POLITECNICO DI TORINO

Master's Degree in Cinema and Media Engineering

Master's Degree Thesis

Study and evaluation of the use of binaural audio processing aimed at professional audio monitoring

Supervisors

Prof. Arianna ASTOLFI

Candidate

Filippo TREMATERRA

2024/2025

The purpose of this thesis is to study and analyze the subjective perception of both expert sound engineers and audio technicians from RAI regarding Head-Related Transfer Functions (HRTFs). More specifically, it aims to determine which of the tested HRTFs most closely resembles loudspeaker listening by subjecting participants to a tailored test designed for the evaluation of selected HRTFs.

The study seeks to compare and analyze various HRTFs, including those from artificial heads such as KEMAR and KU100, as well as personalized HRTFs of the different sound engineers, which are calculated and processed through specific procedures in order to maintain consistency between the test conditions. Additionally, it investigates differences among the selected HRTFs in terms of impulse response length and headphone equalization.

The subjective test is conducted in a 7.1.4 studio at RAI, where participants will listen to various musical and non-musical files to assess various attributes identified in the literature. The evaluation focuses on comparing the different HRTFs against a reference, aiming to determine which choice best replicates loudspeaker listening.

The results aim to understand the preferences of sound engineers and the differences between the HRTFs in terms of quality, spatial rendering, and perception. The test outcomes show a preference for the KU100 Far Field at 128 samples and indicate that personalized HRTFs are preferred with headphone equalization rather than without.

Summary

In recent years the demand for realistic and immersive audio experiences has significantly increased due to advancements in virtual reality (VR), augmented reality (AR), and 3D audio technologies. A key element of these experiences is the accurate spatial rendering of sound, which allows listeners to perceive the direction and distance of sound sources in a three-dimensional environment. The Head-Related Transfer Function (HRTF) plays a fundamental role in ensuring spatial precision and perception. This thesis investigates the use of binaural audio processing for professional audio monitoring, focusing on how sound engineers and audio technicians perceive and select different types of HRTFs. The study was conducted in collaboration with RAI CRITS in Turin and involved a subjective evaluation test in a controlled 7.1.4 surround sound room. The research compares different HRTF types, including artificial head models (Neumann KU100, KEMAR) and personalized HRTFs, considering different impulse response lengths and headphone equalization. The thesis presents a case study on the acquisition and processing of impulse responses (IRs) in a professional studio environment, with a custom pipeline developed for IR measurement, processing, and compensation, ensuring accurate spatial audio reproduction and cohesion between literature-based HRTFs and personalized ones. A listening test was designed following ITU-R standards (ITU-1284, ITU-2132, SAQI) and conducted using webMUSHRA. Sound engineers evaluated various HRTFs based on five perceptual attributes: sound color, externalization, localization accuracy, scene depth, and overall subjective quality. The results were analyzed using statistical methods, including boxplots, histograms, and ranking systems, to identify the most preferred HRTF configurations. The findings provide valuable insights into the role of individualized and non-individualized HRTFs in professional audio production and highlights the importance of selecting appropriate spatialization techniques. These results contribute to the ongoing development of binaural rendering and may guide future research on HRTF personalization in professional applications for RAI CRITS.

Acknowledgements

A Nonno Damiano.
Alla mia famiglia per essere sempre stata vicina a me.
A Luigi e Alessandro, punti saldi della mia vita.
Ai miei amici di Acri, compagni di una vita.
Ai miei fratelli di Torino, un pezzo del mio cuore.
A Giorgia, la persona che crede più di chiunque altro in me e che mi fa apprezzare tutte le piccole cose. Se il mondo mi vedesse con i tuoi occhi, sarei grande.

Table of Contents

List of Tables

List of Figures

Acronyms

3D
: Three-Dimensional

dB
: Decibel

dB SPL
: Decibels Sound Pressure Level

FFT
: Fast Fourier Transform

FIR
: Finite Impulse Response

HRIR
: Head Related Impulse Response

HRTF
: Head Related Transfer Function

IR
: Impulse Response

LFE
: Low-Frequency Effect

PICO
: Population, Intervention, Comparison, Outcome

SOFA

Spatially Oriented Format for Acoustics

Chapter 1

Introduction

1.1 Context and Motivation

In the last years there has been a significant evolution in audio technologies, leading to a remarkable increase in the demand for increasingly realistic and immersive sound experiences. This growth has been driven by technological advancements in digital content, music, video games, cinema, virtual reality (VR), augmented reality (AR), and 3D audio technologies. The ability to accurately position and localize sound sources in a three-dimensional space has become an essential aspect to ensure an engaging and believable experience, enhancing the listener's sense of presence and immersion.

One of the key tools for spatial sound rendering is the Head-Related Transfer Function (HRTF), which is a function that describes how an incoming sound wave is filtered by the anatomical characteristics of a listener before reaching the eardrum. This function depends on individual factors such as the shape of the head, outer ears (pinnae), and torso, influencing the perception of the direction and distance of a sound source. The accuracy of the HRTF is therefore crucial for achieving correct sound localization and ensuring a realistic listening experience.

However, anatomical variability among individuals implies that the use of generic or non-personalized HRTFs can introduce perceptual inaccuracies, reducing the effectiveness of spatial audio reproduction. Despite this, non-individualized HRTFs are often preferred due to their ease of implementation and lower cost compared to personalized ones, making them a viable solution in various application contexts. The choice of the most suitable HRTF is thus a matter of balancing perceptual accuracy, ease of application, and technological accessibility.

1.2 HRTF Acquisition and Types

There are various methods for acquiring HRTFs, each with specific advantages and limitations. Among the most common techniques are measurements conducted on artificial head models, such as the Neumann KU100 and KEMAR, which are widely used in research and audio production for their reliability and standardization. Alongside these traditional methodologies, more advanced techniques have been developed, including HRTF acquisition through 3D mesh scans of the listener's anatomy or the analysis of the subject's anthropometric parameters.

HRTF measurements can vary in terms of the number and placement of positions chosen for data collection. In some cases, a fixed configuration is used, where the subject remains static, and measurements are taken from different angles with strategically positioned loudspeakers. In other approaches, variability is introduced in the position of the listener or the loudspeakers to capture a larger number of spatial points and achieve a more detailed representation of the sound filtering characteristics.

1.3 Research Objectives

The primary objective of this thesis is to investigate the role of HRTFs in spatial sound perception among audio professionals, specifically sound engineers and technicians. In collaboration with RAI CRITS in Turin, an experiment will be conducted to subjectively evaluate different types of HRTFs and determine which one best approximates traditional loudspeaker listening.

The test will be carried out in a controlled listening environment with a 7.1.4 setup, allowing for a comparison between the perception of HRTFs derived from artificial head models and those obtained through standard static measurements. The analysis will be based on well-established methods and metrics from the literature, such as [1],[2] and [3], to achieve an accurate assessment of listener preferences. The ultimate goal is to understand which HRTF best preserves the characteristics of loudspeaker listening, contributing to the selection of the most suitable spatialization techniques for professional audio production and post-production.

1.4 Thesis Structure

The thesis is structured as follows:

- Chapter 2 - Research and Methods: Present the methodes used for the literature search and selection of paper.

- Chapter 3 - Theoretical Background: This chapter will examine the fundamental principles of spatial sound perception.
- Chapter 4 - Head-Related Transfer Function: A detailed description of HRTFs and their measurement methods.
- Chapter 5 - Case Study: Impulse Response Acquisition and Binaural Rendering: Description of the process for acquiring impulse responses, their processing, and their application in binaural rendering. The methodologies used and equipment setup will be detailed.
- Chapter 6 - Results: Visualization of the results.
- Chapter 7 - Discussion: Discussion of the results.
- Chapter 8 - Conclusion

1.5 Research Contribution

This research aims to assess the perceptual effectiveness of different HRTFs compared to loudspeaker listening. By conducting subjective tests with professional audio engineers and audio technicians in a controlled 7.1.4 environment, the study seeks to identify which HRTF best preserves spatial accuracy and realism relative to loudspeaker-based reference listening. The findings will contribute to optimizing HRTF selection for professional audio production and post-production, enhancing spatial audio rendering techniques in immersive applications.

Chapter 2

Research and Methods

The main objective of this section is to outline the methodology used to identify and analyze relevant research from 2019 to 2024 on HRTF measurement techniques and their influence on spatial audio perception. This review provides the foundation for the original work presented in this thesis by offering an overview of key studies and insights that informed the development of the proposed methodology and experiments. The steps included literature search, selection of papers, data analysis, and the synthesis of key findings.

2.1 Literature Search

The present section focuses on the search for relevant works that analyzed HRTF measurement techniques and their influence on spatial audio perception, either in controlled laboratory environments or in unconventional settings using non-standard methods. Specific conditions for binaural rendering were considered, particularly focusing on results and analyses from studies published between 2019 and 2024.

The strategy used to find important and useful articles was based on the Population, Intervention, Comparison and Outcome framework (PICO)[4] and was executed in the Scopus database. The search was based on terms related to HRTF measurement techniques, binaural rendering, and spatial audio perception.

Keywords
HRTF
HRTF Personalization
3D AND Mesh AND HRTF
HRTF Measurements
Individual HRTF

Table 2.1: Table of keywords.

These terms appeared in the title, abstract, or keywords of the documents. When the full text of a paper was not available in Scopus, additional searches were conducted through ResearchGate, Google Scholar, PubMed, Zotero, Research Rabbit, and Pico, a bibliographic search engine that provides integrated access to all bibliographic resources of the Politecnico di Torino. These tools were also used for general web-based literature searches beyond cases where Scopus did not provide access to the full text.

2.2 Selection of Papers

After gathering the papers of interest, a more detailed inclusion or exclusion process was applied, following the criteria outlined in the table below. This selection process was conducted in two separate stages: the first involved screening all of the titles and abstracts of the collected papers to determine their relevance, the second involved reviewing the full text to ensure they met the necessary requirements for inclusion in the review.

Parameter	Value
Years	2019-2024
Language	English
Years	2019-2024
Document Type	Article
Source Type	Journal

Table 2.2: Table of parameters and values.

Thus, by using Scopus and starting from the keywords, 1,617 documents were retrieved. By applying the filters presented in the table above, the number of relevant documents was narrowed down to 511.Considering only those available as open access, the number further reduced to 215.

Finally, by selecting additional keywords related to the topic and present in the titles or abstracts of the documents, 27 results were obtained.

Chapter 3

Theoretical Background

First and foremost, it is essential to provide a solid theoretical background to better understand the concepts and methodologies explored in this thesis. This section aims to introduce the key principles behind the HRTF (Head-Related Transfer Function) which form the foundation for the research and experiments that follow.

3.1 Auditory System

The auditory system elaborates how we hear sounds in the environment. This system is composed of the peripheral and central structures:

- The peripheral auditory structure includes the outer, middle, and inner ear. The outer ear includes the pinna and ear canal, a tube about 2.7 cm long with a resonance frequency around 3 kHz, where our hearing sensitivity is at its peak. The middle ear consists of the malleus, incus, and stapes bones, which have the task of converting the vibrations of the eardrum into amplified pressure waves in the fluid of the cochlea. The middle ear also improves sound transmission by matching the impedance and providing pressure gain through the size difference between the eardrum and the oval window. The inner ear contains the cochlea, which is responsible for the conversion of sound waves into neural signals. Within the cochlea there is the basilar membrane that responds to sound waves, its vibration amplitude varying by frequency [5].
- The central auditory structure includes the cochlear nuclei, superior olivary nuclei, lateral lemniscus, inferior colliculus, medial geniculate nuclei, and auditory cortex [6].

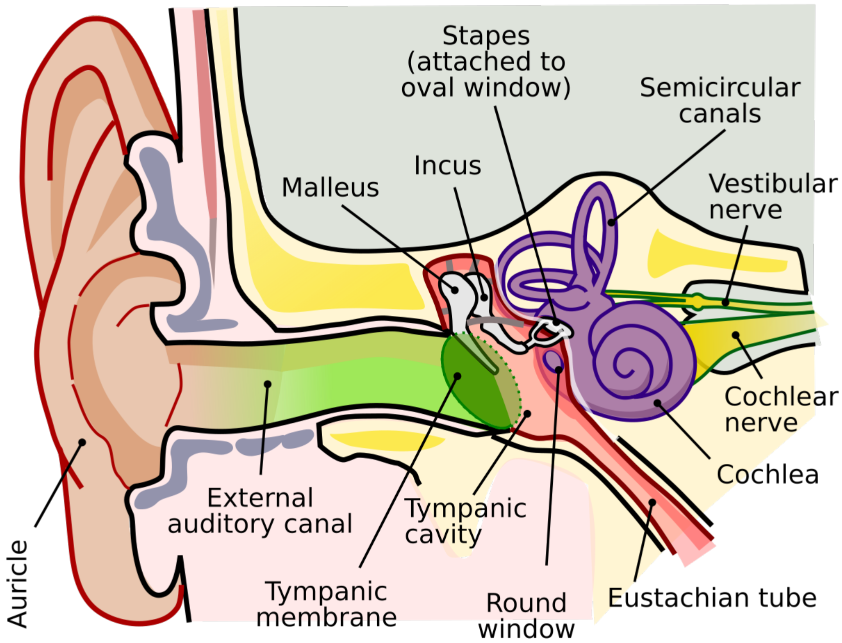


Figure 3.1: Anatomy of the human ear.

3.2 Loudness

The human ears are therefore sensitive to frequencies between 2 and 5 kHz, which is where many of the fundamental frequencies of the human voice and other sounds important for communication are found, in fact the equal-loudness contour shows how the humans perceive the frequencies in relation to the loudness. The curves in the diagram represent the SPL (Sound Pressure Level) in dB required for different frequencies to be perceived as equally loud at a given sound intensity. In other words, they indicate how human hearing perceives volume at different frequencies [7].

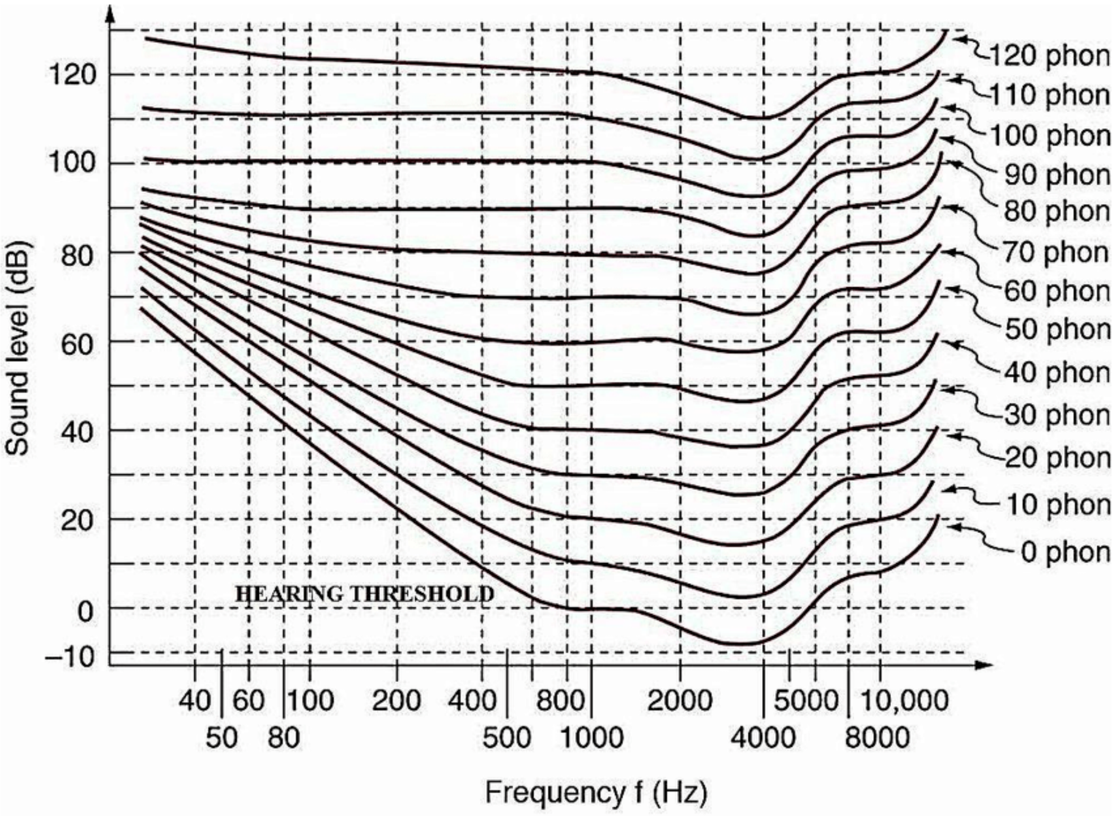


Figure 3.2: Normal equal-loudness-level contours.

The Loudness curves (contour) are expressed in phon, determined by comparison test of pairs of pure tones, and show that at lower frequencies (below about 1 kHz) and higher frequencies (above about 5 kHz), the sound pressure level must be higher for the sound to be perceived as "equally loud" relative to a center frequency such as 3-4 kHz. At the frequency of 1 kHz, by definition, the two scales (dB and phon) coincide [5].

3.3 Sound Localization

The sound localization is our ability to identify the location of a sound, so the distance, elevation and direction of a sound source. Our brain and auditory system use various cues for the localization of a sound source, such as binaural cues and monoaural cues.
Monoaural cues are signals that are detected from a single ear, such as spectral cues.
Binaural cues instead are caused by both ears, for example the Duplex Theory [8]:

- ITD (Interaural Time Difference): the difference in the arrival time of a sound between the two ears. It happens when a sound source is placed near one ear (the ipsilateral ear), causing the sound to reach the eardrum faster than the other ear (the contralateral ear). Its more effective below 1.5 kHz, and various experiments show that ITD relates to the signal frequency f. The ITD for an incident plane wave from azimuth angle θ can be evaluated by [9]:

$$ITD(\theta) = \frac{2a}{c} \sin\theta$$

  More precisely:

$$\text{ITD} = \begin{cases} 3 \times \frac{r}{c} \times \sin\theta, & \text{if } f \leq 4000 \text{ Hz} \\ 2 \times \frac{r}{c} \times \sin\theta, & \text{if } f > 4000 \text{ Hz} \end{cases}$$

  Where the angular position of the acoustic source is θ, the head radius is r, and the acoustic velocity is c.

- ILD (Interaural Level Difference): the difference in the sound pressure level (volume) between the two ears. If a sound is coming from the right it will be louder in the right ear than the left one. The ILD is more effective above 1.5 kHz, in fact is highly frequency-dependent and increase with increasing frequency. It has been demonstrated that the IID relates to the signal frequency f and the angular position of the acoustic source θ. The function of IID is given by [9]:

$$ILD(r,\theta,\phi,f) = 20\log_{10}\left|\frac{P_R(r,\theta,\phi,f)}{P_L(r,\theta,\phi,f)}\right| \ (dB)$$

  where $P_L(r,\theta,\phi,f)$ and $P_R(r,\theta,\phi,f)$ are the frequency-domain sound pressures at the left and right ears, respectively, generated by a sound source at (r,θ,ϕ).

These mechanisms are useful for identifying the sound source in the Azimuth Plane (horizontal plane), but without giving information about the elevation and the distance of the source from the listener. This generates the" Cone of confusion", where all the sounds present the same ILD and ITD, therefore are indistinguishable.

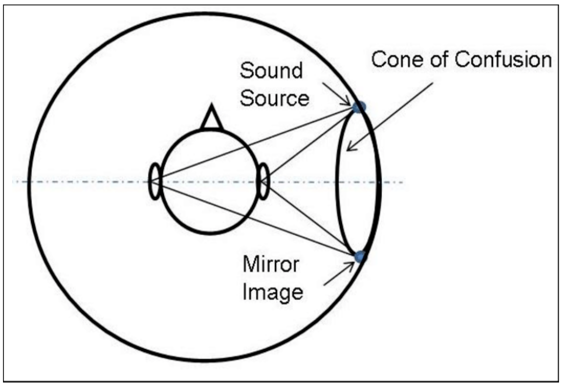


Figure 3.3: Cone of confusion.

In the median plane ILD and ITD are small [5], so humans can distinguish between frontal, up, or back direction due to the monaural cues.

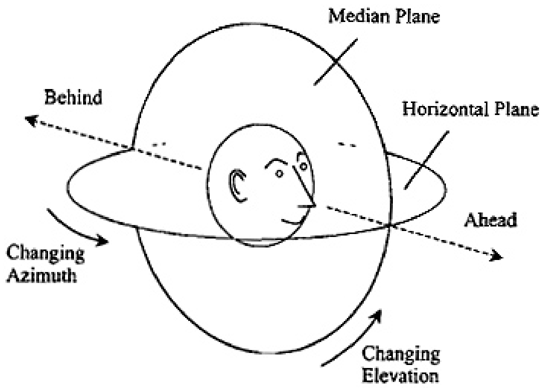


Figure 3.4: The coordinate system for sound direction.

3.4 Overview of Audio Configurations

Nowadays there are lots of audio configurations, each with its own features and advantages that plays an important role for the listener's experience and perception. Here some example of the most common and famous audio configurations:

- Mono Audio: Monophonic audio is the simplest configuration, it uses a single audio channel and the result is an unified sound output. It's commonly used in telephony and voice recordings.

- Stereo Audio: Two audio channels, providing two separate information flows (one for the right channel R and one for the left channel L) that emulate the human ability to recognize spatial provenance.

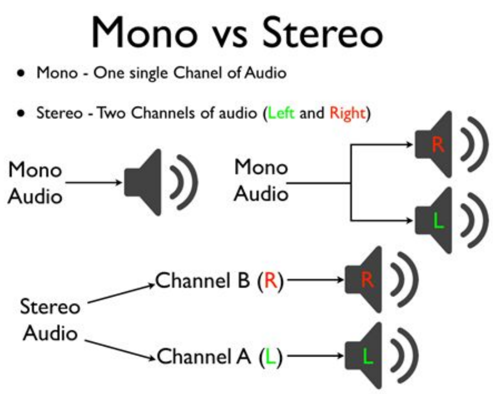


Figure 3.5: Mono vs. Stereo.

- Surroud Sound: Involve multiple audio channels positioned around the listener, creating an immersive audio experience. Common configurations are 5.1 and 7.1 surround sound systems (x.y means x full bandwidth channels and 1 low-frequenxy effect channel). This configurations are highly used to create realistic and immersive audio environment, but their require a complex-setup and non-small space [10].

Channel	Label	Definition
L	Left channel/signal/speaker	Front left position on middle layer
R	Right channel/signal/speaker	Front right position on middle layer
LB	Left Back channel/signal/speaker	Rear left position on middle layer
RB	Right Back channel/signal/speaker	Rear right position on middle layer
C	Centre channel/signal/speaker	Front centre position on middle layer
LFE1	Left Low Frequency Effect channel/signal/speaker	Left side on bottom layer
LS	Left Side channel/signal/speaker	Left position on middle layer
RS	Right Side channel/signal/speaker	Right position on middle layer
LH	Left Height channel/signal/speaker	Front left position on top layer, elevated
RH	Right Height channel/signal/speaker	Front right position on top layer, elevated
CH	Centre Height channel/signal/speaker	Rear centre position on top layer, elevated
LFE2	Right Low Frequency Effect channel/signal/speaker	Right side on bottom layer

Figure 3.6: Channel Definition.

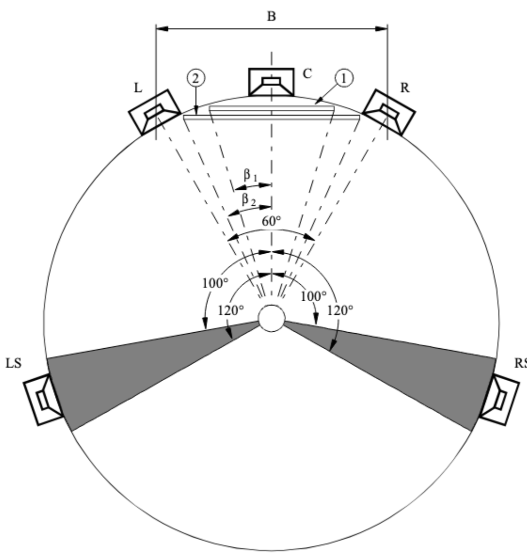


Figure 3.7: 5.1 Layout.

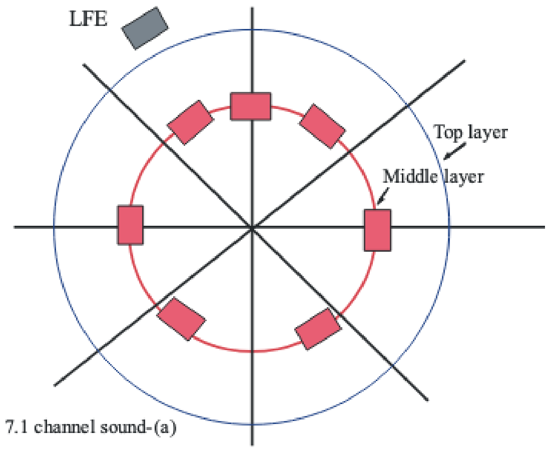


Figure 3.8: 7.1 Layout.

- Object-Based Audio: In an object-oriented system the audio content is created independently of any specific loudspeaker layout. The audio content is represented as audio objects containing the pure audio content, together with metadata describing the position of the audio object along with the properties of the audio object such as directivity in real time [10]. Some examples of Object-Based Audio are: Dolby Atmos, DTS, Apple Spatial Audio, Ambisonics.

Chapter 4

Head-Related Transfer Function

4.1 Definition of HRTF

In the previous chapter, fundamental principles were explored, including the localization cues that are the basis of HRTFs. The head-related transfer function contains both interaural and monaural spectral cues, making it a common tool for synthesizing virtual sound images for headphone playback [11].

It captures all the acoustic transformations before reaching the tympanum [12], so each individual has a unique HRTF shaped by their head, torso, and other anatomical characteristics.

So, a head-related transfer function (HRTF) is an acoustic transfer function that characterizes how sound reaches the ear from a specific point in space.

Having two HRTFs (one per ear) allows for the synthesis of binaural audio, in fact, all of the cues encode the sound location and can be captured by the impulse response (Head-related impulse response, HRIR).

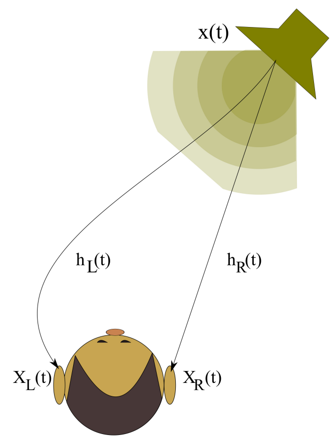


Figure 4.1: Visual concept of HRTF.

As shown in the image below, a Linear Time-Invariant (LTI) system is simply a system that, when given an input signal $s(t)$, generates an output signal $g(t)$ through convolution with the impulse response $h(t)$.
In the context of HRTFs, a generic signal is convolved with the HRTF corresponding to the correct direction to produce the signal that reaches the auditory canal [13].
Therefore, an HRTF can be viewed as an LTI system that defines the relationship between a sound source in free space and a specific position within the listener's ear canal, all under static conditions. Theoretically, any technique used to determine the transfer function of an LTI system can also be applied to HRTF measurements [11].

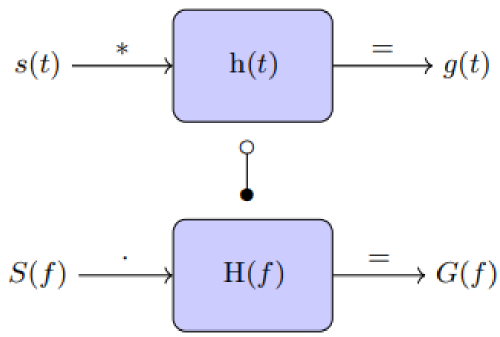
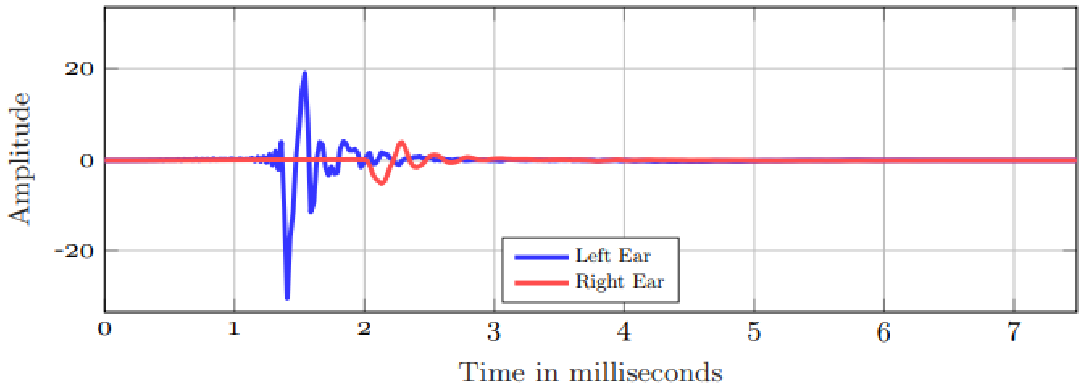
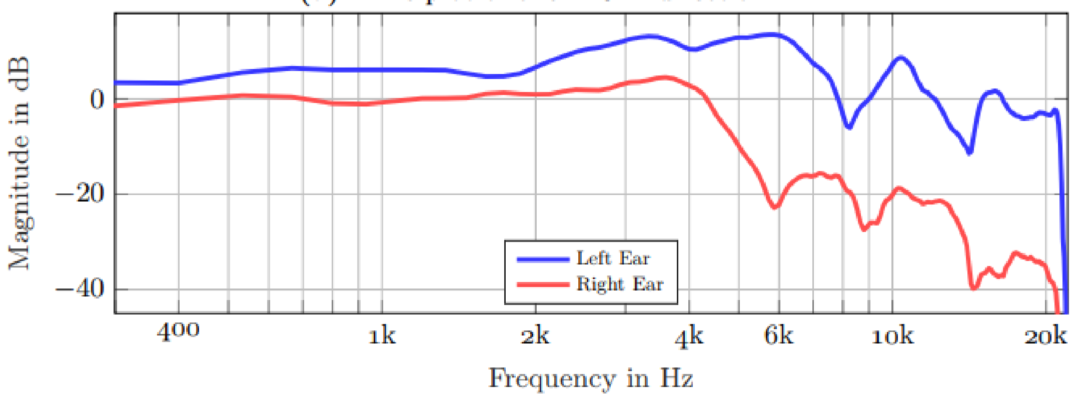


Figure 4.2: Core concept of signal processing in an LTI system [13].



(a) Time plot of one HRTF direction.



(b) Frequency plot of one HRTF direction.

Figure 4.3: Time and Frequency plot of HRTF [13].

To summarize [11]:

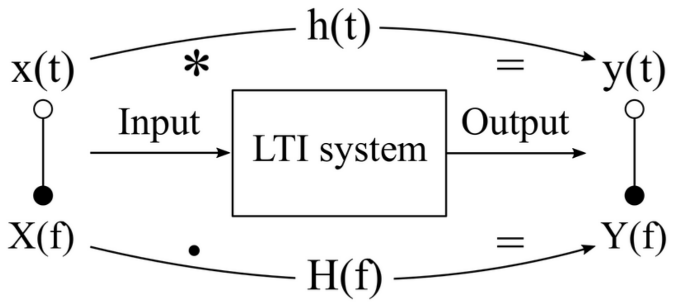


Figure 4.4: Basic principle of signal processing [11].

With regard to an arbitrary source position, a pair of HRTFs, HL and HR, for the left and right ears, respectively, is defined as [9]:

$$H_L = H_L(r, \theta, \phi, f, a) = \frac{P_L(r, \theta, \phi, f, a)}{P_0(r, f)},$$

$$H_R = H_R(r, \theta, \phi, f, a) = \frac{P_R(r, \theta, \phi, f, a)}{P_0(r, f)},$$

where H_L and H_R are the Head-Related Transfer Functions (HRTFs) for the left and right ears, respectively, and P_L and P_R represent the complex-valued sound pressures in the frequency domain at the left and right ears, respectively; P0 represents the free field sound pressure in the frequency domain at the center of the head with the head absent [9].
Therefore, an HRTF is a function of the source position (r, θ, ϕ), frequency f and individual a, which encapsulates the individuality of the listener. The HRTF is the Fourier transform of HRIR.

4.2 Artificial Heads

Before introducing the measurement process of the HRTF it is appropriate to explain what artificial heads are.

Artificial heads are models used to simulate the human head and how sound waves are received and processed by the ear. They are designed to capture the acoustic characteristics of the head, ears, and neck, using in-ear microphones that simulate human hearing. All of this is intended for the reproduction of how sound is perceived by a real listener.

Some of the well-known artificial heads are:

- KEMAR (KEMAR (Knowles Electronics Manikin for Acoustic Research): Mostly used in acoustic research, designed to mimic human anatomy. It is equipped with in-ear microphones to record the acoustic impulse response (HRTF) and study sound localization. It has been designed with median human adult dimensions and the ear simulation matches the acoustic response with an auricle, an ear canal, and an eardrum that equal the median ear in dimensions, acoustic impedance, and modes [14].

Figure 4.5: Artificial heads: KEMAR.

- KU100: Primarily used for binaural recordings. Its shape and structure imitate the human head and allow for capturing detailed information about incoming sound. Due to its natural and transparent sound its typical applications range from classical recordings and radio drama productions to experimental pop and ambient recordings. In addition, the Neumann KU 100 is also often used for industrial applications, e.g. documenting and examining noise in automobiles or work places [15].

Figure 4.6: Artificial heads: KU100.

- HMS II.3 LN HEC: Specifically designed for accurate auditory testing of close-to-the-ear and in-ear transducers, such as those found in ANC headphones, headsets, handsets, and hearing aids. The artificial head is equipped with a low-noise occluded ear simulator, pinnae with a human-like ear canal and a fullband mouth with a two-way loudspeaker design [16].

Figure 4.7: Artificial heads: HMS II.3 LN HEC.

4.3 Measurements of HRTF

The typical measurement procedure of HRTFs is:

- The subject equipped with two miniature microphones, placed inside each the ear canal. The subject is then positioned in an anechoic chamber with a defined distance and orientation to a loudspeaker.
- After the first measurement, the subject is repositioned and a new HRTF is measured. This process is repeated until all required directions are measured [11] [13].

In the image below, a basic and standard setup for HRTF measurement without the repositioning of the subject:

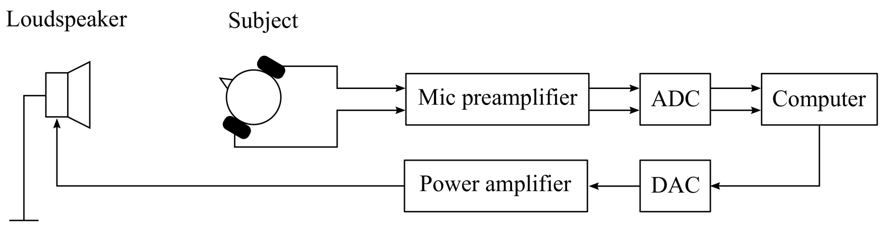


Figure 4.8: Basic setup for HRTF measurements [11].

