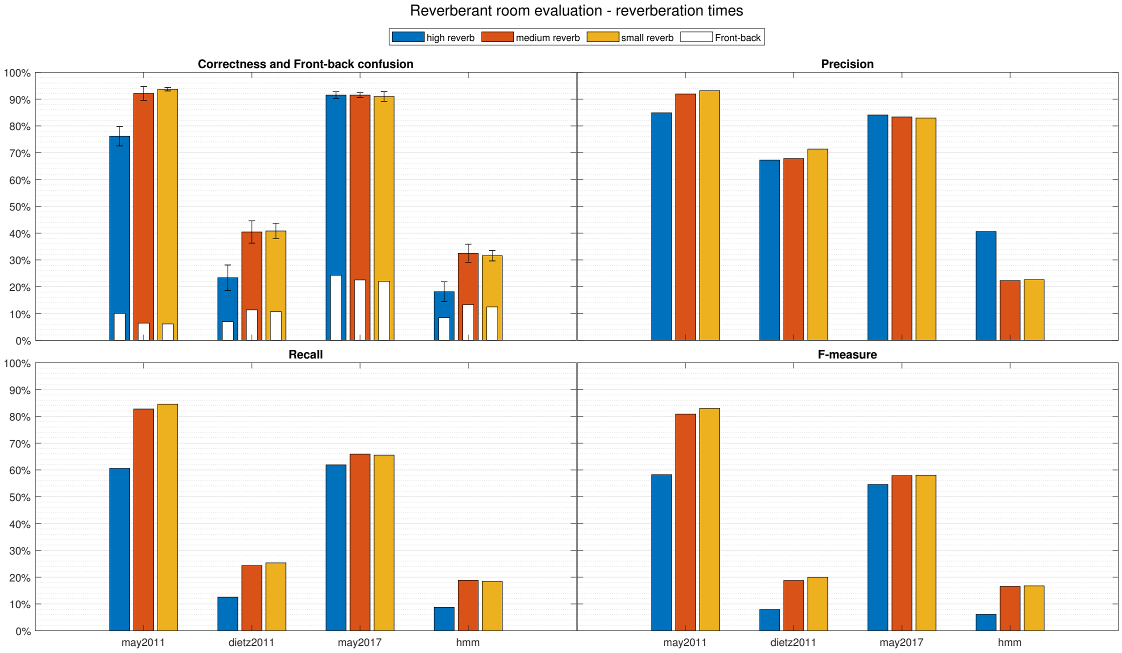

Figure 36: Average results for each model tested in the reverberant room scenario considering different reverberation times. Error bars report the standard deviation of results with different number of voices and positions.

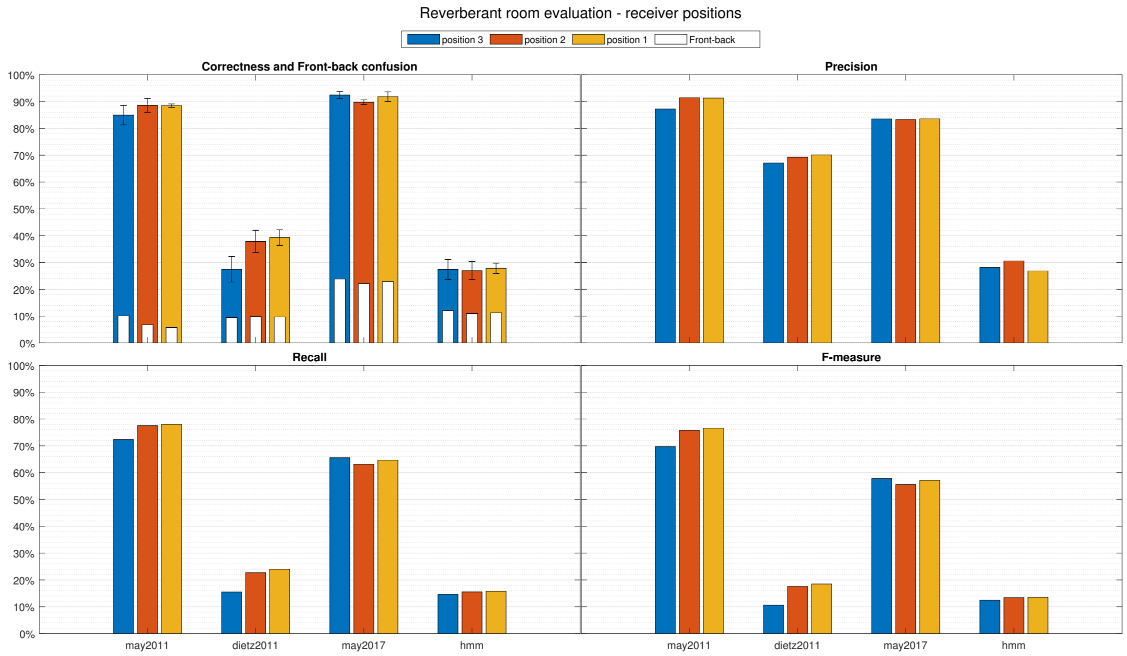

Figure 37: Average results for each model tested in the reverberant room scenario considering different receivers' positions. Error bars report the standard deviation of results with different number of voices and reverberation times.

