UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Smart sound control in acoustic sensor networks: a perceptual perspective

Doctoral Thesis

by

Juan Estreder Campos

Supervisors:

Dr. Gema Piñero Sipán
Dr. María de Diego Antón

Valencia, Spain
February, 2022

Abstract

Audio systems have been extensively developed in recent years thanks to the increase of devices with high-performance processors capable of performing more efficient audio processing. In addition, the expansion of wireless communications has given the possibility of implementing networks in which devices can be placed in different locations without physical limitations, unlike wired networks. The combination of these technologies has led to the emergence of Acoustic Sensor Networks (ASN). An ASN is composed of nodes equipped with audio transducers, such as microphones or speakers. In the case of acoustic field monitoring, only acoustic sensors (or microphones) need to be incorporated into the ASN nodes. However, in the case of control applications, the nodes must interact with the acoustic field through loudspeakers.

The ASN can be implemented through low-cost devices, such as Raspberry Pi or commercial mobile devices, capable of managing multiple microphones and loudspeakers and offering good computational capacity. In addition, these devices can communicate through wireless connections, such as Wi-Fi or Bluetooth. This ASN design provides high processing power and flexibility due to the processors and the wireless communications offered by the current mobile devices. Therefore, in this dissertation, an ASN composed of commercial mobile devices connected to wireless speakers through a Bluetooth link is proposed. Additionally, the problem of synchronization between the devices in an ASN is one of the main challenges to be addressed since the audio processing performance is very sensitive to the lack of synchronism. Therefore, a deep analysis of the synchronization problem between commercial devices connected to wireless speakers in an ASN is also carried out. In this regard, one of the main contributions is the analysis of the audio latency of mobile devices when the acoustic nodes in the ASN are comprised of mobile devices communicating with the corresponding loudspeakers through Bluetooth links. A second significant contribution of this dissertation is the implementation of a method to synchronize the different devices of an ASN, together with a study of its limitations. Finally, the proposed method has been introduced in order to implement personal sound zones (PSZ) applications. Therefore, the implementation and analysis of the performance of different audio applications

over an ASN composed of commercial mobile devices and wireless speakers is also a significant contribution in the area of ASN.

In cases where the acoustic environment negatively affects the perception of the audio signal emitted by the ASN loudspeakers, equalization techniques are used with the objective of enhancing the perception threshold of the audio signal. For this purpose, a smart equalization system is defined and implemented in this dissertation. In this regard, psychoacoustic algorithms are employed in order to implement a smart processing based on the human hearing system capable of adapting to changes in the environment, and thus increase the perception threshold of the audio signal dynamically. Therefore, another important contribution of this thesis focuses on the analysis of the spectral masking between two complex sounds. This analysis will allow to calculate the masking threshold of one sound over the other in a more accurate way than the currently used methods. This method is used to implement a perceptual equalization application that aims to improve the perception threshold of the audio signal in presence of ambient noise. To this end, this thesis proposes two different equalization algorithms: 1) pre-equalizing the audio signal so that it is perceived above the ambient noise masking threshold and 2) designing a perceptual control of ambient noise in active noise equalization (ANE) systems, so that the perceived ambient noise level is below the masking threshold of the audio signal. Therefore, the last contribution of this dissertation is the implementation of a perceptual equalization application with the two different embedded equalization algorithms and the analysis of their performance through the testbed carried out in the GTAC-iTEAM laboratory.

Keywords: Acoustic Sensor Networks, synchronization between devices, personal sound zones, psychoacoustics, smart equalization, perceptual experiments.

Resumen

Los sistemas de audio han experimentado un gran desarrollo en los últimos años gracias al aumento de dispositivos con procesadores de alto rendimiento capaces de realizar un procesamiento de audio cada vez más eficiente. Por otra parte, la expansión de las comunicaciones inalámbricas ha permitido implementar redes en las que los dispositivos pueden estar ubicados en diferentes lugares sin limitaciones físicas, a diferencia de las redes cableadas. La combinación de estas tecnologías ha dado lugar a la aparición de las redes de sensores acústicos (o sus siglas en inglés, ASN). Una ASN está compuesta por nodos equipados con transductores de audio, como micrófonos o altavoces. En el caso de la monitorización del campo acústico, sólo es necesario incorporar sensores acústicos (o micrófonos) en los nodos de la ASN. Sin embargo, en el caso de las aplicaciones de control, los nodos deben interactuar con el campo acústico a través de altavoces.

La ASN puede implementarse a través de dispositivos de bajo coste, como Raspberry Pi o dispositivos móviles comerciales, capaces de gestionar varios micrófonos y altavoces y de ofrecer una buena capacidad computacional. Además, estos dispositivos pueden comunicarse mediante conexiones inalámbricas, como Wi-Fi o Bluetooth. Este diseño de ASN proporciona una gran potencia de procesamiento y una gran flexibilidad gracias a los procesadores y las comunicaciones inalámbricas que ofrecen los dispositivos móviles actuales. Por ello, en esta tesis se propone una ASN compuesta por dispositivos móviles comerciales conectados a altavoces inalámbricos a través de un enlace Bluetooth. El problema de la sincronización entre los dispositivos de una ASN es uno de los principales retos a abordar ya que el rendimiento del procesamiento de audio es muy sensible a la falta de sincronismo. Por lo tanto, también se lleva a cabo un análisis profundo del problema de la sincronización entre los dispositivos comerciales conectados a los altavoces inalámbricos en una ASN. En este sentido, una de las principales contribuciones es el análisis de la latencia de audio de los dispositivos móviles cuando los nodos acústicos en la ASN están compuestos por dispositivos móviles que se comunican con los correspondientes altavoces a través de enlaces Bluetooth. Una segunda contribución significativa de esta tesis es la implementación de un método para sincronizar los diferentes dispositivos de una ASN, junto con un estu-

dio de sus limitaciones. Por último, se ha introducido el método propuesto para implementar aplicaciones de zonas de sonido personal (o sus siglas en inglés, PSZ). Por lo tanto, la implementación y el análisis del rendimiento de diferentes aplicaciones de audio sobre una ASN compuesta por dispositivos móviles comerciales y altavoces inalámbricos es también una contribución significativa en el área de las ASN.

En los casos en los que el entorno acústico afecta negativamente a la percepción de la señal de audio emitida por los altavoces del ASN, se utilizan técnicas de ecualización con el objetivo de mejorar la percepción de la señal de audio. Para ello, en esta tesis se define e implementa un sistema de ecualización inteligente. Para ello, se emplean algoritmos psicoacústicos con el fin de implementar un procesamiento inteligente basado en el sistema auditivo humano capaz de adaptarse a los cambios del entorno, y así aumentar la percepción de la señal de audio de forma dinámica. Por ello, otra contribución importante de esta tesis es el análisis del enmascaramiento espectral entre dos sonidos complejos. Este análisis permitirá calcular el umbral de enmascaramiento de un sonido sobre el otro de forma más precisa que los métodos utilizados actualmente. Este método se utiliza para implementar una aplicación de ecualización perceptiva que pretende mejorar la percepción de la señal de audio en presencia de un ruido ambiental. Para ello, esta tesis propone dos algoritmos de ecualización diferentes: 1) la pre-ecualización de la señal de audio para que se perciba por encima del umbral de enmascaramiento del ruido ambiental y 2) diseñar un control de ruido ambiental perceptivo en los sistemas de ecualización activa de ruido (o sus siglas en inglés, ANE), de modo que el nivel de ruido ambiental percibido esté por debajo del umbral de enmascaramiento de la señal de audio. Por lo tanto, la última aportación de esta tesis es la implementación de una aplicación de ecualización perceptiva con los dos diferentes algoritmos de ecualización embebidos y el análisis de su rendimiento a través del banco de pruebas realizado en el laboratorio GTAC-iTEAM.

Palabras clave: Redes de sensores acústicos, sincronización entre dispositivos, zonas de sonido personal, psicoacústica, ecualización inteligente, experimentos perceptuales.

Resum

El sistemes de so han experimentat un gran desenvolupament en els últims anys gràcies a l'augment de dispositius amb processadors d'alt rendiment capaços de realitzar un processament d'àudio cada vegada més eficient. D'altra banda, l'expansió de les comunicacions inalàmbriques ha permès implementar xarxes en les quals els dispositius poden estar situats a diferents llocs sense limitacions físiques, a diferència de les xarxes cablejades. La combinació d'aquestes tecnologies ha donat lloc a l'aparició de les xarxes de sensors acústics (o les seues sigles en anglés, ASN). Una ASN està composta per nodes equipats amb transductors d'àudio, com micròfons o altaveus. En el cas del monitoratge del camp acústic, només cal incorporar sensors acústics (o micròfons) als nodes de l'ASN. No obstant això, en el cas de les aplicacions de control, els nodes han d'interactuar amb el camp acústic a través d'altaveus.

Una ASN pot implementar-se mitjantçant dispositius de baix cost, com ara Raspberry Pi o dispositius mòbils comercials, capaços de gestionar diversos micròfons i altaveus i d'oferir una bona capacitat computacional. A més, aquests dispositius poden comunicar-se a través de connexions inalàmbriques, com Wi-Fi o Bluetooth. Aquest disseny d'ASN proporciona una gran potència de processament i una gran flexibilitat gràcies als processadors i les comunicacions inalàmbriques que ofereixen els dispositius mòbils actuals. Per això, en aquesta tesi es proposa una ASN composta per dispositius mòbils comercials connectats a altaveus inalàmbrics a través d'un enllaç Bluetooth. El problema de la sincronització entre els dispositius d'una ASN és un dels principals reptes a abordar ja que el rendiment del processament d'àudio és molt sensible a la falta de sincronisme. Per tant, també es duu a terme una anàlisi profunda del problema de la sincronització entre els dispositius comercials connectats als altaveus inalàmbrics en una ASN. En aquest sentit, una de les principals contribucions és l'anàlisi de la latència d'àudio dels dispositius mòbils quan els nodes acústics en l'ASN estan compostos per dispositius mòbils que es comuniquen amb els altaveus corresponents mitjançant enllaços Bluetooth. Una segona contribució significativa d'aquesta tesi és la implementació d'un mètode per sincronitzar els diferents dispositius d'una ASN, juntament amb un estudi de les seves limitacions. Finalment, s'ha introduït el mètode proposat per implemen-

tar aplicacions de zones de so personal (o les seues sigles en anglés, PSZ). Per tant, la implementació i l'anàlisi del rendiment de diferents aplicacions d'àudio sobre una ASN composta per dispositius mòbils comercials i altaveus inalàmbrics és també una contribució significativa a l'àrea de les ASN.

En els casos en què l'entorn acústic afecta negativament a la percepció del senyal d'àudio emesa pels altaveus de l'ASN, es fan servir tècniques d'equalització amb l'objectiu de millorar la percepció del senyal d'àudio. En consequència, en aquesta tesi es defineix i s'implementa un sistema d'equalització intel·ligent. Per això, s'utilitzen algoritmes psicoacústics per implementar un processament intel·ligent basat en el sistema auditiu humà capaç d'adaptar-se als canvis de l'entorn, i així augmentar la percepció del senyal d'audio de manera dinàmica. Per aquest motiu, una altra contribució important d'aquesta tesi és l'anàlisi de l'emmascarament espectral entre dos sons complexos. Aquesta anàlisi permetrà calcular el llindar d'emmascarament d'un so sobre l'altre de manera més precisa que els mètodes utilitzats actualment. Aquest mètode s'utilitza per a implementar una aplicació d'equalització perceptiva que pretén millorar la percepció del senyal d'àudio en presència d'un soroll ambiental. Per això, aquesta tesi proposa dos algoritmes d'equalització diferents: 1) la preequalització del senyal d'àudio perquè es percebi per damunt del llindar d'emmascarament del soroll ambiental i 2) dissenyar un control de soroll ambiental perceptiu en els sistemes d'equalització activa de soroll (o les seues sigles en anglés, ANE) de manera que el nivell de soroll ambiental percebut estiga per davall del llindar d'emmascarament del senyal d'àudio. Per tant, l'última aportació d'aquesta tesi és, doncs, la implementació d'una aplicació d'equalització perceptiva amb els dos algoritmes d'equalització embeguts i l'anàlisi del seu rendiment a través del banc de proves realitzat al laboratori GTAC-iTEAM.

Paraules clau: Xarxes de sensors acústics, sincronització entre dispositius, zones de so personal, psicoacústica, equalització intel·ligent, experiments perceptuals.

Acknowledgements

This thesis has been carried out at the Audio and Communications Signal Processing Group (GTAC) from the Institute of Telecommunications and Multimedia Applications (ITEAM) of the Universitat Politècnica de València, Spain. This research has been supported by the grant BES-2016-077899 of the Spanish Ministry of Economy and Competitiveness.

First and foremost I would like to express my gratitude to my supervisors Dr. Gema Piñero and Dr. María de Diego, who have supported me all these years in this thesis through their knowledge, advice, patience and numerous hours of proofreading. This work has been completed thanks to their complete support.

I am very grateful to the evaluators of this manuscript, Felipe Orduña Bustamante of the Universidad Nacional Autónoma de México who has also been a member of the committee, Máximo Cobos Serrano of the Universitat de València and Stefania Cecchi of the Università Politecnica delle Marche di Ancona, for providing me with important reviews and gentle comments that have been very useful for improving the final manuscript. I am also thankful to the members of the committee Germán Ramos Peinado of the Universitat Politècnica de València and José Antonio Belloch Rodríguez of the Universidad Carlos III de Madrid, for spending their time reading my thesis. I deeply appreciate the time the evaluators and the members of the committee spent on my thesis.

I also want to thank Dr. Miguel Ferrer and Dr. Franciso José Martínez for their continuous support and collaboration in this thesis. They have contributed in a very significant way to the development of it with their experience in the different areas covered in this dissertation.

I would like to thank Prof. Vesa Välimäki and Dr. Jussi Rämö, who hosted me at the Aalto University in Finland, for their support during the time I spent working in their team and offer me the possibility to deepen my knowledge in the psychoacoustics field. I would also to thanks them, together with the Communication Acoustics group, for the welcome

I received, which allowed me to have an easier and more comfortable stay.

It is very necessary to recognize here the help of all the people who have volunteered to take the various perceptual tests and measurements contained in this work. I am very grateful to them.

I would like to show my gratitude to the members of the GTAC who have made our workplace a very positive, friendly and comfortable place to work. I would like to thank specially to Laura Fuster, Marian Simarro, Christian Antoñanzas, Pablo Gutiérrez, Enrique Palazón, Vicent Moles, Ariel Álvarez, and also the former members Emanuel Aguilera, Fabián Aguirre, Javier García and Eric Beaucamps.

Now in Spanish, mi más profunda gratitud a mis padres José y Alfonsa y a mi hermana María José por darme todo su apoyo incondicional desde el principio, no lo podría haber logrado sin ellos. A mi abuela Petra por su apoyo y ternura desde la distancia y al resto de mi familia quienes han estado siempre ahí para apoyarme. Y por último, mi más sincera gratitud a mis amigos de la facultad, Torrent, La Solana y Villarobledo que me han hecho disfrutar de cada momento todos estos años, lo que me ha ayudado a terminar esta tesis con mayor motivación.

Juan Estreder Campos
February, 2022

Contents

List of Figures

List of Tables

List of symbols

The following list contains those symbols and abbreviations that appear in more than one place in the text:

$\mathbf{X}$	Matrix
$\mathbf{x}$	Vector
x	Scalar
$(\cdot)^{\mathrm{T}}$	Transpose
$(\cdot)^{\mathrm{H}}$	Conjugated transpose
$(\cdot)^{*}$	Complex conjugation
$(\cdot)^{-1}$	Inverse
$\widetilde{(\cdot)}$	Estimation
$*$	Discrete convolution
i	Microphone index
j	Speaker index
k	Frequency index
n	Discrete time sample
N	Number of acoustic nodes
J	Number of speakers
M	Number of microphones
N_{FFT}	FFT size
F	Loudness in sones
L_{F}	Loudness in phones
$y(n)$	Recorded signal
$s(n)$	Audio signal
$c(n)$	Electro-acoustic path
ξ	Smoothing constant of the EMA
f_{s}	Sample rate

For the psychoacoustic chapter:

$x(n)$	Noise signal
$q(n)$	Signal to reproduce
$u(n)$	Recorded audio signal
$r(n)$	Recorded noise signal

$d(n)$	Recorded noise signal from the primary source
$w(n)$	Adaptive filter
$e(n)$	Error signal
$e'(n)$	Pseudo-error signal
m	Frame index
ν	Critical band index
N_{c}	Number of critical bands
$(\cdot)^{\mathrm{dB}}$	dB units
$T_x(\nu)$	Masking threshold of the signal $x(n)$ in dB units
$S_x(\nu)$	Overall masking curve of the signal $x(n)$ in dB units
$O_x(\nu)$	Offset of the signal $x(n)$ in dB units
$E_x(\nu)$	Energy of the signal $x(n)$
$P_x(k)$	Power spectrum of the signal $x(n)$
M_{S}	Frame length
$\Delta_\nu(\eta)$	Difference between critical bands
$B_\nu(\eta)$	Masking pattern in dB units
$b_\nu(\eta)$	Masking pattern considering energy of the signal
α	Factor to control the "Additivity of masking"
μ_x	Tonal Factor of the signal $x(n)$
Υ_x	Spectral flatness measure of the signal $x(n)$
W_{T}	Tonal weighting factor
W_{F}	Loudness weighting factor
λ	Neighboring frequency components
T_{H}	Threshold of tonal component
l	Tonal component index
z	Relevant tonal component index
C_{L}	Number of tonal components
C_{Z}	Number of relevant tonal components
ΔL	SPL Excess
$L_{\mathrm{E}\gamma}(f)$	Excitation level
ρ	Terhardt masking pattern
I_{N}	Noise intensity
L_{TH}	Human hearing threshold
$g(\nu)$	Gain levels per critical band in dB units
$g^{\mathrm{sm}}(\nu)$	Smoothed gain levels per critical band in dB units
$H(z)$	Global response of the graphic equalizer filter
$H_\nu(z)$	Response of the ν^{th} filter of the graphic equalizer
$g_{\mathrm{opt}}^{\mathrm{dB}}(\nu)$	Optimal gains per critical band in dB units

$Q(\nu)$	Quality Factor of $H_\nu(z)$
$\varphi_\mathrm{c}(\nu)$	Center frequency of $H_\nu(z)$
$\varphi_\mathrm{u}(\nu)$	Upper frequency of $H_\nu(z)$
$\varphi_\mathrm{l}(\nu)$	Low frequency of $H_\nu(z)$
$\mathbf{A}$	Interaction matrix
g_p	Prototype gain
k_1	Noise frequency index
K	Number of components of the noise
β_{k_1}	Attenuation levels per noise frequency
o	Convergence factor of the adaptive algorithm

For the synchronization chapter:

T_AL	Audio latency in time units
T_DP	Device processing latency in time units
T_OLD	Output latency at the mobile device side in time units
T_IL	Input latency in time units
T_BL	Bluetooth link latency in time units
T_SP	Speaker processing latency in time units
T_A	Acoustic latency in time units
p_AL	Audio latency in samples unit
D_ms	Distance between microphone and speaker in meters
L_B	Buffer size
$L_\mathrm{B,ref}$	Minimum buffer size in Android
$\arg\max$	Argument of the maximum of a set
$\overline{(\cdot)}$	Mean value
$\|\cdot\|$	Euclidean norm
σ	Standard deviation
L_S	Length of the sweep signal in samples unit
L_W	Length of the warm-up in samples unit
T_O	Clock offset
τ_ACK	Transmission time of the ACK frame
τ_R	Transmission time of a frame
T_C	Timestamp of the Client
T_S	Timestamp of the Server
T_L	Local timestamp
T_G	Global timestamp
Δ_M	Waiting time of the Master node
Δ_S	Waiting time of the Slave node

Δ_{T}	Difference between different T_{AL}
N_{M}	Reproduced samples during the processing in Master node
N_{S}	Reproduced samples during the processing in Slave node
L_{Z}	Sweep gap in samples unit
Δp	Difference between different p_{AL}
L_{c}	Impulse response length
$h(n)$	Inverse filter of $c(n)$
L_{h}	Length of $h(n)$
β	Regularization parameter
$\mathbf{I}$	Identity matrix
$v(n)$	Filtered signal

Abbreviations and Acronyms

A2DP	Advanced Audio Distribution Profile
ACK	Acknowledgment frame
ALC	Audio Latency Compensation
ANC	Active Noise Control
ANE	Active Noise Equalizer
ASN	Acoustic Sensor Network
BASN	Body Acoustic Sensor Network
EMA	Exponential Moving Average
FFT	Fourier Fast Transform
FIR	Finite Impulse Response
HT Profile	Hearing Threshold Profile
IEEE	Institute of Electrical and Electronics Engineers
IIR	Infinite Impulse Response
IP	Internet Protocol
ISO	International Organization for Standardization
ITU	International Telecommunication Union
LMS	Least Mean Square
MeFxLMS	Multiple-error filtered-X LMS
MLS	Maximum Length Sequence
MRSS	Multi-Room Speaker System
NM Profile	Noise Masked Profile
NTP	Network Time Protocol
NTS	Node Task Synchronization
OS	Operating System
PEQ Profile	Perceptual Equalization Profile
PESQ	Perceptual Evaluation of Speech Quality
PM	Pressure Matching
PS	Power Spectrum
PSZ	Personal Sound Zones
PTP	Precision Time Protocol
RIR	Room Impulse Response
SFM	Spectral Flatness Measure
SN	Sensor Network

SNR	Signal-to-Noise Ratio
SPL	Sound Pressure Level
SRO	Sample Rate Offset
STP	Synchronization Time Protocol
SYNC	Synchronized Sequence Number
TCP	Transmission Control Protocol
UAS Profile	Unmasked Audio Signal Profile
UDP	User Datagram Protocol
WASN	Wireless Acoustic Sensor Network
WSN	Wireless Sensor Network
XCT	Cross-Talk Canceller

Introduction 1

1.1 Motivation

In the last decades, audio applications have undergone a far-reaching transformation due to progress in certain areas, such as data communication or audio processing. A wide variety of applications have been implemented allowing, for example, the localization of acoustic sources or the rendering of a specific acoustic field in some positions. To do this, audio applications must have the capability of recording and/or reproducing sound signals through microphones and speakers. In addition, a computing processor is required to analyze the recorded signals and generate the signals fed to the loudspeakers, if necessary. Furthermore, if the necessary hardware to make communication links is available, the information can be exchanged, resulting in a network composed by microphones and speakers. This structure is known as acoustic sensor network (ASN), which can be composed of several acoustic nodes, being an acoustic node the combination of a set of microphones, speakers and a processor.

In general, the ASNs focus on the estimation of a common signal or parameter that is measured by some nodes, or on the estimation of specific signals for each node. The information from each node can be exchanged

with the others, and thus implement different applications. Audio applications aiming at environment monitoring [1] or audio event classification [2] only require microphones at each node. However, there are other applications that require sound generation through the loudspeakers of the nodes, as it is illustrated in Figure 1.1. This figure shows how some parameters are exchanged between nodes in order to generate a specific acoustic field through the loudspeakers in the control zone.

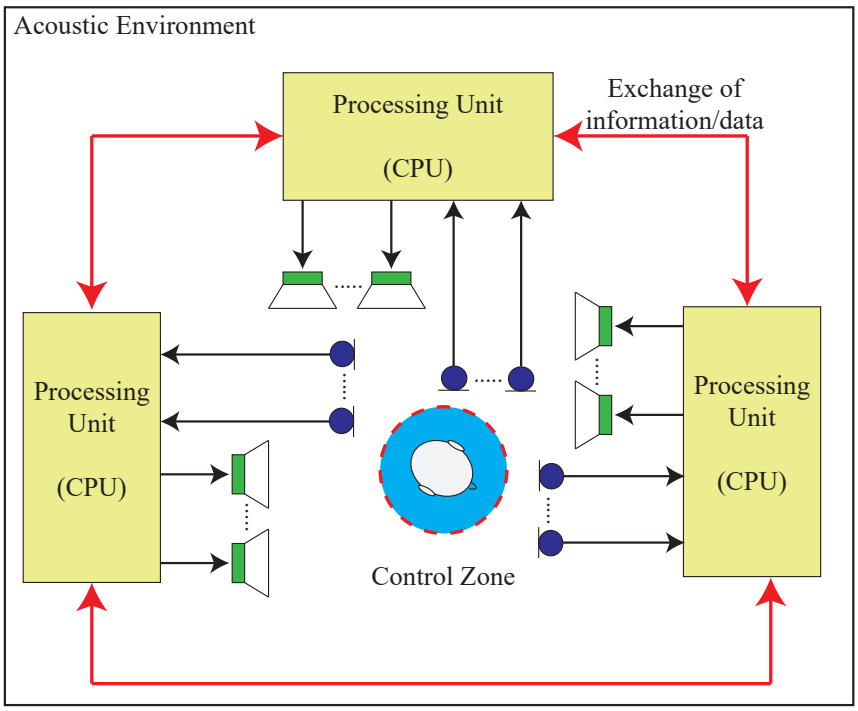


Figure 1.1. ASN composed by multiple acoustic nodes to perform different sound-field applications.

Figure 1.1 shows an ASN where the acoustic nodes include microphones and loudspeakers. This configuration allows the development of more complex audio applications, such as *beamforming* for speech enhancement in noisy environments [3], active noise control [4] or, personal sound zone (PSZ) applications [5]. Since these applications modify the sound field, they are referred to as sound-field applications in this dissertation.

In this regard, this thesis addresses the study of two sound-field applications over ASN: a perceptual equalization based on active noise control techniques and a PSZ application. These applications will be able to implement intelligent algorithms over the ASN, based on different network parameters and environmental information. That is, through this "smart sound processing", the sound-field applications will be able to dynamically adapt to the environment in order to render the corresponding sound in the specific zone.

Reproduction of music signals through speakers in a hostile environment can be annoying due to ambient noise. Since ambient noise can mask some frequency bands of the music signal [6], the perception threshold of the music signal can be seriously degraded, causing an inappropriate user experience. To prevent this result, equalization is required to correct the affected frequency bands. However, because both signals can have non-stationary behavior, an adaptive equalization is needed in order to dynamically adapt the music reproduction at each time. In addition, the study of the masking effect can provide a smart processing and consequently an efficient equalization because it perceptually models the interference between different sounds. Perceptual equalization has been used previously for the same purpose but using headphones [7]. As far as we know, no previous studies have been carried out using loudspeakers. In addition, those works only perform equalization over the music signal. This thesis introduces an adaptive equalization system based on active noise control (ANC) techniques [4] that equalizes the ambient noise instead of the music signal.

PSZ applications, where the objective is the reproduction of some audio signal in a specific area [5], make possible the reproduction of binaural audio through loudspeakers. These applications can be implemented in a wireless ASN, increasing the flexibility since the elements can be placed in different locations without physical restrictions, unlike the wired ASN. Due to this flexibility, portable low-cost audio devices can be a good alternative to perform sound-field applications. Due to the great development of the current mobile devices, they can fulfill the role of performing reproduction applications [8, 9]. Therefore, this dissertation proposes a wireless ASN composed of mobile devices in order to perform PSZ applications. To this end, it is first necessary to address the issue of synchronization [10]. When different devices in an ASN are not properly synchronized, they are not

able to simultaneously perform the corresponding sound-field application, resulting in a poor user experience.

1.2 Objectives

Considering the previous aspects, this thesis is focused on the development of different sound-field applications over ASNs capable of performing smart audio processing to adapt to the specific environment. Therefore, in order to achieve this purpose, the following objectives should be met:

- To study and analyze the algorithms proposed for the sound-field applications to examine their feasibility when being implemented into an ASN.
- To review the effect of the interaction between two simultaneously generated audio signals. Since noise signals (or ambient noise) can be present in the same environment as the signal to be perceived (or target signal), the perception threshold of the target signal can be degraded. Therefore, perceptual algorithms must be considered in order to improve the perception threshold of the target signal in presence of ambient noise. To this end, the masking threshold algorithm is studied and implemented, since it models the way two signals interact psychoacoustically.
- To implement the perceptual equalization application using the masking threshold algorithm. This includes the implementation of two different approaches: a perceptual equalization on the target signal and a perceptual equalization on the ambient noise.
- To evaluate the different approaches of the perceptual equalization application through different perceptual tests carried out by several participants in order to analyze the performance of each approach.
- To implement a network composed of low-cost devices in order to examine the performance of sound-field applications. This includes a review of the corresponding algorithms in terms of computational cost and processing time when they are implemented on low-cost devices.

- To provide a full study of the sound-field reproduction applications over an ASN. The problem of performing accurate synchronization between different acoustic nodes should be analyzed since perfect control of the reproduction and recording of the acoustic nodes is required to produce an optimal result. Therefore, an in-depth study of the synchronization between acoustic nodes must be carried out before implementing the corresponding sound-field reproduction application.

1.3 Organization of the thesis

This thesis describes the research that has been undertaken to develop the previous objectives. The chapters are organized and presented as follows:

- **Chapter 2**: This chapter describes the basic concepts necessary for a full understanding of this dissertation. It contains an introduction to acoustic networks, a brief explanation of current mobile devices and their impact on these networks and finally the fundamental principles of the psychoacoustic theory used in the dissertation.

- **Chapter 3**: This chapter is focused on the psychoacoustic theory. The chapter includes an in-depth explanation of the algorithm for estimating the masking of a signal step by step. In the last part, a perceptual equalization application is presented. It is focused on equalizing sounds based on the masking algorithm presented in the chapter. In addition, this application is evaluated by perceptual experiments in order to study the performance of the masking algorithm.

- **Chapter 4**: This chapter presents a wireless network composed of mobile devices in order to perform an in-depth study about the synchronization between these devices acting as acoustic nodes of an ASN. For this purpose, first a detailed description about the audio latency of the mobile devices is presented. Then, two complementary methods are proposed in order to address accurately the synchronization problem. Finally, a sound-field reproduction application is implemented in order to analyze the performance achieved when implementing the two proposed methods.

- **Chapter 5**: Finally, the conclusions obtained throughout the dissertation are presented. In addition, a list of the publications related to this thesis and the list of the projects that have financially supported this dissertation are also included.
- **Appendix A**: The first appendix shows the structure of a software implemented on mobile devices to address the synchronization and implement the sound-field reproduction application.
- **Appendix B**: In the second appendix, a method of searching for relevant information from ASN acoustic nodes based on Android OS is described.
- **Appendix C**: In the last appendix, a brief study is carried out on the optimal buffer size to be used in the ASN composed of mobile devices and wireless speakers.

State of the Art 2

This chapter presents a brief description of the main concepts necessary to fully understand the methodology used in this dissertation. In the first section, acoustic sensor networks (ASN) are explained, as well as their main design challenges. Afterwards, a brief overview of mobile devices is included, since they are considered in this dissertation as acoustic sensors of an ASN. Subsequent to the introduction of mobile devices, the main wireless communication protocols are described in order to understand the current available options to form a wireless network of acoustic nodes. Finally, since the subjective perception of the audio generated in ASNs is also evaluated, the fundamental psychoacoustics theory is explained.

2.1 Acoustic Sensor Networks

2.1.1 Sensor Networks Overview

Over the past decades, technology has focused on reducing the size of components (such as device processors) and increasing their computational capability [11] in order to improve the performance of the applications. The

development that some areas (such as the computational and the sensing areas) have experienced, along with the improvements in communications and power consumption methods, have led to a technology capable of processing and exchanging a large amounts of data [12]. According to [13], the areas mentioned above are key areas for the sensor networks (SN) technology, where a sensor denotes any component for measuring certain properties of the environment where the network is located and a processor capable of processing the captured data by the sensing hardware [14].

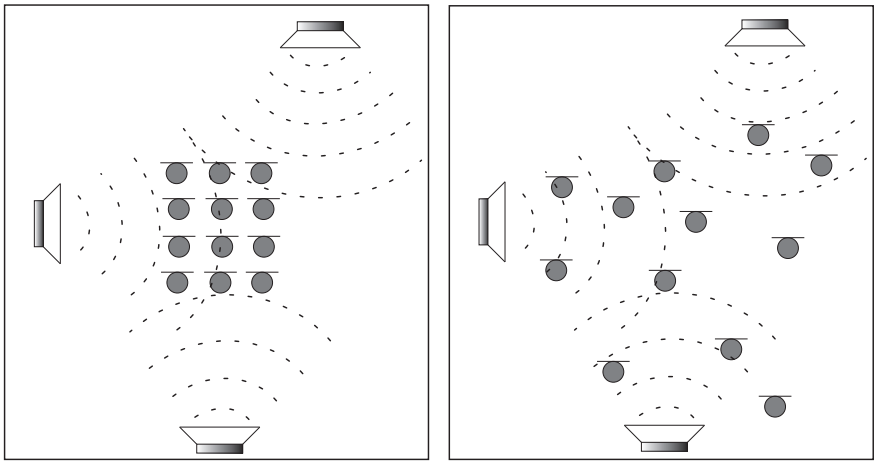
The development of SNs has grown exponentially over the last years because of their use different areas. Depending on the type sensors used, a SN can be able to measure the temperature to detect a fire in a forest [15], monitor the air conditioning system in a building [16], or to monitor the health of a person [17, 18], with the body being the element that delimits the network, resulting in what is known as body area sensor network (BASN). The SN technology can be used in the military field as well [19], where wireless SN (WSN) are used to monitor the enemy forces areas, or in [20], where the main goal is to improve the response to asymmetric threats.

Regarding the previous literature, most of the current applications implemented over SNs use wireless connections to connect their sensors, such as shown in [21, 22], where the tracking and maintenance of industrial equipment is performed. As it is explained in [23], thanks to the wireless connections in SN, larger areas can be covered using same number of sensors compared to traditional wired SN as shown in Figure 2.1, where the sensors of the network are represented by the microphones. Therefore, the WSN provide a greater flexibility and a lower computational cost [24].

Because both types of sensor network (shown in Figure 2.1) only differ in the connection between the sensors, for the sake of simplicity, in this dissertation a sensor network will be denoted as SN, regardless of the connection between sensors used (wireless or wired).

2.1.2 Passive Acoustic Networks

When a network is composed by acoustic sensors, the network is known as Acoustic SN (ASN). These sensors are represented by microphones (as illustrated in Figure 2.1), where acoustic environment can be captured without altering it, resulting in a network capable of obtaining features of the environment through the recorded audio. Since these ASNs are able to get



(a) **(b)**

Figure 2.1. Area covered in a SN (Left) and in a WSN (Right) [10].

parameters from the audio signal, but they cannot modify the sound field, we call them as passive ASN.

Therefore, the main objective of the passive ASNs is to sense the acoustic environment. In this way, they can also obtain useful parameters to characterize that scenario, such as the room impulse response (RIR), which measures the propagation path between two points separated in space [25]. Since ASN are able to obtain useful information about the environment, they are used to solve different issues, such as localization or tracking of targets [26, 27].

In recent years, ASN composed only by microphones have been widely used in different applications. The most common applications are focused on the monitoring, detection and classification of acoustic events. Some examples of monitoring applications are found in [28], where the main goal is to control the state of volcanic eruptions, in [29], where an ASN is used to supervise vehicle traffic, or in [30], where important environmental factors affecting to the people are monitored. Regarding the second group of applications, ASNs are used to detect audible events, such as the gunfire of a sniper [31] or to detect audio events in general (such as opening a door or footsteps) [32, 33], or to classify audio events [2].

Considering the previous literature, passive ASN can be assumed to be formed by a set of microphones capable of exchanging information (using wireless or wired connections) in order to obtain key parameters of the area where the ASN is located. The monitored environment is usually associated with offices, homes or outdoor scenarios, although the deployment of ASN in underwater scenarios has been considered for different applications as well [34]. However they cannot be able to modify the acoustic field.

2.1.3 Active Acoustic Networks

When an ASN is composed of microphones and speakers, it is able not only to monitor the acoustic field but also to modify it. For this reason, in this dissertation these ASNs are called as active ASNs. An example is depicted in Figure 2.1, where loudspeakers are considered as a new element of the acoustic network. Through the introduction of the speakers, the user experience can be improved since the ASN is able to perform more complex tasks beyond monitoring, classification or detection acoustic events.

Nowadays, active ASN are used in different applications that adapt the acoustical environment to some type of requirement. For example, as shown in Figure 2.2, an ASN can be used to actively control ambient noise at certain locations of the ASN through the loudspeakers [4, 35], or to enhance speech using *beamforming* processing techniques [3, 36]. The most popular audio applications implemented over ASNs are the generation of personal sound zones (PSZ), where the objective is to render different sounds in different acoustic zones of the same acoustic environment [37, 38], as shown by Figure 2.2.

In this dissertation an acoustic node is defined as a group of microphones, loudspeakers and a processing unit (CPU) that allows the processing and exchange of information between other nodes. According to this description, an acoustic node can be represented through the model depicted in Figure 2.3, where the left picture is a simplified version that contains only one microphone and a loudspeaker, which will be denoted as single-channel node.

Due to the great development of the ASN technology, these acoustic nodes can include from high-quality microphones and speakers [39] to low-cost devices, such a Rasberry Phi [40]. Therefore, the ASN presents a very flexible structure that can be configured in different ways.

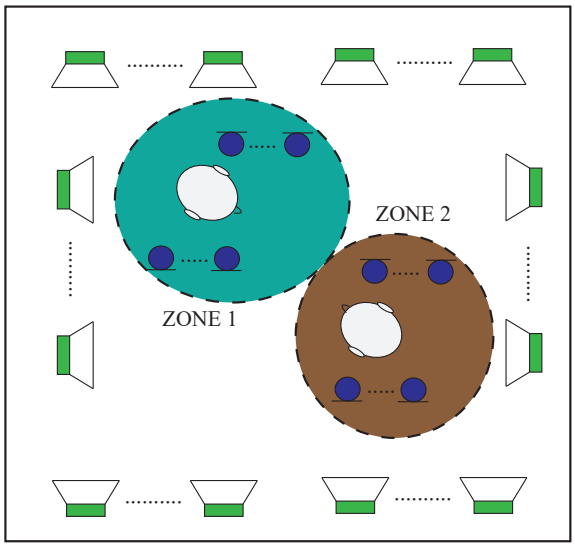


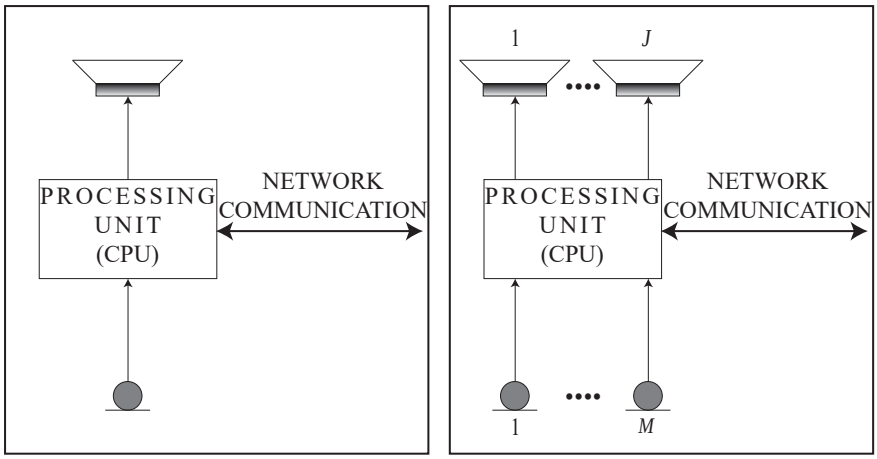
Figure 2.2. Scheme of an arbitrary ASN composed by microphones and speakers.

2.1.4 ASN design challenges

Since different technologies are involved in ASN, an accurate implementation is necessary to take full advantage of the network. To achieve this purpose, different aspects of the network must be considered, as detailed in [1, 10, 41], where an analysis of the ASN design is done. The most important aspects for this dissertation are:

- Topology: It defines the structure of the underlying communication network. The most common are shown in Figure 2.4 (star, fully connected, ...). The choice of the topology depends on several factors, as it is explained in [42], but two main approaches can be taken: either establish the topology and perform the algorithms based on that topology [43], or choose the topology according to physical parameters, such as the impulse responses of the system related with the positions of the nodes [44].

- Processing computation: This feature concerns the approaches (or strategies) to perform the algorithms inside the network. Two strate-



(a) Node with one speaker and one microphone.

(b) Node with J speakers and M microphones.

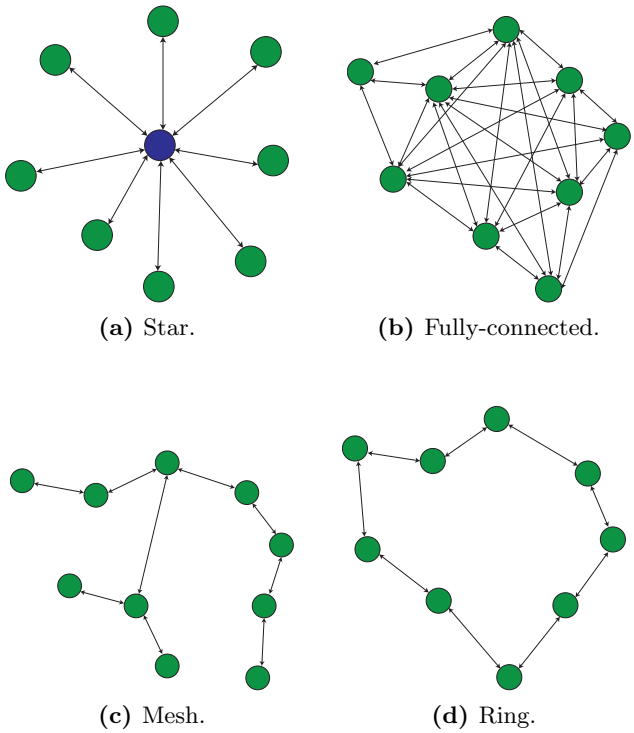
Figure 2.3. Model of acoustic node.

gies are explained in [45, 46, 47], centralized and distributed processing. The first strategy aims to use a central node to process all the data in the network, as the blue node in Figure 2.4a represents. The second strategy focuses on sharing the computational cost of the algorithms among the nodes. This latter strategy will be more efficient in networks with a topology such as shown in Figure 2.4b. Therefore, the approach to process data will be very related with the chosen topology.

- Scalability: This feature is related to the complexity of the algorithms as the network grows. Another goal is to design an algorithm able to scale when the number of the nodes of the network increases [10], meaning that the introduction of more nodes is not a significant drawback.

- Synchronization: Since each node has its own hardware and software, simultaneous procedures in different nodes are difficult to implement. Therefore, in order to address the problem, synchronization mechanisms must be performed [48, 49].

- Minimizing input-output delay: In order to perform real-time audio applications, a key parameter is the audio delay that depends on both the hardware and software of the node [10]. This parameter is closely related to the network synchronization.



(a) Star. **(b)** Fully-connected.

(c) Mesh. **(d)** Ring.

Figure 2.4. Most common topologies in ASN [42].

2.1.5 ASN commercial systems

Nowadays, the most common commercial application for indoor environments based on ASN are focused on reproducing music signals simultaneously by the speakers of the ASN. Figure 2.5 shows the system where these applications are performed, called as Multi-Room Speaker System (MRSS) [50], where all the speakers are connected through the Wi-Fi link in order to reproduce the same signal simultaneously. This system allows

to reproduce different signals by each speaker at the same time as well.

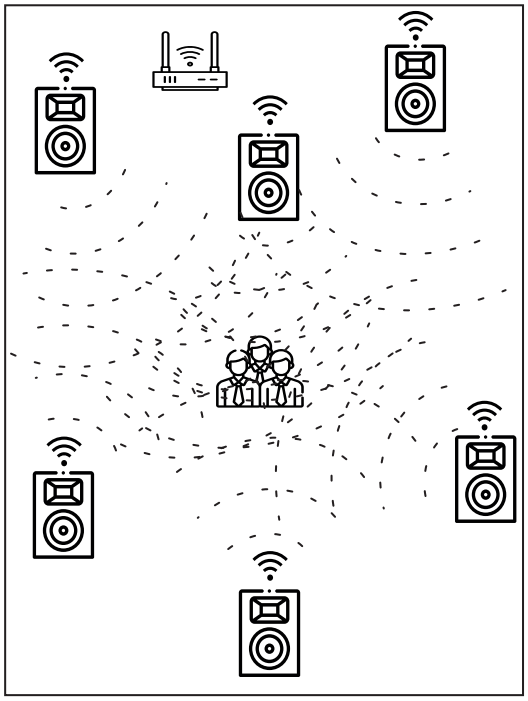


Figure 2.5. Scheme of a MRSS.

Some MRSS are focused not only on reproducing a sound but equalizing it according to the room where system is located in order to enhance the quality of the reproduced sound and improve the user experience [51]. In addition, since the elements of the MRSS (the acoustic nodes) are also composed by microphones, these systems are capable of performing speech detection applications [52] in order to give specific commands to the system to reproduce music.

Although the MRSS can generate acoustic environments within indoor spaces, the available commercial MRSS are focused exclusively on rendering sound without applying any processing except equalization. Furthermore, usually these systems are proprietary systems, i.e. the ASN must be composed by the same type of components (in this case, same speakers) otherwise the application performance is significantly reduced due to the

differences between the speakers of different manufacturers. In addition, each of these components is often very expensive [53]. In this dissertation, more affordable devices will form the active ASN.

2.2 Acoustic Networks based on mobile devices

2.2.1 Mobile device development

According to the data obtained in February 2021 [54], the use of the mobile devices, such smartphones and tablets, have shown an exponential growth in these last years. As Figure 2.6 shows, a data forecast was done for next months, for this reason the last year is highlighted with the $*$ symbol.

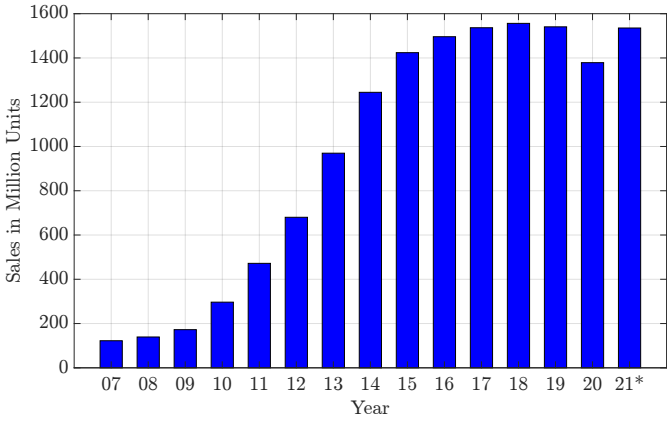


Figure 2.6. Increment of sells of mobile devices from 2007-2021 [54].

As a result of this exponential growth, a great number of audio applications have been developed, such as noise monitoring [55], evaluation of the sound [56], sound classification by using the smartphone to monitor the environmental sounds [57], or even to record ultra-sound signals with an in-build microphone in order to implement a proximity detection system [58].

Moreover, current mobile devices present more than one microphone and speaker that can be controlled in order to reproduce or/and record multi-channel signals. In addition, they offer a powerful processor that

is capable of performing complex algorithms that can be useful to extract information about the acoustic environment. Furthermore, the current mobile devices are capable of exchanging the information obtained with other mobile devices through different communication protocols (such as Bluetooth or Wi-Fi), that will be explained in Section 2.3. Taking into account the particularity of such mobile devices, they can be considered as acoustic nodes of an ASN because they provide all the required elements to act as nodes of an ASN (see Section 2.1.3).

On the other hand, as Figure 2.6 shows, the increase in mobile devices consume becomes significant in the last decade. One of the main reasons of this behavior was the introduction of stable versions of their Operating Systems (OS). In 2008, a wide range of OS were used, but they had a limited number of features. In addition, many updates were required frequently. However, during 2010-2011 two of these OS started to outstand from the rest, as Figure 2.7 shows, because they started to provide a robust and stable OS for the mobile devices. These two OS are well known nowadays and nearly all devices use one of these, and as a consequence, the rest of the OS have already been disappeared.

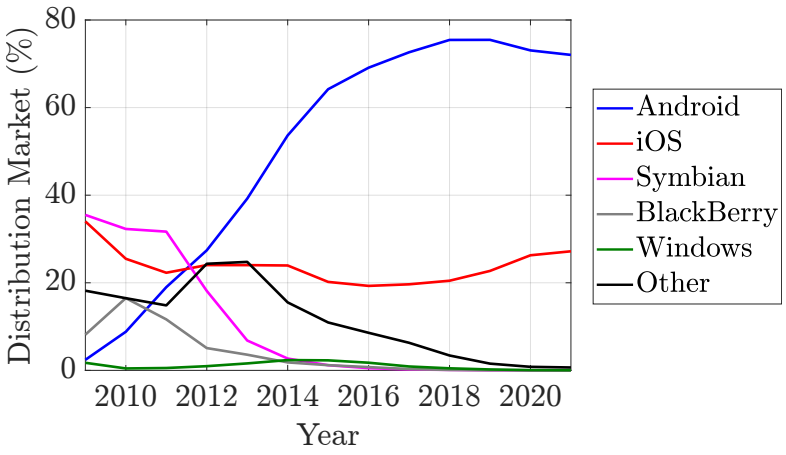


Figure 2.7. Worldwide Operating System distribution [59].

These two OS are: Android, from Google company, and iOS, from Apple company. Since both are very different OS, they are usually compared in order to analyze their differences as well as advantages and disadvan-

tages [60, 61, 62]. Although both OS are the most common used nowadays, according to Figure 2.7, Android is more extended than iOS in terms of number of devices, specifically almost 80% of devices use Android and 20% of devices use iOS. This gap between both OS is due mainly to two reasons:

1. The cost of the mobile device of each OS. Android devices are usually cheaper with a broader range of prices than iOS devices.

2. The system category. Meanwhile Android is an open free system, iOS is a proprietary system. For this reason a greater number of different companies (such as Samsung, LG, Motorola, etc ...) use Android as the OS for their mobile devices. In contrast, iOS only is used by one company, Apple.

As it is shown in Figure 2.7, nowadays a wide range of different devices with Android are presented and for the reasons mentioned above, in this dissertation Android mobile devices are chosen to implement the acoustic nodes of an ASN. The study of the an ASN composed by mobile devices is described in Chapter 4.

Finally, since we focus on Android mobile devices, in the following paragraphs a brief explanation about the structure of the Android OS is given.

2.2.2 Android structure

The structure of the Android OS (also called Android stack) is implemented through different layers in order to allow the developers of each company greater flexibility and a greater functionality in their mobile devices. Figure 2.8 shows the stack of Android, the different layers that compose it and the libraries (set of functionalities) available in each layer.

The performance of the application depends on the stack, or more precisely, on the communication between the different layers that compose the stack. A good performance leads to an optimal communication between the layers, or in other words, the processing time required for one layer to transform the information in order to be processed by the next is minimum. This factor, performance, is one of the most common issues to address in Android devices [64, 65], but at the same time it is one of the most difficult to address since layers are not designed by the same developers, unlike

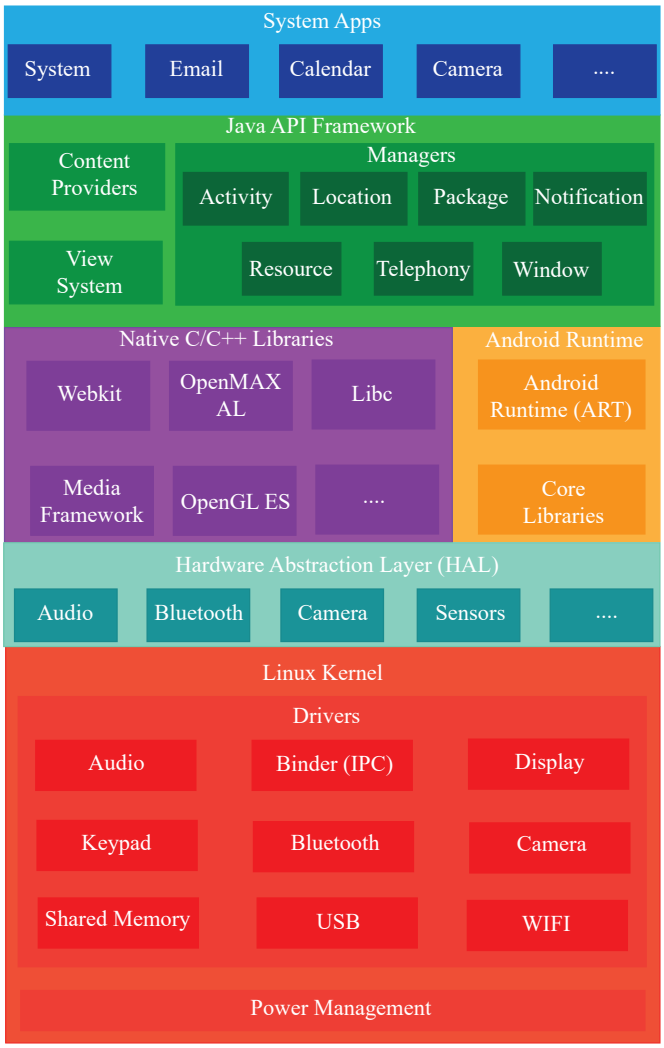


Figure 2.8. Android Stack [63].

iOS. In fact, to implement a multi-platform system, the stack is separated in different sections where each section is designed by different developers resulting in a fragmented stack where upper layers are designed by the developers of Android and lower layers are designed by the developers of each company of mobile devices. In this regard, because different companies are capable of implementing Android systems, the performance of each

specific application can be significantly reduced depending on the mobile device, as indicated in Section 2.2.1. On the other hand, the stack is exclusively software and hardware has not been considered in the performance, but it greatly impacts on it, resulting in better performance for a powerful hardware.

Despite this disadvantage, Android system has significantly improved during the last years through continuous software revisions, where last one corresponds to version 12 [66]. However, when the application demands high system resources, such as multimedia applications, performance is still a weak point. Multimedia operations require a fast processing, specially when real-time processing is the main objective, but this goal becomes very difficult to achieve when Android devices are used [67, 68]. This behavior will be thoroughly explained in the study carried out in Chapter 4 where an ASN composed by Android devices is implemented in order to perform an audio application.

2.3 Communication protocols

The current mobile devices have become very powerful systems capable of exchanging information with similar or different devices due to its capacity of using different communication technologies to transmit data. In order to perform a practical ASN, mobile devices must exchange their data with the purpose of providing the network with a certain intelligence that will allow to implement more efficient algorithms based on the information shared by the nodes.

When using mobile devices, the most common communication protocols are wireless. From these protocols the most known (and therefore the most used) are Wi-Fi (IEEE 802.11 [69]) and Bluetooth (IEEE 802.15.1 [70]). At present, these two protocols are widely used in different applications, although Wi-Fi is used in most of the cases because it provides higher transfer rates and higher coverage than Bluetooth. Nevertheless, Bluetooth is still used in some specific fields, such as the headphones or speakers where most of them use Bluetooth.

On the other hand, according to [71] where a comparison among different communication technologies is carried out, the main differences between Wi-Fi and Bluetooth protocols are:

1. Transmission band. Meanwhile Bluetooth provides the service in the 2.4 GHz band, Wi-Fi is able to transmit in 2.4 GHz and 5 GHz bands.

2. Data transfer rate. Although, the rate has been improved after each version, Wi-Fi achieves much higher rates than Bluetooth, as it is shown in Table 2.1.

Wi-Fi		Bluetooth	
Generation	Rate (Mbps)	Version	Rate (Mbps)
1st (802.11)	2	1	1
2nd (802.11 b)	11	2	2.1
3rd (802.11 g/a)	54	3	24
4th (802.11 n)	600	4	32
5th (802.11 ac)	3600	5	50

Table 2.1. Data transfer rates for Wi-Fi and Bluetooth protocols.

Through the flexibility and versatility that wireless communications are able to provide, current devices are able to build a network similar to the network represented in Figure 2.1b, by placing the devices at any point of the space. As Table 2.1 shows, since Wi-Fi protocol provides better rates, it is the optimal solution for connecting the acoustic nodes of an ASN due to the fact that data transmission is faster.

On the other hand, audio devices (such as speakers or headphones) can be implemented through a wireless protocol in order to receive data from the wireless link and reproduce that data [72, 73]. Considering this performance, when a mobile device uses a wireless speaker in order to reproduce data through that speaker instead of the own loudspeakers of the mobile device, the flexibility of the ASN is increased since microphone and speaker can be located at different spots. However, as it is stated before, most of the current audio devices use the Bluetooth protocol instead of the Wi-Fi one resulting in a slower data transmission between the mobile device and the speaker.

Since the use of wireless speakers in the ASN increases the network flexibility, we will considered them instead of using the own loudspeakers of the mobile devices. However, a comparison should be made between

the features offered by the Wi-Fi and Bluetooth speakers in order to select the best option to use in this dissertation. On one side, Bluetooth provides a variety of speakers greater than Wi-Fi. On the second hand, Bluetooth provides a generic method called Advanced Audio Distribution Profile (A2DP) that explains how the multimedia data is streamed from the mobile device to the speaker [74], while Wi-Fi speakers must be used through a proprietary application that is provided by the speaker manufacturer, in order to exchange the audio information between devices. That means that the application can be different for different speakers of different manufacturers. In addition, no other sound reproduction application can independently transfer data to the speaker because the transmission is managed by the manufacturer application. Finally, since Wi-Fi speakers are composed by more physical components, such as the Ethernet port, as well as additional processing algorithms in order to achieve a better sound quality than Bluetooth speakers, they are usually more expensive than Bluetooth speakers.

Therefore, Wi-Fi speakers achieve a better sound quality and they are able to receive audio data faster, but they cannot be accessed without the manufacturer application, while Bluetooth speakers are more affordable and they provide a generic method to transmit the audio data without any additional application. For these reasons, the best choice is to use Bluetooth speakers.

2.4 Psychoacoustics

Psychoacoustics is the science that studies how a person perceives a sound of the surrounding environment [75]. In this section a revision of the psychoacoustic topics related to how two sounds are perceived simultaneously in the human hearing system will be revised.

2.4.1 Human hearing system

An acoustic sound can be described as a time-varying sound pressure measured in Pascal (Pa), where values from 10^{-5} to 10^{2} are within the range that a sound is perceived [6]. Due to the large extension of this range, the

sound pressure level is defined in logarithmic scale as [6]:

$$L_{\mathrm{SPL}} = 20 \log_{10} \left(\frac{p}{p_0} \right) \ \mathrm{dB}\,, \tag{2.1}$$

where p and p_0 are, respectively, the sound pressure of the captured sound and the reference sound pressure which depends on the propagation medium (in the particular case of the air it takes a value of $p_0 = 20$ uPa). In (2.1), L_{SPL} is the corresponding sound pressure level in logarithmic units, from now called as SPL level.

The SPL level is widely used to provide a real measurement of the audio system. When an audio sound is captured by a microphone, usually this signal is provided within the range of $[-1,\ 1]$ without units of measurement, usually called as Full Scale (FS) units. That means that to obtain a real measurement of its SPL level, we must carried out a calibration process [76]. This process is based on (2.1) and it relates the digital information, in the range of $[-1,\ 1]$, with the physical measurement in dB SPL.

When the sound pressure reaches the human ear, it is analyzed by three different regions called respectively outer, middle and inner ear, as it is illustrated in Figure 2.9 [77]. Firstly, the outer ear transmits the sound pressure to the middle ear where it is transformed to a vibratory stimulus. Secondly, this stimulus is transmitted to the inner ear where the vibration is transformed into electrical information, through the auditory nerves, in order to send the information to the brain where the electrical information is interpreted.

Thus, as stated before, the sound pressure is transmitted by the air in the outer ear, while a vibratory stimulus is sent via liquid in the inner ear [77]. Therefore, a transformation of the sound pressure in a vibratory stimulus is done in the middle ear. In addition, a proper adaptation of impedances, due to the propagation medium change, to avoid signal losses is performed in the middle ear. However, as [6] states, a perfect match of impedances cannot be performed, being the narrowband signals centered on 1 kHz the best adapted.