In principle, any signal was used as an excitation signal for HRTF measurement, it was enough to have a good SNR (Sgnal-Noise Ratio) and to take some precautions with the dynamic range of the devices and the nonlinear behavior of electro-acoustic systems [11]. There are various methods for the measurements of the acoustic impulse responses, but the sweep signal, that is a continuous signal whose frequency continuously changes with time, is the preferred method for the HRTF measurements, beacause of its advantages ,such as the the discrimination of harmonic distortion caused by non-linearities of the measurement system and its robustness against the time-variance effect [17]. There are multiple setup for the measurements, such as Multi-Loudspeaker Setups, Single-Loudspeaker Setups and Multi-Microphone Setups, each one with its own cost, advantages and limitations. For example, a single loudspeaker setup can be useful for fast measurements, with low costs and difficulty, but it is limited in simulating complex sound environments and dynamic scenarios. On the other hand, having a subject in motion, such as rotating, allows for a more realistic simulation of sound perception in movement, but it is more difficult and requires a lot of precision. The most precise and high quality measurement systems involves both a moving subject and multiple speakers, simulating real-world dynamic environments.

The [18] contains a complete measurement setup for the HRTF done by the University of York in their Audio Lab.
The various measurements were taken with high-quality audio equipment and certain rules:

- The microphones used were a pair of Knowles FG-23329-C05 for the humans and, for the artificial head, their built in microphones.
- The configuration featured an acoustically treated HRTF measurement system made up of three fixed vertical semi-circular arcs, each positioned 45° apart in azimuth. Twenty-three Genelec 8010 loudspeakers were arranged at different elevation angles along the arcs, all set at a distance of 1.2 meters.
- The participants were seated on a motorized "saddle stool" chosen for its minimal acoustic occlusion, with their feet tucked under their bodies and supported by a footrest. The inter-aural axis was laser-aligned to the exact center of the loudspeaker array and their head position was tracked in real-time.
- Twenty-four second sweeps were conducted over a frequency range of 200–24 kHz, automated with Max MSP. Measuring the Ambisonic configurations required 64 stops, while a 1° resolution for dummy subjects led to 399 stops and 8,802 unique measurements, taking over 3 hours. Due to discomfort, 11 subjects were measured at a 5° resolution (127 stops) and 18 subjects at a 10° resolution (95 stops), resulting in 2,818 and 2,114 unique measurements, respectively.

.

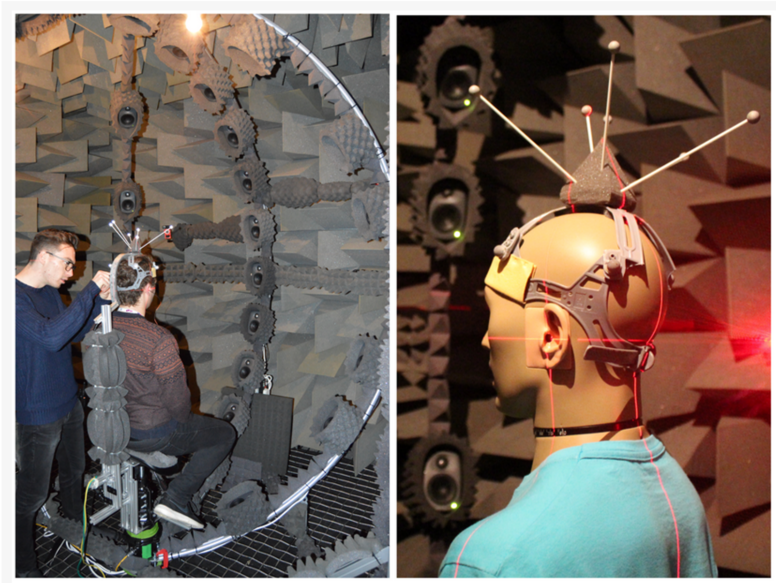


Figure 4.9: On the left a subject being prepared for HRTF measurements, on the right an example of the alignment of a subject and the centre of the loudspeaker array [18].

In addition to all this, various problems have also been taken into account and resolved, including low-frequency compensation caused by the limit of the anechoic chamber and the size of the loudspeakers' diaphragms, and headphone equalization for achieving a flat frequency response for the headphones (HpEQ) using Headphone Impulse Responses (HpIRs).

4.4 Unconventional Approaches for Determining HRTFs

Nowadays there are various methods to obtain HRTFs without needing expensive equipment, though at the expense of a slightly lower quality of personalization compared to traditional methods.
The use of 3D modeling of the head and ears through digital scans is one of the most promising methods, because it has excellent results in relation to its simplicity and cost. Thanks to the spread of user-friendly 3D scanners and advanced software, it is possible to accurately reconstruct an individual's morphology and the data obtained from these scans can be used to generate synthetic HRTFs through acoustic simulations based on physical models. The most famous method is Mesh2HRTF, an open-source project aiming at providing an easy-to-use software package for the numerical calculation of head-related transfer functions (HRTFs) [19].

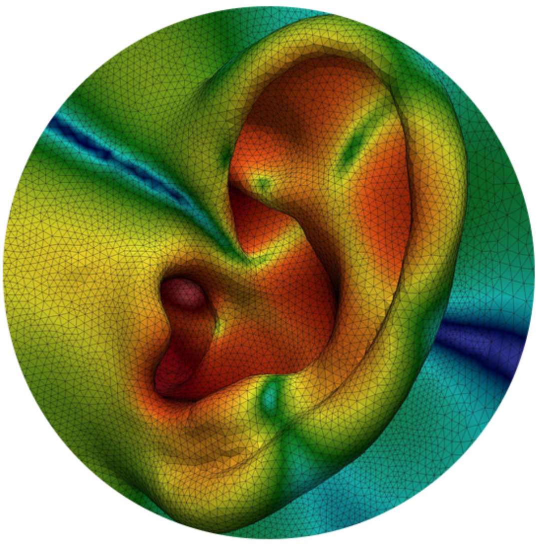


Figure 4.10: 3D mesh of human ear.

Another innovative approach is the use of machine learning techniques combined with anthropometric parameters to generate personalized HRTFs without requiring direct acoustic measurements. For instance, [20] proposed the Equivalent Acoustic Center (EAC) methodology which uses statistical models and large HRTF databases to predict an individual's response with reasonable accuracy, by using user-specific data, such as head and ear geometry, in order to synthesize HRTFs tailored to the listener. Although still in development, this approach holds the potential to revolutionize the calculation and personalization of HRTFs in the future [21].

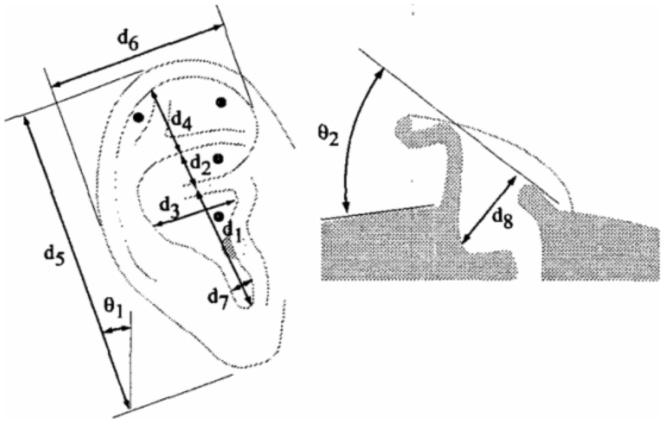


Figure 4.11: Pinna measurement.

Finally, there are also some hybrid solutions that combine simple measurements,like recording a few angular positions, with algorithms that interpolate the missing data. This techniques significantly reduce the number of measurements required, lowering costs and the time needed, while still maintaining good sound quality.
In conclusion, these methods offer a valid and accessible alternative for those without specialized equipment or for applications that do not require the highest level of detail.

4.5 SOFA Files: Characteristics and Applications

The Spatially Oriented Format for Acoustics (SOFA) is a standardized file format designed to store and exchange spatial audio data, particularly for Head-Related Transfer Functions (HRTFs) and Impulse Response (IR) measurements. Developed to facilitate the sharing of spatial audio data among researchers and audio engineers, SOFA files provide a structured way to represent complex audio characteristics [22].
The characteristics of SOFA Files are:

- Structured Data Storage: SOFA files use a hierarchical structure based on the HDF5 format, which allows for efficient organization and retrieval of multi-dimensional data.
- SOFA files can support multi-channel audio data, used for spatial audio representation, enhancing the realism of auditory experiences.
- Metadata Inclusion: they can include extensive metadata, such as information about the measurement environment, equipment used, and specific parameters of the audio data.
- Interoperability: As a standardized format, SOFA promotes interoperability between various software tools and platforms used in spatial audio research and production.

The applications of SOFA Files is various and change from application to application. They are essential for storing and analyzing HRTF measurements, contributing to research on spatial sound perception and binaural rendering, they also play a key role in spatial audio rendering for immersive environments like virtual reality, augmented reality, and gaming, where precise 3D sound positioning is critical. In room acoustics, SOFA files store impulse responses, helping to study and improve the sound characteristics of different spaces and they are also valuable for educational and research purposes, offering standardized and accessible spatial audio data for experiments and analysis.

Chapter 5

Case Study: Impulse Response Acquisition and Binaural Rendering

This chapter is based on the experience gained during a university internship at RAI CRITS (Research and Technological Innovation Centre) in Turin. It presents a case study on the acquisition and processing of Impulse Responses (IRs) for binaural rendering, as well as a related subjective test.

Figure 5.1: Studio ST6, RAI.

5.1 Measurements Setup

The HRTF measurements were conducted in the ST6 studio, located at Via Verdi in the RAI headquarters, which has a 7.1.4 configuration
The setup begins with the Digital Audio Workstation (DAW), where Reaper was used for the recording and processing of audio.

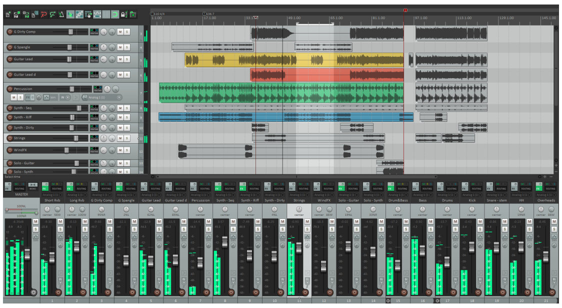


Figure 5.2: Cockos Reaper. Digital Audio Workstation.

The audio interface used is the MADIFACE XT, connected via USB with a DPA CORE 4560 binaural microphone connected to the audio interface via XLR.

(a) MADIFACE XT (b) DPA CORE 4560

Figure 5.3: On the right, the DPA CORE 4560 binaural microphone, and on the left, the MADIFACE XT

The signal then passes through MADI (Multichannel Audio Digital Interface) to the DIRECTOUT MADI M1K2 16x16 matrix, which allows for efficient routing of the audio channels. Following this, the signal is converted to AES format by the Directout Andiamo.aes MADI-AES converter. An RTW Loudness Meter is connected to the AES output for real-time loudness monitoring.
Finally, the AES signal is sent to the TRINNOV D-MON 12, an audio monitoring system, which allows for precise control and monitoring of the audio signals. The loudspeaker setup consists of Genelec 8250A speakers for standard positions and

Genelec 8240A speakers for elevated positions, providing a comprehensive 7.1.4 surround sound configuration.

Figure 5.4: Trinnov D-MON 12.

(a) Genelec 8250a

(b) Genelec 8240a

Figure 5.5: Genelec 8240a and 8250a.

Here below is the speaker arrangement, visualized through the TRINNOV D-MON 12, along with the respective phase, volume and delay adjustments.

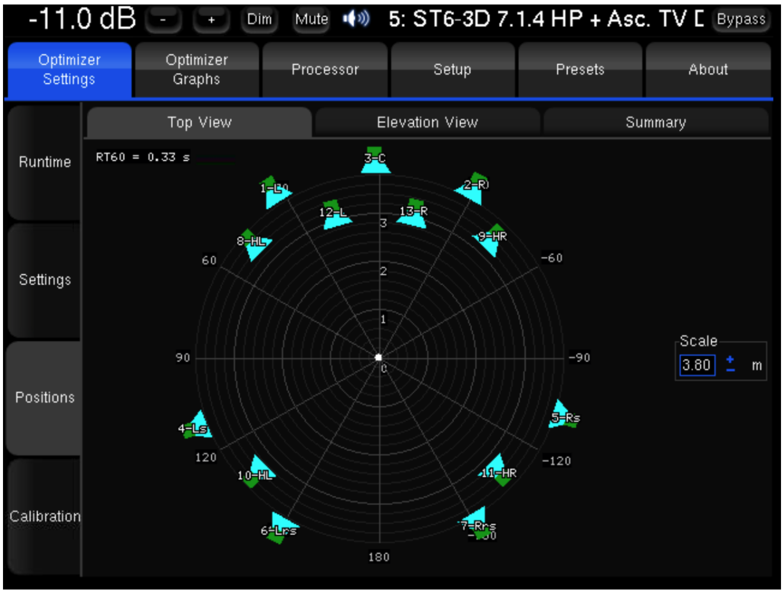


Figure 5.6: Speaker Layout ST6.

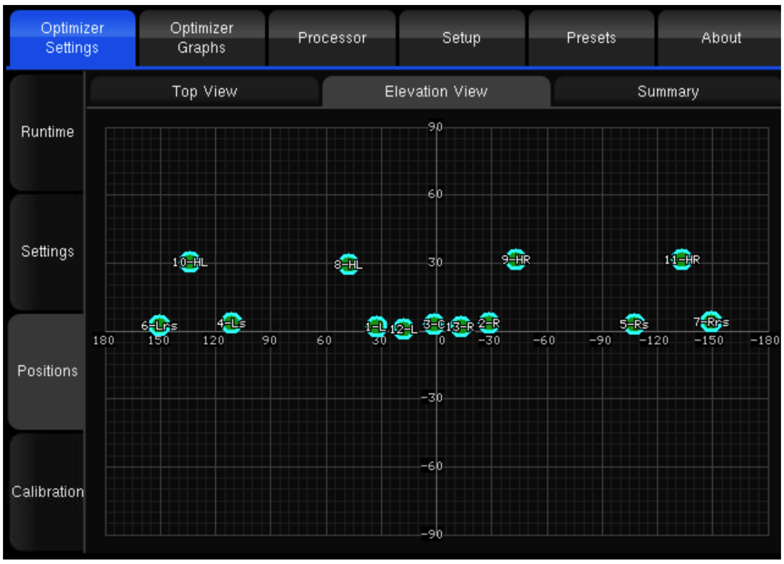


Figure 5.7: Speaker Layout ST6.

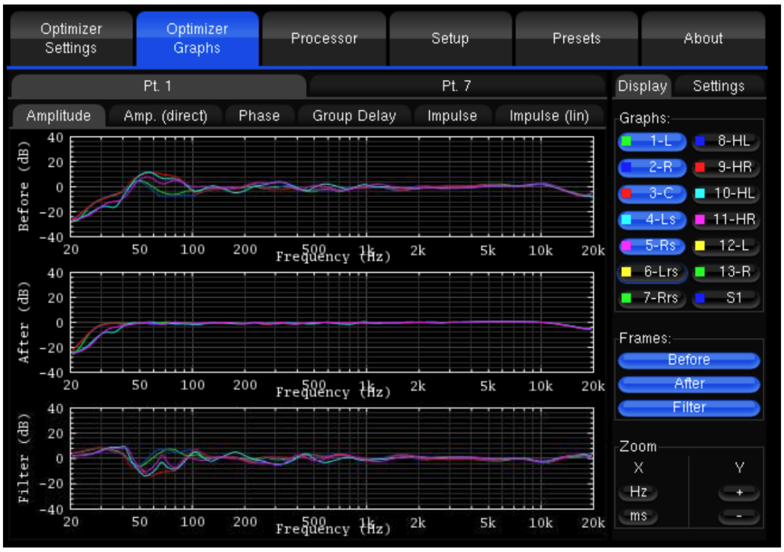


Figure 5.8: Amplitude Adjustment.

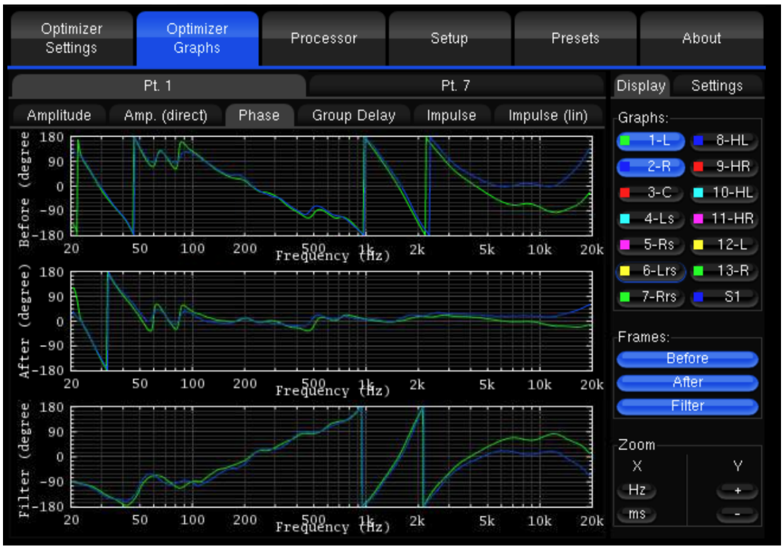


Figure 5.9: Phase Adjustment.

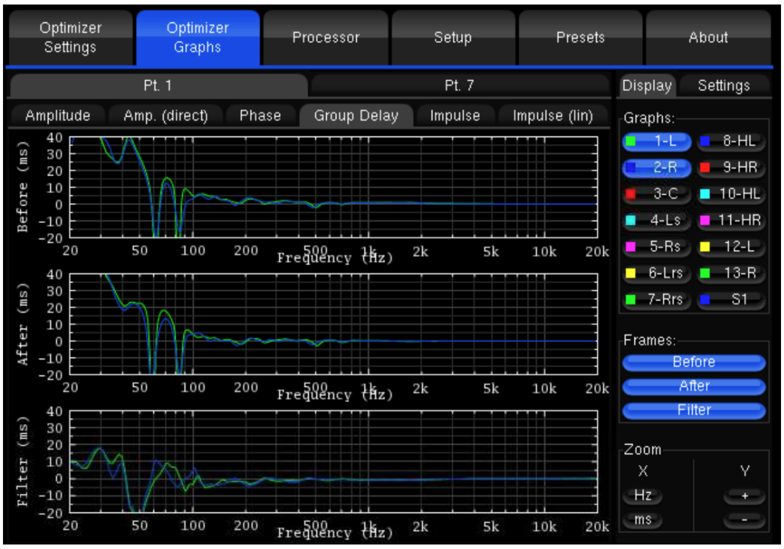


Figure 5.10: Delay Adjustment.

5.2 Impulse Response acquisition and processing

This section describes the entire workflow for acquiring, processing, and compensating impulse responses (IRs) used in binaural rendering for our case study.
The workflow includes capturing IRs in a 7.1.4 studio setup with basic static measurements, followed by a detailed signal processing pipeline that involves convolution, low-frequency compensation, and the concatenation of IRs into a single file, ready for binaural rendering.
As previously mentioned in the earlier section, the DPA CORE 4560 binaural microphone was used to capture the spatial audio. This microphone provides an accurate recording of the sound field generated by the speaker setup. The signal used for recording was a 2-second linear sweep signal, followed by 1 second of silence, spanning the frequency range from 0 Hz to approximately 20 kHz, generated by using the Aurora Plugin by Angelo Farina. The sweep signal ensures a detailed representation of the entire audible frequency spectrum, allowing for the precise capture of the acoustics of the room and the speaker configuration. The binaural microphone was positioned at the optimal listening location, the "sweet spot", within the room, where it could best replicate the sound perception of a listener seated in the center of the setup. This spot is 2.36–2.38 meters away from the speakers and has a height of approximately 1.23 meters from the floor to the KU100 dummy head. The candidates and the KU100 remain fixed in their position, as do the speakers. This is therefore a static measurement, which does not involve any movements or changes in the angles of the speakers or the candidates.
After the initial recording, the next step involved was processing the recorded sweep signals for the extraction of the impulse responses. To achieve this, the Aurora Plugin developed by Angelo Farina was used to generate the inverse sweep simultaneously with the creation of the initial sweep signal. The inverse sweep is essential for the convolution process; in fact, the raw IRs are generated using the X-Volver plugin, a tool developed by Farina that convolves the recorded sweeps with the matrix of the inverse sweep, a stereo file with the inverse sweep first on the left and after on the right channel, resulting in a wav file containing the IRs. This process provides the raw IRs needed for further optimization.

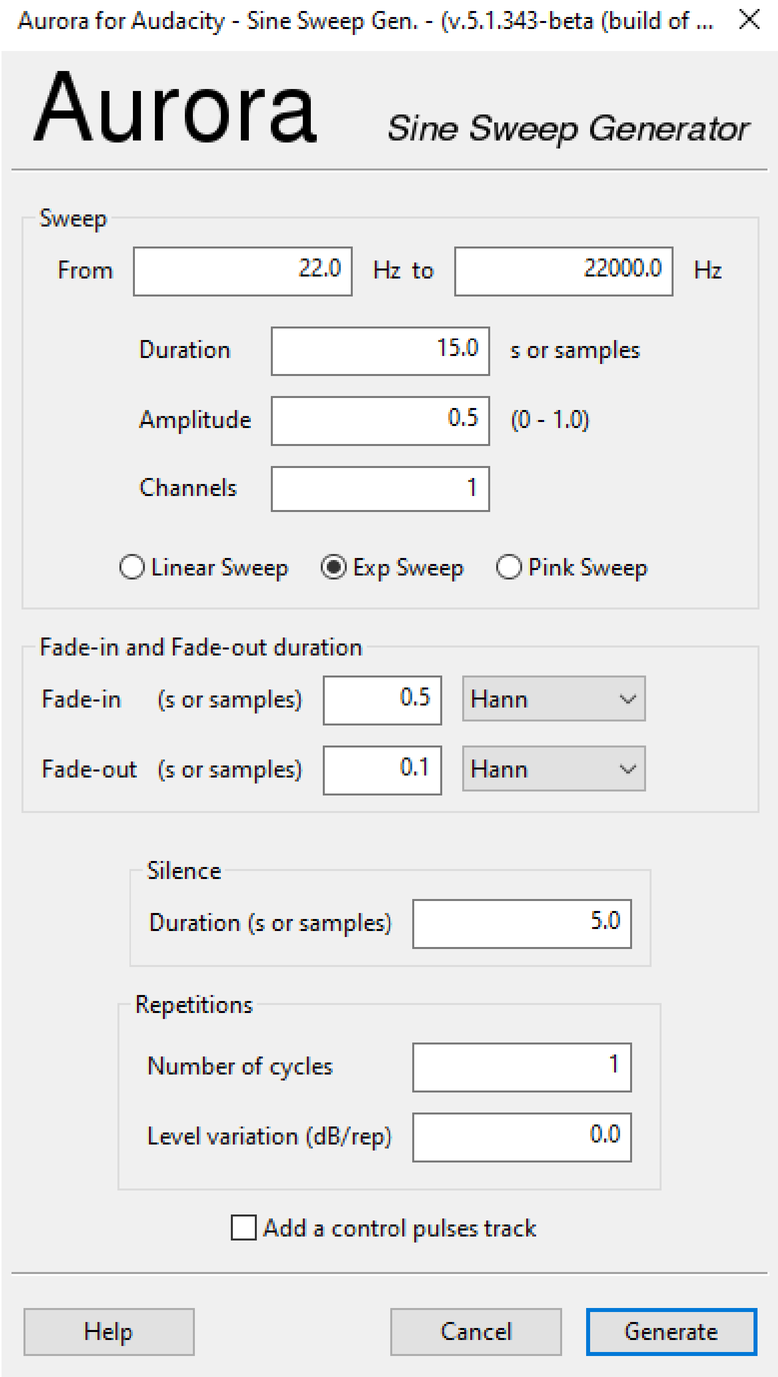


Figure 5.11: Sine Sweep Generator, Aurora Plugin.

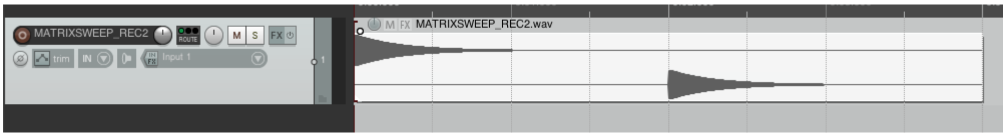


Figure 5.12: Matrix of inverse sweep.

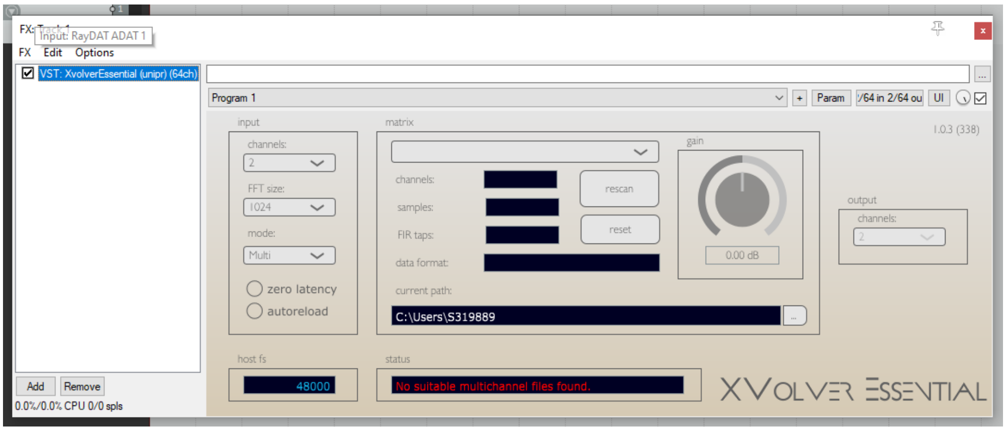


Figure 5.13: X-Volver.

Once the IRs were extracted using X-Volver, the next step was to process them further to compensate the low-frequency inaccuracies. Low-frequency compensation is particularly important because the recording system and room acoustics may introduce losses or coloration at lower frequencies, which can affect perceived spatial accuracy when used in binaural rendering. To address this issue, a custom Python script was developed. This script takes the raw IR files generated by X-Volver as input and applies a low-frequency compensation algorithm. The goal of compensation is to correct for any losses or deviations in the low-frequency response, ensuring that the IRs are more accurate and ready for high-quality binaural reproduction.

The Python script B is designed to perform a series of operations on audio signals, focusing specifically on handling impulse responses (IR). Below is an overview of the main operations carried out:

- Input: The script requires the wav file conteining the IRs, an integer specifying the number of samples for each segment (e.g., 256), the Dirac delta convolved with a low-pass FIR filter at 275 Hz, a high-pass FIR filter at 275 Hz and a sine wave at 275 Hz (ensuring it starts and ends with a slight fade to avoid distortion).

- Main functions:

  - `load_audio()`: Loads an audio file (including stereo files, converting to mono if required).
  - `cut_wav_numpy()`: Cuts an audio segment starting from a given sample and for a defined number of samples.
  - `process_wav_numpy()`: Extracts IR segments from the input file according to pre-defined settings, saves each extracted IR as a separate WAV file and concatenates all extracted IRs into a single output file.
  - `split_stereo_to_mono()`: Splits a stereo file into two mono files (left and right channels), saved in PCM 24-bit format.
  - `envelope_audio()`:Applies an envelope (fade-in and fade-out) to the first and last 10 samples of an audio file.
  - `convolve_audio()`: Convolves an input file with an FIR file, saving the result in 48 kHz/24-bit format.
  - `phase_correlation()`: Calculates the delay between two audio signals using cross-correlation and provides adjustments to balance RMS levels.
  - `apply_delay_and_volume()`: Applies a delay (time shift) and attenuation in dB to a mono audio file.
  - `sum_and_cut()`: (partially visible) Appears to sum and cut processed IRs to create the final output files.

- Process Flow: The main input file containing the IRs is loaded. Each IR is extracted, cut, and processed individually. IRs are convolved with the specified FIR filters and after with the Sine wave, that helps to calculate the phase correlation and delays with the Delta Dirac (also convolved with FIR filter and Sine wave.). After phase correction, delays, and volume adjustments, the processed IRs are saved individually. Finally, all processed IRs are concatenated into a single output file.

- Output: Separate mono or stereo WAV files for each extracted IR. A concatenated WAV file containing all processed IRs.
- Additional Features: It guarantees a 48 kHz sampling rate and generates informative log messages for each function.

Following the compensation process, the optimized IRs were concatenated into a single file. This step is crucial for organizing the IRs in a format that is suitable for binaural rendering software. The IRs were arranged according to the speaker layout of the 7.1.4 system, ensuring that each channel of the audio setup is correctly represented. The resulting file includes the following channels:

- LEFT, RIGHT, CENTER, LFE (Low Frequency Effect), LEFT SURROUND, RIGHT SURROUND, BACK LEFT, BACK RIGHT, UP LEFT, UP RIGHT, UP BACK LEFT, UP BACK RIGHT.

Each of these channels corresponds to a specific position in the 7.1.4 configuration in the ST6 studio, as shown above, guaranteeing that the spatial characteristics of the original setup are accurately preserved in the final IR file.
The concatenated IR file can now be used in convolution-based binaural rendering systems, such as the MCFX Convolver, also developed by Angelo Farina.

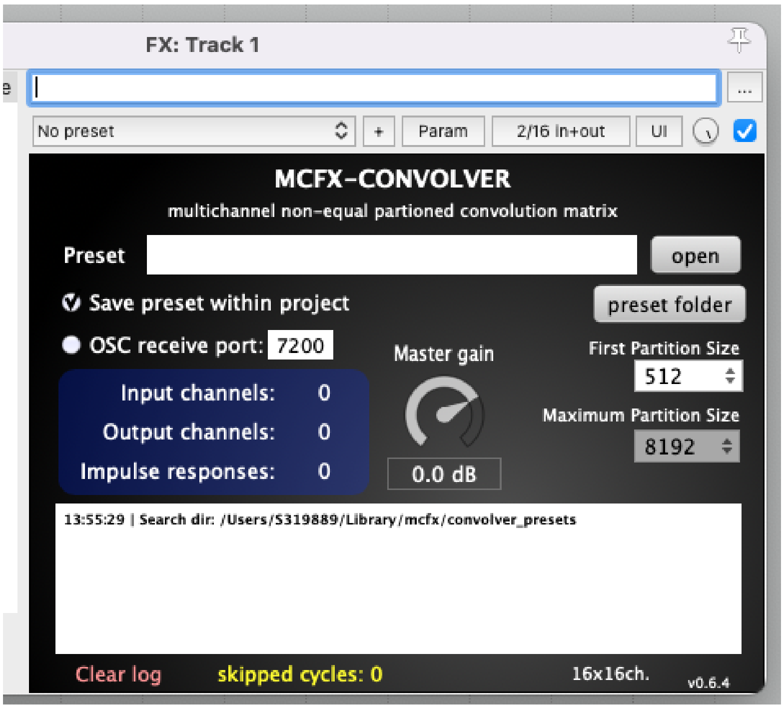


Figure 5.14: MCFX-Convolver, 16 Channel.

After executing the script and completing all the previous processes, this convolver allows for the binaural rendering of a multichannel audio source by applying the compensated, concatenated impulse responses. Furthermore, by utilizing an additional MATLAB script and the individually compensated stereo IRs (that are generated with the same script above), it is possible to create a SOFA (Spatially Oriented Format for Acoustics) file, which can be used for further analysis and applications in 3D audio spatialization and HRTF studies.
Below is a representation of the frequency spectrum of a white noise signal, convolved using the MCFX-Convolver 16x16, with both uncompensated and compensated IRs.

Figure 5.15: Raw IRs.

Figure 5.16: Compensated IRs.

5.3 Test Setup

It is now necessary to evaluate the perceived differences in listening to various HRTFs. Therefore, a subjective test was designed based on the ITU-1284 [2], ITU-2132 [1], and SAQI [23] standards to be administered to sound engineers and audio technicians at RAI Torino. This approach aims to understand the subjective differences in the selected parameters and identify preferences among the chosen HRTFs and their respective implications. The goal is to evaluate which rendering method most closely approximates the listening experience with the 7.1.4 speaker setup, additionally, the test also aims to determine which binaural rendering is preferred by professionals and, consequently, to analyze the various differences between the selected HRTFs.
The audio files used in the test were selected from various musical compositions and non-musical tracks provided by RAI Milan and RAI Turin. This choice allowed for the inclusion of a variety of sound content with different characteristics and different spatial behaviour, enabling a more comprehensive evaluation of the performance of the different HRTFs.

5.3.1 Selected HRTF

The HRTFs selected for the test are as follows:

- KU100 FF: A full sphere far field HRTF of a Neumann KU100 from Sofa Convention [24] [22].
- KU100 NF: A near field HRTF of a Neumann KU100 from Sofa Convention [25] [22].
- KU100 ST6 128samples: KU100 HRTF processed with a 128-sample window, calculated in the ST6 Studio.
- KU100 ST6 128 samples + HpEQ(HD800@Harmann): KU100 HRTF processed with a 128 sample window and Headphone Equalization, calculated in the ST6 Studio.
- KU100 ST6 256 samples + HpEQ(HD800@Harmann): KU100 HRTF processed with a 256 sample window and Headphone Equalization, calculated in the ST6 Studio.
- Kemar 128 samples AAchen + HpEQ(HD800@Flat): An HRTF of a KEMAR from Sofa Convention and Headphone Equalization [26] [22].
- pHRTF 128 samples: Personalized HRTF processed with a 128-sample window, calculated in the ST6 Studio.

- pHRTF 128 samples + HpEQ(HD800@Flat): Personalized HRTF processed with a 128 sample window and Headphone Equalization, calculated in the ST6 Studio.
- pHRTF 256 samples + HpEQ(HD800@Flat): Personalized HRTF processed with a 256 sample window and Headphone Equalization, calculated in the ST6 Studio.

The selection has different HRTFs, including standardized artificial head models (KU100, KEMAR) and personalized HRTFs, both with Correction EQ. From this selection, it is possible to understand which option was most preferred between the 128-sample and 256-sample processing, between the KEMAR, KU100, and personalized HRTFs, and whether the headphone equalization was favored.
It is important to specify that for the HRTFs taken from literature, a Python script B was used to extract the corresponding IRs from the SOFA file in the positions relevant to us, namely those of our 7.1.4 speaker setup. This was done in order to ensure that the final convolution process, and therefore the binaural rendering, is the same across all HRTFs. Moreover, since we do not have head tracking, there is no need for a higher number of points.

5.3.2 Headphone Equalization

As previously mentioned, among the various HRTF options, there are some headphone equalizations aimed at ensuring consistency between the listening of different HRTFs and between headphone and speaker listening. The headphones used for the tests are Sennheiser HD800, while the aimed equalizations are two, created using AutoEQ [27], a tool for automatically equalizing headphones:

- Harmann Curve: The Harman target curve is an ideal frequency response for headphones developed through studies conducted by Harman International.
- Flat: A flat frequency response.