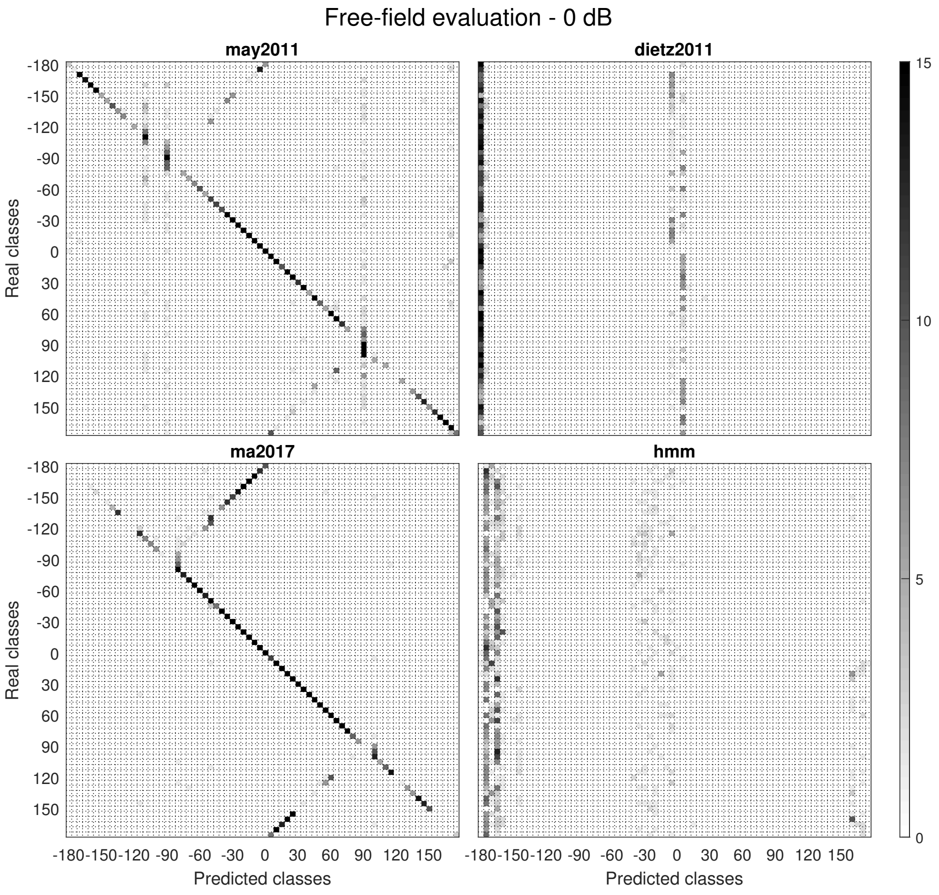

Figure 38: Confusion Matrices for all models in the free-field scenario at 0 dB of SNR

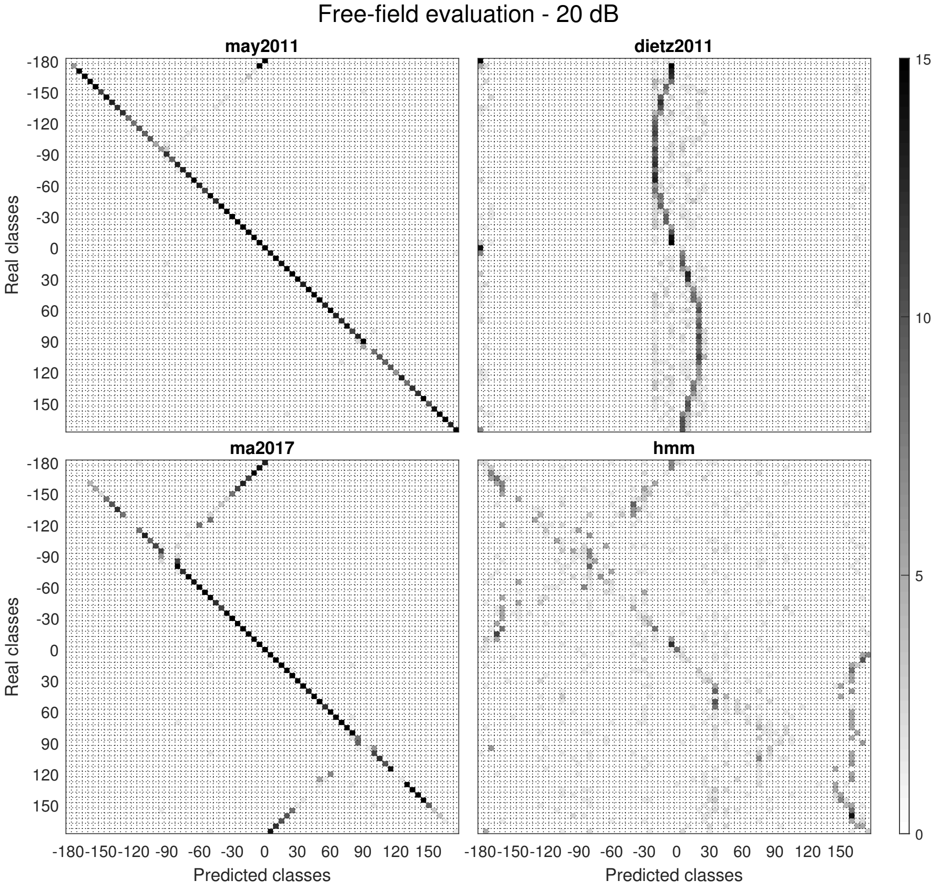

Figure 39: Confusion Matrices for all models in the free-field scenario at 20 dB of SNR