Due to the human ear behavior, the auditory system perceives the sound in a very specific range. This range is called the audible spectrum and it is limited by four boundaries. The first two involve the limit frequencies that are perceptible, that is, 20 Hz and 20 kHz. The other two

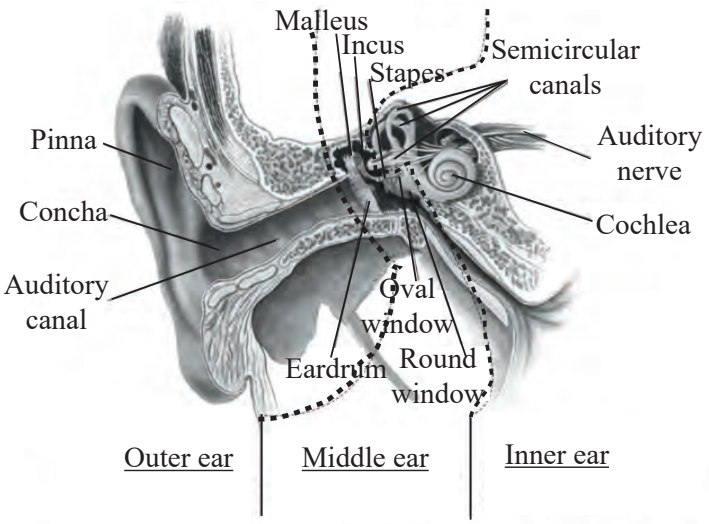


Figure 2.9. Regions of the human auditory system [77].

boundaries were presented above and they involve the limit pressures that are perceptible, that is 10^{-5} Pa to 10^{2} Pa. Through these boundaries, an area can be described that presents the sounds that can be perceived by the human hearing system. Any sound outside this area cannot be perceived by the ear, such as ultra-sounds. This area is called the hearing area [6] and it is usually represented as shown in Figure 2.10, where the SPL level vs frequency is shown.

Figure 2.10 shows important information, such as the regions normally occupied by the music and speech, the region of SPL levels where is located a harmful sound for the auditory system (represented through the upper curves) and finally, the minimum SPL level required for a sound to become audible by our hearing system, also called as the threshold in quiet (or human hearing threshold). According to Figure 2.10, the threshold in quiet is different for different frequencies, making some regions more sensitive than others. In fact, the most sensitive region is located between 2–5 kHz, where most speech sounds are located, while the least sensitive are located in the extremes of the hearing area, or in other words, the least sensitive sounds are located in the lowest and highest frequencies of the hearing area while the most sensitive is located in the upper-middle frequencies. In

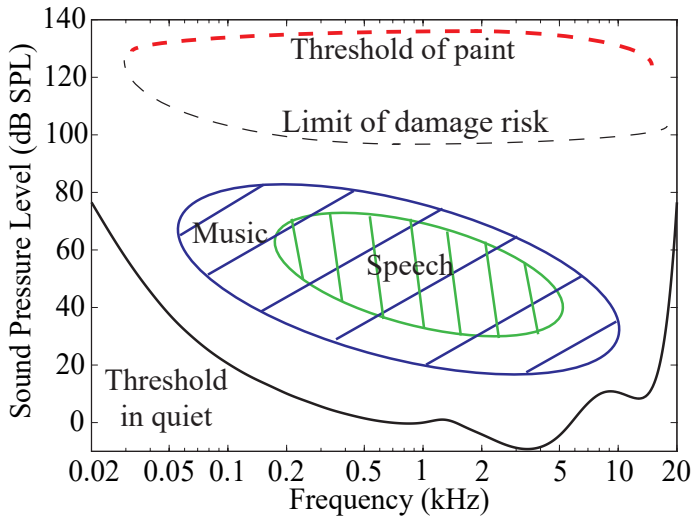


Figure 2.10. Hearing Area of the human hearing system [6].

conclusion, the human hearing system perceives sounds in different ways depending on frequency.

The threshold in quiet is a very useful parameter for understanding how the auditory system deals with the sounds in the environment. It is the result of a large number of subjective tests performed on a large number of subjects. With the aim of obtaining that threshold, a single tone with a variable frequency ranging from low to high values is generated. Then, the subject is able to change its SPL level in order to obtain the perception threshold at the corresponding frequency. This test is called Bekésy-tracking method [78] and the current auditory tests (or audiometry) are based on that method in order to evaluate the hearing thresholds of subjects, even with cochlear implants [79].

2.4.2 Critical Bands

The concept of critical bands was introduced by Fletcher and Munson [80] in order to characterize the response of the auditory system. This model explains how the inner ear decomposes the sound into several frequency bands resulting in the behavior shown in Figure 2.10. These frequency bands are known as critical bands because when two different sounds are

located at the same band, they interfere with each other causing a different perception than when these two sounds are located at different critical bands. In [81], this behavior is studied for the loudness parameter (described in Section 2.4.3), being the perceived loudness greater when sounds (in that case the tones) are located in multiple critical bands instead of within a single one.

As [75] explains, auditory system uses auditory filters according to the critical bands to analyze the sounds. But due to the composition of the ear (see Figure 2.9), the bandwidth of these filters is different depending on the frequency. This relationship between the bandwidth of the filters (also known as critical bandwidth) and the frequency can be seen in Table 2.2, where each critical band (represented through the critical band number, ν) is presented in terms of the upper and center frequency (f_{u} and f_{c} respectively) and its bandwidth (BW).

ν	f_{c} (Hz)	f_{u} (Hz)	BW (Hz)	ν	f_{c} (Hz)	f_{u} (Hz)	BW (Hz)
1	50	100	100	13	1850	2000	280
2	150	200	100	14	2150	2320	320
3	250	300	100	15	2500	2700	380
4	350	400	100	16	2900	3150	450
5	450	510	110	17	3400	3700	550
6	570	630	120	18	4000	4400	700
7	700	770	140	19	4800	5300	900
8	840	920	150	20	5800	6400	1100
9	1000	1080	160	21	7000	7700	1300
10	1170	1270	190	22	8500	9500	1800
11	1370	1480	210	23	10500	12000	2500
12	1600	1720	240	24	13500	15500	3500

Table 2.2. Relationship of frequency and critical band.

As Table 2.2 shows, the critical bandwidth increases as the frequency increases, leading to lower resolution at high frequencies. In addition, this relationship is related to the sampling frequency (f_{s}) of the sound to be analyzed, causing that the last critical band is defined by $\frac{f_{\mathrm{s}}}{2}$. Therefore, assuming a sound sampled at $f_{\mathrm{s}} = 16000$, last critical band to be analyzed

will be the 22^{nd}, while for a sound sampled at $f_s = 8000$, the last critical band will be the 18^{th}.

On the other hand, from Table 2.2 a non-linear increase of the critical bandwidth can be appreciated. In fact, for low frequencies (up to the 5^{th} critical band) the relationship between the center frequency and the bandwidth of the critical band is linear, but from that band onward, a logarithmic relationship can be appreciated. In order to model this non-linear relationship, [82] proposed the following expression:

$$\Delta f_\nu(f) = 25 + 75\left[1 + 1.4\left(\frac{f}{1000}\right)^2\right]^{0.69}, \tag{2.2}$$

where $\Delta f_\nu(f)$ is the critical bandwidth, f is the frequency and ν is the critical band number. To evaluate the accuracy of the model in (2.2), the difference between expression (2.2) and the critical bandwidth data of Table 2.2 is represented in Figure 2.11, where the red line represents the expression of [82] and the blue dots represents the data of Table 2.2.

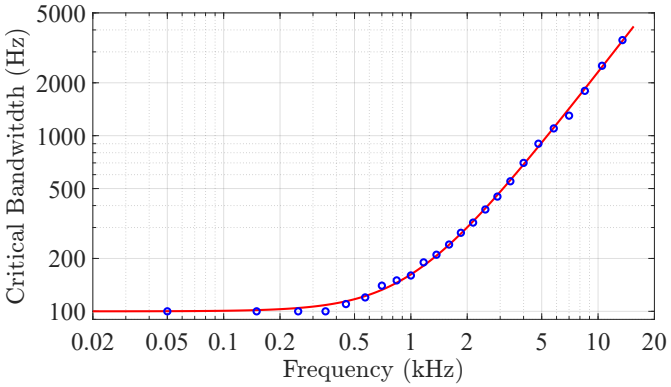


Figure 2.11. Difference between Table 2.2 and expression (2.2) [82].

As Figure 2.11 shows, the model proposed by [82] adequately implements the relationship in most of the frequencies leading to a proper method to define the bandwidth of all the critical bands where according to Table 2.2, 24 bands can be found. The critical bands are usually called as Bark bands as well and are denoted by the same symbol (ν) [6]. Therefore,

in this dissertation both, critical and Bark bands, will be used to refer to these bands.

2.4.3 Masking between signals

Subjective metrics based on the previous critical band model (see Section 2.4.2) have been implemented [6, 83, 84, 85] to model the human hearing perception. One of the most well-known metrics is the masking threshold that models the masking effect between signals.

The masking effect occurs when two different sounds are perceived by the auditory system causing a change in the perception of the first sound (called maskee or target sound) due to the second sound (called masker sound). This effect is extended in both, temporal and frequency domains leading to two different masking effects: temporal and spectral masking [86]. Although in Chapter 3 of this dissertation, spectral masking effect is analyzed in detail in order to implement audio applications, a brief explanation of the temporal masking is also given here. According to [87], temporal masking is an effect that occurs over a certain time interval when two sounds interact in the auditory system. It can be spread before and after of the masker sound, just as [6] explains with the Figure 2.12 where a masker of 200 ms and a short tone burst (maskee sound) are used.

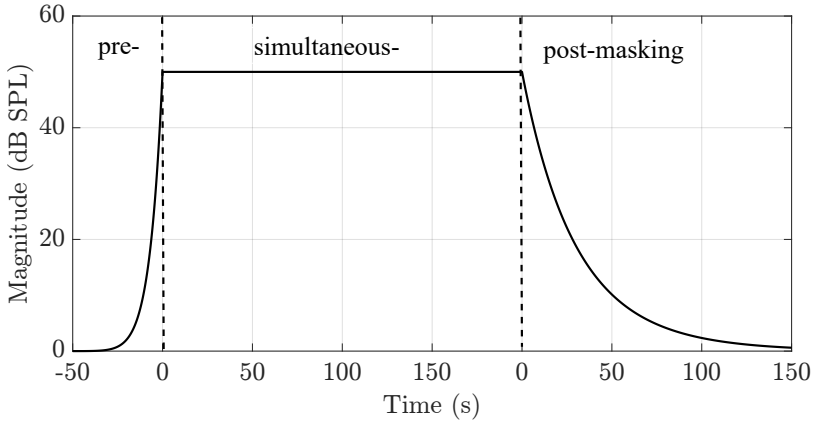


Figure 2.12. Temporal Masking for a masker of 200 ms [6].

On the other hand, spectral masking is produced when two different

sounds, located at the same frequency components (or closely components), are perceived by the auditory system at the same time [88]. The effect of the spectral masking depends on the SPL level of maskee and masker sounds, and depending on these levels, the maskee sound can reduce its perception. The masking effect that the masker sound produces on the maskee sound increases as the SPL level of the masker sound increases. Finally, when the SPL level of the masker sound is much higher than the SPL level of the maskee sound, the masker sound is able to mask completely the maskee sound, making it inaudible. Therefore, the masking effect depends on the frequency content of the masker sound [6].

An example of spectral masking is shown in Figure 2.13, where spectral masking from a narrowband noise (the masker sound) to a pure tone (maskee sound) is shown. In this case, three different scenarios are presented, each one in a different critical band in order to appreciate dependence of the spectral masking effect with the frequency. In all three scenarios, the masker sound is a narrowband noise of bandwidth one critical band and the maskee sound is a pure tone located at the same critical band as the noise.

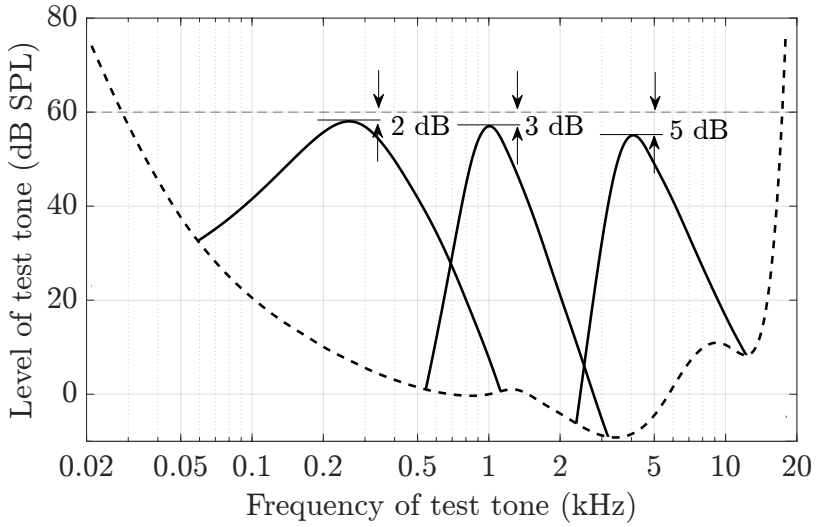


Figure 2.13. Spectral masking of a narrow-band noise with a bandwidth of one critical band [6].

As shown in Figure 2.13, the center frequencies of the maskee and masker sounds are 250 Hz, 1kHz and 4kHz. In addition, the SPL of the masker (60 dBSPL) is indicated with the dotted line, as well as the human hearing threshold (the dashed line). It can be seen that the spectral masking is always lower than the SPL of the masker in any of the three frequencies, leading to assume that the masking produced by a sound is always below its SPL level. In addition, the masking level is lower as frequency is increases, thus, decreasing the effect of the masker sounds at high frequencies. Therefore, as shown in Figure 2.13, the SPL level of the maskee sound must exceed the masking level produced by the masker sound in order the maskee sound to become audible, otherwise the maskee sound will be inaudible.

The spectral masking effect is a very relevant issue in audio applications that demands a proper sound perception. Some of these applications are focused on audio coding [89, 90], where the masker is considered as the quantification noise, speech recognition [91, 92] or music equalization [93, 94]. Given the impact of this effect, in this dissertation a deeply study of the spectral masking will be done through the masking threshold metric. As the Chapter 3 will explain, this metric depends on the perceptual parameter called tonality, that depends at the same time on the loudness parameter. For this reason, in the following paragraphs a brief explanation of both parameters is done.

LOUDNESS

According to [80], loudness is a parameter that subjectively describes the sound pressure level. Additionally, as it was depicted in Figure 2.10, the sound pressure that the auditory system is able to perceive also depends on frequency. For this reason, the loudness is related with the intensity and the frequency composition of the sound. In addition, when sound is composed by several frequency components, the loudness depends on the masking caused among the different components, as [95] explains.

The loudness uses as a measurement unit the “sone”, which is usually denoted by the symbol N. However, in order to avoid confusion with the number of nodes of an ASN, in this dissertation it is denoted as F. The “sone” unit is related with a tone located at 1 kHz in such a way that 1 “sone” represents the perceived sound power of a tone of 40 dB SPL located at 1 kHz. Therefore, if the loudness of an arbitrary sound is 1 sone, it

means that it is as loud as a tone of 40 dB SPL located at 1 kHz. However, another measurement is often used: the "loudness level"(denoted in this case by L_{F}), whose measurement unit is the "phon" [6]. In contrast to the loudness, F, the loudness level is directly related with the SPL of a tone of 1 kHz, in such a way that a sound of "X" phons means that the sound is as loud as a tone of 1 kHz of "X" dB SPL. These two measures, loudness (F) and loudness level (L_{F}), are used equally to evaluate subjectively the sound pressure level of the sounds and they are related through the next expression [6]:

$$L_{\mathrm{F}} = \begin{cases} 40 + 10\log_2(F) & F \geq 1 \\ 40\,(F + 0.005)^{0.35} & F < 1\,. \end{cases} \tag{2.3}$$

Assuming that the sound to analyze is a pure tone, the loudness level can be represented as a curve that depends on the SPL level and frequency of the tone to analyze. The international standard ISO 226 [96] establishes the different curves according to the loudness levels, leading to the equal-loudness contour curves that are represented in Figure 2.14, where loudness level is defined for certain frequencies up to 12.5 kHz.

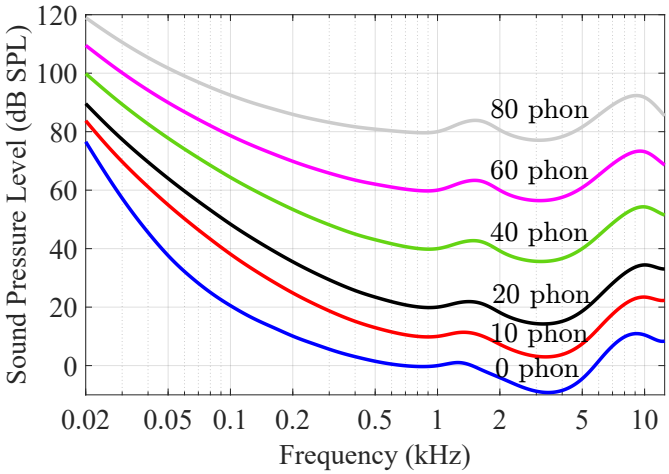


Figure 2.14. Equal loudness contour curves for pure tone [96].

Considering a specific curve, such as the 20 phon curve, it represents the required SPL of a tone in order to be as loud as a tone of 1 kHz of

20 dB SPL. As Figure 2.14 shows, in case of the tone of 1 kHz, the loudness level matches with the SPL level in all the curves, due to the definition of loudness level. Additionally, the lower curve (0 phons) indicates the threshold in quiet, shown in Figure 2.10.

Since the loudness depends on the perception of a sound in the auditory system, it is a non-linear parameter. This non-linearity can be intensified in the case of broadband sounds where frequency and temporal effects (such as masking) of other components can interfere by completely changing the loudness perceived and the loudness level. In order to consider all the contributions of these components, different standards have been defined to obtain the loudness of complex sounds. In this dissertation the ISO 532-b is used as the standard to estimate the loudness, F, [97] (and consequently the loudness level through (2.3)). This standard estimates the loudness through the combination of the specific loudness per critical band. In other words, the standard first estimates the loudness per critical band of the broadband sound and then combines all the contributions to obtain the total loudness (F).

TONALITY

The last psychoacoustic parameter that we will introduce is the tonality. According to [98], the tonality quantifies the perception of the tonal content of a sound. The tonality gives information about the frequency composition of the sound, leading to a specific perceptual analysis of the sound that can be used in different psychoacoustic models, such as masking [99] or perceptual annoyance [100].

As Figure 2.14 shows, narrowband sounds (such as tones) located at different frequencies are not equally perceived. Since the tonality also depends on the frequency of the sound, this behavior leads to consider that tonality and loudness can be related, as [101] explains. The tonality can be estimated through different methods, such as those explained in [101, 102], where tonality is estimated through loudness, or in [98], where the relationship between the tonal and non-tonal parts of a sound are used.

According to [103], the tonality parameter is very important in the masking effect. The masking has been demonstrated to be different in pure tones than in narrow-band noise signals located at the same frequency, being the masking effect of the noise signals stronger than that produced by the tones [104, 105]. Based on the masking model that will be used

and detailed in Chapter 3, the difference in masking is caused because of the tonality of the signals: a white noise causes more masking due to its low tonality whereas a pure tone produces less masking because its high tonality. In order to understand the effect of the tonality over the masking, Figure 2.15 shows the masking level (in dB SPL) of a masker sound in terms of critical band. The SPL per critical band of the masker is represented as a dotted line in order to appreciate the offset (Δ) that the masking level of a masker suffers on its SPL level (as can also be seen in Figure 2.13).

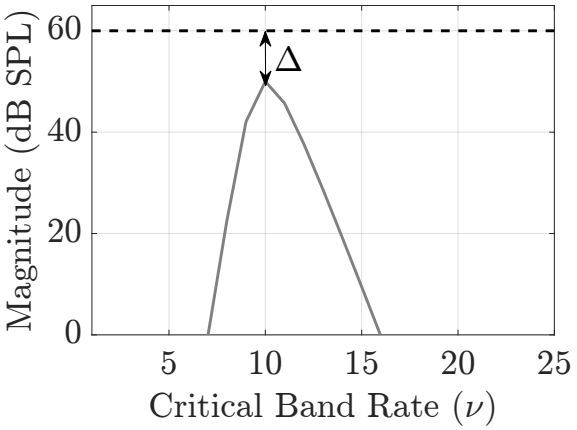


Figure 2.15. Effect of tonality over the masking [103].

The offset or shift shown in Figure 2.15, is the effect of the tonality over the masking. Because tonality depends on the frequency composition of the sound to be analyzed (in this case the masker), it will be different for different masker sounds leading to different masking levels, even when the maskers have the same SPL per critical band.

Sound Masking on ASN

3

This chapter describes processing techniques that allow to estimate the spectral masking threshold of signals through a perceptual analysis. This analysis is based on the human auditory system, which decomposes audio signals in different spectral bands with different bandwidths called critical bands. According to the literature, the masking of a signal is obtained through the spectral masking threshold model that can be divided into the ***auditory masking model*** *and the* ***tonality estimator model****. Both processes are deeply analyzed in this chapter in order to provide an accurate estimator able to measure the masking parameter of the signals. On one side, the auditory masking process provides a masking curve based on a specific pattern. On the other hand, the tonality estimator process estimates an offset that will be applied to the masking curve depending on the features of the signal. Finally, in the two last sections, an equalizer modeled on an ASN is implemented in which the above spectral masking model is applied by using two different methodologies. Therefore, a perceptual equalization that adapts dynamically according to the masking between signals is implemented.*

3.1 Spectral masking model

The analysis that will be developed in this chapter is based on the masking effect, introduced in Section 2.4.3. As explained, the masking effect is produced in both time and frequency domains [86], although the analysis performed in this dissertation is focused on the masking effect in the frequency domain, called from now on as spectral masking.

The spectral masking is analyzed using the spectral masking threshold model that indicates the level of masking that an audio signal is capable of producing at each critical band (see Section 2.4.2). The spectral masking model forms the basis for different applications, such as coding applications [106, 107] where the noise introduced by the encoder is analyzed in order to perform a more accurate coding of the audio signal, the enhancing of audio signals (music or speech) in a noisy environment in headphones [94], or in a car through the loudspeakers [108]. This masking model can also be used to remove irrelevant components of any real-world music or speech sound to facilitate the sound synthesis and design [109], or to enhance the sound quality of the hybrid electric power-train noise perceived inside the passenger compartment [110].

According to [6], the spectral masking model was developed as a result of multiple subjective studies of the masking between different narrowband signals (such as tones or narrowband noise signals located in a single critical band). In this regard, the target audio signal is called the maskee signal and the signal used to mask the target signal is called the masker signal, as it is described in Section 2.4.3. The different studies show that the masking effect can be modelled as a triangular-shaped function (as it will be shown in Section 3.2.2) leading to assume the masking spreads over several critical bands (front and back critical bands) depending on the critical band where masker signal is located. This behavior has been analyzed for years resulting in different masking models, as for instance those based on the ERB filters [106] or based on the masking patterns [111]. Both models are commonly used in audio coding applications [103], although they can be used in applications with different approaches as well, such as equalization [93, 94]. In this dissertation, the masking model based on masking patterns [111] will be discussed. This model can be decomposed as shown in Figure 3.1 where $x_m(n)$ represents the masker signal with m as the frame index. This model is composed by two main processes needed

to obtain the spectral masking threshold of the masker signal ($T_{x,m}(\nu)$) per critical band (see Section 2.4.2).

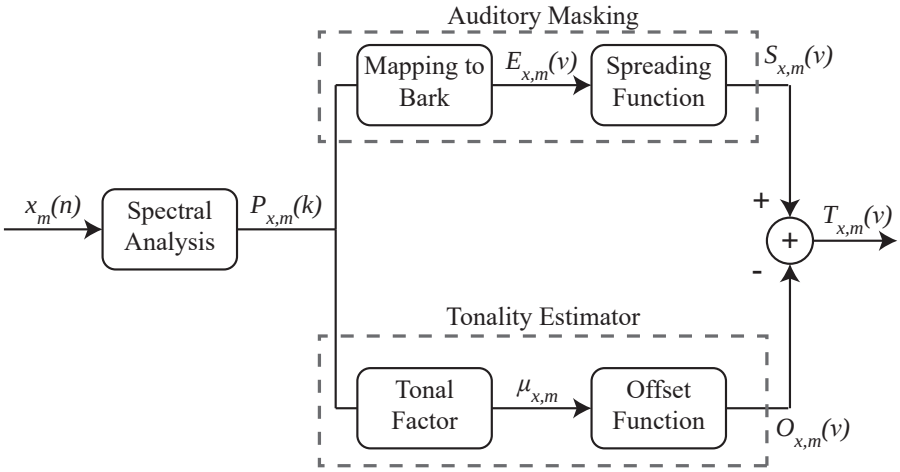


Figure 3.1. Block diagram of the spectral masking model.

According to Figure 3.1, in order to estimate $T_{x,m}(\nu)$ of a specific masker signal, the spectral masking model is divided in two blocks. The upper branch is called the "Auditory Masking Model" because it estimates the masking pattern based on the human auditory system. The process of the bottom branch is called "Tonality Estimator Model" as it estimates the tonality parameter of the masker signal, explained in Section 2.4.3.

As shown in Figure 3.1, the diagram input is the m^{th} frame of the signal of interest $x(n)$, whose length is M_{S} samples. In an initial state, the Power Spectrum (PS) of the $x_m(n)$, $P_{x,m}(k)$, is estimated through the "Spectral Analysis" block by means of the Welch method [112] averaging N_{F} frames, including the current frame (m), such that $M_{\text{T}} = N_{\text{F}}M_{\text{S}}$ samples, using a hamming window, an overlap of 50% and a FFT size of $N_{\text{FFT}} = M_{\text{S}}$ samples. As shown in Figure 3.1, $P_{x,m}(k)$ is estimated in linear units over the discrete frequencies $f_k = \frac{k}{N_{\text{FFT}}} f_{\text{s}}$, where f_{s} is the sample rate used and k is the frequency bin, such that $k = 0, \ldots, \frac{N_{\text{FFT}}}{2}$.

Once $P_{x,m}(k)$ has been estimated, the perceptual processes "Auditory Masking Model" and "Tonality Estimator Model" are performed. These processes are explained in detail in the following sections.

3.2 Auditory masking model

3.2.1 Background

The main objective of the "Auditory Masking" process is to obtain the overall masking curve ($S_{x,m}(\nu)$) that represents the shape of the masking of the m^{th} frame of the masker signal, $x(n)$, [94]. As indicated in Section 3.1, it is based on patterns that provide information of the masking per critical bands (see Section 2.4.2) in order to mimic the human auditory system.

The masking patterns were designed as a result of the analysis of multiple experiments where the masking caused by narrowband signals (signals with a maximum bandwidth of one critical band, such as tones or narrowband noise signals) [6]. They represent the masking caused by one critical band (where the masker is located) over the rest of them [90]. As shown in [103], these patterns are functions with a triangular shape, indicating that the masking of one critical band spreads over the rest. For this reason, the masking patterns are more commonly known as "Spreading Functions" [103, 113, 114].

Although the "spreading functions" represent the auditory masking pattern for narrowband signals, for years they have been used for the analysis of broadband signals as well. Nevertheless, the masking of broadband signals was first studied by [115] where the objective was to predict the masking of broadband signals using narrowband signals, alone and combined, in order to determine whether the masking of the broadband signal could be estimated through the combination of the masking caused by each of their narrowband components. This combination of different masking is known as "additivity of masking" [6, 116], since the contributions of the different maskers are added together.

The "additivity of masking" has been discussed several times due to the complexity of its modeling. As [116] explains, the prediction of the masking of two combined narrowband signals does not correspond to the experimental results, that show a masking level between 10 – 17 dB above the predicted masking. On the other hand, the results in [117] show that the prediction of the masking of two combined narrowband signals matches experimental studies in certain cases, depending on the SPL level and frequency position of the masker signals. Finally, in [118] a study (based on [116]) is provided where the combination of different masking can be

defined through a power-law expression, which will be seen in (3.5), in order to select the way the different contributions of each masker are added together.

The combination of the different masking leads to the overall masking curve ($S_{x,m}(\nu)$), that it is the result of the upper branch of Figure 3.1. For this reason in the following sections the steps that compose the "Auditory Masking Model" are described in detail for a full understanding of this procedure.

3.2.2 Overall masking curve computation

According to Figure 3.1, the first step of the upper branch computes the energy per critical band of the masker signal, $x_m(n)$. This transformation is done in two stages. Firstly, frequency bins are converted into Bark bins through the following expression [82]:

$$\nu = 13 \arctan\left(\frac{0.76 f_k}{1000}\right) + 3.5 \arctan\left(\frac{f_k}{7500}\right)^2, \tag{3.1}$$

where f_k is the frequency in Hz and ν is the critical band index (or Bark index), being $\nu = 1, \cdots, N_c$ with N_c as the number of critical bands. The frequency range of all the critical bands is shown in Table 2.2, specifying the center, low and high frequency of each critical band for the whole spectrum.

This mapping provides a non-linear relationship between frequency and Bark domains, specifically it provides a logarithmic relationship, as Figure 2.11 shows. This relationship causes the critical bandwidth of each critical band to increase according to their center frequency. On the other hand, as (3.1) describes, mapping depends on frequency resolution, since $f_k = \frac{k}{N_{\mathrm{FFT}}} f_s$, with N_{FFT}, f_s and k defined in Section 3.1. Therefore, the FFT size, N_{FFT}, will define the number of frequency components that each critical band will have. Since the bandwidth is narrower in the lower critical bands (see Figure 2.11), a poor frequency resolution will cause the low critical bands to be composed by a few frequency components, resulting in an inaccurate estimate of the lower bands.

Once the mapping to Bark domain has been done, in a second stage,

the energy per critical band is obtained through [111]:

$$E_{x,m}(\nu) = \sum_{k=\inf(\nu)}^{\sup(\nu)} P_{x,m}(k)\,, \tag{3.2}$$

where $E_{x,m}(\nu)$ is the energy per critical band in linear units and $\inf(\nu)$ and $\sup(\nu)$ correspond to the frequency bin of the lower and upper boundary of the Bark band ν, respectively. This expression gathers the frequency components of the power spectrum according to the critical band where they are included. As an important aspect to consider in this relationship is the fact that the "power spectrum" is considered as the "energy spectrum" divided by the frequency interval (Δ_{f}) where that energy has been calculated. But, in this case $P_{x,m}(k)$ already considers the frequency interval, causing that $\Delta_{\mathrm{f}} = 1$ in (3.2).

According to Figure 3.1, the second block of the upper branch is the "spreading function" block where the "additivity of masking" is performed. As stated before, the masking patterns provides information about the masking caused by a specific critical band (named as masker band) over the rest of them (named as maskee bands).

A large number of masking patterns based on different approaches to the human perception of sounds have been proposed in the literature. In [103], an extensive analysis of different masking patterns can be found. However, the process to obtain $S_{x,m}(\nu)$ is identical regardless to the used pattern. The masking pattern proposed in [107] is the base of modern audio coding standards [90, 119] and we will use it as the masking pattern in the following steps to illustrate how the method of the "additivity of masking" works. That pattern can be expressed as follows:

$$B_\nu(\eta) = 15.81 + 7.5\,(\Delta_\nu(\eta) + 0.474) - 17.5\sqrt{1 + (\Delta_\nu(\eta) + 0.474)^2}\,, \tag{3.3}$$

where ν is the maskee band, η is the masker band, $\Delta_\nu(\eta) = (\nu - \eta)$ and $B_\nu(\eta)$ is expressed in dB units. The Figure 3.2 shows the masking pattern obtained for $\eta = 8$ where a triangular shape can be appreciated. This triangular pattern is common to all the spreading functions [103].

For broadband signals, the spreading function in (3.3) must be obtained

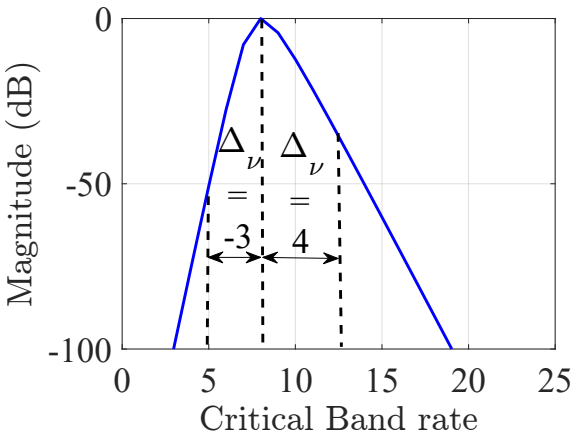


Figure 3.2. Masking pattern proposed in [107] for $\eta = 8$.

for each critical band η. For this reason both, η and ν are defined in the range of 1 to N_{c}.

In (3.3), we can see that masking patterns do not consider the level provided by the masker signal, or from an alternative point of view, the masking patterns assume a SPL level of the masker signal of 0 dB SPL (as Figure 3.2 shows). Therefore, in order to provide the corresponding SPL level to the masking patterns, the energy per critical band of the masker signal (see (3.2)) must be included as:

$$b_{\nu}(\eta) = 10^{\frac{B_{\nu}(\eta)}{10}} E_{x,m}(\eta)\,, \quad \eta = 1, \ldots, N_{\mathrm{c}}\,, \tag{3.4}$$

where $b_{\nu}(\eta)$ are the corresponding masking patterns, considering the SPL level of the masker signal, $E_{x,m}(\eta)$, expressed in linear units.

We will illustrate the meaning of (3.4) with an example where a white Gaussian noise is used. The noise is sampled at $f_{\mathrm{s}} = 44100$ Hz and filtered with a low-pass filter of 1 kHz in order to provide a broadband signal. In addition, to estimate the energy per critical band of the noise, the SPL level of the masker signal must be known. Thus, the measurement must be done in a calibrated system. In this case we will assume an energy per critical band of 70 dB SPL. The result when the previous expressions ((3.3) and (3.4)) are used for this specific signal is shown in Figure 3.3, where the

energy per critical band, $E_{x,m}(\eta)$, of the masker signal is represented by a blue line.

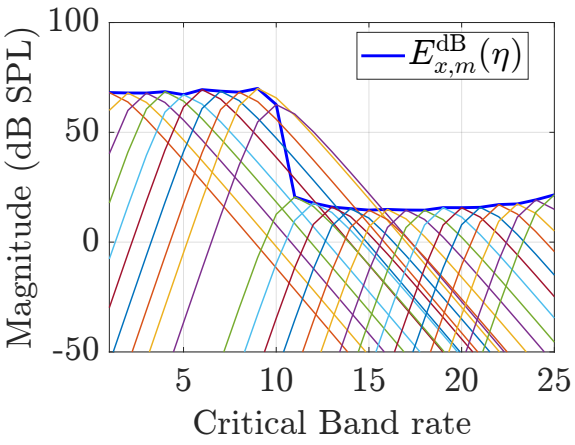


Figure 3.3. Masking patterns, $b_\nu(\eta)$, calculated by (3.4). Their maximum values correspond to the band of interest η.

As Figure 3.3 shows, each masking pattern is calculated according to the energy per critical band. It can be appreciated how both, energy and masking pattern, decrease from the 10^{th} critical band in advance, which corresponds with a noise low-pass filtered at 1 kHz (see Table 2.2).

Once the masking patterns have been particularized for the specific masker level, next step is to combine the different contributions of the masking patterns in order to obtain $S_{x,m}(\nu)$, performing what is known as the "additivity of masking". This final step is performed through the next expression that represents the overall masking curve (in dB units) [94]:

$$S_{x,m}(\nu) = 10\log_{10}\left(\left(\sum_{\eta=1}^{N_{\mathrm{c}}}[b_\nu(\eta)]^\alpha\right)^{\frac{1}{\alpha}}\right), \quad 0<\alpha<\infty\ ;\ \nu=1,\ldots,N_{\mathrm{c}}\,, \tag{3.5}$$

where α indicates the weight of the different contributions in order to perform the summation. This expression can be seen as the alpha-norm of a vector of N_{c} components since $b_\nu(\eta) \geq 0$. Therefore, setting $\alpha = 1$

corresponds to the simple summation of the terms while taking $\alpha \to \infty$ corresponds to using the highest masking pattern. When $\alpha < 1$ is used, the masking obtained produces a higher level than the simple summation of the terms, just as the study in [116] presents, where the best results are obtained when $\alpha = 0.33$.

Returning to our illustrative example, the value of $S_{x,m}(\nu)$ with $\alpha = 1$ has been obtained from (3.5) using the masking patterns in Figure 3.3. The curve for $\nu = 8$ is shown in Figure 3.4 to illustrate the summation of the terms.

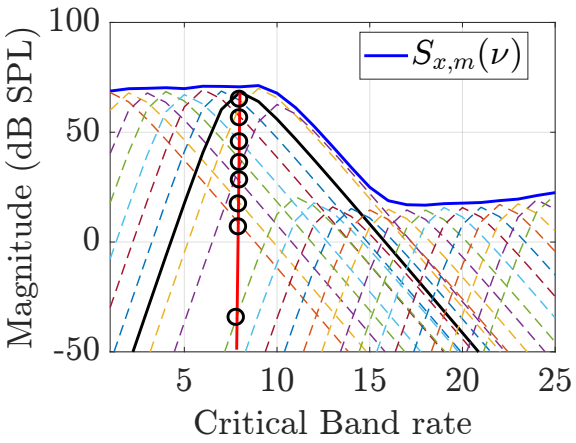


Figure 3.4. "Additivity of masking" for the critical band $\nu = 8$.

As Figure 3.4 shows, the "additivity of masking" for a specific critical band is performed by adding all the components of the rest of critical bands in that specific band, represented through the circles in Figure 3.4. A softer decrease in the SPL level of $S_{x,m}(\nu)$ after the 10^{th} critical band can be appreciated compared to the energy per critical band shown in Figure 3.3. This behavior is the result of adding the different masking patterns, since as stated before, they spread across the rest of critical bands, specially the closest ones, as Figure 3.4 shows.

After performing the "additivity of masking", the process performed by the upper branch of Figure 3.1 is completed and the overall masking curve, $S_{x,m}(\nu)$, is obtained.

3.3 Tonality Estimator Model

3.3.1 Background

In this section, the bottom branch of Figure 3.1 is described. According to [103], the overall masking curve, $S_{x,m}(\nu)$, suffers a decrease in the energy that depends on the tonal features of the masker signal. This effect can be appreciated in Figure 2.15 (and the corresponding explanation in Section 2.4.3).

Multiple studies of the masking of signals provide results where masking is different even when the different maskers lie in the same critical band and they have identical SPL levels. Some of these examples are described in [105, 120, 121] where a comparative study between tone signals masking noise signals and vice-versa is performed. Their experimental results conclude that masking does not present a symmetrical behavior, that is, narrowband noise signals and tone signals do not mask equally even though they have the same SPL level and they are located at the same critical band.

Considering the proposed model, the effect of the asymmetry of the masking is estimated through the bottom branch of Figure 3.1. Although [6] explains that tonality influence should be obtained through empirical test, nowadays different methods are implemented to estimate the tonality in the spectral masking threshold model. One of the most used methods at present was designed by Johnston [111] where an estimated tonal factor produces an offset per critical band used to estimate the lost energy of $S_{x,m}(\nu)$. Similar approaches to the method implemented by Johnston have been used in perceptual audio coding [119, 122], perceptual speech enhancement [113, 114] and music equalization in presence of a noise [93, 94, 123]. Although this approach has been widely used, it can be inaccurate for applications involving complex noise signals [120, 124] or involving speech and high frequency noise signals [125]. Alternative methods can be found in [124, 126] where tonality is based on different methods such as the temporal envelope rate (TE-R) method, the linear prediction method or the "Auditory Image Correlation" (AIC) method where tonality is estimated through a time-domain approach that analyzes the envelope fluctuations according to the auditory system.

In this dissertation, two different approaches to estimate tonality in the spectral masking model are presented and compared between them with the

purpose of studying which one performs a more realistic spectral masking effect. The first method is described in [111] and it is the most commonly method used in this spectral masking threshold model. The second one is based on the method explained in [127] and it considers several features of the masker sound. Each one estimates a tonal factor, $\mu_{x,m}$ in Figure 3.1, in order to use it in the same "Offset Function". The estimation of this tonal factor is explained in detail for each method in Sections 3.3.2 and 3.3.3.

3.3.2 Spectral Flatness Method

This method was originally described in [111]. It estimates the tonal factor, $\mu_{x,m}$, from the so-called Spectral Flatness Measure (SFM), thus, we have denoted this method as the "Spectral Flatness" (SF) method.

To estimate $\mu_{x,m}$, the two steps shown in Figure 3.5 have to be carried out. The "SFM Estimator" block computes the SFM, denoted by $\Upsilon_{x,m}$, as the ratio between the geometric and arithmetic mean of $P_{x,m}(k)$ of the masker signal [128]:

$$\Upsilon_{x,m} = 10\log_{10}\left(\frac{\left[\prod_{k=1}^{N_{\mathrm{FFT}}} P_{x,m}(k)\right]^{\frac{1}{N_{\mathrm{FFT}}}}}{\frac{1}{N_{\mathrm{FFT}}}\sum_{k=1}^{N_{\mathrm{FFT}}} P_{x,m}(k)}\right). \tag{3.6}$$

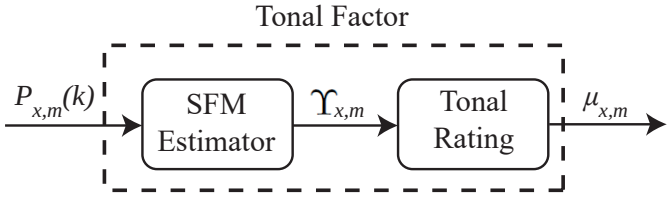


Figure 3.5. Block diagram of the SF method to estimate the tonal factor, $\mu_{x,m}$.

According to the definition of the SFM parameter, their values range between two boundaries. Its highest level is 0 dB, corresponding to a white noise, and its lowest level is $-\infty$ dB, which corresponds to the case of a

tone signal. According to this behavior, a masker signal would represent a greater "whiteness" in its spectrum as its SFM level becomes closer to 0 dB. Once the SFM is estimated by (3.6), the "Tonal Rating" is computed in order to obtain the tonal factor:

$$\mu_{x,m} = \min\left(\frac{\Upsilon_{x,m}}{\Upsilon_{1\text{k}60}}, 1\right), \tag{3.7}$$

where $\Upsilon_{1\text{k}60}$ corresponds to the SFM of a tone of 60 dB SPL located at 1 kHz. In [111], this constant is set as $\Upsilon_{1\text{k}60} = -60$ dB. Basically, the expression compares the SFM of the masker signal with the SFM of the tone signal, leading a value between 1 (a tone signal) and 0 (a white noise signal). Therefore, the masker signal provides a more noise-like tonality as $\mu_{x,m}$ gets closer to 0 and a tone-like tonality as $\mu_{x,m}$ gets closer to 1.

3.3.3 Aures Method

The second method to obtain the tonal factor that we will study here is based on the Aures method [127]. This method provides an elaborated estimation of the tonality since several perceptual characteristics of the masker signal are considered to calculate $\mu_{x,m}$.

The block diagram with all the steps performed in the method is shown in Figure 3.6, and it based in the block diagrams depicted in [127, 129, 130]. As it can be seen in Figure 3.6, $\mu_{x,m}$ is obtained at the end of the process through two main branches. The upper branch considers the frequency, bandwidth and SPL of $x_m(n)$ (through $P_{x,m}$, that has been estimated in the "Spectral Analysis" block, see Section 3.1), while the lower branch takes into account its loudness.

Due to the high complexity of the method, in the following paragraphs a detailed explanation of the steps shown in Figure 3.6 is included, where for the sake of simplicity, the subscript m, denoting the m^{th} frame, is omitted.

Extraction of the tonal components

According to Figure 3.6, the first step is the extraction of the tonal and noise components of the masker signal, $x(n)$. This step is based on the pitch extraction algorithm modeled by Terhardt [131, 132] that uses the estimated power spectrum, $P_x(k)$, of the masker signal. According to this method, all the frequency bins of $P_x(k)$ are analyzed to determine whether

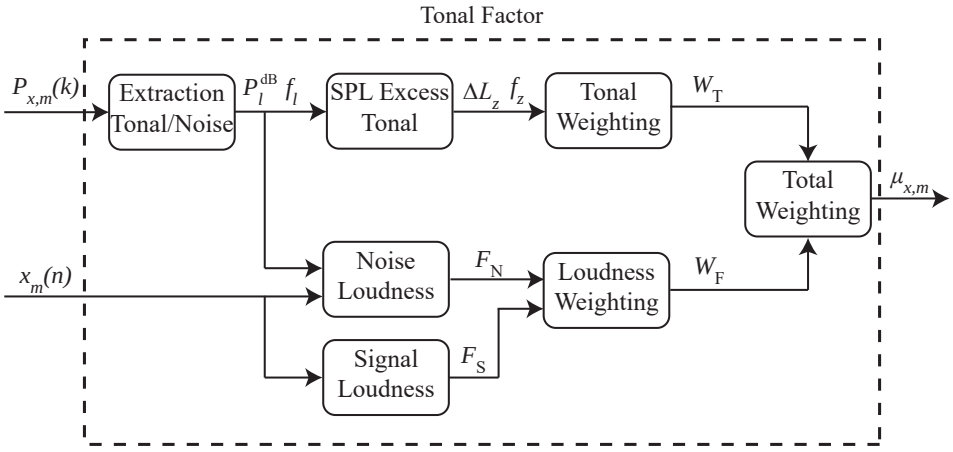


Figure 3.6. Block diagram of the Aures method to estimate the tonal factor,$\mu_{x,m}$.

they satisfy the following two conditions:

$$P_x^{\mathrm{dB}}(k-1) < P_x^{\mathrm{dB}}(k) \geq P_x^{\mathrm{dB}}(k+1)\,, \tag{3.8a}$$

$$P_x^{\mathrm{dB}}(k) - P_x^{\mathrm{dB}}(k \pm \lambda) \geq T_{\mathrm{H}}\,, \tag{3.8b}$$

where $P_x^{\mathrm{dB}}(k)$ is $P_x(k)$ expressed in dB units, T_{H} is a threshold stated to detect tonal components and λ represents the neighboring components to be checked, excluding ± 1 since the first condition already considers them. Therefore, the k^{th} component of $P_x^{\mathrm{dB}}(k)$ will be considered "tonal" if the following two conditions are fulfilled:

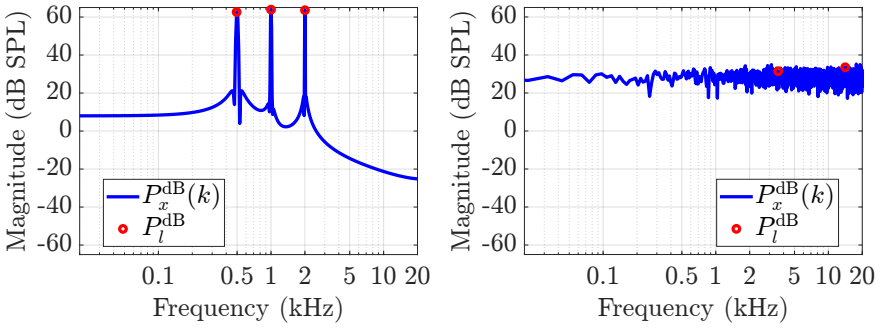
1. It must be the largest component considering its nearest neighbors.
2. It must be, at least, T_{H} dB larger than its $\pm\lambda$-separated neighboring components, with $\lambda = 2, 3, \ldots$

Both λ and T_H are estimated according to the frequency resolution used to estimate $P_x^{\mathrm{dB}}(k)$. The original method proposed by Aures [132] used a frequency spacing of 12.5 Hz, a set of λ values given by $\lambda = \{2, 3\}$

and a threshold of $T_{\mathrm{H}} = 7$ dB, where this last parameter was empirically set from the analysis of different sounds [132].

In this dissertation, we have used a different frequency resolution, which results in different values of λ and T_{H}. Using a sample rate of $f_{\mathrm{s}} = 44100$ Hz, a value of $N_{\mathrm{FFT}} = 4096$ samples is required in order to estimate efficiently $P_x(k)$ and obtain a similar frequency resolution than the resolution used in [132]. More specifically, through these parameters the frequency resolution is $\frac{f_{\mathrm{s}}}{N_{FFT}} = 10.76$ Hz. Regarding the neighborhood range in (3.8), the maximum value of $\lambda = 3$ in the original method produced a maximum frequency separation of $3 \times 12.5 = 37.5$ Hz. For this case, a range of $\lambda = \{2, 3, 4\}$ has been used, obtaining a maximum frequency separation of $4 \times 10.76 = 43.06$ Hz. Similarly to procedure in [132], the value of the threshold, T_{H} in (3.8), has been obtained through empirical test resulting in a value of $T_{\mathrm{H}} = 5.5$ dB.

As shown in Figure 3.6, the SPL level (P_l^{dB}) and the frequency (f_l) of the tonal components are obtained at the output of this block. The index l indicates the tonal component ranging from $l = 1, \ldots, C_{\mathrm{L}}$ where C_{L} is the number of tonal components found. An illustrative example of detected tonal components is shown in Figure 3.7 for a multi-tone signal (left) and a white noise signal (right).



(a) Multi-tonal signal of three tones. **(b)** White Gaussian noise.

Figure 3.7. Tonal components detected for two different signals.

As Figure 3.7 shows, in this case the algorithm is able to detect all the tonal components when a multi-tonal signal is analyzed, while it detects

two tonal components in the white noise signal. This behavior is caused because of the randomization of samples that compose the noise. We have observed that the number of tonal components detected in flat spectrum signals decreases when T_{H} is increased. However, an excessive level of the threshold can decrease significantly the tonal components in complex broadband signals. For this reason, T_{H} should not be overestimated in order to accurately detect the tonal components in any type of signal.

Sound Pressure Level Excess (SPL Excess)

According to [132], once the tonal components have been found, the next step is to estimate which ones are aurally relevant, that is, which ones can be appreciated. For this purpose, the SPL excess, denoted by ΔL, is estimated for each tonal component l detected in the previous step:

$$\Delta L_l = P_l^{\mathrm{dB}} - 10\log_{10}\left(\left[\sum_{\substack{\gamma=1\\ \gamma\neq l}}^{C_{\mathrm{L}}} 10^{\frac{L_{\mathrm{E}\gamma}(f_l)}{20}}\right]^2 + I_{\mathrm{N},l} + 10^{\frac{L_{\mathrm{TH}}(f_l)}{10}}\right), \tag{3.9}$$

where $L_{\mathrm{E}\gamma}(f_l)$ is the excitation level caused over the l^{th} tonal component due to the tonal component γ, $I_{\mathrm{N},l}$ is the noise intensity in the critical band where the l^{th} tonal component is located and finally, $L_{\mathrm{TH}}(f_l)$ is the level of the hearing threshold at the frequency f_l.

The excitation level in (3.9) can be modeled as [132]:

$$L_{\mathrm{E}\gamma}(f_l) = P_\gamma^{\mathrm{dB}} - \rho(f_\gamma, f_l)(\nu_\gamma - \nu_l)\,, \tag{3.10}$$

where P_γ^{dB} is the SPL level of the tonal component γ expressed in dB units, ν_γ and ν_l are the frequencies of the γ^{th} and l^{th} tonal components respectively, expressed in the Bark domain through (3.1), and the parameter $\rho(f_\gamma, f_l)$ specifies the masking pattern and it is expressed as (in dB/Bark):

$$\rho(f_\gamma, f_l) = \begin{cases} 27 & \text{if } f_l \leq f_\gamma \\ -24 - \frac{230}{f_\gamma} + 0.2P_\gamma^{\mathrm{B}} & \text{if } f_l > f_\gamma\,. \end{cases} \tag{3.11}$$

The noise intensity, $I_{\mathrm{N},l}$ in (3.9), is obtained through the addition of all the components located within the critical band defined between $\nu_l - 0.5$ and $\nu_l + 0.5$, without including the tonal components l and their neighbors, defined by (3.8b). In addition, we have considered suitable to remove the influence of the other tonal components (such as γ) in this term if these components are located in the critical band defined between $\nu_l - 0.5$ and $\nu_l + 0.5$ since their impact is already considered through (3.10). Finally, the hearing threshold, L_{TH}, of (3.9) in dB units can be obtained as [131]:

$$L_{\mathrm{TH}}(f_l) = 3.64\left(\frac{f_l}{1000}\right)^{-0.8} - 6.5e^{-0.6\left(\frac{f_l}{1000}-3.3\right)^2} + 10^{-3}\left(\frac{f_l}{1000}\right)^4. \quad (3.12)$$

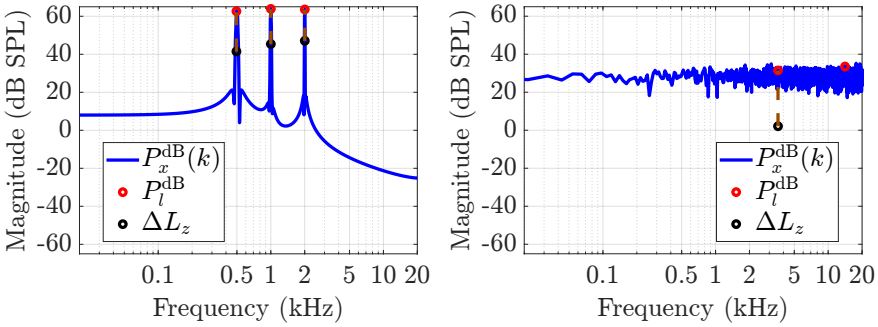
The tonal components with a SPL excess such that $\Delta L > 0$ will be considered as aurally relevant tonal components. Since not all the tonal components l might be considered relevant, a new subscript is considered, such that:

$$\Delta L_z = \Delta L_l\,, \quad \text{for} \quad \Delta L_l > 0\,, \quad (3.13)$$

where z represents the index of the relevant tonal components, ranging from $z = 1, \ldots, C_{\mathrm{Z}}$ with C_{Z} as the number of relevant tonal components found. Similar to the previous step, the SPL excess (ΔL_z) and the frequency (f_z) of the relevant tonal components are obtained in this step.

The process described above is illustrated in Figure 3.8 where same signals than Figure 3.7 has been analyzed in order to appreciate the effect of this step over the tonal components. As Figure 3.8 shows, the red circles represent the SPL levels of the candidates to be relevant components (denoted through l), whereas the black circles show the SPL excess, ΔL, of the remaining relevant components (denoted by z).

As Figure 3.8 shows, for the multi-tonal signal on the left, all the tonal components are relevant. It can be seen that all the components exhibit a ΔL above 40 dB SPL. For the noise signal on the right, in this case, only one component is relevant, this means that the other component exhibit a value of ΔL lower than 0 dB SPL. Furthermore, although some tonal components are detected as relevant in noise signals, they have a low impact over the tonality measurement due to their small SPL excess value, unlike the signals composed by tones.



(a) Multi-tonal signal of three tones. **(b)** White Gaussian noise.

Figure 3.8. Estimation of the relevant tonal components (SPL excess).

Tonal Weighting

According to Figure 3.6, the tonal weighting is performed over the relevant tonal components considering their bandwidth in the Bark domain ($\Delta\nu_z$), frequency (f_z) and SPL excess magnitude (ΔL_z). The tonal weighting, W_{T}, can be estimated as [127]:

$$W_{\mathrm{T}} = \sqrt{\sum_{z=1}^{C_{\mathrm{Z}}} \left[\omega_1(\Delta\nu_z)\,\omega_2(f_z)\,\omega_3(\Delta L_z)\right]^2}\,, \tag{3.14}$$

where the weighting functions denoted by $\omega_n(\cdot)$ for $n = 1, 2, 3$ are defined as [127]:

$$\omega_1(\Delta\nu_z) = \left(\frac{0.13}{\Delta\nu_z + 0.13}\right) \tag{3.15a}$$

$$\omega_2(f_z) = \left(\frac{1}{\sqrt{1 + 0.2\left(\frac{f_z}{700} + \frac{700}{f_z}\right)^2}}\right) \tag{3.15b}$$

$$\omega_3(\Delta L_z) = \left(1 - e^{\frac{-\Delta L_z}{15}}\right)\,, \tag{3.15c}$$

where ω_1 has been modified with respect to the original method [127, 133],

$\omega_1(\Delta\nu_z) = \left(\frac{0.13}{\Delta\nu_z+0.13}\right)^{\frac{1}{0.29}}$. This innovation has been done to be consistent with the other two weighting functions, ω_2 and ω_3.

Loudness Weighting

In the lower branch of Figure 3.6, the Aures method considers the effect of the loudness (in sones, see expression (2.3)) of the tonal components. For this purpose two types of loudness are calculated: the "Signal Loudness" , F_{S}, that computes the loudness of $x_m(n)$, and the "Noise Loudness", F_{N}, that computes the loudness of the noisy part of $x_m(n)$. The noisy part is the remaining signal once the l tonal components have been eliminated. Both loudness are estimated according to the Zwicker method (ISO 532) [97].

In order to eliminate these tonal components, IIR (Infinite Impulse Response) notch filters of second order are designed for each frequency, f_l, with a minimum attenuation of 30 dB in the suppressed band. These filters must be adapted to the features of the tonal components, more specifically to its bandwidth in order to remove their contribution as much as possible. For this reason, the bandwidth of all the IIR filters is set to be equal to the bandwidth of the tonal components, that is:

$$\mathrm{BW} = \left\lceil \frac{2\lambda_{\mathrm{M}} N_{\mathrm{FFT}}}{f_{\mathrm{s}}} \right\rceil , \tag{3.16}$$

where BW is expressed in hertz and λ_{M} corresponds to the maximum value of λ, that in this case stands for $\lambda_{\mathrm{M}} = 4$ (see (3.8) and its corresponding explanation).

Then, $x(n)$ is filtered through the cascade of notch filters, obtaining the noisy signal in time domain, $x_{\mathrm{N}}(n)$. Once $x_{\mathrm{N}}(n)$ is obtained, its loudness level, F_{N}, is computed [97] and the loudness weighting is obtained as:

$$W_{\mathrm{F}} = 1 - \frac{F_{\mathrm{N}}}{F_{\mathrm{S}}}. \tag{3.17}$$

Considering the ratio between F_{N} and F_{S} as the loudness percentage of the noisy part of the m^{th} frame of $x(n)$, it can be stated that W_{F} represents the loudness percentage of its tonal part.

Total Weighting

Finally, the tonal factor (μ_x) for the m^{th} frame of $x(n)$ is defined as the combination of the Tonal Weighting (3.14) and the Loudness Weighting (3.17):

$$\mu_x = C_{1\text{k}60} W_{\text{T}}^{0.29} W_{\text{F}}^{0.79} \,, \tag{3.18}$$

where $C_{1\text{k}60}$ is a calibration variable that sets $\mu_x = 1$ for a pure tone of 60 dB SPL located at 1 kHz. In [130], this constant is set as $C_{1\text{k}60} = 1.09$ under ideal conditions. The exponents of W_{T} and W_{F} are correction factors introduced in the model according to empirical experiments on the matter [127].

In order to clarify all the operations carried out by the Aures method to obtain (3.18), a pseudocode is provided in Algorithm 1, where the equations related to each step are given as a comment at the end of the corresponding lines.

3.3.4 Offset Function

Once the tonal factor has been estimated, either by the SF method (3.7) or the Aures method (3.18), the next step to perform is the "Offset Function".

According to [103] a pure tone signal located at the ν^{th} critical band suffers a shift (or offset) of $(14.5 + \nu)$ dB over its overall masking curve $S_{x,m}(\nu)$, while the offset suffered by the overall masking curve of a white noise ranges between 3 and 6 dB, usually considered a constant value of 5.5 dB [94, 111]. This offset can also be estimated even for masker signals that are not completely tone-like or noise-like. For this purpose, $\mu_{x,m}$ is used to geometrically weight the above mentioned thresholds, as follows [111]:

$$O_{x,m}(\nu) = \mu_{x,m}(14.5 + \nu) + (1 - \mu_{x,m})5.5 \text{ dB} \,, \quad \nu = 1, \ldots, N_{\text{c}} \,. \tag{3.19}$$

Finally, according to the last step of Figure 3.1, the spectral masking threshold (in dB units) for every frame m of the masker signal $x(n)$ is obtained as [94]:

$$T_{x,m}(\nu) = S_{x,m}(\nu) - O_{x,m}(\nu) \,, \quad \nu = 1, \ldots, N_{\text{c}} \,. \tag{3.20}$$

Algorithm 1 Aures tonality method.