Here below are the frequency responses of the Sennheiser HD800 and the equalizations of the headphones:

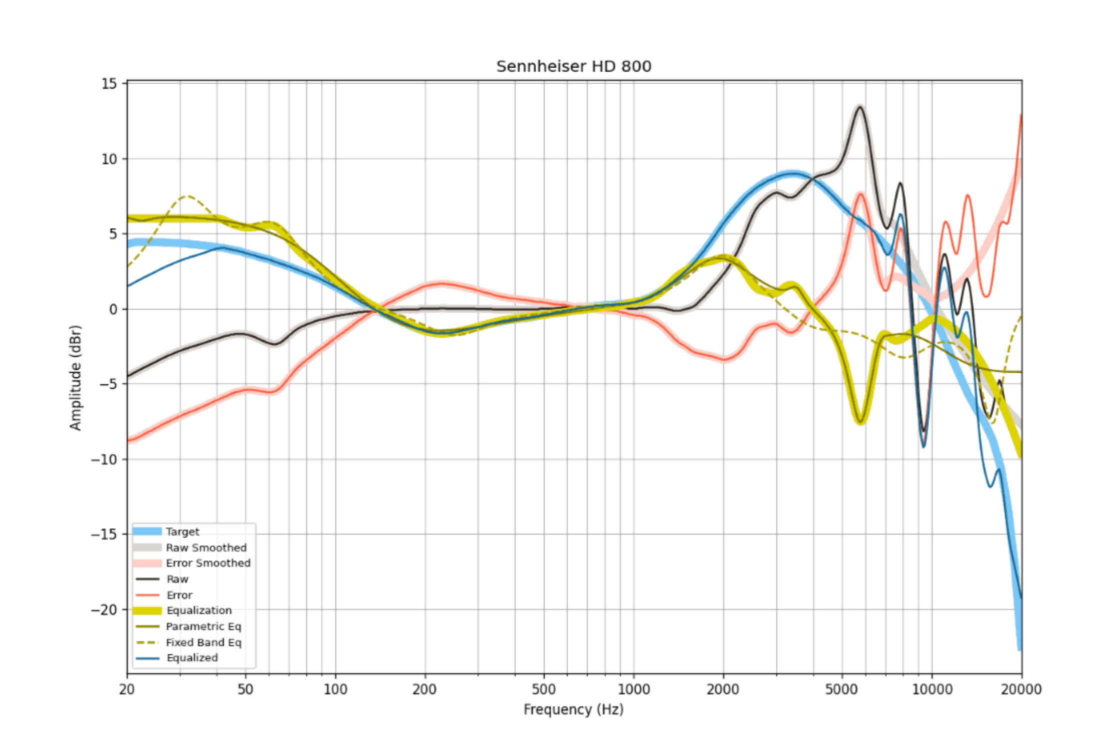


Figure 5.17: Frequency response of HD800.

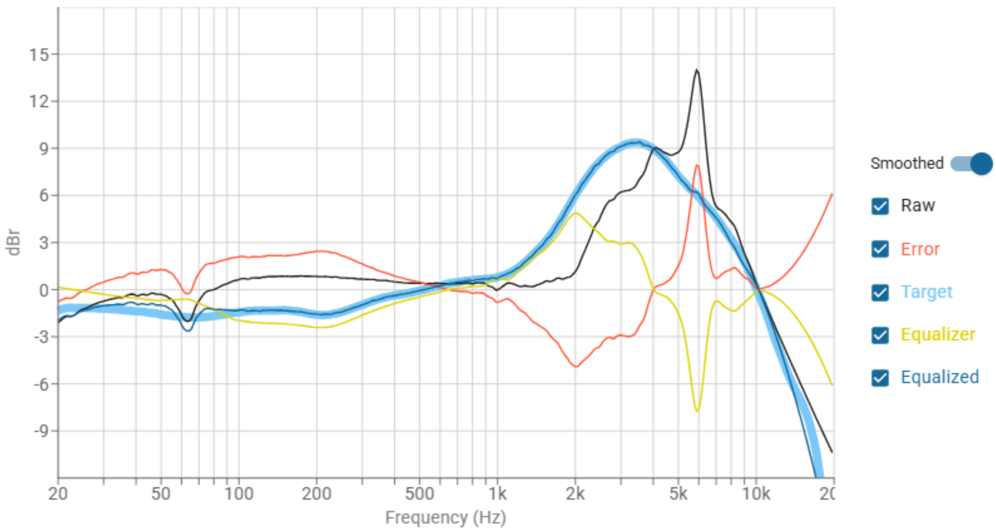


Figure 5.18: Harman over-ear 2018 without bass Equalization.

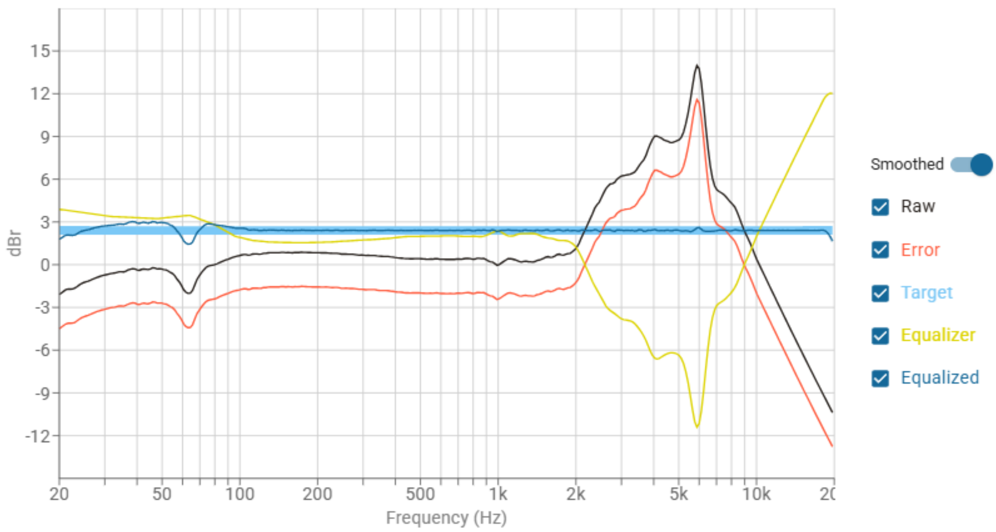


Figure 5.19: Flat Equalization.

These responses were used to achieve the most similar listening experience possible between the KU100, KEMAR, personalized HRTFs and beetween loudpeaker and headphone. The KU100 has a flat frequency response and therefore requires a diffuse field equalization, which includes a slight boost in the high frequencies. On the other hand the personalized HRTFs and the KEMAR already have a similar type of Harmann frequency response, so a flat equalization is used. This type of reasoning arises from the fact that the target frequency response for studio headphones, according to [28] is defined as follows (Sennheiser HD800 respects this target):

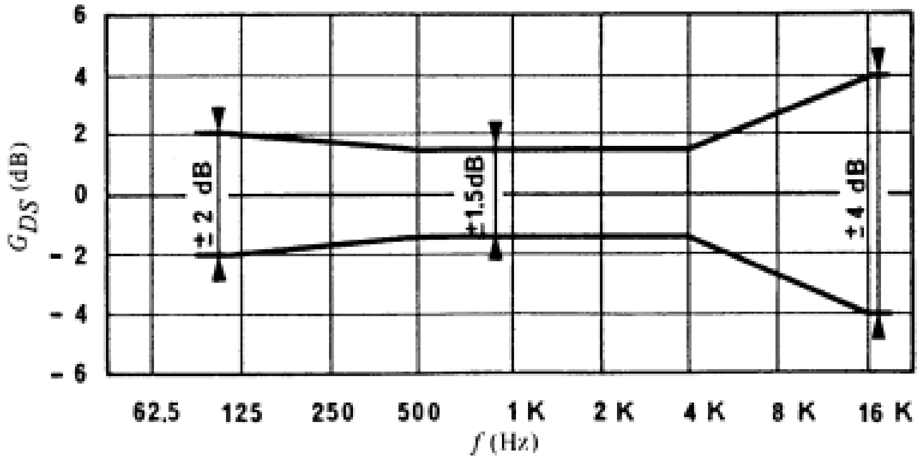


Figure 5.20: The tolerance mask for the diffuse-field frequency response .

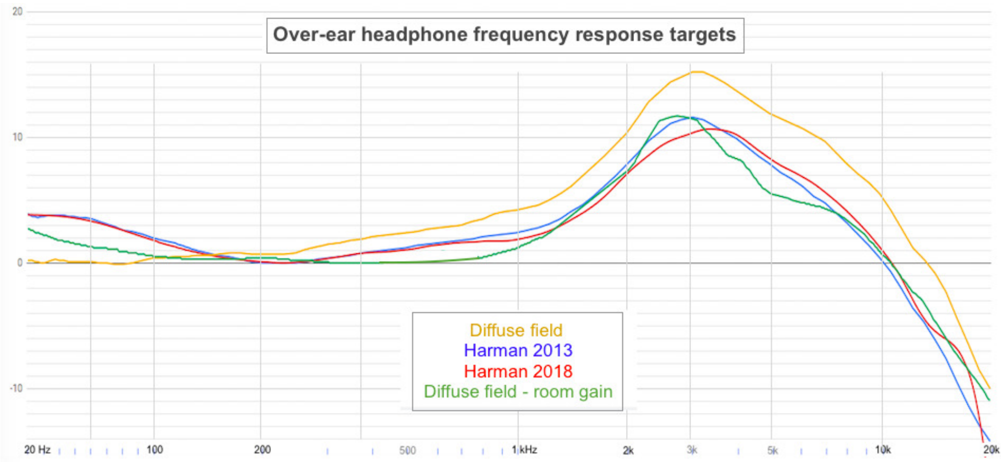


Figure 5.21: Diffuse Field Curve.

5.3.3 Loudness Normalization

Volume normalization ensures that the source clips are adjusted to a level of –23 LUFS, while the test clips are normalized to the track level using a convolver on pink noise, with a target value of –23 LUFS $\pm$ 0.2. Subsequently, to achieve the same loudness between headphones and speakers during the test, the appropriate tools available in the ST6 studio are used, such as the headphone amplifier and the speaker loudness meter. The loudness measurements of the speakers and headphones are madde using the KU100.

5.3.4 Evaluation Attributes

The attributes used in the test are based on the ITU-1284 [2], ITU-2132 [1], and SAQI [23] standards and aim to assess the various subjective differences in the key aspects of HRTFs. Each attribute will be evaluated not only for a single audio file but across multiple files to analyze the variations and changes in results depending on the audio content and its characteristics. For example, this approach allows for the assessment of how perception and outcomes for the same attribute may differ when evaluated with a musical excerpt compared to a film clip.

Attribute	**Description**	**Lower Label**	**Upper Label**
Sound Color [2]	The subjective impression of an appropriate sound for each source including all its characteristic harmonic elements.	Bad	Excellent
Overall subjective quality [1]	The attribute basic audio quality includes all aspects of the sound quality being assessed.	Bad	Excellent
Scene depth [1]	The perception of the depth of the sound image. Takes into consideration both overall depth of scene, and the relative distance of the individual sound sources.	Flat	Deep
Localisation accuracy [1]	How well are the individual instruments and voices placed and separated in the spatial sound image? How precise are the individual sound sources positioned in the room?	Inaccurate	Accurate
Geometry: Externalization [23]	If localizability is low, spatial extent and location of a sound source are difficult to estimate, or appear diffuse. If localizability is high, a sound source is clearly delimited. Low/high localizability is often associated with high/low perceived extent of a sound source. Examples: sound sources in highly diffuse sound field are poorly localizable.	More Difficult	Easier

Table 5.1: Attributes for the test.

5.3.5 webMUSHRA

Once the attributes and audio files to be listened to have been selected, the next step is to calculate their binaural renders by following the previously described procedure. This involves opening the audio track in Reaper and using the 16-channel MCFX-Convolver, feeding it the concatenated and compensated IR file for each sound engineer, previously calculated using the provided script, and finally exporting each result. After completing this process for all audio files and HRTFs, the actual test can be created using webMUSHRA [3].
webMUSHRA is a MUltiple Stimuli with Hidden Reference and Anchor (MUSHRA) compliant web audio API based experiment software.
Listening tests are a common method for evaluating the quality of audio systems. In recent years, web-based experiments have become increasingly popular due to the growing accessibility of online testing platforms. Previously, limitations in web standards, such as the inability to manipulate audio streams, restricted the types of listening tests that could be conducted online, however, with the introduction of the Web Audio API, it is now possible to perform MUSHRA tests directly in web browsers while adhering to ITU-R Recommendation BS.1534 [29] (MUSHRA) [3].
The test starts with a verbal explanation of the various attributes and testing methods, which is a continuous quality scale (CQS) [30] that consists of a 100-point linear scale divided into five equal intervals with five regional verbal anchors used in the five-grade scales.

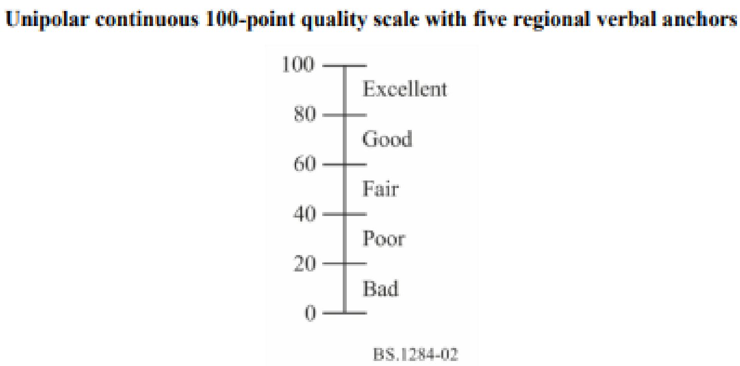


Figure 5.22: Unipolar continuous 100-point quality scale with five regional verbal anchors [2].

Upon starting the actual test, there will be the option to adjust the volume as desired. Then, the evaluation of the various attributes will begin, with the possibility to listen to the reference at any time, move forward and backward through the audio files as needed, listen to a single part on loop, and also navigate freely between the different pages.

To play the reference through the speakers and the binaural renders through the headphones, a particular method was used. Specifically, each file consists of 16 channels: for the binaural render files, the first 14 channels are empty, and the last two contain the stereo audio files of the binaural render, instead, the reference file includes the first 12 channels, while the remaining 4 channels are empty, allowing the first 12 channels to be played through the speakers of the 7.1.4 setup and the last two through the headphone output. This setup was made possible thanks to the equipment available in the ST6 studio. With this method it is possible to switch from the reference to the render without having problem with the headphone or loudpeaker setup.

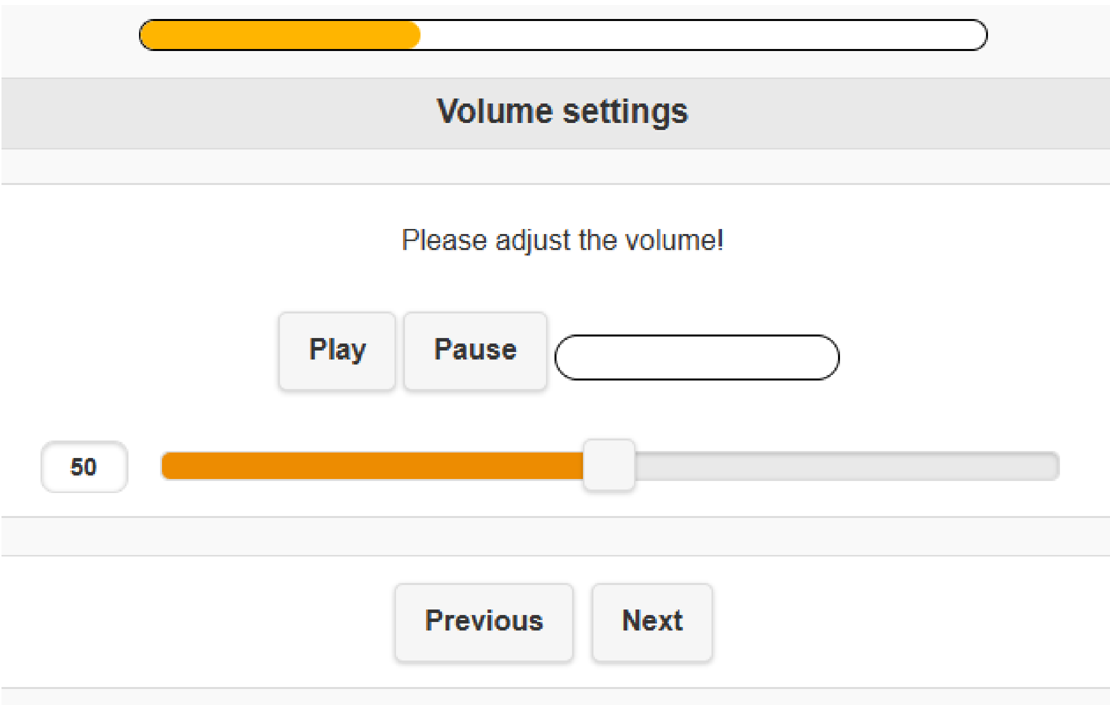


Figure 5.23: First page of the test.

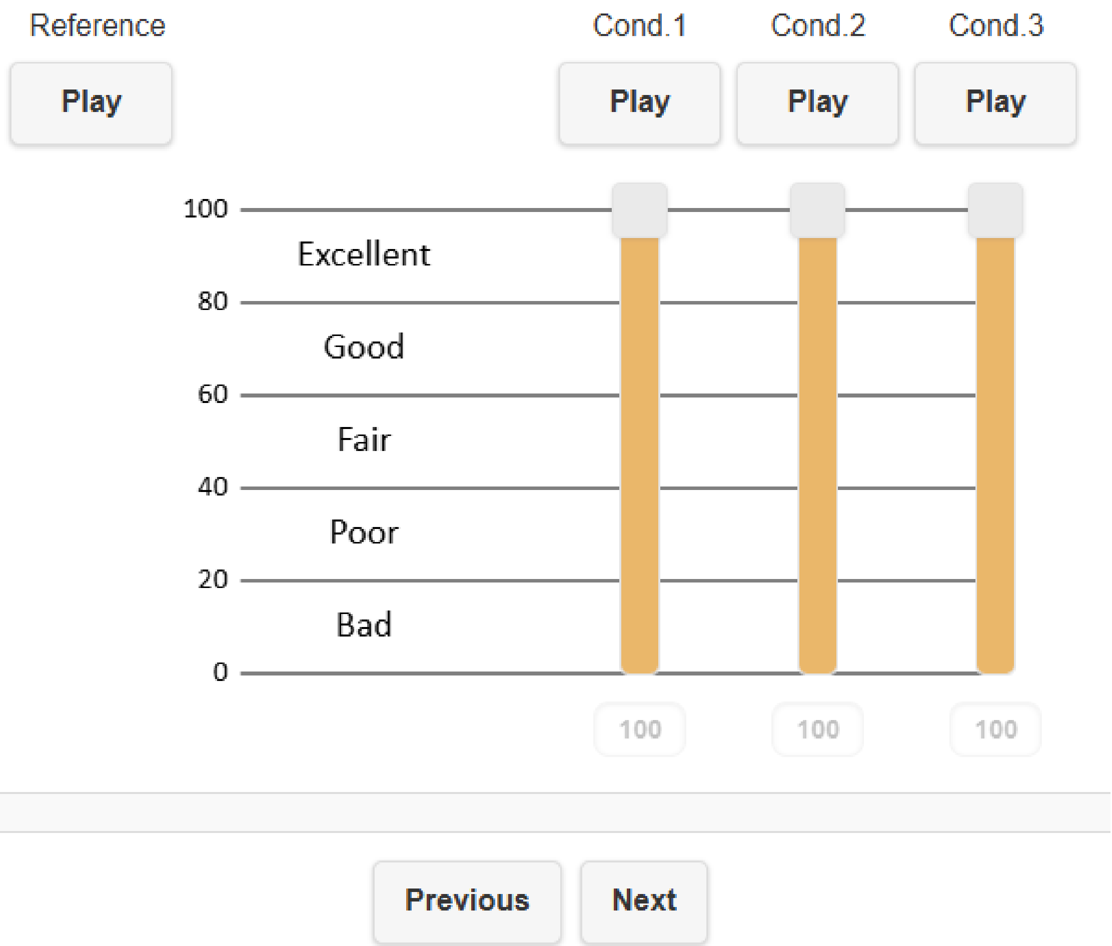


Figure 5.24: CQS.

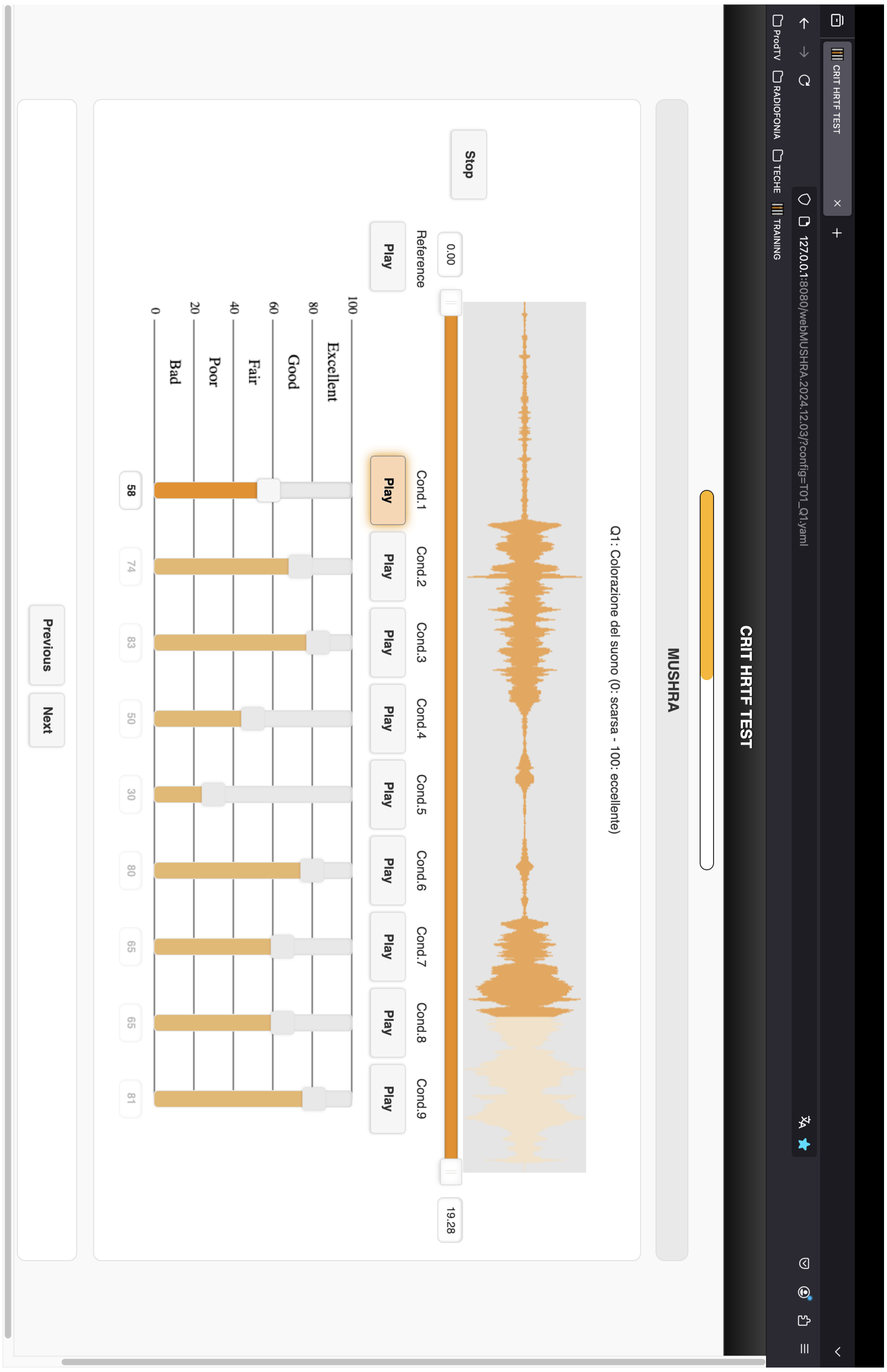


Figure 5.25: Type of page of the test.

Chapter 6

Results

6.1 Introduction

This chapter presents the results of the subjective test conducted to evaluate the perception and preferences of the different HRTFs.
The group of participants in the test consisted of 17 candidates, divided into two categories: 9 experienced sound engineers and 8 sound technicians. The sound technicians had an age range from 24 to 39 years, with an average age of 28.7 years, while the sound engineers had ages ranging from 27 to 60 years, with an average age of 41.36 years.
To provide a clear and concise representation of the data, the results are illustrated using boxplots and histograms:

- Boxplots: display the distribution of responses for each HRTF, highlighting the median, quartiles, and potential outliers, offering insights into the variability of the ratings.

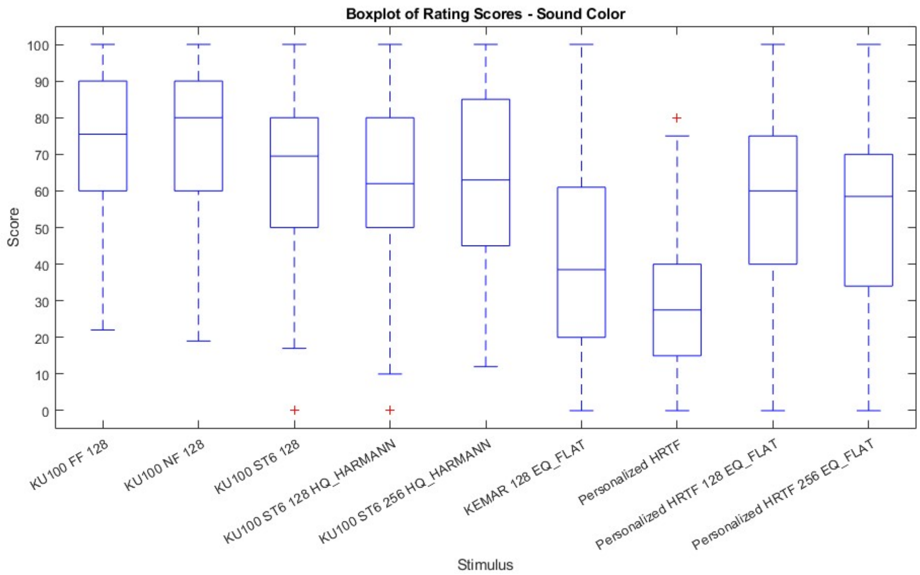
- Histograms: present the mean and standard deviation of the responses for each HRTF, allowing for an immediate comparison between the tested conditions.

The following sections provide a detailed analysis of the results for each of the five questions, highlighting notable trends and differences among the tested HRTFs.

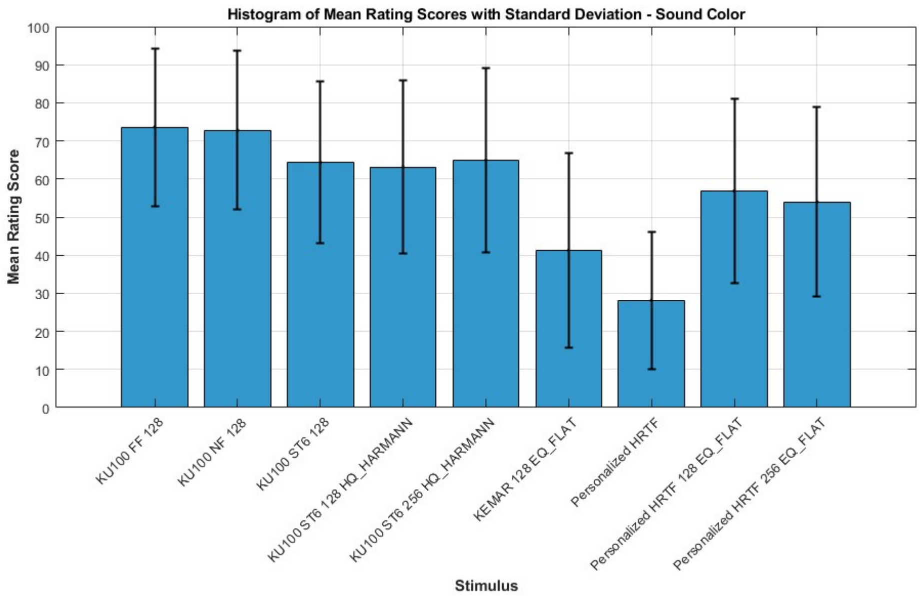
6.2 Results for the Five Attributes

Below are the plots of the various HRTFs calculated on the candidates for each attribute (Sound Color, Externalization, Localisation Accuracy, Scene Depth, Overall Subjective Quality):

Sound Color



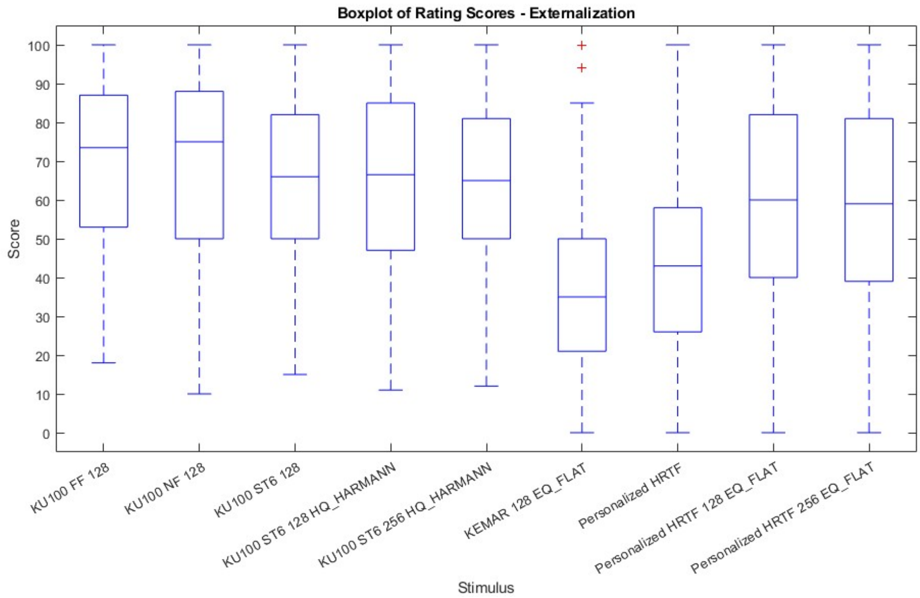
(a) Boxplot



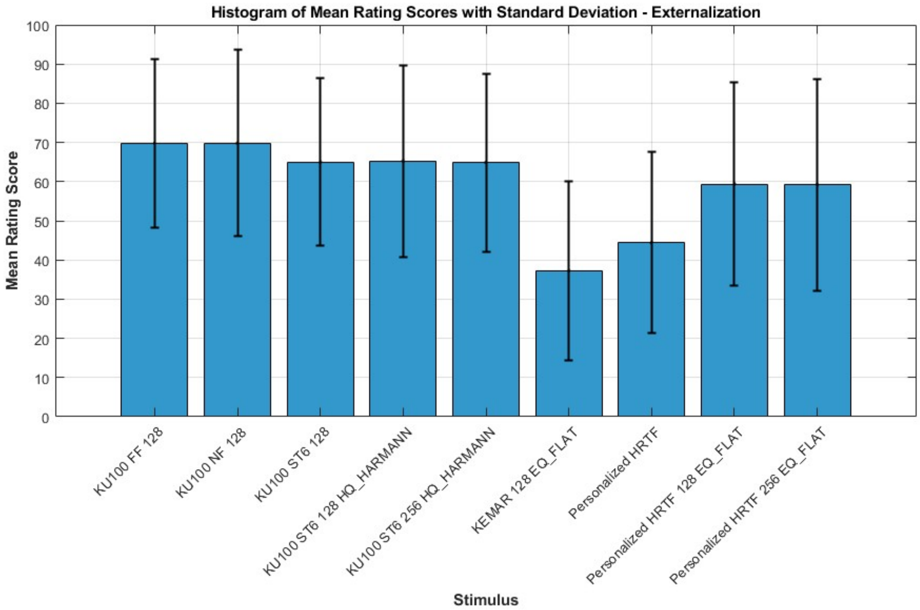
(b) Histogram

Figure 6.1: Boxplot and Histogram for Question 1 (Sound Color).

Externalization



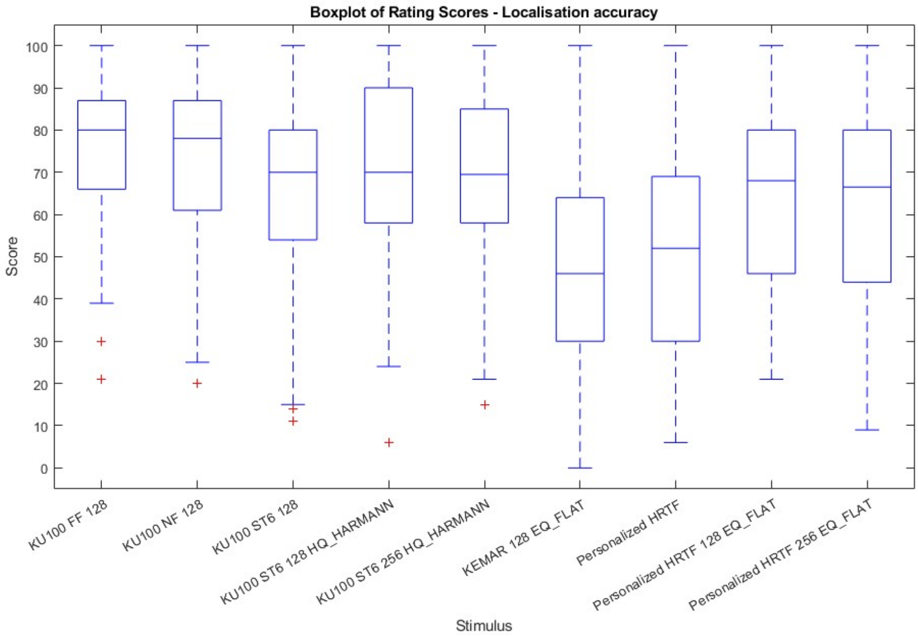
(a) Boxplot



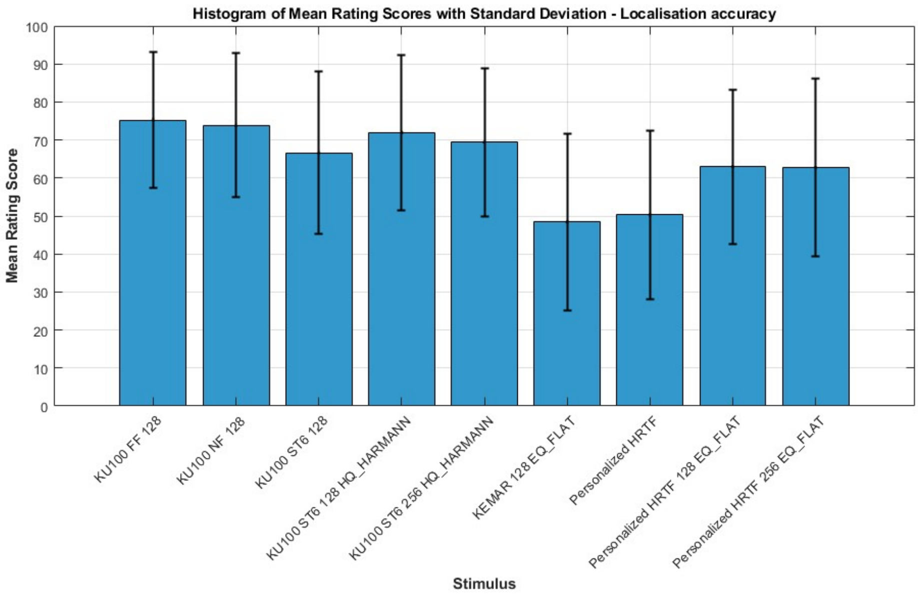
(b) Histogram

Figure 6.2: Boxplot and Histogram for Question 2 (Externalization).