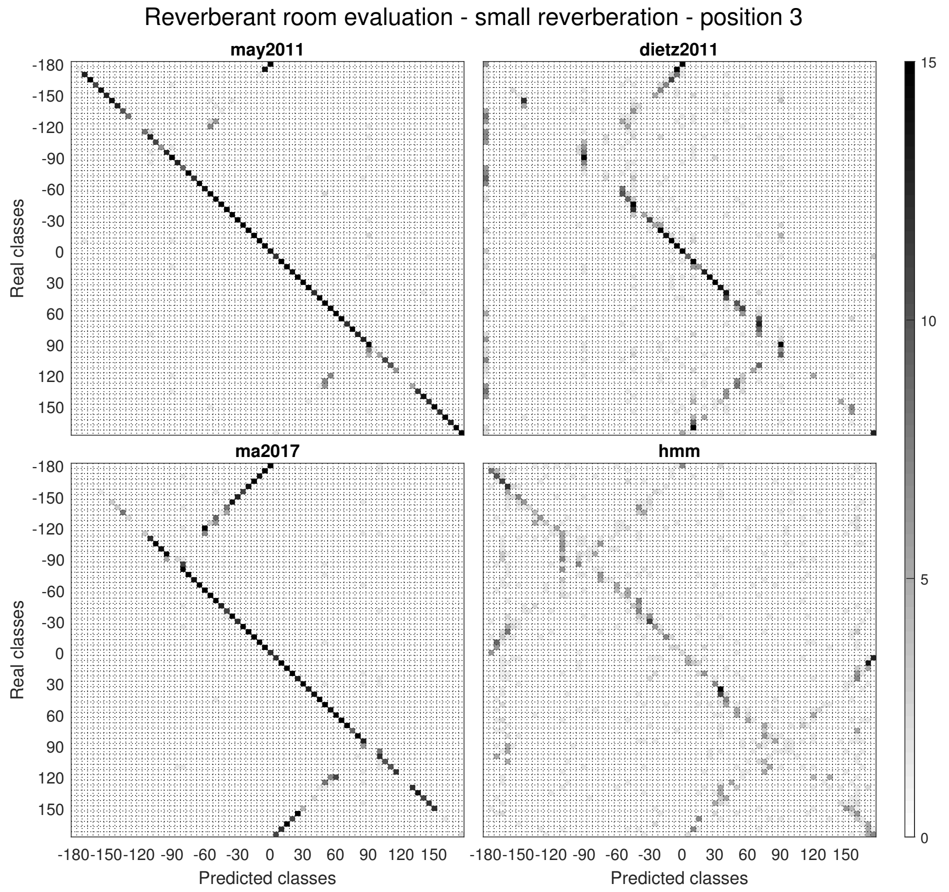

Figure 40: Confusion Matrices for all models in the reverberant room scenario with small reverberation time.

SECTION 5

DISCUSSION

In this section, results in section 4 will be analyzed and discussed.

5.1 Reproduction of papers' experiments

5.1.1 May's 2011 model paper's reproduction

The plot in Figure 31 denotes an important performance difference between the May's and the Dietz's models. May's model reported an accuracy of 100% for all the experiments. This means that the May's model can detect sources' azimuths with more precision and less errors, even considering more concurrent voices. The fact that the Dietz's model performs worse could depend on a bad fitting or on intrinsic problems of the auditory model not forseen by its authors, because evaluations in [13] has been done only with a small dataset of HRTFs in a free-field environment.

5.1.2 Dietz's 2011 model paper's reproduction

The results of the reproduced experiment in Fig. 32 were aligned with the ones reported in the Dietz's paper [13], both for 0 dB and -6 dB scenarios. The plot shows, for each source-receiver configuration as in section 3.2.2, the rate of detection of each azimuth in the range [-180 °, +175 °], where higher bars represent more frequent azimuth detection in audio's frames. Both Dietz's and May's models succeeded in the detection of the main sound sources. The presence of the IVS mask introduced by the Dietz's model allowed to filter the overall noise with respect to the May's model. In addition to the noise difference, peaks' shapes showed different characteristics: May's model ones were more defined and aligned to the real azimuth, while the Dietz's model ones were larger and less centered to the target azimuth. In this case, the comparison was qualitative rather than quantitative and it showed that May's model operates better if the number of simultaneous voices or the noise increases.

5.2 Additional Experiments

Overall models' comparison A deeper analysis on the results shows again that the May's 2011 auditory model performs significantly better than the Dietz's one: the percentage of correctly detected azimuths for the first model in the worst case was very high, while, for the second model, a large quantity of test data was predicted wrong. Ma's model seemed to perform better than May's one in noisy environments, while the HMM model performed better than the Dietz's model in free-field environments. F-measure and Recall seem to follow the Correctness metric, where the others, as Precision, Specificity and Accuracy gave very high results even if the previous "best" metrics showed low values. Recalling metrics' formulas in section 3, it can be noticed that Precision is a division between the number of correctly-classified samples for a class and the total number of predictions for that class. This can be misleading because, for example, in Dietz's model experiments where detected azimuths are compressed in a restricted interval, there are many cases where true and false positives are zero. For the definition of Precision, if the number of false positives is zero, the metric value is 100% because there aren't cases predicted wrong. Similar facts can be derived for Specificity: there are many cases where false negatives are zero as true negatives, bringing to the same exact condition seen in Precision. Accuracy also can be affected by these problems if false negatives and positives are zero. Recall, instead, tends to be more informative for the needs of the evaluation, since it measures the number of correct detection on the total number of samples with the same DOA. F-measure also demonstrated trends similar to the ones of the other two reasonably-good measures, but its behaviour depends on Recall and Precision. This last metric, however, gave controversial results. F-measure, however, if reported with other metrics which can explain the trend of data, can be also informative.

Effect of noise in azimuth estimation The noise, in this case speech-shaped with different SNRs of 20, 10 and 0 dB, influences the perceived azimuth correctness. The more is the noise, the more is the localization error and the front-back confusion in all models analyzed. The Ma's DNN is the more robust again when noise increases, while the HMM model has the highest ratio between front-back confusion and correctness values. This could depend on the better generalization given by the DNNs learning model using the full cross-correlation function compared to the GMMs and HMMs with only ITD and ILD.

Effect of reverberation in azimuth estimation Reverberation acts as the noise in the previous experiments, making the sound localization more difficult. As expected, when the reverberation time increases, the front-back confusion and the correctness return worse trends. This effect is particularly noticeable in the high reverberation room scenario, where all models decrease their performances significantly. An exception is represented by the Ma's model, whose performances decrease, but not as much as other models.