Initialize: $f_\mathrm{s} = 44.1$ kHz; $N_\mathrm{FFT} = 4096$
Initialize: $\lambda = 2, 3, 4$, $\lambda_\mathrm{M} = 4$; $\mathrm{T_H} = 5.5$ dB; $l = 0$; $z = 0$
Initialize: $C_\mathrm{1k60} = 1.09$; $W_\mathrm{T} = 0$; $W_\mathrm{F} = 0$
for all freqbins $(\lambda_\mathrm{M} + 1) \leq k \leq \left(\frac{N_\mathrm{FFT}}{2}\right) - (\lambda_\mathrm{M} + 1)$ **do**
    if $P_x^\mathrm{dB}(k) \geq P_x^\mathrm{dB}(k \pm 1)$ and $P_x^\mathrm{dB}(k) - P_x^\mathrm{dB}(k \pm \lambda) \geq \mathrm{T_H}$ **then** ▷ see (3.8)
        $l = l + 1$
        $P_l^\mathrm{dB} = P_x^\mathrm{dB}(k)$; $f_l = \left\lceil \frac{k}{N_\mathrm{FFT}} f_\mathrm{s} \right\rceil$
    end if
end for
if $l > 0$ **then**
    for all TonCompon $1 \leq l \leq C_\mathrm{L}$ **do**
        Compute $L_{\mathrm{E}\gamma}(f_l)$ ▷ (3.10)-(3.11)
        Compute $I_{\mathrm{N},l}$
        Compute $L_\mathrm{TH}(f_l)$ ▷ see (3.12)
        Obtain ΔL_l ▷ see (3.9)
        if $\Delta L_l > 0$ **then**
            $z = z + 1$
            $\Delta L_z = \Delta L_l$; $f_z = f_l$
        end if
    end for
    if $z > 0$ **then**
        for all RelevantCompon $1 \leq z \leq C_\mathrm{Z}$ **do**
            Compute BW ▷ see (3.16)
            Map BW to $\Delta\nu_z$ ▷ see (3.1)
            Compute $w_1(\Delta\nu_z)$; $w_2(f_z)$ and $w_3(\Delta L_z)$ ▷ see (3.15)
        end for
        Obtain W_T ▷ see (3.14)
    end if
    Compute loudness of $x(n)$, F_S ▷ Zwicker loudness [97]
    for all TonCompon $1 \leq l \leq C_\mathrm{L}$ **do**
        Design notch filter for f_l
    end for
    Compute $x_\mathrm{N}(n)$ as $x(n)$ filtered out by the C_L notch filters in cascade
    Compute loudness of $x_\mathrm{N}(n)$, F_N ▷ Zwicker loudness [97]
    Obtain W_F ▷ see (3.17)
end if
Obtain μ ▷ see (3.18)

3.3.5 Subjective Experiments

In this section a subjective test has been carried out in order to validate the performance of the different tonality methods described in Sections 3.3.2 and 3.3.3. However, as stated in Section 3.3.3, the original Aures method has been slightly modified from the mathematical point of view, as Table 3.1 shows, where the motivation of the modifications carried out are presented. As a result, in this section three different tonality methods have been compared:

1. The SF method explained in Section 3.3.2 that it is commonly used in the masking threshold estimation [7, 99].
2. The original Aures (OA) method that estimates the tonal factor from the approach seen in [127].
3. The proposed improved Aures (IA) method that estimates the tonal factor according to the procedure seen in Section 3.3.3.

In order to compare these tonality methods, the spectral masking threshold, $T_{x,m}(\nu)$, has been computed for each tonality method as the average value of (3.20) over all the time frames, and compared to the average masking threshold obtained in the subjective test. Therefore, the four averaged masking thresholds that will be referred to along this section are:

- $\tilde{T}_x(\nu)$: Obtained from the subjective test.
- $T_x^{\mathrm{SF}}(\nu)$: Obtained by the SF method (3.7).
- $T_x^{\mathrm{IA}}(\nu)$: Obtained by IA method (3.18), according to the values given in Table 3.1.
- $T_x^{\mathrm{OA}}(\nu)$: Obtained by the OA method (3.18), according to the values given in Table 3.1.

Since the main difference between $T_x^{\mathrm{IA}}(\nu)$, $T_x^{\mathrm{OA}}(\nu)$ and $T_x^{\mathrm{SF}}(\nu)$ lies on the calculation of the $\mu_{x,m}$ parameter, the overall masking curve $S_{x,m}(\nu)$, obtained in (3.5), and the offset function, $O_{x,m}(\nu)$, obtained in (3.19), will be the same for the different methods considered. Additionally, for

Feature	OA Method	IA Method	Motivation
Frequency resolution (Hz)	12.5	10.76	FFT analysis carried out through the efficient FFT. FFT size as power of 2.
FFT size (samples)	3528	4096	
Tonal Threshold, T_{H} (dB)	7	5.5	This value is a compromise in order to detect most of the single-tone signals as tonal components, but only a few in noise signals.
Neighboring separation, λ (samples)	3	4	Similar bandwidth to the OA method, but using the frequency resolution given by the FFT.
Tonal bandwidth (Hz)	75	86.08	
Exponent of ω_1 (3.15a)	$(0.29)^{-1}$	1	In correspondence with the other two weighting functions, see (3.15b) and (3.15c).

Table 3.1. Differences between the OA method and the proposed IA method.

this comparison, the overall masking curve, $S_{x,m}(\nu)$, has been generated through the masking pattern of (3.3) with $\alpha = 1$.

In order to describe in detail the subjective test, the main features as well as the results obtained are explained in the following paragraphs.

Generation of the stimuli

We have designed two sets of sounds to be used in the subjective tests: multi-tonal signals and narrowband noise signals. Each stimuli is formed by the combination of a multi-tonal signal and a noise signal, as shown in Table 3.2. On one side, three different multi-tonal signals, each one composed by three different tones between 350 and 5800 Hz have been used. Additionally, nine narrowband noise signals have been generated with a bandwidth of one critical band (whose center frequency is shown in Table 3.2). These signals can be considered stationary, thus the mean value of the estimated masking threshold over time, $\tilde{T}_x(\nu)$, can be considered unbiased for long enough frames. As it can be appreciated from Table 3.2, the chosen frequencies for both types of signals are the center frequency of certain critical bands. We have tried to use a wide part of the perceptual spectrum, covering eight of the twenty-five critical bands, particularly those more sensitive according to the human audibility threshold.

	Center Frequency (Hz)		
Stimuli #	Multi-tonal signal	Noise signal	Critical Band
1	700, 840, 1000	700	
2	700, 840, 1000	840	7, 8, 9
3	700, 840, 1000	1000	
4	350, 450, 5800	350	
5	350, 450, 5800	450	4, 5, 20
6	350, 450, 5800	5800	
7	1000, 2150, 3400	1000	
8	1000, 2150, 3400	2150	9, 14, 17
9	1000, 2150, 3400	3400	

Table 3.2. List of stimuli generated for the subjective test.

There are only three different multi-tonal signals, as shown Table 3.2. The first multi-tone signal is formed by three tones lying in consecutive critical bands, the seventh, eighth and ninth, whereas the other two multi-tonal signals are formed by three tones spread along the frequency spectrum. In this way, the design of the subjective test takes into account a diverse set

of complex signals with a different energy distribution along the frequency spectrum. The 1000-Hz tone has been used in two multi-tonal signals, once as the highest tone and once as the lowest, in order to analyze the masking produced on a single critical band by their adjacent (pre and post) critical bands within a complex sound.

Both set of signals, multi-tonal and noise signals, have been generated with a sample rate of $f_s = 44100$ Hz. Each of the tones that compose the multi-tonal signals were generated as a zero initial phase sine and all having the same amplitude. The narrowband noise signals were generated as the output of IIR band-pass filters centered at the frequencies shown in the second column of Table 3.2, being their inputs white Gaussian noise, and with a minimum attenuation of 60 dB in the stop bands. The stop bands of these filters started at the center frequencies of their adjacent critical bands, thus, each filter had a bandwidth of one critical band.

Apparatus and design

The perceptual test was carried out inside the listening room of the Audio Processing laboratory of the Institute of Telecommunications and Multimedia Applications (iTEAM) [134]. The reproduction system, shown in Figure 3.9, was formed by a M-Track Quad sound card and a pair of Sennheiser HD 600 headphones connected to a laptop. The laptop is equipped with an Intel i7 processor, 8 GB of RAM and a NVIDIA GeForce 940MX graphics card.

The reproduction system was calibrated in order to provide 60 dB SPL for a single tone signal located at 1 kHz, resulting in a digital amplitude of -14.7 dB with respect to the same tone ranging the full digital scale of $[-1, 1]$. Additionally, the signals shown in Table 3.2, have been weighted in loudness with the loudness of the tone signal used for calibrating the system in order to make the perceptual test as comfortable as possible. For this purpose, firstly the loudness of a tone of 60 dB SPL located at 1 kHz has been estimated (through the Zwicker method [97]). Then, the loudness of each signal shown in Table 3.2 is estimated and matched to the loudness of the tone at 1 kHz. Both, the calibration and the perceptual test were implemented on Matlab [135].

Finally, as Section 3.3.2 explains, the SF method requires the SFM value of a tone signal of 60 dB SPL located at 1 kHz ($\Upsilon_{1\text{k}60}$, see (3.7)). Therefore, in order to perform an accurate comparison with the SF method,

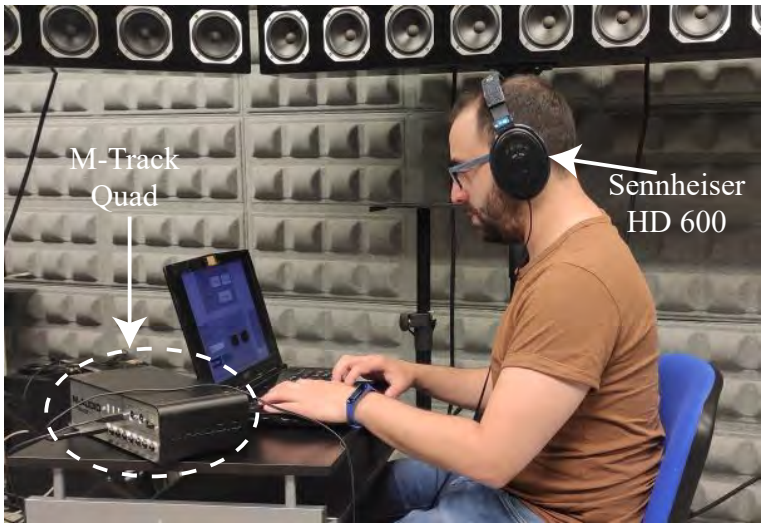


Figure 3.9. System used in the tonality study.

this value must be calculated for the system described previously. Additionally, M_{T} and M_{S} (that is equal to N_{FFT}) must be indicated in order to determine the frequency resolution of $P_{x,m}(k)$, that affects to Υ_{1k60} as well. Since the IA method requires $N_{\mathrm{FFT}} = 4096$ samples (see Section 3.3.3), for this study we will consider that value for the IA and SF methods, that is, $M_{\mathrm{S}} = N_{\mathrm{FFT}} = 4096$ samples, while for the OA method, this value is set to $N_{\mathrm{FFT}} = 3528$ samples in order to obtain a frequency resolution of 12.5 Hz (as Table 3.1 indicates). In addition, the power spectrum, $P_{x,m}(k)$, has been estimated by averaging four frames, that is, $M_{\mathrm{T}} = 4M_{\mathrm{S}}$. Considering these features, we obtain that $\Upsilon_{\mathrm{1k60}} = -54$ dB.

Participants

Nineteen people participated in the subjective evaluation. Before starting the subjective test, an audiometry was carried out at each participant using the same reproduction system available for the test. For this purpose, we also programmed an ad-hoc application in Matlab. In this way, the audibility threshold of each participant was assessed in order to assure that all of them presented normal hearing.

Inspection of the results revealed some subjects whose results appeared

to be outliers and their results were excluded from the statistical analysis. Consequently, only the assessment given by 16 subjects have been considered for the results. Therefore, the jury panel was formed by 7 males, 9 females, aged between 18 and 50 years, although most of the participants aged between 24 and 30 years, and only three of them were familiarized with the psychoacoustic research field.

Procedure

The perceptual test aims to obtain the masking threshold, $T_{x,m}(\nu)$, of the multi-tonal signal at the specific critical band where noise signals are located, thus, one perceived masking threshold is estimated per stimuli. For this purpose, the multi-tonal signals are reproduced with a constant SPL while the noise levels of each stimuli can be increased or decreased by the subject as it will be described in the following.

The perceptual test is performed through an application designed in Matlab, whose main interface is shown in Figure 3.10. At the bottom left, there is the possibility of selecting the stimuli to be played (labelled as "Signal 1" in the figure). At the bottom center, there are two buttons that allow to increase ("+") or decrease ("-") the SPL level of the narrowband noise. The application also provides two examples of multi-tonal signals with narrowband noise signals (none of them used in the real test) in order to introduce the participant into the dynamic of the test. Both examples can play the multi-tonal signal alone ("Play Clean") or in addition with the narrowband noise ("Play dirty").

Each participant manages the interface shown in Figure 3.10 in order to set the spectral masking threshold for each stimuli following the flow diagram of Figure 3.11. The different options of this flow diagram are described in the following:

1. The participant starts the test by selecting one of the stimuli from the list at the bottom left of Figure 3.10. There are nine stimulus labeled from "Signal 1" to "Signal 9" corresponding to those of Table 3.2.

2. Then, the participant must press "Play Signal" to reproduce the stimuli selected. The multi-tone is presented with a fixed level, while the noise signal is presented with an attenuation of 40 dB with respect to the SPL of a noise whose loudness would be equal to that of the multi-tonal signal. In this way, we assure the noise is masked.

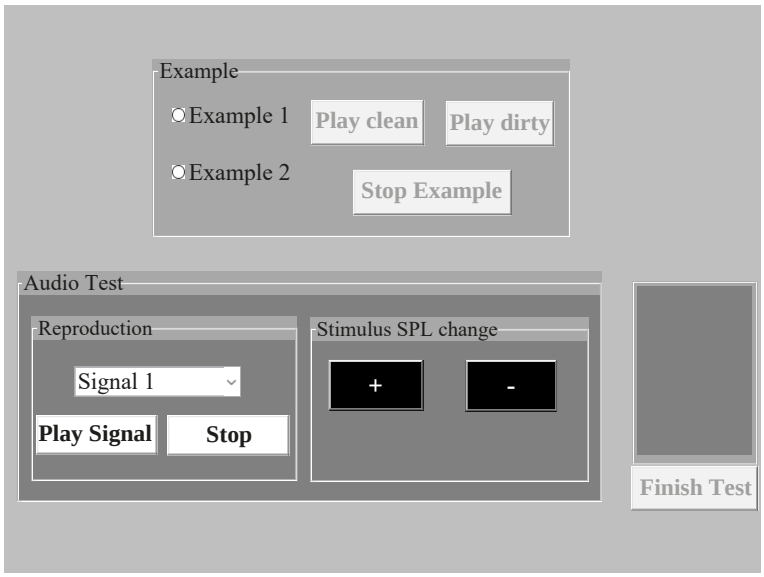


Figure 3.10. Matlab interface specifically designed to run the subjective tests.

3. Then, the participant presses the plus ("+") button as many times as necessary in order to make the noise audible. At this point, the SPL level of the noise increases in steps of 5 dB each time the button "+" is pressed.

4. Once the participant is able to hear the added noise, the participant presses the minus ("-") button as many times as necessary in order to make the noise inaudible again. The first time that the "-" button is pressed, the noise level is decreased by 3 dB. The second and successive times that the button "-" is pressed, the level values decrease in steps of 2 dB. This reduction of the step size as the number of selections increases is a common technique in subjective tests whose goal is the estimation of a threshold, as in the hearing in noise test (HINT) [136], or as in the perceptual test used in Lutfi [116].

5. The participant repeats steps 3 and 4 within a loop making the noise almost audible through step 3 and almost inaudible through step 4. However, at this point, every press of the plus ("+") or minus ("-")

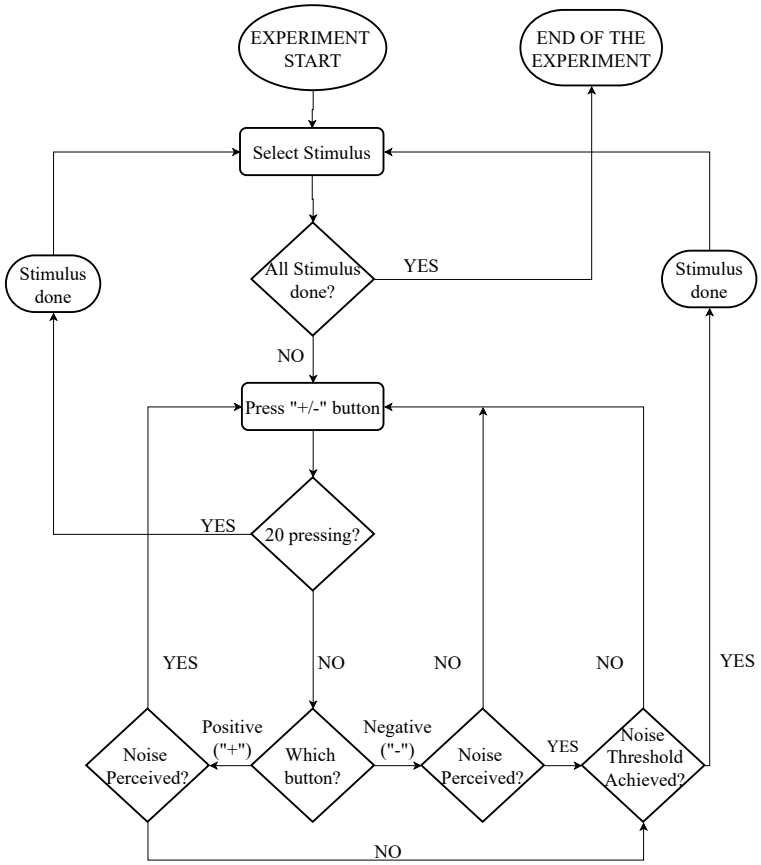


Figure 3.11. Flow diagram of the steps followed by any participant when they carried out the subjective test.

buttons means an increase or decrease of the noise SPL of only 2 dB respectively. The loop terminates when the participant presses the "+" (or "-") button once and then presses the "-" (or "+") button once, or alternatively when the participant have pressed any "+" or "-" button 20 times.

6. Once the evaluation of the stimuli has finished, the noise signal is saved and the application allows for the selection of a new stimuli, going to step 1.

7. Once all the stimuli have been evaluated by the participant, the "Finish Test" button at the bottom right of Figure 3.10 is enabled in order to finish the subjective test.

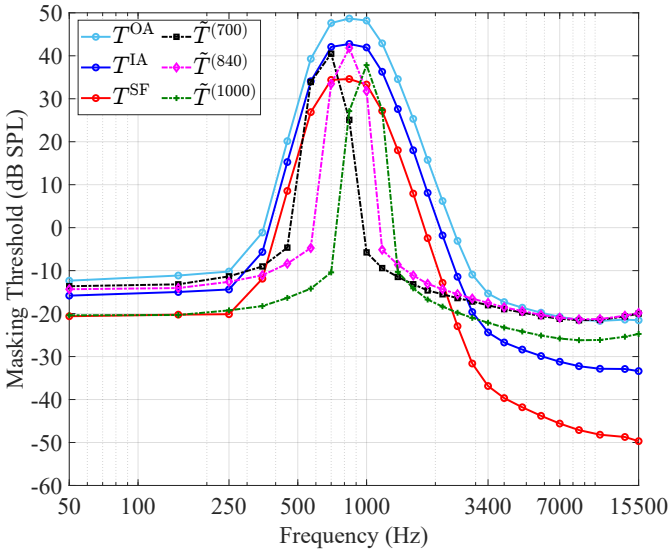
The final SPL level of the narrowband noise obtained in step 6 will indicate the level of the masking threshold produced by that particular multi-tonal signal over the critical band covered by the noise. More specifically, the energy of the noise $E_x(\nu)$ is obtained for that critical band through the average of (3.2) over time, and the masking threshold is set as $T_x(\nu) = E_x^{\mathrm{dB}}(\nu)$. Finally, the subjective masking threshold, $\tilde{T}_x(\nu)$, is obtained as the average value over the sixteen participants for every stimuli and critical band shown in Table 3.2.

Results and discussion

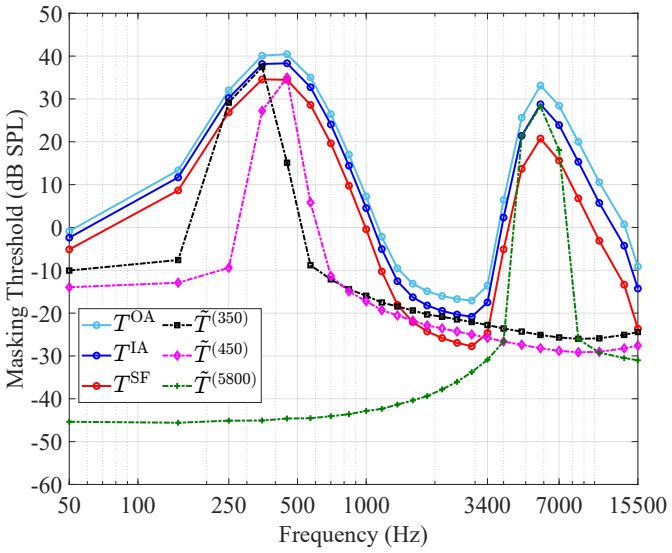
The results of the subjective test in comparison with the SF, the OA and the IA methods are shown in Figure 3.12. For all the sub-figures, the SF masking threshold, T^{SF}, is represented by a red line, the IA masking threshold, T^{IA}, is represented by a blue line and the OA masking threshold, T^{OA}, is represented by the cyan line. Additionally, each sub-figure represents three different masking thresholds obtained for the same multi-tonal signal of Table 3.2. The masking thresholds obtained from the perceptual test are labeled as $\tilde{T}^{(f_n)}$, where f_n indicates the center frequency of each of the noise signals used for each of the stimuli ($n = 1, \ldots, 9$) shown in Table 3.2.

As it was described above, the perceived masking threshold $\tilde{T}^{(f_n)}$ is obtained from the energy of the noise in the critical band where the noise is located. However, Figure 3.12 show the energy of the noise for the whole spectrum, although the only value of interest is their maximum, which coincides with the value of the perceived masking threshold $\tilde{T}^{(f_n)}$.

From Figure 3.12, it can be seen that T^{SF} provides a lower masking level than the masking levels obtained in the perceptual test. This means that T^{SF} always underestimates the masking level, specially in middle and high frequency ranges, and therefore the masking levels provided T^{SF} do not mask the noise in practice. The OA method produces the opposite performance than the SF method, that is, T^{OA} overestimates the masking level, specially for frequencies below 1 kHz, as it can be seen from Figure 3.12a. Regarding the IA method, T^{IA} provides the closest masking levels to those obtained in the perceptual test for most of the stimuli.

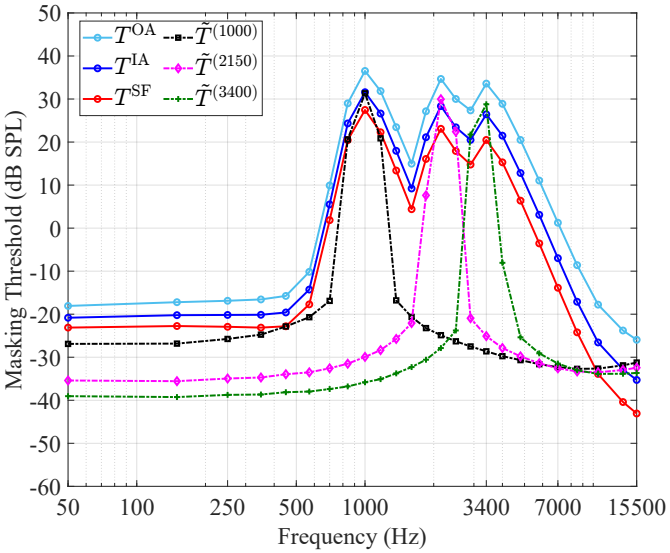


(a)



(b)

Figure 3.12. Masking thresholds of the SF (T^{SF}), OA (T^{OA}) and IA (T^{IA}) methods compared to the masking thresholds obtained in the subjective test for the (a) first, (b) second and (c) third multi-tonal signal.



(c)

Figure 3.12. (continued).

The difference between the masking levels provided by each method (SF, OA and IA) and the masking levels obtained by the perceptual test are shown in Figure 3.13, where the difference in dB units is defined as:

$$\epsilon(n) = T(\nu_n) - \tilde{T}^{(f_n)}\,, \tag{3.21}$$

where $n = 1, \ldots, 9$ corresponds to the stimulus of Table 3.2, f_n is the center frequency of the narrowband noise of the nth stimuli and ν_n is the critical band where f_n is located. The average differences are $\bar{\epsilon} = 5.8$ dB for the OA model, $\bar{\epsilon} = -5.3$ dB for the SF and $\bar{\epsilon} = 0.8$ dB for the IA. This error indicates the deviation of each method with respect to the perceptual results. On the other hand, the values of the mean absolute error are $|\bar{\epsilon}| = 5.8$ dB for the OA model, $|\bar{\epsilon}| = 5.3$ dB for the SF and $|\bar{\epsilon}| = 1.7$ dB for the IA. The difference of values in the IA method is caused because is the only method that provides positive and negative differences, as opposed to the OA and the SF method, as it is illustrated in Figure 3.13.

As Figure 3.13 shows, the masking levels of the three first stimulus (see Figure 3.12a) present a significant difference with the masking levels

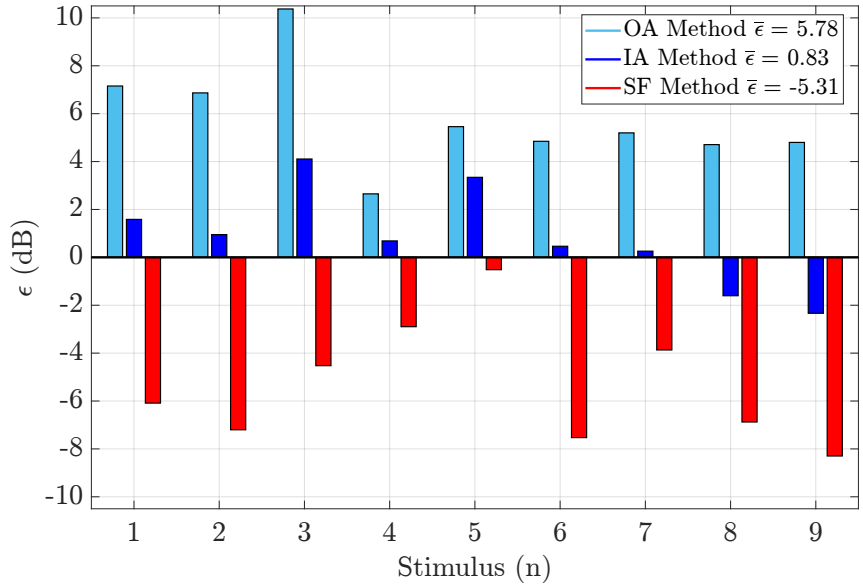


Figure 3.13. Difference in dB units as defined in (3.21) of the IA method (blue bars), the OA method (cyan bars) and the SF method (red bars) for the stimuli listed in Table 3.2.

obtained in the perceptual test, $\tilde{T}^{(f_n)}$, for the OA and SF methods. More specifically, T^{OA} levels are 6 dB above for the seventh and eighth critical bands, and around 10 dB above for the ninth critical band. For the SF method, the T^{SF} levels are 6 dB below for the seventh and eighth critical bands, and around 4 dB below for the ninth critical band. Regarding the IA method, the T^{IA} levels present a lower difference with the $\tilde{T}^{(f)}$ levels for the seventh and eighth critical band, around 1 dB above, while it is 4 dB greater for the ninth critical band.

The next three stimulus in the Figure 3.13 correspond to the multi-tonal signal shown in Figure 3.12b. The tonal components of this multi-tonal signal are 350 Hz, 450 Hz and 5800 Hz, corresponding to the fourth, fifth and twentieth critical bands. Notice that the first two tones lie in consecutive bands, but the third tone cover a separated Bark band. In addition, the first two tones are located at the low part of the spectrum where the human audibility threshold is higher [6], whereas the third tone

presents the highest frequency used in the test, increasing the contribution of the tonality (in a negative sense) to the masking. In this case, Figure 3.13 shows that the T^{SF} levels are 3 dB, 0.8 dB and 7.5 dB below the $\tilde{T}^{(f_n)}$ levels for the fourth, fifth and twentieth critical bands respectively. In contrast to the SF method, the T^{OA} levels are 3 dB, 5.7 dB and 5.3 dB above the $\tilde{T}^{(f_n)}$. Regarding the IA method, T^{IA} provides values slightly above of the $\tilde{T}^{(f_n)}$ levels. Specifically, the T^{IA} levels are 0.5 dB, 3 dB and 0.2 dB above the $\tilde{T}^{(f_n)}$ levels for the fourth, fifth and twentieth critical bands respectively.

Finally, the last three stimulus in the Figure 3.13 correspond to the multi-tonal signal shown in Figure 3.12c. The tonal components of this multi-tonal signal are 1000 Hz, 2150 Hz and 3400 Hz, corresponding to the ninth, fourteenth and seventeenth critical bands. The three tones are located at non-consecutive critical bands of the most sensitive area of the human hearing system. As Figure 3.13 shows, the T^{SF} levels are 4 dB below the $\tilde{T}^{(f_n)}$ levels in the ninth critical band, 7 dB below in the fourteenth critical band and 8 dB below in the seventeenth critical band. Regarding the OA method, the T^{OA} levels are around 5 dB above the $\tilde{T}^{(f_n)}$ levels for the three critical bands. The IA method provides levels closer to those obtained in the subjective test, specifically, the T^{IA} levels are 0.1 dB above the $\tilde{T}^{(f_n)}$ levels and 1.8 dB and 2.5 dB below the $\tilde{T}^{(f_n)}$ levels.

According to these results, the IA method provides a lower tonality estimation than the SF method and a higher tonality estimation than the OA method, providing the most accurate estimate of the masking threshold level obtained in the subjective tests.

3.3.6 Conclusions

In this section, the tonality parameter has been studied in depth providing different approaches to estimate it. The first method is a well-known method commonly used to estimate the masking threshold called, in this dissertation, as spectral flatness (SF) method (see Section 3.3.2). The second method is a modified version of the Aures method [127] with minor changes of the original method from a mathematical point of view. Due to these changes, explained in Section 3.3.3, this method is called as improved Aures (IA).

Moreover, a perceptual test has been carried out in order to evaluate the performance of the different tonality methods regarding the subjective

perception of broadband signals. Since the IA method is slightly different from the original Aures method (OA), the study has evaluated the three different tonality methods (the SF, IA and OA methods). The test has been carried out by 16 participants and it has used three multi-tone signals covering the most sensitive range of the human hearing system (350 – 5800 Hz) with the purpose of obtaining the perceived masking threshold in the presence of a narrowband noise. The objective masking thresholds of the OA, IA and SF methods have been computed and compared to the levels obtained in the subjective test. The results (see Figures 3.12 and 3.13) show that the SF method underestimates the masking, while the OA method overestimates the masking. In contrast, the IA method presents the most accurate masking levels, regarding the subjective masking levels, among the three methods. In fact, according to Figure 3.13, the IA method presents an absolute mean error($|\bar{\epsilon}|$) of 1.7 dB, while the SF method and the OA method present an absolute mean error of 5.3 dB and 5.8 dB respectively. Therefore, the SF and OA methods produce a similar mean error (above 5 dB), while the mean error of the IA method is reduced in 3 dB approximately. This behavior can be appreciated as well in the average difference ($\bar{\epsilon}$), where the IA presents a value of 0.8 dB with respect to the masking thresholds provided by the subjective test, while for the SF and OA methods the value reaches -5.3 dB and 5.8 dB respectively.

Therefore, the IA tonality method improves the estimation of the masking threshold regarding the traditional tonality method (the SF method). Additionally, the IA method has been implemented straightforwardly and simply from the OA method giving a very significant improvement on the estimation of the masking threshold, as Figures 3.12 and 3.13 show.

3.4 Perceptual Audio Equalization

3.4.1 Overview

The perception of audio signals may be severely impaired when they are reproduced in the presence of ambient noise. A solution to this problem is the use of headphones to prevent ambient noise from interfering with the audio signal. There are, however, some cases where this solution is not possible, such as for the driver of a car. And even for the other passengers of the car, a non-headphone-based solution will be more pleasant. To this

end, a method that avoids the use of headphones is proposed in this section.

Audio equalization is a large area of research that aims to adjust certain frequencies (or frequency bands) of a specific audio signal and thus, modify its spectral shape. To this end, a bank of filters, that can be designed in many different ways [137, 138], is used in order to modify the spectral shape of the specific audio signal. The equalizers provide a great flexibility since a different number of filters with different bandwidths can be defined, leading to different ways of performing an equalization depending on the requirements of the corresponding application [139, 140]. Therefore, the spectral shape of the equalized signal is governed by the different filters that compose the equalizer.

In the above situation, when the audio signal may be affected by the ambient noise, the equalization can be used to modify the frequency response of the audio signal with the purpose of improving its perception in presence of ambient noise. For this purpose, an analysis of both signals must be performed in order to obtain the gains corresponding to each of the filters of the equalizer [141, 142]. In addition, as stated in Section 2.4.3, when two sounds are perceived simultaneously, the masking effect is presented, causing a change in the perception of one of the sounds, considered as the maskee signal, in presence of the other, that acts as the masker signal. In Section 3.1, we presented an algorithm to estimate the masking threshold of a signal (see Figure 3.1). Therefore, the level of masking caused by the ambient noise to the audio signal (and vice-versa) can be known through the masking threshold algorithm. In fact, this algorithm can be used in order to obtain the corresponding gains for each filter of the equalizer and adjust the audio signal spectral shape, resulting in the so-called perceptual equalizer [7, 93].

In [7] a study of a single-user perceptual equalizer is carried out when using headphones. By using a single microphone to capture the ambient noise, the proposed algorithm in [7] is able to adaptively modify the frequency response of the audio signal, and consequently improve its perception according to the masking caused by the ambient noise. This approach is highly interesting since the reproduction system is capable of adapting in real-time to the changes of the ambient noise automatically through an analysis that mimics the auditory system. In this regard, this section proposes and analyzes a single-user perceptual equalizer that does not require the use of headphones.

The proposed system is formed by one microphone and one loudspeaker to record and reproduce the corresponding signals, and a processor to implement the algorithms for the perceptual equalization. Accordingly to Section 2.1.3, this organization corresponds to a single-channel acoustic node of an ASN. Therefore, the single-user perceptual equalizer can be implemented over an acoustic node, as Figure 3.14 shows, where N nodes (or N single-user perceptual equalizers) are shown.

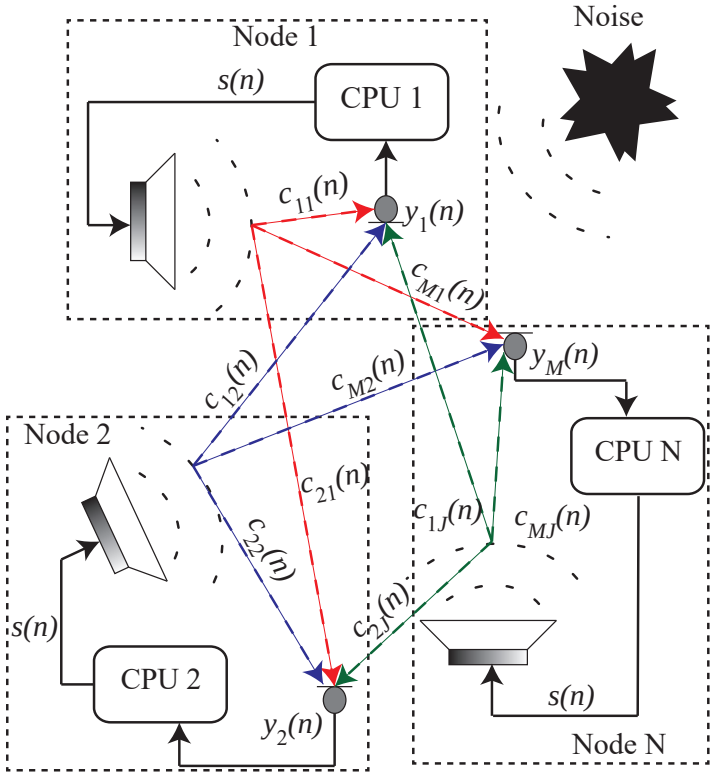


Figure 3.14. Single-user perceptual equalizers running on an ASN.

The system shown in Figure 3.14 can also be considered as a multiple-user perceptual equalizer where the nodes communicate with each other in order to exchange important details about the local equalization of each node, performing a perceptual processing more precise. In addition, this processing can be performed in two different ways. If we consider a small scenario, such as the cabin of a car, a centralized system with a single pro-

cessor, which will perform the required perceptual analysis and send the corresponding parameters to each node can be used. But, for larger scenarios, such as trains or airplanes, an ASN with distributed processors can be a more reliable solution since they provide a more flexible and scalable network [4, 143, 144].

Nevertheless, before studying the system shown in Figure 3.14, a deep analysis of the single-user perceptual equalizer is essential in order to evaluate its performance and, thus simplify as much as possible the extension to the multiple-user. In this regard, in this section, a single node of the ASN shown in Figure 3.14 is considered (such as the node labelled as "Node 1"). Considering this scenario, two main studies are carried out in this section:

1. A comparison of different masking patterns in order to obtain an optimal spectral masking model. This study will allow to find the lowest complexity pattern to be used in the masking estimation causing the extension to multiple users to be simpler.

2. Based on the optimal frequency model found, we perform an experimental study for a single-user case to analyze the performance of the perceptual equalizer.

First of all, the steps that compose the process to equalize the audio signal will be explained in detail. Then, once the equalization process has been explained, each of the studies defined above will be discussed in depth.

3.4.2 Perceptual equalization algorithm

The block diagram of Figure 3.15 shows the process to compute the perceptual equalization over the m^{th} audio signal frame, $s_m(n)$, composed by M_{S} samples. This process estimates the m^{th} equalized audio signal frame, that is emitted by the j^{th} speaker ($q_{j,m}(n)$), based on the analysis of the m^{th} audio signal and noise signal frame at the i^{th} microphone position ($\widetilde{u}_{i,m}(n)$ and $\widetilde{r}_{i,m}(n)$ respectively).

As Figure 3.15 shows, the process of equalization can be divided into three main steps:

1. Source Separation: This block allows to independently obtain the contributions $\widetilde{u}_{i,m}(n)$ and $\widetilde{r}_{i,m}(n)$ from the total recorded signal, $y_{i,m}(n)$.

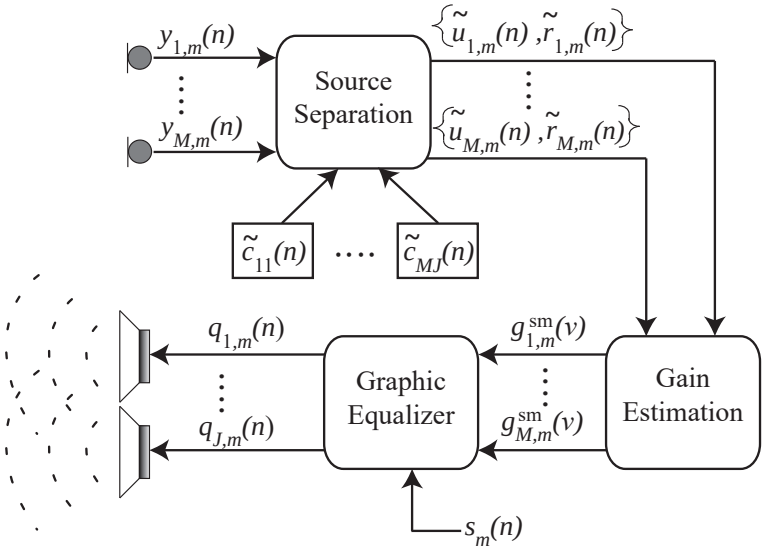


Figure 3.15. Block diagram of the audio perceptual equalization.

2. Gain Estimation: This block estimates the gain levels ($g_m^{\mathrm{sm}}(\nu)$) that will feed the equalizer in order to improve the audio signal perception. However, the perception can be improved in different ways leading to different strategies, called in this dissertation as "equalization profiles". According to these profiles, the perceptual analysis is different and consequently the equalized audio signal, $q_{j,m}(n)$, is also different.

3. Graphic Equalizer: The final block equalizes the audio signal frame, $s_m(n)$. The transfer function of the graphic equalizer is controlled by specifying the gains of each band [138]. Since these gains are obtained through the perceptual analysis, based on critical bands, the bands that compose the graphic equalizer are designed according to the critical bands as well (see Table 2.2).

In order to fully understand each step, in the following paragraphs a detailed description of each one is made, where for the sake of simplicity, the subscript m denoting the m^{th} frame is omitted.

Source Separation

Consider that the signal picked up by the i^{th} microphone ($y_i(n)$ in Figure 3.15) is modeled as:

$$y_i(n) = u_i(n) + r_i(n)\,, \tag{3.22}$$

where $u_i(n)$ and $r_i(n)$ are the audio signal and the ambient noise signal at the position of the i^{th} microphone respectively. Assuming the audio signal ($s(n)$) is known and the electro-acoustic paths between i^{th} microphone and all the speakers ($c_{ij}(n)$ with $j = 1, \ldots, J$) have been estimated in a previous stage, the recorded audio signal can be estimated as:

$$\widetilde{u}_i(n) = s(n) * \sum_{j=1}^{J} \widetilde{c}_{ij}(n)\,, \tag{3.23}$$

where $\widetilde{c}_{ij}(n)$ is an estimate of $c_{ij}(n)$. If we consider the single-user case, only one acoustic path ($c_{11}(n)$) is involved, and equation (3.23) is developed for $J = 1$. Finally, considering an accurate estimate of the electro-acoustic path, the recorded ambient noise, $r_i(n)$, could be estimated by using (3.22).

Gain Estimation

Once each contribution has been identified, the perceptual analysis block is performed in order to estimate the gain levels in dB units, $g^{\text{sm}}(\nu)$. This analysis depends on the different equalization profiles implemented. In this case, because two different masking thresholds can be estimated (from the audio signal and from the noise signal), two different profiles are implemented: 1) Unmasked Audio Signal (UAS) profile and 2) Masked Noise (MN) profile. Because each one estimates the gains differently, in the following paragraphs, the process to obtain the gains for each profile is described.

Unmasked Audio Signal (UAS) profile

This profile is based on the strategy used in [94] that aims to prevent masking of the audio signal from the ambient noise signal. To this end, the masking threshold of the recorded ambient noise ($\widetilde{r}_i(n)$) is estimated in order to find the critical bands that present a higher level in comparison with the SPL level of the recorded audio signal ($\widetilde{u}_i(n)$), which must

be expressed in terms of critical band (see (3.2)) to perform an accurate comparison. For those bands where the masking of $\widetilde{r}_i(n)$ is higher than the SPL level of $\widetilde{u}_i(n)$, the gain level will be a positive value, otherwise the gain level will be zero. Therefore, the gain levels for the UAS profile can be estimated as follows:

$$g(\nu) = \max\left(\left[T_r(\nu) - E_u^{\mathrm{dB}}(\nu)\right] + 2, 0\right) , \tag{3.24}$$

where $g(\nu)$ is the gain in dB units for the critical band ν. $E_u^{\mathrm{dB}}(\nu)$ (3.2) is the energy per critical band of the recorded audio signal and $T_r(\nu)$ is the masking threshold (3.20) of the recorded ambient noise, both expressed in dB SPL. According to [94], the value +2 dB is introduced since sounds above 2 dB the masking threshold of a masker sound showed a clear improvement in its perception.

Noise Masked (NM) profile

The NM profile is used to mask the ambient noise through the audio signal. In this case, the masking threshold of the recorded audio signal ($\widetilde{u}_i(n)$) is estimated in order to analyze which are the critical bands where the threshold is lower than the SPL level of the ambient noise. Identically to the previous profile, to perform an accurate comparison, the energy per critical band of the recorded ambient noise is estimated, causing the following expression to estimate the gains for the NM profile:

$$g(\nu) = \max\left(E_r^{\mathrm{dB}}(\nu) - T_u(\nu), 0\right) , \tag{3.25}$$

where $g(\nu)$ is included in the same range than previous profile, such that $g(\nu) \geq 0$, $E_r^{\mathrm{dB}}(\nu)$ is the energy per critical band of the recorded ambient noise and $T_u(\nu)$ is the masking threshold of the recorded of the audio signal, both expressed in dB SPL.

Additionally, a gain level limitation is used to prevent undesired non-linear effects over the equalization, such as saturation in the reproduction. In this case, regardless of the profile, $g(\nu)$ has been limited to 15 dB for all the critical bands. Furthermore, when audio signal is not detected, such that $E_u^{\mathrm{dB}}(\nu) \leq 0$ dB SPL, gain levels must not be introduced ($g(\nu) = 0$ dB) in those critical bands.

As a final stage in the perceptual analysis, a frame averaging over

the gains is performed. Since both, audio signal and noise signal, can be non-stationary, the estimated gain levels could be very different between frames. This behavior can lead to large fluctuations in the gain levels for the critical bands causing an annoying effect. Therefore, an EMA (Exponential Moving Average) [145] is applied over the gain levels to provide a smoother transition between frames for each critical band. However, depending on the gain level tendency, two cases can be distinguished: 1) increase of the gain level between frames and 2) decrease of the gain level between frames. The first case requires a fast adaptation in order to achieve the desired condition (see (3.24) and (3.25)). In contrast, for the second case a slow adaptation is a better option because a fast reduction of the gain leads to a perceptible reduction of the loudness and consequently to a reduction of the perception of the audio signal [6] (even if the ambient noise is reduced). Therefore, the introduction of a slow adaptation will solve the problem, as the reduction will be gradual and imperceptible. Since two cases are presented, two EMA must be performed, one for the critical bands falling under the first case and the other for the critical bands falling under the second case. Although two EMA are used, the expression is identical in both cases:

$$g^{\mathrm{sm}}(\nu) = \xi_a g(\nu) + (1 - \xi_a) g^{\mathrm{sm}}(\nu)\,, \tag{3.26}$$

where $g^{sm}(\nu)$ are the averaged gains in dB units and ξ_a is the smoothing constant that depends on the cases previously described, where subscript $a = 1, 2$ defines the case. For the first case (a fast adaptation) $\xi_1 = 0.3$ and for the second case (slow adaptation) $\xi_2 = 0.1$. As Figure 3.15 shows, the averaged gains are introduced in the graphic equalizer.

Graphic Equalizer

The graphic equalizer is composed by as many filters as critical bands are presented, where each filter is designed for each critical band. According to [138], these filters can be represented by a second-order digital peaking and shelving filters where the whole frequency response is expressed as

$$H(z) = \prod_{\nu=1}^{N_{\mathrm{c}}} H_\nu(z), \tag{3.27}$$

where N_{c} are the total number of critical bands and $H_\nu(z)$ is the filter for

the critical band ν that can be expressed as:

$$H_\nu(z) = \frac{b_0(\nu) + b_1(\nu)z^{-1} + b_2(\nu)z^{-2}}{1 + a_1(\nu)z^{-1} + a_2(\nu)z^{-2}}, \tag{3.28}$$

where the coefficients of the denominator and the numerator can be expressed as:

$$\begin{aligned} a_2(\nu) &= \frac{2Q(\nu) - \sin\varphi_\mathrm{c}(\nu)}{2Q(\nu) + \sin\varphi_\mathrm{c}(\nu)} \\ a_1(\nu) &= b_1(\nu) = -\left(1 + a_2(\nu)\right)\cos\varphi_\mathrm{c}(\nu) \\ b_0(\nu) &= \frac{1}{2}\left(1 + a_2(\nu)\right) + \frac{1}{2}\left(1 - a_2(\nu)\right) g_\mathrm{opt}(\nu) \\ b_2(\nu) &= \frac{1}{2}\left(1 + a_2(\nu)\right) - \frac{1}{2}\left(1 - a_2(\nu)\right) g_\mathrm{opt}(\nu)\,, \end{aligned} \tag{3.29}$$

where $Q(\nu)$ is the quality factor, expressed as:

$$Q(\nu) = \frac{1}{2}\left[\frac{g_\mathrm{opt}(\nu)\sin^2\varphi_\mathrm{c}(\nu)\left(\cos\varphi_\mathrm{l}(\nu) + \cos\varphi_\mathrm{u}(\nu)\right)}{2\cos\varphi_\mathrm{c}(\nu) - \cos\varphi_\mathrm{l}(\nu) - \cos\varphi_\mathrm{u}(\nu)}\right]^{\frac{1}{2}}, \tag{3.30}$$

$\varphi_\mathrm{c}(\nu)$ is the center frequency, expressed as:

$$\varphi_\mathrm{c}(\nu) = \arccos\left\{\kappa(\nu) - \mathrm{sign}\{\kappa(\nu)\}\left(\kappa(\nu)^2 - 1\right)^{\frac{1}{2}}\right\}, \tag{3.31}$$

being:

$$\kappa(\nu) = \frac{1 + \cos\varphi_\mathrm{l}(\nu)\cos\varphi_\mathrm{u}(\nu)}{\cos\varphi_\mathrm{l}(\nu) + \cos\varphi_\mathrm{u}(\nu)}, \tag{3.32}$$

where $\varphi_\mathrm{u}(\nu)$ and $\varphi_\mathrm{l}(\nu)$ are the upper and lower boundaries respectively of the critical band ν, that is, the high and low frequencies of the specific critical band. The parameter $g_\mathrm{opt}(\nu)$ is described as the optimal gain per critical band in linear units, obtained from (3.26).

According to [146] and in order to obtain an specific global frequency response, $g^\mathrm{sm}(\nu)$, estimated in (3.26), should be re-adjusted based on the

global frequency response of the graphic equalizer (given in (3.27)). Otherwise the equalizer will provide different gain levels than desired in each critical band due to the contributions of the different filters, $H_\nu(z)$, as Figure 3.16 shows.

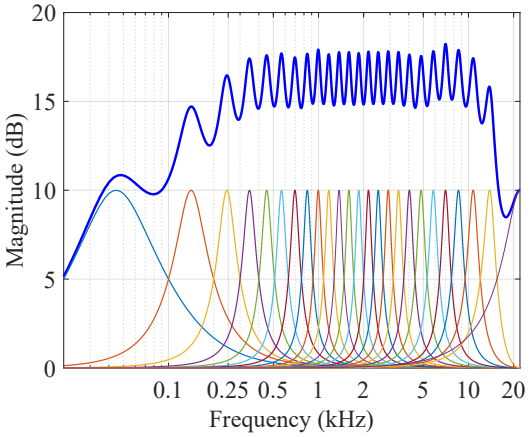


Figure 3.16. Global frequency response (in blue line) of the graphic equalizer assuming 10 dB gain per critical band.

Figure 3.16 represents an equalizer with a gain of 10 dB per critical band. However, the resulting global frequency response, $H(z)$ (represented by the blue line), exhibits a response with a magnitude higher than the desired one. For this reason, an additional processing is required to adjust the gains at each critical band:

$$\mathbf{g}_{\text{opt}}^{\text{dB}} = \mathbf{A}^{-1}\mathbf{g}^{\text{sm}}\,, \tag{3.33}$$

where $\mathbf{g}_{\text{opt}}^{\text{dB}}$ is the array of optimal gains expressed in dB units, such that $\mathbf{g}_{\text{opt}}^{\text{dB}} = \left(\left[g_{\text{opt}}^{\text{dB}}(1)\ \ldots\ g_{\text{opt}}^{\text{dB}}(N_{\text{c}})\right]\right)^T$ and $\mathbf{g}^{\text{sm}}$ is the array of gains estimated in the previous step, such that $\mathbf{g}^{\text{sm}} = \left(\left[g^{\text{sm}}(1)\ \ldots\ g^{\text{sm}}(N_{\text{c}})\right]\right)^T$. The matrix $\mathbf{A}$ is an interaction matrix of $[N_{\text{c}} \times N_{\text{c}}]$ that stores the normalized amplitude response of all the filters at the corresponding center frequencies. The generation of this matrix is described in [146] and it is estimated through the following procedure:

1. Select a prototype gain, g_{p}, such as $g_{\text{p}} = 10$ dB.

2. Assume the first critical band ($\nu = 1$) gets the prototype gain, g_{p}, while the rest of the critical bands offer 0 dB of gain.

3. Generate the filters $H_{\nu}(z)$ through (3.28).

4. Obtain the global frequency response of the graphic equalizer, $H(z)$, through (3.27).

5. Obtain the magnitude at the center frequency of each critical band inside the global frequency response. These contributions will represent the first row of $\mathbf{A}$.

6. Repeat the process for all critical bands assuming the prototype gain, $g_{\mathrm{p}} = 10$ dB, is used only in one of them at the same time.

Finally, once all the rows have been estimated, $\mathbf{A}$ must be divided by g_{p}, such as $\mathbf{A} = \mathbf{A}/g_{\mathrm{p}}$, in order to normalize the results and use $\mathbf{A}$ for any gain level.

Considering the case illustrated in Figure 3.16, the processing performed through (3.33) leads to the result depicted in Figure 3.17. In contrast to Figure 3.16, the gain levels of the global response of the graphic equalizer is around 10 dB for each critical band.

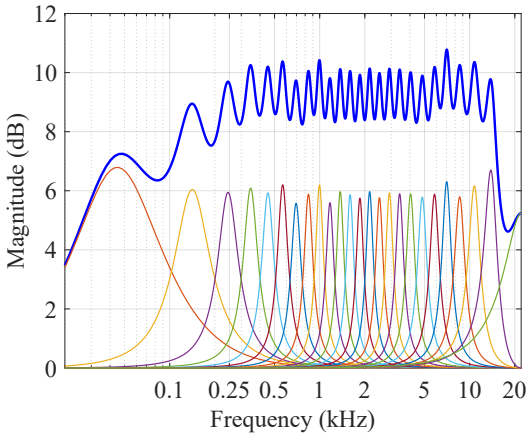


Figure 3.17. Global frequency response (in blue line) of the graphic equalizer assuming 10 dB gain per critical band considering $\mathbf{A}$ matrix.

Thus, the theoretical gain of each filter must be reduced to achieve the corresponding target response, as shown in Figure 3.17.

3.4.3 Efficient spectral masking Model

The spectral masking model is analyzed in this section with the purpose of obtaining a straightforward method to estimate the masking threshold. In this way, we can facilitate the extension from the single user to the multiple user case. The process explained in Section 3.4.2 is based on the study performed in [7], where the masking pattern used to estimate the masking threshold depends on the energy level of the masker signal. It also provides a non-linear addition between the different contributions of the masker signal. As a result, the complexity for computing the gain levels is high for the single-user case. This means that when it is extended to multiple users, the complexity can be significantly increased. Therefore, this section proposes to study different masking patterns in order to reduce their complexity while maintaining their performance compared to the masking pattern used in [7].

In this study, three different masking patterns are compared (including the one used in [7]). The first pattern can be found in [103, 107] and it is commonly used for audio coding applications. Because it was defined in [107], in this study it is named as the "Schroeder" pattern and it will be denoted as $B_\nu^{\mathrm{S}}(\eta)$. The second pattern is described in [109] and is used for removing irrelevant components of the signal. This pattern is named as the "Balazs" pattern and it will be denoted as $B_\nu^{\mathrm{B}}(\eta)$. Finally, the third patter is used in [7] for perceptual equalization. This pattern is named as the "Rämö" pattern and it will be denoted as $B_\nu^{\mathrm{R}}(\eta)$. These patterns are expressed in dB units as follows:

$$
\begin{aligned}
B_\nu^{\mathrm{S}}(\eta) &= 15.81 + 7.5\left(\Delta_\nu(\eta) + 0.474\right) - 17.5\sqrt{1 + \left(\Delta_\nu(\eta) + 0.474\right)^2} \\
B_\nu^{\mathrm{B}}(\eta) &= 13.94 + 1.5\left(\Delta_\nu(\eta) + 0.03\right) - 25.5\sqrt{0.3 + \left(\Delta_\nu(\eta) + 0.03\right)^2} \\
B_\nu^{\mathrm{R}}(\eta) &= \left[-27 + 0.37\max\{E_{x,m}^{\mathrm{dB}}(\eta) - 40, 0\}\theta\left(\Delta_\nu(\eta)\right)\right]\left|\Delta_\nu(\eta)\right| ,
\end{aligned}
\tag{3.34}
$$

where $\Delta_\nu(\eta)$ is defined in (3.3), $E_{x,m}^{\mathrm{dB}}(\eta)$ is the energy per critical band in dB units (3.2) and $\theta(\Delta_\nu(\eta))$ in the Rämö pattern is a step function equal

to 0 for negative values of $\Delta_\nu(\eta)$ and 1 for positive values. Significant differences can be found between the three masking patterns of (3.34). The most relevant difference is that the Rämö pattern depends on the energy per critical band, while the other two patterns are independent of this factor.

In order to better appreciate the differences between these patterns, Figure 3.18 shows the effect of each masking pattern for a single critical band, specifically the 8^{th} band. In addition, to show the effect of the energy dependency on the Rämö pattern, two different examples are used. The first one considers a masker of $E^{\text{dB}}_{x,m}(8) = 70$ dB SPL, while he second one assumes a masker of $E^{\text{dB}}_{x,m}(8) = 40$ dB SPL.

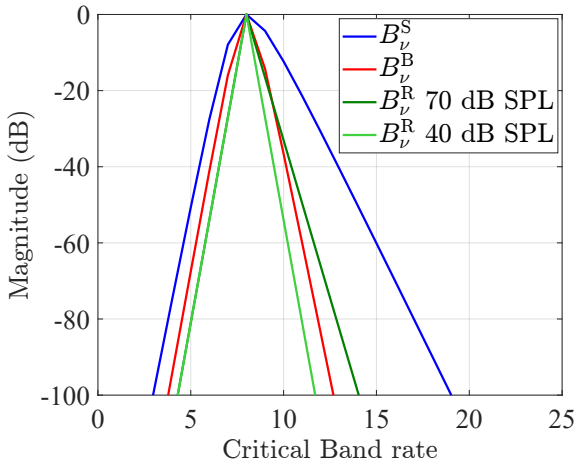


Figure 3.18. Difference between the patterns of (3.34) assuming the 8^{th} band as the masker band.

As Figure 3.18 shows, the three patterns have a triangular shape, but with certain differences, specially in the right slope where the Schroeder pattern has a smoother fall than the others. That is, the Schroeder pattern considers that the masking has a greater spreading for the bands after the masker band, causing that the overall spreading curve is higher than the rest. Regarding the Rämö pattern, it presents a smoother fall of the right slope when masker has a SPL level of 70 dB SPL, while the left slope does not show any difference. This means that the energy significantly affects to the right slope of the pattern, and it does not affect to the left slope. As a

result, a higher energy leads to a higher spreading, causing a higher overall spreading curve. Finally, the Balazs pattern provides a pattern independent of the energy per critical band, as the Schroeder pattern. However, the Balazs pattern produces a lower spreading than the Schroeder pattern, specially in the right slope, and even a lower spreading than the Rämö pattern when $E_{x,m}^{\mathrm{dB}}(8) = 70$ dB SPL.