Localisation accuracy



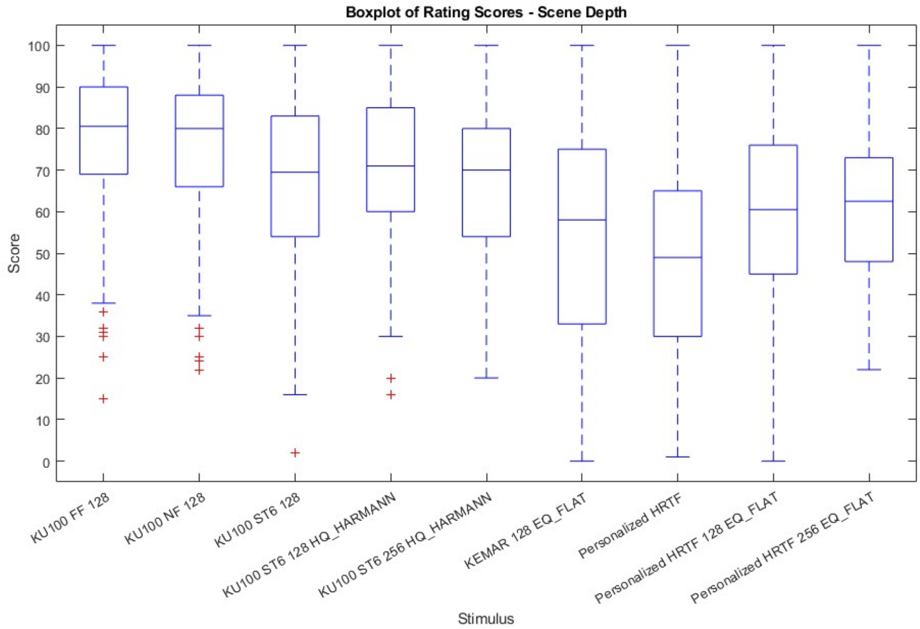
(a) Boxplot



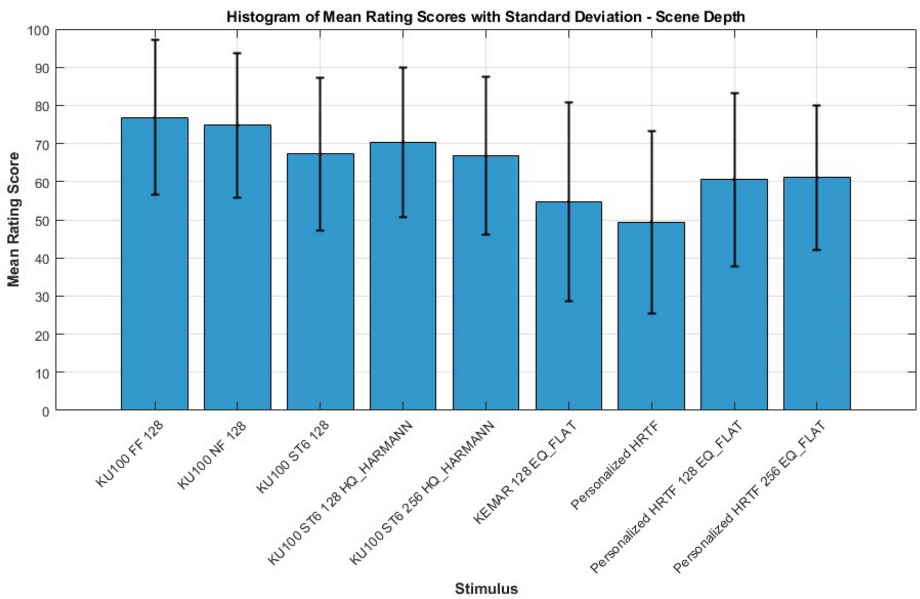
(b) Histogram

Figure 6.3: Boxplot and Histogram for Question 3 (Localisation accuracy).

Scene Depth



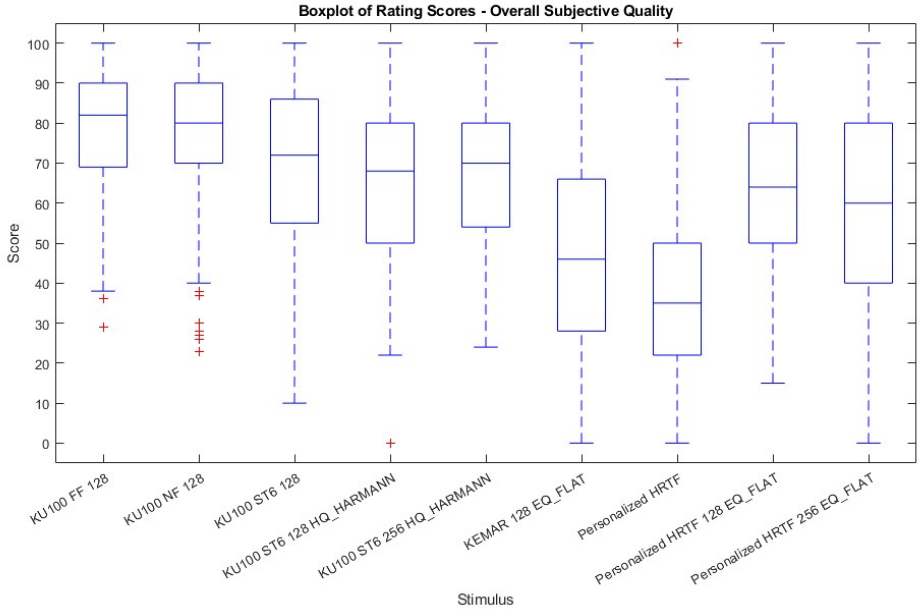
(a) Boxplot



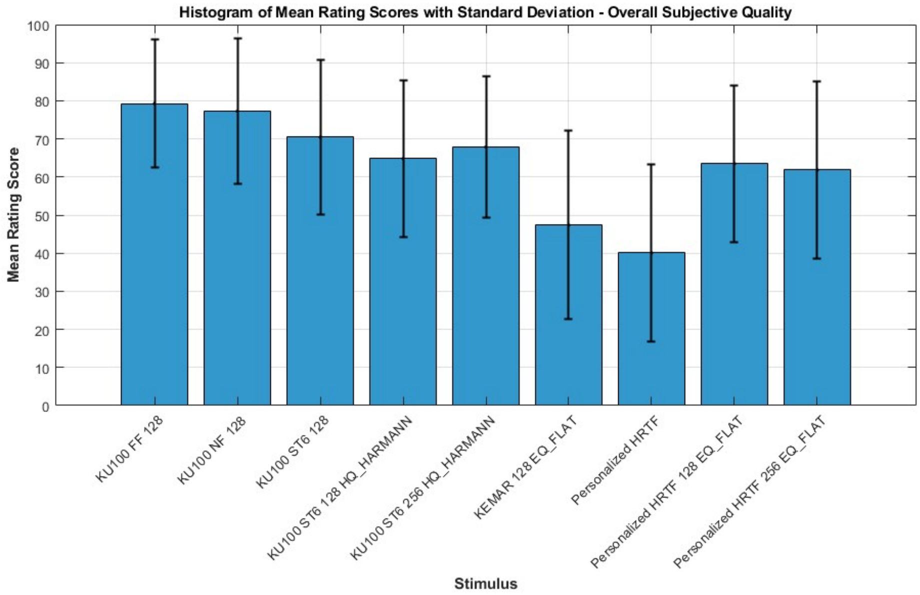
(b) Histogram

Figure 6.4: Boxplot and Histogram for Question 4 (Scene Depth).

Overall Subjective Quality



(a) Boxplot

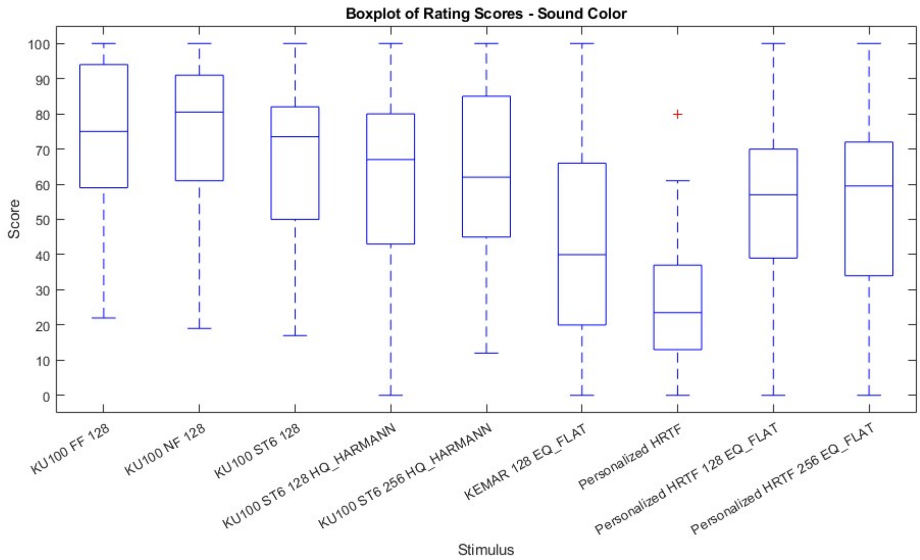


(b) Histogram

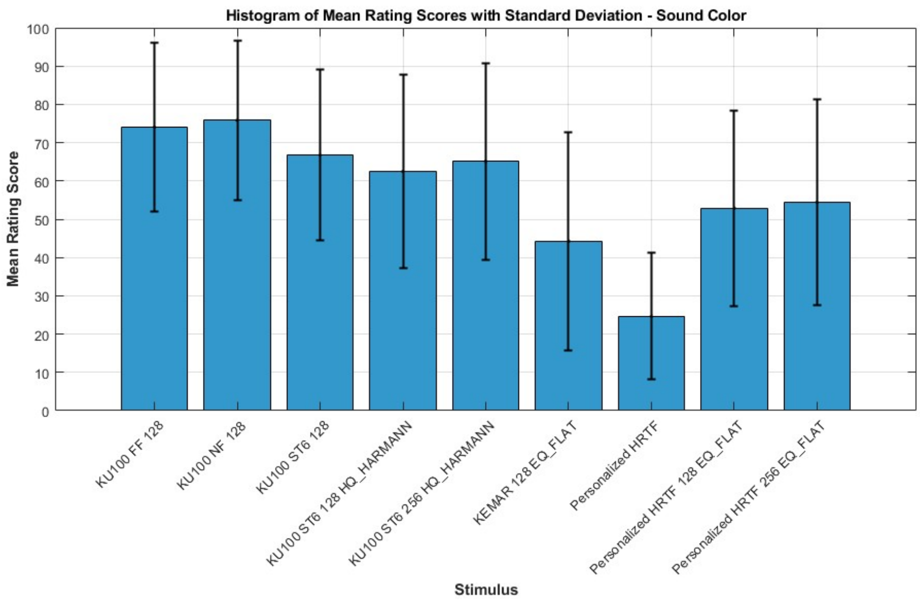
Figure 6.5: Boxplot and Histogram for Question 5 (Overall Subjective Quality).

Below are the plots divided into the two groups of expert sound engineers (Group 1) and sound technicians (Group 2):

Sound Color



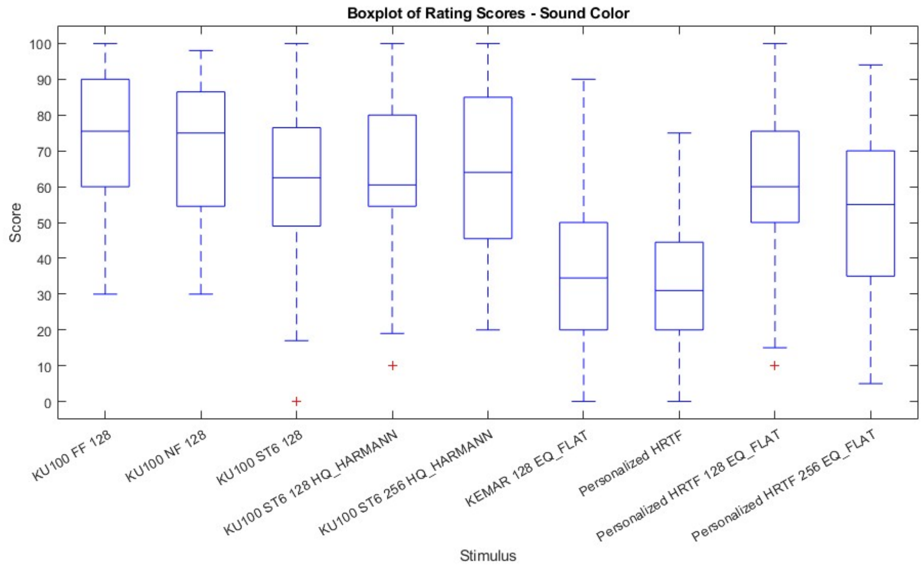
(a) Boxplot



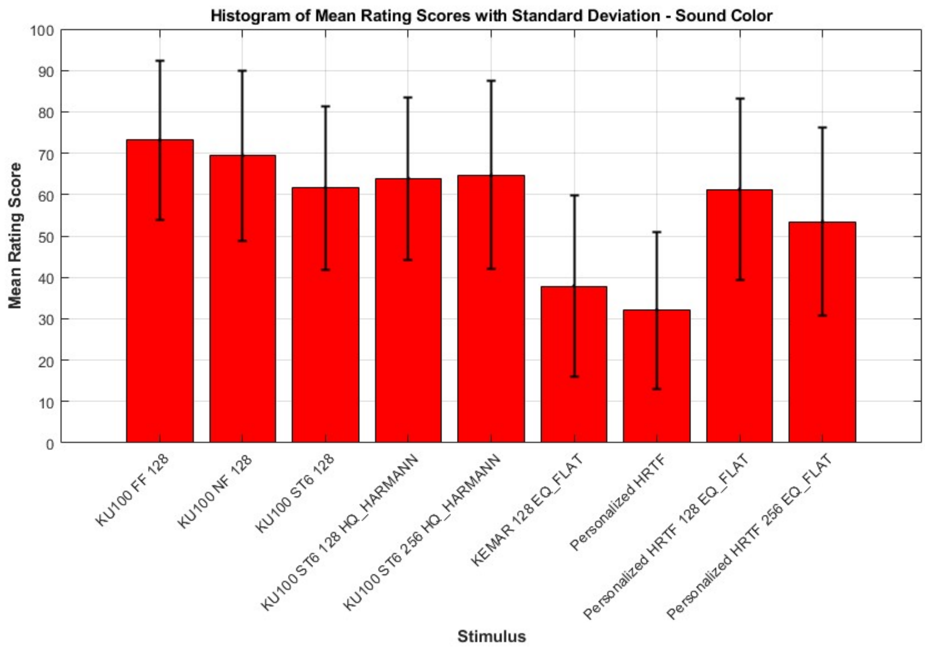
(b) Histogram

Figure 6.6: Boxplot and Histogram for Question 1 for Group 1 (Sound Color).

Sound Color



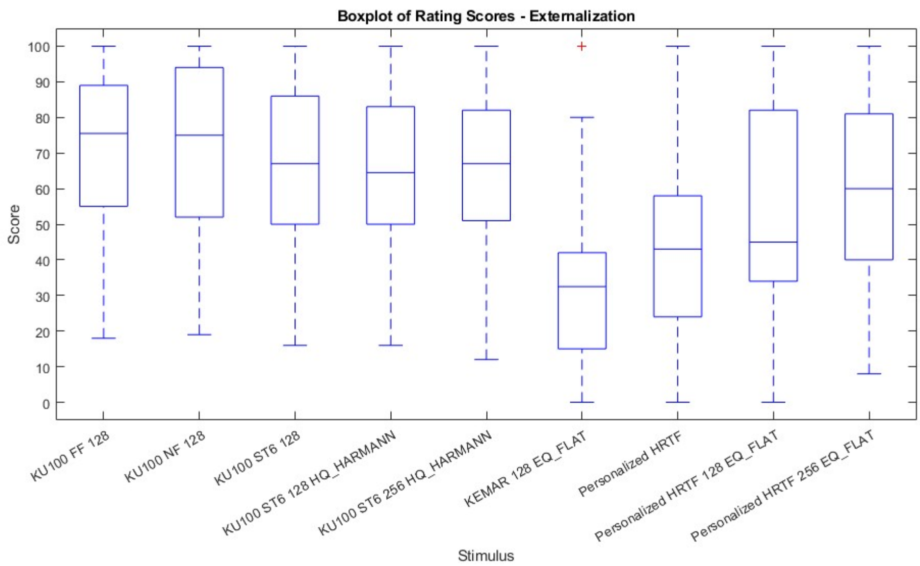
(a) Boxplot



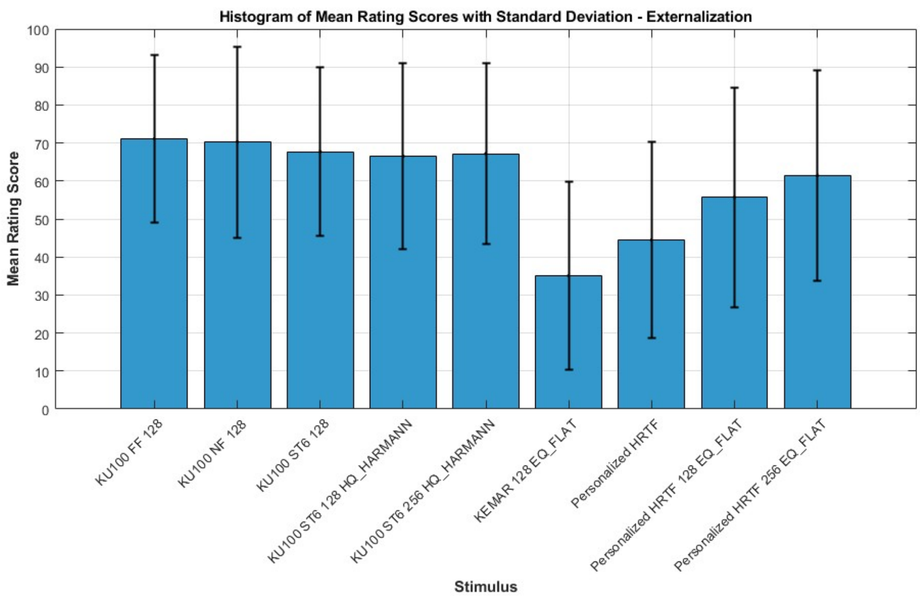
(b) Histogram

Figure 6.7: Boxplot and Histogram for Question 1 for Group 2 (Sound Color).

Externalization



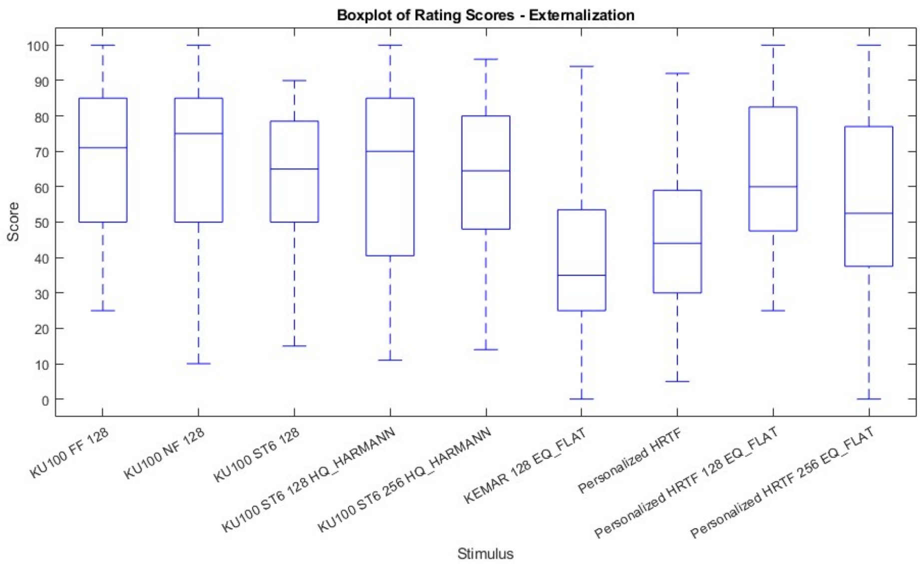
(a) Boxplot



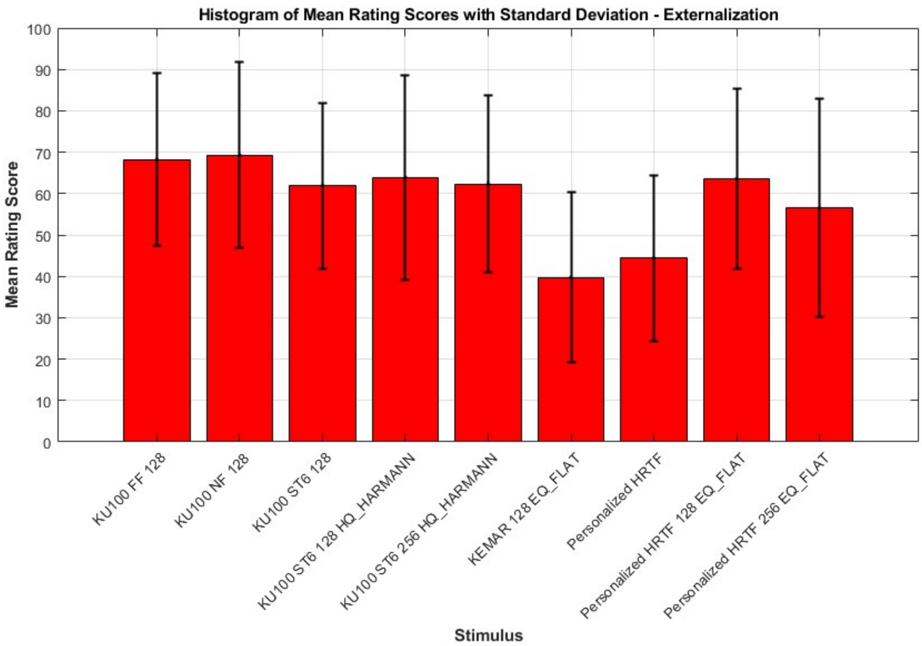
(b) Histogram

Figure 6.8: Boxplot and Histogram for Question 2 for Group 1 (Externalization).

Externalization



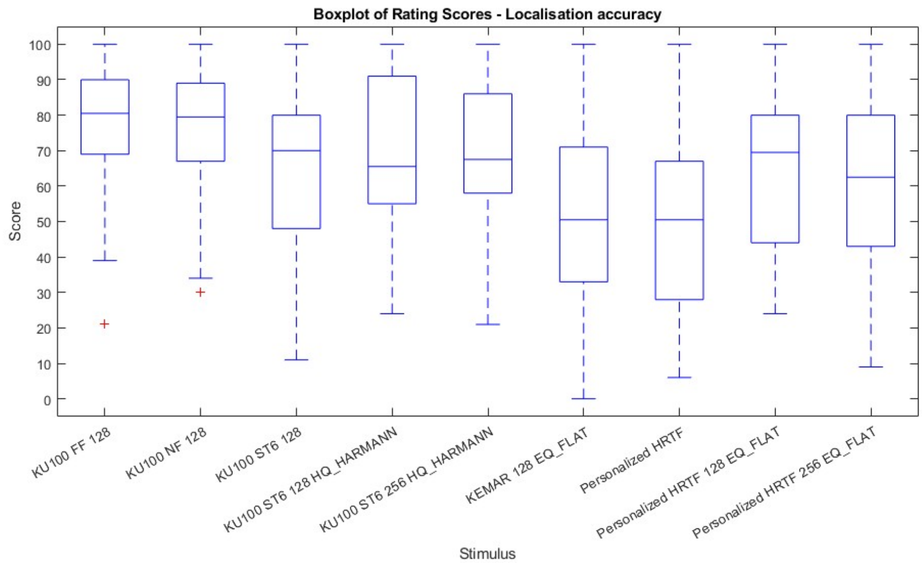
(a) Boxplot



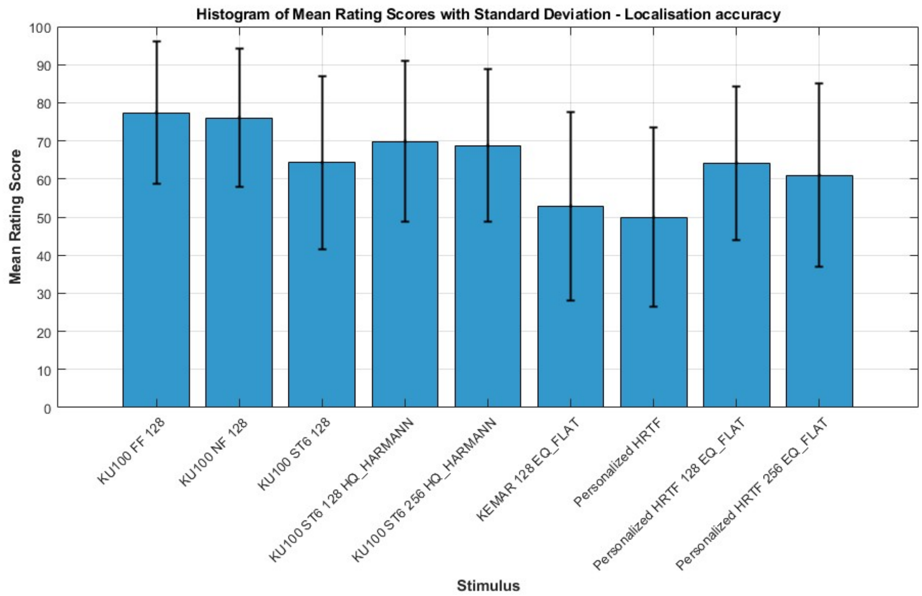
(b) Histogram

Figure 6.9: Boxplot and Histogram for Question 2 for Group 2 (Externalization).

Localisation accuracy



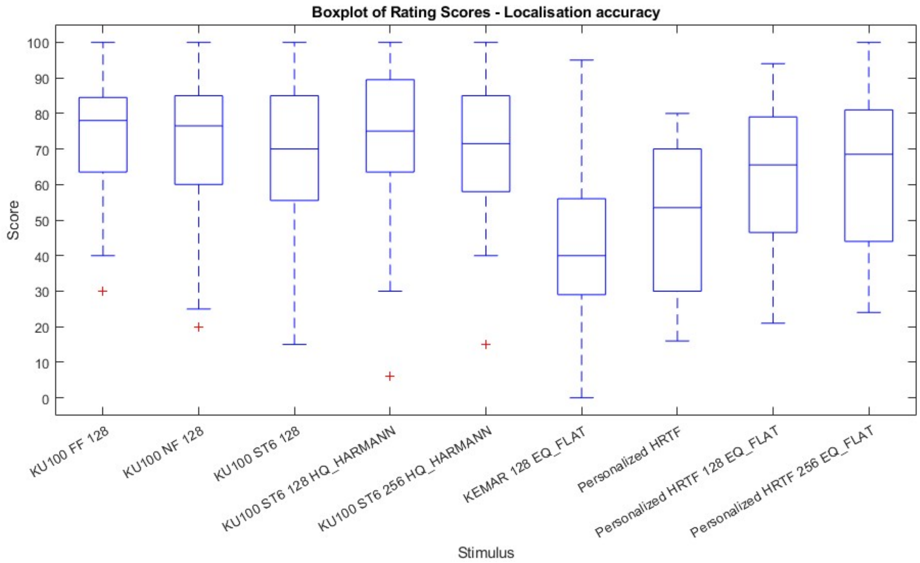
(a) Boxplot



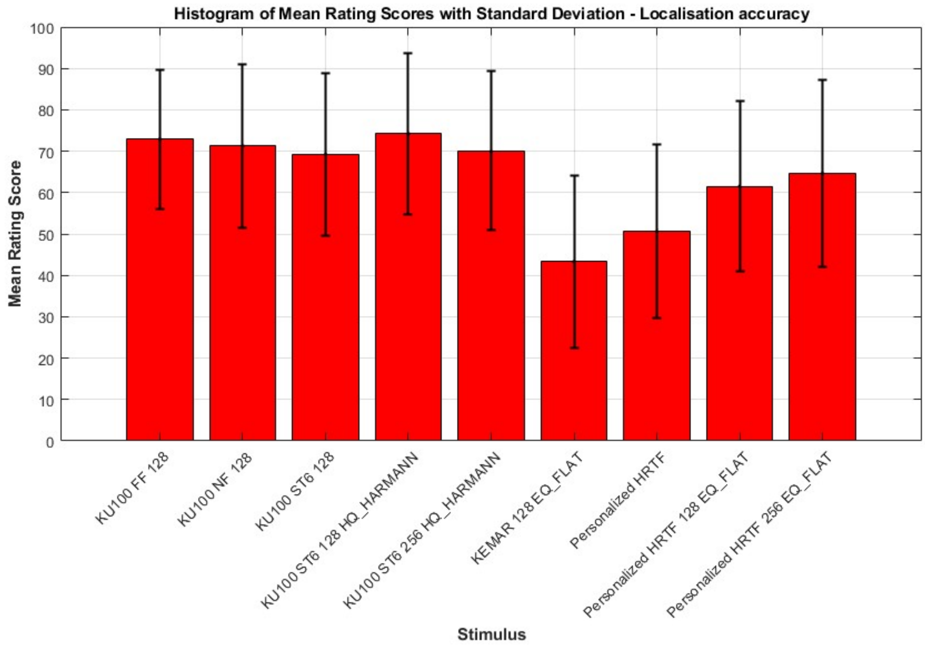
(b) Histogram

Figure 6.10: Boxplot and Histogram for Question 3 for Group 1 (Localisation accuracy).

Localisation accuracy



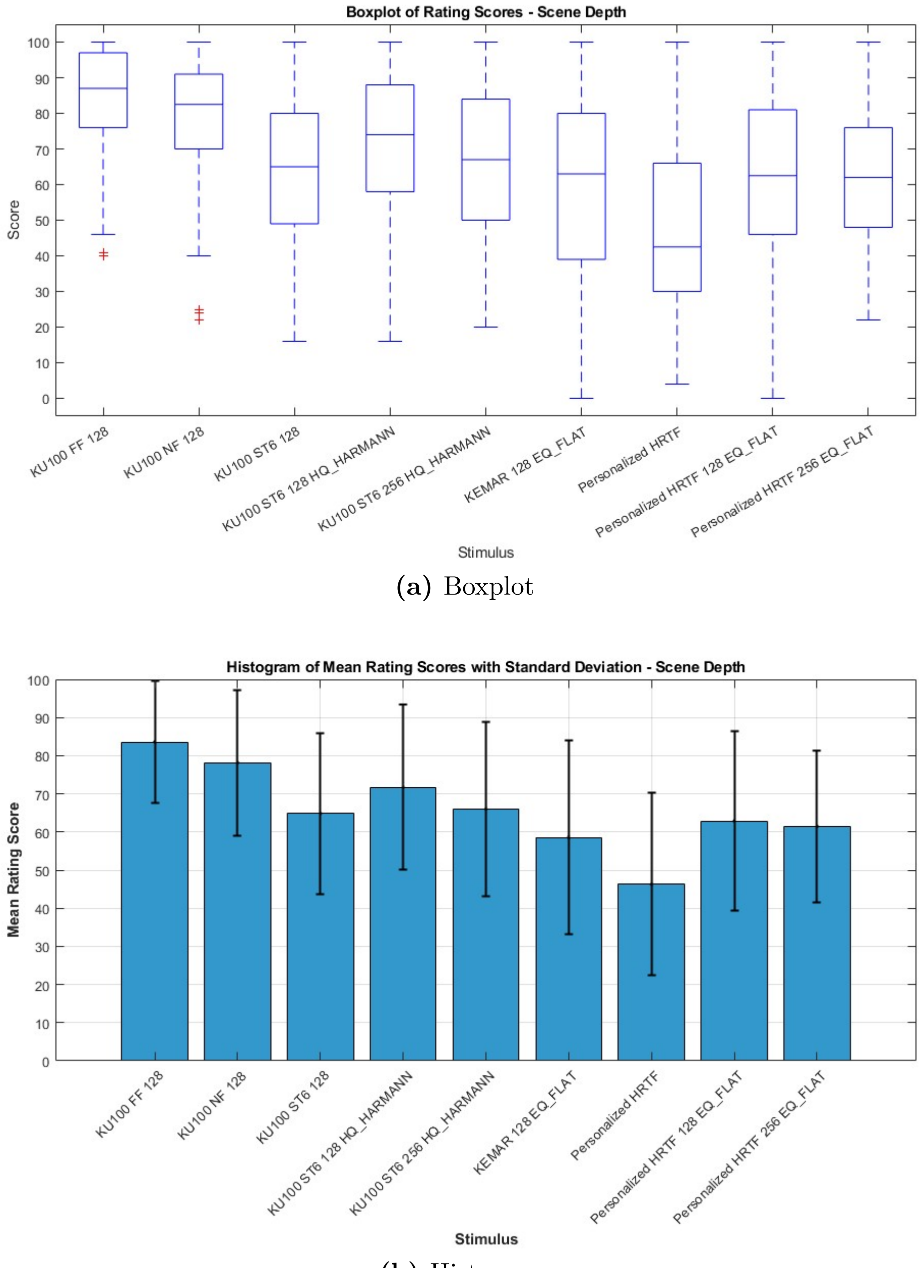
(a) Boxplot



(b) Histogram

Figure 6.11: Boxplot and Histogram for Question 3 for Group 2 (Localisation accuracy).