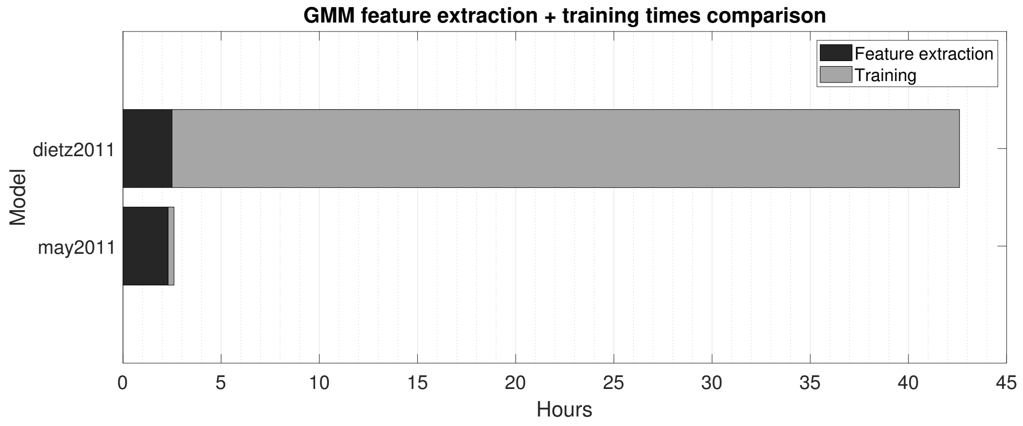

Figure 41: GMMs computation times for May's and Dietz's 2011 models trained on full dataset.

Effect of receivers' positions in azimuth estimation Performances related to receivers' position were also investigated: a position nearer to the walls makes the localization more complicated. This makes sense because a wall can make appearing a ghost sound source which can have the same or more relevance than the real sound source. In a more central position, as position 3 reported in Fig. 27, these effects are reduced, the reverberation is better distributed and it is easier for the model to find the correct azimuth. This effect is less noticeable in Ma's and HMM models since the number of correct-evaluated audio samples in percent remains more or less the same.

5.3 General Discussion

Complexity-related issues during training Some complexity problems have been found during the training of the Dietz's model with GMMs. This model, indeed, does not evaluate segments of the incoming signal but each sample: this resulted in a number of features for each incoming audio file which was about 150 times the number of features of the May's model. This leads to very long training times for GMMs using EM. Training on full dataset has been done for both models using the "Blade" cluster at the Department of Information Engineering (DEI) with 32 dedicated cores of Intel® Xeon Gold® 5518 processors and 1.5 TB of RAM shared by all logged users. To overcome this issue, it was decided to run the model-selection with a dataset 10 times smaller than the original one (72 DOAs with 3 SNRs of 20, 10 and 0 dB and 3 voices for each SNR, resulting in 9 voices for each azimuth). With this configuration, the model-selection for the Dietz's model took about one week with the cluster machines in the department, while the same procedure for the May's model took some hours.

An investigation on GMMs' covariance matrix type has also been done. It has been found that the best types were diagonal and full, with a very small difference between these two types (less than 2%). This leads to prefer diagonal covariance matrix for less computational costs.

At the end, it has been decided to pick 11 components for the May's model and 9 components for the Dietz's one. Features' extraction and training times for the best model chosen can be seen in Fig. 41.

Dietz's 2011 extended GMM model results The Dietz's model showed very low results with the GMM training. A further investigation shows that, in the free-field case, the model is not capable to detect correctly any azimuth with SNR of 0 dB, as shown in Fig. 38. The May's model, instead, is very reliable even in this noisy situation.

When SNR increases, another unexpected effect appears: the range of azimuth detected tends to remain constrained to central values, as shown in Fig. 39.

The situation changes in the second simulation with reverberant rooms: the model becomes more precise, as can be seen in Fig. 40.

The last Confusion Matrix shows an high tax of front-back confusion, starting from -90° to -180° and from 90° to 180° .

In addition to what has been stated before, ITD and ILD from the Dietz's model have not been processed with the IVS mask, because the mask is used after the computation of the IVS in a following step after binaural features' calculation.

GMM Binaural Feature Space comparison The difference in performances between May's and Dietz's GMM extended models can lead to considerations about the models' feature space: while in the May's model points are distributed according to regular patterns, in the Dietz's model the feature space seems to be more affected by outliers. Examples of binaural feature spaces for an audio sample of the training set and different filters can be seen in Fig. 42 The localization of these ITD-ILD points is very important to permit to GMMs to adequately fit the problem. The feature space provided by the Dietz's model does not show regular patterns as in the May's model and this makes very difficult for the GMM training procedure to find the position and the shape of clusters of points.

Ma's 2017 model implementation and analysis The Ma's model, as reported in the Section 2, has been trained in a slightly different manner with respect to the original work. For convenience, MATLAB has been used and a custom Neural Network has been developed to fulfill as much as possible the training characteristics described in [36]. Differences are about the training method, training epochs and validation-stop criterion. A maximum number of 1000 iterations for each NN and a 20 epochs validation-stop criterion have been used. Training has been done with CPUs because of lack of support for the `mapstd` function with GPUs. The entire training of 64 NNs requested more than two days to complete with the same machines used to train GMM models.