According to [116], the addition between the different contributions of the different critical bands is controlled by the "α" parameter (see (3.5)). As stated before, the Rämö pattern provides a non-linear summation, specifically $\alpha = 0.33$, in order to increase the masking, compared to the simple summation ($\alpha = 1$). In [107, 109], the Schroeder pattern and the Balazs pattern use $\alpha = 1$, leading to a linear summation between the different contributions of the critical bands. Additionally, a non-linear summation, specifically $\alpha = 0.5$, for the Schroeder and the Balazs patterns are proposed to perform a more complete study. Furthermore, same as Section 3.3.5, both tonality methods (Aures and SF) are used in the study as well, leading to a collection of ten different combinations.

To compare the patterns shown in (3.34), the real system depicted in Figure 3.15 has been simulated. This system represents the position of a seated passenger inside a vehicle and it is composed by one microphone and one speaker (one acoustic node) and an additional speaker to emulate the ambient noise. The signal to be processed ($y(n)$ in Figure 3.15) has been simulated by using the real-time acoustic response measured in the system shown in Figure 3.19 located in the listening room of the Audio Processing Laboratory of the Polytechnic University of Valencia [134]. As Figure 3.19 shows, the loudspeaker that reproduces the audio signal is separated 50 cm from the microphone, while the speaker that emulates the ambient noise is separated 1.5 m from the microphone. In addition, the microphone of the real system have been calibrated, thus the SPL level of the simulated signal, $y(n)$, can be obtained.

Considering this scenario, the single-channel perceptual equalizer of Figure 3.15 has been implemented for the different combinations to be compared. The audio signal used, $s(n)$, is an excerpt of the song "Tell me something good" by Chaka Khan, while the ambient noise is an except of traffic noise whose spectral composition is shown in Figure 3.22. Both signals are sampled at $f_{\mathrm{s}} = 44100$ Hz and have a duration of 30 seconds. Additionally, the audio signal has been weighted in loudness as

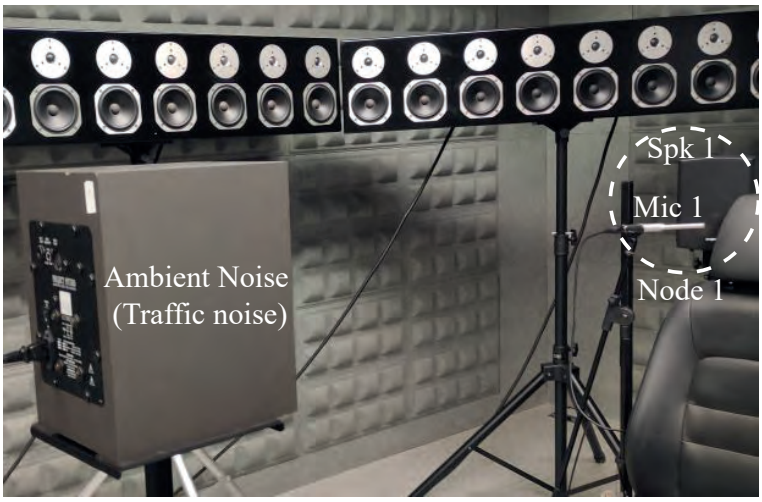
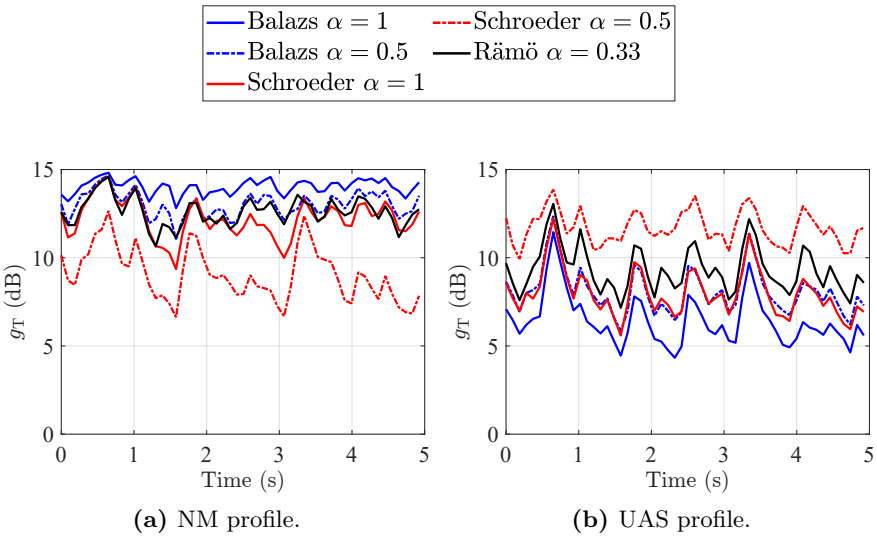


Figure 3.19. Scenario simulated for the perceptual equalizer.

Section 3.3.5 explains, but using a tone of 70 dB SPL located at 1 kHz for the reference loudness. The traffic noise has been weighted as well with respect to the loudness weighted audio signal in order to produce a signal-to-noise ratio (SNR) of -3 dB. Regarding to the processing, the audio perceptual equalizer shown in Figure 3.15 is designed using $M_{\mathrm{T}} = 2M_{\mathrm{S}}$ and $M_{\mathrm{S}} = N_{\mathrm{FFT}} = 4096$ samples (see the "Spectral Analysis" block in Section 3.1).

Each of the ten combinations defined above have been used for the different equalization profiles (UAS and NM), resulting in twenty different gain levels, $g^{\mathrm{sm}}(\nu)$. The gain levels indicate how the spectral shape of the audio signal is modified. Therefore, the more similar the gain levels are between the different combinations, the more similar the equalized audio signal will be. This means that similar gain levels will produce a similar perception in presence of the ambient noise.

Figures 3.20 and 3.21 compare the different combinations according to the equalization profile (UAS or NM) and the tonality method (SF or Aures) used. These results show the 75 % percentile of the sum of the smoothed gains, such as $g_{\mathrm{T}} = \sum_{\nu} g^{\mathrm{sm}}(\nu)$, from the 6 to the 11 seconds (represented from 0 to 5 in the figures) since this fragment shows a significant change in the audio signal energy.

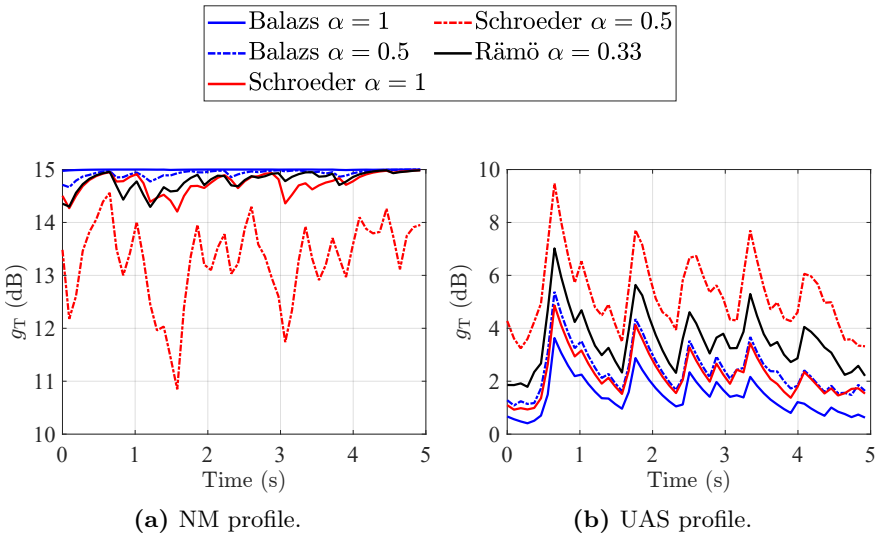


(a) NM profile. **(b)** UAS profile.

Figure 3.20. Curves of the 75 % percentile of g_T for the different combinations when tonality is estimated through the Aures method.

Figure 3.20 shows the gain levels when the tonality has been computed with the Aures method. The patterns of Schroeder with $\alpha = 1$, Balazs with $\alpha = 0.5$ and Rämö show a very similar gain levels regardless the profile. However, they are more similar for the NM profile since the difference is less than 1 dB, while for the UAS profile they present a difference around 2 dB in the worst case. In contrast, the patterns of Schroeder with $\alpha = 0.5$ and Balazs with $\alpha = 1$ present significant differences with the Rämö pattern. For the NM profile, (see Figure 3.20a), the pattern of Schroeder with $\alpha = 0.5$ shows the lowest gain level among all the combinations, while for the UAS profile it shows the highest gain level. Regarding the Balazs pattern with $\alpha = 1$, it presents the opposite performance as the Schroeder pattern with $\alpha = 0.5$.

Figure 3.21 shows the gain levels when the tonality has been computed with the SF method. Similar to Figure 3.20, the patterns of Schroeder with $\alpha = 1$, Balazs with $\alpha = 0.5$ and Rämö presents a similar behavior with a difference less than 1 dB for the NM profile and a difference around 2 dB for the UAS profile in the worst case. In this case, these three patterns are very close of the gain limit value (15 dB) for the NM profile. In fact,



(a) NM profile. **(b)** UAS profile.

Figure 3.21. Curves of the 75 % percentile of g_T for the different combinations when tonality is estimated through the SF method.

these combinations are limited from the second 4 to 5. The other two combinations present the same behavior as Figure 3.20. However, it should be highlighted the fact that the Balazs pattern with $\alpha = 1$ always provides a result higher than 15 and for this reason this combination is always limited, as opposed to the case shown in Figure 3.20.

As Figures 3.20 and 3.21 show, the patterns of Schroeder with $\alpha = 0.5$ and Balazs with $\alpha = 1$ show a different behavior to the rest. In this regard, we can assure that the Balazs pattern with $\alpha = 1$ estimates a lower masking threshold than the rest of combinations, while the Schroeder pattern with $\alpha = 0.5$ estimates a higher one. As a result, the behavior of each one of these patterns are the opposite depending on the profile. Regarding the other three patterns, they present a similar estimation of the masking threshold, resulting in similar gain level values. Additionally, according to these results, the SF tonality method provides higher gain levels with the NM profile than the Aures tonality method, while for the UAS profile the performance is just the opposite. According to (3.24) and (3.25), a higher masking threshold produces lower gain levels for the NM profile, but higher gain levels for the UAS profile. Therefore, the Aures method

estimates a higher masking threshold than the SF method, just as described in Section 3.3.5.

Given these results, the patterns of Schroeder with $\alpha = 1$, Balazs with $\alpha = 0.5$ estimate a similar equalized audio signal than the Rämö pattern and consequently they provide a similar perception of the audio signal in presence of the ambient noise. However, the patterns of Balazs with $\alpha = 0.5$ and Rämö compute the masking through a non-linear addition of the different contributions of the different critical bands, and moreover, the latter depends on the energy per critical band. As a result, the pattern of Schroeder with $\alpha = 1$ is selected, since it provides both, a pattern independent of the energy of the masker signal and a linear addition between the different contributions of the critical bands.

Considering the masking pattern of Schroeder with $\alpha = 1$, the masking threshold (3.20) is computed in linear units as:

$$\mathbf{t}_{x,m} = \mathbf{S}_{x,m}\mathbf{O}_{x,m}^{-1} = \mathbf{e}_{x,m}\mathbf{B}\mathbf{O}_{x,m}^{-1}\,, \tag{3.35}$$

where $\mathbf{e}_{x,m}$ is defined as the vector that contains the energy per critical band at the m^{th} frame:

$$\mathbf{e}_{x,m} = [E_{x,m}(1)\ E_{x,m}(2)\ \ldots\ E_{x,m}(N_{\text{c}})]\,, \tag{3.36}$$

$\mathbf{B}$ is defined as a $[N_{\text{c}} \times\ N_{\text{c}}]$ matrix with the values of the Schroeder pattern in linear units(see (3.34)):

$$\mathbf{B} = \left(\left[B_1(\eta)^T\ B_2(\eta)^T\ \ldots\ B_{N_{\text{c}}}(\eta)^T\right]\right)^T\,, \tag{3.37}$$

and $\mathbf{O}$ is defined as a diagonal $[N_{\text{c}} \times\ N_{\text{c}}]$ matrix whose non-zero elements are the N_{c} values of the tonality offset ($O_{x,m}(\nu)$) obtained at the lower branch of Figure 3.1.

From (3.35), the gains in (3.24) and (3.25) are derived in vectorial way and in linear units as:

$$\begin{aligned} \mathbf{g}_m^{\mathrm{NM}} &= \max\left(\frac{\mathbf{e}_{r,m}}{\mathbf{e}_{u,m}\mathbf{B}\mathbf{O}_{u,m}^{-1}}, 1\right) \\ \mathbf{g}_m^{\mathrm{UAS}} &= \max\left(\frac{\mathbf{e}_{r,m}\mathbf{B}\mathbf{O}_{r,m}^{-1}}{\mathbf{e}_{u,m}}, 1\right), \end{aligned} \tag{3.38}$$

where $\mathbf{e}_{r,m}$ and $\mathbf{e}_{u,m}$ are the vectors of the noise and audio signal energy values respectively, and $\mathbf{O}_{r,m}$ and $\mathbf{O}_{u,m}$ are the matrices that contains the offset estimated through the noise and audio signal respectively.

Considering now the multi-user scenario in Figure 3.14 and focusing on the first node (labeled as "Node 1") to estimate the gains to equalize the audio signal, the recorded signal can be modeled as:

$$y_{1,m}(n) = c_{11}(n) * s_m(n) + c_{\mathrm{net}}(n) * s_m(n) + r_1(n)\,, \tag{3.39}$$

where $c_{\mathrm{net}}(n)$ models the network contribution to the recorded signal at the first node, which is represented by the combination of c_{1j} with $j = 2, \ldots, J$ in Figure 3.15. This contribution can be seen as an additional energy to each band of the audio signal leading to the following expressions to estimate the gains:

$$\begin{aligned} \mathbf{g}_m^{\mathrm{NM}} &= \max\left(\frac{\mathbf{e}_{r,m}}{(\mathbf{e}_{u,m} + \mathbf{e}_{\mathrm{net},m})\,\mathbf{B}\mathbf{O}_{u,m}^{-1}}, 1\right) \\ \mathbf{g}_m^{\mathrm{UAS}} &= \max\left(\frac{\mathbf{e}_{r,m}\mathbf{B}\mathbf{O}_{r,m}^{-1}}{(\mathbf{e}_{u,m} + \mathbf{e}_{\mathrm{net},m})}, 1\right), \end{aligned} \tag{3.40}$$

where $\mathbf{e}_{\mathrm{net},m}$ is the energy vector per critical band provided by the rest of the nodes of the network. Unlike the UAS profile where the additional network energy does not significantly increase the complexity of the whole algorithm in the multi-user case, the NM profile requires to know the masking threshold of the recorded audio signal. Therefore, for the Rämö pattern (that is dependent of the energy), the $\mathbf{B}$ matrix must be estimated for each of the different contributions of the energy. However, for the Schroeder

pattern, the $\mathbf{B}$ matrix is estimated once, reducing the computational complexity when using multiple users. In addition, since the pattern selected uses $\alpha = 1$, the addition between the contributions of the different critical bands is linear, resulting in a lower computational cost compared to the combinations where $\alpha \neq 1$. For this reason, the selected pattern facilitates the computation of the masking pattern in the multi-user case.

3.4.4 Perceptual experiment

A subjective test has been carried out in order to analyze the performance of the different profiles explained in Section 3.4.2. To this end, the Schroeder pattern with $\alpha = 1$ is used because of the results of Section 3.4.3. Additionally, both tonality methods (the SF method, explained in Section 3.3.2, and the Aures method, explained in Section 3.3.3) are used in this study. Furthermore, the profile where no equalization is performed has also been considered in the perceptual test as a baseline and it has been labelled as "NONE" profile in the results. Therefore, five different profiles are studied in this test:

1. The NM Aures profile. The NM profile based on the masking threshold when the Aures tonality method is used.
2. The NM SF profile. The NM profile based on the masking threshold when the SF tonality method is used.
3. The UAS Aures profile. The UAS profile based on the masking threshold when the Aures tonality method is used.
4. The UAS SF profile. The UAS profile based on the masking threshold when the SF tonality method is used.
5. The NONE profile. A profile where no equalization is performed, that is, $q(n) = s(n)$ in the simulated scenario shown in Figure 3.19

In order to describe in detail the subjective test, the main features as well as the results obtained are explained in the following paragraphs.

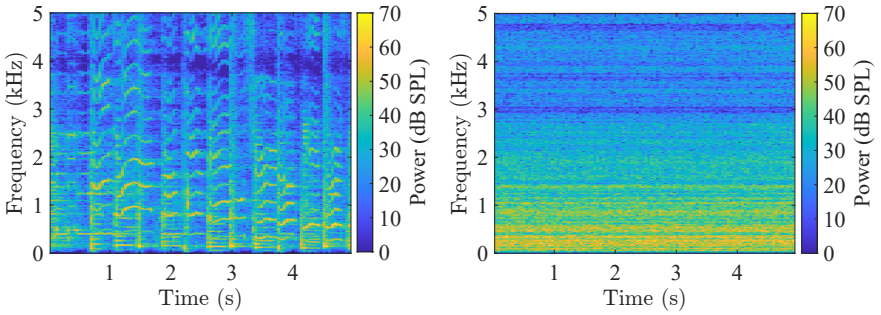
Generation of the stimuli

In this experiment, both audio signal and noise signal are the same as those used in Section 3.4.3, this is, an excerpt of the song "Tell me something

good" by Chaka Khan as the audio signal and a traffic noise as the ambient noise. Both signals are sampled at $f_s = 44100$ Hz with a duration of 30 seconds. In addition, the audio signal has been weighted in loudness as Section 3.4.3 described, while the traffic noise has been weighted in order to provide a SNR of -3 dB regarding the audio signal.

To obtain the different stimulus used in the perceptual test, the system shown in Figure 3.19 has been simulated. To this end, the real-time acoustic response of the system have been measured and the microphone of the real system have been calibrated. Then, using the perceptual equalizer of Figure 3.15, the SPL level of signal $y(n)$ for the five different profiles to be evaluated (NM SF, NM Aures, UAS SF, UAS Aures and NONE) have been obtained. These five simulated signals are the stimulus used in the perceptual test, but they have been trimmed to five seconds to avoid the jury to get tired. More specifically, the simulated signals have been trimmed from the sixth to the eleventh second because the song presents a high variation of its energy during that period.

In order to provide a more in-depth analysis of the audio signal and the traffic signal, their spectrograms are presented in Figure 3.22. Since the traffic noise decreases significantly its power level from 5 kHz on, the spectrograms represent the power spectrum of the signals up to that frequency. In addition, only the five seconds selected are shown.



(a) Excerpt of the song "Tell me something good".

(b) Traffic noise signal.

Figure 3.22. Spectrograms of the simulated signals at the microphone position.

According to Figure 3.22, the traffic noise presents an stationary behavior. In addition, it exhibits the highest power levels at frequencies below 1 kHz in contrast to the audio signal. The power level of the traffic noise decreases as the frequency increases, specially for frequencies above 3 kHz, where the lowest power level of the traffic noise is shown. Regarding the audio signal, it presents a non-stationary behavior, as opposed to the traffic noise. In addition, in Figure 3.22, a lack of frequency components around 4 kHz for the audio signal can be appreciated since the power level observed in that range is significant lower than the rest.

Apparatus and design

The perceptual test was carried out in the same room than the tonality experiment of Section 3.3.5. The same setup has been considered: a M-Track Quad sound card and a pair of Sennheiser HD 600 headphones connected to a laptop with an Intel i7 processor, 8 GB of RAM and a NVIDIA GeForce 940MX graphics card.

A calibration process is required in order to obtain the corresponding SPL values. For this purpose, the dummy head Neumann KU100 has been used in order to measure the sound pressure emitted by the headphones, as shown in Figure 3.23, in order to match the SPL levels between the simulated scenario of Figure 3.19 and the scenario of the perceptual test. Both, the calibration and the perceptual test were implemented on Matlab software.

Finally, for obtaining the stimulus of the perceptual test, M_{T} and M_{S} must be indicated in order to determine the frequency resolution that affects $P_{x,m}(k)$ and consequently the processing to obtain the stimulus. In this case, $P_{x,m}(k)$ has been estimated averaging two frames of 4096 samples with an overlap of 50%, that is, $M_{\mathrm{T}} = 2M_{\mathrm{S}}$ samples and $M_{\mathrm{S}} = N_{\mathrm{FFT}} = 4096$ samples.

Participants

The perceptual test was carried out by 13 participants aged between 20 and 40 years. All of them presented normal hearing and five of them were familiarized with the psychoacoustic research field. None of them were discarded in the results, so the final jury panel was formed by all the participants, 6 males and 7 females.

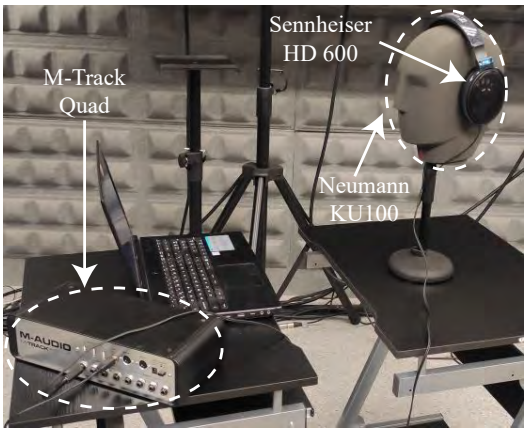


Figure 3.23. System used in the perceptual study.

Procedure

The main objective of the perceptual test is to compare the perception of the different audio signals in presence of the traffic noise for the different profiles, explained Section 3.4.2. For this purpose, the stimulus obtained in the simulated system, depicted in Figure 3.19, are reproduced in the scenario shown in Figure 3.23. These stimulus are compared between them in order to evaluate different audio features. Therefore, the perceptual test is performed through a paired comparison among the different stimuli. A similar test is described in [147], where a jury test evaluated some features of different sound signals, including the preference.

This test has been performed through an application implemented in Matlab, whose main interface is shown in Figure 3.24. This Matlab application can be found in the Audio Processing group website [134], where the steps to design a new test, and execute and analyze an existing test are described. In this case, two features were used to compare the signals: the preference of the participant and the clarity of the audio signal. The first feature indicates which of the two signals to be compared the participant likes most, given the scenario of Figure 3.19. The clarity of the signal indicates which of the two signals to be compared provides a clearer audio signal in presence of the traffic noise, that is, which one is perceived less noisy.

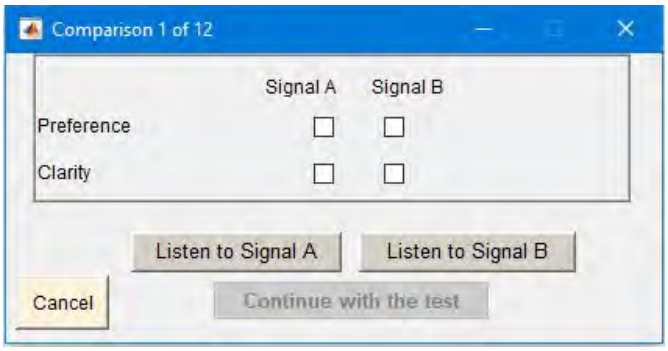


Figure 3.24. Matlab interface for the paired comparison.

The different combinations to be compared are shown in Table 3.3 where the ones assessed in the perceptual test are marked with an "X", leading to a total of twelve different comparisons to be evaluated. As Table 3.3 shows, some combinations have been discarded in order to avoid a very extensive test.

NM SF		X			X
NM Aures	X		X	X	
UAS SF				X	X
UAS Aures	X	X	X		
NONE		X		X	
	NM SF	NM Aures	UAS SF	UAS Aures	NONE

Table 3.3. Combinations of the perceptual test.

Each participant manages the interface shown in Figure 3.24 in order to evaluate the audio signal perception for each combination of Table 3.3 following the next steps:

1. The participant starts the test with the first combination.
2. The participant pushes the buttons to play each of the two stimulus in Figure 3.24.
3. After listening to each of the two stimulus, the participant should

choose one stimuli for every feature, meaning that the chosen stimuli is preferred, or it sounds clearer, than the other. The participant must check all the parameter check boxes.

4. Then, the button "Continue with the test" is enabled and the participant is able to perform the next comparison, where an interface identical to Figure 3.24 is presented with the new comparison.

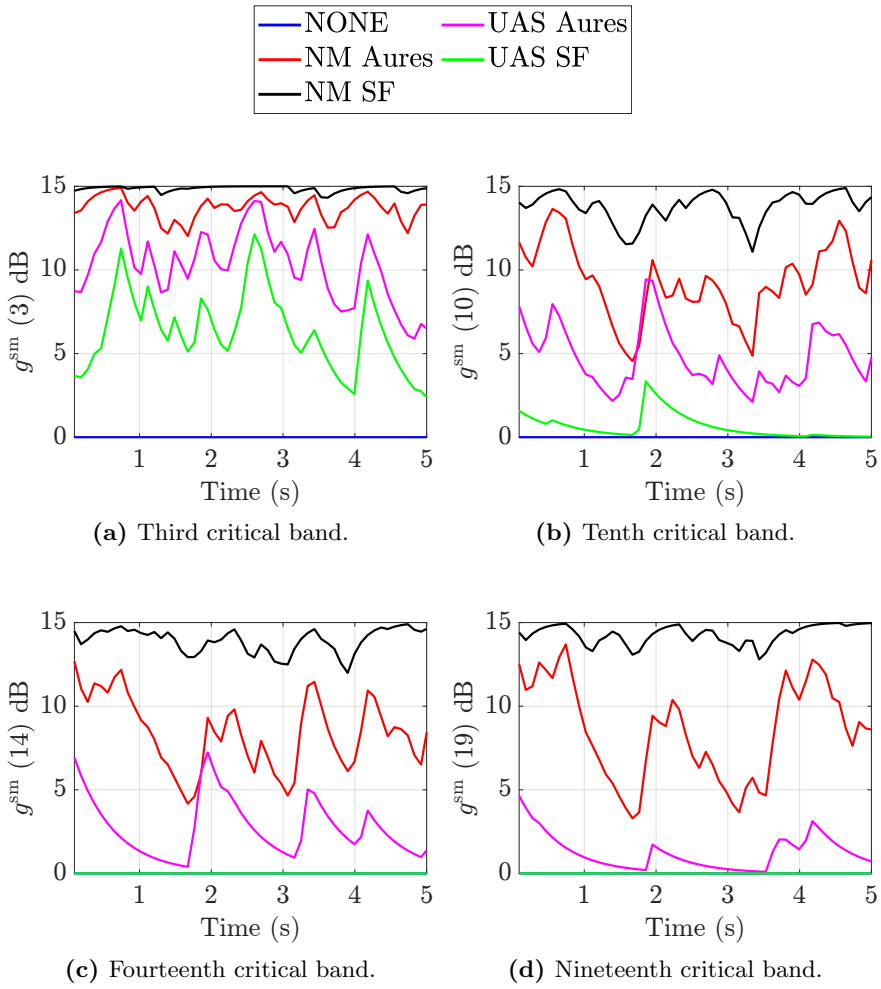
5. Once all the comparisons have been evaluated, the results are automatically saved and the participant finishes the test.

The choice performed by the participants for each comparison will indicate the profiles that present the best improvement of the perception of the audio signal, as well as the preferred profile by the participants.

Results and discussion

The results of the subjective test are analyzed in the following paragraphs. In addition, the gain levels provided by each profile are analyzed in order to support the results of the subjective study. The gain levels over the test audio fragment are shown in Figure 3.25 where four different critical bands are shown since each of them presents significant changes in the power level of the noise signal (see Figure 3.22). For frequencies below 1 kHz, where noise signal presents the highest power level, two critical bands are used, the third critical band (with a center frequency around 250 Hz) and the tenth critical band (with a center frequency around 1 kHz). For frequencies included between 1 kHz and 3 kHz, where the noise power level has decreased with respect to the previous range of frequencies, the fourteenth critical band (with a center frequency around 2 kHz) has been used. Finally, for frequencies above 3 kHz, where the noise signal exhibits its lowest power level, the nineteenth critical band (with a center frequency around 4 kHz) has been used.

As Figure 3.25 shows, for frequencies below 1 kHz (shown through the third and tenth critical bands) all the profiles introduce gain levels. This means that the power level of the traffic noise signal is significantly higher than the power level of the audio signal for frequencies below 1 kHz. Consequently, all the profiles must introduce a gain level in order to accomplish their specific objective (see (3.24) and (3.25)). In addition, for the third critical band, the gain levels estimated through the NM SF profile need



(a) Third critical band.

(b) Tenth critical band.

(c) Fourteenth critical band.

(d) Nineteenth critical band.

Figure 3.25. Gains for each profile.

to be limited (15 dB, see Section 3.4.2). For the tenth critical band, the gain levels introduced by each profile are reduced compared to the previous critical band. This behavior is caused because from 1 kHz, the noise power level starts to decrease, causing each profile is able to fulfill its objective with a lower gain level. The fourteenth critical band presents a very similar gain levels as the tenth critical band for both NM profiles. Therefore, the relationship between the masking of audio signal and the energy of the

noise presents a similar behavior than the tenth critical. In this band, the UAS SF profile does not introduce any gain level, as it also happens with the NONE profile, and therefore, the profile does not consider any masking effect over the audio signal. The last critical band (the nineteenth critical band) exhibits a behavior very similar to that of the fourteenth band. However, the UAS Aures profile present lower gains, as expected given the low power level of the traffic noise at this band (see Figure 3.22).

As Figure 3.25 shows, for the UAS profile, the Aures tonality method introduces higher gains than the SF method, as opposed to the NM profile where an opposite behavior is appreciated. An expected behavior since the Aures tonality method estimates a masking threshold higher than the SF method (see Section 3.3.5).

The gain levels illustrated in Figure 3.25 provide a different equalization over the audio signal according to the profile and tonality method used. Therefore, in order to evaluate this equalization, Figure 3.26 is provided, where the power spectrum of the equalized audio signal at the microphone position is shown. This representation have been performed through the Welch method [112] with an average between the 2^{nd} to the 3^{rd} second, assuming a hamming window of $M_{\text{S}} = 4096$ samples with an overlap of 50% and $N_{\text{FFT}} = 4096$ samples.

As Figure 3.26 shows, the audio signal is only modified (regarding the NONE profile) for frequencies below 1 kHz when UAS SF profile is considered. Regarding the UAS Aures profile, it produces a higher power level audio signal than the UAS SF profile, although from frequencies above 3 kHz the difference with the UAS SF profile is minimal. Finally, the NM profiles always cause an increase in the power level of the audio signal with respect to the NONE profile. These results would seem to imply that the NM SF causes a higher audio signal and consequently a greater improvement of the audio signal perception in presence of the noise signal since it achieves a greater power level in the audio signal, while the UAS SF profile produces the lowest improvement of the perception of the audio signal, compared with the rest of the profiles (excluding the NONE profile), since it provides the lowest power level of the audio signal.

The previous results show a significant difference of the received audio signal between the NM profiles and the UAS profiles. But, in order to evaluate the performance correctly, the results of the perceptual test must be analyzed. The values of merit of this test are shown in Figure 3.27 for

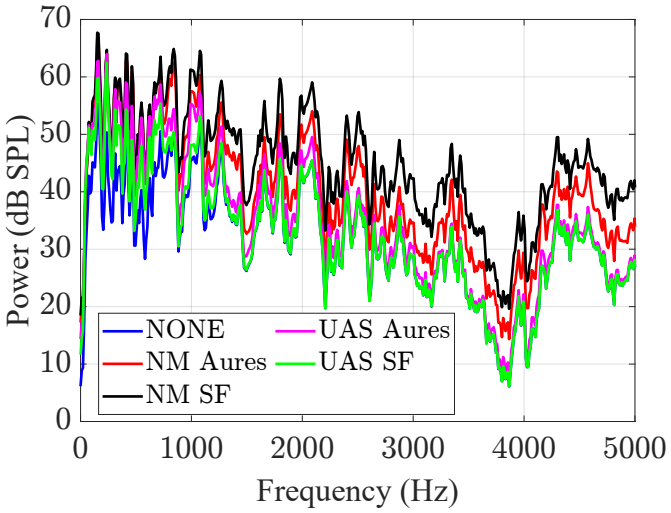


Figure 3.26. Power level of the equalized audio signal at the microphone position.

the different features (preference and clarity of the audio signals), which represent the probability of each profile. That is, the profile with the lowest or the highest value of merit is the least or the most selected profile, respectively, for the specific feature. In contrast, similar values of merit for different profiles indicate that those profiles do not present significant differences for the specific feature, that is, they are difficult to differentiate between them. The sum of the values of merit for an specific feature is equal to 0.

Figure 3.27 shows that the combinations that present greater clarity in presence of the noise (red bars) are the two based on the NM profile. In contrast, a lower clarity of the audio signal is achieved when the UAS profile is considered. According to Figure 3.25, this is an expected behavior since the NM profiles always introduce gain levels greater than the UAS profile, causing a higher power level of the audio signal and consequently a greater masking over the noise. Focusing on combinations based on the NM profile alone, and according to Figure 3.25, the NM SF produces higher gain levels and thus a higher masking. Despite this behavior, the NM Aures presents a better result. This phenomenon is due to the high level of the gains

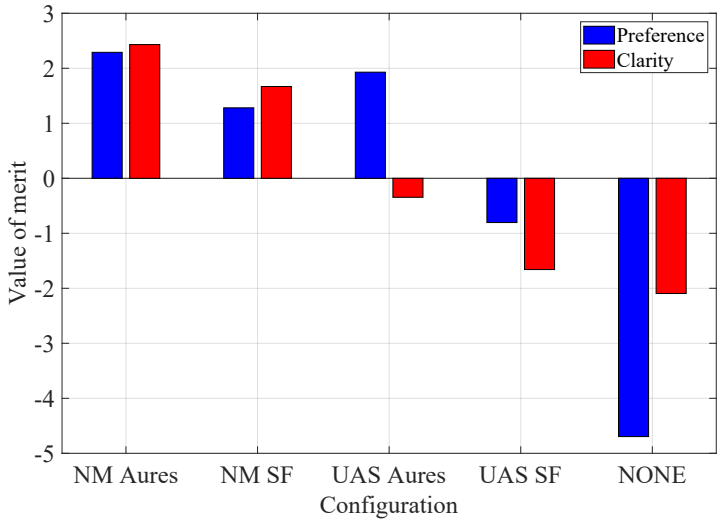


Figure 3.27. Values of merit of the perceptual test.

(and consequently of the audio signal). An extremely high level causes the audio signal to result uncomfortable. This means that the audio signal suffers a deterioration in its perception, reducing its clarity in presence of the noise. As a result, the NM SF profile, that introduces a higher gain levels compared to the NM Aures profile, shows worse values of merit than those of the NM Aures profile. When UAS profile is implemented, the combination that presents a better result is the UAS Aures since, as shown in Figure 3.25, it introduces gain levels greater than those of the UAS SF profile. Finally, the profile labeled as NONE presents the lowest value of merit, which means that the noise is significantly perceived compared to the other profiles.

Regarding the preference (blue bars), the results show a similar behavior to those of the clarity in presence of ambient noise: The best combination is the NM Aures while the worst combination (excluding the NONE profile) is the UAS SF.

3.4.5 Conclusions

In this section, a perceptual application has been implemented with the main objective of increasing the perception of the audio signal in presence of noise through the equalization of the audio signal. The equalizer is performed through a graphic equalizer that is controlled through the gain levels estimated in the perceptual algorithm, that is based on the masking threshold algorithm explained in previous sections.

First of all, different masking patterns have been studied in order to analyze their performance when the perceptual equalizer of this section is used. The results (see Figures 3.20 and 3.21) show that certain combinations present similar behavior than to that shown with the pattern used in [7]. In addition, the proposed patterns are independent of the power level of the specific signal leading to more straightforward methods to estimate the masking threshold. Moreover, the pattern used in this perceptual equalizer produces a linear addition between the different contributions (see "Additivity of masking" process in Section 3.2.2), leading to a masking pattern that presents the most straightforward method for estimating the masking threshold. This could be particularly interesting when the perceptual equalizer is extended to multiple users where the received signal in each microphone is a composition of the emitted signals by each one of the nodes that compose the ASN. Since the contributions of the nodes can be seen as an additional energy to each band of the audio signal regarding the single user case, a masking pattern that produces a linear additivity and does not depends on the energy of the audio signal, will provide an easier analysis.

Once the masking pattern has been selected, a perceptual study has been done in order to analyze the behavior of each one of the profiles implemented in this equalizer. In this case, two different profiles (estimated in two different ways) have been compared in the study, the UAS profile and the NM profile. First of all, a frequency analysis has been done where the estimated gain levels and the power spectrum of the equalized signals of each profile has been shown (see Figures 3.25 and 3.26). According to this data, the NM SF profile provides the best perception of the audio signal in presence of the traffic noise, while the UAS SF profiles provides the worst. Finally, the perceptual test results are shown in Figure 3.27, where the NM Aures profile achieves a significant improvement of the audio signal perception in presence of traffic noise regarding both UAS profiles. In

contrast, the NM SF profile, that produces the highest gains, provides a worse result than NM Aures. This behavior is caused because the NM SF profile generates an uncomfortable audio signal, that is, it generates a very high gain levels, thus decreasing its performance. Focusing exclusively on the UAS profiles, the UAS Aures produces an audio signal in presence of traffic noise clearer than the UAS SF, matching results with the frequency analysis. Therefore, given these results, the profile (NM or UAS) that uses the Aures tonality method achieves a better performance for equalizing audio signals than the same profile using the SF tonality method.

3.5 Smart Active Noise Equalizer

3.5.1 Overview

In the previous section an audio signal is equalized in order to enhance its perception in presence of ambient noise. In contrast, in this section, the ambient noise is controlled for the same purpose, that is, enhance the perception of the audio signal. To this end, algorithms based on Active Noise Control (ANC) principle are used [148], which are commonly applied to cancel undesired noise [149, 150]. In particular, ANE (Active Noise Equalizer) algorithms, that allows to modify the spectral shape of the noise, are considered [151, 152]. Therefore, instead of canceling the noise, an equalization of the noise is proposed.

The ANE algorithms could be used in the considered scenario to reduce the ambient noise and obtain a specific spectral shape. This new approach prevents modifying the audio signal, as opposed to the previous application, avoiding possible undesired effects, such as distortion or saturation of the audio signal, due to a high gain level. Therefore, in this section, similar to the previous one, an equalizer is performed where main objective is to enhance the perception of the audio signal in presence of ambient noise, but with the main difference that the signal to equalize is the ambient noise instead of the audio signal. As a result, a similar scenario where headphones are not available, such as the inside of a car or airplane, is considered. In this case, the scenario shown in Figure 3.28 is considered to perform the equalizer, where two loudspeakers and two microphones are used. Each loudspeaker and microphone are group together and defined a single-channel acoustic node, leading to a two-node ASN with the same

number of microphones (M) and speakers (J) than nodes (N). An additional speaker is also used to emulate the ambient noise, called it "primary noise source".

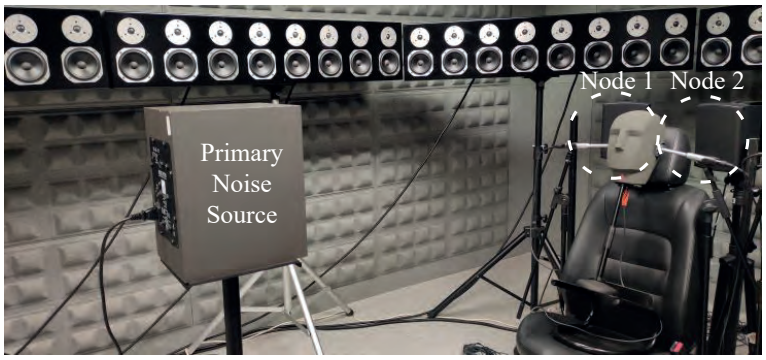


Figure 3.28. Scenario used for equalizing the noise.

The ANE algorithm used in this dissertation is based on the algorithm explained in [152, 153], where the equalization is performed over a multi-frequency noise. According to [152], the multi-frequency noise can be controlled by adjusting the equalization parameters that control each single frequency of the noise, denoted as k_1 with $k_1 = 1, \ldots, K$ being K the total number of frequency components of the ambient noise. These equalization parameters are the attenuation levels (β) and they control each of the frequencies of the noise in order to modify its spectral shape as required. Therefore, the equalization of the noise depends on the attenuation levels that can be estimated in many different ways, same as the gain levels of the previous section (see Section 3.4.2). That is, different strategies (or "equalization profiles") can be used to estimate the attenuation levels.

Therefore, an equalizer based on an ANE algorithm that provides different equalization profiles to modify the spectral shape of the ambient noise is proposed. Additionally, in order to study the performance of this equalizer and its different profiles, a perceptual study is designed. This study evaluates the signals recorded by the microphones of the scenario shown in Figure 3.28 for the different implemented profiles of the equalizer. However, in order to fully understand the perceptual study, firstly the complete algorithm used for equalizing the ambient noise is explained.

3.5.2 Active noise equalization algorithm

The block diagram of Figure 3.29 shows the process to equalize the ambient noise signal. The subscript m in all the signals in Figure 3.29 denote the m^{th} time frame of duration of M_{S} samples. The audio signal (speech or music) emitted by all the speakers is $s(n)$, while the anti-noise signal (or equalized signal) emitted by the speakers ($j = 1, \ldots, J$) is $q_{j,m}(n)$. The attenuation levels ($\beta_{k_1,m}$) used to obtain $q_{j,m}(n)$ are estimated through the audio signal and noise signal at the i^{th} microphone position ($\widetilde{u}_{i,m}(n)$ and $\widetilde{r}_{i,m}(n)$ respectively).

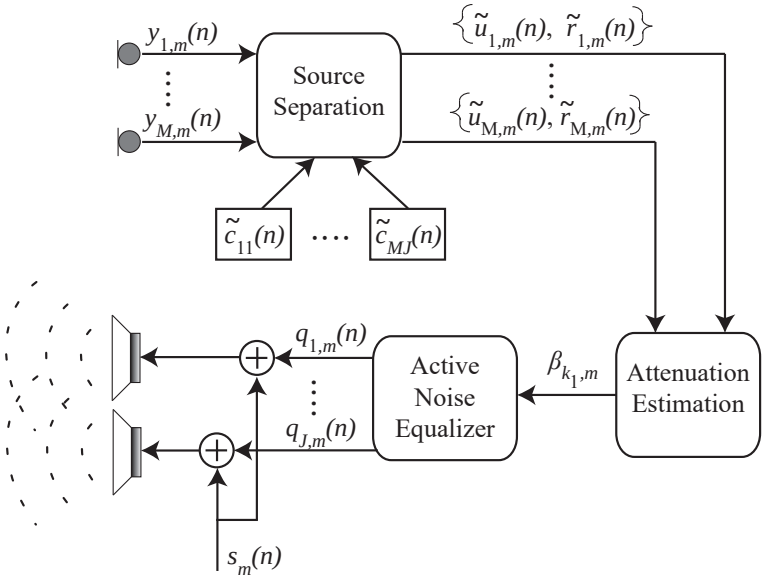


Figure 3.29. Block diagram of Smart Active Noise Equalizer.

This process is very similar to the one shown in Figure 3.15, where the main differences lie on the last block. As Figure 3.29 shows, this algorithm is also divided in three main steps:

1. Source Separation: Identical block to the one shown in Figure 3.15. It is used to obtain separately $\widetilde{u}_{i,m}(n)$ and $\widetilde{r}_{i,m}(n)$.
2. Attenuation Estimation: This block estimates the different attenuation level for each frequency component of the multi-tonal noise,

$\beta_{k_1,m}$. Similar to the previous application, different profiles are implemented to provide different configurations to the spectral shape of the ambient noise.

3. ANE Algorithm: The final block computes the anti-noise signal, $q_{j,m}(n)$, from the attenuation levels estimated in the previous step by means of an adaptive filtering.

In order to fully understand each step, in the following paragraphs a detailed description of each one is made, where for the sake of simplicity, the subscript m denoting the m^{th} frame is omitted. However, the "Source Separation" block will not be explained since it has been previously explained in Section 3.4.2.

Attenuation estimation

Once each contribution ($\widetilde{u}_{i,m}(n)$ and $\widetilde{r}_{i,m}(n)$) has been identified in the "Source Separation" block, the attenuation levels that controls the spectral shape of the anti-noise signal are estimated. These attenuation levels depends on the profile to be used. Since the main objective of this equalizer is to attenuate the ambient noise, the implemented profiles aim to reduce the perception of the ambient noise, that is, they are similar to the NM profile (see Section 3.4.2). As a result, the profiles must reduce the noise below a certain threshold. For this reason, two different profiles are implemented: 1) the hearing threshold (HT) profile and 2) perceptual equalization (PEQ) profile. Each profile estimate the attenuation levels through a different threshold, that is, they perform a different analysis to estimate β_{k_1}. Therefore, in the following paragraphs each profile is described in order to present the main different between them.

Hearing Threshold (HT) profile

This profile aims to force the ambient noise below the human hearing threshold. As a result, the noise power level, $P_{x,m}(k)$, and the standard curve of the human hearing threshold, $L_{\text{TH}}(k)$, must be estimated. The ISO 226 loudness standard [96] is used for estimating $L_{\text{TH}}(k)$ since the curve of 0 phon matches the human hearing threshold (see Figure 2.14). The noise power level at the i^{th} microphone position, $P_{i,r}(k)$, is estimated through the process labeled "Spectral Analysis" in Figure 3.1, that is explained in Section 3.1. Because of the frequency resolution of the FFT (that depends on f_{s} and N_{FFT}), the frequencies related to $P_{i,r}(k)$ may not match the

frequencies of the ambient noise. As a result, $P_{i,r}(k_1)$ is defined as the PS for the closest frequency of the k_1^{th} frequency component of the ambient noise. Similarly, the ISO 226 provides the hearing threshold at certain frequencies that may not match the frequencies of the ambient noise. Then, $L_{\text{TH}}(k_1)$ is defined as human hearing threshold at the closest frequency of the k_1^{th} frequency component of the ambient noise. Considering these definitions, the attenuation levels for the HT profile are estimated as:

$$20\log_{10}\left(\beta_{i,k_1}\right) = \min\left(L_{\text{TH}}(k_1) - P_{i,r}^{\text{dB}}(k_1), 0\right) , \tag{3.41}$$

where β_{i,k_1} is the attenuation level of the k_1^{th} frequency in linear units such that $0 < \beta_{i,k_1} \leq 1$, and $L_{\text{TH}}(k_1)$ and $P_r^{\text{dB}}(k_1)$ are defined in dB units.

Perceptual Equalization (PEQ) profile

This profiles aims to to attenuate the ambient noise below the masking threshold of the audio signal, that is, it aims to mask the ambient noise through the audio signal. It should be noted that system has the same objective than the one analyzed in Section 3.4. As a result, the perceptual analysis explained in Sections 3.2 and 3.3 must be performed in order to obtain the masking threshold of the audio signal, $T_{i,u}(\nu)$, and the energy per critical band of the ambient noise signal, $E_{i,r}^{\text{dB}}(\nu)$, at the i^{th} microphone position. Once both parameters have been estimated, the attenuation levels for the PEQ profile are estimated as:

$$20\log_{10}\left(\beta_i(\nu)\right) = \min\left(T_{i,u}(\nu) - E_{i,r}^{\text{dB}}(\nu), 0\right) , \tag{3.42}$$

where $\beta_i(\nu)$ are the attenuation levels per critical band obtained from the signals of the i^{th} microphone and $T_{i,u}(\nu)$ and $E_{i,r}^{\text{dB}}$ are defined in dB units. Since the attenuation levels must be introduced in the ANE algorithm by frequency component of the ambient noise (f_{k_1}), a reverse mapping from the Bark domain to frequency domain is required. This conversion is performed through next relationship:

$$\beta_{i,k_1} = \beta_i(\nu)\,, \forall\,, f_{k_1} \in \text{BW}_\nu\,, \tag{3.43}$$

where BW_ν is the bandwidth of the critical band ν in hertz (see (3.1)). These attenuations are included in the same range than the previous profile, that is, $0 < \beta_{i,k_1} \leq 1$.

According to [153], the attenuation levels are uniform for different microphones. Therefore, in order to obtain a single attenuation level per frequency, the most restricted attenuation level is considered, that is:

$$\beta_{k_1} = \min_i \left(\beta_{i,k_1}\right) . \tag{3.44}$$

Due to both microphones are close to each other (see Figure 3.28), the attenuation levels can be considered to be similar. Consequently, the same attenuation levels can be used for estimating the anti-noise signals of each speaker.

ANE Algorithm

In this dissertation, the ANE algorithm is not studied in depth since it is not the main topic to be analyzed. However, for a full understanding of the whole process, a brief explanation is given in the following paragraphs. Although, for a deeper understanding of the ANE algorithm, a complete study is performed in [42]. Additionally, one last aspect to consider in order to understand perfectly this algorithm is the fact that no audio signal is reproduced, that is, the microphone only records the noise signal ($y(n) = r(n)$ in Figure 3.29).

In this case, the ANE algorithm is based on [153], where one adaptive filter per frequency component is used. Each filter is composed by two coefficient referring to the in-phase and quadrature components of the reference signal (the ambient noise), that is not necessary to obtain since it can be generated internally if the controlled frequencies are previously known [152]. These filters are estimated through the multiple-error filtered-X LMS (MeFxLMS) algorithm [154, 155], where the objective is to minimize a pseudo-error signal instead of the error signal.

The ANE algorithm is explained through the block diagram of Figure 3.30 that is based on [152, 153]. It describes the signal processing for one generic error sensor i and one generic loudspeaker j. The output of the sinusoidal generator is the reference signal, $x(n)$, composed by K sine waves denoted by $x_{k_1}(n)$ with $k_1 = 1, \ldots, K$.

As Figure 3.30 shows the reference signal is filtered by the adaptive filters, $W(z)$, in order to obtain the outputs signals, $q_j(n)$, that are the anti-noise signals emitted by each speaker of the network. Therefore, the

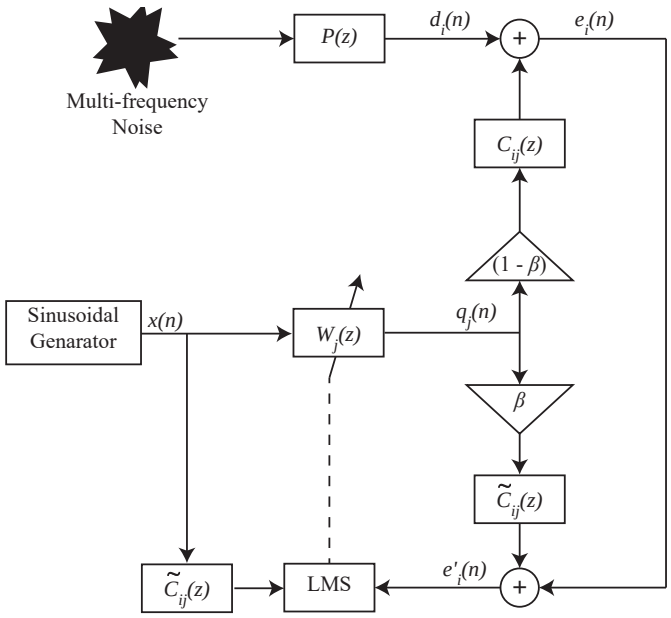


Figure 3.30. Active Noise Equalization block diagram.

main objective is to estimate the adaptive filters, $W(z)$. Firstly, according to Figure 3.30, the error signal in the i^{th} microphone is obtained as:

$$e_i(n) = d_i(n) + \sum_{j=1}^{J} q_j(n) * c_{ij}(n)\,, \tag{3.45}$$

where $*$ stands for the discrete convolution, $c_{ij}(n)$ is the electro-acoustic path between the j^{th} speaker and the i^{th} microphone and $d_i(n)$ is the recorded noise emitted by the primary noise source.

Then, once $e_i(n)$ has been estimated, the pseudo-error signal is computed as:

$$e'_{k_1,i}(n) = e_i(n) + \beta_{k_1} \sum_{j=1}^{J} \left[q_{j,k_1}(n) * \widetilde{c}_{ij}(n)\right], \quad k_1 = 1, \ldots, K;\ j = 1, \ldots, J\,, \tag{3.46}$$

where $\widetilde{c}_{ij}(n)$ is an estimate of the impulse response, $c_{ij}(n)$, and $q_{j,k_1}(n)$ are the anti-noise signals for each of the k_1 frequencies of the reference signal, $x_{k_1}(n)$. Then, the adaptive filters can be estimated through the pseudo-error signal, as the next expression shows:

$$w_{k_1,j}(n+1) = w_{k_1,j}(n) - 2o_{k_1}\sum_{i=1}^{M} e'_{k_1,i}(n)\left[x_{k_1}(n) * \widetilde{c}_{ij}(n)\right] , \tag{3.47}$$

where o_{k_1} is the step-size parameter for the frequency f_{k_1}. Once these filters are estimated, each $x_{k_1}(n)$ is filtered by its corresponding adaptive filter, $w_{k_1,j}(n)$, in order to obtain $q_{j,k_1}(n)$. Finally, all the filtered signal, $q_{j,k_1}(n)$, are combined, in order to obtain the signal to be reproduced by the j^{th} speaker, as:

$$q_j(n) = \sum_{k_1=1}^{K} (1-\beta_{k_1}) q_{j,k_1}(n) . \tag{3.48}$$

3.5.3 Perceptual experiment

In this section, a subjective test has been carried out in order to study the performance of the different profiles explained in Section 3.5.2. Since the PEQ profile depends on the masking threshold, a masking pattern must be determined. In this case, the Schroeder pattern with $\alpha = 1$ is used to estimate the masking since according to Section 3.4.3, it is the most straightforward pattern. In addition, the masking threshold depends on the tonality method, studied in Section 3.3 through two different methods. Therefore, the PEQ profile is studied for the two different tonality methods explained in Sections 3.3.2 and 3.3.3, named hereinafter as "PEQ SF" and "PEQ Aures" profiles for the sake of clarity. Finally, the profile where no equalization is performed also has been considered in the perceptual test as a baseline and it has been labelled as "NONE" profile in the shown results. As a result, four different profiles are compared in the perceptual test:

1. The HT profile. It is based on the hearing threshold and it does not depends on the audio signal.
2. The PEQ SF profile. It is based on the masking threshold (3.20) estimated through the SF tonality method (see Section 3.3.2).

3. The PEQ Aures profile. It is based on the masking threshold (3.20) estimated through the Aures tonality method explained in Section 3.3.3.

4. The NONE profile. A profile where no equalization is performed, that is, $q_j(n) = 0$ for all the speakers.

In order to describe in detail the subjective test, the main features as well as the results obtained are explained in the following paragraphs.

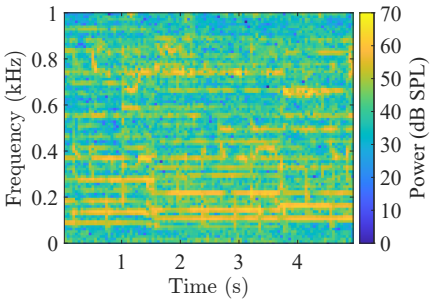
Generation of the stimuli

In this experiment, two different audio signals sampled at $f_s = 44100$ Hz have been used, an excerpt of 30 seconds from the song "Tell me something good" by Chaka Khan, and an excerpt of 30 seconds from the song "Numb" by the group Linkin Park. Additionally, the audio signals have been weighted in loudness as Section 3.4.4 explains in order to make the perceptual test as comfortable as possible. The multi-frequency noise has been implemented in order to emulate the engine car noise. More specifically, a synthesized noise sampled at $f_s = 44100$ Hz with $K = 10$ tones (that simulate the different periodic components of the engine noise), such as $f_{k_1} = (k_1 + 1)47.05$ Hz, with $k_1 = 1, 2, \ldots, 9, 11$ has been used. This noise has been weighted in order to provide a SNR of -5 dB regarding the audio signals.

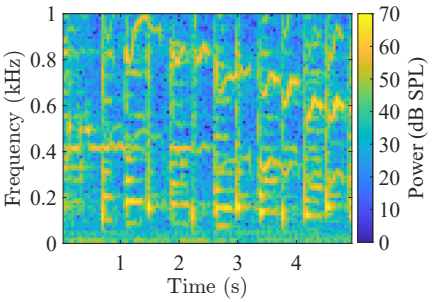
Both set of signals are introduced in the scenario shown in Figure 3.28, where synthesized noise is emitted by the primary noise source and the audio signals are emitted by the speakers of the acoustic nodes. Through the real-time recording and reproduction of these signals and the processing seen in Figure 3.29, the recorded signals, $y_i(n)$, at each microphone are obtained. These recorded signals are the stimulus used in the perceptual test. However, they have been trimmed to five seconds in the perceptual test to avoid performing a large test. The recorded signals that belong to the excerpt of Chaka Khan have been trimmed from the sixth to the eleventh second, while the recorded signals of the other excerpt have been trimmed from the tenth to the fifteenth second. In the first excerpt, these five seconds have been selected since the audio signal presents a high variation in the energy of the signal, due to the introduction of the loud female voice. The five seconds selected in the second excerpt present a more stationary behavior, but with a wider spectrum than the first one.

Additionally, the Figure 3.31 shows the spectrograms of each audio

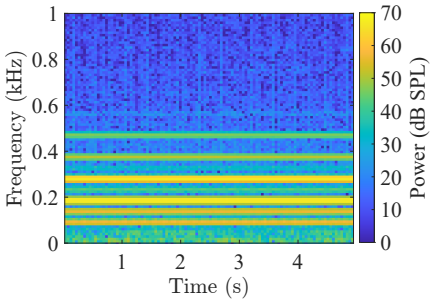
signal and synthesized noise at the position of the first microphone of Figure 3.28 during the five seconds selected (from the sixth to the eleventh second in the "Tell me something good" excerpt and from tenth to the fifteenth second in the "Numb" excerpt). Because the noise is located at frequencies below 1 kHz, all the spectrograms of Figure 3.31 have been represented up to that frequency.



(a) Excerpt of the song "Numb".



(b) Excerpt of the song "Tell me something good".



(c) Recorded engine noise.

Figure 3.31. Spectrograms of the recorded signals at the first microphone position.

As Figure 3.31 shows, the received noise is a stationary noise composed by several tonal components with more energy for the frequency components below 400Hz. The recorded signal of the excerpt of the song "Tell me something good" exhibits a non-stationary spectrum with frequency components of changing levels depending on the frame. Finally, the excerpt of the song "Numb" presents a more stationary behavior for frequencies

below 1 kHz than the previous excerpt. In fact, this excerpt shows a similar representation than the received noise in some frequency components, specially in components around 200 Hz.

Apparatus and design

The apparatus and design of this test is identical to the subsection "Apparatus and design" described in Section 3.4.4.

Participants

The perceptual test was carried out by 14 participants aged between 20 and 40 years. All of them presented normal hearing and five of them were familiarized with the psychoacoustic research field. None of them were discarded in the results, so the final jury panel was formed by all the participants, 8 males and 6 females.

Procedure

The perceptual test aims to compare the perception of the different audio signals in presence of the ambient noise for the different profiles, explained in Section 3.5.2. Therefore, the four different profiles are compared in order to study their performance from a subjective point of view. For this purpose, the stimulus recorded in the scenario shown in Figure 3.28 are reproduced in the scenario shown in Figure 3.23 through a paired comparison test, where different combinations are tested. These combinations are shown in Table 3.4, where the combinations to be compared are identified by an "X" and the rest are discarded.

PEQ SF		X		X
PEQ Aures	X		X	X
HT	X			X
NONE		X		
	PEQ SF	PEQ Aures	HT	NONE

Table 3.4. Combinations of the perceptual test.

From Table 3.4, eight different paired comparisons are appreciated. However, these combinations only represent the comparisons made for a single audio signal. Therefore, considering both audio signals ("Numb"

and "Tell me something good"), a total of sixteen paired comparisons are carried out in the test.