Scene Depth

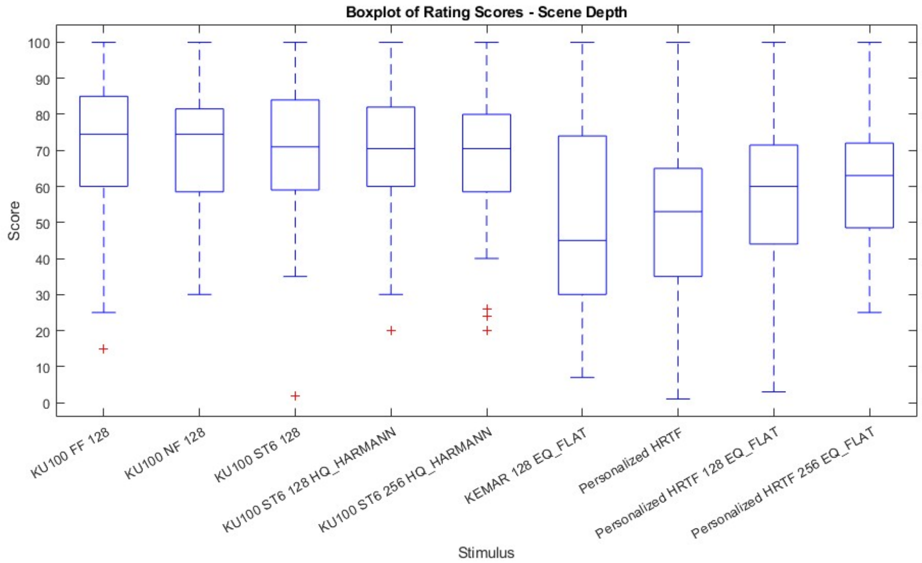


(a) Boxplot

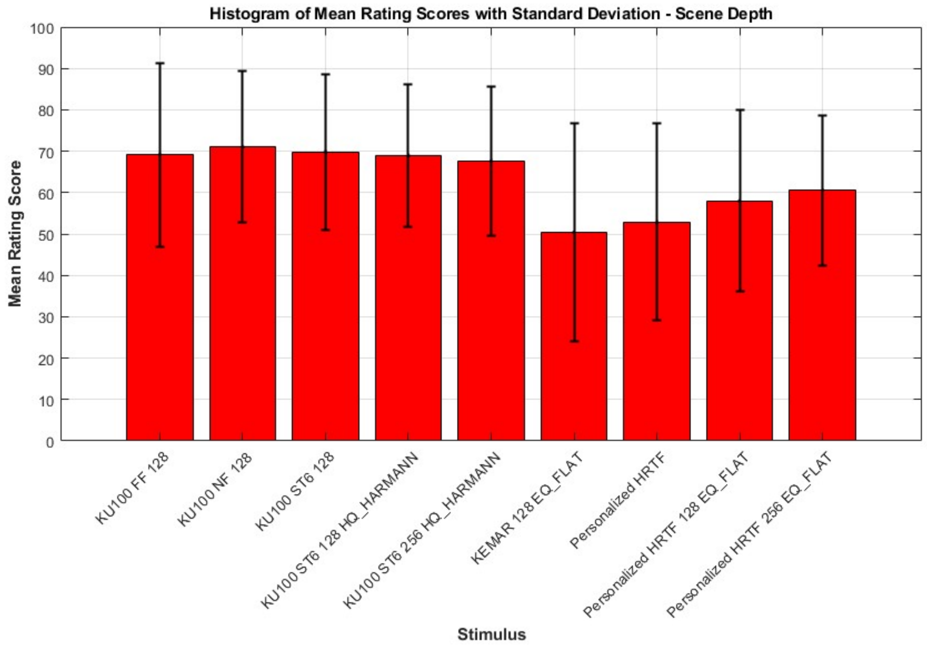
(b) Histogram

Figure 6.12: Boxplot and Histogram for Question 4 for Group 1 (Scene Depth).

Scene Depth



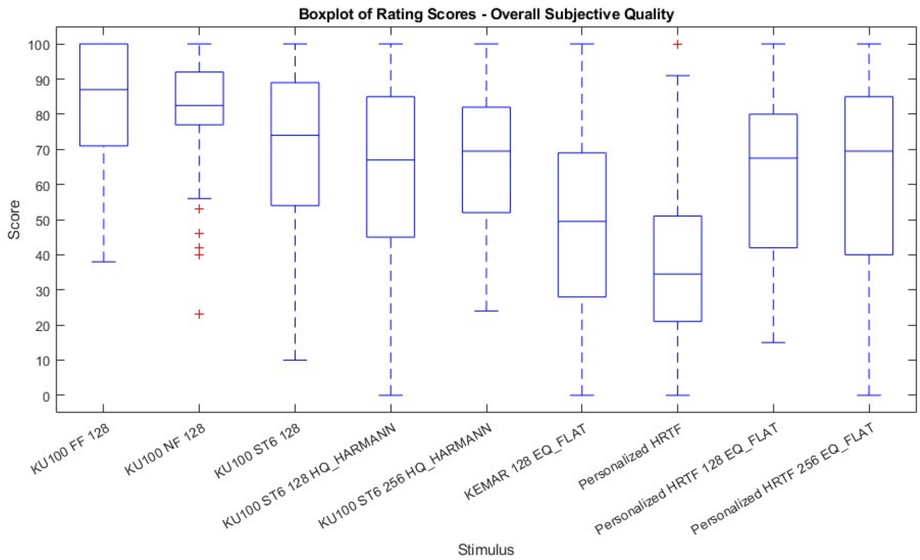
(a) Boxplot



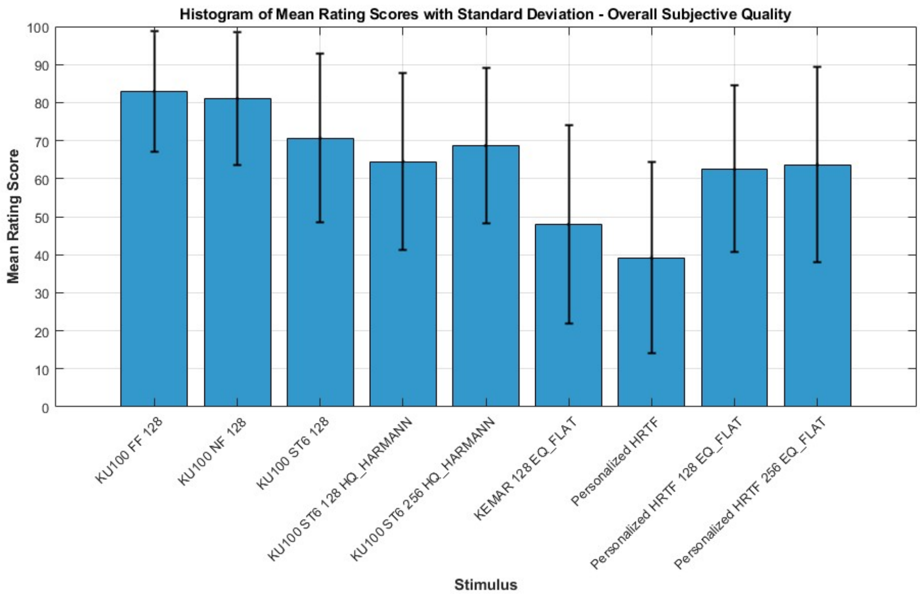
(b) Histogram

Figure 6.13: Boxplot and Histogram for Question 4 for Group 2 (Scene Depth).

Overall Subjective Quality



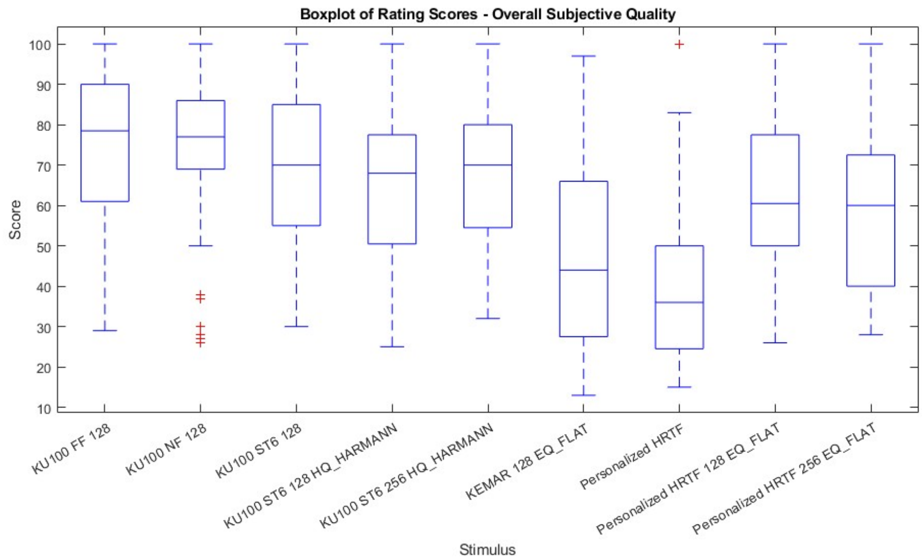
(a) Boxplot



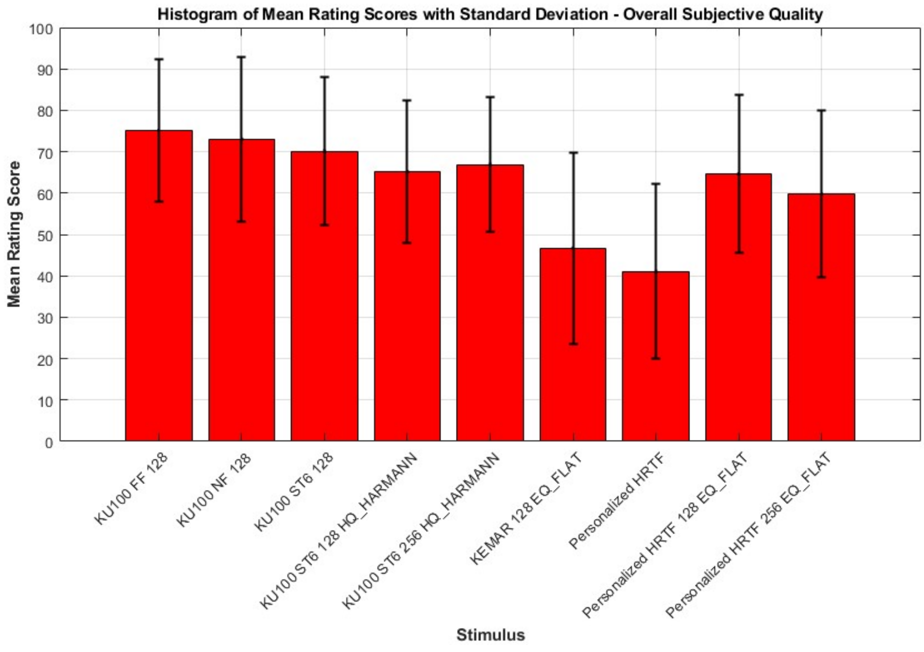
(b) Histogram

Figure 6.14: Boxplot and Histogram for Question 5 for Group 1 (Overall Subjective Quality).

Overall Subjective Quality



(a) Boxplot



(b) Histogram

Figure 6.15: Boxplot and Histogram for Question 5 for Group 2 (Overall Subjective Quality).

Overall Boxplots by Group

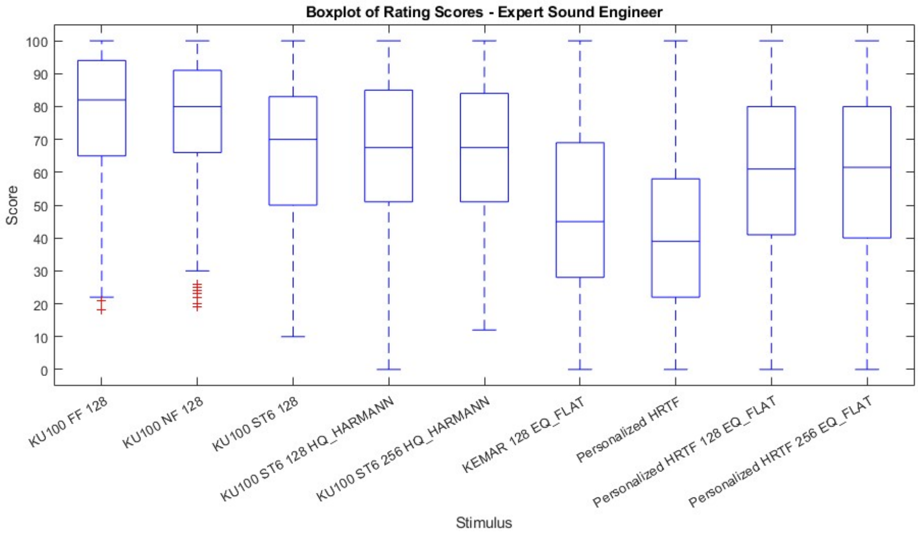


Figure 6.16: Boxplot for Group 1

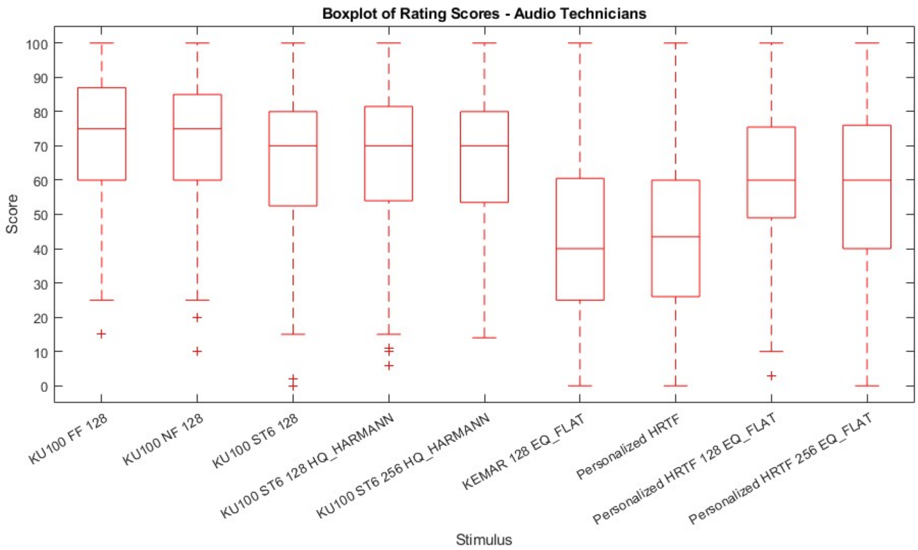


Figure 6.17: Boxplot for Group 2

6.3 Multi-Histogram

This charts present the mean rating scores of the different HRTFs across the five attributes, each bar represents the average rating for a given HRTF within a specific evaluation group, while the error bars indicate the standard deviation, providing insight into the variability of responses. This visualization allows for a comparative analysis of the perceived quality of different HRTF processing methods, highlighting both trends and potential inconsistencies in subjective assessments.

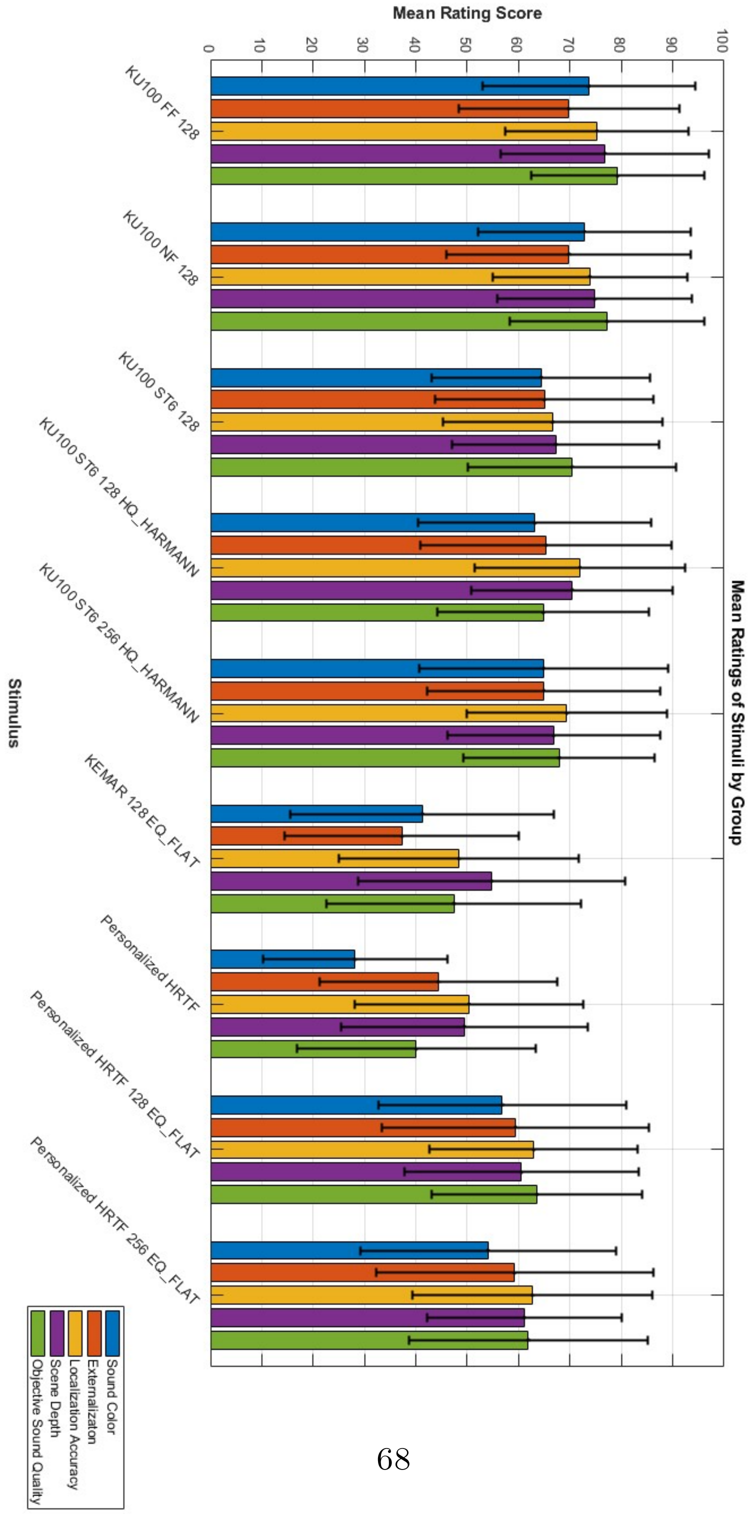


Figure 6.18: Multi-Histogram.

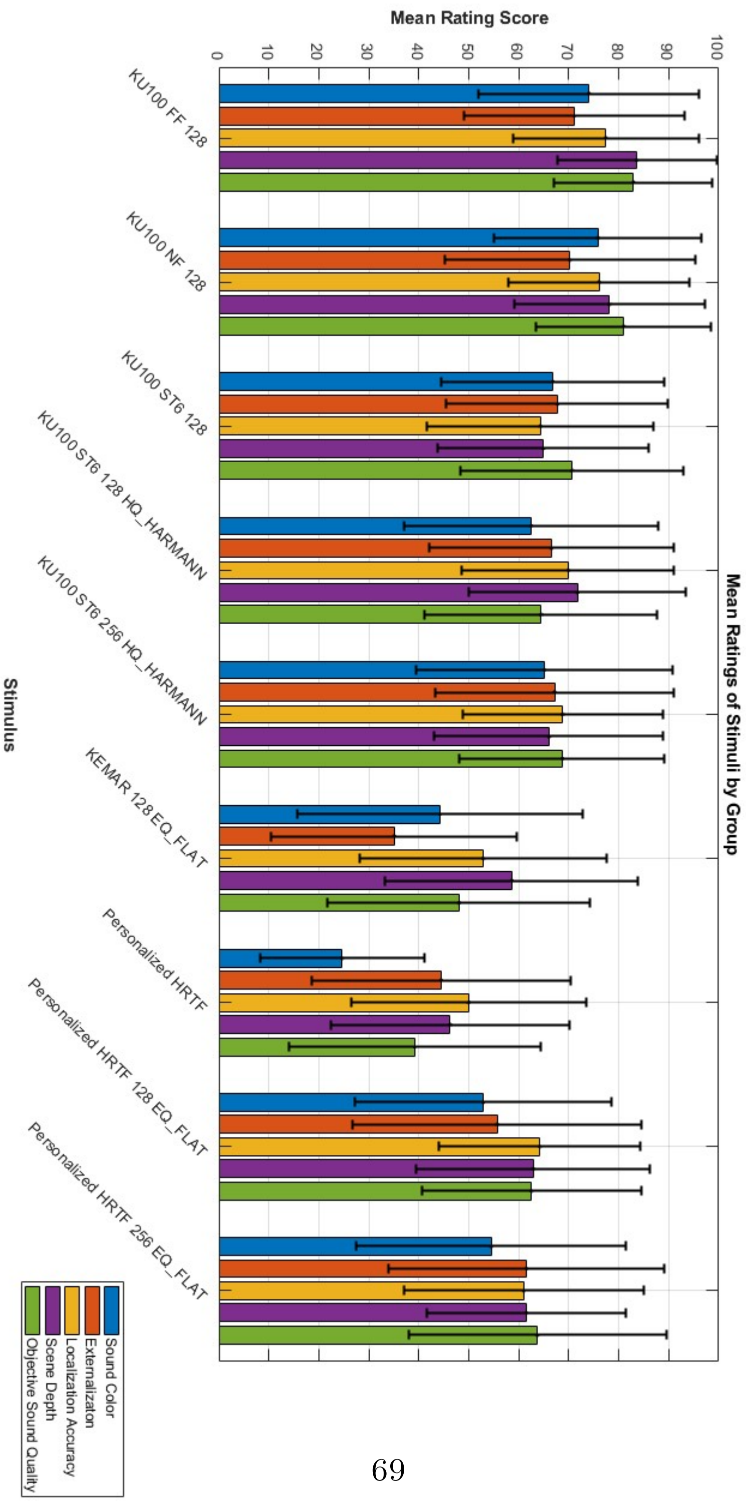


Figure 6.19: Multi-Histogram for Group 1.

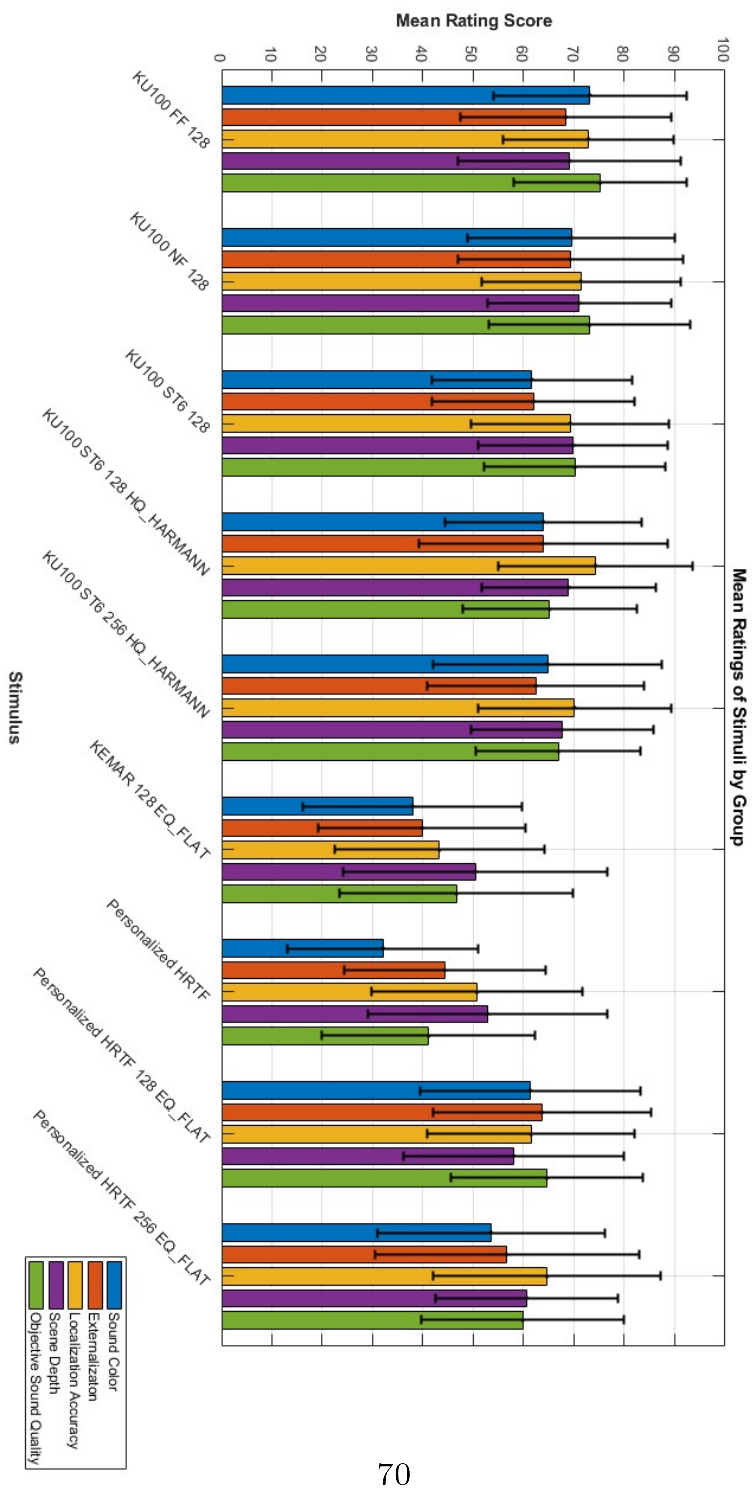
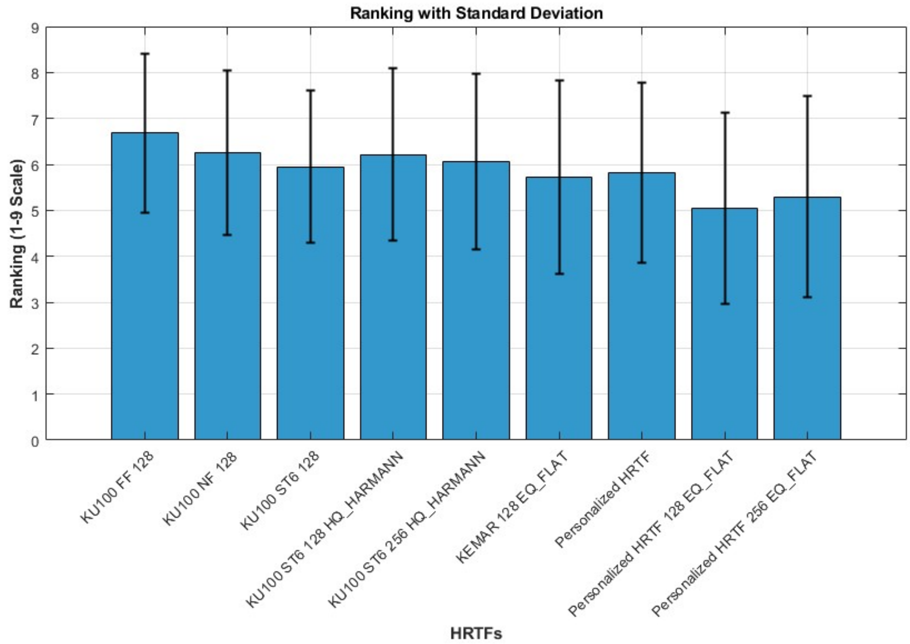


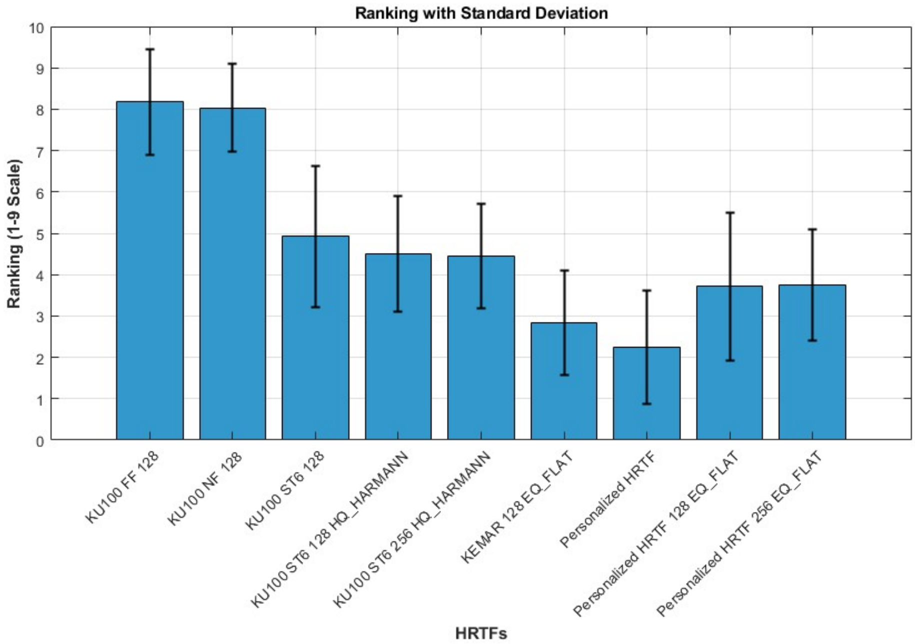
Figure 6.20: Multi-Histogram for Group 2.

6.4 Ranking

The ranking method was used to visualize the results of individual candidates, calculated based on the average score of each HRTF across the different attrubutes. This approach allows for a more cohesive interpretation of the data, making it easier to compare the performance of different HRTFs across participants: by ranking the results, variations in individual preferences become clearer, providing a structured overview that highlights the most and least preferred HRTFs in a more intuitive way.

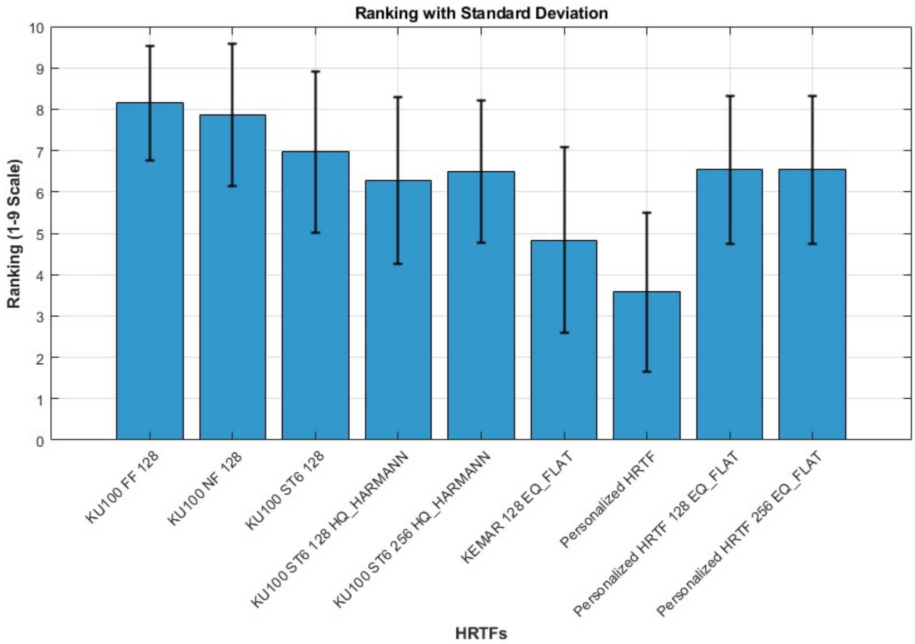


(a) Ranking Candidate 01.

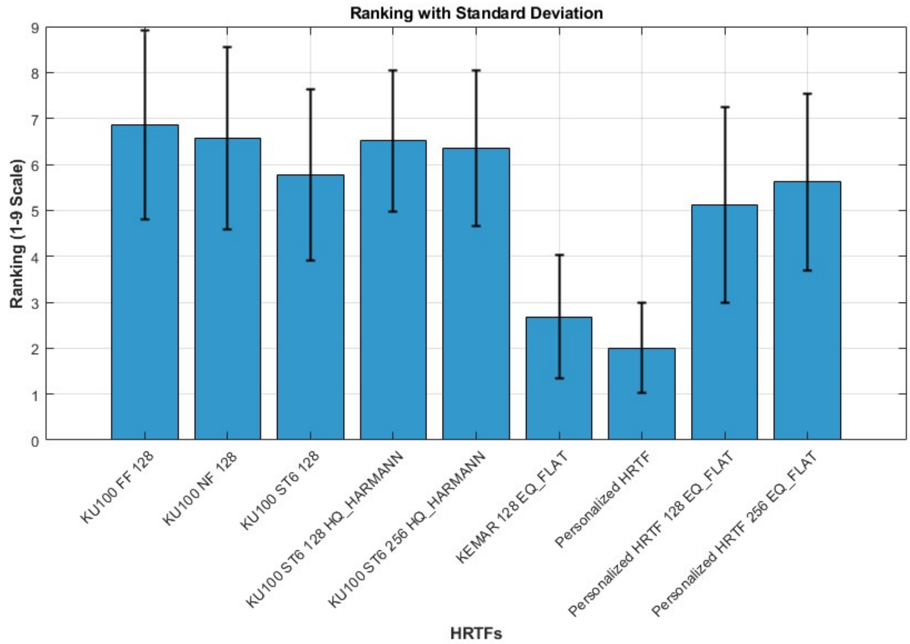


(b) Ranking Candidate 02.

Figure 6.21: Ranking Candidates 01 and 02.

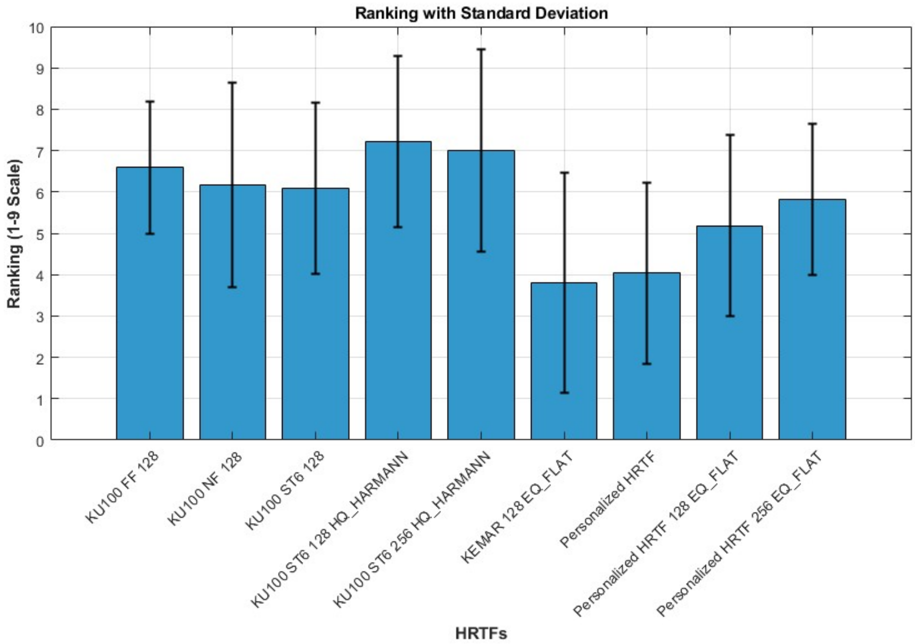


(a) Ranking Candidate 03.