Sample Confusion Matrices for this model can be seen in Figs. 38, 39 and 40

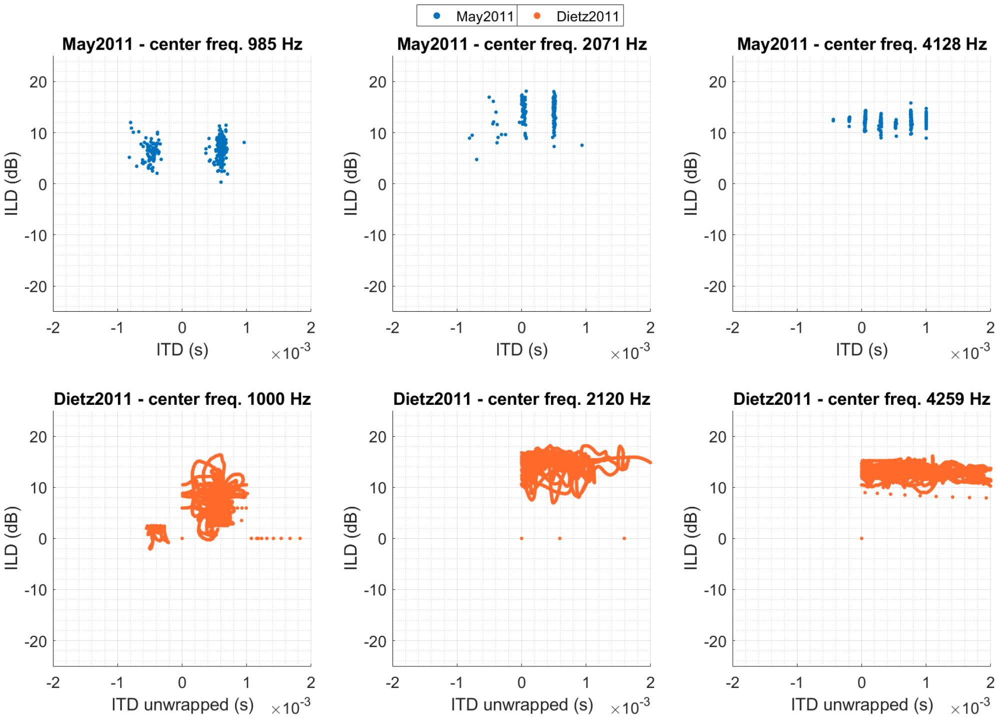

Figure 42: Binaural Feature Spaces for May's and Dietz's models. Audio sample from the training set, SNR: 20 dB and DoA: 60°.

The results presented in [36] are better than ones obtained in all simulations done in this manuscript; nevertheless, our results show a good generalization even if there is some front-back confusion. It seems that, with a Deep Learning framework which supports all parameters as in the original work and with the addition of the head rotation, the results in [36] can be reached.

HMM model analysis As stated in Section 2, the HMM model uses the same feature space of the May's 2011 model in [38]. Despite this, its performances are not so good. This can depend on several things:

- *ITD-ILD median filtering*: using median to extract characteristic values of binaural features for each azimuth can lead to an excessive approximation of real features' values and consequently to misleading detections. This problem can be overcomed by using a framework which lets to implement HMMs with continuous values.

- *ITD-ILD closeness for front-back angles*: For each angle of azimuth, there's another one (complementary) which have a very similar ITD and ILD for what has been stated in Section 1. Median filtering and discretization can determine two values for the correct and the front-back angle which are very near each other and can lead to incorrect azimuth detection. This is confirmed by the fact that there is a high ratio between correctness and front-back confusion.

- *Bad Transition/Emission matrices initialization*: this is not a simple task because the error function to be optimized is very complex and making a initial guess on these matrices is challenging.

The generalization on data, however, is satisfactory and better than Dietz's model looking at confusion matrices, as in Figs. 38, 39 and 40. As in the Dietz's 2011 extended model, in a very noisy environment as the free-field simulation at 0 dB, the confusion matrix showed a compression of all predictions to the 0° and 180° positions.

SECTION 6

CONCLUSION

In this work, two auditory models, the May's and Dietz's 2011 as in [38] and [13], have been reproduced and compared. In addition, two other Machine Learning techniques have been evaluated and compared with the results ot the extended GMM models. Regarding the model's comparison, it seems that the best model analyzed for localization is the May's extended GMM model, which performs better than all other models on average. This model can generalize well from the given training set and its performances are good due to the robust binaural features extracted from the data for each filter and DOA trained. The Ma's 2017 model is another valid model for azimuth estimation and its performances, compared to results in [36], have been partially proven by previous simulations and results in section 4. In addition, this last model has more stable performances than the May's model ones, especially in environments with a lot of noise. The Dietz's 2011 model with GMMs and the HMM one resulted very inaccurate and not suitable for reliable DoA estimation in the extended experiments, but the second model can be improved to obtain better performances because of the quality of the binaural feature space that is used by the model.

All the four models seen in this manuscript can be certainly improved. The May's extended GMM model can be trained with a superior number of iterations and give better performance. As stated in section 3, the original model has been trained with maximum 300 iterations instead of 100 of the current model. An internally-tested version of the model trained with maximum 600 iterations has given slightly better performances, but it has not been reported in this manuscript because the main focus was to compare the two GMM models with the same training configuration. The Dietz's extended model with GMM, instead, can be surely improved with other refinements in the auditory part: the theory behind the non-cross-correlation approach based on rate code theory proposed is relatively young and didn't enjoy the perfectioning of cross-correlation auditory models based on place theory, as stated by the same author of the model in [11]. In addition, the effect of more iterations of EM algorithm on the Dietz's extended model during GMM training has not been investigated for matters of time, but could improve performances even dramatically. The Ma's tested DNN model can be obviously improved by implementing the full pipeline for front-back confusion detection with head rotation and the complete training setup, but performances, even in this phase, are good and reliable for azimuth estimation. The HMM model, finally, demonstrates the difficulty of setting up a similar ML approach for DOA estimation. This also can be seen on academic literature: there are very few attempts of pursuing this task with HMMs, which are very suitable for speech recognition instead, as demonstrated by the Rabiner's work in [51].