Same as the perceptual test performed in Section 3.4.4, this one is based on [147], where the different combinations shown in Table 3.4 are compared in order to evaluate different features. In this case, same features than the test performed in Section 3.4.4 are evaluated, this is, the clarity of the audio signal and the preference. Due to the similarity with the test explained in Section 3.4.4, same application implemented in Matlab is used, see Figure 3.24. Each participant manages the interface shown in Figure 3.24 in order to evaluate the audio signal perception for each combination of Table 3.4 following the next steps:

1. The participant starts the test with the first combination.
2. The participant reproduce each of the two stimulus in Figure 3.24.
3. After reproducing each of the two stimulus, the participant should choose one stimuli for every feature, meaning that the chosen stimuli is preferred, or with a clearer audio signal, than the other. The participant must check all the parameter check boxes.
4. Then, the button "Continue with the test" is enabled and the participant is able to perform the next comparison, where an interface identical to Figure 3.24 is presented with the new comparison.
5. Once all the comparisons have been evaluated, the results are automatically saved and the participant finishes the test.

The choice performed by the participants for each comparison will indicate the profiles that present the best improvement of the perception of the audio signal, as well as the preferred profile by the participants. Additionally, to avoid fatigue in the participants (and consequently avoid errors), the test is divided into two, where each one is focused on a specific audio signal ("Numb" and "Tell me something good") leading to two different test of eight paired comparisons each one.

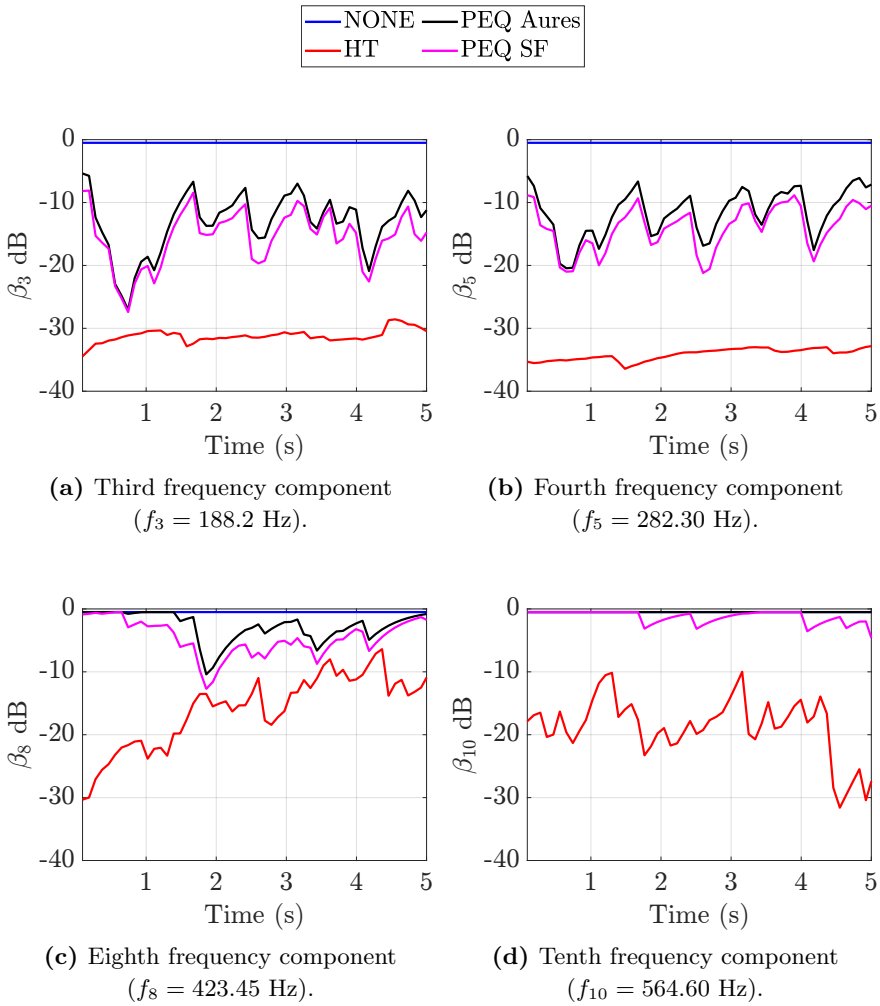
Results and discussion

The results of the subjective test are analyzed in the following paragraphs. In addition, the attenuation levels provided by each profile are analyzed

in order to perform a more complete study. These results are shown in Figure 3.32 and Figure 3.33 where the first one is focused on the excerpt of the song "Tell me something good" and the second one is focused on the excerpt of the song "Numb". Only four frequency components are presented since each of them presents differences in the power level of the received noise (see Figure 3.31. Additionally, each of them represents a different critical band in the perceptual analysis performed in the PEQ profile. These results are represented for the five seconds selected in each signal

Figure 3.32 shows the attenuation levels for the five seconds selected (from sixth to the eleventh second) when the excerpt of the song "Tell me something good" is used. For the third and fifth frequency components, both PEQ profiles produce a similar attenuation levels that are included between -10 and -20 dB, while for the HT profile an attenuation level around -30 and -35 dB is presented. Since both PEQ profiles depends on the audio signal, the attenuation levels present a wide fluctuation due to the non-stationarity of the audio signal. In contrast, the HT profile presents very stable attenuation levels compared to those of the PEQ profile due to the stationary behavior of the ambient noise. For the eighth frequency component, the attenuation levels are lower than the previous frequency components since this one has a lower power level (see Figure 3.31). Finally, for the last frequency component, the PEQ profiles consider a very low attenuation level, specially the PEQ Aures profile, where we observe a behavior very similar to that of the NONE profile. This means that the PEQ Aures profile does not introduce attenuation because it assumes that audio signal masks the ambient noise in that frequency component.

Nevertheless, the HT profile introduces a higher attenuation than both PEQ profiles since it considers the human hearing threshold, a threshold far below the masking threshold of the audio signal. As a result, the HT profile will produce a higher attenuation on the noise. It should be noticed that the eighth and tenth components do not presents a stable attenuation levels in the HT profile. This result is caused because of the estimate of the electro-acoustic path ($\widetilde{c}(n)$) is slightly inaccurate, causing that recorded noise ($r(n)$) cannot be completely extracted by using the "Source Separation" block (see Figure 3.29). Then, the noise obtained in the "Source Separation" block contains a small contribution of the audio signal, causing the fluctuation of the attenuation levels in those components.



(a) Third frequency component ($f_3 = 188.2$ Hz).

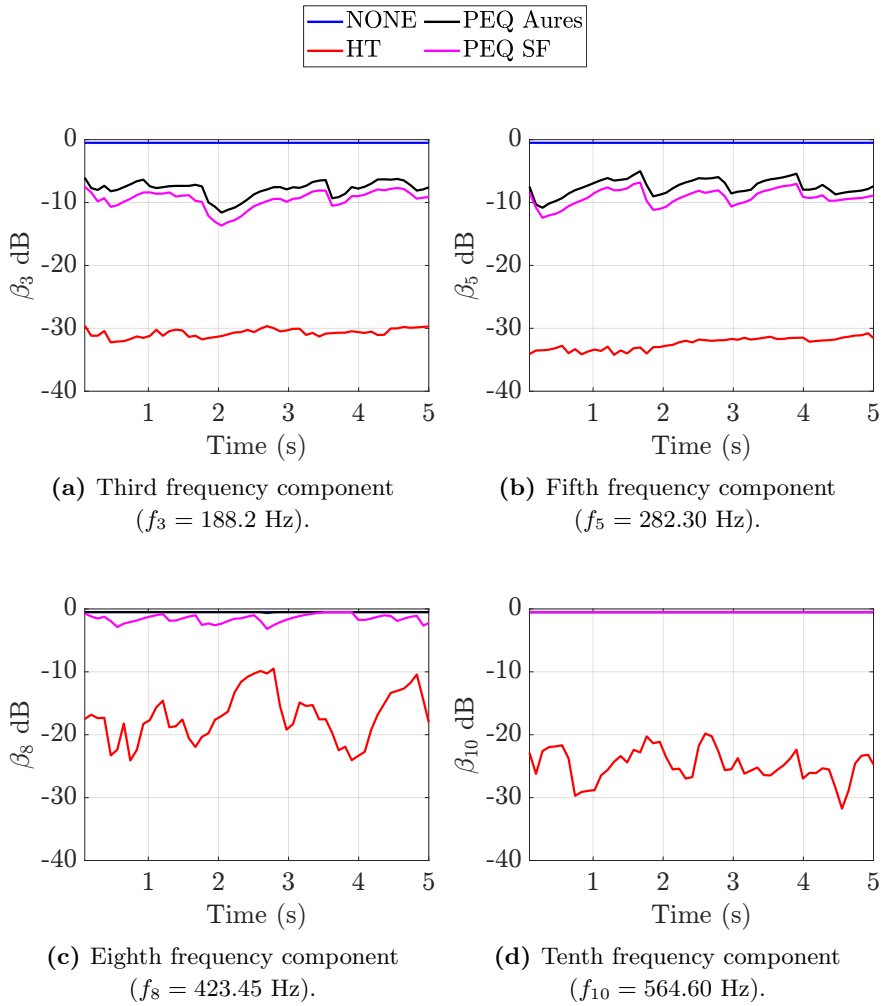
(b) Fourth frequency component ($f_5 = 282.30$ Hz).

(c) Eighth frequency component ($f_8 = 423.45$ Hz).

(d) Tenth frequency component ($f_{10} = 564.60$ Hz).

Figure 3.32. Attenuation levels for the excerpt of the song "Tell me something good".

Figure 3.33 shows the attenuation levels for the five seconds selected (from tenth to the fifteenth second) when the excerpt of the song "Numb" is used. A similar behavior than the attenuation levels shown in Figure 3.32 can be appreciated. That is, the attenuation levels of the HT profile are higher than the PEQ profiles in any case due to the hearing threshold is



(a) Third frequency component ($f_3 = 188.2$ Hz).

(b) Fifth frequency component ($f_5 = 282.30$ Hz).

(c) Eighth frequency component ($f_8 = 423.45$ Hz).

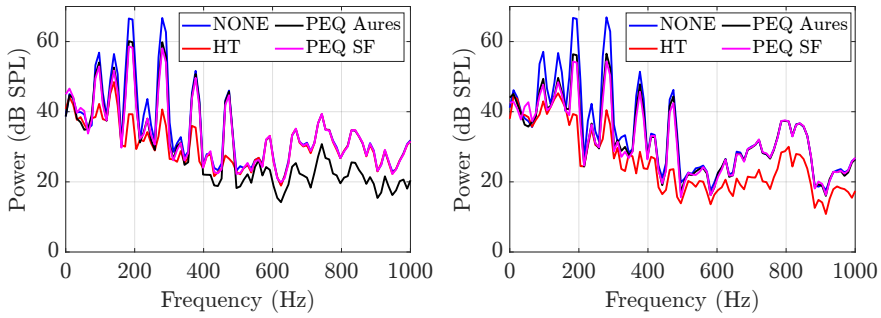
(d) Tenth frequency component ($f_{10} = 564.60$ Hz).

Figure 3.33. Attenuation levels for the excerpt of the song "Numb".

a more restrictive curve than the masking of the audio signal. Regarding the PEQ profiles, both of them present a similar attenuation levels, being the PEQ Aures the profile that introduces a lower attenuation levels. Same as the previous results, the attenuation levels are lower as the frequency increase due to the ambient noise present a lower power level in the last frequency components (see Figure 3.31). In addition, since this excerpt

presents a more stationary behavior than the previous one, the attenuation levels are more stable. Furthermore, both PEQ profile do not introduce any attenuation for the last two frequency components. This means that the PEQ profiles consider that audio signal is already masking the ambient noise for those components. As Figure 3.33 shows, the last component present a higher attenuation level for the HT profile, disagreeing with the above argument. This behavior is caused because of the inaccuracy of the estimate of the electro-acoustic path ($\widetilde{c}(n)$), just as the previous results.

These attenuation levels will cause a specific anti-noise that is reproduced by the speakers of the acoustic nodes ($q_j(n)$, see Figure 3.28) in order to reduce as much as possible the noise in the microphones and increase the perception of the specific audio signal. To appreciate the amount of noise that the microphones receive, Figure 3.34 illustrates the power spectrum of the noise recorded by the first microphone for the five seconds selected of each audio signal is shown. The power spectrum has been estimated through the Welch method [112] with an average between the 2$^{\text{nd}}$ to the 3$^{\text{rd}}$ second assuming a hamming window of M_{S} samples with an overlap of 50% and $N_{\text{FFT}} = 4096$ samples.



(a) Excerpt of the song "Numb".

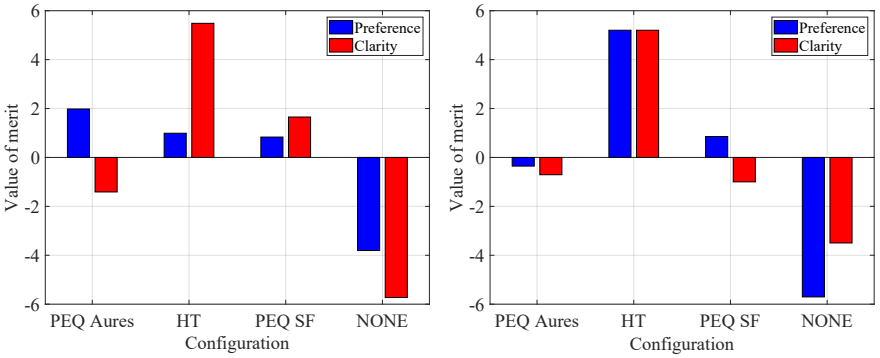
(b) Excerpt of the song "Tell me something good".

Figure 3.34. Received noise power level at the first microphone.

As Figure 3.34 shows, the received noise is lower for the PEQ profiles when the excerpt of "Tell me something good" is used because attenuation levels are higher in this excerpt. This behavior can be better appreciated in the frequency components between 200 and 400 Hz where a difference

around 5 dB between both excerpts can be appreciated. When the HT profile is considered, the recorded noise presents the highest attenuation regardless the frequency component and audio signal. It should be noted that this behaviour is expected since the HT profile introduces the highest attenuation levels among all the profiles.

The previous results show a significant difference of the received noise between the HT and both PEQ profiles. But, in order to evaluate the performance correctly, the results of the perceptual test must be analyzed. The values of merit for each specific feature (preference and clarity) are shown in Figure 3.35, which represent the probability of each profile for each feature. That is, the profile with the lowest or the highest value of merit is the least or the most selected profile, respectively, for the specific feature. In contrast, similar values of merit for different profilesindicate that those profiles do not present significant differences for the specific feature, that is, they are difficult to differentiate between them. The sum of the values of merit for an specific feature is equal to 0.



(a) Excerpt of the song "Numb".

(b) Excerpt of the song "Tell me something good".

Figure 3.35. Values of merit of the perceptual test.

According to Figure 3.35, the profile that presents a clearer audio signal is the HT profile. Therefore, the HT profile achieves the best improvement of the perception of the audio signal in presence of the ambient noise, an expected behavior since it introduces the highest attenuation levels. Regarding the PEQ profiles, the PEQ SF shows a slightly higher values of

merit than the PEQ Aures, that is, the PEQ SF provides a slightly better performance than the PEQ Aures. However, a further study should be carried out since the differences are minimal, specially for the excerpt of the song "Tell me something good". This behavior is caused because, as stated in Section 3.3.5, the SF tonality method estimates a lower masking threshold than the Aures method, leading in this case to higher attenuation levels, see (3.42). Finally, the profile labeled as NONE is the profile that participants less select since no equalization is performed in this profile.

Regarding the preference feature, the participants share a general view on the clarity in the excerpt of the song "Tell me something good". Although, certain differences can be appreciated for the PEQ profiles, they are not relevant since the difference between them is minimal. However, for the excerpt of the song "Numb" the preference feature presents a different behavior. In that case, the values of merit of both PEQ profiles and the HT profile are similar, that is, the participants are equally divided between the PEQ SF, PEQ Aures and HT profiles. The different views have emerged due to the spectral composition of this specific audio signal. As Figure 3.31 shows, the excerpt of the song "Numb" presents similar stationary components that the noise, specially around 200 Hz. As a result, the attenuation level also affects to the audio signal causing that the participants are more declined to the PEQ profiles since they introduce a lower attenuation levels.

Therefore, the HT profile achieves a better increase of the perception of the audio signal since it introduces a higher gains regarding the PEQ profiles. However, with a lower attenuation levels the PEQ profiles also achieves to increase the perception of the audio signal regarding the NONE profile since when any of the PEQ profiles are compared to the NONE profile, the participants are able to distinguish them without error.

3.5.4 Conclusions

A second approach for equalizing in order to increase the audio signal perception has been presented in this section. In this case, through the ANE algorithm, the noise is equalized in order to reduce it according to the implemented profiles. These different profiles are used to provide a specific spectral shape to the noise, increasing the flexibility of the equalizer.

Similar to the previous section, in order to analyze the performance of the different profiles, an experimental study is performed. However, firstly

the attenuation levels for certain frequencies are shown. According to these results (see Figure 3.32 and Figure 3.33), the HT profile presents the highest attenuation levels between all the profiles to be compared. Both PEQ profiles present similar attenuation levels, although the PEQ SF profile provides slightly higher attenuation levels, an expected behavior since the PEQ SF profile estimates a lower masking threshold. This behavior leads to generate a noise in the speakers of the acoustic nodes with a power level higher for the HT profile in comparison to the PEQ profiles, as Figure 3.34 shows.

In a second stage, a perceptual test between the different profiles has been performed. A paired caparison test with sixteen different combinations are implemented. The results, shown in Figure 3.35, conclude that all the profiles (HT, PEQ Aures and PEQ SF) improve the perception of the audio signal, being the HT profile the best one. However, these results show that the preference of the participants is more equally distributed when audio signal provides frequency components similar to those of the ambient noise. Finally, the PEQ Aures and PEQ SF are difficult of differentiate due to their similar values of merit in both audio signals. This means that the additional attenuation level introduced in the PEQ SF profile is not significant because the improvement provided by the PEQ SF cannot be significantly perceived, in contrast to the improvement provided by the HT profile.

3.6 Conclusions

In this chapter, the perceptual algorithm based on the masking effect has been explained in detail and use it in order to implement a perceptual application over an indoor acoustic environment, such as the cabin of a car. Since recording and reproduction operations are needed as well as processing, this scenario is presented through an ASN composed by single-channel acoustic nodes.

According to Figure 3.1, the masking of a signal is estimated through two main steps, the process of the upper branch called “Auditory Masking Model” and the process of the bottom branch called “Tonality Estimator Model” that are deeply explained along the respective sections. In addition, since the process of the bottom branch is inaccurate in certain cases

(such as complex noise signals [120] or speeches [125], in this chapter an alternative method is proposed, the Aures method [127]. Finally, in order to compare the conventional method (called in this dissertation as Spectral Flatness method) with the alternative method, a perceptual study is performed. This study obtains the masking threshold obtained from the multiple test made by several participants to obtain a real measurement and compare this result with the analytical results provided by each of the methods. According to the results of this study (see from Figure 3.12a to Figure 3.12c) both tonality methods provide a similar masking threshold in low frequencies than the perceptual results. On the other hand, for middle and high frequencies, the masking threshold provided by the Aures method show a better performance than the masking threshold of the Spectral Flatness method, in other words, the results provided by the Aures method fit better with the perceptual results. Therefore, given these results, the Aures method can be assumed to be a more accurate method, since he Spectral Flatness method is inaccurate for middle and high frequencies, just as concluded in [124, 125].

The masking method is used to implement a perceptual analysis in order to perform an application that aims to enhance the perception of the audio signal in presence of the noise signal in the acoustic environment presented above. For this purpose, an equalizer based on the above masking method is introduced, where two different approaches are implemented.

On one side, an equalization on the audio signal is presented through a graphic equalizer. For this purpose, the audio signal must be boosted in order to enhance its perception. This behavior is achieved by using the perceptual analysis to obtain the corresponding gains that will feed the graphic equalizer. Although, first at all a study to analyze the best masking pattern is performed in order to propose the most straightforward method to estimate the masking threshold. This method establishes the basis for the extension of the perceptual equalizer to multiple users (see Section 3.4.4). On the other hand, using that masking pattern, a perceptual study is performed in order to analyze the performance of the different equalization profiles performed in this application (see Section 3.4.1). According to the results of this study, the combinations where masking threshold has been estimated through the Aures method tonality offers a more comfortable equalization than the combinations where masking threshold has been estimated through the Spectral Flatness method. In other words, the masking

threshold estimated through the Aures tonality method achieves a better performance in this equalizer.

On the other hand, the second approach aims to enhance the perception of the audio signal in presence of the noise by means of an ANE algorithm in order to control the spectral shape of the noise. Similar to the previous approach, a perceptual test has been performed in order to analyze the different profiles implemented in this application. According to the results, in this case, the perceptual profile (PEQ) presents very similar behavior regardless to the tonality method employed, in other words, the difference between the attenuations of each PEQ profile is not perceptible. On the other hand, the HT profile presents the best results since it provides the highest attenuations, in other words, the equalization through the PEQ profiles is not able to mask the noise signal. This behavior is caused because the estimated masking threshold does not match perfectly with the real masking threshold, an expected behavior according to the results of Section 3.3.5 where the estimated masking thresholds are not identical to the real masking thresholds. In addition, in this case the noise signal is composed by more than four tones, unlike the signals considered in Section 3.3.5. For this reason, in order to adjust even more the estimated masking threshold of the broadband signals a further study is proposed.

4 Analysis of an Android ASN

This chapter presents an ASN composed by commercial mobile devices connected to wireless speakers. The mobile devices use the Android operating system and the wireless connections are based on Bluetooth and Wi-Fi protocols. The purpose of this chapter is to study the synchronization between the mobile devices of the ASN in order to perform synchronized reproduction through multiple speakers. Therefore, first at all, the design of the network is explained in order to present the scenario where synchronization is studied. Once the network is established, all the parameters that affect the audio latency are described and analyzed. Subsequently, a solution to overcome the synchronization problems is proposed and evaluated through experiments in a real scenario. Finally, different Personal Sound Zones (PSZ) applications are presented with the purpose of analyzing the performance of these applications once the synchronization problems have been addressed.

4.1 Acoustic Network Model

In Section 2.2, the mobile devices have been proposed as an acoustic node since they are equipped with one or more microphones and speakers and a very powerful processor capable of computing large amount of data. Therefore, a mobile device is able to play, record and process multimedia data in order to produce a specific audio performance inside different acoustic environments. Since the loudspeakers of the mobile device are usually very poor in quality, then wireless speakers can be added to the network, as it is shown in Figure 4.1. As it was described in Section 2.2.1, the mobile devices that form the network use the Android Operating System (OS).

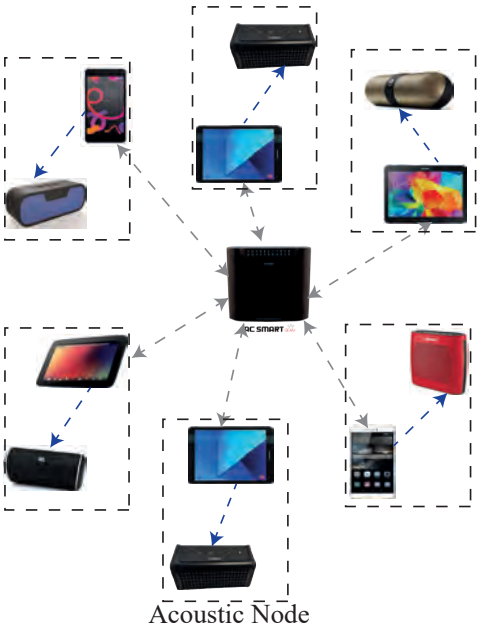


Figure 4.1. Example of an ASN composed by mobile devices and wireless loudspeakers

As shown in Figure 4.1, all the mobile devices are connected to a Wi-Fi router in order to exchange information between them. The Wi-Fi communication is represented by the gray dotted line in Figure 4.1 and it is a two-way communication. It can also be appreciated that each mobile

device is connected to a wireless speaker through the Bluetooth protocol in order to transmit audio streams with enough quality. The Bluetooth communication is represented by the blue dotted line and it is a one-way communication, from the mobile device to the speaker. This last communication allows separating the microphone and the speaker at any distance within the coverage area of the Bluetooth link. Due to the characteristics of Bluetooth connection (A2DP [74]), only one connection can be performed at the same time, unlike the Wi-Fi protocol where several connections can be made simultaneously. Therefore, each mobile device manages two different wireless connections at the same time: one based on Wi-Fi to exchange local information through the network and the other based on Bluetooth in order to reproduce sound over the speaker.

Since a single Bluetooth connection can be performed and mobile device usually is able of recording single-channel signals, in this dissertation, each mobile device and speaker will manage single-channel audio signals. Therefore, the set formed by one mobile device and one speaker combined, as Figure 4.1 shows, will be considered as a single-channel acoustic node (see Figure 2.3a). As said before, the network shown in Figure 4.1 can be used to perform different audio applications involving audio reproduction [156, 157]. As it is explained in Section 2.1.3, the network shown in Figure 4.1 can be considered an active ASN.

Reproduction applications over acoustic networks need a precise control of synchronization between the nodes. When the acoustic nodes are perfectly synchronized, they can record or play audio signal simultaneously as a wired system with a single processor would do it. Therefore, perfect synchronization of ASN, in particular, ASN over Android devices and Bluetooth and Wi-Fi connections, is of great interest. Therefore, in order to perform a detailed analysis on the relevant factors that affects the synchronization, the study of this chapter is focused on a two-node acoustic network, such as the one shown in Figure 4.2.

The synchronization issue is a well-known design challenge in an ASN (see Section 2.1.4) because of the uncertainly with wireless communications. Due to its complexity, in this dissertation, the synchronization challenge is divided into two problems that will be addressed separately for the sake of clarity. The first problem to be addressed will be denoted as the **clock synchronization** problem [158, 159] because it is related to the different time at any device clock. The clock synchronization problem causes that

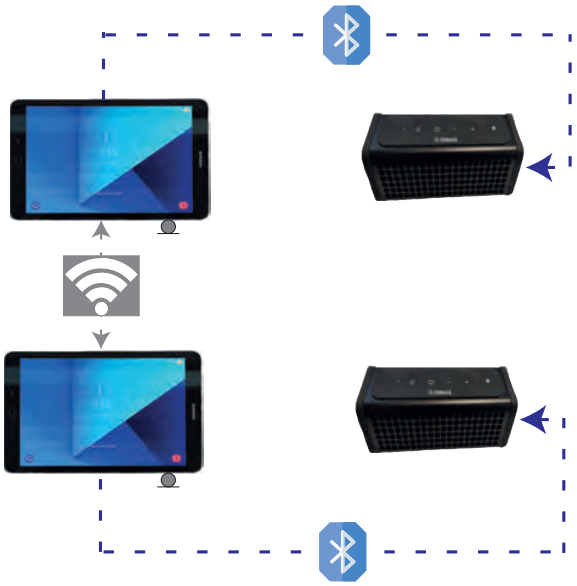


Figure 4.2. Two-node ASN with two Android devices and two wireless speakers.

the acoustic nodes are not able to start a task (such as an audio task) at the same exact time all together. It will be explained in detail in Section 4.3. The second problem to be addressed is related to physical aspects of the loudspeakers and microphones (or devices since they are built-in microphones) and the software used. It prevents the audio signals arriving at the microphones to be temporarily not aligned, even when the first problem (clock synchronization) is corrected. Due to the different physical locations and software and hardware between the nodes, the signals emitted by the speakers do not reach the microphones at the same time. This problem can be considered as another type of synchronization related to the audio application when the signals from all the loudspeakers must arrived at the same time at one particular location. But to prevent confusion with the other problem, in this dissertation, it is called the **temporal alignment** problem and it is addressed in Section 4.4.

Since synchronization deals with the time alignment of the signals emitted by each node, an objective metric to measure the time differences be-

tween signals is required. For this reason, the parameter called here audio latency is firstly studied and analyzed in the following section for a full understanding of the methods employed to solve each of the problems mentioned above. The fundamental actions affecting the audio latency will be explained, and afterwards, each of the above problems, clock synchronization and temporal alignment, will be addressed, solved and tested through experimental measurements. Finally, certain audio applications will be implemented and their performance will be studied. More specifically, as an specific application of ASN, Personal Sound Zone (PSZ) applications will be implemented, whose main objective is to generate different audio zones within the same room.

Finally, the proposed algorithms able to manage the synchronization between nodes and implement multi-speaker audio reproduction has been developed over Android and can be installed as an application. Since its use is not essential for the correct understanding of the issues proposed in this chapter, the application is explained step by step in Appendix A, where the block diagram of their different interfaces is shown. The application has been created with the environment called **Android Studio** that is specific for programming Android applications. The application creates the ASN shown in Figure 4.2 through the corresponding wireless connections. In addition, it addresses the synchronization of the network regarding the two problems mentioned above, and finally, it implements the different PSZ systems explained in Section 4.5.

4.2 Study of the audio latency

4.2.1 Definition of the audio latency

Considering the whole chain of playing and recording an audio signal, the audio latency is usually defined as the time an audio sample needs to throughout the whole audio chain [160]. In the context of an acoustic node as those shown in Figure 4.2, the audio latency, denoted hereafter by T_{AL}, is defined as the time between sending an audio sample for being played and the time that the same audio sample is recorded by the microphone. More specifically, it is the delay that the sound suffers due to the electro-acoustic path of the system, which includes delays caused by the software, the wireless link and the sound propagation between the

speaker and the microphone, as shown Figure 4.3. Since T_{AL} depends on the specific electro-acoustic path, in the ASN of Figure 4.2 there will be as many T_{AL} as electro-acoustic paths. In a network of N single-channel acoustic nodes, there will be N^2 electro-acoustic paths, resulting in N^2 different T_{AL}. In order to simplify the description of T_{AL}, only one acoustic node is considered to study the generation of T_{AL}. As it can be seen from Figure 4.3, the audio latency can be modelled as the sum of four different delays:

$$T_{AL} = T_{DP} + T_{BL} + T_{SP} + T_A \,, \tag{4.1}$$

where T_{DP} is the delay caused by the software and hardware of the mobile device, T_{BL} is the delay of the Bluetooth link, T_{SP} is the delay caused by the software and hardware of the wireless speaker and T_A is the acoustic delay.

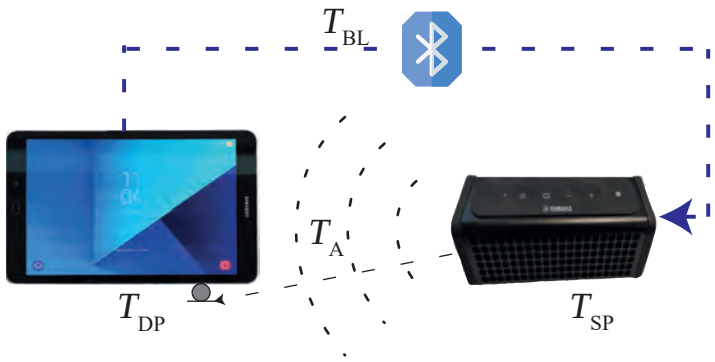


Figure 4.3. Electro-acoustic path for a single acoustic node.

Due to their influence over the total measurement of T_{AL} a comprehensive and detailed explanation of each factor is given in the following paragraphs. All the factors are explained assuming that Android mobile devices are used as acoustic nodes.

Device Processing Latency

The delay caused by the data processing in the device is represented by T_{DP}. Basically, it can be assumed that T_{DP} is caused by the Android OS, since it manages and controls all the processes, including the audio processing.

Since the device runs both, recording and playing processes, T_{DP} is formed by the audio output and audio input latency, $T_{\mathrm{DP}} = T_{\mathrm{OLD}} + T_{\mathrm{IL}}$, where T_{OLD} is the output latency of the mobile device side and T_{IL} is the input latency. The input latency, T_{IL}, can be defined as the elapsed time the system needs to make the audio data accessible in digital form from the microphone recording, while the output latency, T_{OLD}, can be defined as the elapsed time the system needs to render a sound from the stored digital format. In the case of Figure 4.3, where a Bluetooth connection is used, T_{OLD} is re-defined as the time the Android OS requires to process the sound and sending it to the Bluetooth link.

On the other hand, as indicated in Section 2.2.2, Android OS is decomposed into several layers named as the Android stack (see Figure 2.8). Since only audio tasks are involved in this definition, the stack can be specified in order to only take into account the important features related with the audio processing, as it is shown in Figure 4.4 [161], where the different colors represent the different layers of the stack.

This stack shows the steps that audio signals must follow in order to be reproduced or recorded, that is, affecting T_{OLD} and T_{IL}. The audio reproduction starts in the "Application Framework" layer where the entire feature set of the Android OS is available. Then, audio reproduction proceeds through the intermediate layers, where the core components of the OS are accessed. The audio reproduction ends in the "Linux Kernel" Layer, where the drivers of the hardware are managed. The recording process considers the same steps, but in reverse order (from the "Linux Kernel" Layer to the "Application Framework" layer).

Therefore, T_{OLD} and T_{IL}, and consequently T_{DP}, highly depend on the performance of the stack shown in Figure 4.4, leading to longer delays when the performance of the stack is bad. In addition, when a Bluetooth connection is enabled, additional operations are required [162] in order to adapt the audio signal to the Bluetooth link, causing an increase of T_{OLD}. However, as explained in [163], the performance of T_{OLD} can be slightly controlled through some parameters that are specific for each Android device model. These parameters are the sample rate (f_{s}), the size or length of the data buffer [164] (L_{B}) measured in number of audio samples and the bit-depth defined in bits. The sample rate and the bit-depth are related with the quality of the sound signal while the buffer length defines the amount of samples to be processed at the same time (in recording and

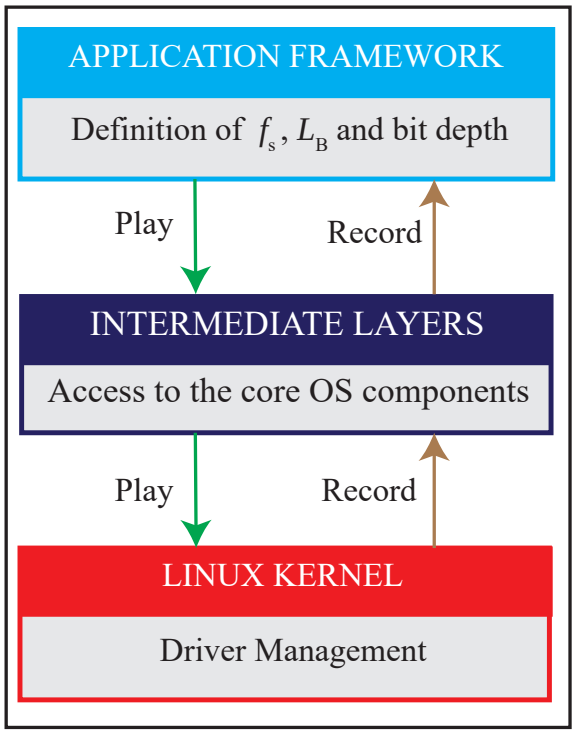


Figure 4.4. Simplified Android audio stack from [161].

playing). Regarding the bit-depth parameter, the highest value in Android devices corresponds to the floating point data format, but according to [165], it presents certain disadvantages. For the sake of simplicity, all the tests performed along this chapter will be carried out with a bit-depth of 16 bits.

Once the specific value of these three features is established, a chunk of L_B samples (also called audio frame or audio buffer) goes through the different layers of Figure 4.4 to be reproduced or recorded at the corresponding f_s. Therefore, T_{DP} is strongly related to f_s and L_B, specially with L_B where a bigger L_B leads to a higher processing cost in each layer, increasing the value of T_{DP}. In addition, when a Bluetooth connection is used, T_{DP} is increased due to the additional processing that the device must perform to adapt the audio signal to the Bluetooth link (such as coding [74, 166]).

In summary, T_{DP} depends on the performance of the Android stack shown in Figure 4.4, but it can be slightly controlled through f_{s} and L_{B}.

Bluetooth Link Latency

The Bluetooth link latency, T_{BL}, is the elapsed time between an audio buffer leaving the mobile device and the same buffer reaching the wireless speaker, that is, it is the radio propagation delay. Since the distance between the device and the speaker is within a range of a few meters, the time due to the propagation of radio waves can be considered negligible compared to the rest of factors contributing to T_{AL} in Figure 4.3, that is $T_{\mathrm{BL}} = 0$.

Speaker Processing Latency

As it is shown in Figure 4.3, the acoustic node can be composed by a wireless speaker. When a Bluetooth connection is used, some basic operations such as data transfer or decoding [74] must be carried out. Moreover, additional advanced audio processing [167] can be used to enhance the quality of the reproduced sound, leading to a wide range of T_{SP} values, depending on the model and manufacturer of the wireless speaker.

At the present, most of the Bluetooth speakers are dummy devices capable of receiving information to reproduce it, but unable to send back important control information, such as the L_{B} used or the number of underruns [168], which can produce annoying audio glitches due to an unstable reproduction. If the connection between the device and the loudspeaker in Figure 4.3 is wired, the audio received by the speaker is an analogue signal and it can be directly reproduced. Nevertheless, even in this case there can be small differences depending on the model and the manufacturer of the loudspeaker.

Acoustic Latency

Finally, we denote by T_{A} the time needed by the sound wave to travel from the loudspeaker to the built-in microphone of the mobile device, that is, the acoustic propagation time. It can be estimated through next expression:

$$T_{\mathrm{A}} = \frac{D_{\mathrm{ms}}(m)}{V(m/s)}\,, \tag{4.2}$$

where $D_{\mathrm{ms}}(m)$ is the distance between the mobile device and the wireless

speaker expressed in meters and $V(m/s)$ is the sound speed of the medium of propagation, which for the air is $V(m/s) = 343$ [169]. It must be noted that T_{A} is known once the distance ($D_{\mathrm{ms}}(m)$) is known, as will be considered in the experiments of Section 4.2.3.

Final Relationship

According to the previous definitions, the audio latency T_{AL} of the acoustic node shown in Figure 4.3 can be expressed as:

$$T_{\mathrm{AL}} = T_{\mathrm{DP}} + T_{\mathrm{SP}} + T_{\mathrm{A}}\,, \tag{4.3}$$

where $T_{\mathrm{DP}} = T_{\mathrm{OLD}} + T_{\mathrm{IL}}$, being all the delays measured in seconds.

4.2.2 Audio Latency Estimation

The audio latency, T_{AL}, is estimated as the maximum value of the cross-correlation function between the audio signal reproduced by the speaker and the audio signal recorded by the microphone of the node of Figure 4.3 [170]. Denoting by $x(n)$ as the signal emitted by the speaker and $y(n)$ as the signal captured by the microphone, the cross-correlation is defined as:

$$c(n) = \mathrm{E}\left[x(m+n)y(m)\right] = \sum_{m=-\infty}^{\infty} x(m+n)y(m)\,, \tag{4.4}$$

where $c(n)$ is the result of the cross-correlation. In addition, when the signal to reproduce is the Maximum-Length Sequence (MLS) [171] or the sine sweep (specifically a logarithmic sweep) [172], the result of the cross-correlation is equal to the RIR measurement [25]. This parameter provides information about shape of the system between microphone and speaker and it can be used to perform different applications, such as a CrossTalk canceller [173]. For this reason, these signals are proposed to estimate T_{AL}.

Since the cross-correlation function involves a high computational cost, the Fast Fourier Transform (FFT) is used in order to compute the cross-correlation efficiently. This approach is expressed as [174]:

$$c(n) = \mathrm{FFT}^{-1}\left(Y(k)F(k)\right)\ ,\ k = 0, 1, \ldots, N_{\mathrm{FFT}} - 1\,, \tag{4.5}$$

where $Y(k)$ is the k^{th} frequency bin of the FFT of the recorded signal, N_{FFT} is the closest power of two of $L_{\text{x}} + L_{\text{y}} - 1$ with L_{x} and L_{y} as the length of the reproduced and recorded signal respectively and FFT^{-1} as the inverse FFT. Regarding $F(k)$, it depends on the reproduced signal to use. In case of the MLS, $F(k) = X^*(k)$ where $X(k)$ is the k^{th} frequency bin of the FFT of the MLS signal and $(\cdot)^*$ means conjugated. When the sweep signal is used, $F(k)$ is the k^{th} frequency bin of the FFT of the $f(n)$ signal with $f(n)$ as the reversed time signal of the sweep signal using and envelope of -6 dB per octave, as [172] explains.

As stated above, the audio latency can be obtained through the position of the maximum of $c(n)$, as the following expression shows [170]:

$$p_{\text{AL}} = \arg\max_n c(n)\,, \tag{4.6}$$

where p_{AL} is T_{AL} expressed in samples, that is:

$$T_{\text{AL}} = \frac{1}{f_{\text{s}}} p_{\text{AL}}\,. \tag{4.7}$$

In this dissertation, a logarithmic sweep signal is considered since it provides an estimate better than that of the MLS signal [175]. The Figure 4.5 shows the RIRs estimated using the two signals, sweep and MLS. It can be appreciated that the estimated value of T_{AL} is identical for both signals since their respective maximum values (p_{AL}) are located at the same sample. However, it can also be noticed how the RIR estimated with the MLS signal is noisier than that estimated with the logarithmic sweep. In Section 4.5, the RIR measure is needed to perform the corresponding audio applications. For this reason, the sweep signal is used in this dissertation to estimate the value of T_{AL}.

4.2.3 Experimental study

As explained in Section 4.2.1, T_{AL} highly depends on the mobile device and the speaker used in Figure 4.3. Therefore, an experimental study of the behavior of T_{AL} is carried out in this section in order to analyze T_{AL} estimates when Android mobile devices and wireless speakers are used. For this purpose, different mobile devices and different speakers have been used and placed at different distances from each other. The mobile devices

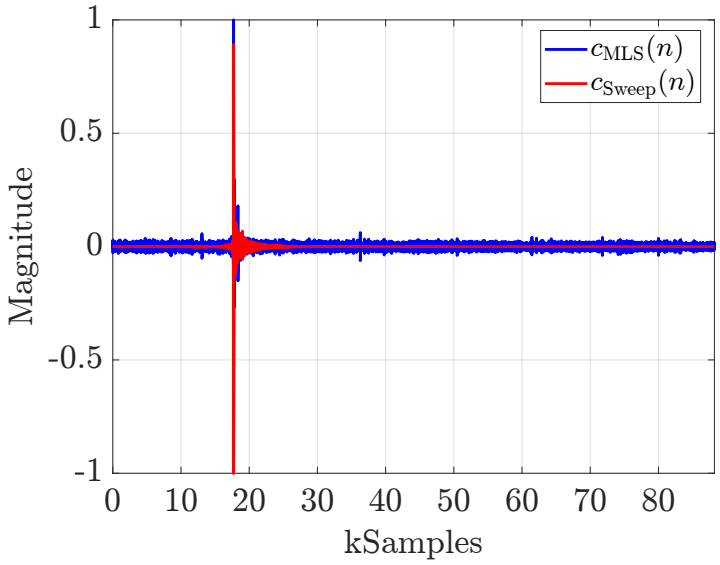


Figure 4.5. Estimated RIR by means of the MLS (blue) and the logarithmic sweep (red) signals.

and speakers used in the experiment are shown in Table 4.1 where each mobile device is combined with each of the speakers. Additionally, the Android version of the mobile device as well as the Bluetooth version of the loudspeakers are also shown. It should be highlighted that mobile devices are sorted such that the device with highest computational capability is located on the top row and the device with lowest computational capability on the bottom row. In addition, each of these combinations are evaluated at difference distances, $D_{\text{ms}} \in \{0.1, 0.2, 0.5, 1, 2, 5\}$ meters.

The different setups have been evaluated in the same room and the same day. In addition, each mobile device is located opposite to its corresponding speaker and at one meter approximately over the ground, as Figure 4.6 shows.

Before the sweep signal is reproduced and recorded in order to estimate T_{AL}, a previous stage must be performed. As [163] explains, the first time that an audio signal is reproduced or recorded, a certain amount of time is required in order to warm up the hardware. This stage is based on recording and playing a specific amount of zero samples (samples whose value is 0)

Mobile Device	Processor	Android Version
Samsung Galaxy S3	Snapdragon 820	8.0
HTC Nexus 9	Nvidia Tegra K1	7.0
Motorola Moto G5 Plus	Snapdragon 625	8.1
Samsung Galaxy Tab 3	Intel Atom Z2560	4.4.2

(a)

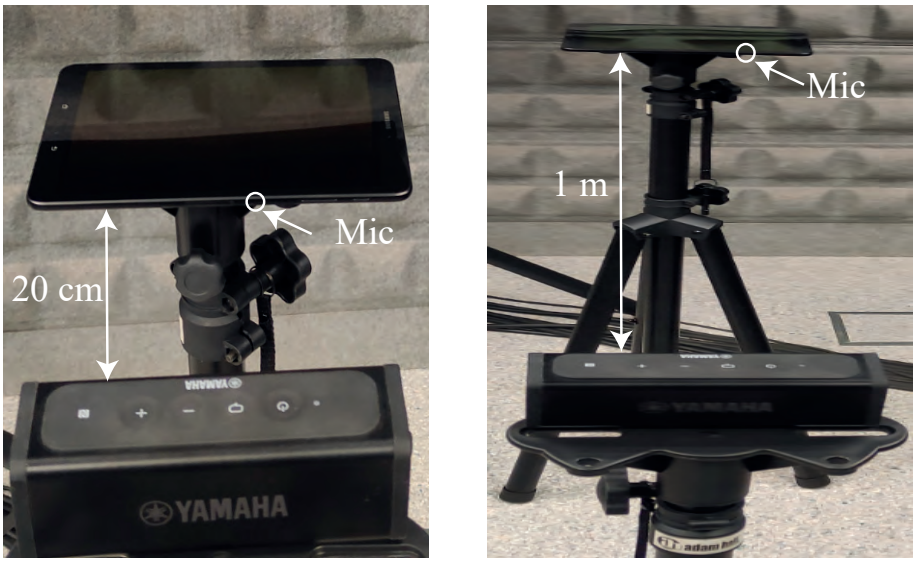
Speaker	Bluetooth Version
Yamaha NX P-100	2.1
Sony SRS-X3	2.1
JBL Flip 2	3.0
JBL Charge 4	4.2

(b)

Table 4.1. Description of the different (a) mobile devices and (b) speakers used for implementing the acoustic nodes.

that will be discarded since they are used only to warm up the hardware of the elements that form the acoustic node. After some experiments, the loudspeaker will be emitting zero samples during 3 seconds, thus, an amount of $L_{\mathrm{W}} = 3f_{\mathrm{s}}$ zero samples will be emitted before the sweep signal.

As stated in Section 4.2.1, reproduction and recording can be controlled through the parameters f_{s} (sample rate) and L_{B} (buffer length). In this experimental study $f_{\mathrm{s}} = 44100$ Hz for all the mobile devices and speakers. However, L_{B} is a parameter that depends on the mobile device, as Section 4.2.1 stated. The Android OS provides information about the minimum buffer size to use on the "Application Framework" Layer (see Figure 4.4) based on the mobile device properties. This buffer size will be denoted as $L_{\mathrm{B,ref}}$ in this dissertation. $L_{\mathrm{B,ref}}$ depends on the Bluetooth connection used to reproduce audio, that means that the Android OS also considers the type of audio connection to obtain $L_{\mathrm{B,ref}}$. The Table 4.2 shows $L_{\mathrm{B,ref}}$ in audio samples when the Bluetooth connection is considered for each mobile device used. These values are obtained without considering



(a) $D_{\text{ms}} = 20$ cm. **(b)** $D_{\text{ms}} = 1$ m.

Figure 4.6. Scenario of the experimental study for a distance between mobile device and loudspeaker of (a) 20 cm and (b) 1 m.

the speaker model since the Android OS considers the type of connection, but it does not consider the specific speaker to estimate $L_{\text{B,ref}}$. In Appendix C a deep study about the optimal L_{B} based on $L_{\text{B,ref}}$ is done. That study concludes that the optimal buffer size considering this scenario is given for $L_{\text{B}} = \frac{L_{\text{B,ref}}}{2}$.

Mobile Device	$L_{\text{B,ref}}$
Samsung Galaxy S3	10944
Samsung Galaxy Tab 3	10240
Motorola Moto G5 Plus	10800
HTC Nexus 9	12672

Table 4.2. Bluetooth $L_{\text{B,ref}}$ (in audio samples).

Once the warm-up stage has been introduced, and the sample rate and

buffer length have been established for the whole experiment, the methodology used to estimate T_{AL} for each device-loudspeaker pair and distance, is summarized in the next steps:

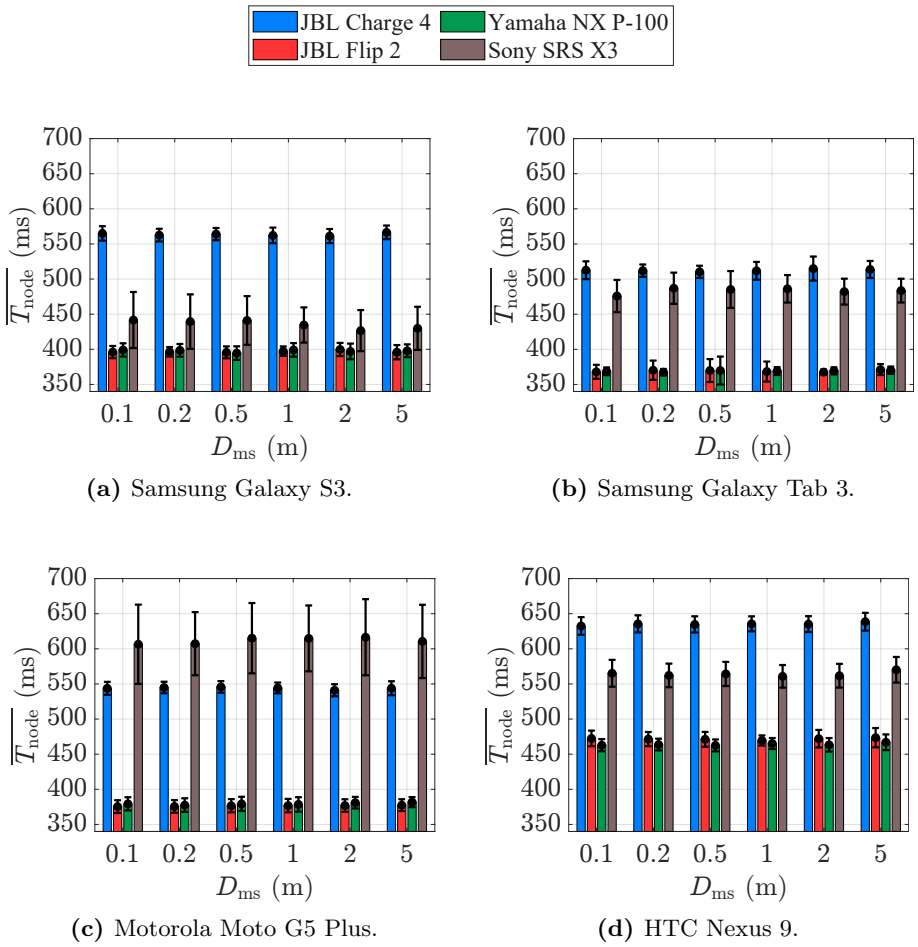
1. Choose a combination from Table 4.1 and select a distance, D_{ms}.
2. Start the warm-up process reproducing and recording $L_{\mathrm{W}} = 3f_{\mathrm{s}}$ zero samples.
3. Reproduce and record the sweep signal. For all the experiments a sweep signal of $L_{\mathrm{S}} = 2f_{\mathrm{s}}$ samples has been used.
4. Estimate the RIR through (4.5).
5. Estimate T_{AL} through (4.7).
6. Wait 3 seconds to release resources in the Android OS to avoid interfering with the next test.
7. Repeat steps 2 to 6 and record the measurements 50 times for the same pair and the same distance.

The previous procedure allows to estimate T_{AL} in terms of the distance and the device-loudspeaker pair. However, since D_{ms} is known, T_{A} can be estimated by using (4.2) and extracted from (4.3), that is:

$$T_{\mathrm{node}} = T_{\mathrm{AL}} - T_{\mathrm{A}} = T_{\mathrm{DP}} + T_{\mathrm{SP}}\,, \tag{4.8}$$

where T_{node} is the latency caused exclusively by the device-loudspeaker pair. This metric is illustrated in Figure 4.7 for a specific mobile device, considering both all the distances and speakers shown in Table 4.1. In particular, Figure 4.7 shows the average value of T_{node} ($\overline{T_{\mathrm{node}}}$) after 50 measurements depending on the distance, D_{ms}. In addition, the standard deviation of the T_{node} measurements ($\sigma_{T_{\mathrm{node}}}$) is also represented by using the error bars in the figure.

Figure 4.7 shows similar results for T_{AL}, for a specific mobile device and speaker regardless the distance, D_{ms}. Therefore, it can be concluded that low-quality communications, possibly caused by a large separation between the mobile device and the speaker, do not significantly affect on T_{node} (and consequently on T_{AL}). This statement is confirmed by the values of the



(a) Samsung Galaxy S3.

(b) Samsung Galaxy Tab 3.

(c) Motorola Moto G5 Plus.

(d) HTC Nexus 9.

Figure 4.7. $\overline{T_{\text{node}}}$ expressed in ms for each device-loudspeaker pair using the Bluetooth link.

standard deviation showed in the error bars, since for a specific combination of mobile device and speaker $\sigma_{T_{\text{node}}}$ is almost identical for all the distances. Therefore, it can be assumed that T_{DP} and T_{SP} are the main contributors to the behavior of $\overline{T_{\text{node}}}$ seen in Figure 4.7.

Regarding the analysis of the latency due to the Android OS, T_{DP}, we focus on the results obtained by the same speaker, that is, we assume that

the speaker latency is constant and $\overline{T_{\text{node}}}$ depends only on T_{DP}. To this end, the bars with the same colors in the different sub-figures of Figure 4.7 are compared. When the $\overline{T_{\text{node}}}$ are compared, a significant difference can be appreciated between the figures, reaching a maximum difference of 150 ms in the worst case (see the measurements of the Samsung Galaxy Tab 3 and the Motorola Moto G5 Plus mobile devices for the Sony SRS X3 speaker). This behavior leads to assume that two nodes composed by identical speakers and different mobile devices can be desynchronized up to 150 ms between them. This difference is explained by the fact that the Android OS implements the audio reproduction and recording processes in different ways depending on the mobile manufacturer. The main difference is caused because the processing in each layer of the stack (see Figure 4.4) is different in each mobile device. In addition, as Table 4.2 shows, each mobile device uses a different buffer length, L_{B}, that can cause a larger delay. Moreover, a different $\sigma_{T_{\text{node}}}$ can be appreciated for the different combinations of mobile devices when the same speaker is used. Although in most of the cases the difference is not significant, in Figure 4.7c the Sony SRS X3 speaker presents a significantly higher $\sigma_{T_{\text{node}}}$ than in the rest of the figures.

As a final remark, it can be appreciated from the figures that the Samsung Galaxy Tab 3 device presents the lowest T_{node} in average for all the speakers and distances. Surprisingly, the oldest mobile device featuring the oldest software and the lowest computational capacity is the fastest device. This behavior can be explained because an older Android version can use less processing at each layer of the stack.

Regarding now the analysis of the latency due to the speaker, T_{SP}, we focus on one of the four sub-figures in Figure 4.7. Then, we assume that the device latency T_{DP} is constant and $\overline{T_{\text{node}}}$ depends only on T_{SP}. In this case a maximum difference of 200 ms can be appreciated in the worst case (see the measurements of the Motorola Moto G5 Plus mobile device). Therefore, when using different speakers with identical mobile devices, the two acoustic nodes can be desynchronized up to 200 ms. This big difference in the values of T_{SP} can be explained by differences on the hardware and on the processes run by each speaker, such as different decoding algorithms or different processing to enhance the signal quality. Even if the same decoding method is used in all the speakers (the basic one is called SBC, Low Complexity Subband Codec [74]), the algorithm can be faster or slower leading to different processing times. In addition, similar to the mobile

devices, the speakers can use different buffer lengths, L_{B}, causing a higher or lower T_{node} (and consequently a higher or lower T_{AL}) according to the L_{B}. As seen in Figure 4.7, the speakers that offer the lower T_{AL} are Yamaha NX P-100 and JBL Flip 2, whereas the JBL Charge 4 exhibits the longest T_{node} for three of the four mobile devices.

Finally, a remark should be made about the fact that T_{node} depends on both T_{DP} and T_{SP}. This behavior can be explained through the next example, where an ASN of two acoustic nodes with the same mobile device and different speakers in each node is considered. When the Samsung Galaxy S3 device and the JBL Charge 4 and Sony SRS X3 speakers are used, the difference of T_{node} between the different nodes is around 100 ms (see Figure 4.7a). However, when the mobile device is changed to the Motorola Moto G5 Plus device, this difference is around 50 ms (see Figure 4.7c). This behavior demonstrates that T_{node} does not depends on T_{DP} and T_{SP} considered separately, since T_{node} (and consequently T_{AL}) depends on how the mobile device and the speaker collaborate between them.

We would like to analyze this mutual dependence between T_{DP} and T_{SP} when the mobile device and the speaker are connected through a wireless Bluetooth link. For this purpose, we have studied an alternative wired analog connection between the mobile device and the speaker. For this experiment, the same combinations shown in Table 4.1 and the same methodology than in the wireless connection has been used, but assuming a single distance of $D_{\mathrm{ms}} = 50$ cm, since the distance was not a relevant factor once T_{A} has been subtracted from T_{AL}. The same buffer length, L_{B}, than for the Bluetooth connection was used (see Table 4.2). Results are shown in Figure 4.8, where in this case, $\overline{T_{\mathrm{node}}}$ is represented for different mobile devices instead than of the distance as in Figure 4.7.

Generally speaking, the values of $\overline{T_{\mathrm{node}}}$ are lower than for the Bluetooth case, specially in the HTC Nexus 9 mobile device where it achieves the lowest $\overline{T_{\mathrm{node}}}$ regardless the used speaker, even providing that in Figure 4.7 the HTC Nexus 9 presented the highest $\overline{T_{\mathrm{node}}}$ in most of the cases. The JBL Flip 2, Sony SRS X3 and Yamaha NX P-100 show similar $\overline{T_{\mathrm{node}}}$ values for each mobile device, while the JBL Charge 4 speaker produces a $\overline{T_{\mathrm{node}}}$ around 40 ms higher than the other three for any mobile device. This additional latency could be caused by an additional processing of the speaker, suggesting that the speakers run some kind of processing independently of the connection, wired or wireless. Furthermore, the variability

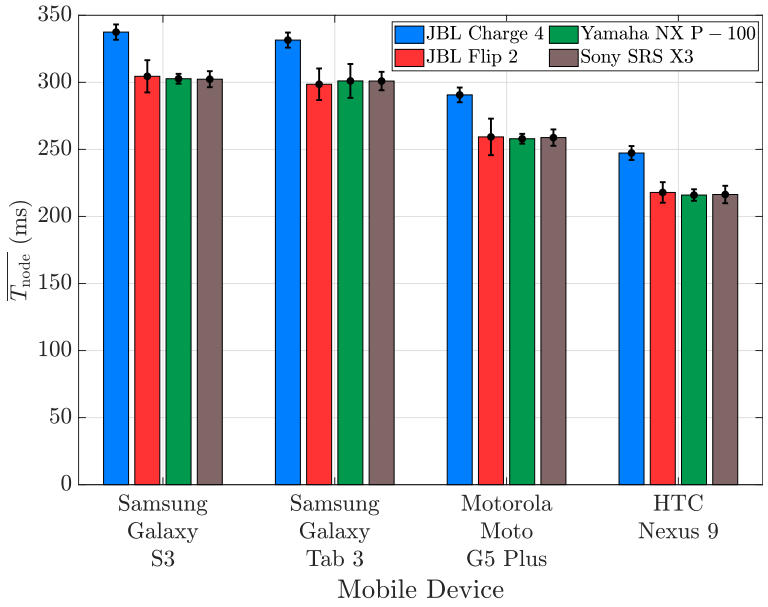


Figure 4.8. $\overline{T_{\text{node}}}$ expressed in ms for each device-loudspeaker pair using the wired link.

of the measurements obtained for a wired connection is smaller than that for the Bluetooth. In this regard, the typical deviation is represented in Figure 4.8 by the error bars. In most of the cases this behavior is slightly significant, but the dispersion of the Sony SRS X3 speaker greatly improves with respect to their ranges shown in Figure 4.7. Therefore, when an analog wired connection is used, the T_{node} (and consequently T_{AL}) measurements present a higher stability and lower values since the mobile devices and the speakers perform less processing as well as the audio transmission between the mobile device and the speaker is faster.