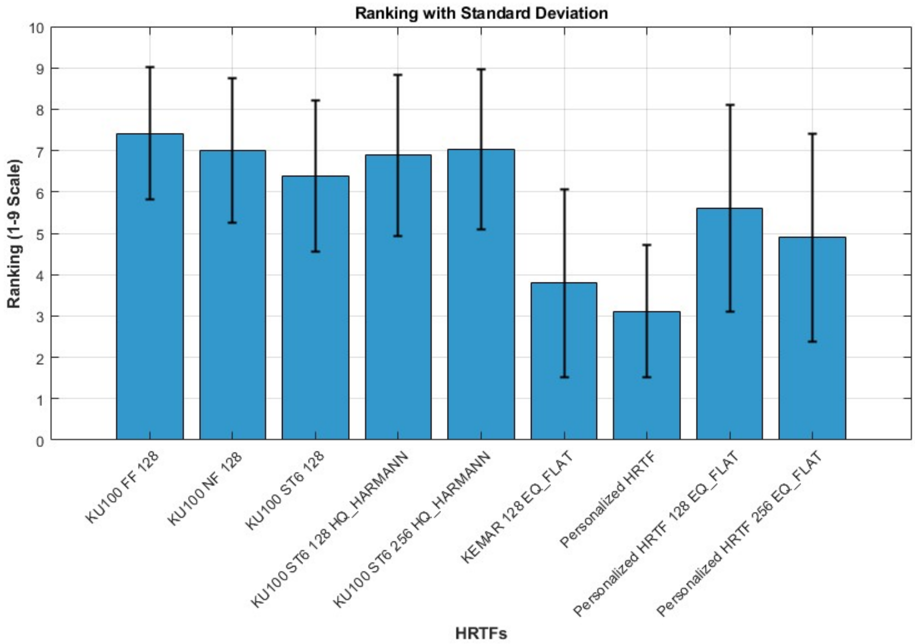


(b) Ranking Candidate 04.

Figure 6.22: Ranking Candidates 03 and 04.

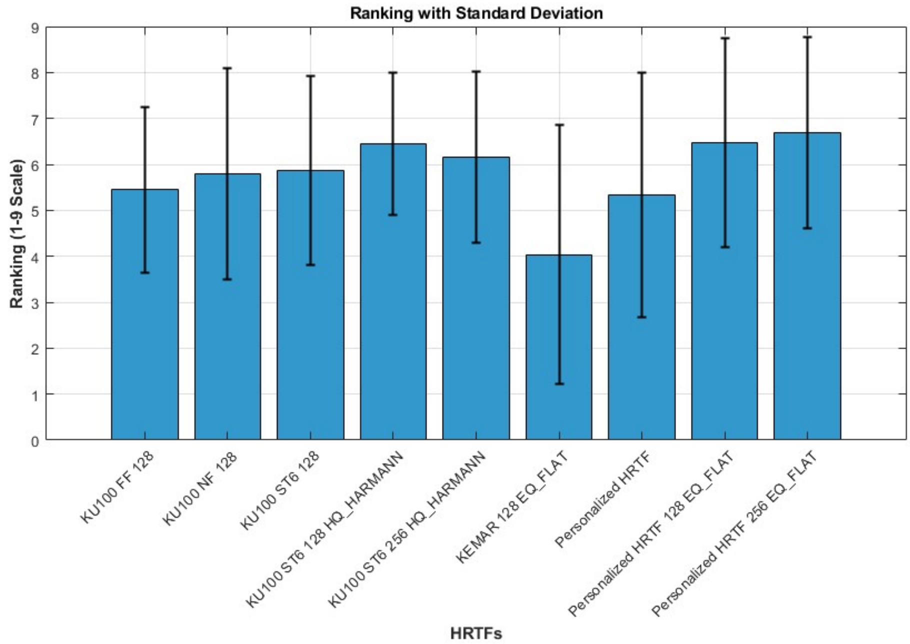


(a) Ranking Candidate 05.

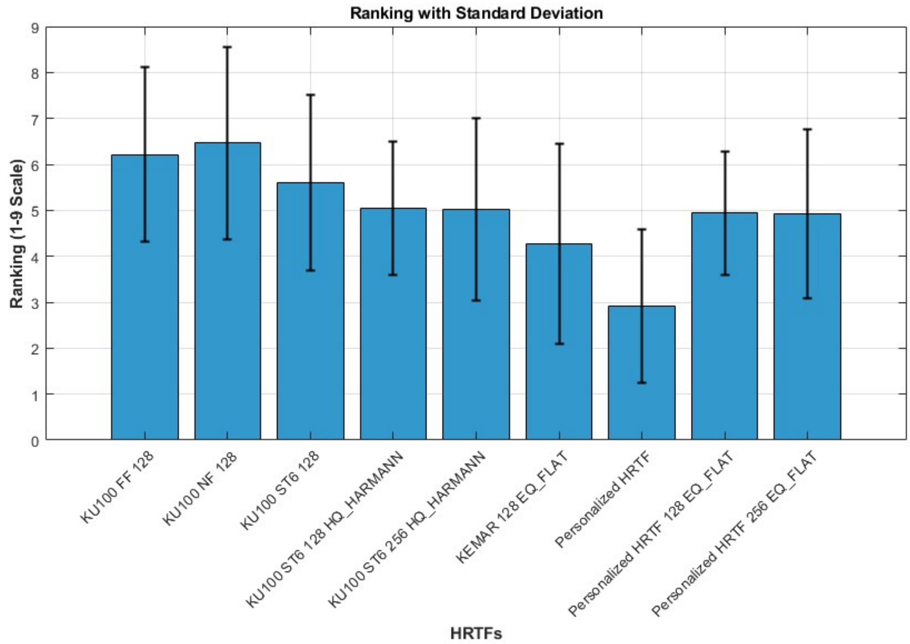


(b) Ranking Candidate 06.

Figure 6.23: Ranking Candidates 05 and 06.

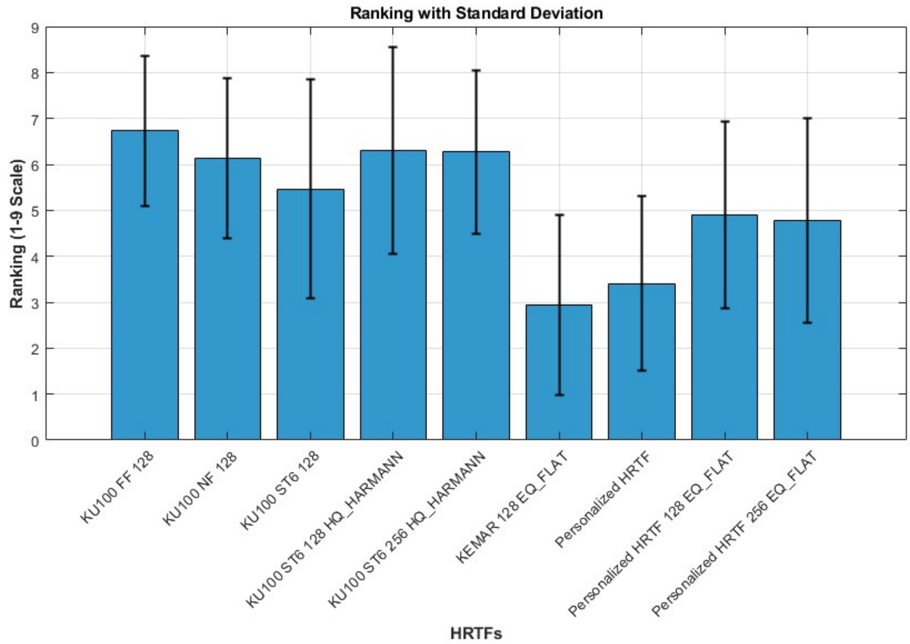


(a) Ranking Candidate 07.

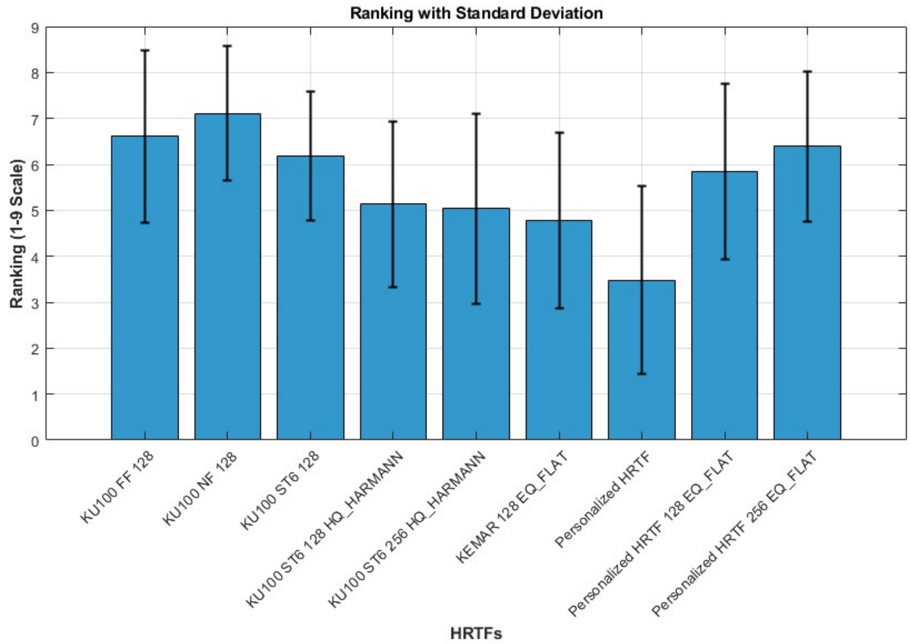


(b) Ranking Candidate 08.

Figure 6.24: Ranking Candidates 07 and 08.

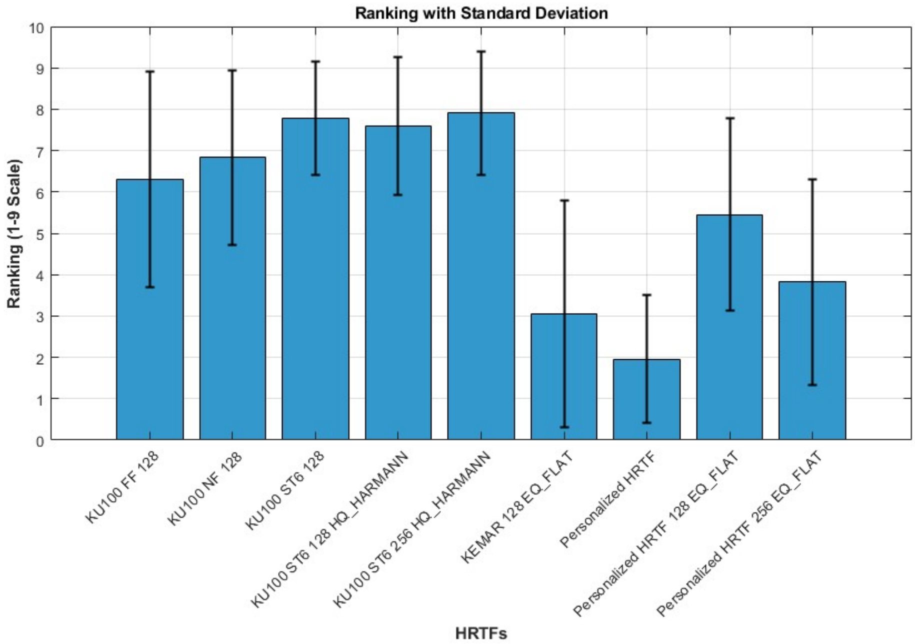


(a) Ranking Candidate 09.

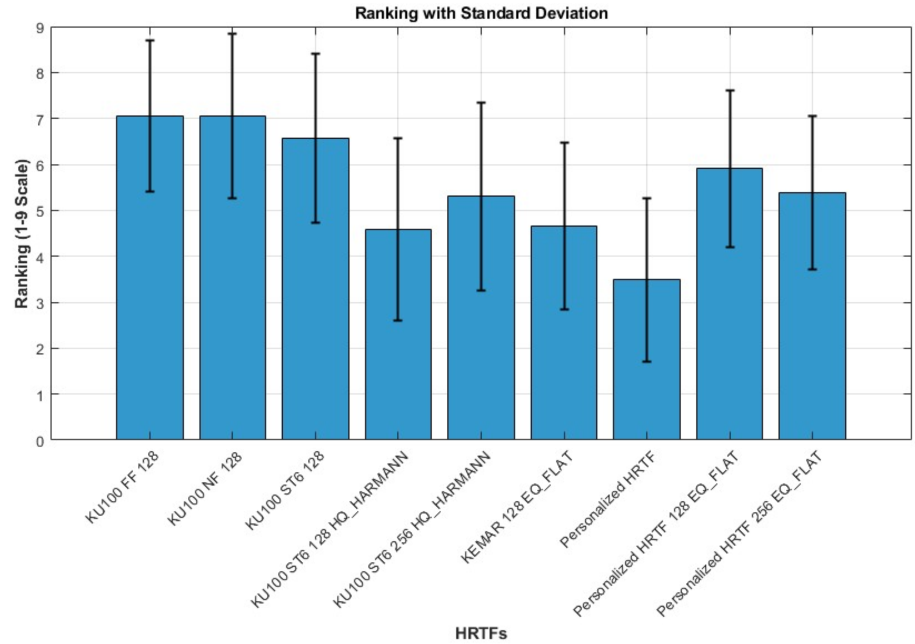


(b) Ranking Candidate 10.

Figure 6.25: Ranking Candidates 09 and 10.

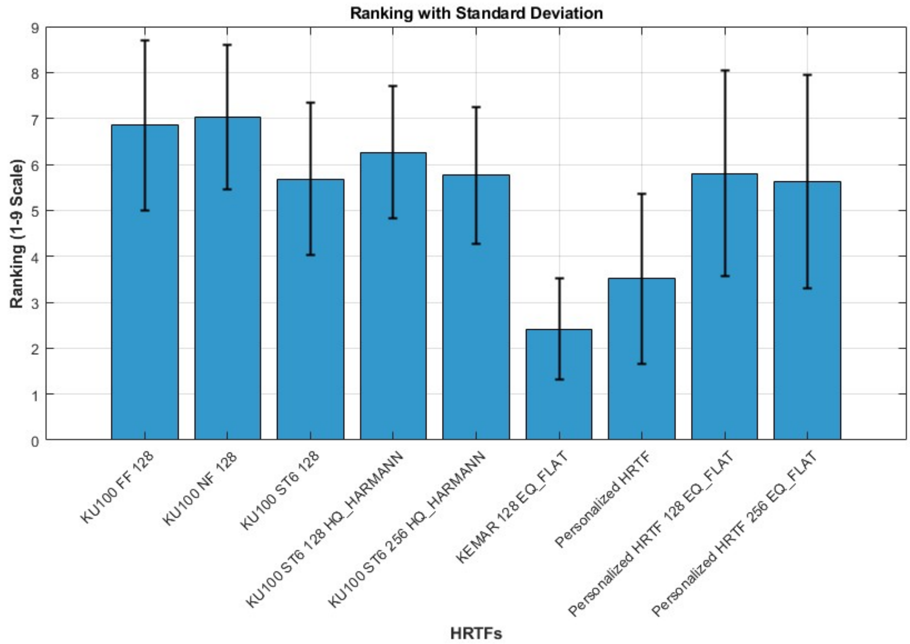


(a) Ranking Candidate 11.

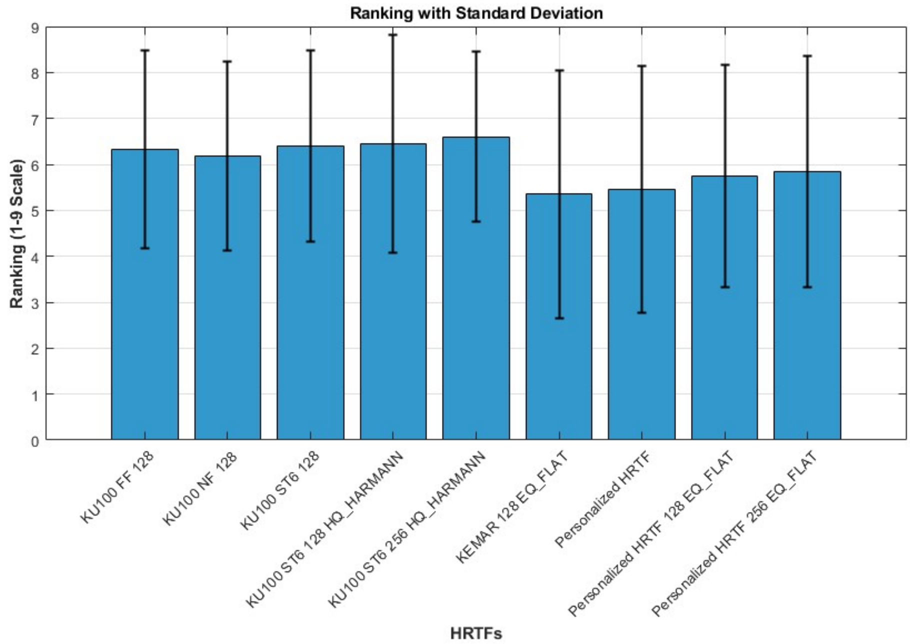


(b) Ranking Candidate 12.

Figure 6.26: Ranking Candidates 11 and 12.

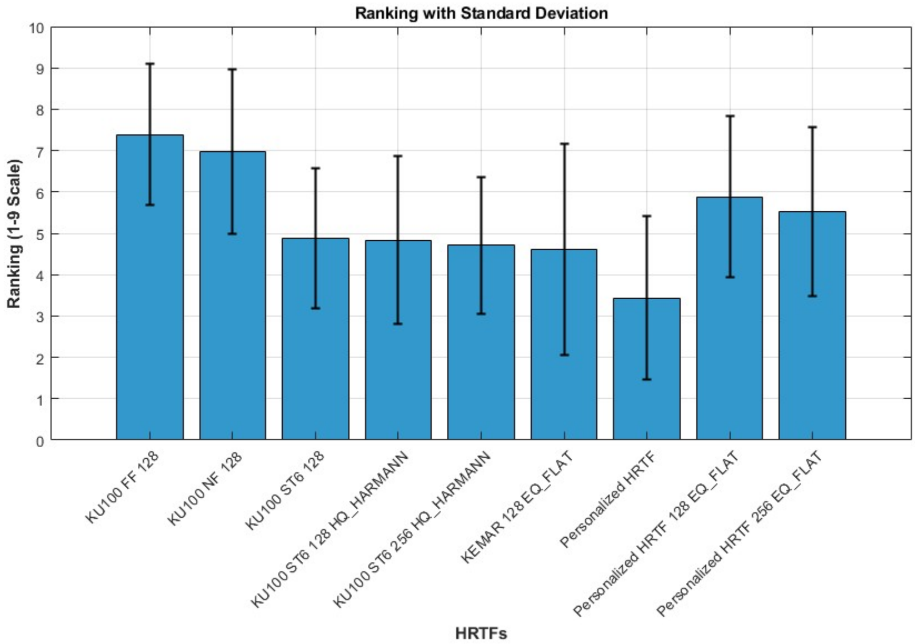


(a) Ranking Candidate 13.

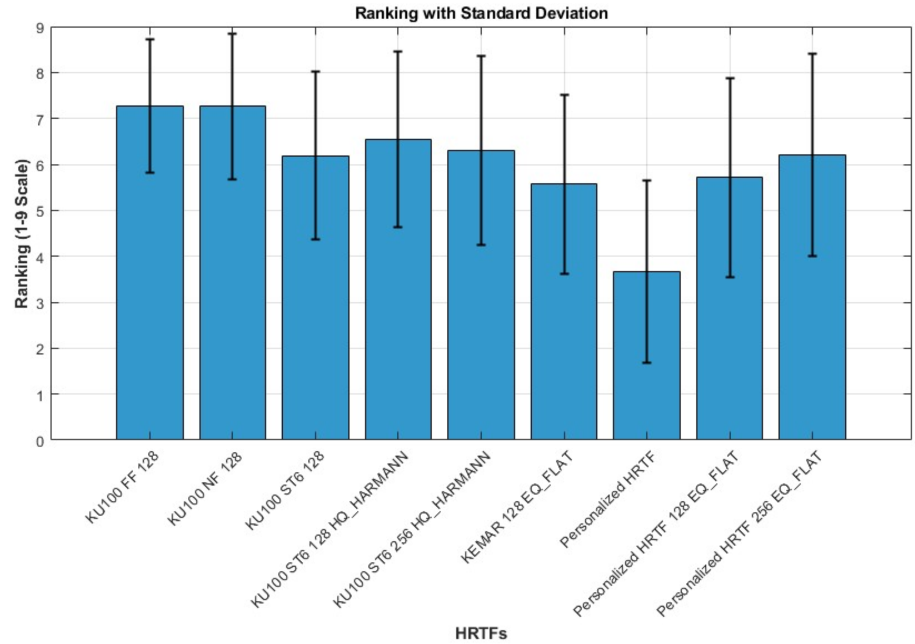


(b) Ranking Candidate 14.

Figure 6.27: Ranking Candidates 13 and 14.



(a) Ranking Candidate 15.



(b) Ranking Candidate 16.

Figure 6.28: Ranking Candidates 15 and 16.

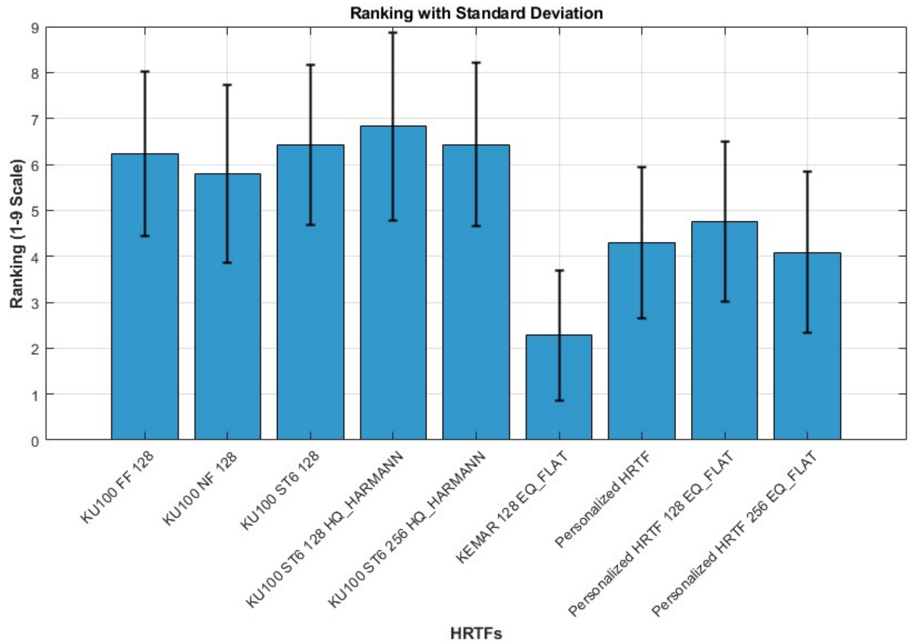


Figure 6.29: Ranking Candidates 17.

Chapter 7

Discussion

Before discussing the data, specific statistical tests were applied based on the nature of the data to assess differences between experimental conditions and subject groups. First, the assumption of normality was verified, followed by pairwise comparisons using non-parametric tests. The first statistical test applied was the Shapiro test B, for all distributions compared in pairs, the test was applied to each individual distribution to verify the assumption of data normality. The null hypothesis (H_0) of the test states that the tested distribution X is normal. For all analyzed distributions, the test reported p-values lower than $p < 0.05$, leading to the rejection of the null hypothesis and indicating that the distributions do not follow a normal distribution. For this reason, all subsequent pairwise comparisons were performed using non-parametric tests.
The first non-parametric test was the Wilcoxon test: the objective of this comparison was to determine whether there are differences between the various HRTFs while keeping the subject group constant (experts, non-experts, overall). For this reason, data from individual tracks, for each attribute, were aggregated. The Wilcoxon test was applied to each pair of HRTFs to test the null hypothesis of equality between two non-normal distributions, with continuous data, within the same group of subjects (dependent distributions). A total of 35 comparisons were conducted.
Results:

- Expert group: The hypothesis of equality between two HRTFs can always be rejected, except in the following cases:
  - HRTF1-HRTF2 (KU100 FF 128 - KU100 NF 128)
  - HRTF3-HRTF4-HRTF5 (ST6 128 - ST6 128 EQ - ST6 256 EQ)
  - HRTF8-HRTF9 (pHRTF 128 EQ - pHRTF 256 EQ)

- Non-expert group: The hypothesis of equality between two HRTFs can always be rejected, except in the following cases:
  - HRTF1-HRTF2 (KU100 FF 128 - KU100 NF 128)
  - HRTF3-HRTF4-HRTF5 (ST6 128 - ST6 128 EQ - ST6 256 EQ)
  - HRTF6-HRTF7 (KEMAR EQ - pHRTF)

Overall, the results from the expert group are more pronounced. Below are the tables of the test results.

Expert	HRTF 1	HRTF 2	HRTF 3	HRTF 4	HRTF 5	HRTF 6	HRTF 7	HRTF 8	HRTF 9
HRTF 1	1	0.1863	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
HRTF 2	0.1863	1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
HRTF 3	0.0000	0.0000	1	0.8880	0.8006	0.0000	0.0000	0.0001	0.0010
HRTF 4	0.0000	0.0000	0.8880	1	0.8753	0.0000	0.0000	0.0002	0.0028
HRTF 5	0.0000	0.0000	0.8006	0.8753	1	0.0000	0.0000	0.0002	0.0005
HRTF 6	0.0000	0.0000	0.0000	0.0000	0.0000	1	0.0001	0.0000	0.0000
HRTF 7	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	1	0.0000	0.0000
HRTF 8	0.0000	0.0000	0.0001	0.0002	0.0002	0.0000	0.0000	1	0.4151
HRTF 9	0.0000	0.0000	0.0010	0.0028	0.0005	0.0000	0.0000	0.4151	1

Figure 7.1: Wilcoxon Test - Experts.

Non-Expert	HRTF 1	HRTF 2	HRTF 3	HRTF 4	HRTF 5	HRTF 6	HRTF 7	HRTF 8	HRTF 9
HRTF 1	1	0.5781	0.0018	0.0156	0.0037	0.0000	0.0000	0.0000	0.0000
HRTF 2	0.5781	1	0.0160	0.0432	0.0129	0.0000	0.0000	0.0000	0.0000
HRTF 3	0.0018	0.0160	1	0.7544	0.5361	0.0000	0.0000	0.0065	0.0002
HRTF 4	0.0156	0.0432	0.7544	1	0.3547	0.0000	0.0000	0.0092	0.0001
HRTF 5	0.0037	0.0129	0.5361	0.3547	1	0.0000	0.0000	0.0236	0.0003
HRTF 6	0.0000	0.0000	0.0000	0.0000	0.0000	1	0.7380	0.0000	0.0000
HRTF 7	0.0000	0.0000	0.0000	0.0000	0.0000	0.7380	1	0.0000	0.0000
HRTF 8	0.0000	0.0000	0.0065	0.0092	0.0236	0.0000	0.0000	1	0.0221
HRTF 9	0.0000	0.0000	0.0002	0.0001	0.0003	0.0000	0.0000	0.0221	1

Figure 7.2: Wilcoxon Test - Audio Technicians.

Overall	HRTF 1	HRTF 2	HRTF 3	HRTF 4	HRTF 5	HRTF 6	HRTF 7	HRTF 8	HRTF 9
HRTF 1	1	0.1790	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
HRTF 2	0.1790	1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
HRTF 3	0.0000	0.0000	1	0.7598	0.5422	0.0000	0.0000	0.0000	0.0000
HRTF 4	0.0000	0.0000	0.7598	1	0.6966	0.0000	0.0000	0.0000	0.0000
HRTF 5	0.0000	0.0000	0.5422	0.6966	1	0.0000	0.0000	0.0000	0.0000
HRTF 6	0.0000	0.0000	0.0000	0.0000	0.0000	1	0.0049	0.0000	0.0000
HRTF 7	0.0000	0.0000	0.0000	0.0000	0.0000	0.0049	1	0.0000	0.0000
HRTF 8	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1	0.4099
HRTF 9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.4099	1

Figure 7.3: Wilcoxon Test - Overall.

The second non-parametric test was the Umann-Whitney test: the objective of this comparison is to verify whether there are differences between the two subject groups (experts vs. non-experts) while keeping the HRTF constant. Again, all data from individual tracks for each attribute were aggregated. For each HRTF, the distribution of expert subjects was compared with that of non-expert subjects to test the null hypothesis of equality between two non-normal distributions, with continuous data, from independent groups.
Results: For all HRTFs, the hypothesis of equality between the two groups can be rejected, except for:

- HRTF 7 (pHRTF)
- HRTF 9 (pHRTF 256 EQ)

Below are the tables of the test results.

Mann-Whitney Test				
HRTF 1	p-value = 0.013	H = true		
HRTF 2	p-value = 0.010	H = true		
HRTF 3	p-value = 0.000	H = true		
HRTF 4	p-value = 0.000	H = true		
HRTF 5	p-value = 0.000	H = true		
HRTF 6	p-value = 0.000	H = true		
HRTF 7	p-value = 0.054	H = false		
HRTF 8	p-value = 0.000	H = true		
HRTF 9	p-value = 0.069	H = false		

Figure 7.4: Umann-Whitney Test.

General observation: The differences between the HRTFs exhibit a consistent trend across the different subject groups.

So, from the general results, without distinguishing between groups, it is observed that among the 9 HRTFs, the two most preferred were KU100 FF 128 [24] and KU100 NF 128 [25], both taken from the literature. On the other hand, KEMAR 128 EQ FLAT [26] and the Personalized HRTF for each candidate were the least preferred. Equalization generally increased the preference for the Personalized HRTF, both for the 128-sample and 256-sample versions. Regarding the KU100 recorded in ST6, neither equalization nor different sample lengths had a significant impact, as the results between the non-equalized version and the two equalized versions with different lengths showed little variation. This lack of difference between the equalized and non-equalized versions of KU100 ST6 with different sample lengths was also observed in the Wilcoxon test for both groups.

When analyzing the groups separately, these preference trends remain consistent. In particular, in the group of expert sound engineers, KEMAR was almost always preferred over the Personalized HRTF, while KU100 FF 128 was slightly favored over the near-field version, despite the fact that the two KU100 HRTFs taken from the literature did not reject the null hypothesis of equality in the Wilcoxon test. Moreover, the equalized version of the Personalized HRTF was generally preferred, whereas sample length did not produce significant differences.
In the sound technicians group, however, more pronounced and variable differences emerge. Firstly, the Personalized HRTF was slightly preferred over KEMAR. Additionally, the different versions of KU100 ST6 showed slightly higher values compared to the other group, while maintaining the same behavior observed in the Wilcoxon test. However, even in this case, the KU100 FF 128 remained the most preferred.

It is important to note that adding headphone equalization to the Personalized HRTF resulted in an increase across all five evaluated attributes, in both the sound engineers and audio technicians groups. This increase was particularly evident for Sound Color and Objective Sound Quality. Additionally, for KU100 NF 128 and KU100 FF 128, the sound engineers group assigned higher values for Scene Depth and Overall Subjective Quality.

Regarding the KU100 recorded in ST6, the equalized versions were preferred in terms of Localisation Accuracy, while both equalized and non-equalized versions exhibited almost identical values for Externalization. Furthermore, there was a slight preference for the 128-sample version over the 256-sample version in terms of Scene Depth and Localisation Accuracy.

Finally, for the Personalized HRTF, the values between the 128-sample and 256-sample versions varied depending on the attribute and the group. A slight preference for the 256-sample version was observed among sound engineers, whereas the audio technicians group showed a slight preference for the 128-sample version.

In addition to the non-parametric tests, it was also useful to examine the correlations between the various attributes for each HRTF, in order to better understand the relationships between the different perceptual dimensions.The correlation matrix, along with p-values, was used to determine the significance of these correlations. The results provide crucial insights into how different perceptual parameters relate to one another for the tested HRTFs. Significant correlation values ($p < 0.05$) between these parameters reveal important trends across the different HRTFs (for the expert group):

- HRTF1 and HRTF2 show a strong correlation between scene depth (Q4) and overall subjective quality (Q5), indicating that the perception of depth is a key factor in the final evaluation of these HRTFs.
- HRTF3 does not exhibit significant correlations, suggesting that the perceptual parameters do not follow clear patterns or that the evaluations are more variable among the experts.
- HRTF4 and HRTF5 highlight an interconnection between all parameters, suggesting that for these HRTFs, each perceptual dimension directly influences the others. In particular, strong relationships are observed between timbre, externalization, localization accuracy, and scene depth, all elements that contribute to the perceived quality.
- HRTF6, HRTF7, and HRTF8 indicate that overall subjective quality (Q5) is mainly influenced by externalization (Q2), scene depth (Q4), and localization accuracy (Q3), with variations in the weight of each parameter depending on the HRTF.
- HRTF9 shows a relationship between timbre (Q1) and localization accuracy (Q3), suggesting that sound color could influence the perception of spatial position.

These results confirm that the perceived subjective quality of an HRTF is influenced by multiple factors, with a predominant role of scene depth, externalization, and localization accuracy. However, some HRTFs show specific correlations that could influence their choice in practical applications. In particular:

- If the goal is to maximize subjective quality, HRTFs with a strong correlation between Q4-Q5 and Q2-Q5 are preferable.
- For applications where spatial accuracy is crucial, the relationship between Q3 and the other parameters becomes relevant, as seen in HRTF9.
- HRTFs with a strong interconnection between all parameters (HRTF4, HRTF5) may offer a more consistent experience but could also be more sensitive to individual variations among listeners.

The results obtained in the present study can be compared with those reported in the research conducted within the University of York [18]. While the present work focuses on the selection of the HRTF that most faithfully replicates the experience of listening through loudspeakers, as judged by experienced sound engineers and audio technicians, the SADIE II study investigated the preferences of a general audience, identifying three main findings: (1) significant differences in the attribute ratings assigned to different HRTF sets, (2) correlations between some of these attributes, though not particularly high, and (3) a suboptimal perception of individualized measurements compared to the HRTFs of binaural mannequins.

One of the most relevant aspects emerging from the SADIE II study is the general preference for binaural mannequin HRTFs, particularly those of the KU100, despite the fact that they received similar or even lower ratings compared to other HRTFs in relation to specific attributes—making this result quite similar to that of the present study. This finding suggests that perceptual factors not directly considered in the evaluation of attributes may influence the overall preference for a given HRTF. Moreover, methodological differences between measurements conducted on human subjects and those on binaural mannequins have undoubtedly played a fundamental role: the inevitable movement of human participants and the use of larger, more high-performance microphones in mannequins are among the most significant differences. Another possible factor is the simplicity of acquiring personalized HRTFs, as it did not require changes in direction for either the speakers or the subjects.
A particularly noteworthy element is the correlation between externalization and preference, reported in the SADIE II study with a correlation coefficient of 0.46, indicative of a moderate relationship. This data confirms that externalization is a relevant factor in evaluating the quality of an HRTF, while not being the sole determinant of subjective preference. In the present study, when analyzing the group of expert sound engineers, externalization precisely aligns with the highlighted preferences and non-preferences.
One of the most critical aspects emerging from the SADIE II analysis, as well as in the present study, is the suboptimal perception of individualized measurements.

The fact that 81 percent of SADIE participants preferred the KU100 HRTF over their own raises questions about the actual usefulness of HRTF personalization, at least in contexts where perceived audio quality is prioritized over mere localization accuracy. This result is particularly relevant to the present study, as it suggests that individualized HRTFs may not always be the best choice and that the subjective perception of audio quality might be better satisfied by standardized, widely accepted HRTFs.

Finally, the SADIE II study highlights that the choice of HRTF, and consequently personalization, also depends on the context of use. For instance, in video games, localization accuracy is crucial, whereas in cinema, timbral quality is more important. Therefore, analyzing the application context is fundamental to determining the most appropriate HRTF selection.