APPENDIX A

CODE DOCUMENTATION

This Appendix has been written as documentation for the code used in this document. The source code of this thesis can be found in the repository azim-doa in the private GitLab of the DEI department, accessible at the URL `https://gitlab.dei.unipd.it/SMCrepos-auditory/azim_doa`.

A.1 Main structure

The contents of the repo have been arranged according to the following folder structure:

```
root
├─ experiments
│   ├─ modelselection
│   ├─ replicas
│   │   ├─ may2011
│   │   └─ dietz2011
│   └─ new
│       ├─ freefield
│       └─ room
├─ models
│   ├─ may2011
│   ├─ dietz2011
│   ├─ ma2017
│   └─ hmm
├─ libraries
│   ├─ ltfat
│   ├─ netlab
│   ├─ roomsim
│   └─ SOFA
├─ datasets
│   ├─ TIMIT
│   └─ HRTF
└─ results
    ├─ scripts
    ├─ dietz2011_replica
    ├─ may2011_replica
    └─ new_exps
```

A.1.1 Experiments

This folder contains the code used to replicate some experiments inside the Dietz's and the May's papers and the experiments for the evaluation of the extended models inside the thesis. The `may2011` folder contains the code for the replica of the experiment of Fig. 4 inside the model's paper, while the `dietz2011` folder contains the code for the replica of the Fig. 5 of the related paper. Both replicas can be simply launched using the scripts `launch_may2011_replica` and `launch_dietz2011_replica`. New experiments are inside the folder `new` and subdivided into the `freefield` and `room` evaluations, as documented in the thesis. These evaluations can be launched by using `launch_freefield` and `launch_room` MATLAB scripts. Model selection procedures are inside the `modelselection` folder, where the scripts `may2011_modelselection` and `dietz2011_modelselection` can be adjusted for training with the reduced or the full dataset using the MCT training as in the Ma's 2017 IEEE paper with the script `generateMCTDataset`.

A.1.2 Models

Here models' implementations are hosted. For the May's and the Dietz's models, both orginal and extended implementation have been inserted. The original implementations did not have a training procedure, which has been created for the extended versions of the model, but the code can be also used to retrain the GMM models using the preferred azimuth range. May's and Dietz's models code have been extracted from the Auditory Modeling Toolbox. Each model contains a function for the evaluation (with simply the name of the model) and the training function (followed by the `train` suffix). Details about the implementation of each training or evaluation procedure are inside the folder `misc` for every model analyzed into the thesis.

A.1.3 Libraries

Libraries used for the work are the Auditory Modeling Toolbox, from where Dietz, May's auditory models and the SOFA and LTFAT libraries have been extracted, and the NETLAB library for GMM training. The SOFA library allowed to process HRTFs while the LTFAT library was used for filters' implementations inside the Dietz's model. The `roomsim` folder contains the Schimmel's *roomsim* MEX executables for Windows and Linux platforms, with also the source code in C for further compilations. This folder also contains the functions `RIR_generation` and other utilities (`pointPolarToXYZ` and `estimateRT60`), used to interface every reverberant room simulation with the room simulator's code and to verify the RT_{60} obtained with the simulator.

A.1.4 Datasets

Datasets used for simulations are HRTFs from MIT's KEMAR in the compact version and the full TIMIT database. HRTFs are in the SOFA file format, while the TIMIT database is subdivided into two folders, `TRAIN` and `TEST`, used respectively as training/validation and test sets. The *roomsim* simulator uses the same HRTFs inside the used SOFA file.

A.1.5 Results

This folder contains both results obtained by running the scripts inside the `experiments` folder and the `scripts` to generate the plots inside the thesis using results' data. The `plot_bars` function can be used to plot bar charts inside the Results section of the thesis, while the `plot_CM` function can be used to plot confusion matrices as in the last figures of the same thesis' section. The plots used inside the paper can be generated with the functions `plot_paper_may2011_replica`, `plot_paper_dietz2011_replica`, `plot_paper_feature_spaces`, `plot_paper_freefield` and `plot_paper_room`.

A.1.6 Usage

The first step to use the script set of this repository is to run the script `initialize_all`, which will add all paths of the scripts inside the toolbox. Using the script `help` will show direct commands to launch the fundamental experiments, both replicas and new simulations. Results for new simulations have been already uploaded inside the toolbox, so scripts to plot data and confusion matrices can be already used without other operations.

APPENDIX B

ICA 2019 PUBLICATION

The abstract of the submitted publication for the International Congress of Acoustics in Aachen (Germany) from 9 to 13 September 2019 can be seen at the address `https://www.researchgate.net/publication/333603759_Auditory_models_comparison_for_horizontal_localization_of_concurrent_speakers_in_adverse_acoustic_scenarios`. The paper in [3] is a synthesis of what has been discovered about the May's and Dietz's 2011 models extended to the full horizontal plane and working with the GMM Machine Learning approach.

Many thanks to Roberto Barumerli, Michele Geronazzo, Profs. Giorgio Maria Di Nunzio and Federico Avanzini for the great tips and suggestions given for all the work!

REFERENCES

[1] *Glossary of Terms*, volume 30. Kluwer Academic Publishers, February 1998.

[2] R. Barumerli, M. Geronazzo, and F. Avanzini. Round robin comparison of inter-laboratory hrtf measurements – assessment with an auditory model for elevation. In *2018 IEEE 4th VR Workshop on Sonic Interactions for Virtual Environments (SIVE)*, pages 1–5, March 2018.

[3] Roberto Barumerli, Andrea Almenari, Michele Geronazzo, Giorgio Maria Di Nunzio, and Federico Avanzini. Auditory models comparison for horizontal localization of concurrent speakers in adverse acoustic scenarios. September 2019.

[4] Robert Baumgartner, Piotr Majdak, and Bernhard Laback. Assessment of Sagittal-Plane Sound Localization Performance in Spatial-Audio Applications. In *The Technology of Binaural Listening, Modern Acoustics and Signal Processing*, pages 93–119. January 2013.