4.2.4 Conclusions

In this section, the latency T_{AL} has been introduced in (4.1) and Figure 4.3. The parameter T_{AL} models the time elapsed between the Android device starts the process of reproducing a sound through a wireless speaker connected via Bluetooth and the time the sound is recorded by the microphone of the device. The latency T_{AL} is made up of the addition of four terms,

although only three of them have been considered relevant: the time spent by the Android device to send the audio to the speaker, T_{DP}, the time spent by the speaker to decode and reproduce the digital audio, T_{SP}, and the time due to the physical propagation of the sound from the speaker to the microphone of the device.

An experimental study has been done in order to provide a full analysis of the behavior of T_{AL}. The results have proven that T_{AL} is highly dependent on the latencies T_{DP} and T_{SP}, obtaining different values for different combinations of mobile device and speaker, that are illustrated in Figure 4.7 shows. For this reason, a new latency term, T_{node}, formed by the addition of these two terms have been defined, in order to specifically characterize the latency of the acoustic nodes. In addition, it has been shown in Figure 4.7 that T_{node} depends on both T_{DP} and T_{SP}. Therefore, T_{node} is not only affected by the individual characteristics of the mobile device and the speaker, but also by the relationship between them regarding the audio processing required to make they work in a coordinated way.

As a final step, a similar study on the latency of the audio reproduction process has been carried out, where the Bluetooth connection has been replaced by a wired connection. Results in Figure 4.8 show a better performance than in the case of Bluetooth connection. Lower values of T_{node} have been obtained for every device-speaker combination, together with lower values of their standard deviation, leading to a higher reliability in the measurements.

Finally, it can be stated that a Bluetooth connection presents a T_{node} values ranging from 350 ms to 650 ms with standard deviations from a few ms till more than 50 ms. Moreover, given the dependence on the hardware and software of both the device and speaker, the T_{node} values cannot be controlled, only observed. Therefore, a wireless ASN comprised of acoustic nodes with Bluetooth connections must estimate the T_{node} latency for every device-speaker pair prior to carry out audio applications where synchronism between nodes is critical.

4.3 Clock Synchronization

4.3.1 Background

As it is stated in Section 4.1, in order to carry out distributed audio application on different acoustic nodes of the ASN, the synchronization problem must be addressed. For the sake of clarity, the study on the ASN synchronization will be split in two parts. Along this section, we will focus on how to achieve the clock synchronization of the Android devices that form the ASN. For this purpose, a novel approach for wireless networks and Android devices is proposed. In addition, the proposed method is compared with previously proposed methods throughout some experiments. Finally, the accuracy of the achieved synchronism between the devices will be evaluated when a two-node ASN is used to reproduce a sound simultaneously by both loudspeakers.

The problem of clock synchronization is a well-known issue that affects all devices, regardless whether devices are based on Android or not, or even if all the devices are the same model of the same manufacturer, because their clocks depend on the hardware of the device and, in general, each device presents a different timestamp. As a result, in order to perform simultaneously a task by all the devices inside an ASN, a synchronization between them is required otherwise the task will probably not be performed at the same time in each device. This behavior can be appreciated in Figure 4.9, where a particular task has been scheduled to start on time T_{I} by the clock of the device labeled as "Device A". Since "Device A" and "Device B" present different clocks, the task is performed at different times in each device. More specifically, "Device A" performs the task at time T_{A} (that corresponds to T_{I}), while "Device B" performs it at time T_{B}. This difference between the reference values of the clocks is usually known as clock offset [176], denoted hereafter as T_{O}.

Consequently, in order to be able to run synchronized operations in different devices, it is essential to compensate the time differences [177, 178]. Several methods have been implemented to address this problem, such as [179] where an accurate method for middle-large networks is implemented, and [180] where the implemented method is used in two Android devices in order to measure the stability of the proposed method on the Android OS. These methods are based on the master-slave architecture, being the slave (or slaves) the device that requests information to the mas-

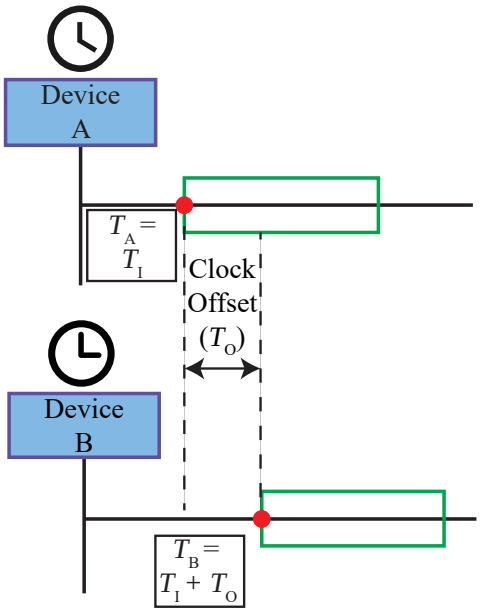


Figure 4.9. Example of clock offset, T_O, between two devices when a scheduled task is carried out in the ASN.

ter in order to compensate its clock time according to the clock time of the master. To this end, master and slave exchange messages containing their time information or timestamps. These timestamps are used to estimate T_O and to compensate the different clock times in the devices [181, 182, 183]. Following the scenario shown in Figure 4.9, "Device B" will be able to compensate its clock according to the clock of "Device A" once T_O has been estimated, causing the task to be performed simultaneously in both devices. As said before, the devices must communicate and exchange messages between them through a communication link, such as the Wi-Fi link.

One of the most widely used methods firsts proposed several decades ago is the network time protocol (NTP) method [184]. Apart from exchanging their timestamps between master and slave, it assumes that master and slave belong to different networks. For this reason, the slave device requires an Internet connection to communicate with the master. Despite the solid results that offers the NTP method, it can provide inaccurate results when networks are heavily loaded [178], or when the OS of the devices are not

specifically designed to deal with time-controlled applications [185], such as Android devices. Even presenting inaccuracies, NTP is still widely used since it offers synchronization of the order of tens of milliseconds [186], which is suitable in applications where a tight clock synchronization is not needed. A native procedure is available in Android OS based on NTP to synchronize its own clock [187].

New methods have been implemented to improve the accuracy of the T_O estimation, such as the method shown in [188], based on distributed algorithms, and [189, 190] where different methods to estimate T_O are explained considering the challenges of a SN (specially a WSN). In [191], the proposed method is used to synchronize the data collected by the different sensors of the WSN and fuse them. Other methods are presented in [192, 193] as well, whose objective is to increase the reliability and robustness of the offset estimation while keeping the algorithm simple. Despite all these methods from the literature, the IEEE organization standardized the precision time protocol (PTP) method, which is detailed in the IEEE 1588 standard [194, 195]. However, in this dissertation, we propose a new method specially defined for Android devices connected throughout wireless link.

4.3.2 Proposed Synchronization Time Protocol (STP)

The proposed method is based on the methods described in [193, 194] because they present a simple design and a great robustness of the estimation of T_O. Therefore, the proposed method will exchange the timestamps between two devices in order to obtain T_O. This method is implemented over the TCP protocol and takes advantage of the way the protocol TCP must establish the connection between the devices. The establishment of the TCP connection is known as the 3-Way Handshake process [196] and is based on the server-client architecture, where the client requests the connection and the server receives that request. The establishment of a TCP connection is shown in Figure 4.10 where three different messages are exchanged, as explained in the following steps:

1. The Client device sends a request of connection with a sequence number (Synchronize Sequence Number, SYNC).
2. The Server device answers the request by providing its own sequence number (SYNC + ACK).

3. Finally, the Client device sends an ACK to let the Server device know that the connection has been established.

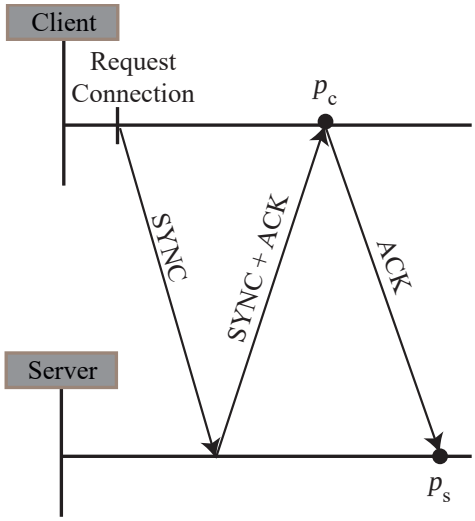


Figure 4.10. The 3-Way Handshake process.

In this process, two moments are very significant for the STP method, shown as p_c and p_s in Figure 4.10. These two moments refer to the timestamps when the Client and Server devices become aware of the connection, respectively. While the Client is aware after the SYNC+ACK message, the Server is aware after the 3-Way Handshake process is completed. Therefore, each device becomes aware of the TCP connection at different times. This difference is caused by the communication protocol (in this case Wi-Fi), that is, the time difference between p_c and p_s is the time that the ACK frame takes to arrive from the Client device to the Server device.

As explained above, the STP method takes timestamps p_c and p_s when each device is aware of the connection, this is, two timestamps are taken in p_c and p_s respectively. Assuming that the delay of the ACK frame is always the same, Figure 4.11 shows the time sequence of the STP method to estimate the clock offset, T_O. For this purpose, four different roles are defined:

- The "GrandMaster" and "GrandSlave" roles. They indicate which

device has the reference clock time (GrandMaster) and which one must compensate its own clock (GrandSlave).

- The "Server" and "Client" roles. They indicate which device requests the TCP connection and which one listens to the TCP connection.

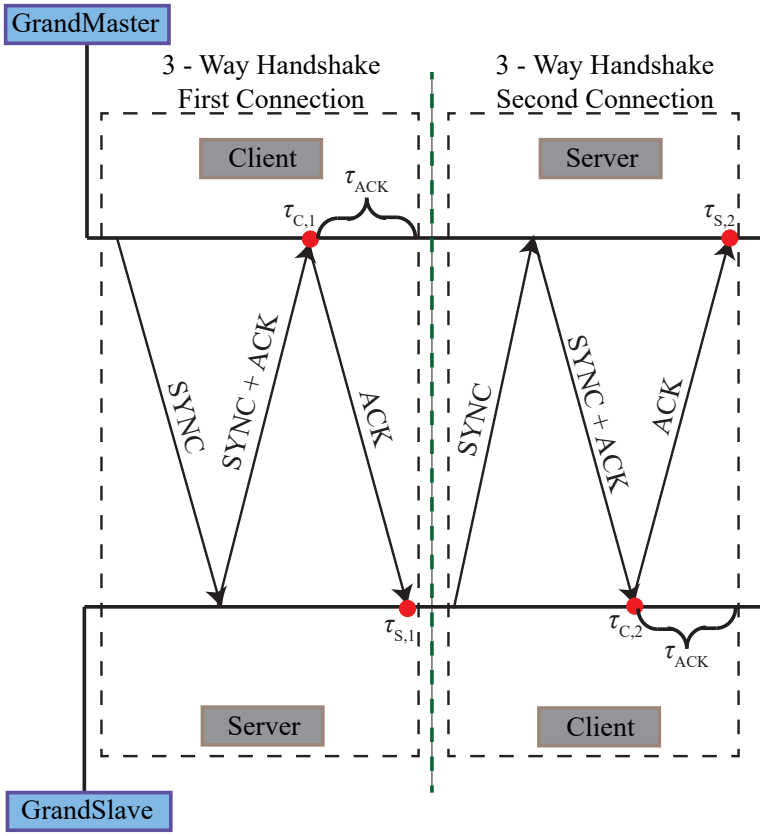


Figure 4.11. STP method to estimate the clock offset, T_O.

Through these roles, the devices perform the STP method as follows in order to obtain the time difference between their clocks, T_O:

1. The GrandMaster (acting as a Client) requests a connection to the GrandSlave (SYNC message).

2. After receiving SYNC+ACK message, the GrandMaster is aware of the connection. Therefore it sends the ACK message to the Grand-Slave and it takes a timestamp, $\tau_{C,1}$.

3. After receiving the ACK message, the GrandSlave is aware of the connection and it takes a timestamp, $\tau_{S,1}$.

4. First connection is closed and the connection roles (Server and Client) are switched.

5. The GrandSlave (acting as a Client) requests a connection to the GrandMaster (SYNC message).

6. The same steps than steps 2 and 3 are repeated, but the timestamps taken are $\tau_{S,2}$ for the GrandMaster and $\tau_{C,2}$ for the GrandSlave.

7. Second connection is closed, finishing the STP method.

As Figure 4.11 shows, through this method four different timestamps are taken such that:

$$\tau_{S,1} - \tau_{C,1} = \tau_{ACK} + \tau_O \tag{4.9a}$$

$$\tau_{S,2} - \tau_{C,2} = \tau_{ACK} - \tau_O \,, \tag{4.9b}$$

where τ_{ACK} is the time the last frame takes to be transmitted from the Client to the Server in any connection, as shown in Figure 4.11, and τ_O is the current estimate of the clock offset between devices. Note that τ_O can be positive or negative according to the clock of each device. In (4.9), τ_O shows a different sign because the connection roles (Server and Client) have been exchanged. This role switch causes that the GrandMaster and GrandSlave clock times are reversely subtracted in (4.9a) and (4.9b), and τ_O presents a different sign for each expression. On the other hand, τ_{ACK} is not affected by this switch since it refers to the time taken by the communication process between the Client and the Server, regardless of the roles of GrandMaster or GrandSlave. Assuming this scenario, the clock offset can be estimated by the difference between (4.9a) and (4.9b), such as:

$$\tau_O = \frac{(\tau_{S,1} - \tau_{C,1}) - (\tau_{S,2} - \tau_{C,2})}{2} \,. \tag{4.10}$$

The estimate of τ_O obtained by the STP method of Figure 4.11 is valid only for a limited time interval because the clock of each device can suffer a time shift, also known as the clock drift [197], deprecating the current estimate of τ_O. Therefore, the STP method must be performed periodically in order to overcome the drift of the clocks and obtain an accurate estimate of τ_O at any time. In addition, with the purpose of reducing as much as possible the effect of outliers (caused by network errors), all the estimations are averaged by using an Exponential Moving Average (EMA) [145], as:

$$T_O(n) = \xi\tau_O + (1-\xi)T_O(n-1)\,, \tag{4.11}$$

where n and $(n-1)$ are the current and previous iteration, respectively, $T_O(n)$ is the current estimate of the clock offset and ξ is the smoothing constant. This parameter lies within the range of $[0,1]$, causing a slow adaptation to the current value (τ_O) for low values of ξ and a fast adaptation for large values of ξ. To minimize the influence of network errors, a slow adaptation with a value of $\xi = 0.1$ is considered.

It is important to relate the STP method with the accuracy of the measurement of the timestamps when the devices use Android OS. In this case, the highest level of precision of the timestamps in Figure 4.11 is within the millisecond range [198], thus, the estimate of T_O will be also obtained within the millisecond range.

4.3.3 Node Task Synchronization (NTS) method

Once T_O has been estimated through the STP method, the time offset between the clocks of the two devices is known. This value, T_O, is used to change the clock of the GrandSlave device and thus schedule a task to be simultaneously performed by all the devices. However, the Android OS does not allow to change the clock time by third party applications (such as the used in this dissertation, see Appendix A). Therefore, it should not be possible to schedule a simultaneous task in multiple devices supporting Android OS. To overcome this drawback, an additional method denoted by Node Task Synchronization (NTS) method, is proposed and implemented in this section.

The NTS method defines two different clocks: 1) the global clock (GC) and 2) the local clock (LC). The time provided by the GC is shared by all the nodes of the network, that is, the GC time is known by all the

nodes. The time provided by the LC corresponds to the system clock of each acoustic node that cannot be modified by third-party applications. Considering the definition of T_{O} shown in Figure 4.9 that is estimated by (4.11), and assuming that device A is the GrandMaster whose clock is denoted by GC, the relationship between the clocks of both devices A and B can be expressed as

$$T_{\mathrm{LC}} = T_{\mathrm{GC}} + T_{\mathrm{O}}\,, \tag{4.12}$$

where T_{LC} is the timestamp provided by the device B (considered as LC) and T_{GC} is the timestamp provided by the device A (considered as GC). Therefore, using (4.12), any acoustic node of the network can manage a local and a global, o reference, timestamp, just estimating their local time offset through (4.11) with respect to the device that provides the GC time.

The proposed NTS method acts as shown in Figure 4.12. First of all, the "Master" and "Slave" roles have to be defined, where the first one indicates the node that defines the time to start the task and the second one the node that receives that time. These two roles should not be confused with the roles used in the Section 4.3.2 (GrandMaster and GrandSlave), since they are not used with the same purpose. In fact, the GrandMaster node can take the Master or the Slave role in the NTS method. Regarding the NTS method shown in Figure 4.12, the timestamps labeled as $T_{\mathrm{LC,M}}$ and $T_{\mathrm{LC,S}}$ indicate the timestamps taken by the LC of the Master and Slave devices respectively, while the timestamp labeled as $T_{\mathrm{GC}}^{\mathrm{st}}$ indicates the exact time the task must start in reference to the clock of the GrandMaster, GC.

As Figure 4.12 shows, the device with the Master role decides when the task must start and calculates the corresponding global time, $T_{\mathrm{GC}}^{\mathrm{st}}$, as:

$$T_{\mathrm{GC}}^{\mathrm{st}} = T_{\mathrm{LC,M}} - T_{\mathrm{O,M}} + \Delta_{\mathrm{M}}\,, \tag{4.13}$$

where $T_{\mathrm{LC,M}}$ is the current time of the Master device (taken with its LC), $T_{\mathrm{O,M}}$ is the clock offset of the Master, estimated through (4.11), and Δ_{M} is a parameter that can be set to any value by the Master, such as one second.

Once the Master device has estimated $T_{\mathrm{GC}}^{\mathrm{st}}$ through (4.13), it sends the global timestamp to the device with the Slave role and it waits till the time

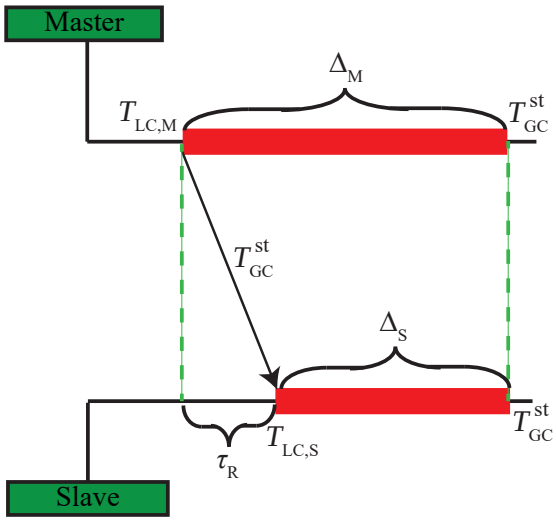


Figure 4.12. NTS method to synchronize any task (such as an audio task) on different devices.

T^{st}_{GC} is reached. Then, the Slave device receives T^{st}_{GC} and it uses the global timestamp to compute the waiting interval (Δ_S) shown in Figure 4.12 as:

$$\Delta_S = T^{st}_{GC} - (T_{LC,S} - T_{O,S}) \ , \tag{4.14}$$

where $T_{LC,S}$ is the time, taken by the LC of the Slave, when T^{st}_{GC} is received and $T_{O,S}$ is the clock offset of the Slave, estimated through (4.11).

After estimating Δ_S, the Slave waits till the time T^{st}_{GC} is reached. Once both sides (Master and Slave) reach their corresponding local timestamps, that corresponds to the global timestamp T^{st}_{GC} (4.12), they will be able to start the task simultaneously.

To provide a proper result, the NTS method must accomplish $\Delta_M > \tau_R$ in Figure 4.12, where τ_R is the time needed to transmit the global timestamp, T^{st}_{GC}. If Δ_M is lower than τ_R, the Master will start the task before the global timestamp reaches the Slave and will prevent the synchronization of both devices. Since τ_R is due to the Wi-Fi link, its range can be assumed to be a few milliseconds. As a result, setting Δ_M long enough, i.e., $\Delta_M \geq 1$ second, will ensure a correct performance of the NTS method.

4.3.4 Experimental analysis of the clock offset estimates

Once the NTS method has been introduced, it will be used to analyze the performance of the estimation of the clock offset, T_{O}. Additionally, in this section we will compare the estimate of T_{O} obtained through the proposed STP method, shown in Figure 4.11, with the estimate of T_{O} obtained through the NTP and the PTP methods, explained in Section 4.3.1. Both, the PTP and NTP methods have been also implemented in Android: the PTP method is based on the process explained in [199], while the NTP method is based on the algorithm provided by the Apache Commons Net library [200].

Each of the three methods will be used in combination with the NTS method, shown in Figure 4.12, in order to estimate the value of T_{O} between two mobile devices. For this experiment, the Samsung Galaxy S3 and HTC Nexus 9 mobile devices are used, although similar results have been obtained for the other devices.

The experiment has been carried out considering the Samsung Galaxy S3 device is the GrandMaster and the HTC Nexus 9 is the GrandSlave for any of the three methods. This experiment is based on obtaining the local timestamp (through the LC) of each device when both of them are synchronized, this is, when they reach $T^{\mathrm{st}}_{\mathrm{GC}}$ in Figure 4.12. Since the local time of the GrandMaster device is the global time (GC), the difference of both timestamps will provide the value of T_{O}, as (4.12) indicates. As a result, T_{O} is obtained as:

$$T_{\mathrm{O}} = T^{\mathrm{st}}_{\mathrm{LC,GS}} - T^{\mathrm{st}}_{\mathrm{LC,GM}}\,, \tag{4.15}$$

where $T^{\mathrm{st}}_{\mathrm{LC,GS}}$ is the exact time the task must start in reference to the LC of the GrandSlave and $T^{\mathrm{st}}_{\mathrm{LC,GM}}$ the exact time the task must start in reference to the LC of the GrandMaster (that is equal to $T^{\mathrm{st}}_{\mathrm{GC}}$).

This process is repeated 1000 times, being the Samsung Galaxy S3 device the Master in the NTS method the first 500 times, and the HTC Nexus 9 device the Master in the NTS method the last 500 times. This role interchange is to evaluate the correct behavior of the NTS method. Therefore, this experiment stores 1000 local timestamps of each device for each method, and through (4.15) 1000 values of T_{O} are estimated for each method.

Due to the wireless connection between the two devices used in this experiment, the groundtruth value of T_{O} cannot be obtained, as [201] shows, where the evaluation of its method is carried out through a wired link. Therefore, this study will be focused on analyzing the variability of the T_{O} measurement provided by each of the three methods (STP, PTP and NTP). In order to analyze the stability of the methods, the mean value of the 1000 values of T_{O} is subtracted from these values, such as:

$$\Delta T_{\mathrm{O}}(n) = T_{\mathrm{O}}(n) - \overline{T_{\mathrm{O}}}\,, \tag{4.16}$$

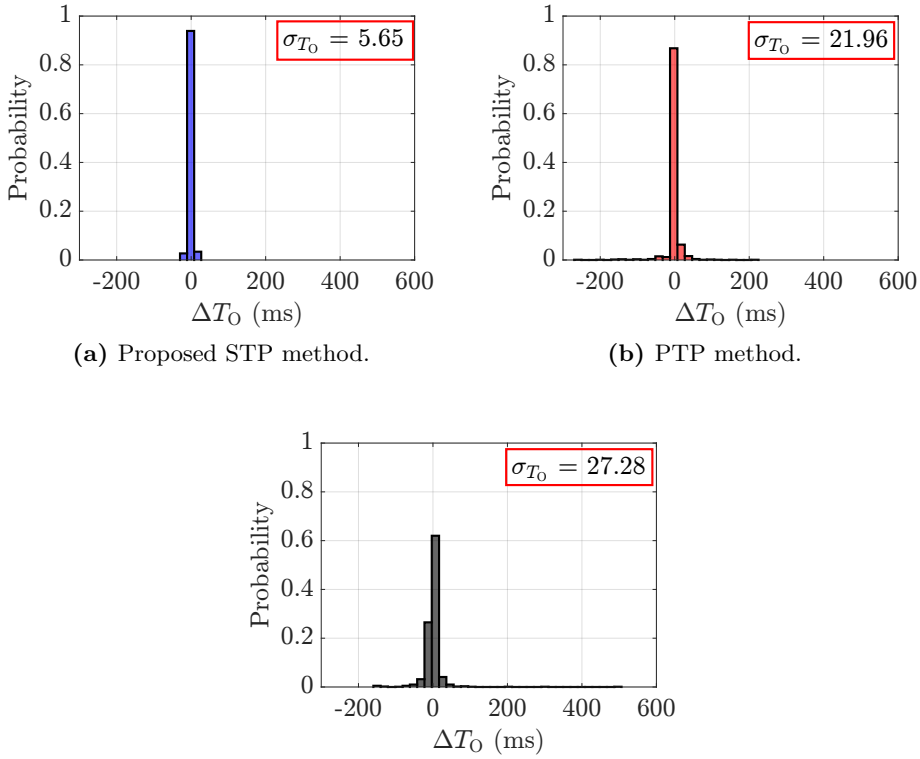
where $n = 1, \ldots, 1000$ and $\overline{T_{\mathrm{O}}}$ is the mean value of $T_{\mathrm{O}}(n)$ that is shown in Table 4.3 for each method in milliseconds.

Method	$\overline{T_{\mathrm{O}}}$ (ms)
STP	4859
PTP	4866
NTP	4878

Table 4.3. The mean value of T_{O} in ms for each method.

Figure 4.13 shows the histogram of the difference, $\Delta T_{\mathrm{O}}(n)$ in (4.16), for each method where the bars have a width of 20 ms. In addition the standard deviation of $T_{\mathrm{O}}(n)$ in ms ($\sigma_{T_{\mathrm{O}}}$) is included in the upper right side of the figures.

As Figure 4.13c shows, the NTP method presents the highest standard deviation of the three methods, meaning that T_{O} values estimated by the NTP method present significant differences (the maximum is 500 ms) with respect to the average value. Therefore, the NTP can be considered the less stable method since only 60% of its T_{O} values are included in the range of $\overline{T_{\mathrm{O}}} \pm 10$ ms. Regarding the PTP method, it shows a standard deviation smaller than that of the NTP method. The PTP method presents a maximum deviation around 200 ms, a standard deviation of $\sigma_{T_{\mathrm{O}}} = 21.96$ ms and 90% of its values is within the range of $\overline{T_{\mathrm{O}}} \pm 10$ ms. As a result, the PTP presents a more stable behavior than the NTP method. Regarding the STP method, it achieves the narrowest deviation of the three methods, with a maximum deviation of 30 ms with respect to its average, a standard deviation of $\sigma_{T_{\mathrm{O}}} = 5.65$ ms and 94% of its values are within the range of



(a) Proposed STP method.

(b) PTP method.

(c) NTP method.

Figure 4.13. Comparison between the three methods.

$\overline{T_O} \pm 10$ ms. Note that the proposed STP method presents a significant improvement on the maximum extent of the deviation of the estimate (30 ms) with respect to the widely used PTP method (200ms).

Summarizing, the NTP method presents the highest standard deviation (σ_{T_O}), being the least accurate method. Such behavior can be explained because the devices must communicate with a NTP server that is located outside of the local network. As a result, the communication with the NTP server is unpredictable, causing an unstable measurement of T_O. The PTP method overcomes the NTP method because the communications between the devices are carried out inside the local network. Finally, the STP method improves the results obtained by the PTP method mainly because

it averages the instantaneous T_O estimates through (4.11), and because the exchanged messages of this method do not require to go through the entire Android stack, as opposed to the PTP method. As a result, the proposed STP method has shown to be the most accurate method and it will be used to estimate T_O in the following.

4.3.5 Synchronization evaluation for a reproduction task

In this section, we propose to evaluate the performance of the STP method to synchronize audio tasks between the different acoustic nodes of an ASN. For this purpose, the NTS method of Figure 4.12 will be used for reproducing simultaneously the same signal from two devices. The difference between their respective audio latencies, defined in (4.3) as T_{AL}, will be used as a metric to evaluate the performance.

As it is explained in Section 4.2.2, the estimation of T_{AL} involves the recording and playing processes. In order to reduce as much as possible the influence of the electro-acoustic path (see Figure 4.3), an audio analyzer[1] is used, as shown in Figure 4.14. It can be seen how the analogue audio output of each mobile device is connected through a wire to the analogue inputs of the audio analyzer, labelled as channels.

Therefore, the audio is directly sent from the devices to the audio analyzer through a wired channel. This way, the acoustic propagation time, T_A, and the speaker delay, T_{SP}, are removed from T_{AL} in equation (4.3). In addition, since no recording is carried out by the devices, the input latency $T_{IL} = 0$, and the only delay is due to the data processing needed to reproduce the audio signals, that is, $T_{DP} = T_{OLD}$. Consequently, in our experiment the measured audio latency coincides with the latency caused by the mobile output data processing, $T_{AL} = T_{OLD}$.

Summarizing, the experiment consists in scheduling the reproduction task in both devices as shown in Figure 4.14, by applying the NTS method, and measuring the difference of their respective T_{AL} by means of the audio

[1]The audio analyzer is the R&S UPV Analyzer from Rohde & Schwarz. Specifications can be found in: https://scdn.rohde-schwarz.com/ur/pws/dl_downloads/dl_common_library/dl_brochures_and_datasheets/pdf_1/UPV_dat_sw_en_0758-1306-22_v0400.pdf

Figure 4.14. Experiment setup to evaluate the accuracy of the synchronization method.

analyzer. This difference is denoted as Δ_{T} and can be expressed as:

$$\Delta_{\mathrm{T}} = T_{\mathrm{AL},1} - T_{\mathrm{AL},2}\,, \tag{4.17}$$

where $T_{\mathrm{AL},1}$ and $T_{\mathrm{AL},2}$ are the audio latencies of the devices connected to the first and the second channel of the audio analyzer respectively.

The difference Δ_{T} range is within $[-\infty, \infty]$, where positive differences mean that the first device provides a bigger T_{AL} than the second one and negative differences means the opposite case, being $\Delta_{\mathrm{T}} = 0$ the case of perfect synchronization. The value Δ_{T} is directly measured using the audio analyzer with the help of its markers. Figure 4.15 shows the screen of the audio analyzer when the two input signals are captured and depicted versus time. It can be seen how the markers can provide precise information of the specific point where the markers are located. Moreover, the audio analyzer can also estimate the difference of the markers of the different channels. This difference is displayed in the upper-left side of the screen as ΔX for the difference of the horizontal axis, and ΔY for the difference of the vertical axis. In this case, ΔX is the temporal difference of the audio signals captured by each channel, that is, $\Delta\mathrm{X} = \Delta_{\mathrm{T}}$. Therefore, the audio analyzer estimates directly the synchronization error between both devices.

In order to reduce as much as possible the differences due to the physical components, same wires with the same length are used to connect the audio output (3.5 mm mini-jack port) of the mobile devices with the analogue

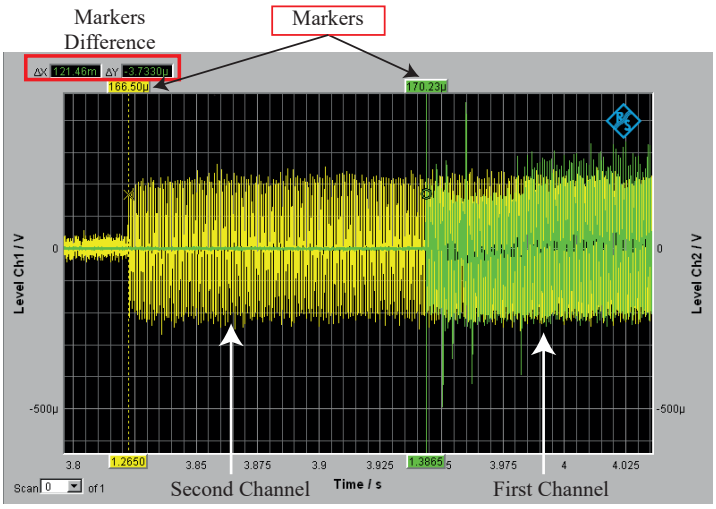


Figure 4.15. Example of a measure in the audio analyzer.

inputs of the audio analyzer. As it is explained in Section 4.2.3, the audio latency, T_{AL}, depends on the type of mobile device. Therefore, different pairs of devices have been used in the experiment. These pairs are shown in Table 4.4, where the first column corresponds to the devices of the first channel of the audio analyzer and the second column corresponds to the devices of the second one. It should be note that, the listed devices have different hardware and software, which causes differences in the Android stack, and consequently in T_{AL}, as Section 4.2.3 discussed.

Device CH1	Device CH2
Samsung Galaxy S3	Samsung Galaxy S3
Samsung Galaxy S3	HTC Nexus 9
Samsung Galaxy S3	Samsung Galaxy Tab 3
Samsung Galaxy Tab 3	HTC Nexus 9

Table 4.4. Pairs of devices used in the experiment.

In order to facilitate the precise measurement of Δ_{T}, a tone located at 1 kHz with a length of 2 seconds has been selected. Similarly to the study

of Section 4.2.3, the mobile devices must decide their sample rate, f_s, and its data buffer size, L_B. The sample rate is set to $f_s = 48000$ Hz, whereas the buffer size is set to $L_B = \frac{L_{B,ref}}{2}$, as it was derived from the analysis in Appendix C. The $L_{B,ref}$ values that correspond to a wired connection are shown in Table 4.5. It should be noted that these values are lower to the ones obtained for the Bluetooth connection (see Table 4.2).

Mobile Device	$L_{B,ref}$
Samsung Galaxy S3	4032
Samsung Galaxy Tab 3	4096
Motorola Moto G5 Plus	4080
HTC Nexus 9	2048

Table 4.5. Buffer size, $L_{B,ref}$, for wired connections (in audio samples).

The NTS method is used in order to schedule the reproduction of the tone on both mobile devices and thus measuring Δ_T. As it was described in Section 4.2.3, a warm-up stage must be performed before the reproduction of the tone signal in order to warm up the circuits and provide an accurate measurement of T_{AL}. Consequently, once the NTS has finished, the warm-up stage followed by a reproduction of the tone is carried out.

Assuming that the GrandMaster is the device connected to the first channel, the next steps are followed in order to measure the synchronization error between two Android devices performing audio reproduction tasks:

1. Select a combination of Table 4.4.
2. The GrandMaster acts as the Master in the NTS method of Figure 4.12 in order to schedule the reproduction task using $\Delta_M = 1$ second.
3. Once the NTS method has been completed, each device starts the reproduction process generating the corresponding single tone signal and previously performing the warm-up stage.
4. The audio analyzer samples both tones using a sample rate of 400 kHz (that corresponds to a temporal resolution of 2.5 us) and displays them.

5. Obtain the time when each signal starts by using the markers. In this case, the maximum of the first period of the tone is selected as the initial time.

6. By using the audio analyzer, the difference Δ_T is estimated, such that $\Delta_{\mathrm{T}} = \Delta\mathrm{X}$.

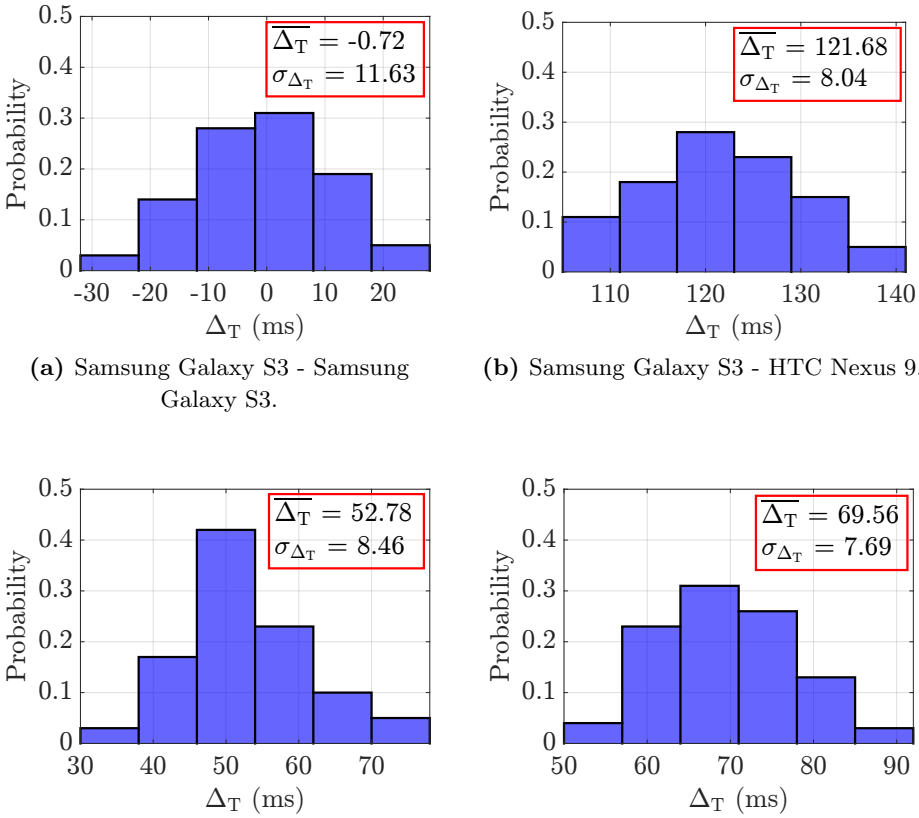
7. Repeat from step 2 this process 50 times.

8. Repeat all the previous steps (from step 2 to step 7) considering now the device of the second channel as the GrandMaster.

For the different pairs shown in Table 4.4, 100 measurements of Δ_{T} are performed, where the first 50 measurements are obtained assuming that the GrandMaster is the device connected to the first channel, while the other 50 measurements are performed assuming that the GrandMaster is the device connected to the second channel. The normalized histograms of Δ_{T} (in ms) are shown in Figure 4.16 where each sub-figure corresponds to a different device combination and each bar expands 10 ms. In addition, the standard deviation ($\sigma_{\Delta_{\mathrm{T}}}$) and the mean ($\overline{\Delta_{\mathrm{T}}}$) of the 100 measurements of Δ_{T} are included in the upper right side of the figures.

Figure 4.16a illustrates the measures obtained when two identical devices Samsung Galaxy S3 are used in both channels of the audio analyzer. The results produce a dispersion of Δ_{T} around ± 25 ms regarding the mean value ($\overline{\Delta_{\mathrm{T}}}$). It is the only figure whose mean value is almost 0 ms, which is a reasonable result since both devices are identical in hardware and software. Regarding the range of values of Δ_{T} in Figure 4.16a, it is due to slightly differences between the Android OS of each mobile device. Since the Android OS is focused on multiple tasks for the proper operation of the mobile device others than the reproduction task, the starting time of the reproduction task can significantly oscillate between two different measurements. In this regard, the number of tasks implemented in each mobile can be different (even with same hardware and software), each Android OS can run the reproduction task at a slightly different time than expected, causing the reproduction task to be delayed differently on each mobile device and resulting in the scattering shown in Figure 4.16a.

Figures 4.16b to 4.16d represent the results when different mobile devices are used in each channel of the audio analyzer. Similarly to the results shown in Figure 4.16a, all these measures present a similar dispersion in



(a) Samsung Galaxy S3 - Samsung Galaxy S3.

(b) Samsung Galaxy S3 - HTC Nexus 9.

(c) Samsung Galaxy S3 - Samsung Galaxy Tab 3.

(d) Samsung Galaxy Tab 3 - HTC Nexus 9.

Figure 4.16. Results of the experimental study.

Δ_T, caused because of the same reasons explained above. However, their mean value $\overline{\Delta_T}$ is different. This means that, the reproduction task produces a different T_{AL} in each device (as it was concluded in Section 4.2.3), resulting in a different $\overline{\Delta_T}$ depending on the combination of devices. This is caused by: 1) different software, that is, different Android version in each device, and 2) different hardware components. In addition, Figures 4.16b and 4.16d that use the HTC Nexus 9 present a higher mean value of Δ_T because its stack (see Figure 4.4) is implemented differently on the lower layers.

Finally, from the previous combinations, the next relationship can be obtained assuming that $T_{\mathrm{AL,Nex}}$ refers to T_{AL} of the HTC Nexus 9 device and $T_{\mathrm{AL,S3}}$ and $T_{\mathrm{AL,ST3}}$ refer to T_{AL} of Samsung Galaxy S3 and Samsung Galaxy Tab 3 devices respectively (4.17):

$$\begin{aligned}\Delta_{\mathrm{T,S3Nex}} - \Delta_{\mathrm{T,S3ST3}} &= (T_{\mathrm{AL,S3}} - T_{\mathrm{AL,Nex}}) - (T_{\mathrm{AL,S3}} - T_{\mathrm{AL,ST3}}) \\ &= T_{\mathrm{AL,ST3}} - T_{\mathrm{AL,Nex}} \\ &= \Delta_{\mathrm{T,ST3Nex}}\,.\end{aligned} \tag{4.18}$$

This relationship must be satisfied otherwise the STP method and the NTS method would not provide a satisfactory performance in the estimation of T_{O}. As Figure 4.16 shows, $\overline{\Delta_{\mathrm{T,S3Nex}}} \approx 120$ ms and $\overline{\Delta_{\mathrm{T,S3ST3}}} \approx 50$ ms, which results in a difference of 70 ms, a value that corresponds to $\overline{\Delta_{\mathrm{T,ST3Nex}}}$ shown in Figure 4.16d. Therefore, an accurate performance of the STP method for reproduction tasks can be assumed, although there is always a random delay that cannot be controlled and ranges a few milliseconds around the respective mean values of Δ_{T}. Summarizing, results in Figure 4.16 show that even addressing the clock synchronization problem, the acoustic nodes cannot perform simultaneous reproduction because the performance of the Android OS is significantly different, even for devices with identical software and hardware.

4.3.6 Conclusions

The clock synchronization problem has been addressed in this section by using the proposed STP method to compensate the clock difference between devices, T_{O}. The proposed STP method has been presented step by step in order to describe how T_{O} is obtained. Additionally, the Node Task Synchronization (NTS) method has been described to schedule a task such that it is carried out simultaneously in multiple devices.

At this point, with the purpose of validate the performance of the STP method, an experimental comparison has been done with the two most used methods nowadays: the PTP and the NTP methods. The results of this comparison have shown that the proposed STP method is the most stable accurate to estimate T_{O}. As a result, the STP method will be used throughout this dissertation.

Finally, with the purpose of analyzing the performance of the STP

method in the context of audio applications, an experiment has been carried out. The experiment involves two devices connected to an audio analyzer through their analogue audio outputs, so that the audio latency T_{AL} of each device depends only on the device processing. The results obtained when different pair of devices were used show very similar standard deviation values ($\sigma_{\Delta_{\mathrm{T}}}$), but quite different mean values ($\overline{\Delta_{\mathrm{T}}}$). The differences in $\overline{\Delta_{\mathrm{T}}}$ are caused because of the Android OS and hardware differences of the devices. In addition, ideal synchronization, that is $\Delta_{\mathrm{T}} = 0$, cannot be reached even when hardware and software are the same, since each Android OS decides to run the reproduction task at slightly different times. In summary, addressing the clock synchronization problem through the STP method does not solve the synchronization problem for audio applications due to the hardware and software differences of mobile devices.

4.4 Audio Latency Compensation

4.4.1 Overview

The Android OS is currently the most widely used system for mobile devices, as it is illustrated in Figure 2.7. Its popularity is mainly due to the fact that the Android OS can be implemented on a large number of devices from different companies. However, this feature also presents a drawback for synchronizing audio tasks when Android devices are used as nodes in an ASN. Due to the hardware and software differences among devices, a synchronous reproduction between two Android devices cannot be achieved, even when the clock synchronization problem has been addressed. Thus, on a typical scenario, such as the one illustrated in Figure 4.2, where the audio latency T_{AL} depends on more factors than the latency of device (T_{OLD}), a synchronous reproduction between two devices can not be achieved either.

Furthermore, in the considered scenario of Figure 4.2, T_{AL} depends on the acoustic latency, T_{A} (4.3). Therefore, in this section, the main objective is to compensate the audio latency, T_{AL}, of two devices so that a signal emitted by two loudspeakers is synchronized at a specific point in the space, more specifically, where the microphone of one of the nodes is located. Figure 4.17 shows an example of a reproduction through two speakers whose audio signals reach the microphone simultaneously, that is, the audio signals are temporally aligned in the microphone of the first node.

The synchronization at a specific point in the space must be addressed in order to implement sound-field applications such as the CrossTalk canceller presented in Section 4.5.

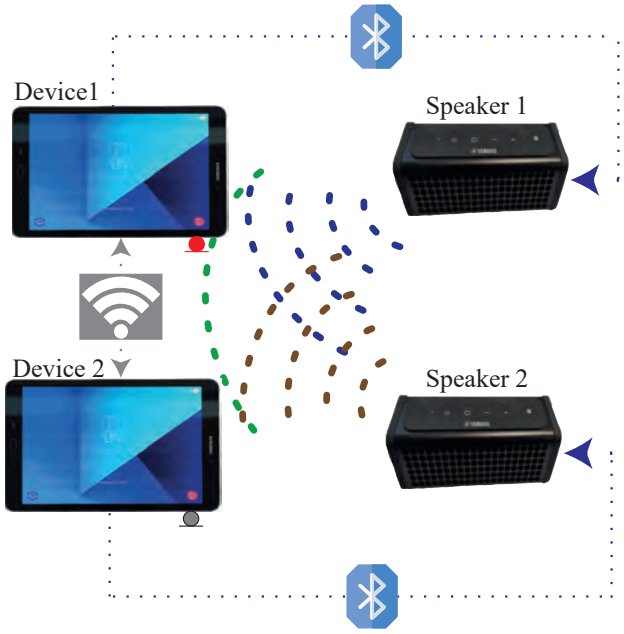


Figure 4.17. Synchronization in the microphone of the first node of two signals emitted by the two speakers.

According to the results shown in Figure 4.16 the reproduction task cannot be perfectly synchronized between two devices, which means that the audio signals emitted by each mobile device are not temporally aligned. As it was stated above, this behavior is due to the difference on T_{OLD} of both devices, denoted in the results as Δ_{T}. Therefore, for the scenario shown in Figure 4.2, the difference Δ_{T} could be even higher than the one obtained in the scenario illustrated in Figure 4.14, since T_{AL} depends on more factors (see (4.3)), causing that the synchronization of the audio signals at the specific point of the space cannot be achieved. In order to improve the synchronization level achieved, an additional method is required. Since Δ_{T} depends on the T_{AL} of each acoustic node, the method explained in this section aims to estimate both T_{AL} and then to align them.

4.4.2 The audio latency compensation (ALC) method

The method explained in this section, from now on the audio latency compensation (ALC) method, estimates the value of T_{AL} of both nodes. For this purpose, a sweep signal of L_{S} samples is simultaneously reproduced and recorded in order to estimate T_{AL} of each node as Section 4.2.2 explains. Since simultaneously reproducing the sweep signal through both devices would interfere between them, the signal used to estimate T_{AL} has been slightly modified, as shown in Figure 4.18, where the desired probe signal is composed of three parts:

1. L_{S} samples of the sweep signal or zeroes. The node labeled as the first node will reproduce the sweep signal and the node labeled as the second one will reproduce zeroes.
2. L_{Z} samples of zeroes to avoid overlapping between both recorded sweeps.
3. L_{S} samples of the sweep signal or zeroes. Similar to the first part, but exchanging the nodes. In this case the second node is the node that reproduces the sweep signal.

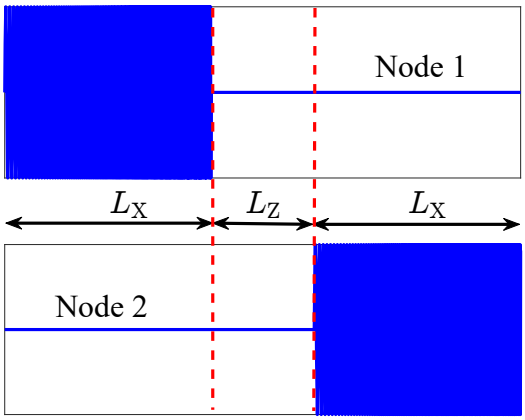


Figure 4.18. Designed probe signal for each acoustic node.

L_{Z} is necessary to separate the second sweep from the first one in the recorded signals. As the results of Figure 4.7 show, the value of T_{AL} can

present differences up to 300 ms between devices. For this reason, it seems reasonable to design a time gap of 1 second, that is, $L_{\mathrm{Z}} = f_{\mathrm{s}}$ samples. In addition, both sweep parts last 2 seconds each, thus $L_{\mathrm{S}} = 2f_{\mathrm{s}}$ samples.

When the signals in Figure 4.18 are reproduced, the microphone of each device will record a signal similar to that shown in Figure 4.19, where for the sake of simplicity the effect of the room is not represented. The Figure 4.19 shows the free-field recorded signal in the i^{th} microphone, where the first part corresponds to the sweep that is reproduced by the first speaker and the second part corresponds to the sweep of the second speaker. Due to each part correspond to a different electro-acoustic path, two audio latencies, T_{AL}, can be estimated. The first one, $p_{\mathrm{AL},i1}$, is the latency of the electro-acoustic path between the i^{th} microphone and the first speaker in samples (4.6), and the second one, $p_{\mathrm{AL},i2}$ is the latency of the electro-acoustic path between the i^{th} microphone and the second speaker in samples.

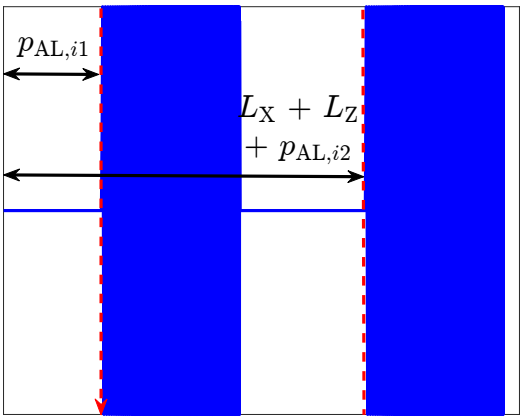


Figure 4.19. Example of a free-field recorded signal in the i^{th} microphone.

Once the reproduction and recording of the sweep signals have been finished, T_{AL} is estimated as Section 4.2.2 indicates. It has to be noticed that T_{AL} can vary for each different recording and reproduction process. In order to overcome this drawback, the audio loop of reproducing and recording must remain open in Android OS. Therefore, each node must perform both processes (reproduction and recording) without any pause.

Figure 4.20 shows the proposed audio latency compensation (ALC) method to deal with the estimation and compensation of the difference between the audio latencies, T_{AL}, of two or more devices. It uses the NTS method of Figure 4.12 to schedule the recording and reproduction tasks, assigning the same roles, Master and Slave, during the whole duration of the ALC process, where Master and Slave correspond to the nodes labelled as "Node 1" and "Node 2" respectively in Figure 4.18. The implementation of the ALC method is shown in Figure 4.20, whose steps are detailed in the following:

1. The Master starts the NTS method where global time is defined ($T^{\mathrm{st}}_{\mathrm{GC1}}$).

2. Once both nodes reach the global time, $T^{\mathrm{st}}_{\mathrm{GC1}}$, each one starts the "Warm-Up" stage (explained in Section 4.2.3), where L_{W} samples of zeroes are played and recorded.

3. Afterwards, the nodes start the "Set-Up" stage defined as the reproduction and recording of the signals presented in Figure 4.18, where $2L_{\mathrm{S}} + L_{\mathrm{Z}}$ samples are reproduced and recorded at each node.

4. The "Audio Processing" stage estimates all the possible T_{AL} such that every node estimates two values of audio latency: those corresponding to the audio loop between two loudspeakers and its own microphone. Afterwards, the estimate values of T_{AL} are exchanged between the two nodes through the Wi-Fi link. While the processing is enabled, both nodes keep on recording and reproducing zeroes.

5. The Slave notifies the Master when it finishes the "Audio Processing" stage.

6. Once the Master has received the message from the Slave and having finished its own "Audio Processing" stage, it starts a second NTS process where a second global time is defined ($T^{\mathrm{st}}_{\mathrm{GC2}}$).

7. Once both nodes reach the second global time, $T^{\mathrm{st}}_{\mathrm{GC2}}$, they compute the time in the number of samples from the beginning of the "Audio Processing" stage, denoted by N_{M} for the Master and N_{S} for the Slave.

8. At this moment, the ALC method reaches its final stage ("Audio Balance"), which compensates the possible mismatch of samples between N_M and N_S. For this purpose, both nodes start to reproducing and recording L_W samples (same number of samples as the "Warm-Up" stage) and the Master sends the value of N_M to the Slave.

9. When the Slave receives N_M from the Master, it modifies its samples to be reproduced from L_W to $L_W + (N_M - N_S)$, in order to compensate the excess or lack of samples with respect to the Master.

10. Once the "Audio Balance" stage is finished, the nodes are able to perform a synchronous reproduction.

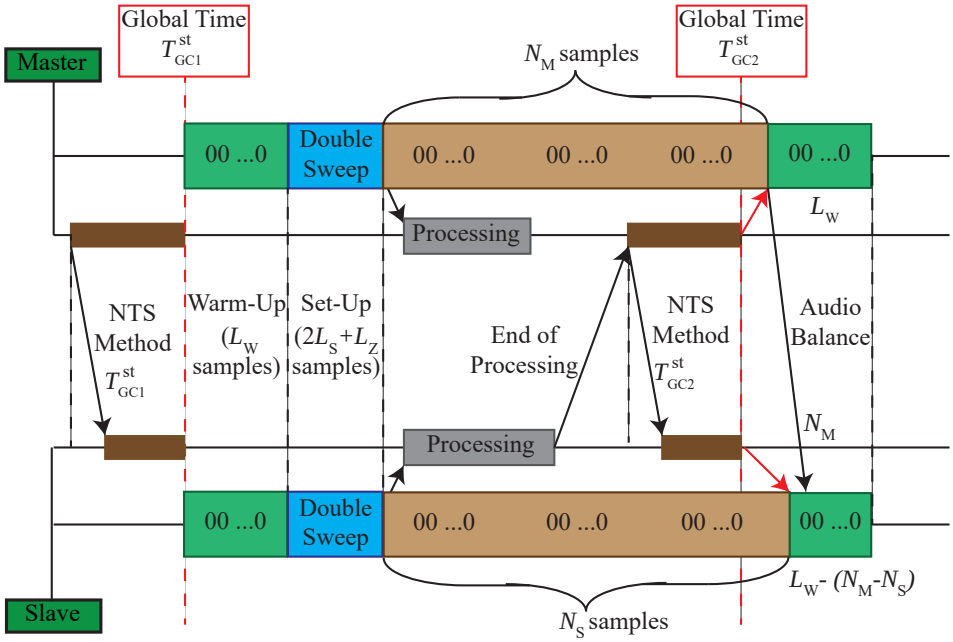


Figure 4.20. Implementation of the Audio Latency Compensation (ALC) method.

As shown in Figure 4.20, one of the most important stages is the "Audio Processing" stage since it estimates all the values of T_{AL} in the ASN. In addition, in this stage the necessary audio signal processing for different applications can be performed, such as that described in Section 4.5.

In contrast to the "Warm-Up" and the "Set-Up" stages, the number of samples to be reproduced and recorded in the "Audio Processing" stage are unknown since the elapsed time to perform the necessary operations in that stage are unknown beforehand. In addition, the processing time of this stage can differ between different mobile devices, even for devices with identical software and hardware. In Figure 4.20 this difference is indicated in the Slave side through a larger "Audio Processing" stage indicating that Slave is slower than the Master, but it can also happen the opposite.

Therefore, due to these factors, both devices are desynchronized again. As a result, a second NTS method is implemented in order to synchronize both devices again. As Figure 4.20 shows, the reproduction and recording tasks are implemented in a different processing thread to that of the processing task, where the NTS method is carried out. This means that the thread of the processing task must communicate to the thread of the reproduction and recording tasks that the NTS method has finished. However, as a brief reminder how Android OS works, the communication time between the threads can be different in each device, as the red arrows show in Figure 4.20. Therefore, even with the second NTS method, the devices will not be synchronized. For this reason, the "Audio Balance" stage is required. This last stage balances the difference of reproduced samples between both devices, causing the two of them to be synchronized.

As Figure 4.20 shows, the node that acts as the Slave corrects the difference of samples reproducing $L_{\mathrm{W}} - (N_{\mathrm{M}} - N_{\mathrm{S}})$ instead of L_{W} samples. That is, the Slave is the node that must adapt the reproduced (and recorded) samples accordingly to the Master. However in order to be able to perform this stage correctly, two main constraints must be considered:

1. $L_{\mathrm{W}} \geq (N_{\mathrm{M}} - N_{\mathrm{S}})$, otherwise the Slave will not be able to compensate the excess or lack of samples regarding the Master.

2. $L_{\mathrm{W}} - (N_{\mathrm{M}} - N_{\mathrm{S}})$ must be greater than the time to send the value N_{M} from the Master to the Slave.

Therefore, according to these constraints, since N_{M} and N_{S} cannot be modified, L_{W} must be big enough to: 1) obtain a positive value of samples to be reproduced and recorded at the Slave side for the "Audio Balance" stage and 2) give the Slave enough time to receive the message of the

Master. Empirical experiment have provided the value of $L_{\mathrm{W}} = 3f_{\mathrm{s}}$, that is, the same value as the "Warm-Up Stage" (see Section 4.2.3).

4.4.3 Performance of the ALC method

In this section, the performance of the ALC method is evaluated by measuring the degree of synchronization achieved between two Android devices. For this purpose, a wireless ASN comprised of two Android devices connected through a Bluetooth link to two speakers (see Figure 4.2) will be used to execute the ALC method shown in Figure 4.20. Once the "Audio Balance" stage has finished, the devices will reproduce and record an audio signal specifically designed to facilitate the measurement of the difference between their respective audio latencies, T_{AL} (4.3).

As it was stated in Section 4.4.1, the temporal alignment can only be solved for one of the two positions where the microphones are located. Without loss of generality, the first microphone has been selected to evaluate the temporal alignment, being identical the process for the second microphone. Since the first microphone is chosen, the audio compensation is carried out through the estimation of $T_{\mathrm{AL},11}$ and $T_{\mathrm{AL},12}$ in number of samples instead of seconds, being $T_{\mathrm{AL},1j}$ the audio latency of the connection between the j^{th} loudspeaker and the first device. Consequently, along this section, it will be estimated the audio latency according to the method described in Section 4.2.2, thus, the variable used to describe this magnitude in samples will be p_{AL} (4.6).

The experiment designed to evaluate the accuracy of the ALC method is the following: both devices will perform simultaneously the ALC method of Figure 4.20 and once the "Audio Balance" stage is finished, a particular audio signal will be reproduced and recorded at each device. These particular audio signals must be designed to accurately estimate the audio latencies of both devices, but they should also avoid to interfere with each other. The signals reproduced are shown in Figure 4.21, where the only difference regarding the signals shown in Figure 4.19 is the addition of a time period of $\Delta p^{(0)}$ zero samples, where $\Delta p^{(0)}$ is calculated as:

$$\Delta p^{(0)} = p^{(0)}_{\mathrm{AL},11} - p^{(0)}_{\mathrm{AL},12}\,, \tag{4.19}$$

where $p^{(0)}_{\mathrm{AL},11}$ and $p^{(0)}_{\mathrm{AL},12}$ are the audio latencies in samples (4.6), between the

first and the second speaker respectively to the first microphone, estimated in the "Audio Processing" stage of Figure 4.20.