In line with the findings of [31], which investigated the localization accuracy with non-individual and individual HRTFs comparing static and dynamic reproduction methods, no significant preference for individualized HRTFs was observed in this thesis. The study found that the use of individual HRTFs led to a slight reduction in front-back confusions, but no significant differences were found in horizontal localization accuracy. Similarly, in this study, there was no significant improvement in localization performance with individualized HRTFs. Although it was hypothesized that personalized HRTFs would enhance localization, this effect was marginal and only evident under dynamic reproduction conditions. Furthermore, no significant differences in elevation localization accuracy were found between individual and non-individual HRTFs, which aligns with the findings of the present study.

In this thesis, it is noted that KU100 HRTFs were preferred over customized HRTFs because, as highlighted in a plausibility experiment [32], non-individual HRTFs like those of the Neumann KU100 can produce a plausible sound virtualization, even though it is not perfectly identical to loudspeaker listening. This result suggests that, despite the lack of personalization, non-individual HRTFs can still offer high-quality immersive experiences, similar to those one might achieve in an acoustically controlled environment. Nevertheless, in some studies, such as [33], which explores the impact of non-individualized HRTFs on speech recognition performance in virtual environments with noise, it is highlighted that the use of personalized HRTFs significantly improves localization and speech accuracy, especially in 'cocktail party' scenarios. This result suggests that, although non-personalized HRTFs can have a positive impact, individualized ones lead to decisive improvements in specific acoustic situations.

Chapter 8

Conclusion

This research explored the effects of using Head-Related Transfer Functions (HRTFs) in spatial sound reproduction, with a particular focus on the importance of HRTF selection in professional contexts such as audio monitoring. The results revealed that while personalized HRTFs (pHRTFs) can provide better sound reproduction in individual contexts, their actual applicability heavily depends on the usage conditions and listening environment.

In particular, the comparison between different types of HRTFs, including those measured on artificial head models such as the KU100, demonstrated that HRTFs derived from standardized models like the KU100 are particularly effective in professional audio monitoring scenarios. Although it was expected that pHRTFs would offer superior performance, the results suggest that more traditional models, like the KU100, are better suited for certain applications.

This work also emphasized the importance of post-measurement treatments such as equalization and adaptation in enhancing spatial performance. In fact, proper equalization of HRTFs proved to be a crucial step in optimizing spatial perception, improving the accuracy and consistency of the final result, expecially on the pHRTFs.

The implications of this study are significant for audio professionals. While the adoption of personalized HRTFs remains a valid option, choosing the right model based on the context of use is essential. In professional listening environments, such as music recording or audio production, the use of standardized HRTF models, properly treated, can be equally effective. The effectiveness of these HRTFs is not solely dependent on their origin (measured or artificial), but also on the ability to adapt and calibrate the parameters according to the specific needs of the user and the type of listening environment.

To further validate these findings, the next step will be to expand the listener sample size to better understand individual variations in HRTF perception. Improving the test setup will be crucial, aiming to measure HRTFs in more

complex and complete environments, ensuring HRTFs with more available points and also implementing head tracking. Additionally, a promising area of research lies in analyzing HRTFs obtained from 3D scans and deep learning-based methods to assess whether these approaches can offer a valid and precise alternative to traditional measured HRTFs.
In conclusion, this research has highlighted the importance of carefully selecting HRTFs in professional audio environments, emphasizing that while personalized HRTFs can be useful, more traditional models like the KU100 often provide superior performance in spatial sound reproduction. Making these technologies more accessible and precise will be essential to meet the evolving needs of the audio industry.

Appendix A

Script

PIPELINEperIR.py

```
#!/usr/bin/env python3

import argparse
import os
from pydub import AudioSegment
import numpy as np
import soundfile as sf
from scipy.signal import fftconvolve
import librosa
import sys
#import time

def load_audio(file_path):
    """Carica un file audio e ritorna il segnale e la frequenza di
    campionamento."""
    signal, samplerate = sf.read(file_path)
    # Se il file è stereo, si prende solo un canale
    if len(signal.shape) > 1:
        signal = signal[:, 0]
    return signal, samplerate

# Funzione per tagliare l'audio con numpy
def cut_wav_numpy(audio_data, start_sample, num_samples):
    return audio_data[start_sample:start_sample + num_samples]

# Funzione per processare il file, tagliare le IR e concatenarle
def process_wav_numpy(file_path, segment_duration, offset_seconds,
    num_samples, sample_rate, output_dir):
    """Taglia l'audio, concatena le IR e salva il risultato in un
    singolo file WAV."""
```

```
    # Carica i dati audio
    audio_data, sr = sf.read(file_path)

    if sr != sample_rate:
        raise ValueError(f"Il file ha un sample rate di {sr}, ma ci si aspettava {sample_rate}.")

    total_samples = len(audio_data)
    segment_duration_samples = int(segment_duration * sample_rate)
    offset_samples = int(offset_seconds * sample_rate)

    # Estrae il nome del file senza estensione
    base_filename = os.path.splitext(os.path.basename(file_path))[0]
    if not os.path.exists(output_dir):
        os.makedirs(output_dir)

    all_ir_data = []  # Lista per concatenare tutte le IR

    # Cicla ogni segmento e taglia la porzione richiesta
    for i in range(0, total_samples, segment_duration_samples):
        segment_start = i
        segment_end = segment_start + segment_duration_samples

        if segment_end > total_samples:
            break  # Se eccede la lunghezza del file

        # Definisce il punto di taglio dall'offset
        cut_start = segment_start + offset_samples  # Offset + inizio segmento

        # Verifica che i tagli siano entro i limiti del segmento
        if cut_start + num_samples <= total_samples:
            # Taglia i campioni esatti
            slice_audio = cut_wav_numpy(audio_data, cut_start, num_samples)
            all_ir_data.append(slice_audio)  # Aggiunge la porzione alla lista

            # Salva il risultato come file WAV separato
            output_file = f"{output_dir}/{base_filename}_IR_{i // segment_duration_samples + 1}.wav"
            sf.write(output_file, slice_audio, sample_rate)
            print(f"Salvato: {output_file} - Lunghezza: {len(slice_audio)} campioni")
        else:
            print(f"Taglio fuori limiti per il segmento {i // segment_duration_samples + 1}, skipping.")

    # Concatenazione di tutte le porzioni IR
```

```python
    concatenated_ir_data = np.concatenate(all_ir_data)

    # Salva il risultato finale in un unico file WAV
    concatenated_output_file = f"{output_dir}/{base_filename}_concatenated_IRs.wav"
    sf.write(concatenated_output_file, concatenated_ir_data, sample_rate)
    print(f"Salvato il file concatenato in: {concatenated_output_file}")

    # Restituisce i nomi dei file generati
    return [f"{output_dir}/{base_filename}_IR_{i + 1}.wav" for i in range(len(all_ir_data))]

def rms_db(signal):
    """Calcola il livello RMS in dBFS di un segnale audio."""
    rms = np.sqrt(np.mean(signal**2))
    # Evita logaritmi di zero
    rms_db = 20 * np.log10(rms) if rms > 0 else -np.inf
    return rms_db

def split_stereo_to_mono(input_file):
    """Divide un file stereo in due file mono (sinistro e destro) a 24 bit utilizzando pydub."""

    # Verifica se il file esiste
    if not os.path.exists(input_file):
        raise FileNotFoundError(f"Il file {input_file} non è stato trovato.")

    # Crea i nomi dei file mono per i canali sinistro e destro
    base_name = os.path.splitext(os.path.basename(input_file))[0]
    output_dir = os.path.dirname(input_file)
    left_file = os.path.join(output_dir, f"{base_name}_left.wav")
    right_file = os.path.join(output_dir, f"{base_name}_right.wav")

    # Carica il file stereo come AudioSegment
    stereo_audio = AudioSegment.from_file(input_file, format="wav")

    sample_rate = stereo_audio.frame_rate
    ir_length_in_samples = len(stereo_audio.get_array_of_samples())//2

    # Usa il metodo split_to_mono per separare il file stereo nei due canali mono
    mono_audios = stereo_audio.split_to_mono()

    # Esporta i file mono (sinistro e destro) in formato 24-bit PCM
```

```python
    mono_audios[0].export(left_file, format="wav", parameters=["-acodec", "pcm_s24le"])
    mono_audios[1].export(right_file, format="wav", parameters=["-acodec", "pcm_s24le"])

    print(f"File stereo diviso in:\n- {left_file} (24-bit)\n- {right_file} (24-bit)")

    return left_file, right_file, ir_length_in_samples

def envelope_audio(input_file):
    # Carica il file audio
    audio, samplerate = sf.read(input_file)

    # Verifica che il file audio sia mono (1 canale)
    if audio.ndim != 1:
        raise ValueError("Il file audio deve essere mono per questa funzione.")

    # Durata del fade (5 campioni)
    fade_duration = 10

    # Creazione dell'inviluppo per i primi 5 campioni (da 0 a 1)
    start_fade = np.linspace(0, 1, fade_duration)

    # Creazione dell'inviluppo per gli ultimi 5 campioni (da 1 a 0)
    end_fade = np.linspace(1, 0, fade_duration)

    # Modifica dei primi 5 campioni
    audio[:fade_duration] = audio[:fade_duration] * start_fade

    # Modifica degli ultimi 5 campioni
    audio[-fade_duration:] = audio[-fade_duration:] * end_fade

    # Crea il nome del file di output aggiungendo "_modified" al nome originale
    base_name, ext = os.path.splitext(input_file)
    output_file = f"{base_name}_modified{ext}"

    # Salvataggio del file modificato in 24 bit e con la stessa frequenza di campionamento (48000 Hz)
    sf.write(output_file, audio, samplerate, subtype='PCM_24')

    return output_file  # Ritorna il nome del file di output

def convolve_audio(input_file, fir_file, output_dir):
    # Carica il file audio e il FIR
    input_signal, input_samplerate = sf.read(input_file)
    fir_signal, fir_samplerate = sf.read(fir_file)
```

```

    # Verifica e riscalamento alla frequenza di campionamento di 48 kHz, se necessario
    target_samplerate = 48000
    if input_samplerate != target_samplerate:
        input_signal = librosa.resample(input_signal, orig_sr=input_samplerate, target_sr=target_samplerate)
        input_samplerate = target_samplerate

    if fir_samplerate != target_samplerate:
        fir_signal = librosa.resample(fir_signal, orig_sr=fir_samplerate, target_sr=target_samplerate)
        fir_samplerate = target_samplerate

    # Verifica se l'audio è mono
    if len(input_signal.shape) > 1:
        raise ValueError("L'input deve essere un file audio mono.")

    # Convolvi direttamente
    convolved_signal = fftconvolve(input_signal, fir_signal, mode='full')

    # Crea il nome del file di output
    input_name = os.path.splitext(os.path.basename(input_file))[0]
    fir_name = os.path.splitext(os.path.basename(fir_file))[0]
    output_file = os.path.join(output_dir, f"convolution_{input_name}_{fir_name}.wav")

    # Salva il risultato in un nuovo file audio a 24 bit
    sf.write(output_file, convolved_signal.astype(np.float32), target_samplerate, subtype='PCM_24')
    print(f"Convoluzione completata. Risultato salvato come '{output_file}' con 48 kHz e 24 bit.")

    return output_file

def phase_correlation(file1, file2):
    """Calcola la correlazione di fase tra due file audio e il ritardo."""
    # Carica i segnali audio
    signal1, samplerate1 = load_audio(file1)
    signal2, samplerate2 = load_audio(file2)

    # Assicura che i segnali abbiano la stessa frequenza di campionamento
    if samplerate1 != samplerate2:
        raise ValueError("Entrambi i file devono avere la stessa frequenza di campionamento.")

```

```
    # Trova la correlazione incrociata
    correlation = np.correlate(signal1, signal2, mode='full')
    lag = np.argmax(correlation) - (len(signal2) - 1)

    # Calcola il ritardo in secondi e in campioni
    delay_samples = abs(lag)
    delay_seconds = delay_samples / samplerate1

    # Determina se i segnali sono in fase o sfasati
    in_phase = (lag == 0)

   # Calcola i livelli RMS dei due segnali in dB
    rms_db1 = rms_db(signal1)
    rms_db2 = rms_db(signal2)

    # Calcola la differenza in dB per allineare i livelli RMS
    db_adjustment = rms_db1 - rms_db2

    print(f"Ritardo in campioni: {delay_samples}")
    print(f"Ritardo in secondi: {delay_seconds}")
    print(f"Livello RMS del file1: {rms_db1:.2f} dB")
    print(f"Livello RMS del file2: {rms_db2:.2f} dB")
    print(f"Per uguagliare i livelli RMS, il file2 deve essere regolato di {db_adjustment:.2f} dB")

    return lag, delay_samples, delay_seconds, db_adjustment

def apply_delay_and_volume(input_file, delay_samples, volume_db, output_dir, channel, nameIR):
    """Applica il ritardo e il volume modificato a un file audio, indicando L o R nel nome del file in base al canale."""
    # Carica il segnale audio
    signal, samplerate = load_audio(input_file)

    # Applica il ritardo (shift temporale)
    delayed_signal = np.roll(signal, delay_samples)

    # Applica la modifica del volume (abbassamento di volume in dB)
    volume_factor = 10**(volume_db / 20)
    delayed_signal = delayed_signal * volume_factor

    # Crea il nome del file di output includendo L o R
    input_name = os.path.splitext(os.path.basename(input_file))[0]
    input_name2 = os.path.splitext(os.path.basename(nameIR))[0]
    output_file = os.path.join(output_dir, f"delayed_volume_{input_name}_{channel}_{input_name2}.wav")

    # Salva il risultato modificato
```

```
    sf.write(output_file, delayed_signal.astype(np.float32), samplerate, subtype='PCM_24')
    print(f"Ritardo e volume applicati. Risultato salvato come '{output_file}' con 48 kHz e 24 bit.")

    # Ritorna il segnale modificato
    return delayed_signal

def sum_and_cut(input_file, ir_left_fir, ir_right_fir, adjusted_dirac_signal_left, adjusted_dirac_signal_right, ir_length_in_samples, output_dir, samplerate=48000):
    """Somma le IR filtrate con la Dirac aggiustata, crea un file stereo e lo taglia a metà, aggiungendo un taglio di 128 campioni.
    """

    # Carica i segnali IR filtrati
    ir_left_signal, _ = load_audio(ir_left_fir)
    ir_right_signal, _ = load_audio(ir_right_fir)

    # Allinea le lunghezze degli array aggiungendo zeri (padding) al segnale più corto
    max_length = max(len(ir_left_signal), len(adjusted_dirac_signal_left))
    ir_left_signal = np.pad(ir_left_signal, (0, max_length - len(ir_left_signal)), 'constant')
    adjusted_dirac_signal_left = np.pad(adjusted_dirac_signal_left, (0, max_length - len(adjusted_dirac_signal_left)), 'constant')

    max_length = max(len(ir_right_signal), len(adjusted_dirac_signal_right))
    ir_right_signal = np.pad(ir_right_signal, (0, max_length - len(ir_right_signal)), 'constant')
    adjusted_dirac_signal_right = np.pad(adjusted_dirac_signal_right, (0, max_length - len(adjusted_dirac_signal_right)), 'constant')

    # Somma le IR filtrate con i rispettivi segnali di Dirac aggiustati
    sum_left = ir_left_signal + adjusted_dirac_signal_left
    sum_right = ir_right_signal + adjusted_dirac_signal_right

    # Combina i segnali sinistro e destro in un array stereo
    combined_signal = np.vstack((sum_left, sum_right)).T

    # Calcola la lunghezza del file in campioni
    length_in_samples = combined_signal.shape[0]

    # Trova il punto di metà lunghezza e il punto finale del taglio
    cut_point = (length_in_samples // 2) - (ir_length_in_samples//2)
    cut_end = cut_point + ir_length_in_samples
```

```python

    # Estrai la porzione del file tra il punto di taglio e il punto finale
    cut_signal = combined_signal[int(cut_point):int(cut_end)]

    # Crea il nome del file di output
    input_name = os.path.splitext(os.path.basename(input_file))[0]
    output_file = os.path.join(output_dir, f"combined_adjusted_{input_name}_dirac_FINAL.wav")

    # Salva il risultato come file stereo a 24 bit (sintassi PCM_24)
    sf.write(output_file, cut_signal.astype(np.float32), samplerate, subtype='PCM_24')
    print(f"File combinato e tagliato salvato come '{output_file}'")

    return output_file

def concat_audio_files(input_files, output_dir, name, samples):
    """
    Concatena i file IR stereo e salva il risultato in un unico file di output.

    :param input_files: Lista di percorsi ai file IR da concatenare.
    :param output_dir: Cartella di destinazione dove salvare il file finale.
    :param name: Nome base per il file di output.
    """
    # Lista per contenere i dati audio
    all_data = []

    # Leggi ogni file e aggiungi i dati alla lista
    for input_file in input_files:
        audio_data, samplerate = sf.read(input_file, dtype='float32')

        # Controlla che tutti i file abbiano la stessa frequenza di campionamento
        if samplerate != 48000:
            print(f"Attenzione: il file {input_file} ha una frequenza di campionamento diversa da 48000 Hz.")

        # Verifica che il file sia stereo (due canali)
        if audio_data.ndim != 2 or audio_data.shape[1] != 2:
            print(f"Attenzione: il file {input_file} non è stereo. Verrà ignorato.")
            continue  # Ignora i file non stereo

        all_data.append(audio_data)

    # Concatena i dati audio solo per i file stereo
```

```python
    if all_data:
        concatenated_data = np.concatenate(all_data, axis=0)

        input_name = os.path.splitext(os.path.basename(name))[0]
        output_file = os.path.join(output_dir, f"{input_name}_Final_Concatenated_{samples}.wav")

        # Salva il file concatenato in formato 24-bit e campionamento 48000 Hz
        sf.write(output_file, concatenated_data, 48000, subtype='PCM_24')

        print(f"File concatenato salvato in: {output_dir}")


def main():
    # Definisci gli argomenti della riga di comando
    parser = argparse.ArgumentParser(description="Pipeline di convoluzioni per IR e FIR.")
    parser.add_argument("ir", type=str, help="Path al file  IR")
    parser.add_argument("num_samples", type=int, help="Numero di campioni da mantenere per ogni taglio")
    parser.add_argument("dirac_filtered", type=str, help="Path alla Dirac già filtrata")
    parser.add_argument("fir_hp", type=str, help="Path al filtro FIR HP")
    parser.add_argument("tone", type=str, help="Path al tono")
    parser.add_argument("--log", type=str, help="Path del file di log", default="output_log.txt")

    args = parser.parse_args()

    segment_duration = 3  # Durata di ogni segmento (in secondi)
    offset_seconds = 2.057  # Offset di 2.056 secondi
    sample_rate = 48000  # Sample rate del file

    # Genera la cartella di output utilizzando il nome del file IR
    ir_name = os.path.splitext(os.path.basename(args.ir))[0]
    output_dir = os.path.join(os.getcwd(), ir_name+"_"+str(args.num_samples))

    # Crea la cartella di output se non esiste
    if not os.path.exists(output_dir):
        os.makedirs(output_dir)
        print(f"Cartella di output '{output_dir}' creata.")

    # Reindirizza stdout a un file
    log_path = os.path.join(output_dir, args.log)
    sys.stdout = open(log_path, "w")
```

```

    ir_files = process_wav_numpy(args.ir, segment_duration, offset_seconds, args.num_samples, sample_rate, output_dir)
    ir_final_list = []

    for ir_file in ir_files:

        ir_L, ir_R, ir_length_in_samples = split_stereo_to_mono(ir_file)
        #ir_L2 = envelope_audio(ir_L)
        #ir_R2 = envelope_audio(ir_R)

        # Esegui la convoluzione usando i percorsi specificati dalla riga di comando
        # Convolvi l'IR sinistra con il filtro FIR HP
        ir_left_fir_result = convolve_audio(ir_L, args.fir_hp, output_dir)

        # Convolvi l'IR destra con il filtro FIR HP
        ir_right_fir_result = convolve_audio(ir_R, args.fir_hp, output_dir)

        # Convolvi la Dirac filtrata con il tono
        dirac_tone_result = convolve_audio(args.tone, args.dirac_filtered, output_dir)

        #time.sleep(5)

        # Convolvi il risultato della convoluzione IR sinistra + FIR con il tono
        ir_left_fir_tone_result = convolve_audio(args.tone, ir_left_fir_result, output_dir)

        # Convolvi il risultato della convoluzione IR destra + FIR con il tono
        ir_right_fir_tone_result = convolve_audio(args.tone, ir_right_fir_result, output_dir)

        #time.sleep(5)

        # Calcola la correlazione di fase tra i risultati delle convoluzioni
        print("Correlazione tra IR sinistra convoluta con FIR e tono e Dirac filtrata convoluta con tono:")
        lag_left, delay_samples_left, delay_seconds_left, db_adj_left = phase_correlation(ir_left_fir_tone_result, dirac_tone_result)
        print(f"Ritardo per IR sinistra: {delay_seconds_left:.4f} secondi ({delay_samples_left} campioni)")

```

```
        print("Correlazione tra IR destra convoluta con FIR e tono e
    Dirac filtrata convoluta con tono:")
        lag_right, delay_samples_right, delay_seconds_right,
    db_adj_right = phase_correlation(ir_right_fir_tone_result,
    dirac_tone_result)
        print(f"Ritardo per IR destra: {delay_seconds_right:.4f} secondi
     ({delay_samples_right} campioni)")

        adjusted_dirac_signal_left = apply_delay_and_volume(args.
    dirac_filtered, delay_samples_left, db_adj_left, output_dir, "L",
    ir_file)
        adjusted_dirac_signal_right = apply_delay_and_volume(args.
    dirac_filtered, delay_samples_right, db_adj_right, output_dir, "R"
    , ir_file)
        summed_audio = sum_and_cut(ir_file, ir_left_fir_result,
    ir_right_fir_result, adjusted_dirac_signal_left,
    adjusted_dirac_signal_right, ir_length_in_samples, output_dir)
        ir_final_list.append(summed_audio)

    concat_audio_files(ir_final_list, output_dir, args.ir, args.
    num_samples)


if __name__ == '__main__':
    main()
```

ExportIR.py

```
import os
import numpy as np
import soundfile as sf
import pysofaconventions as sofa

# Percorso del file SOFA
sofa_file_path = r"directory del file sofa"

# Crea la directory di output se non esiste
output_directory = "output_IRs_NF"
if not os.path.exists(output_directory):
    os.makedirs(output_directory)

# Carica il file SOFA
sofa_data = sofa.SOFAFile(sofa_file_path, 'r')

# Verifica se il file è valido
if not sofa_data.isValid():
    print("Il file SOFA non è valido.")
else:
    print("Esportazione delle IR per la configurazione 7.1.4:")

    # Ottiene le posizioni delle sorgenti
    source_positions = sofa_data.getVariableValue('SourcePosition')
    impulse_responses = sofa_data.getVariableValue('Data.IR')

    # Controlla se l'attributo 'DataSamplingRate' esiste
    try:
        sampling_rate = int(sofa_data.getGlobalAttributeValue('DataSamplingRate'))
    except sofa.SOFAError:
        print("L'attributo 'DataSamplingRate' non trovato. Utilizzo di 48000 Hz come valore predefinito.")
        sampling_rate = 48000  # Valore predefinito

    # Definisci le posizioni target della configurazione 7.1.4
    target_positions_7_1_4 = {
        "LEFT": (30, 0),
        "RIGHT": (330, 0),
        "CENTER": (0, 0),
        "LFE": (0, -90),
        "LEFT SURROUND": (110, 0),
        "RIGHT SURROUND": (250, 0),
        "BACK LEFT": (150, 0),
        "BACK RIGHT": (210, 0),
        "UP LEFT": (45, 30),
```

```
        "UP RIGHT": (315, 30),
        "UP BACK LEFT": (135, 30),
        "UP BACK RIGHT": (225, 30)
    }

    tolerance = 4  # Tolleranza in gradi
    exported_files = set()  # Insieme per tenere traccia dei file già esportati

    # Itera attraverso le posizioni delle sorgenti e esporta le IR
    for i, position in enumerate(source_positions):
        azimuth = position[0] % 360  # Normalizza l'azimuth
        elevation = position[1]

        for label, (target_azimuth, target_elevation) in target_positions_7_1_4.items():
            if (abs(azimuth - target_azimuth) <= tolerance and
                abs(elevation - target_elevation) <= tolerance):

                ir_data = impulse_responses[i]

                # Controlla la forma dell'IR
                print(f"IR {i}: forma originale {ir_data.shape}.")

                if ir_data.ndim == 1:  # Se l'IR è mono, duplica il canale
                    ir_data = np.column_stack((ir_data, ir_data))  # Crea un array stereo
                    print(f"IR {i} era mono, convertita in stereo con forma {ir_data.shape}.")
                elif ir_data.ndim == 2:  # Se è già 2D
                    if ir_data.shape[0] > 2:  # Più di 2 canali
                        ir_data = ir_data[:2, :]  # Prendi solo i primi 2 canali
                        print(f"IR {i} ha più di 2 canali, mantenuti solo i primi 2 con forma {ir_data.shape}.")
                    elif ir_data.shape[0] < 2:  # Meno di 2 canali
                        print(f"IR {i} non ha abbastanza canali per l'esportazione, forma: {ir_data.shape}.")
                        continue
                    print(f"IR {i} è già in formato valido con forma {ir_data.shape}.")
                else:  # Gestione di IR non stereo
                    print(f"IR {i} non è in un formato valido per l'esportazione, forma: {ir_data.shape}.")
                    continue

                # Crea un nome di file univoco
                filename = f"{label}_Az{azimuth}_El{elevation}.wav"
```

```
                if filename not in exported_files:
                    exported_files.add(filename)  # Aggiunge il file
    all'insieme
                    # Salva l'IR come file WAV
                    sf.write(os.path.join(output_directory, filename)
    , ir_data.T, sampling_rate)  # Trasponi per soundfile
                    print(f"IR salvata come {filename} con forma {
    ir_data.shape}.")
                else:
                    print(f"Duplicato trovato per {filename}, non
    salvato.")

# Chiude il file SOFA
sofa_data.close()
```

Appendix B

Visualization of Shapiro Test and Correlation Matrix

Shapiro Test:

EXPERT		
HRTF 1 vs HRTF 2	H1 = true, p = 4.4206e-08, W = 0.9491	H2 = true, p = 3.4159e-09, W = 0.93839
HRTF 1 vs HRTF 3	H1 = true, p = 4.4206e-08, W = 0.9491	H2 = true, p = 0.0076119, W = 0.98543
HRTF 1 vs HRTF 4	H1 = true, p = 4.4206e-08, W = 0.9491	H2 = true, p = 0.00025261, W = 0.97716
HRTF 1 vs HRTF 5	H1 = true, p = 4.4206e-08, W = 0.9491	H2 = true, p = 0.00097872, W = 0.9806
HRTF 1 vs HRTF 6	H1 = true, p = 4.4206e-08, W = 0.9491	H2 = true, p = 1.9245e-06, W = 0.96276
HRTF 1 vs HRTF 7	H1 = true, p = 4.4206e-08, W = 0.9491	H2 = true, p = 3.8658e-05, W = 0.97201
HRTF 1 vs HRTF 8	H1 = true, p = 4.4206e-08, W = 0.9491	H2 = true, p = 0.0046289, W = 0.9843
HRTF 1 vs HRTF 9	H1 = true, p = 4.4206e-08, W = 0.9491	H2 = true, p = 0.0032337, W = 0.98347
HRTF 2 vs HRTF 3	H1 = true, p = 3.4159e-09, W = 0.93839	H2 = true, p = 0.0076119, W = 0.98543
HRTF 2 vs HRTF 4	H1 = true, p = 3.4159e-09, W = 0.93839	H2 = true, p = 0.00025261, W = 0.97716
HRTF 2 vs HRTF 5	H1 = true, p = 3.4159e-09, W = 0.93839	H2 = true, p = 0.00097872, W = 0.9806
HRTF 2 vs HRTF 6	H1 = true, p = 3.4159e-09, W = 0.93839	H2 = true, p = 1.9245e-06, W = 0.96276
HRTF 2 vs HRTF 7	H1 = true, p = 3.4159e-09, W = 0.93839	H2 = true, p = 3.8658e-05, W = 0.97201
HRTF 2 vs HRTF 8	H1 = true, p = 3.4159e-09, W = 0.93839	H2 = true, p = 0.0046289, W = 0.9843
HRTF 2 vs HRTF 9	H1 = true, p = 3.4159e-09, W = 0.93839	H2 = true, p = 0.0032337, W = 0.98347
HRTF 3 vs HRTF 4	H1 = true, p = 0.0076119, W = 0.98543	H2 = true, p = 0.00025261, W = 0.97716
HRTF 3 vs HRTF 5	H1 = true, p = 0.0076119, W = 0.98543	H2 = true, p = 0.00097872, W = 0.9806
HRTF 3 vs HRTF 6	H1 = true, p = 0.0076119, W = 0.98543	H2 = true, p = 1.9245e-06, W = 0.96276
HRTF 3 vs HRTF 7	H1 = true, p = 0.0076119, W = 0.98543	H2 = true, p = 3.8658e-05, W = 0.97201
HRTF 3 vs HRTF 8	H1 = true, p = 0.0076119, W = 0.98543	H2 = true, p = 0.0046289, W = 0.9843
HRTF 3 vs HRTF 9	H1 = true, p = 0.0076119, W = 0.98543	H2 = true, p = 0.0032337, W = 0.98347
HRTF 4 vs HRTF 5	H1 = true, p = 0.00025261, W = 0.97716	H2 = true, p = 0.00097872, W = 0.9806
HRTF 4 vs HRTF 6	H1 = true, p = 0.00025261, W = 0.97716	H2 = true, p = 1.9245e-06, W = 0.96276
HRTF 4 vs HRTF 7	H1 = true, p = 0.00025261, W = 0.97716	H2 = true, p = 3.8658e-05, W = 0.97201
HRTF 4 vs HRTF 8	H1 = true, p = 0.00025261, W = 0.97716	H2 = true, p = 0.0046289, W = 0.9843
HRTF 4 vs HRTF 9	H1 = true, p = 0.00025261, W = 0.97716	H2 = true, p = 0.0032337, W = 0.98347
HRTF 5 vs HRTF 6	H1 = true, p = 0.00097872, W = 0.9806	H2 = true, p = 1.9245e-06, W = 0.96276
HRTF 5 vs HRTF 7	H1 = true, p = 0.00097872, W = 0.9806	H2 = true, p = 3.8658e-05, W = 0.97201
HRTF 5 vs HRTF 8	H1 = true, p = 0.00097872, W = 0.9806	H2 = true, p = 0.0046289, W = 0.9843
HRTF 5 vs HRTF 9	H1 = true, p = 0.00097872, W = 0.9806	H2 = true, p = 0.0032337, W = 0.98347
HRTF 6 vs HRTF 7	H1 = true, p = 1.9245e-06, W = 0.96276	H2 = true, p = 3.8658e-05, W = 0.97201
HRTF 6 vs HRTF 8	H1 = true, p = 1.9245e-06, W = 0.96276	H2 = true, p = 0.0046289, W = 0.9843
HRTF 6 vs HRTF 9	H1 = true, p = 1.9245e-06, W = 0.96276	H2 = true, p = 0.0032337, W = 0.98347
HRTF 7 vs HRTF 8	H1 = true, p = 3.8658e-05, W = 0.97201	H2 = true, p = 0.0046289, W = 0.9843
HRTF 7 vs HRTF 9	H1 = true, p = 3.8658e-05, W = 0.97201	H2 = true, p = 0.0032337, W = 0.98347
HRTF 8 vs HRTF 9	H1 = true, p = 0.0046289, W = 0.9843	H2 = true, p = 0.0032337, W = 0.98347

Figure B.1: Shapiro test for Group 1.

NON EXPERT		
HRTF 1 vs HRTF 2	H1 = true, p = 3.4933e-09, W = 0.91634	H2 = true, p = 6.2505e-09, W = 0.92018
HRTF 1 vs HRTF 3	H1 = true, p = 3.4933e-09, W = 0.91634	H2 = true, p = 4.1896e-06, W = 0.95582
HRTF 1 vs HRTF 4	H1 = true, p = 3.4933e-09, W = 0.91634	H2 = true, p = 5.2231e-07, W = 0.95319
HRTF 1 vs HRTF 5	H1 = true, p = 3.4933e-09, W = 0.91634	H2 = true, p = 2.0072e-07, W = 0.9493
HRTF 1 vs HRTF 6	H1 = true, p = 3.4933e-09, W = 0.91634	H2 = true, p = 0.00010517, W = 0.9718
HRTF 1 vs HRTF 7	H1 = true, p = 3.4933e-09, W = 0.91634	H2 = true, p = 6.2029e-05, W = 0.97015
HRTF 1 vs HRTF 8	H1 = true, p = 3.4933e-09, W = 0.91634	H2 = true, p = 2.6641e-05, W = 0.96743
HRTF 1 vs HRTF 9	H1 = true, p = 3.4933e-09, W = 0.91634	H2 = true, p = 6.4468e-05, W = 0.97028
HRTF 2 vs HRTF 3	H1 = true, p = 6.2505e-09, W = 0.92018	H2 = true, p = 4.1896e-06, W = 0.95582
HRTF 2 vs HRTF 4	H1 = true, p = 6.2505e-09, W = 0.92018	H2 = true, p = 5.2231e-07, W = 0.95319
HRTF 2 vs HRTF 5	H1 = true, p = 6.2505e-09, W = 0.92018	H2 = true, p = 2.0072e-07, W = 0.9493
HRTF 2 vs HRTF 6	H1 = true, p = 6.2505e-09, W = 0.92018	H2 = true, p = 0.00010517, W = 0.9718
HRTF 2 vs HRTF 7	H1 = true, p = 6.2505e-09, W = 0.92018	H2 = true, p = 6.2029e-05, W = 0.97015
HRTF 2 vs HRTF 8	H1 = true, p = 6.2505e-09, W = 0.92018	H2 = true, p = 2.6641e-05, W = 0.96743
HRTF 2 vs HRTF 9	H1 = true, p = 6.2505e-09, W = 0.92018	H2 = true, p = 6.4468e-05, W = 0.97028
HRTF 3 vs HRTF 4	H1 = true, p = 4.1896e-06, W = 0.95582	H2 = true, p = 5.2231e-07, W = 0.95319
HRTF 3 vs HRTF 5	H1 = true, p = 4.1896e-06, W = 0.95582	H2 = true, p = 2.0072e-07, W = 0.9493
HRTF 3 vs HRTF 6	H1 = true, p = 4.1896e-06, W = 0.95582	H2 = true, p = 0.00010517, W = 0.9718
HRTF 3 vs HRTF 7	H1 = true, p = 4.1896e-06, W = 0.95582	H2 = true, p = 6.2029e-05, W = 0.97015
HRTF 3 vs HRTF 8	H1 = true, p = 4.1896e-06, W = 0.95582	H2 = true, p = 2.6641e-05, W = 0.96743
HRTF 3 vs HRTF 9	H1 = true, p = 4.1896e-06, W = 0.95582	H2 = true, p = 6.4468e-05, W = 0.97028
HRTF 4 vs HRTF 5	H1 = true, p = 5.2231e-07, W = 0.95319	H2 = true, p = 2.0072e-07, W = 0.9493
HRTF 4 vs HRTF 6	H1 = true, p = 5.2231e-07, W = 0.95319	H2 = true, p = 0.00010517, W = 0.9718
HRTF 4 vs HRTF 7	H1 = true, p = 5.2231e-07, W = 0.95319	H2 = true, p = 6.2029e-05, W = 0.97015
HRTF 4 vs HRTF 8	H1 = true, p = 5.2231e-07, W = 0.95319	H2 = true, p = 2.6641e-05, W = 0.96743
HRTF 4 vs HRTF 9	H1 = true, p = 5.2231e-07, W = 0.95319	H2 = true, p = 6.4468e-05, W = 0.97028
HRTF 5 vs HRTF 6	H1 = true, p = 2.0072e-07, W = 0.9493	H2 = true, p = 0.00010517, W = 0.9718
HRTF 5 vs HRTF 7	H1 = true, p = 2.0072e-07, W = 0.9493	H2 = true, p = 6.2029e-05, W = 0.97015
HRTF 5 vs HRTF 8	H1 = true, p = 2.0072e-07, W = 0.9493	H2 = true, p = 2.6641e-05, W = 0.96743
HRTF 5 vs HRTF 9	H1 = true, p = 2.0072e-07, W = 0.9493	H2 = true, p = 6.4468e-05, W = 0.97028
HRTF 6 vs HRTF 7	H1 = true, p = 0.00010517, W = 0.9718	H2 = true, p = 6.2029e-05, W = 0.97015
HRTF 6 vs HRTF 8	H1 = true, p = 0.00010517, W = 0.9718	H2 = true, p = 2.6641e-05, W = 0.96743
HRTF 6 vs HRTF 9	H1 = true, p = 0.00010517, W = 0.9718	H2 = true, p = 6.4468e-05, W = 0.97028
HRTF 7 vs HRTF 8	H1 = true, p = 6.2029e-05, W = 0.97015	H2 = true, p = 2.6641e-05, W = 0.96743
HRTF 7 vs HRTF 9	H1 = true, p = 6.2029e-05, W = 0.97015	H2 = true, p = 6.4468e-05, W = 0.97028
HRTF 8 vs HRTF 9	H1 = true, p = 2.6641e-05, W = 0.96743	H2 = true, p = 6.4468e-05, W = 0.97028

Figure B.2: Shapiro test for Group 2.