[5] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[6] Jens Blauert. *Spatial Hearing, Revised Edition*. The MIT Press, 1996.

[7] Douglas R Campbell, Kalle J Palomäki, and Guy J Brown. A MATLAB simulation of "shoebox" room acoustics for use in research and teaching. page 4.

[8] David Fitzpatrick Lawrence C Katz Anthony-Samuel LaMantia James O McNamara Dale Purves, George J Augustine and S Mark Williams. *Neuroscience*. Sinauer Associates, 2nd edition, 2001.

[9] Mitchell L Day, Kanthaiah Koka, and Bertrand Delgutte. Neural encoding of sound source location in the presence of a concurrent, spatially separated source. *Journal of neurophysiology*, 108(9):2612–2628, November 2012.

[10] Prof. Fabio Vandin – Università di Padova. Neural networks slides for the machine learning course, not publicly available.

[11] M. Dietz, J.-H. Lestang, P. Majdak, R. M. Stern, T. Marquardt, S. D. Ewert, W. M. Hartmann, and D. F. M. Goodman. A framework for testing and comparing binaural models. *106*, November 2017.

[12] Mathias Dietz, Stephan D Ewert, and Volker Hohmann. Lateralization of stimuli with independent fine-structure and envelope-based temporal disparities. *The Journal of the Acoustical Society of America*, 125:1622–35, April 2009.

[13] Mathias Dietz, Stephan D. Ewert, and Volker Hohmann. Auditory model based direction estimation of concurrent speakers from binaural signals. 53(5):592–605.

[14] Mathias Dietz, Stephan D. Ewert, Volker Hohmann, and Birger Kollmeier. Coding of temporally fluctuating interaural timing disparities in a binaural processing model based on phase differences. *Brain research*, 1220:234–245, July 2008.

[15] J. F. Feuerstein. Monaural versus binaural hearing: ease of listening, word recognition, and attentional effort. *Ear and hearing*, 13(2):80–86, April 1992.

[16] J. L. Flanagan. Models for Approximating Basilar Membrane Displacement. *Bell System Technical Journal*, 39(5):1163–1191, 1960.

[17] William G Gardner and Keith D Martin. Hrtf measurements of a kemar. *The Journal of the Acoustical Society of America*, 97(6):3907–3908, 1995.

[18] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett. DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1. 93.

[19] Theodoros Giannakopoulos and Aggelos Pikrakis. *Audio Alignment and Temporal Modeling*, pages 185–207. 12 2014.

[20] Brian R. Glasberg and Brian C. J. Moore. Derivation of auditory filter shapes from notched-noise data. *Hearing Research*, 47(1):103 – 138, 1990.

[21] Dan FM Goodman, Victor Benichoux, and Romain Brette. Decoding neural responses to temporal cues for sound localization. *eLife*, 2:e01312, December 2013.

[22] Simon Haykin and Zhe Chen. The cocktail party problem. *Neural computation*, 17:1875–902, 10 2005.

[23] Gregory Hickok and David Poeppel. The cortical organization of speech processing. *Nature Reviews Neuroscience*, 8:393, April 2007.

[24] A J Hudspeth. Making an effort to listen: mechanical amplification in the ear. *Neuron*, 59(4):530–545, August 2008.

[25] Lloyd A. Jeffress. A place theory of sound localization. *Journal of Comparative and Physiological Psychology*, 41(1):35–39, 1948.

[26] Philip Joris and Tom Yin. Joris, p. & yin, t.c. a matter of time: internal delays in binaural processing. trends neurosci. 30, 70-78. *Trends in neurosciences*, 30:70–8, 03 2007.

[27] Eric R. Kandel, Thomas M. Jessell, James H. Schwartz, Steven A. Siegelbaum, and A. J. Hudspeth. *Principles of Neural Science, Fifth Edition*. McGraw Hill Professional, 2013.

[28] Saurabh Kataria, Clement Gaultier, and Antoine Deleforge. Hearing in a shoe-box: Binaural source position and wall absorption estimation using virtually supervised learning. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 226–230, New Orleans, LA, March 2017. IEEE.

[29] Takuji Koike, Hiroshi Wada, and Toshimitsu Kobayashi. Modeling of the human middle ear using the finite-element method. *Acoustical Society of America Journal*, 111(3):1306–1317, Mar 2002.

[30] Pat Langley. *The changing science of machine learning*, volume 82. March 2011.

[31] S. E. Levinson and Danfeng Li. A Bayes-rule based hierarchical system for binaural sound source localization. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, volume 5, pages V–521, April 2003.

[32] W. Lindemann. Extension of a binaural cross-correlation model by contralateral inhibition. I. Simulation of lateralization for stationary signals. *The Journal of the Acoustical Society of America*, 80(6):1608–1622, December 1986.

[33] R. F. Lyon and C. Mead. An analog electronic cochlea. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7):1119–1134, July 1988.

[34] Steven M. Schimmel, Martin F. Muller, and Norbert Dillier. A fast and accurate "shoebox" room acoustics simulator. pages 241–244.

[35] Ning Ma, Jose A. Gonzalez, and Guy J. Brown. Robust binaural localization of a target sound source by combining spectral source models and deep neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26:1–1, 07 2018.

[36] Ning Ma, Tobias May, and Guy J. Brown. Exploiting deep neural networks and head movements for robust binaural localization of multiple sources in reverberant environments. 25(12):2444–2453.

[37] Piotr Majdak, Yukio Iwaya, Thibaut Carpentier, Rozenn Nicol, Matthieu Parmentier, Agnieszka Roginska, Yôiti Suzuki, Kankji Watanabe, Hagen Wierstorf, Harald Ziegelwanger, et al. Spatially oriented format for acoustics: A data exchange format representing head-related transfer functions. In *Audio Engineering Society Convention 134*. Audio Engineering Society, 2013.