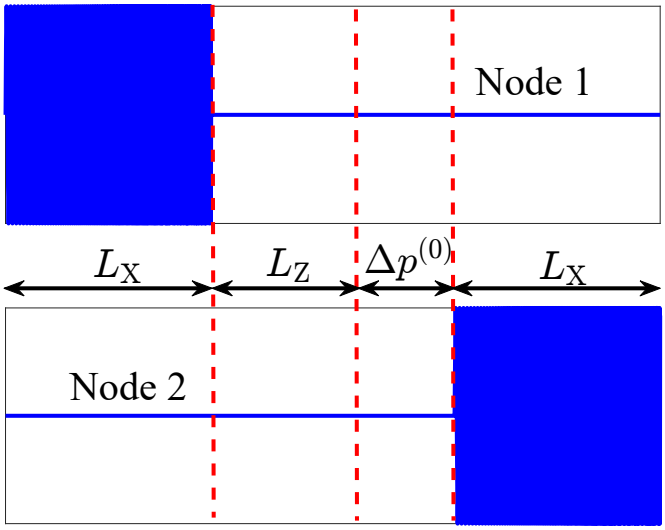


Figure 4.21. Signals to reproduce after "Audio Balance" stage.

As Figure 4.21 shows, $\Delta p^{(0)}$ affects exclusively the starting time of the sweep signal of the second node (the Slave in the ALC method). Therefore, once the particular signals of Figure 4.21 have been reproduced, the first device estimates the corresponding audio latencies in samples, $p_{\mathrm{AL},11}$ and $p_{\mathrm{AL},12}$ through (4.6), and the final difference between the audio latencies can be estimated as:

$$\Delta p = p^{(1)}_{\mathrm{AL},11} - p^{(1)}_{\mathrm{AL},12}\,. \tag{4.20}$$

The difference Δp represents the degree of synchronization (in number of samples) that can be achieved in the WASN of Figure 4.2, that is, Δp is the measured synchronization error of the ALC method.

As it was stated above, two Android devices and two speakers are needed at the same time to perform this study. The mobile devices and speakers have been placed symmetrically in order to provide similar distances between the microphones and the speakers, reducing the difference between the delay due to the physical propagation of the sounds as much as

possible. A picture of the physical disposition of the devices and loudspeakers is shown in Figure 4.22, where distances between all the elements are approximately 50 cm. In addition, as Figure 4.22 shows, all the elements have been placed over cardboard boxes to avoid the influence of the table vibration.

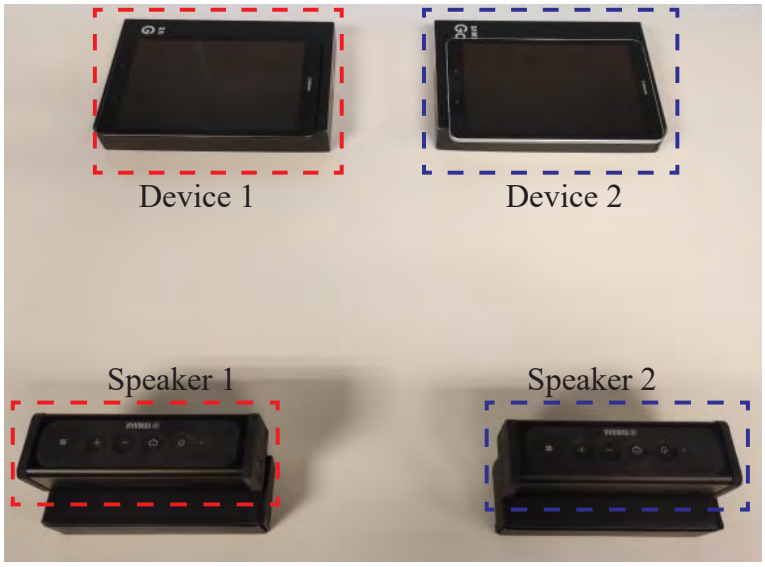


Figure 4.22. Devices and loudspeakers disposition in the scenario of the experimental validation.

Although in Figure 4.22 a homogeneous combination with identical mobile devices and speakers is shown, in this section different combinations of both elements are used, since the audio latency, p_{AL}, depends on the mobile device and the speaker, as it was concluded in Section 4.2.3. These combinations are shown in Table 4.6, where in the second node the mobile device is combined as many times as speakers are listed in the SPK subtable, that is, each mobile device of the second node is combined with four different speakers. Therefore, in this study a total of 16 different combinations are used.

Similarly to the study of Section 4.2.3, the mobile devices must decide its sample rate, f_{s}, and its data buffer size, L_{B}. The sample rate is set to $f_{\mathrm{s}} = 44100$ Hz, whereas the buffer size is set to $L_{\mathrm{B}} = \frac{L_{\mathrm{B,ref}}}{2}$ (using $L_{\mathrm{B,ref}}$ as

First Node	Second Node
Samsung Galaxy S3 & Yamaha NX P-100	Samsung Galaxy S3 & SPK
	HTC Nexus 9 & SPK
	Motorola Moto G5 Plus & SPK
HTC Nexus 9 & Yamaha NX P-100	Motorola Moto G5 Plus & SPK

(a) First and second node configurations.

SPK			
Yamaha NX P-100	Sony SRS X3	JBL Flip 2	JBL Charge 4

(b) Speakers of the second node.

Table 4.6. Combinations of the study.

Table 4.2 indicates), as a result of the study performed in Appendix C.

Considering all these scenarios and constraints, the next steps are followed in order to measure the synchronization error at the control point of an Android-based WASN:

1. Select a combination of the first and second node from Table 4.6.
2. Perform the ALC method of Figure 4.20, establishing first node as the Master and the second node as the Slave and assuming $\Delta_{\mathrm{M}} = 1$ second in the NTS method shown in Figure 4.12.
3. In the "Audio Processing" stage:
   (a) The first node estimates $p^{(0)}_{\mathrm{AL},11}$ and $p^{(0)}_{\mathrm{AL},12}$ and its difference $\Delta p^{(0)}$.
   (b) The first nodes sends the value $\Delta p^{(0)}$ to the second node and each one generates the corresponding audio signal of Figure 4.21.
4. Signals in Figure 4.21 are reproduced once the "Audio Balance" stage finishes.

5. The estimation of $p^{(1)}_{\mathrm{AL},11}$ and $p^{(1)}_{\mathrm{AL},12}$ is carried out in the first device from the signal recorded at its microphone. Afterwards, Δp is estimated through (4.20).

6. Return to step 2 and repeat the process 50 times.

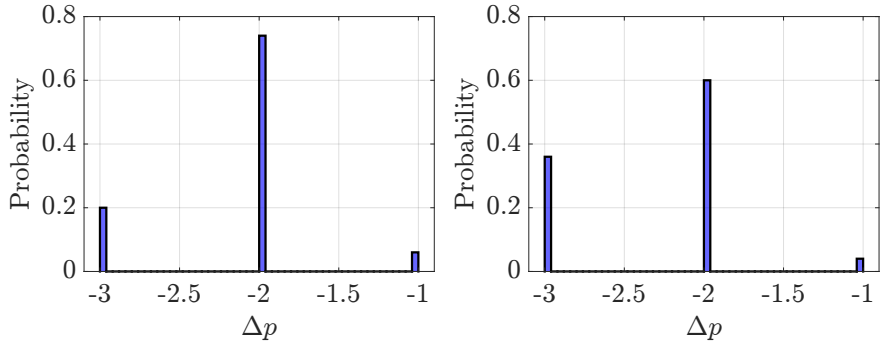
7. Return to step 1 in order to select a different combination.

The results obtained from the previous procedure have been classified by speaker combination. According to this classification and Table 4.6, four different groups of results are presented: 1) Yamaha NX P-100-Yamaha NX P-100, 2) Yamaha NX P-100-Sony SRS X3, 3) Yamaha NX P-100-JBL Flip 2 and 4) Yamaha NX P-100-JBL Charge 4. The results depicted in Figures 4.23-4.26 show the normalized histograms of Δp for each speaker combination.

Figure 4.23 presents the results when the Yamaha NX P-100 speaker is used in both nodes. In this case, the temporal difference between both nodes is -2 ± 1 samples whatever the mobile device combination is. That is, the differences in hardware and software between the mobile devices do not affect the ALC synchronization method, as opposed to the results shown in Section 4.3.4. Despite this behavior, a slightly inaccurate synchronization can be appreciated in all the cases of Figure 4.23, since $\Delta p = -2$ samples with a deviation of ±1 sample, which in seconds represent 45 ± 22 μs.

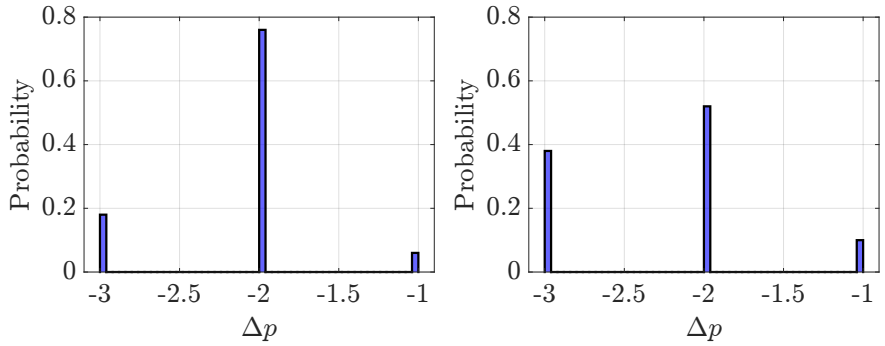
The results where the Yamaha NX P-100 and Sony SRS X3 are used as the speakers of the acoustic nodes are shown in Figure 4.24. Similar to the results shown in Figure 4.23, Δp does not show difference between the different mobile device combinations. In fact, the only difference with the previous results is the value of Δp, being in this case 2 ± 1 samples. Although the change of sign is explained below, this behavior means that the change of the speaker has a significant impact over the temporal difference between the signals emitted by each node since it does not change for the different mobile device combinations, but it changes when the speaker combination is changed. Nevertheless, the results of this speaker combination confirm a very stable behavior of the ALC method, as in the previous case, since in both speaker combinations the deviation of Δp is $\pm$ 1 sample.

Figure 4.25 shows the histogram of the Δp values when the Yamaha NX P-100 and the JBL Flip 2 are used as the speakers of the acoustic nodes. In this case a very different behavior can be appreciated since the difference



(a) Samsung Galaxy S3 - Samsung Galaxy S3.

(b) Samsung Galaxy S3 - HTC Nexus 9.

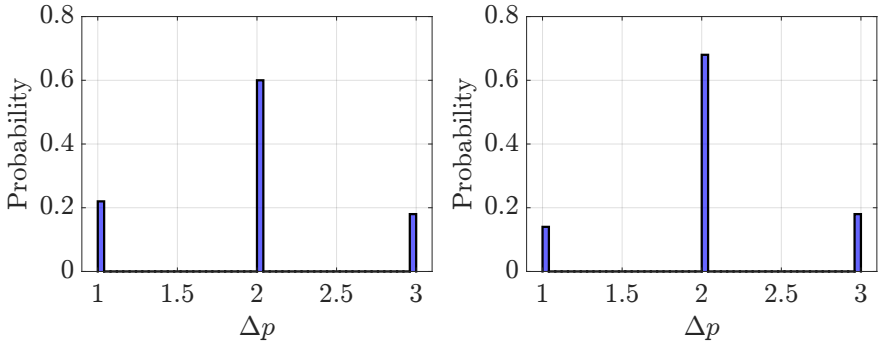


(c) Samsung Galaxy S3 - Motorola Moto G5.

(d) HTC Nexus 9 - Motorola Moto G5.

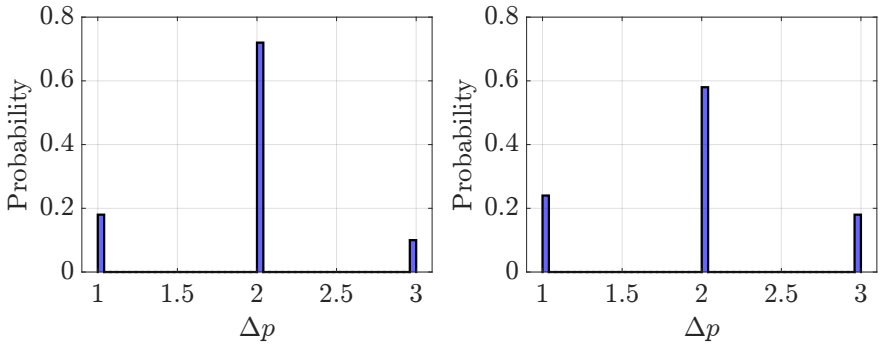
Figure 4.23. Synchronization error using the Yamaha NX P-100 - Yamaha NX P-100 speaker combination.

between the nodes reaches 800 samples, an extremely high value of Δp compared to the previous results. Notice that the bars of the histogram represent a width of 15 samples. Since the only difference compared to the previous scenarios is the integration of the JBL Flip 2 speaker in the second node, the high synchronization errors are caused by the JBL Flip 2 speaker. Despite this average error of 800 samples (17 ms), a similarity with the previous combinations can be appreciated in the sense that the average value of Δp is the same regardless of the mobile device combination.



(a) Samsung Galaxy S3 - Samsung Galaxy S3.

(b) Samsung Galaxy S3 - HTC Nexus 9.

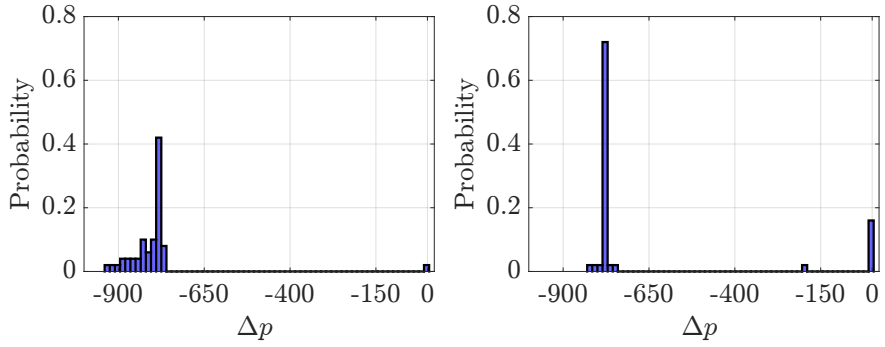


(c) Samsung Galaxy S3 - Motorola Moto G5.

(d) HTC Nexus 9 - Motorola Moto G5.

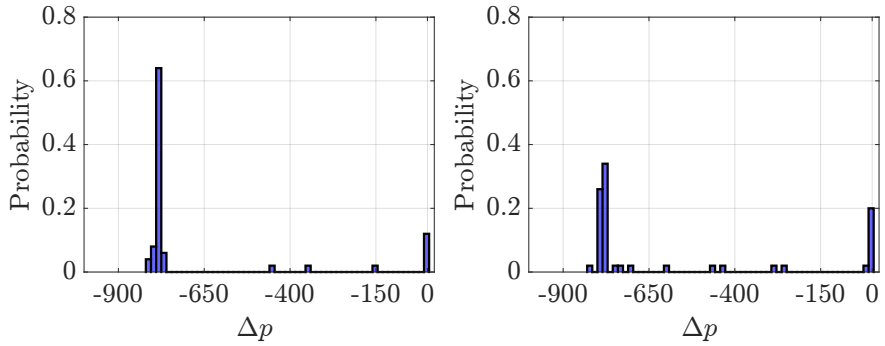
Figure 4.24. Synchronization error using the Yamaha NX P-100 - Sony SRS X3 speaker combination.

Finally, Figure 4.26 shows the results where the Yamaha NX P-100 and the JBL Charge 4 are used. In this case, the results show a high dispersion of the synchronization error (around 250 samples or 5.7 ms). However, for this combination of speakers, the average value of Δp is smaller than that of Figure 4.25, that is, the WASN synchronization is better. Summarizing, the value of Δp is high and unstable for the two last speaker combinations. Their behavior shown in Figures 4.25 and 4.26 confirm that they are not suitable for implementing simultaneous audio applications over ASN.



(a) Samsung Galaxy S3 - Samsung Galaxy S3.

(b) Samsung Galaxy S3 - HTC Nexus 9.

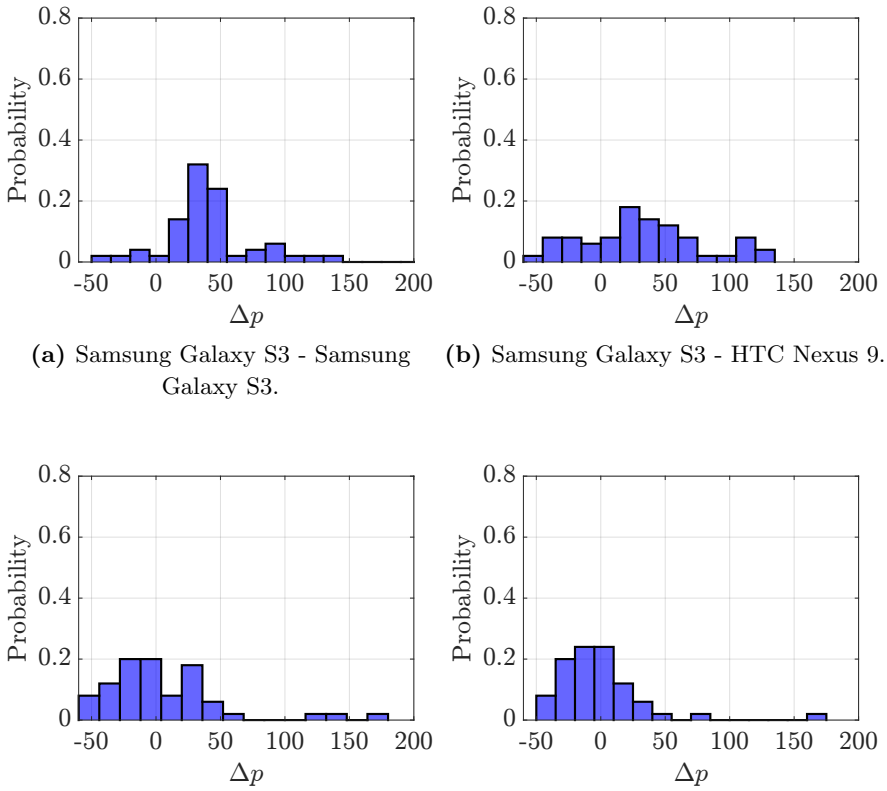


(c) Samsung Galaxy S3 - Motorola Moto G5.

(d) HTC Nexus 9 - Motorola Moto G5.

Figure 4.25. Synchronization error using the Yamaha NX P-100 - JBL Flip 2 speaker combination.

Given all these results, a perfect synchronization between nodes for audio tasks cannot be guaranteed since $\Delta p \neq 0$ in any case. According to these results, Δp suffers an offset, preventing both nodes from carrying out synchronized audio tasks. Since the offset suffered by Δp changes significantly when the commercial speaker combination is changed, it is considered that the offset depends on the speakers. Due to Δp is estimated through the difference between $p^{(1)}_{\mathrm{AL},11}$ and $p^{(1)}_{\mathrm{AL},12}$, this offset can only be



(a) Samsung Galaxy S3 - Samsung Galaxy S3.

(b) Samsung Galaxy S3 - HTC Nexus 9.

(c) Samsung Galaxy S3 - Motorola Moto G5.

(d) HTC Nexus 9 - Motorola Moto G5.

Figure 4.26. Synchronization error using the Yamaha NX P-100 - JBL Charge 4 speaker combination.

caused because one of these two reasons: 1) $p_{\mathrm{AL},11}$ or $p_{\mathrm{AL},12}$ suffer an offset between the first and the second estimation (exclusively one of them) or 2) both of them suffer an offset between the first and the second estimation, but each one suffers a different one. In this regard, in the following paragraphs, a more in-depth study is performed in order to analyze the offset suffered by p_{AL}. To do this, two additional steps are carried out in addition to the steps of the previous study. Figure 4.27 illustrates the whole procedure performed in this new study.

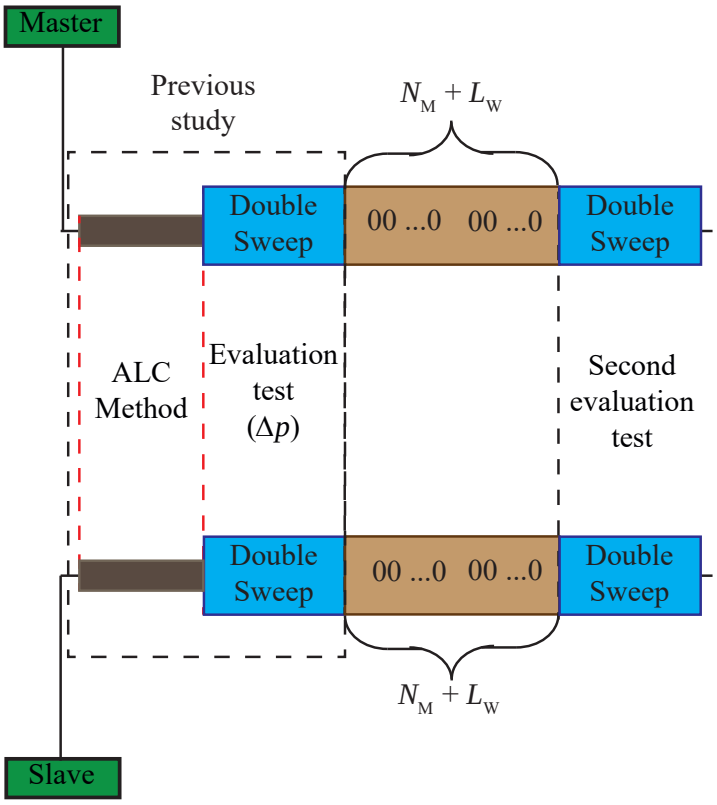


Figure 4.27. Procedure to study the offset suffered by p_{AL}.

According to Figure 4.27, after the steps of the previous study, a reproduction and recording of $N_{\mathrm{M}}+L_{\mathrm{W}}$ zero samples is carried out. Afterwards, the reproduction and recording of the signals of Figure 4.21 is performed again. These steps result in a third estimation of the audio latencies $p^{(2)}_{\mathrm{AL},11}$ and $p^{(2)}_{\mathrm{AL},12}$ through (4.6). The use of $N_{\mathrm{M}}+L_{\mathrm{W}}$ zero samples in Figure 4.27 is required in order to keep the same conditions between the estimation of $\Delta p^{(0)}$ and Δp and the estimation of Δp and the values of $p^{(2)}_{AL,11}$ and $p^{(2)}_{\mathrm{AL},12}$. The third estimation is required because between the "Set-Up" stage (where $p^{(0)}_{\mathrm{AL},11}$ and $p^{(0)}_{\mathrm{AL},12}$ are obtained) and the reproduction and recording of the signals of Figure 4.21 (where $p^{(1)}_{\mathrm{AL},11}$ and $p^{(1)}_{\mathrm{AL},12}$ are obtained), the Android OS controls both audio and processing tasks simultaneously. This behavior

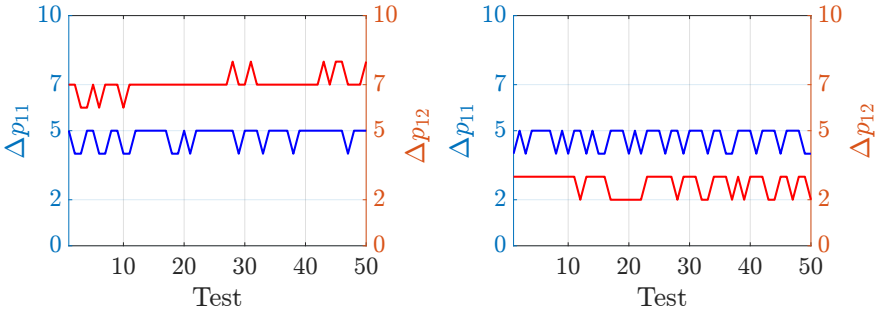
can affect the computation time of the audio tasks since the Android OS needs to share the available resources. Therefore, a third estimation with no processing is used in order to obtain the offset suffered by each link between the second and the third estimation of p_{AL}, which is estimated as:

$$\begin{aligned} \Delta p_{11} &= p^{(2)}_{\mathrm{AL},11} - p^{(1)}_{\mathrm{AL},11} \\ \Delta p_{12} &= p^{(2)}_{\mathrm{AL},12} - p^{(1)}_{\mathrm{AL},12}\,. \end{aligned} \tag{4.21}$$

Same disposition shown in Figure 4.22 is used in this study. Additionally, since the previous results showed similar Δp values regardless the combination of mobile devices, for the sake of simplicity, only one combination is analyzed in the following. More specifically, the Samsung Galaxy S3 device is used in both nodes.

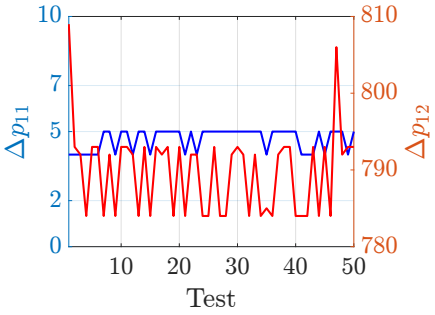
The results of this study are shown in Figure 4.28 for the 50 tests carried out. The offset suffered by $p_{\mathrm{AL},11}$ (Δp_{11}) is shown on the left Y axis in blue line, and the offset suffered by $p_{\mathrm{AL},12}$ (Δp_{12}) is shown on the right Y axis in red line. Each Y axis has its own scale due to the different values of the offset of Δp_{11} and Δp_{12}, specially for the two last combinations.

As Figure 4.28 shows, the offset suffered by $p_{\mathrm{AL},11}$ and $p_{\mathrm{AL},12}$ is always different, even when identical speakers are used in Figure 4.28a. In this case the difference between them match the results seen in Figure 4.23, where a difference of -2 ± 1 samples was shown. The offset in $p_{\mathrm{AL},11}$ and $p_{\mathrm{AL},12}$ is due to the fact that T_{DP} and T_{SP} present a behavior that is not constant, resulting in a reproduction that is not continuous and consequently modifying $p_{\mathrm{AL},11}$ and $p_{\mathrm{AL},12}$. However, the offset on $p_{\mathrm{AL},11}$ and $p_{\mathrm{AL},12}$ is very low and stable and it can be predicted and compensated, thus improving the synchronization between the acoustic nodes. When different speakers are used, two different cases can be appreciated. The first case, when the Yamaha NX P-100 and Sony SRS X3 speakers are used, presents a very similar behavior than the previous case. In fact the only difference is that Δp_{12} is lower than Δp_{11}, causing the change of sign seen in Figure 4.24. The second case, when the Yamaha NX P-100 and any of the JBL speakers are used, shows that Δp_{12} is significantly different than Δp_{11}, as opposed to the previous cases. In addition, the value of Δp_{12} match the results of Δp shown in Figures 4.25 and 4.26. That is, the high value and the unstable behavior seen in those results are caused exclusively

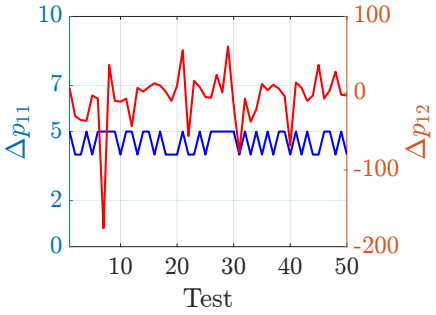


(a) Yamaha NX P-100 - Yamaha NX P-100.

(b) Yamaha NX P-100 - Sony SRS X3.



(c) Yamaha NX P-100 - JBL Flip 2.



(d) Yamaha NX P-100 - JBL Charge 4.

Figure 4.28. Latency offset using the Samsung Galaxy S3 device in both nodes.

by the JBL speakers since the value of Δp_{11} is similar to the previous cases.

Contrary to what might be assumed, Δp_{11} and Δp_{12} are different when identical mobile and speakers are used, as it is shown in Figure 4.28a. According to [202], two devices (even with identical hardware and software) suffer an offset between them due to slightly differences in the value of their actual sample rate f_s. This offset is caused by hardware components, more specifically, by the clock of the digital to analog converter (DAC). This difference causes the signals to be sampled at slightly different sample rates, producing an offset over time. In the particular case that the results are obtained from the microphone of the first node, the reference is given

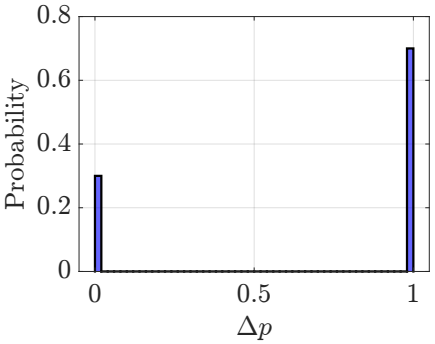
by the first node and as a result the latency from the speaker of the second node ($p_{\mathrm{AL},12}$) suffers an additional offset that in this case is positive. This additional offset is called Sample Rate Offset (SRO) [202].

In Figure 4.28b, when different speakers are used, a slightly difference between Δp_{11} and Δp_{12} can be appreciated as well. However, in this case, this difference may not be caused exclusively by the SRO, but by additional offsets introduced by other differences between the speakers, such as the buffer length, L_{B}. In contrast to the previous speaker combination (see Figure 4.28a), an independent measurement of these additional offsets cannot be carried out. Therefore, their impact cannot be accurately demonstrated. Nevertheless, the additional offset suffered by Δp_{12} demonstrates to be low and stable, resulting in a similar performance to that shown by the combination of identical speakers. Regarding to the combinations where the JBL speakers are used, the SRO is also present, but due to the unstable behavior it cannot be easily detected.

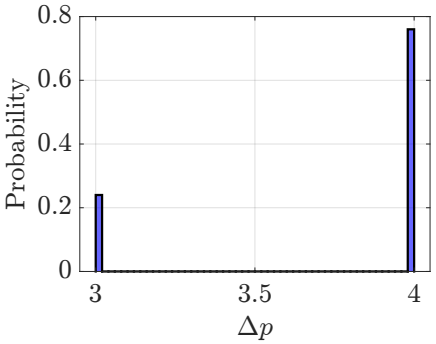
The results of Figure 4.28 show that the offset suffered by $p_{\mathrm{AL},11}$ and $p_{\mathrm{AL},12}$ highly depends on the speaker. Combinations using the Yamaha NX P-100 and Sony SRS X3 loudspeakers show a very stable behavior and a high level of synchronization, but those using the JBL speakers show a poor level of synchronization and a high dispersion of the synchronization error.

A deeper study on the behavior of this offset has been developed and it will be explained in the following. For this purpose, we have implemented all the experiments performed to obtain the results shown in Figures 4.23 - 4.28, but replacing the Bluetooth link by a wired link. Additionally, for the sake of simplicity, only a two-speaker combinations and two mobile device combinations have been considered. For the loudspeakers combination: 1) the Yamaha NX P-100 speaker is used in both speakers and 2) the Yamaha NX P-100 speaker and the JBL Flip 2 are used. For the mobile device combination: 1) the Samsung Galaxy S3 is used in both nodes and 2) the Samsung Galaxy S3 and the HTC Nexus 9 have been used.

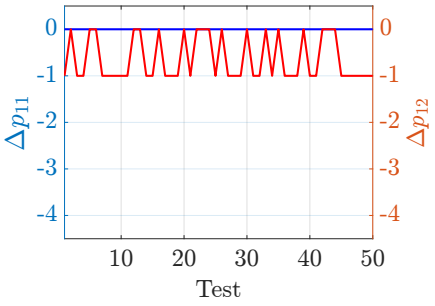
The synchronization error, Δp, obtained in this new experiment is shown in Figures 4.29 and 4.30, where each figure represents different speaker combination. The first row of each figure shows the normalized histograms of Δp for each speaker combination with a bar width of one sample. The second row of each figure shows the offset that suffer $p_{\mathrm{AL},11}$ (Δp_{11}) and $p_{\mathrm{AL},12}$ (Δp_{12}).



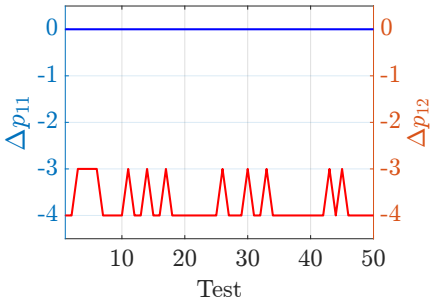
(a) Samsung Galaxy S3 - Samsung Galaxy S3.



(b) Samsung Galaxy S3 - HTC Nexus 9.



(c) Samsung Galaxy S3 - Samsung Galaxy S3.

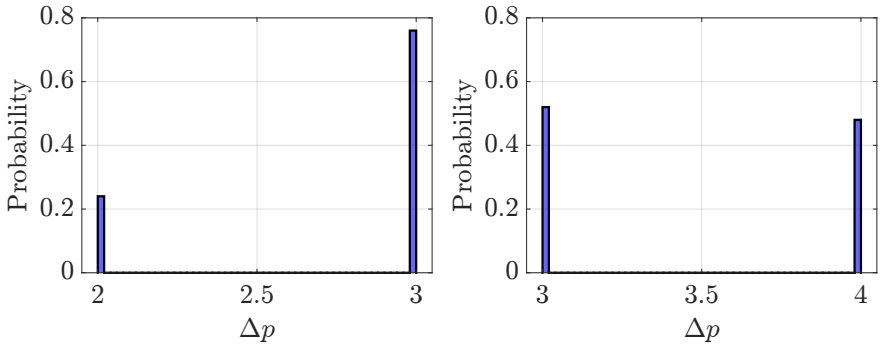


(d) Samsung Galaxy S3 - HTC Nexus 9.

Figure 4.29. Results using Yamaha NX P-100 as speakers in both nodes, where (a) and (b) show Δp and (c) and (d) show Δp_{11} and Δp_{12} respectively.

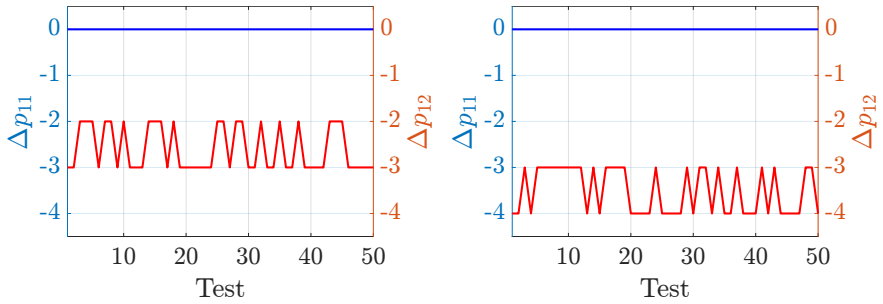
Figure 4.29 represents the results when Yamaha NX P-100 is used in both nodes. In this case, when identical mobile devices are used, the SRO is clearly detected since $p_{\mathrm{AL},11}$ does not suffers any offset, as opposed to $p_{\mathrm{AL},12}$. In fact, the variations in the value of the offset Δp_{12} are exclusively caused by the SRO since the Bluetooth link has been replaced by a wired link. Regarding Figure 4.29a where identical devices are used, it can be appreciated how the synchronization error is 1 sample (22 μs) the 70% of the times and 0 the rest, which s a very good result and can also be explained by

the SRO. However, when two different devices are used (Figure 4.29b), Δp changes unlike the results when the Bluetooth link is used. That is, when wired connection is used, the mobile device combination has an impact on the value Δp. Similar to the case where identical mobile devices are used, in this case the offset suffered by $p_{\mathrm{AL},12}$ is caused because of slightly differences between the acoustic nodes, such as f_{s}. Although in this case the mobile device affects the offset as well, other parameters could have a certain influence, although they cannot be measured individually in this scenario.



(a) Samsung Galaxy S3 - Samsung Galaxy S3.

(b) Samsung Galaxy S3 - HTC Nexus 9.



(c) Samsung Galaxy S3 - Samsung Galaxy S3.

(d) Samsung Galaxy S3 - HTC Nexus 9.

Figure 4.30. Results using Yamaha NX P-100 and JBL Flip 2 speakers, where (a) and (b) shows Δp and (c) and (d) Δp_{11} and Δp_{12} respectively.

Figure 4.30 shows the results when the Yamaha NX P-100 and JBL Flip 2 speakers are used. In this case a similar performance to that shown in Figure 4.29 can be appreciated. In fact, when different mobile devices are used, the behavior is identical to the results shown in Figure 4.29. However, when mobile devices are the same, a slightly difference compared with the results shown in Figure 4.29 can be appreciated. This difference is caused by an additional offset, apart from SRO, that suffers $p_{\mathrm{AL},12}$ due to the processing of the speaker when a wired connection is used, that is, T_{SP}. In the combination of two different devices, the SRO is also present, but additive offsets due to different hardware and software may also affect the result, causing the same offset of Figure 4.29. Nevertheless, Figure 4.30 shows that the offset suffered by $p_{\mathrm{AL},12}$ is significantly lower than that shown in Figure 4.28. In addition, the dispersion of the synchronization error shown in Figure 4.28 has disappeared, resulting in an ASN that presents a very stable offset between the acoustic nodes.

Given these results, we can conclude that the additional factor that causes the offset in $p_{\mathrm{AL},11}$ and $p_{\mathrm{AL},12}$ is the processing burden needed by the Bluetooth connection. This increase on the processing time causes an unstable reproduction process that changes p_{AL} between different estimations. For this reason, the change in p_{AL} depends on this Bluetooth processing and in particular on the speaker that performs that processing. In this regard, a lower and more stable offset is obtained when the Yamaha and Sony loudspeakers are used in the nodes and a higher and more unstable offset is obtained when nodes are composed by JBL loudspeakers.

4.4.4 Conclusions

The temporal alignment problem has been addressed in this section and the ALC method has been proposed to perform a simultaneous recording and reproduction between two devices. The ALC method compensates the audio latency (T_{AL}) between two devices, and performs a synchronized reproduction at a specific point in the space, more specifically, where the microphone of one of the devices is located. This approach will be the solution to some critical applications such as the CrossTalk canceller of Section 4.5.

With the purpose of validate the performance of the ALC method, two experiments has been carried out. The first experiment consists in two devices connected through a Wi-Fi link between them and connected each one

through a Bluetooth link to a wireless speaker. With the aim of determine very precisely the offset due to the devices and speakers features, a second experiment has been carried out. That experiment considers the same set up that in the first one, but replacing the Bluetooth links by wired ones. The results of the first experiment showed that synchronization error, Δp, significantly depends on the combination of the speakers. The combinations using the Yamaha NXP-100 and Sony SRS X3 loudspeakers presented a low value and a very stable behavior of Δp, while the combinations using the JBL loudspeakers presented a high value and an unstable behavior of Δp. However, the ALC is not able to achieve $\Delta p \neq 0$ since the latencies of the links, between the two loudspeakers and the first device, suffer an offset. This offset is different for each loudspeaker, causing a different value for the synchronization error, Δp, for each combination of speakers, which in the case of the JBL speakers, results in an unstable behavior. Regarding the results of the second experiment, Δp showed a low value and a stable behavior for all the combinations of speakers. When the loudspeakers are connected via a wired link, the combination that uses the Yamaha speakers was able to achieve perfect synchronization 30% of the times the experiment was run. As a result, we can conclude that the offset suffered by $p_{\mathrm{AL},11}$ and $p_{\mathrm{AL},12}$ and, consequently, the behavior of Δp, depends significantly on the processing burden needed by the speakers when using Bluetooth connections. Therefore, for audio applications implemented on a wireless ASN composed by Android devices and wireless speakers, the processing burden of the speakers is critical for the synchronization problem.

4.5 Personal Sound Zone (PSZ)

4.5.1 Introduction

In this section, the synchronization algorithms of the previous sections will be used to implement a real-time reproduction application known as Personal Sound Zone (PSZ) [203], where the main objective is to reproduce different audio signals to different users in the same room with minimum interference between them without using headphones [204]. To this end, a set of filters for each audio signal and each loudspeaker considering the local RIRs of the zones to cover is considered [5].

When dealing with a single audio signal, the aim of a PSZ system

is usually the generation of two zones: the bright zone, where the sound can be perceived, and the dark zone, where the sound is attenuated [205]. For the multi-sound zone reproduction problem, where different sounds are rendered to multiple users (generating multiple bright zones), the superposition of multiple bright and dark zones is required [206]. An early example of this approach is the Cross-Talk canceller application [207], where the left and the right channel of an stereo sound are the audio signals and the two bright zones would be the ears of the listener. In the following, the basic model of a PSZ system with a bright and a dark zone will be explained.

Consider the PSZ system of Figure 4.31 that is composed by J speakers and M microphones, where the bright zone is delimited by M_{B} microphones, and the dark zone is delimited by M_{D} microphones, such that $M = M_{\mathrm{B}} + M_{\mathrm{D}}$. The RIR between the j^{th} speaker and the i^{th} microphone is modelled as the finite impulse response (FIR) filter:

$$\mathbf{c}_{ij} = \left[\begin{array}{cccc} c_{ij}(0) & c_{ij}(1) & \cdots & c_{ij}(L_{\mathrm{c}}-1) \end{array} \right]^T , \tag{4.22}$$

where $i = 1, \ldots, M$, $j = 1, \ldots, J$ and L_{c} is the maximum length in samples.

According to Figure 4.31, the signals recorded by the microphones, $y_i(n)$, can be expressed in terms of the signals reproduced by the speakers, $v_j(n)$, as:

$$y_i(n) = \sum_{j=1}^{J} \mathbf{c}_{ij} * v_j(n) \qquad i = 1, \ldots, M , \tag{4.23}$$

where $*$ stands for the discrete-time convolution.

As it was stated above, the PSZ system requires a set of filters in order to achieve the corresponding effect at each zone. The sound source in the PSZ system, $s(n)$, is the input to that set of filters modelled as FIR filters of L_{h} coefficients, such as

$$\mathbf{h}_j = \left[\begin{array}{cccc} h_j(0) & h_j(1) & \cdots & h_j(L_{\mathrm{h}}-1) \end{array} \right]^T \qquad j = 1, \ldots, J. \tag{4.24}$$

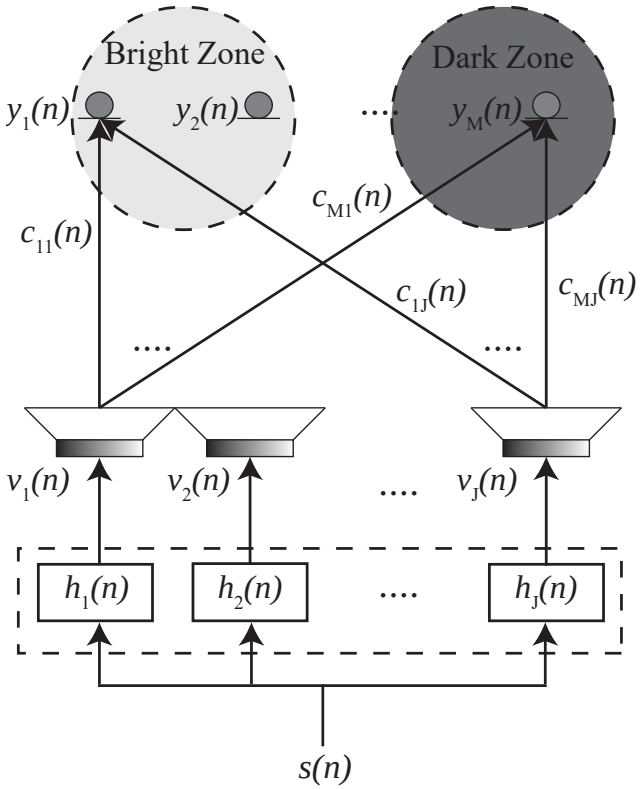


Figure 4.31. PSZ System of J speakers and M microphones.

Considering the filters, $\mathbf{h}_j$, the expression (4.23) can be rewritten as:

$$y_i(n) = \sum_{j=1}^{J} \mathbf{c}_{ij} * \mathbf{h}_j * s(n) = \mathbf{g}_i * s(n)\,, \tag{4.25}$$

where $\mathbf{g}_i = \sum_{j=1}^{J} \mathbf{c}_{ij} * \mathbf{h}_j$ is the global room impulse response between the source $s(n)$ and the i^{th} microphone with $L_{\text{g}} = L_{\text{c}} + L_{\text{h}} - 1$ coefficients.

In order to produce the bright and the dark zone in the same room, the filters of each speaker ($\mathbf{h}_j$) must be designed such that $\mathbf{g}_i$ fulfills a certain requirement in each zone. The filters can be designed using a time-domain approach [208] or a frequency-domain approach [209]. We will

use the frequency-domain approach since it requires lower computational cost [210, 211].

Let us define $C_{ij}(k)$ and $H_j(k)$ as the FFT of $\mathbf{c}_{ij}$ and $\mathbf{h}_j$ respectively at the k^{th} frequency bin. The matrix including the room transfer functions between all the speakers (J) and all the microphones (M) can be defined as

$$\mathbf{C}(k) = \begin{bmatrix} C_{11}(k) & C_{12}(k) & \cdots & C_{1J}(k) \\ C_{21}(k) & C_{22}(k) & \cdots & C_{2J}(k) \\ \vdots & \vdots & \ddots & \vdots \\ C_{M1}(k) & C_{M2}(k) & \cdots & C_{MJ}(k) \end{bmatrix}, \tag{4.26}$$

and the frequency responses of all the filters are defined as

$$\mathbf{h}(k) = \begin{bmatrix} H_1(k) & H_2(k) & \cdots & H_J(k) \end{bmatrix}^T, \tag{4.27}$$

where $k = 0, \ldots, N_{\text{FFT}} - 1$ and N_{FFT} is the FFT size.

In the frequency domain, the FFT ($G_i(k)$) of the global room transfer functions (see (4.25)) can be computed as a simple matrix multiplication:

$$\mathbf{g}(k) = \mathbf{C}(k)\mathbf{h}(k), \tag{4.28}$$

where $\mathbf{g}(k) = [G_1(k)\ G_2(k)\ \cdots\ G_M(k)]$.

Therefore, considering that the global impulse responses have a length of L_{g} samples, the FFT size must be at least of that length, that is, $N_{\text{FFT}} \geq L_{\text{g}}$. As it was stated above, the global responses $\mathbf{g}(k)$ must accomplish a specific objective in order to generate a bright and a dark zone. Let us define the objective global impulse response as

$$\mathbf{d}(k) = \begin{bmatrix} D_1(k) \\ D_2(k) \\ \vdots \\ D_{M_{\text{B}}}(k) \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{4.29}$$

where the bright zone is defined through $D_i(k)$ with $i = 1, \ldots, M_{\mathrm{B}}$ and the dark zone is defined by zeroes for the rest of the microphones (M_{D}) since $s(n)$ must be cancelled there. Therefore, the filters $\mathbf{h}(k)$ are designed such that the global impulse responses $\mathbf{g}(k)$ are equal to $\mathbf{d}(k)$. To this end, the optimal coefficients of $\mathbf{h}(k)$ must minimize the following cost function:

$$J(\mathbf{h}(k)) = \left\|\mathbf{g}(k) - \mathbf{d}(k)\right||^2 + \beta||\mathbf{h}(k)\|^2 \,. \tag{4.30}$$

where β is a regularization parameter [212]. The solution to the minimization of (4.30) is given by [212, 213]:

$$\mathbf{h}(k) = \left(\mathbf{C}^H(k)\mathbf{C}(k) + \beta\mathbf{I}\right)^{-1}\mathbf{C}^H(k)\mathbf{d}(k)\,, \tag{4.31}$$

where $\mathbf{I}$ is the identity matrix.

When the PSZ system aims to generate two bright zones instead of one bright and one dark zones, a superposition between two bright and two dark zones must be carried out. In this sense two different PSZ systems of one bright and one dark zones each one are designed. The first PSZ system considers that the first zone is the bright zone and the second zone is the dark one, while the second PSZ system considers the opposite case. Then, the combination of both PSZ systems will provide a PSZ system with two bright zones. Consequently, the expression in (4.31) must be solved twice, each one with a different objective vector $\mathbf{d}(k)$. However, the same solution expressed in (4.31) can be used for the case of two bright zones but considering $\mathbf{H}(k) = [\mathbf{h}_1(k)\ \mathbf{h}_2(k)]$ and $\mathbf{D}(k)$ as:

$$\mathbf{D}(k) = \begin{bmatrix}\mathbf{d}_1(k) & \mathbf{d}_2(k)\end{bmatrix} = \begin{bmatrix} D_1(k) & 0 \\ D_2(k) & 0 \\ \vdots & \vdots \\ D_{M_{\mathrm{B}}}(k) & 0 \\ 0 & D_{M_{\mathrm{B}}+1}(k) \\ \vdots & \vdots \\ 0 & D_{M_{\mathrm{B}}+M_{\mathrm{D}}}(k) \end{bmatrix}, \tag{4.32}$$

where $\mathbf{h}_1(k)$ and $\mathbf{d}_1(k)$ are the filters and the global room response related to the first PSZ system and $\mathbf{h}_2(k)$ and $\mathbf{d}_2(k)$ are the filters and the global room response related to the second PSZ system.

4.5.2 PSZ system in a two-node Android wireless ASN

PSZ systems can be implemented in an ASN using its nodes to generate the sound zones [214]. This approach is shown in Figure 4.32, where a two-node ASN formed by commercial devices is used. In this ASN, each node is composed by a single microphone and a single speaker, thus, $M = 2$ and $J = 2$. Since the objective is to generate two bright zones, there are two filters per loudspeaker in Figure 4.32, that can be expressed as:

$$\mathbf{h}_{jz} = \begin{bmatrix} h_{jz}(0) & h_{jz}(1) & \cdots & h_{jz}(L_{\mathrm{h}}-1) \end{bmatrix}^T \qquad j = 1, \ldots, J\,, \tag{4.33}$$

where the subscript $z = 1, 2$ refers to its input signal $s_z(n)$. To generate two bright zones in the scenario of Figure 4.32, one microphone per zone must be used, that is, $M_{\mathrm{B}} = M_{\mathrm{D}} = 1$. The first zone is represented by the microphone of the first device ("Device 1") and the second zone is represented by the microphone of the second device ("Device 2").

Particularizing (4.25) for this scenario, the recorded signals are given by expressed as:

$$\begin{aligned} y_1(n) &= \mathbf{c}_{11} * v_1(n) + \mathbf{c}_{12} * v_2(n) \\ y_2(n) &= \mathbf{c}_{22} * v_2(n) + \mathbf{c}_{21} * v_1(n)\,, \end{aligned} \tag{4.34}$$

The reproduced signals $v_1(n)$ and $v_2(n)$ can be expressed as:

$$\begin{aligned} v_1(n) &= \mathbf{h}_{11} * s_1(n) + \mathbf{h}_{12} * s_2(n) \\ v_2(n) &= \mathbf{h}_{21} * s_1(n) + \mathbf{h}_{22} * s_2(n)\,, \end{aligned} \tag{4.35}$$

where $\mathbf{h}_{jz}$ correspond to the filter of the j^{th} speaker when system is solved using $\mathbf{d}_z(k)$. These filters are estimated through (4.31), leading to:

$$\mathbf{H}(k) = \left(\mathbf{C}^H(k)\mathbf{C}(k) + \beta\mathbf{I}\right)^{-1} \mathbf{C}^H(k)\mathbf{D}(k)\,. \tag{4.36}$$

As it was stated above, different objectives stated through $\mathbf{d}(k)$ result in different sound perceptions at each zone. For this reason, in the following

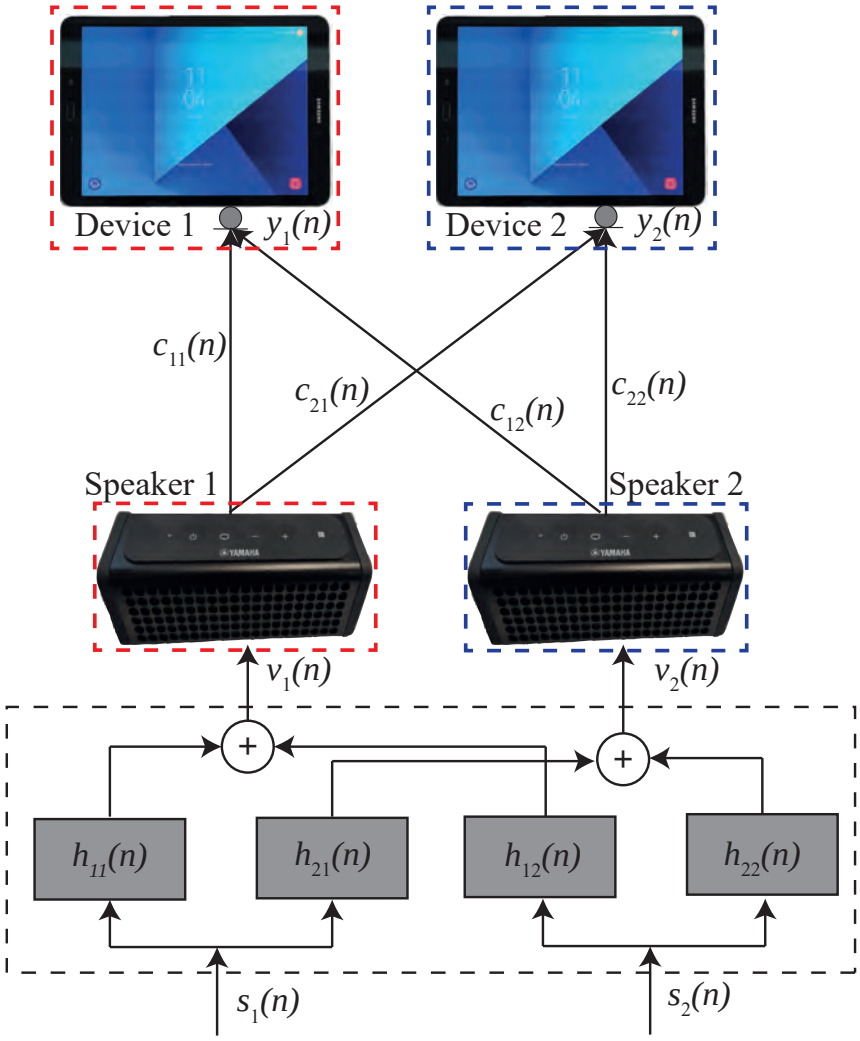


Figure 4.32. Implementation of a PSZ system over a two-node ASN of commercial devices.

paragraphs we define three different audio applications characterized by their own objective matrix $\mathbf{D}(k)$:

-**CrossTalk Cancellation Application (XTC)**. It aims to cancel the contributions of the cross paths without altering the direct paths [211].

Therefore, this application aims to match the pressure captured by the microphones with its corresponding audio signals propagated by their direct paths, such that:

$$\begin{aligned} y_1(n) &\approx \mathbf{c}_{11} * s_1(n) \\ y_2(n) &\approx \mathbf{c}_{22} * s_2(n)\,. \end{aligned} \tag{4.37}$$

As a result, the cross contributions are reduced as much as possible. For this purpose, the objective matrix $\mathbf{D}(k)$ in (4.36) is denoted by $\mathbf{D}_0(k)$ and defined as

$$\mathbf{D}_0(k) = \begin{bmatrix} C_{11}(k)e^{-j2\pi k\frac{\mathrm{L_h}/2}{N_{\mathrm{FFT}}}} & 0 \\ 0 & C_{22}(k)e^{-j2\pi k\frac{\mathrm{L_h}/2}{N_{\mathrm{FFT}}}} \end{bmatrix}, \tag{4.38}$$

where the electro-acoustic paths have been delayed $L_\mathrm{h}/2$ samples to assure causality [208].

-**Pressure Matching Application (PM)**. The main objective of the pressure matching algorithm is to generate a desired sound field in the bright zone [215]. That allow to us to render the audio signals $s_1(n)$ and $s_2(n)$ to their corresponding bright zones as if they would have been originated from a particular speaker. To this end, the objective matrix $\mathbf{D}(k)$ in (4.36) that will be used in this application are denoted by $\mathbf{D}_1(k)$ when the objective relates to the first speaker:

$$\mathbf{D}_1(k) = \begin{bmatrix} C_{11}(k)e^{-j2\pi k\frac{L_\mathrm{h}/2}{N_{\mathrm{FFT}}}} & 0 \\ 0 & C_{21}(k)e^{-j2\pi k\frac{L_\mathrm{h}/2}{N_{\mathrm{FFT}}}} \end{bmatrix}, \tag{4.39}$$

and $\mathbf{D}_2(k)$ when the objective relates to the second speaker:

$$\mathbf{D}_2(k) = \begin{bmatrix} C_{12}(k)e^{-j2\pi k\frac{L_\mathrm{h}/2}{N_{\mathrm{FFT}}}} & 0 \\ 0 & C_{22}(k)e^{-j2\pi k\frac{L_\mathrm{h}/2}{N_{\mathrm{FFT}}}} \end{bmatrix}. \tag{4.40}$$

4.5.3 Implementation of the PSZ applications

As seen before, the PSZ applications only require the knowledge of the involved room impulse responses $\mathbf{c}_{ij}$ to estimate the filters $\mathbf{h}_{jz}$ and then obtain their output signals $v_j(n)$. In order to use the ALC procedure with the PSZ applications to assure the best possible synchronization of the reproduced signals $v_j(n)$, we can observe at what stage of the ALC procedure shown in Figure 4.20 the estimated RIRs are available. It can be appreciated that the estimation of the RIRs is carried out in the "Audio Processing" stage, after the recording of the double sweep signal. Therefore, the "Audio Processing" stage will include now the processing needed to compute the filters and their output signals. To this end, all the estimated RIRs $\mathbf{c}_{ij}$ are started at a same time given by the minimum of the four audio latencies T_{AL} (4.7) minus 25 ms, that is, the common position to start trimming the estimated RIRs is:

$$p_{\min} = (\min(p_{\mathrm{AL},ij})) - 0.025 f_{\mathrm{s}} \qquad i = 1, 2, j = 1, 2\,, \tag{4.41}$$

where $p_{\min}$ is expressed in samples.

Additionally, T_{AL} must also be considered to set the common length of the RIRs involved, L_{c}, since it must be large enough to include most of the energy content of all impulse responses, which means that all possible values of p_{AL} must be included in L_{c}. As it is illustrated in Figure 4.7, the difference between the average audio latency $\overline{T_{\mathrm{AL}}}$ of different acoustic nodes can be higher than 300 ms. Since these results show averages, even bigger differences can be measured. From experimental results, we have selected L_{c} to last 500 ms as a compromise between the computational cost and the accurate estimation of the RIRs. For the sake of simplicity, the length of the filters have been selected as $L_{\mathrm{h}} = L_{\mathrm{c}}$. Additionally L_{c} has been rounded up to the nearest power of two and the length of the FFT is given by $N_{\mathrm{FFT}} = 2L_{\mathrm{c}} > L_{\mathrm{c}} + L_{\mathrm{h}} - 1$.

Therefore, the "Audio Processing" stage of Figure 4.20 has been extended to include PSZ applications by carrying out the following steps:

1. Estimate the RIRs $\mathbf{c}_{ij}$ by means of the method explained in Section 4.2.2.

2. Obtain the $p_{\mathrm{AL},ij}$ of each electro-acoustic path and exchange them between the nodes.
3. Estimate $p_{\min}$ (4.41).
4. Trim the impulse responses according to the values of $p_{\min}$ and L_{c}.
5. Exchange the corresponding impulse responses $\mathbf{c}_{ij}$ between nodes.
6. Estimate the corresponding filters $\mathbf{h}_{jz}$ according to the selected PSZ application by means of (4.36) and the corresponding objective matrix $\mathbf{D}_0(k)$ (4.38), $\mathbf{D}_1(k)$ (4.39) or $\mathbf{D}_2(k)$ (4.40).
7. Generate the output signals $v_j(n)$ (4.35).