OVERALL

HRTF 1 vs HRTF 2	H1 = true, p = 4.4076e-14, W = 0.93521	H2 = true, p = 1.2101e-14, W = 0.93052
HRTF 1 vs HRTF 3	H1 = true, p = 4.4076e-14, W = 0.93521	H2 = true, p = 2.3343e-08, W = 0.97171
HRTF 1 vs HRTF 4	H1 = true, p = 4.4076e-14, W = 0.93521	H2 = true, p = 8.5819e-09, W = 0.96958
HRTF 1 vs HRTF 5	H1 = true, p = 4.4076e-14, W = 0.93521	H2 = true, p = 2.0614e-08, W = 0.97145
HRTF 1 vs HRTF 6	H1 = true, p = 4.4076e-14, W = 0.93521	H2 = true, p = 3.8725e-08, W = 0.97276
HRTF 1 vs HRTF 7	H1 = true, p = 4.4076e-14, W = 0.93521	H2 = true, p = 9.6086e-08, W = 0.97458
HRTF 1 vs HRTF 8	H1 = true, p = 4.4076e-14, W = 0.93521	H2 = true, p = 2.9734e-06, W = 0.98082
HRTF 1 vs HRTF 9	H1 = true, p = 4.4076e-14, W = 0.93521	H2 = true, p = 1.0308e-06, W = 0.979
HRTF 2 vs HRTF 3	H1 = true, p = 1.2101e-14, W = 0.93052	H2 = true, p = 2.3343e-08, W = 0.97171
HRTF 2 vs HRTF 4	H1 = true, p = 1.2101e-14, W = 0.93052	H2 = true, p = 8.5819e-09, W = 0.96958
HRTF 2 vs HRTF 5	H1 = true, p = 1.2101e-14, W = 0.93052	H2 = true, p = 2.0614e-08, W = 0.97145
HRTF 2 vs HRTF 6	H1 = true, p = 1.2101e-14, W = 0.93052	H2 = true, p = 3.8725e-08, W = 0.97276
HRTF 2 vs HRTF 7	H1 = true, p = 1.2101e-14, W = 0.93052	H2 = true, p = 9.6086e-08, W = 0.97458
HRTF 2 vs HRTF 8	H1 = true, p = 1.2101e-14, W = 0.93052	H2 = true, p = 2.9734e-06, W = 0.98082
HRTF 2 vs HRTF 9	H1 = true, p = 1.2101e-14, W = 0.93052	H2 = true, p = 1.0308e-06, W = 0.979
HRTF 3 vs HRTF 4	H1 = true, p = 2.3343e-08, W = 0.97171	H2 = true, p = 8.5819e-09, W = 0.96958
HRTF 3 vs HRTF 5	H1 = true, p = 2.3343e-08, W = 0.97171	H2 = true, p = 2.0614e-08, W = 0.97145
HRTF 3 vs HRTF 6	H1 = true, p = 2.3343e-08, W = 0.97171	H2 = true, p = 3.8725e-08, W = 0.97276
HRTF 3 vs HRTF 7	H1 = true, p = 2.3343e-08, W = 0.97171	H2 = true, p = 9.6086e-08, W = 0.97458
HRTF 3 vs HRTF 8	H1 = true, p = 2.3343e-08, W = 0.97171	H2 = true, p = 2.9734e-06, W = 0.98082
HRTF 3 vs HRTF 9	H1 = true, p = 2.3343e-08, W = 0.97171	H2 = true, p = 1.0308e-06, W = 0.979
HRTF 4 vs HRTF 5	H1 = true, p = 8.5819e-09, W = 0.96958	H2 = true, p = 2.0614e-08, W = 0.97145
HRTF 4 vs HRTF 6	H1 = true, p = 8.5819e-09, W = 0.96958	H2 = true, p = 3.8725e-08, W = 0.97276
HRTF 4 vs HRTF 7	H1 = true, p = 8.5819e-09, W = 0.96958	H2 = true, p = 9.6086e-08, W = 0.97458
HRTF 4 vs HRTF 8	H1 = true, p = 8.5819e-09, W = 0.96958	H2 = true, p = 2.9734e-06, W = 0.98082
HRTF 4 vs HRTF 9	H1 = true, p = 8.5819e-09, W = 0.96958	H2 = true, p = 1.0308e-06, W = 0.979
HRTF 5 vs HRTF 6	H1 = true, p = 2.0614e-08, W = 0.97145	H2 = true, p = 3.8725e-08, W = 0.97276
HRTF 5 vs HRTF 7	H1 = true, p = 2.0614e-08, W = 0.97145	H2 = true, p = 9.6086e-08, W = 0.97458
HRTF 5 vs HRTF 8	H1 = true, p = 2.0614e-08, W = 0.97145	H2 = true, p = 2.9734e-06, W = 0.98082
HRTF 5 vs HRTF 9	H1 = true, p = 2.0614e-08, W = 0.97145	H2 = true, p = 1.0308e-06, W = 0.979
HRTF 6 vs HRTF 7	H1 = true, p = 3.8725e-08, W = 0.97276	H2 = true, p = 9.6086e-08, W = 0.97458
HRTF 6 vs HRTF 8	H1 = true, p = 3.8725e-08, W = 0.97276	H2 = true, p = 2.9734e-06, W = 0.98082
HRTF 6 vs HRTF 9	H1 = true, p = 3.8725e-08, W = 0.97276	H2 = true, p = 1.0308e-06, W = 0.979
HRTF 7 vs HRTF 8	H1 = true, p = 9.6086e-08, W = 0.97458	H2 = true, p = 2.9734e-06, W = 0.98082
HRTF 7 vs HRTF 9	H1 = true, p = 9.6086e-08, W = 0.97458	H2 = true, p = 1.0308e-06, W = 0.979
HRTF 8 vs HRTF 9	H1 = true, p = 2.9734e-06, W = 0.98082	H2 = true, p = 1.0308e-06, W = 0.979

Figure B.3: Overall Shapiro test.

Correlation Matrix:

Expert	C1_Q1	C1_Q2	C1_Q3	C1_Q4	C1_Q5
C1_Q1	1.00 (p=1.0000)				
C1_Q2	-0.10 (p=0.7915)	1.00 (p=1.0000)			
C1_Q3	0.48 (p=0.1929)	0.54 (p=0.1327)	1.00 (p=1.0000)		
C1_Q4	0.14 (p=0.7213)	0.41 (p=0.2677)	0.11 (p=0.7759)	1.00 (p=1.0000)	
C1_Q5	0.39 (p=0.2961)	0.47 (p=0.2020)	0.58 (p=0.1044)	0.75 (p=0.0191	1.00 (p=1.0000)

Non-expert	C1_Q1	C1_Q2	C1_Q3	C1_Q4	C1_Q5
C1_Q1	1.00 (p=1.0000)				
C1_Q2	0.36 (p=0.3770)	1.00 (p=1.0000)			
C1_Q3	0.82 (p=0.0132)	0.59 (p=0.1239)	1.00 (p=1.0000)		
C1_Q4	0.61 (p=0.1095)	0.59 (p=0.1261)	0.55 (p=0.1558)	1.00 (p=1.0000)	
C1_Q5	0.36 (p=0.3855)	0.75 (p=0.0328)	0.32 (p=0.4472)	0.82 (p=0.0118	1.00 (p=1.0000)

Overall	C1_Q1	C1_Q2	C1_Q3	C1_Q4	C1_Q5
C1_Q1	1.00 (p=1.0000)				
C1_Q2	0.14 (p=0.6018)	1.00 (p=1.0000)			
C1_Q3	0.61 (p=0.0100)	0.53 (p=0.0276)	1.00 (p=1.0000)		
C1_Q4	0.43 (p=0.0881)	0.45 (p=0.0707)	0.29 (p=0.2550)	1.00 (p=1.0000)	
C1_Q5	0.34 (p=0.1760)	0.51 (p=0.0365)	0.50 (p=0.0422)	0.76 (p=0.0004	1.00 (p=1.0000)

Figure B.4: Correlation Matrix for KU100 FF 128.

Expert	C2_Q1	C2_Q2	C2_Q3	C2_Q4	C2_Q5
C2_Q1	1.00 (p=1.0000)				
C2_Q2	0.43 (p=0.2489)	1.00 (p=1.0000)			
C2_Q3	0.09 (p=0.8121)	0.65 (p=0.0599)	1.00 (p=1.0000)		
C2_Q4	-0.09 (p=0.8123)	0.33 (p=0.3911)	0.19 (p=0.6192)	1.00 (p=1.0000)	
C2_Q5	0.11 (p=0.7851)	0.22 (p=0.5626)	-0.22 (p=0.5663)	0.78 (p=0.0125)	1.00 (p=1.0000)

Non-expert	C2_Q1	C2_Q2	C2_Q3	C2_Q4	C2_Q5
C2_Q1	1.00 (p=1.0000)				
C2_Q2	0.56 (p=0.1527)	1.00 (p=1.0000)			
C2_Q3	0.40 (p=0.3293)	0.84 (p=0.0083)	1.00 (p=1.0000)		
C2_Q4	0.41 (p=0.3158)	0.60 (p=0.1162)	0.55 (p=0.1605)	1.00 (p=1.0000)	
C2_Q5	0.71 (p=0.0469)	0.57 (p=0.1444)	0.43 (p=0.2899)	0.64 (p=0.0853)	1.00 (p=1.0000)

Overall	C2_Q1	C2_Q2	C2_Q3	C2_Q4	C2_Q5
C2_Q1	1.00 (p=1.0000)				
C2_Q2	0.50 (p=0.0412)	1.00 (p=1.0000)			
C2_Q3	0.27 (p=0.3018)	0.71 (p=0.0015)	1.00 (p=1.0000)		
C2_Q4	0.13 (p=0.6151)	0.37 (p=0.1485)	0.32 (p=0.2106)	1.00 (p=1.0000)	
C2_Q5	0.38 (p=0.1280)	0.38 (p=0.1290)	0.04 (p=0.8680)	0.70 (p=0.0016)	1.00 (p=1.0000)

Figure B.5: Correlation Matrix for KU100 NF 128.

Expert	C3_Q1	C3_Q2	C3_Q3	C3_Q4	C3_Q5
C3_Q1	1.00 (p=1.0000)				
C3_Q2	0.61 (p=0.0828)	1.00 (p=1.0000)			
C3_Q3	0.38 (p=0.3077)	0.51 (p=0.1567)	1.00 (p=1.0000)		
C3_Q4	0.43 (p=0.2500)	0.54 (p=0.1366)	0.53 (p=0.1389)	1.00 (p=1.0000)	
C3_Q5	0.35 (p=0.3520)	0.21 (p=0.5803)	-0.07 (p=0.8526)	0.04 (p=0.9223)	1.00 (p=1.0000)

Non-expert	C3_Q1	C3_Q2	C3_Q3	C3_Q4	C3_Q5
C3_Q1	1.00 (p=1.0000)				
C3_Q2	0.06 (p=0.8897)	1.00 (p=1.0000)			
C3_Q3	-0.59 (p=0.1246)	0.35 (p=0.3887)	1.00 (p=1.0000)		
C3_Q4	0.03 (p=0.9483)	0.18 (p=0.6700)	0.50 (p=0.2025)	1.00 (p=1.0000)	
C3_Q5	-0.17 (p=0.6849)	-0.03 (p=0.9362)	0.20 (p=0.6372)	0.70 (p=0.0534)	1.00 (p=1.0000)

Overall	C3_Q1	C3_Q2	C3_Q3	C3_Q4	C3_Q5
C3_Q1	1.00 (p=1.0000)				
C3_Q2	0.60 (p=0.0114)	1.00 (p=1.0000)			
C3_Q3	0.30 (p=0.2440)	0.55 (p=0.0231)	1.00 (p=1.0000)		
C3_Q4	0.50 (p=0.0415)	0.65 (p=0.0048)	0.59 (p=0.0131)	1.00 (p=1.0000)	
C3_Q5	0.44 (p=0.0790)	0.55 (p=0.0234)	0.26 (p=0.3205)	0.62 (p=0.0084)	1.00 (p=1.0000)

Figure B.6: Correlation Matrix for KU100 ST6 128.

Expert	C4_Q1	C4_Q2	C4_Q3	C4_Q4	C4_Q5
C4_Q1	1.00 (p=1.0000)				
C4_Q2	0.68 (p=0.0451)	1.00 (p=1.0000)			
C4_Q3	0.66 (p=0.0524)	0.91 (p=0.0007)	1.00 (p=1.0000)		
C4_Q4	0.67 (p=0.0492)	0.66 (p=0.0548)	0.81 (p=0.0086)	1.00 (p=1.0000)	
C4_Q5	0.67 (p=0.0462)	0.31 (p=0.4091)	0.43 (p=0.2538)	0.68 (p=0.0423)	1.00 (p=1.0000)
Non-expert	C4_Q1	C4_Q2	C4_Q3	C4_Q4	C4_Q5
C4_Q1	1.00 (p=1.0000)				
C4_Q2	0.76 (p=0.0301)	1.00 (p=1.0000)			
C4_Q3	0.19 (p=0.6478)	0.61 (p=0.1047)	1.00 (p=1.0000)		
C4_Q4	0.85 (p=0.0077)	0.59 (p=0.1223)	0.02 (p=0.9566)	1.00 (p=1.0000)	
C4_Q5	0.03 (p=0.9394)	0.45 (p=0.2615)	0.27 (p=0.5181)	-0.01 (p=0.9832)	1.00 (p=1.0000)
Overall	C4_Q1	C4_Q2	C4_Q3	C4_Q4	C4_Q5
C4_Q1	1.00 (p=1.0000)				
C4_Q2	0.68 (p=0.0026)	1.00 (p=1.0000)			
C4_Q3	0.49 (p=0.0479)	0.74 (p=0.0007)	1.00 (p=1.0000)		
C4_Q4	0.73 (p=0.0009)	0.69 (p=0.0023)	0.51 (p=0.0371)	1.00 (p=1.0000)	
C4_Q5	0.42 (p=0.0911)	0.43 (p=0.0829)	0.31 (p=0.2200)	0.48 (p=0.0504)	1.00 (p=1.0000)

Figure B.7: Correlation Matrix for KU100 ST6 128 EQ.

Expert	C5_Q1	C5_Q2	C5_Q3	C5_Q4	C5_Q5
C5_Q1	1.00 (p=1.0000)				
C5_Q2	0.91 (p=0.0006)	1.00 (p=1.0000)			
C5_Q3	0.89 (p=0.0011)	0.95 (p=0.0001)	1.00 (p=1.0000)		
C5_Q4	0.69 (p=0.0398)	0.65 (p=0.0606)	0.76 (p=0.0179)	1.00 (p=1.0000)	
C5_Q5	0.95 (p=0.0001)	0.86 (p=0.0032)	0.89 (p=0.0012)	0.77 (p=0.0142)	1.00 (p=1.0000)

Non-expert	C5_Q1	C5_Q2	C5_Q3	C5_Q4	C5_Q5
C5_Q1	1.00 (p=1.0000)				
C5_Q2	0.71 (p=0.0485)	1.00 (p=1.0000)			
C5_Q3	-0.00 (p=0.9939)	0.30 (p=0.4685)	1.00 (p=1.0000)		
C5_Q4	0.40 (p=0.3241)	0.06 (p=0.8868)	0.01 (p=0.9796)	1.00 (p=1.0000)	
C5_Q5	0.54 (p=0.1672)	0.45 (p=0.2638)	-0.43 (p=0.2827)	0.60 (p=0.1183)	1.00 (p=1.0000)

Overall	C5_Q1	C5_Q2	C5_Q3	C5_Q4	C5_Q5
C5_Q1	1.00 (p=1.0000)				
C5_Q2	0.80 (p=0.0001)	1.00 (p=1.0000)			
C5_Q3	0.57 (p=0.0166)	0.79 (p=0.0001)	1.00 (p=1.0000)		
C5_Q4	0.57 (p=0.0161)	0.60 (p=0.0106)	0.57 (p=0.0178)	1.00 (p=1.0000)	
C5_Q5	0.70 (p=0.0017)	0.79 (p=0.0001)	0.52 (p=0.0332)	0.80 (p=0.0001)	1.00 (p=1.0000)

Figure B.8: Correlation Matrix for KU100 ST6 256 EQ.

Expert	C6_Q1	C6_Q2	C6_Q3	C6_Q4	C6_Q5
C6_Q1	1.00 (p=1.0000)				
C6_Q2	0.43 (p=0.2511)	1.00 (p=1.0000)			
C6_Q3	0.52 (p=0.1484)	0.15 (p=0.6948)	1.00 (p=1.0000)		
C6_Q4	0.41 (p=0.2705)	0.64 (p=0.0626)	0.60 (p=0.0887)	1.00 (p=1.0000)	
C6_Q5	0.30 (p=0.4321)	0.69 (p=0.0396)	0.46 (p=0.2167)	0.87 (p=0.0020)	1.00 (p=1.0000)

Non-expert	C6_Q1	C6_Q2	C6_Q3	C6_Q4	C6_Q5
C6_Q1	1.00 (p=1.0000)				
C6_Q2	0.58 (p=0.1344)	1.00 (p=1.0000)			
C6_Q3	0.75 (p=0.0309)	0.06 (p=0.8841)	1.00 (p=1.0000)		
C6_Q4	0.93 (p=0.0007)	0.44 (p=0.2733)	0.76 (p=0.0274)	1.00 (p=1.0000)	
C6_Q5	0.97 (p=0.0001)	0.53 (p=0.1725)	0.80 (p=0.0179)	0.92 (p=0.0014)	1.00 (p=1.0000)

Overall	C6_Q1	C6_Q2	C6_Q3	C6_Q4	C6_Q5
C6_Q1	1.00 (p=1.0000)				
C6_Q2	0.60 (p=0.0107)	1.00 (p=1.0000)			
C6_Q3	0.70 (p=0.0016)	0.21 (p=0.4196)	1.00 (p=1.0000)		
C6_Q4	0.71 (p=0.0013)	0.53 (p=0.0303)	0.69 (p=0.0020)	1.00 (p=1.0000)	
C6_Q5	0.71 (p=0.0013)	0.64 (p=0.0056)	0.67 (p=0.0032)	0.88 (p=0.0000)	1.00 (p=1.0000)

Figure B.9: Correlation Matrix for KEMAR 128 EQ.

Expert	C7_Q1	C7_Q2	C7_Q3	C7_Q4	C7_Q5
C7_Q1	1.00 (p=1.0000)				
C7_Q2	0.67 (p=0.0464)	1.00 (p=1.0000)			
C7_Q3	0.65 (p=0.0588)	0.76 (p=0.0177)	1.00 (p=1.0000)		
C7_Q4	0.72 (p=0.0301)	0.93 (p=0.0003)	0.88 (p=0.0019)	1.00 (p=1.0000)	
C7_Q5	0.60 (p=0.0868)	0.79 (p=0.0121)	0.78 (p=0.0137)	0.92 (p=0.0004)	1.00 (p=1.0000)

Non-expert	C7_Q1	C7_Q2	C7_Q3	C7_Q4	C7_Q5
C7_Q1	1.00 (p=1.0000)				
C7_Q2	0.23 (p=0.5875)	1.00 (p=1.0000)			
C7_Q3	0.09 (p=0.8395)	0.84 (p=0.0093)	1.00 (p=1.0000)		
C7_Q4	0.66 (p=0.0730)	0.84 (p=0.0091)	0.67 (p=0.0704)	1.00 (p=1.0000)	
C7_Q5	0.38 (p=0.3493)	0.63 (p=0.0926)	0.47 (p=0.2452)	0.64 (p=0.0884)	1.00 (p=1.0000)

Overall	C7_Q1	C7_Q2	C7_Q3	C7_Q4	C7_Q5
C7_Q1	1.00 (p=1.0000)				
C7_Q2	0.53 (p=0.0283)	1.00 (p=1.0000)			
C7_Q3	0.44 (p=0.0747)	0.79 (p=0.0002)	1.00 (p=1.0000)		
C7_Q4	0.65 (p=0.0051)	0.89 (p=0.0000)	0.78 (p=0.0002)	1.00 (p=1.0000)	
C7_Q5	0.48 (p=0.0487)	0.71 (p=0.0014)	0.64 (p=0.0056)	0.81 (p=0.0001)	1.00 (p=1.0000)

Figure B.10: Correlation Matrix for pHRTF 128.

Expert	C8_Q1	C8_Q2	C8_Q3	C8_Q4	C8_Q5
C8_Q1	1.00 (p=1.0000)				
C8_Q2	0.35 (p=0.3524)	1.00 (p=1.0000)			
C8_Q3	0.24 (p=0.5332)	0.69 (p=0.0385)	1.00 (p=1.0000)		
C8_Q4	0.29 (p=0.4415)	0.66 (p=0.0536)	0.32 (p=0.4016)	1.00 (p=1.0000)	
C8_Q5	0.46 (p=0.2076)	0.93 (p=0.0003)	0.69 (p=0.0414)	0.74 (p=0.0235)	1.00 (p=1.0000)

Non-expert	C8_Q1	C8_Q2	C8_Q3	C8_Q4	C8_Q5
C8_Q1	1.00 (p=1.0000)				
C8_Q2	-0.10 (p=0.8098)	1.00 (p=1.0000)			
C8_Q3	-0.44 (p=0.2730)	0.38 (p=0.3541)	1.00 (p=1.0000)		
C8_Q4	0.47 (p=0.2445)	0.10 (p=0.8228)	0.54 (p=0.1631)	1.00 (p=1.0000)	
C8_Q5	0.14 (p=0.7444)	0.40 (p=0.3282)	0.74 (p=0.0341)	0.88 (p=0.0038)	1.00 (p=1.0000)

Overall	C8_Q1	C8_Q2	C8_Q3	C8_Q4	C8_Q5
C8_Q1	1.00 (p=1.0000)				
C8_Q2	0.20 (p=0.4450)	1.00 (p=1.0000)			
C8_Q3	-0.05 (p=0.8392)	0.64 (p=0.0053)	1.00 (p=1.0000)		
C8_Q4	0.38 (p=0.1329)	0.48 (p=0.0496)	0.52 (p=0.0329)	1.00 (p=1.0000)	
C8_Q5	0.33 (p=0.2029)	0.83 (p=0.0000)	0.68 (p=0.0024)	0.76 (p=0.0004)	1.00 (p=1.0000)

Figure B.11: Correlation Matrix for pHRTF 128 EQ.

Expert	C9_Q1	C9_Q2	C9_Q3	C9_Q4	C9_Q5
C9_Q1	1.00 (p=1.0000)				
C9_Q2	0.51 (p=0.1647)	1.00 (p=1.0000)			
C9_Q3	0.70 (p=0.0377)	0.68 (p=0.0419)	1.00 (p=1.0000)		
C9_Q4	0.27 (p=0.4820)	0.36 (p=0.3398)	0.53 (p=0.1443)	1.00 (p=1.0000)	
C9_Q5	0.36 (p=0.3432)	0.89 (p=0.0014)	0.71 (p=0.0314)	0.33 (p=0.3894)	1.00 (p=1.0000)

Non-expert	C9_Q1	C9_Q2	C9_Q3	C9_Q4	C9_Q5
C9_Q1	1.00 (p=1.0000)				
C9_Q2	0.49 (p=0.2205)	1.00 (p=1.0000)			
C9_Q3	0.65 (p=0.0823)	0.40 (p=0.3204)	1.00 (p=1.0000)		
C9_Q4	0.70 (p=0.0540)	0.87 (p=0.0049)	0.75 (p=0.0338)	1.00 (p=1.0000)	
C9_Q5	0.53 (p=0.1806)	0.91 (p=0.0017)	0.37 (p=0.3623)	0.78 (p=0.0230)	1.00 (p=1.0000)

Overall	C9_Q1	C9_Q2	C9_Q3	C9_Q4	C9_Q5
C9_Q1	1.00 (p=1.0000)				
C9_Q2	0.45 (p=0.0667)	1.00 (p=1.0000)			
C9_Q3	0.66 (p=0.0041)	0.52 (p=0.0341)	1.00 (p=1.0000)		
C9_Q4	0.53 (p=0.0289)	0.50 (p=0.0397)	0.64 (p=0.0057)	1.00 (p=1.0000)	
C9_Q5	0.41 (p=0.0988)	0.89 (p=0.0000)	0.52 (p=0.0336)	0.48 (p=0.0532)	1.00 (p=1.0000)

Figure B.12: Correlation Matrix for pHRTF 256 EQ.

Appendix C

Declaration of AI Tools Usage

During the writing of this thesis, the OpenAI ChatGPT artificial intelligence model was utilized for textual refinement, including grammar and clarity improvements.

Bibliography

[1] International Telecommunication Union. *Objective assessment of sound quality – Methods for system and component evaluation.* Tech. rep. Recommendation ITU-R BS.2132-0. Geneva, Switzerland: International Telecommunication Union, 2019. URL: `https://www.itu.int/rec/R-REC-BS.2132-0-201906-I/en` (cit. on pp. 2, 39, 44, 45).

[2] International Telecommunication Union. *General methods for the subjective assessment of sound quality.* Tech. rep. Recommendation ITU-R BS.1284-2. Geneva, Switzerland: International Telecommunication Union, 2019. URL: `https://www.itu.int/rec/R-REC-BS.1284-2-201903-I/en` (cit. on pp. 2, 39, 44–46).

[3] AudioLabs. *webMUSHRA: An open-source framework for web-based listening tests.* `https://github.com/audiolabs/webMUSHRA`. Accessed: 2024-01-15. 2016 (cit. on pp. 2, 46).

[4] N. Sayers. «Tips and tricks in performing a systematic review: chapter 4». In: *Br. J. Gen. Pract.* 58.547 (2008), p. 136. DOI: `10.3399/bjgp08X277168`. URL: `https://doi.org/10.3399/bjgp08X277168` (cit. on p. 4).

[5] Michael Vorländer. *Auralization: Fundamentals of Acoustics, Modelling, Simulation, Algorithms and Acoustic Virtual Reality.* Springer Berlin, Heidelberg, 2008 (cit. on pp. 6, 8, 10).

[6] Diana C. Peterson, Vamsi Reddy, Marjorie V. Launico, and Renee N. Hamel. «Neuroanatomy, Auditory Pathway». In: (Oct. 2023) (cit. on p. 6).

[7] International Organization for Standardization. *ISO 226:2023 - Acoustics - Normal equal-loudness-level contours.* 2023 (cit. on p. 8).

[8] John William Strutt Baron Rayleigh. *The theory of sound.* Macmilan, 1896 (cit. on p. 9).

[9] Bosun Xie. *Head-related transfer function and virtual auditory display.* Plantation, FL : J. Ross Publishing, 2013 (cit. on pp. 9, 17).

[10] International Telecommunication Union. *Multichannel sound technology in home and broadcasting applications*. ITU-R BS.2159-7. BS Series Broadcasting service (sound). Feb. 2015 (cit. on pp. 11, 13).

[11] Jürgen Peissig Song Li. «Measurement of Head-Related Transfer Functions: A Review». In: *MDPI Open Access Journal* (2020) (cit. on pp. 14, 15, 17, 20).

[12] Brian D. Simpson Giullaume Andéol. «How, and Why, Does Spatial-Hearing Ability Differ among Listeners? What is the Role of Learning and Multisensory Interactions?» In: (2016) (cit. on p. 14).

[13] Jan-Gerrit Richter. *Fast Measurement of Individual Head-Related Transfer Functions*. Aachener Beiträge zur Akustik. Aachen, Germany: Shaker Verlag, 2019 (cit. on pp. 15, 16, 20).

[14] GRAS Acoustics. *KEMAR*. URL: `https://www.grasacoustics.com/industries/audiology/kemar` (visited on 10/09/2024) (cit. on p. 18).

[15] Neumann. *KU 100 - Dummy Head*. URL: `https://www.neumann.com/en-us/products/microphones/ku-100` (visited on 10/09/2024) (cit. on p. 18).

[16] HEAD Acoustic. *HMS II.3 LN HEC – Fullband artificial head with very low inherent noise and a human-like ear canal*. URL: `https://www.head-acoustics.com/products/artificial-head-binaural-recording/hms-ii3-ln-hec` (visited on 10/09/2024) (cit. on p. 19).

[17] Angelo Farina. «Simultaneous Measurement of Impulse Response and Distortion with a Swept-Sine Technique». In: *Proceedings of the 2000 International Conference on Audio Engineering*. Audio Engineering Society, 2000 (cit. on p. 20).

[18] Cal Armstrong, Lewis Thresh, Damian Murphy, and Gavin Kearney. «A Perceptual Evaluation of Individual and Non-Individual HRTFs: A Case Study of the SADIE II Database». In: *Audio Engineering Society Convention 144*. Audio Engineering Society. Malmö, Sweden, 2018 (cit. on pp. 21, 22, 86).

[19] H. Ziegelwanger, W. Kreuzer, and P. Majdak. *Mesh2HRTF*. URL: `https://github.com/Any2HRTF/Mesh2HRTF/wiki` (cit. on p. 23).

[20] Davir C. Carvalho. «Individualized HRTF Synthesis». `https://github.com/davircarvalho/Individualized_HRTF_Synthesis`. PhD thesis. Universidade Federale De Santa Maria, 2022 (cit. on p. 23).

[21] V.R. Algazi, R.O. Duda, D.M. Thompson, and C. Avendano. *The CIPIC HRTF Database*. 2001. URL: `http://interface.cipic.ucdavis.edu` (cit. on p. 24).

[22] SOFA Conventions. *SOFA (Spatially Oriented Format for Acoustics)*. URL: https://www.sofaconventions.org/mediawiki/index.php/SOFA_(Spatially_Oriented_Format_for_Acoustics) (visited on 10/21/2024) (cit. on pp. 25, 39).

[23] Alexander Lindau, Vera Erbes, Steffen Lepa, Hans-Joachim Maempel, Fabian Brinkmann, and Stefan Weinzierl. *A Spatial Audio Quality Inventory (SAQI)*. https://www.ingentaconnect.com/contentone/dav/aaua/2014/00000100/00000005/art00023?crawler=true. 2014 (cit. on pp. 39, 44, 45).

[24] Henning Kayser and Fabian Brinkmann. *Spherical Far-Field HRIR Compilation of the Neumann KU100*. https://zenodo.org/records/3928297. Zenodo Dataset. 2020 (cit. on pp. 39, 84).

[25] Fabian Brinkmann, Alexander Lindau, and Stefan Weinzierl. *Spherical Near-Field (NF) HRIR Compilation of the Neumann KU100*. URL: https://zenodo.org/records/4297951 (cit. on pp. 39, 84).

[26] Piotr Majdak, Peter Balazs, and Bernhard Laback. *Aachen high-resolution KEMAR HRTF database*. URL: https://sofacoustics.org/data/database/aachen%20(high-resolution%20kemar) (cit. on pp. 39, 84).

[27] Jaakko Pasanen. *AutoEQ: A Tool for Automatic Headphone Equalization*. https://github.com/jaakkopasanen/AutoEq. Accessed: 2025-01-23. 2021 (cit. on p. 41).

[28] International Telecommunication Union. *RECOMMENDATION ITU-R BS.708: Considerations for Selecting Listening Conditions for Assessing the Quality of Sound Programme Material*. Tech. rep. ITU-R BS.708. Accessed: 2025-01-23. International Telecommunication Union, 1990. URL: https://www.itu.int/rec/R-REC-BS.708/en (cit. on p. 43).

[29] International Telecommunication Union. *Recommendation ITU-R BS.1534-3: Method for the subjective assessment of intermediate quality levels of audio systems (MUSHRA)*. Tech. rep. Geneva, Switzerland: International Telecommunication Union, 2015 (cit. on p. 46).

[30] International Telecommunication Union. *Recommendation ITU-R BT.500-13: Methodology for the subjective assessment of the quality of television pictures*. Tech. rep. Geneva, Switzerland: International Telecommunication Union, 2012 (cit. on p. 46).

[31] Josefa Oberem, Jan-Gerrit Richter, Dorothea Setzer, Julia Seibold, Iring Koch, and Janina Fels. «Experiments on localization accuracy with non-individual and individual HRTFs comparing static and dynamic reproduction methods». In: *bioRxiv* (2020). DOI: 10.1101/2020.03.31.011650. URL: https://www.biorxiv.org/content/10.1101/2020.03.31.011650v1.full (cit. on p. 87).

[32] Christoph Porschmann Tim Lubeck. «Investigating the plausibility of non-individual HRTFs in binaural reproduction». In: *Proceedings of the FA 2023 Conference*. 2023. URL: https://dael.euracoustics.org/confs/fa2023/data/articles/000633.pdf (cit. on p. 87).

[33] Arcadio Reyes-Lecuona Maria Cuevas-Rodriguez Daniel Gonzalez-Toledo and Lorenzo Picinali. *Impact of non-individualised head related transfer functions on speech-in-noise performances within a synthesised virtual environment.* 1 April 2021. 2021. URL: https://doi.org/10.1121/10.0004220 (cit. on p. 87).