[38] T. May, S. van de Par, and A. Kohlrausch. A probabilistic model for robust localization based on a binaural auditory front-end. 19(1):1–13.

[39] D. McAlpine, D. Jiang, and A. R. Palmer. A neural code for low-frequency sound localization in mammals. *Nature neuroscience*, 4(4):396–401, April 2001.

[40] R. Meddis. Simulation of mechanical to neural transduction in the auditory receptor. *The Journal of the Acoustical Society of America*, 79(3):702–711, March 1986.

[41] John E. Mendoza. Trapezoid Body. In Jeffrey S. Kreutzer, John DeLuca, and Bruce Caplan, editors, *Encyclopedia of Clinical Neuropsychology*, pages 2549–2549. Springer New York, New York, NY, 2011.

[42] B. C. Moore and B. R. Glasberg. Suggested formulae for calculating auditory-filter bandwidths and excitation patterns. *The Journal of the Acoustical Society of America*, 74(3):750–753, September 1983.

[43] Jean K. Moore. Organization of the human superior olivary complex. *Microscopy Research and Technique*, 51(4):403–412, 2000.

[44] Ian Nabney. *NETLAB: Algorithms for Pattern Recognition*. Springer Science & Business Media, 2002.

[45] Stuart Russell; Peter Norvig. *Artificial Intelligence: A Modern Approach (2nd Edition)*. Pearson, 2016.

[46] Douglas L. Oliver. Ascending efferent projections of the superior olivary complex. *Microscopy Research and Technique*, 51(4):355–363, 2000.

[47] J P Demanez and L Demanez. Anatomophysiology of the central auditory nervous system: Basic concepts. *Acta oto-rhino-laryngologica Belgica*, 57:227–36, 02 2003.

[48] DN Pandya. Anatomy of the auditory cortex. *Revue neurologique*, 151(8-9):486—494, 1995.

[49] Marvin Papert, Seymour; Minsky. *Perceptrons*. The MIT Press, 1969.

[50] Prof. Davide Maltoni – Università di Bologna. Clustering slides, available at http://bias.csr.unibo.it/maltoni/ml/dispensepdf/6_ml_clustering.pdf.

[51] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[52] Patterson RD. Time-domain modeling of peripheral auditory processing: a modular architecture and a software platform. - PubMed - NCBI.

[53] Michele Rucci and Jonathan Wray. Binaural cross-correlation and auditory localization in the barn owl: a theoretical study. *Neural networks : the official journal of the International Neural Network Society*, 12(1):31–42, January 1999.

[54] Filip Munch Rønne, Torsten Dau, James Harte, and Claus Elberling. Modeling auditory evoked brainstem responses to transient stimuli. *The Journal of the Acoustical Society of America*, 131(5):3903–3913, May 2012.

[55] Jan Schnupp, Israel Nelken, and Andrew King. *Auditory neuroscience: Making sense of sound.* Auditory neuroscience: Making sense of sound. MIT Press, Cambridge, MA, US, 2011.

[56] Christopher Schymura, Thomas Walther, Dorothea Kolossa, Ning Ma, and Guy Brown. Binaural Sound Source Localisation using a Bayesian-network-based Blackboard System and Hypothesis-driven Feedback. September 2014.

[57] Alex Southern, D Murphy, Guilherme Campos, and Paulo Dias. Finite difference room acoustic modelling on a general purpose graphics processing unit. *128th Audio Engineering Society Convention 2010*, 3:1393–1403, 01 2010.

[58] Richard Stern and H. Steven Colburn. Theory of binaural interaction based on auditory-nerve data. iv. a model for subjective lateral position. *The Journal of the Acoustical Society of America*, 64:127–40, 08 1978.

[59] Peter Søndergaard and Piotr Majdak. The auditory modeling toolbox. In Jens Blauert, editor, *The Technology of Binaural Listening*, pages 33–56. Springer.

[60] Zühre Sü and Semiha Yilmazer. The acoustical characteristics of the kocatepe mosque in ankara, turkey. *Architectural Science Review*, 51:21–30, 03 2008.

[61] Sonia Tabibi, Andrea Kegel, Wai Kong Lai, and Norbert Dillier. Investigating the use of a Gammatone filterbank for a cochlear implant coding strategy. *Journal of Neuroscience Methods*, 277:63 – 74, 2017.

[62] Georg v. Békésy. Zur theorie des hörens bei der schallaufnahme durch knochenleitung. *Annalen der Physik*, 405(1):111–136, 1932.

[63] Sarah Verhulst, Torsten Dau, and Christopher A. Shera. Nonlinear time-domain cochlear model for transient stimulation and human otoacoustic emission. *The Journal of the Acoustical Society of America*, 132(6):3842–3848, December 2012.

[64] Eric Verschooten, Shihab Shamma, Andrew J. Oxenham, Brian C.J. Moore, Philip X. Joris, Michael G. Heinz, and Christopher J. Plack. The upper frequency limit for the use of phase locking to code temporal fine structure in humans: A compilation of viewpoints. *Hearing Research*, 377:109–121, June 2019.

[65] D. Wang and G.J. Brown. *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications (Wang, D. and Brown, G.J., Eds.; 2006)*.

[66] Xiaoqin Wang. The harmonic organization of auditory cortex. *Frontiers in systems neuroscience*, 7:114–114, December 2013.

[67] Jeremy M Wolfe. *Sensation and perception*. Sunderland, MA : Sinauer Associates, 2006.

[68] Muhammad S. A. Zilany and Ian C. Bruce. Representation of the vowel /epsilon/ in normal and impaired auditory nerve fibers: model predictions of responses in cats. *The Journal of the Acoustical Society of America*, 122(1):402–417, July 2007.