4.5.4 Experimental validation

An experimental study of the performance of the PSZ applications according to the synchronization between nodes is performed in this section. For this study, the real scenario shown in Figure 4.22 (that is represented in the scheme of Figure 4.32) is used, where the distance between the microphone and the speaker of the same node is 50 cm and the distances between both microphones and both speakers are 30 cm.

Since the PSZ applications aim to generate two bright zones, two different speech signals are selected: $s_1(n)$ is a male speech signal, and $s_2(n)$ is a female speech signal. These signals are filtered by the corresponding filters and their output signals $v_j(n)$ are then synchronously reproduced by the loudspeaker of the corresponding node. Afterwards, the signals $y_i(n)$ are recorded in order to evaluate the performance of the PSZ applications. In addition, the best and the worst case scenarios are presented, where:

- The best case scenario is the computer simulation of the corresponding PSZ application.
- The worst case scenario is obtained by reproducing the clean speech signals, $s_1(n)$ and $s_2(n)$, instead of the output signals $v_1(n)$ and $v_2(n)$ without using the ALC method.

For the performance comparison, we will use the PESQ[2] score [216]. The PESQ value is defined between $[-0.5\,, 4.5]$, where a high score indicates

[2] Perceptual Evaluation of Speech Quality.

a high quality of the speech. To estimate the PESQ value, two signals are required: 1) the original signal, or clean speech, which in our case should be $s_1(n)$ or $s_2(n)$ and 2) the degraded signal, which corresponds to the recorded signal, $y_1(n)$ or $y_2(n)$ respectively. The PESQ is defined for signals sampled at 8000 or 16000 Hz [216], thus all the signals involved have been resampled to 16kHz.

As said before, the PESQ score provides information related to the deterioration suffered by a speech signal with respect to the recorded signal. However, in this case, the comparison of $s_1(n)$ and $s_2(n)$ with $y_1(n)$ and $y_2(n)$ respectively will be unfair since $y_1(n)$ and $y_2(n)$ include the effect of the selected RIR, as it can be seen from the definitions of $\mathbf{D}_0(k)$ (4.38), $\mathbf{D}_1(k)$ (4.39) or $\mathbf{D}_2(k)$ (4.40). In order to analyze the performance of the PSZ applications exclusively related to the synchronization between nodes, the influence of the electro-acoustic path between microphones and speakers should not be considered. To this end, the original signals that are used in the PESQ procedure are the objective signals designed for each PSZ application (pressure matching, PM, or crosstalk, XTC) according to (4.38), (4.39) and (4.40) respectively:

$$\begin{aligned} x_{\mathrm{or},i}^{\mathrm{XTC}}(n) &= \mathbf{c}_{ii} * s_i(n) \\ x_{\mathrm{or},i}^{\mathrm{PM1}}(n) &= \mathbf{c}_{i1} * s_i(n) \qquad i = 1, 2\,, \\ x_{\mathrm{or},i}^{\mathrm{PM2}}(n) &= \mathbf{c}_{i2} * s_i(n) \end{aligned} \tag{4.42}$$

where $x_{\mathrm{or},i}(n)$ refers to the original signal introduced in the PESQ procedure and the superscript refers to the PSZ application where PM1 and PM2 refer to the pressure matching (PM) application taking loudspeakers 1 and 2 as references, respectively.

The same speaker and mobile device combinations used in Section 4.4.3 are used here to analyze the performance of the PSZ applications thus, there will be 16 different ASNs obtained from the combinations exhibited in Table 4.6. A sample rate, $f_{\mathrm{s}} = 44100$ Hz, and a buffer data length, $L_{\mathrm{B}} = \frac{L_{\mathrm{B,ref}}}{2}$, are used, and the length of the impulse responses are set to $L_{\mathrm{c}} = 32768$ samples.

Our study has been carried out through the following steps:

1. Select one combination from Table 4.6 .

2. Perform the ALC method of Figure 4.20, but including in the "Audio Processing" stage the computation of the filters $\mathbf{H}(k)$ (4.36) and the filter output signals $v_j(n)$ (4.35).

3. Emit signals, $v_1(n)$ and $v_2(n)$ through the corresponding loudspeakers. Record the received signals at the corresponding microphones. These will be the degraded signals for the computation of the PESQ.

4. Obtain the original signals needed for the PESQ and shown in (4.42).

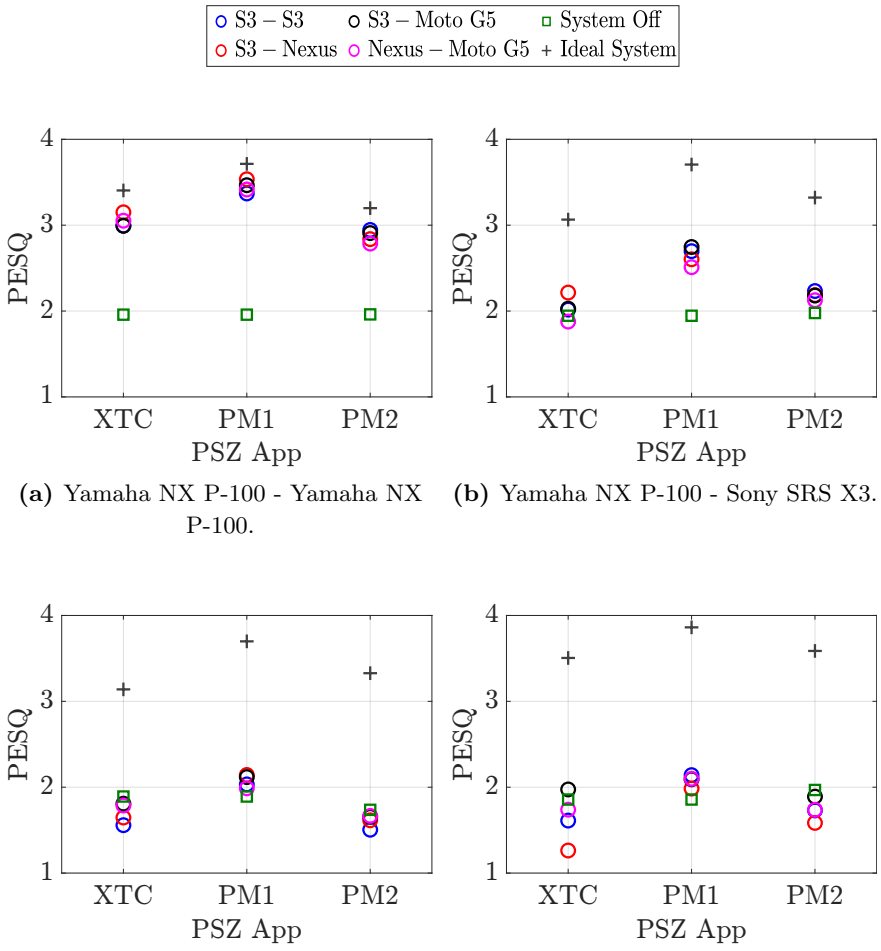
5. Obtain the worst case scenario by reproducing and recording the speech signals $s_1(n)$ and $s_2(n)$ without any latency compensation. The best case scenario is computed at the devices, although their resulting signals are simulated, not recorded.

6. Obtain the PESQ parameter comparing the original and degraded signals for the three PSZ applications and the best and worst case scenarios.

7. Return to step 1 to select other combination.

The results of this study are depicted in Figures 4.33 and 4.34, where the first one corresponds to the PESQ score of the male speech (the first node) and the second one refers to the PESQ score of the female speech (the second node). The results are classified by the selected loudspeaker combination, as the caption of each sub-figure shows. Each sub-figure represents the PESQ score for the three PSZ applications. The best and worst case scenarios are represented by the "+" and the "□" markers and denoted by "Ideal System" and "System off" respectively, while each mobile device combination is represented by the "○" marker using a different color per combination.

According to the results shown in Figures 4.33 and 4.34, the PESQ score presents a similar value to that achieved in the best case scenario when the Yamaha NX P-100 speaker is used in both acoustic nodes, independently of the PSZ application, although the best combination of mobile devices may differ. In particular, for the male speech signal, the best case scenario presents a PESQ score of 3.4, 3.7 and 3.1 for the XTC, PM1 and PM2 applications respectively, while the PESQ score for the real cases ranges between $[3\,,3.1]$, $[3.3\,,3.5]$ and $[2.8\,,2.9]$ for each application respectively. In contrast, for the female speech signal, best case scenario presents



(a) Yamaha NX P-100 - Yamaha NX P-100.

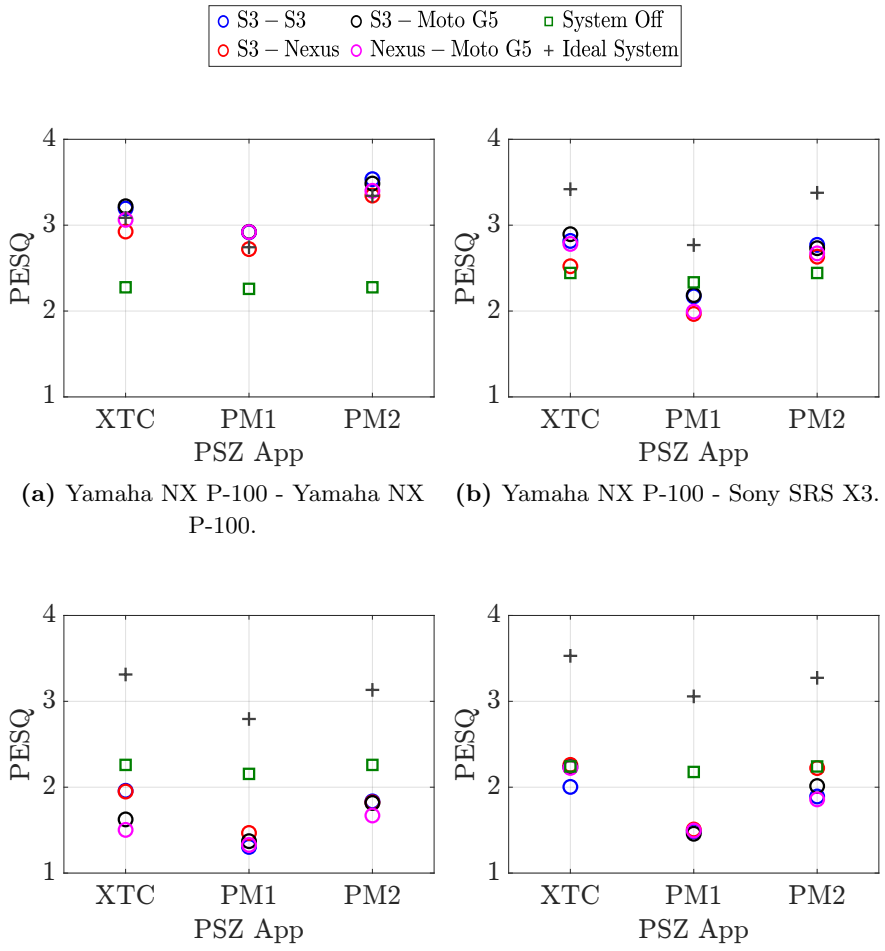
(b) Yamaha NX P-100 - Sony SRS X3.

(c) Yamaha NX P-100 - JBL Flip 2.

(d) Yamaha NX P-100 - JBL Charge 4.

Figure 4.33. The PESQ score for the male speech signal.

a PESQ score of 3, 2.7 and 3.3, while the PESQ score for the real cases ranges between [2.9 , 3.2], [2.7 , 2.9] and [3.3 , 3.5] for each application respectively. This good value of the PESQ score can be explained by the small offset between the acoustic nodes (shown in Figure 4.23) achieved by the ALC procedure. As a result, the performance of the PSZ applications in a real scenario is very similar to the performance of an ideal scenario.



(a) Yamaha NX P-100 - Yamaha NX P-100.

(b) Yamaha NX P-100 - Sony SRS X3.

(c) Yamaha NX P-100 - JBL Flip 2.

(d) Yamaha NX P-100 - JBL Charge 4.

Figure 4.34. The PESQ score for the female speech signal.

When the JBL speakers are used (JBL Flip 2 or JBL Charge 4), certain combinations show that the PESQ score is worse than the worst case scenario where no synchronization and no filtering is performed. In Figure 4.33c, the best case scenario presents a PESQ score of 3.1, 3.7 and 3.3 and the worst case presents a PESQ score approximately of 2 for the XTC, PM1 and PM2 applications respectively, while the PESQ score for the real cases ranges between [1.5 , 1.8], [2 , 2.1] and [1.5 , 1.6]. In contrast,

in Figure 4.34a, the best case scenario presents a PESQ score of 3.3, 2.8 and 3.1 and the worst case presents a PESQ score approximately of 2.2, while the PESQ score for the real cases ranges between [1.5 , 1.9], [1.3 , 1.4] and 1.8 for each application respectively. When JBL Charge 4 is used, see similar performance can be appreciated for both speech signals. This low value is caused because of the high offset between the acoustic nodes and unstable behavior of the offset (shown in Figures 4.25 and 4.26). Therefore, for the JBL speakers, the performance of the PSZ applications is reduced significantly regardless the mobile device combination.

The results in Figures 4.33 and 4.34 show that the performance of the PSZ applications for the speaker combination Yamaha NX P-100 - Sony SRS X3 is worse than expected given the small Δp obtained in Figure 4.24. In this case, the PESQ values of the real cases are significant different than the PESQ values of the best case scenario. In particular, for the male speech signal, the PESQ values of real scenario are included between [2 , 2.2], [2.6 , 2.7] and [2.1 , 2.2], while the PESQ values of the best case scenario are 3, 3.6 and 3.3 for the XTC, PM1 and PM2 applications respectively. In contrast, for the female speech, the PESQ values of the real cases are included between [2.5 , 2.8], [1.9 , 2.1] and [2.6 , 2.7] while the PESQ values of the best case scenario are 3.4, 2.7 and 3.3 for the XTC, PM1 and PM2 applications respectively. This impairment could be caused because of some kind of equalization of the Sony SRS X3 speaker, that avoids the cancellation of the interfering signals.

Additionally, the results show that, for any combination, the PESQ score in the male speech signal is higher than the PESQ score in the female speech signal for the PM1 application, while an opposite behavior is appreciated for the PM2 application. This outcome is caused because of the objective matrix $\mathbf{D}(k)$ used in each application. In the PM1 application, the direct path $\mathbf{c}_{11}$ is used for the male speech signal, while the cross path $\mathbf{c}_{21}$ is used for the female speech signal. In the PM2 application, the direct path $\mathbf{c}_{22}$ and the cross path $\mathbf{c}_{12}$ are used for the female speech signal and the male speech signal, respectively. Therefore, the PM1 application considers in the first zone (male speech signal) a greater energy than in the second zone (female speech signal), while PM2 considers the opposite. Regarding the XTC application, both male and female speech signals show similar PESQ score, an expected behavior since in this application both direct paths, $\mathbf{c}_{11}$ and $\mathbf{c}_{22}$, are used in the objective matrix $\mathbf{D}(k)$ (see (4.38)).

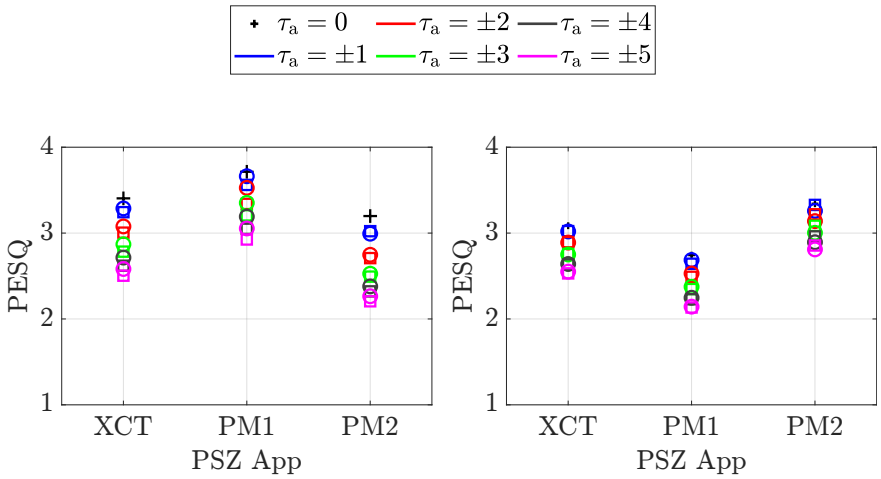
These results have shown that the synchronization between nodes is critical for the implementation of high quality audio applications over wireless ASNs. In order to show the dependence of the audio applications performance with the synchronization between nodes, a set of degraded signals have been simulated such that a varying delay has been added to one of the filtered signals. Therefore, expression (4.34) has been rewritten as:

$$\begin{aligned} y_1(n) &= \mathbf{c}_{11} * v_1(n-\tau_\mathrm{a}) + \mathbf{c}_{12} * v_2(n) \\ y_2(n) &= \mathbf{c}_{22} * v_2(n) + \mathbf{c}_{21} * v_1(n-\tau_\mathrm{a})\,, \end{aligned} \tag{4.43}$$

where τ_a is the additional delay, expressed in samples, whose values have been set to $\tau_\mathrm{a} \in \{\pm 5, \pm 4, \pm 3, \pm 2, \pm 1, 0\}$.

The results of this study have been obtained using the Yamaha NX P-100 speaker on both nodes (as in the Figures 4.33a and 4.34a) and the Samsung Galaxy S3 in both nodes as well. Since all the degraded signals have been simulated, in order to obtain a correct PESQ score, the original signals used for obtain the PESQ score have been also simulated (see (4.42)). Figure 4.35a represents the PESQ score of the signal of the first node (the male speech) and Figure 4.35b represents the PESQ score of the signal of the second node (the female speech). Similar to the previous results, the ideal case ($\tau_\mathrm{a} = 0$) has been represented through the "+" marker. In addition, the "□" marker represents the positive delays and the "○" marker represents the negative delays. Therefore, the delays differ in their colors and positive and negative delays differ in their markers. The rest of features are identical to the results shown in Figures 4.33 and 4.34.

As Figure 4.35 shows, the PESQ value decreases as $|\tau_\mathrm{a}|$ increases, which is an expected result since the lack of synchronism mainly affects to the cancellation of the interference. However, no significant differences can be appreciated in the performance of the PSZ applications regarding the sign of the delay, since the PESQ obtained is quite the same for $\pm\tau_\mathrm{a}$. A comparison between these PESQ values and the real PESQ values illustrated Figures 4.33a and 4.34a is shown in Table 4.7, where the value of ΔPESQ is represented according to τ_a. The value of ΔPESQ represents difference between the averaged PESQ values of the three applications in Figures 4.33a and 4.34a (using the Samsung Galaxy S3 in both nodes) and the averaged PESQ values of the three applications in Figure 4.35 for $\tau_\mathrm{a} \in \{0\ 1\ 2\ 3\ 4\ 5\}$.



(a) Ideal PESQ of the male speech. **(b)** Ideal PESQ of the female speech.

Figure 4.35. Ideal PESQ with different delays between the acoustic nodes.

	$\tau_a = 0$	$\tau_a = 1$	$\tau_a = 2$	$\tau_a = 3$	$\tau_a = 4$	$\tau_a = 5$
Male Speech	-0.336	-0.172	0.085	0.275	0.438	0.557
Female Speech	0.158	0.217	0.347	0.486	0.607	0.71

Table 4.7. ΔPESQ for different values of τ_a.

In Table 4.7, only positive values of τ_a have been considered since, as it was stated before, the PESQ score is quite the same for $\pm\tau_a$. The results for the male speech signal show that the ΔPESQ values are negatives for $\tau_a < 2$ samples, meaning the simulated system presents a performance better than the real system, that is, an expected behavior. However from $\tau_a \geq 2$ samples the values are positive, that is, the real scenario presents a better performance. In contrast, for the female speech, the ΔPESQ values are positive from the start ($\tau_a = 0$ samples). According to the values of ΔPESQ, a desynchronization of a few of samples results in a

lower PESQ score, causing a higher ΔPESQ value. Consequently higher values of τ_a than those shown in Table 4.7, such as $\tau_a = 10$ samples (225 μs) or $\tau_a = 20$ samples (450 μs), can reduce significantly the performance of the PSZ applications, as shown in Figures 4.33 and 4.34 when JBL speakers are used.

4.5.5 Conclusions

The synchronization methods explained in sections 4.3 and 4.4 have been used in order to analyze the performance of two PSZ applications. The PSZ applications have been tested in a 2x2 wireless ASN and the PESQ score (ITU standard P.862 [216]) of the recorded speech signals have been computed.

The experiment consisted in two devices connected through a Wi-Fi link between them and connected each one through a Bluetooth link to a wireless speaker. Then, using the ALC method, the filtered signals, $v_1(n)$ and $v_2(n)$ in (4.35) are reproduced and recorded. The results showed that the PESQ score is higher for the combinations that include the Yamaha NX P-100 speakers and lower for those using the JBL speakers, thus, the speech quality is higher as the synchronization between nodes is better, as shown in Figures 4.33 and 4.34. In contrast, the Yamaha NX P-100 - Sony SRS X3 speaker combination, that exhibited a similar synchronization error (Δp) to that of the Yamaha NX P-100 - Yamaha NX P-100 speaker combination, presented a low PESQ score. This behavior is assumed to be caused by the internal signal processing performed by the Sony SRS X3 speaker.

In order to measure more accurately the dependency of the PSZ performance with the lack of synchronization, a computer simulation of the experiment using the real acoustic paths $\mathbf{c}_{ij}$ was performed, where different delays, τ_a, were used in order to simulate the lack of synchronization between nodes. The results have shown that the PESQ score, thus, the perceived quality of the resulting speech, is seriously affected and it gets worse as $|\tau_a|$ is higher.

4.6 Conclusions

In this chapter, a wireless ASN composed by Android commercial devices and Bluetooth speakers has been presented and its performance has been studied. Within the ASN each mobile device is connected to one speaker through a Bluetooth link and the mobile devices are connected between them through a Wi-Fi link. Based on this scenario, the main objective is to generate an environment where any audio application can be carried out over in the different acoustic nodes of the ASN with the lowest synchronization mismatch between nodes. To this end, a deep analysis about the synchronization problem has been along this chapter.

As an initial stage, the audio latency, denoted as T_{AL}, has been introduced. This parameter provides temporary information about the audio processes implemented in the acoustic nodes. After a detailed description of T_{AL} and a brief explanation about its estimation, an experiment to analyze the behavior of T_{AL} has been carried out. Different combinations of mobile devices and wireless speakers have been used and placed at different distances from each other. The results of this experiment showed that the averaged values of T_{AL} range from 350 ms to 650 ms, concluding that the value of T_{AL} depends highly on the processing of the mobile device and the speaker, that is T_{DP} and T_{SP} respectively.

The synchronization problem has been studied separately throughout this chapter to facilitate its analysis. Initially, the difference between the time clock of the devices, denoted as T_O, has been studied. This difference causes that two (or more) devices are not able to start an audio task simultaneously. To solve this drawback, the STP method has been introduced (see Figure 4.11) in order to compensate T_{O}. After a detailed description of the STP method, an experimental comparison with the NTP and PTP methods is carried out. The results of this comparison showed that the proposed STP method is the most accurate to estimate T_{O}. Afterwards, in order to evaluate the performance of the STP method in the context of audio applications, a second experiment has been proposed. Two mobile devices connected to an audio analyzer are used in this experiment to obtain the difference between their respective audio latencies, Δ_{T}. The results showed that the STP method does not solve the synchronization problem due to the significant differences in hardware and software of the mobile devices that cause audio latencies to be different, and thus $\Delta_{\mathrm{T}} \neq 0$.

According to the previous results, due to the difference between T_{AL} of the devices, the audio applications cannot be simultaneously performed unless a precise delay compensation is carried out. In this sense, the audio latency compensation (ALC) method has been presented in this chapter (see Figure 4.20). After a detailed description of the ALC method, an experiment has been carried out in order to evaluate its performance. Similar to the previous experiment, two mobile devices connected between them and connected to a wireless speaker each have been used to obtain the difference between their respective audio latencies in samples, Δp. The results have shown that Δp depends significantly on the processing burden performed by the speakers since Δp is different according to the speakers used, being in some cases very unstable. However, in the cases that Δp presents a low value and a stable behavior, it can be predicted and compensated, resulting in an almost perfect synchronization with a maximum audio latency of 1 sample for a sample rate of 44100 Hz.

Finally, two personal sound zone (PSZ) applications have been implemented in order to analyze the influence of the synchronization mismatch and the benefit of the ALC method on the perceived speech quality at each zone. The results have shown that the PESQ score is higher as the synchronization error between the nodes is lower, whereas low PESQ values are obtained for the speaker combinations that presented a high value and an unstable behavior of Δp. Moreover, the PESQ results obtained for the pair of loudspeakers Yamaha NX P-100 - Sony SRS X3 in both PSZ applications is very low and does not correspond to the high degree of synchronization presented in the ALC experiment. This behavior is assumed to be caused by the audio processing performed by the Sony SRS X3 speaker, which seems to depend on the signal to be reproduced. However, when the Yamaha NX P-100 speaker is used in both nodes, the ALC method demonstrates to increase significantly the synchronization performance for the PSZ applications, obtaining PESQ values similar to the ones obtained in the computer simulation.

Conclusions 5

This chapter summarizes the findings of this work, reviewing the main objectives set out in the introduction chapter. At the beginning of this chapter, a review of the contents of the studies of the dissertation is done to outline the main conclusions drawn from each chapter. The next section presents different recommendations for future research. The final part contains a list of works published during the course of candidature for the doctoral degree. Additionally, institutional acknowledgments supporting this work are given.

5.1 Summary

In this dissertation, we have focused on performing two different groups of sound-field applications using an ASN in order to analyze their performance. The first group considers that environmental sounds (such as ambient noise) negatively affect to the sound emitted by the ASN (such as music), resulting in a worse perception threshold of the music. To overcome this drawback, equalization and psychoacoustic techniques are used in order to improve the perception threshold of the sounds emitted by the ASN. The second group is focused on performing a synchronous reproduction of

all the acoustic nodes of the ASN. Specifically, in this dissertation PSZ applications are implemented, where the main goal is to generate different acoustic zones with a different sensation in each zone through a synchronous reproduction of the nodes of the ASN.

In Chapter 3, an ASN composed by traditional microphones and speakers is considered in order to perform a perceptual audio application. In this dissertation, a perceptual equalization with two different approaches is proposed. However, in a first stage the masking threshold algorithm is explained in detail since it provides the perceptual analysis needed in the equalization application. The explanation of this algorithm is completed with a perceptual test that analyzes the tonality factor (required for the estimation of the masking threshold) calculated by three methods. The results in Figure 3.12 shows that the proposed method provides a more accurate estimation of the masking threshold since its values are closer to those obtained in the perceptual test. Afterwards, the masking threshold has been used to perform the perceptual equalization application with the two different approaches. In the first one, the equalization is performed on the music, while the second one the equalization is performed on the ambient noise using ANC techniques. Each approach is explained step by step through a block diagram to fully understand the process of each equalization. Finally, each approach is evaluated by a perceptual test performed by several participants in order to study the performance of each one. The results in Figures 3.27 and 3.35 show that the proposed solutions are able to increase significantly the perception threshold of the music in presence of ambient noise. Finally, an important fact to highlight is that the second approach (equalization over ambient noise) provides a better performance than the first approach (equalization over music) because the equalization is based on reducing the power level of the noise instead of increasing the power level of the music, avoiding non-linear effects, such as saturation or distortion of the music signal.

In Chapter 4 an ASN composed by commercial Android mobile devices and Bluetooth speakers is proposed to perform a reproduction application, more specifically a PSZ application, simultaneously. To this end, an accurate control of the synchronization between the acoustic nodes of the ASN must be ensured. For this reason, a deep analysis of the synchronization between the acoustic nodes is performed throughout the Chapter 4. As a first step, the audio latency (T_{AL}) parameter is explained in deep, con-

cluding this section with an experimental study where T_{AL} is estimated for different acoustic nodes. The results in Figure 4.7 show that T_{AL} is dependent on the hardware and software of both mobile device and Bluetooth speaker. Next, to synchronize the different acoustic nodes, the synchronization time protocol (STP) is proposed to estimate the clock difference, T_{O}. Additionally, an study is carried out, comparing the STP method with PTP and NTP methods, the most commonly used synchronization methods nowadays. The results in Figure 4.13 demonstrate that the STP method is the most stable in estimating T_{O} and therefore the method used in this dissertation. A second experiment is proposed, where the STP is used to simultaneously reproduce a tone signal by two devices. The results of this experiment, shown in Figure 4.16, conclude that devices cannot perform a simultaneous reproduction because of the difference of T_{AL}. For this reason, a last method is implemented to consider the difference of T_{AL}, called in this dissertation as the audio latency compensation (ALC) method. The ALC method has been evaluated in an experiment using different combinations of mobile devices and speakers. The results from Figures 4.23 to 4.30 show that the achieved synchronization between nodes is higher to that of the previous test. Despite this behavior, a perfect synchronization is not possible because of the processing when the Bluetooth connection is enabled leading to the conclusion that the synchronization depends on the performance of such processing. Finally, a last study has been performed in order to measure the performance of the PSZ application, where synchronization is critical. The results in Figures 4.33 to 4.35, present a better performance in the PSZ application as the synchronization between nodes is more accurate, that is, the perception of the sounds in each zone is improved as the synchronization is higher.

5.2 Future Work

The research produced in this thesis opens up different interesting fronts to be considered for future research. In this case, the main lines of research open are:

- In Chapter 3 a perceptual study has been carried out in order to analyze different tonality methods of the masking threshold algorithm. Since the signals to be tested have been multi-tonal signals

and narrow-band noise signals, an interesting study to be performed is the same perceptual study, but using signals with more spectral content in order to study the masking threshold for signals with a wider bandwidth.

- In Chapter 3 the first perceptual equalizer is performed considering a single-user. However, in order to benefit from all the features provided by an ASN, an equalizer for multiple users should be considered. This study can be based on the preliminary study of the optimal masking pattern for multiple users performed in this dissertation.

- Finally, in Chapter 3, two different equalizers are used to study the perceptual algorithms. However since the equalizers act on different signals, the combination of the two should be considered in order to implement a single equalizer that is able to act on both signals at the same time.

- The ASN used in Chapter 4 was focused on a two-node network to minimize the complexity of the algorithms. In practice, however, larger ASN are usually considered in order to increase the performance of the audio applications. Therefore, an interesting issue to evaluate would be to perform same studies used in this dissertation, but increasing the number of nodes of the ASN in order to study the scalability of the synchronization methods and the performance of the PSZ applications.

- The ASN used in Chapter 4 only uses a Bluetooth connection to connect the mobile device and the speaker. But, in real scenarios, a Wi-Fi connection usually presents a better approach since this connection is faster. For this reason, a new approach can be proposed using Wi-Fi speakers. However, since the current Wi-Fi speakers only can be used with proprietary software, the use of a Rasberry Phi or Arduino is proposed. Assuming that this additional device can be connected through a wire to a speaker, a Wi-Fi speaker can be implemented. Thus, through the Wi-Fi connection, the mobile device will be able to send the audio signal to the Rasberry Phi or Arduino in order to reproduce the signal through the speaker.

5.3 List of Publications

A list of published work produced during the course of candidature for the degree is presented in what follows.

International Journal Papers

- **J. Estreder**, G. Piñero, M. de Diego, J. Rämö, V. Välimäki, "Improved Aures Tonality Model for the Perceived Masking Threshold of Complex Sounds". *Journal of Sound and Vibration, Elsevier.* Submitted for review.

Peer-reviewed non-ISI Journals

- C. Antoñanzas **J. Estreder**, G. Piñero, M. Ferrer, M. de Diego, A. González, "A complete simulation tool for sound control applications over wireless acoustic sensor networks". *Waves*, vol. 7, pp.47-56, 2015. ISSN: 1889-8297

International Journal Conferences

- G. Piñero, **J. Estreder**, F. J. Martínez-Zaldívar, M. Ferrer, M. de Diego, "Sound-field reproduction system over a two-node acoustic network of mobile devices". *IEEE 2nd World Forum on Internet of Things (WF-IoT).* Milan, Italy, 14–16 December 2015. `https://doi.org/10.1109/WF-IoT.2015.7389131`

- G. Piñero, **J. Estreder**, F. J. Martínez-Zaldívar, M. de Diego, M. Ferrer, "Reshaping of Room Impulse Responses over Wireless Acoustic Networks". *AES International Conference on Sound Field Control*, Guildford, United Kingdom, 18–20 July 2016. `http://www.aes.org/e-lib/browse.cfm?elib=18320`

- **J. Estreder**, G. Piñero, F. Aguirre, M. de Diego, A. González, "On Perceptual Audio Equalization for Multiple Users in Presence of Am-

bient Noise". *IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, Sheffield, United Kingdom, 8–11 July 2018. https://doi.org/10.1109/SAM.2018.8448591

- **J. Estreder**, G. Piñero, M. Ferrer, M. de Diego, A. González, "Perceptual Active Equalization of Multi-Frequency Noise". *18th International Conference on Signal Processing and Multimedia Applications (SIGMAP)*, Online Streaming, 6–8 July 2021. https://doi.org/10.5220/0010648100390047

National Journal Conferences

- C. Antoñanzas, **J. Estreder**, M. Ferrer, A. González, "Simulador de redes de sensores acústicos para aplicaciones de control de campo sonoro". *Proceedings of the 46th Congreso Español de Acústica (SEA - TECNIACÚSTICA 2015)*, Valencia, Spain, 21–23 October 2015. http://www.sea-acustica.es/fileadmin/publicaciones/AFS-1_009.pdf

- **J. Estreder**, C. Antoñanzas, G. Piñero, M. de Diego, "Gestor de aplicaciones de procesado de sonido sobre Android para la creación de redes de nodos acústicos con dispositivos comerciales". *Proceedings of the 46th Congreso Español de Acústica (SEA - TECNIACÚSTICA 2015)*, Valencia, Spain, 21–23 October 2015. http://www.sea-acustica.es/fileadmin/publicaciones/AFS-1_008_01.pdf

5.4 Institutional Acknowledgements

This work has received financial support of the following projects:

- SSPRESING: Smart Sound Processing for the Digital Living (Reference: TEC2015-67387-C4-1-R. Entity: Ministerio de Economia y Empresa. Spain).

- FPI: Ayudas para contratos predoctorales para la formación de doctores (Reference: BES-2016-077899. Entity: Agencia Estatal de Investigación. Spain).

- DANCE: Dynamic Acoustic Networks for Changing Environments (Reference: RTI2018-098085-B-C41-AR. Entity: Agencia Estatal de Investigación. Spain)

- DNOISE: Distributed Network of Active Noise Equalizers for Multi-User Sound Control (Reference: H2020-FETOPEN-4-2016-2017. Entity: I+D Colaborativa competitiva. Comisión de las comunidades europea).

Appendices

A Android Application

An Android application has been implemented with the purpose of providing an interactive tool to perform the PSZ systems explained in Section 4.5. The app presents several graphical interfaces related to the main tasks of the PSZ applications. Figure A.1 shows the structure of the Android application together with the connections between the graphical interfaces. The interfaces are assigned the following tasks:

- Main Interface (MI): It addresses the clock synchronization problem by using the proposed method (see Section 4.3.2).
- Bluetooth Interface (BI): It generates an acoustic node through the connection between the mobile device and speaker. This task is optional, providing for the case that a former Bluetooth connection between device and loudspeakers exists.
- Network Interface (NI): It generates the network through the connection between the acoustic node and the router device.
- Sound Interface (SI): It performs the different PSZ applications (see Section 4.5) using the ALC method, explained in Section 4.4.2.

Since each task is performed by a different interface, each one can be reached and configured independently from the rest. However, the Android app presents two internal restrictions in order to create the PSZ applications without errors. Regarding to the first restriction, the Android application will not allow to address the clock synchronization problem unless the acoustic node is connected to a Wi-Fi network. As for the second restriction, the Android application will not allow to carry out the PSZ applications unless the clock synchronization problem has been addressed. Therefore, the Android application will guide the user through the different interfaces to correctly perform the audio applications. For this reason, in the following, a briefly description of how to deal with each interface is given.

Main Interface (MI)

This is the first interface that appears when the Android application is called. Figure A.2 shows its appearance, where the rest of interfaces can

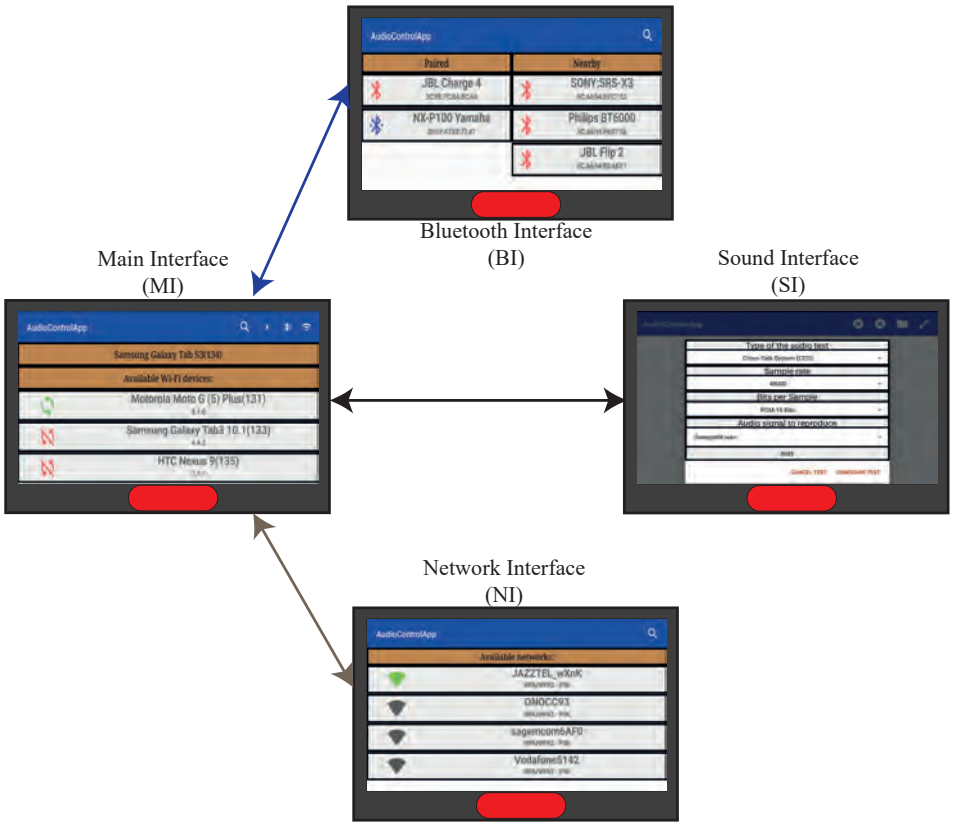


Figure A.1. Structure of the Android application.

be accessed by pressing the corresponding button located at its upper-right corner. This interface is responsible for addressing the clock synchronization problem (deeply studied in Section 4.3) once the device has been connected to a router device through the Network Interface. Therefore, the first step that must be carried out is to access the Network Interface to connect to the underlying Wi-Fi network.

Assuming that a Wi-Fi connection is already done, the magnifying glass icon allows the node to search for other nodes in the network and to perform the clock synchronization between them. It has to be noticed that the Android OS does not provide any direct method to perform this kind of device search, thus a new procedure has been developed for this purpose in

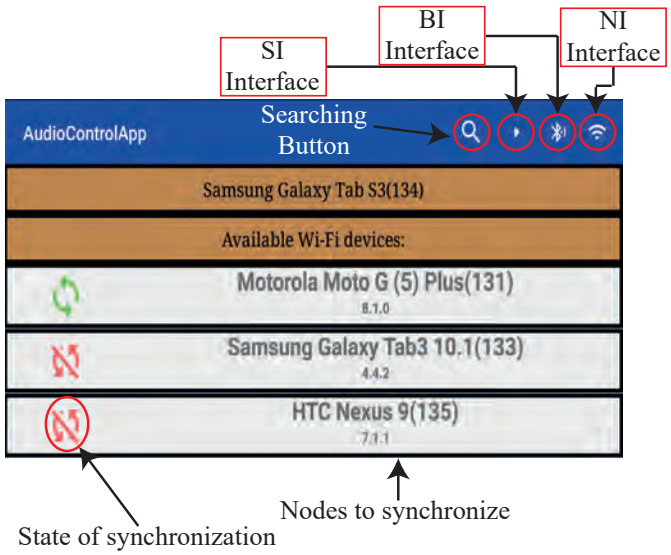


Figure A.2. Main Interface.

this dissertation, and it is explained in Appendix B. Once a node is found, some properties of the node are shown, as indicated in Figure A.2, as the name and the Android version of the device. In order to identify the nodes that are synchronized, a green icon at the left indicates that clock synchronization has been addressed with that node, while a red icon indicates the opposite. By pressing any device with a red icon, the clock synchronization will be addressed automatically for that specific acoustic node. Once the Wi-Fi network has been successfully created, the Sound Interface (SI) can be accessed in order to implement different audio applications.

Bluetooth Interface (BI)

This interface is shown in Figure A.3 and can be accessed in order to pair a wireless loudspeaker to the acoustic node. The connection between the mobile device and the wireless speaker is carried out through the A2DP profile [74], which allows for the transmission of audio streams between devices.

As Figure A.3 shows, the interface presents a list where two different groups of loudspeakers are identified, “Paired” and “Nearby”. In the Blue-

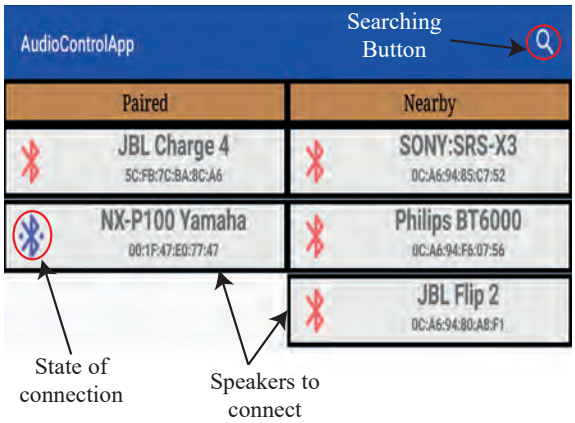


Figure A.3. Bluetooth Interface.

tooth protocol, a "pairing" process is done before the connection process. This pairing process is used to ensure a secure communication between the device and the speaker [217]. Therefore, the BI distinguishes between the speakers that have carried out the pairing process (the "Paired" group) and those that have not (the "Nearby" group).

Same as in Figure A.2, the magnifying glass icon is used to search for Bluetooth speakers, but in this case the searching procedure is provided by the Android OS as described in [218]. Once a speaker is found, its name and its media access control (MAC) are shown. Finally, the connection to the selected speaker is carried out by pressing on its name in the interface, where in a few moments its state will change from no connected (red Bluetooth icon) to connected (blue Bluetooth icon).

Network Interface (NI)

Figure A.4 shows the NI interface, whose main objective is to connect the acoustic node to a Wi-Fi network through an existing router device. According to Figure A.4, once the NI is accessed, a list of available networks is visualized. Same as BI, the search to find different network is performed through the magnifying glass icon using the method specified in [219]. When a network is found, the name of the network and the encryption method [220, 221] are shown. Identically to MI and BI, the connection is performed by pressing the name of the network. To identify

the network connected to the node, a green icon indicates that node is connected to that network, while a gray color indicates no connection.

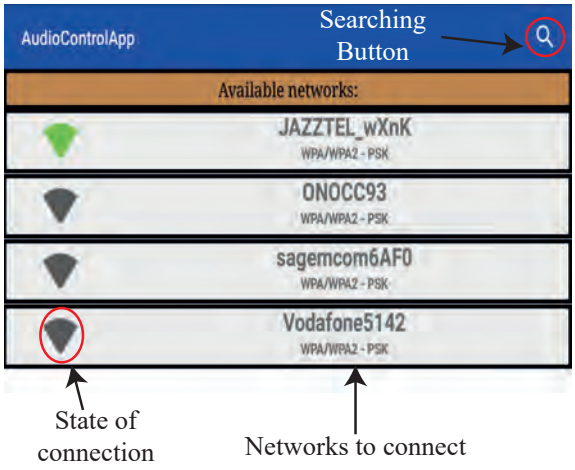
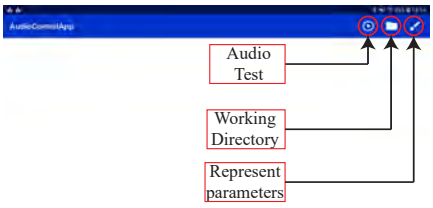


Figure A.4. Network Interface.

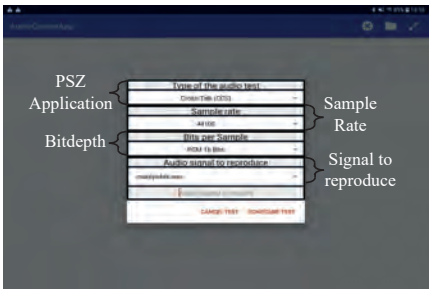
Sound Interface (SI)

This interface is shown in Figure A.5 and it is the responsible for performing the different audio applications presented in Section 4.5 through the ALC method of Section 4.4. The interface is able to configure and perform an audio test, select the directory to save the audio information achieved from the audio test or even to represent the most important information of the test, as Figure A.5 shows.

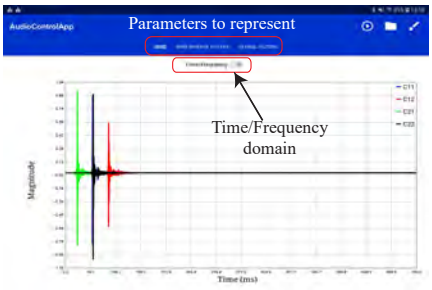
This interface starts with the blank screen showed in Figure A.5a since no information can be represented yet. By pressing the button labeled as "Audio Test", the interface will show a box with different fields, as Figure A.5b shows. In this box, the configuration of the audio test, such as the type of the PSZ application (see Section 4.5) or sample rate, is selected. The audio test is configured in both of them using the connection between the nodes. Once the audio test of the selected PSZ application is finished, a graphical representation of the most important parameters pressing the "paint brush" icon can be carried out. The plot can depict the impulse responses of the ASN acoustic paths (shown in Figure A.5c) and the filters of the PSZ application and their corresponding frequency



(a) Default screen.



(b) Audio Test configuration.



(c) Representation Mode screen.

Figure A.5. Sound Interface.

responses. Finally, the selection of the working directory is enable through the "folder" icon of the SI. Since the audio signals to reproduce and record have to be located in the memory of the mobile device, this information is necessary.

B Procedure to search acoustic nodes inside the network

The connections between the acoustic nodes are one of the most important procures to carried out in an ASN in order to implement different audio applications. But, as opposed to the Bluetooth search or the network search (see Appendix A), the Android OS does not provide any direct procedure to search for devices on a network. As a result, in this dissertation we have designed a method to perform this search. Since this method is used to search mobile devices (or acoustic nodes), we have called it the Node Network Searching (NNS) method.

The Wi-Fi connections between devices only require two parameters: 1) the connection port and 2) the IP address of the device to be connected with. The first parameter is predetermined by the Android application, that is, all the acoustic nodes know the connection port. The IP address, which is different for different devices of the same network can be introduced in the application by the user, but in order to automate the connections, the NNS is proposed.

The NNS method uses the "multicast" communications that allows to send information from one or multiple devices to multiple devices without knowing the specific IP address of each device. Therefore a device can communicate with a group of devices in order to obtain information from them, such as their IP. To accomplish this objective, first at all a group must be generated. The groups are defined through a special IP address (included in the range between [224.0.0.0 - 239.255.255.255]). In the case of the NNS method, the IP address 224.1.2.3 has been chosen to generate the group. Once the group has been created, all the acoustic nodes join the group in order to receive information from any device of to the group.

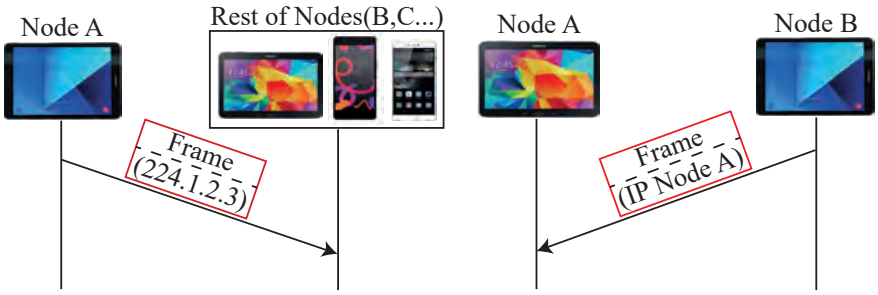
Since the NNS method is implemented in the Main Interface of the application (see Appendix A), it is designed in order to provide a human-readable information. As a result, in addition to the IP address, the features shown in Figure B.1 are exchanged between the nodes.

According to Figure B.1, the frame is divided into three fields. The first field is the name of the mobile device (i.e. "Samsung Galaxy Tab 3"), together with the last octet of its IP address in order to avoid confusions when identical devices are found. The second field is the IP address of the

Figure B.1. Frame sent by the nodes.

specific device. Finally, the third field is additional information, such as the Android version.

Assuming all the nodes have joined the specific group, defined by the address 224.1.2.3, the NNS method is carried out as Figure B.2 shows. First, a specific node, denoted as "Node A", sends the frame of Figure B.1 to the IP address of the group. The rest of the nodes (B, C, ...) receive that frame and each one sends the frame of Figure B.1 with their own information to the Node A, as Figure B.2b shows. Finally, "Node A" receives as many frames as nodes belong to the group, thus obtaining all the IP addresses of the ASN.



(a) First step. **(b)** Second step.

Figure B.2. The NNS method.

The multicast communications are based on the UDP protocol, which means that the information exchanged between the nodes can be lost. As a result, in order to increase robustness against network errors, the frame shown in Figure B.2a is sent 15 times at intervals of 1 second, while the frame shown in Figure B.2b is sent 5 times.

C Optimal Buffer Length for Android acoustic nodes

The acoustic tests performed in Chapter 4 are designed with an audio configuration that is explained in each test, such as the test performed in Section 4.2.3. This configuration is mainly based on the characterization of the sample rate, f_s, and the buffer size (L_B). Since L_B is a parameter with a large number of options (as opposed to f_s, where the most common options are 44100 or 48000 Hz), its optimal value will be discussed in the following.

The Android OS estimates a buffer length according to the mobile device features. This buffer size has been denoted in this dissertation by $L_{B,ref}$ and it is used to configure the recording and playing programming objects. After this initial configuration, the playing and recording operations can use a different size than $L_{B,ref}$. This size is denoted in this dissertation by L_B. Therefore, the Android devices can use two different sizes:

1. The buffer size to create the audio programming objects, $L_{B,ref}$, that is estimated by the Android OS.
2. The buffer size to perform the recording and playing operations, L_B, that can be set to any value.

As Section 4.4.3 shows, T_{AL} (4.3) suffers an offset due to the processing performed by the Bluetooth connection. Since the processing time is also related to the L_B value, as Section 4.2.1 explains, this appendix studies the dependence of the offset of T_{AL} with the value of L_B for a single acoustic node. For this purpose, the same combinations of speaker and mobile device shown in Table 4.1 are used, leading to a study with 16 combinations.

Regarding the possible values of L_B, a large number of values could be used, but for the sake of simplicity, only four values will be considered. Since the Android OS provides the reference size $L_{B,ref}$, three of these values are based on this reference, specifically: 1) $L_B = L_{B,ref}$, 2) $L_B = \frac{L_{B,ref}}{2}$ and 3) $L_B = \frac{L_{B,ref}}{3}$. In addition, according to [163], the native buffer size is considered, that is, $L_B = L_{B,native}$. Since the parameters $L_{B,ref}$ and $L_{B,native}$ depend on the specific mobile device, the values of each one (in samples) are given in Table C.1, where $L_{B,ref}$ is shown in terms of $L_{B,native}$.

The scenario used to perform the study is based on Figure 4.6 where

Mobile Device	$L_{\mathrm{B,ref}}$	$L_{\mathrm{B,native}}$
Samsung Galaxy S3	$57L_{\mathrm{B,native}}$	192
Samsung Galaxy Tab 3	$10L_{\mathrm{B,native}}$	1024
Motorola Moto G5 Plus	$45L_{\mathrm{B,native}}$	240
HTC Nexus 9	$99L_{\mathrm{B,native}}$	128

Table C.1. Buffer sizes (in samples unit) of each mobile device.

$f_{\mathrm{s}} = 44100$ Hz and the mobile device and the speaker are separated a distance of 50 cm. It has to be noticed that, as Section 4.2.3 concludes, the separation distance is not a relevant factor in T_{AL}. Since this study aims to analyze the offset that suffers T_{AL}, similarly to the study carried out in Section 4.4.3, multiple sweeps are recorded and reproduced. In this case, a total of five sweeps are reproduced and recorded, resulting in four different measurements of the offset. In addition, the sweeps are not sequentially recorded and reproduced between them, but a period of L_{P} zeroes samples will be reproduced and recorded in order to produce a similar behavior that seen in Section 4.4.3. Therefore, the study is performed as follows:

1. Select an acoustic node from Table 4.1.
2. Select a size for L_{B}.
3. Perform the warm-up stage, that reproduces and records L_{W} samples of zeroes, being $L_{\mathrm{W}} = 3f_{\mathrm{s}}$.
4. Reproduce and record a sweep signal of L_{S} samples, being $L_{\mathrm{S}} = 2f_{\mathrm{s}}$.
5. Reproduce and record L_{P} zeroes. Since in Section 4.4.3 the elapsed time in the "Audio Processing" stage is around 10 seconds, $L_{\mathrm{P}} = 10f_{\mathrm{s}}$ samples of zeroes.
6. Repeat four times steps 4 and 5.
7. Close recording and reproduction and estimate p_{AL} (4.6) through the method explained in Section 4.2.2 for the five sweep signals.
8. Return to step 3 and repeat the process 20 times.

9. Return to step 2, change the value of L_{B} and repeat steps 2 to 8.

10. Return to step 1, change the combination and repeat steps 1 to 9.

As Section 4.4.3 states, the offset that suffers p_{AL} can be obtained through the difference between two consecutive sweeps, such as

$$\Delta p^{(\mathrm{n})} = p_{\mathrm{AL}}^{(\mathrm{n}+1)} - p_{\mathrm{AL}}^{(\mathrm{n})}\,, \tag{1}$$

where $n = 1, \ldots, 4$ is the number of the estimation and Δp indicates the offset that suffers p_{AL} (or T_{AL}). When the four values of $\Delta p^{(\mathrm{n})}$ have been estimated, the standard deviation (σ_{p}) is calculated in order to observe the stability of Δp. Since the process for each acoustic node and buffer size is performed 20 times (step 8), σ_{p} is estimated 20 times, and its maximum value is saved for comparison. In this way, the saved value of σ_{p} represents the worst performing case for that combination and is the limiting case for that acoustic node an L_{B}.

The results are shown in Table C.2, Table C.3, Table C.4 and Table C.5, where each table refers to a different speaker. For each table, the best options have been highlighted in bold typeface.

	Mobile Device			
Buffer size (L_{B})	Samsung S3	HTC Nexus 9	Motorola Moto G5	Samsung Tab 3
$L_{\mathrm{B,ref}}$	188.35	191.83	184.16	1881
$L_{\mathrm{B,ref}}/2$	**4.50**	**0.57**	**3.68**	**0.50**
$L_{\mathrm{B,ref}}/3$	6.35	0.57	4.19	0.50
$L_{\mathrm{B,native}}$	6.35	0.57	4.27	0.50

Table C.2. Maximum value of σ_{p} for the Yamaha NX P-100 speaker.

Table C.2 shows the maximum value of σ_{p} when the Yamaha NX P-100 speaker is used in the acoustic node. Results show that the stability exhibited by L_{B} sizes $\frac{L_{\mathrm{B,ref}}}{2}$, $\frac{L_{\mathrm{B,ref}}}{3}$ and $L_{\mathrm{B,native}}$ is equal for the HTC Nexus 9 and Samsung Tab 3 devices, whereas $\frac{L_{\mathrm{B,ref}}}{2}$ is slightly better than $\frac{L_{\mathrm{B,ref}}}{3}$ and $L_{\mathrm{B,native}}$ for the other two devices. However, the stability significantly

decreases when $L_B = L_{\mathrm{B,ref}}$, reaching a value of 1880 samples in the worst case. This means that T_{AL} can change up to 42 ms even if reproduction and recording are carried out in a continuous way. One plausible explanation for this result is because of the large size of L_{B} and the large number of samples that have to be processed at once, causing an unstable reproduction process.

	Mobile Device			
Buffer size (L_{B})	Samsung S3	HTC Nexus 9	Motorola Moto G5	Samsung Tab 3
$L_{\mathrm{B,ref}}$	297.20	191.83	192.16	20.12
$L_{\mathrm{B,ref}}/2$	**3.10**	**0.57**	**0.50**	**0.57**
$L_{\mathrm{B,ref}}/3$	3.10	0.57	0.50	0.57
$L_{\mathrm{B,native}}$	3.20	0.57	0.50	0.57

Table C.3. Maximum value of σ_{p} for the Sony SRS X3 speaker.

Table C.3 shows the maximum value of σ_{p} when the Sony SRS X3 speaker is used in the acoustic node. Results show a similar behavior to those shown in Table C.2. That is, a very stable Δp is obtained when options from $\frac{L_{\mathrm{B,ref}}}{2}$ to $L_{\mathrm{B,native}}$ are used. Therefore, these three options indicate that the offset that suffers T_{AL} is low and stable. Additionally, the option of $L_{\mathrm{B,ref}}$ continues to provoke an unstable behavior because of the same reason as explained above.

	Mobile Device			
Buffer size (L_{B})	Samsung S3	HTC Nexus 9	Motorola Moto G5	Samsung Tab 3
$L_{\mathrm{B,ref}}$	363.68	686.63	**359.12**	198.26
$L_{\mathrm{B,ref}}/2$	183.33	417.18	398.47	**192.26**
$L_{\mathrm{B,ref}}/3$	176.02	**363.38**	443.24	193.07
$L_{\mathrm{B,native}}$	**133.37**	376.75	444.96	389.75

Table C.4. Maximum value of σ_{p} for the JBL Flip 2 speaker.

Table C.4 shows the maximum value of σ_p when the JBL Flip 2 is used in the acoustic node. Results show that all the possible sizes of L_B present a very unstable behavior and a clear option on the best L_B size cannot be concluded. Indeed the best value of L_B is different for each mobile device, as opposed to the results shown in Tables C.2 and C.3. Results in Table C.4 show that, regardless to the size of L_B, a high deviation is unavoidable, thus, the offset suffered by T_{AL} is unpredictable, confirming the bad results of Figure 4.28. Consequently, none of the available options of L_B is better than the other.

	Mobile Device			
Buffer size (L_B)	Samsung S3	HTC Nexus 9	Motorola Moto G5	Samsung Tab 3
$L_{B,ref}$	214.54	175.03	151.17	10.46
$L_{B,ref}/2$	**42.35**	50.94	**27.41**	13.93
$L_{B,ref}/3$	63.90	**40.93**	30.74	**9.94**
$L_{B,native}$	43.93	56.19	70.25	10.24

Table C.5. Maximum value of σ_p for the JBL Charge 4 speaker.

Finally, Table C.5 shows the maximum value of σ_p when the JBL Charge 4 is used in the acoustic node. In this case, a better behavior compared to the results in Table C.4 can be appreciated and the offset suffered by T_{AL} is less unstable. However, the results in Table C.5 are still very high compared to those provided by Tables C.2 and C.3. Regarding the worst stability option, it is caused when $L_B = L_{B,ref}$, while the other three options present similar behavior.

To conclude, the option $L_B = L_{B,ref}$ should be avoided as an optimal buffer size since it represents the worst results for all the acoustic nodes. The other three options ($\frac{L_{B,ref}}{2}$, $\frac{L_{B,ref}}{3}$ and $L_{B,native}$) present a similar stable offset for the loudspeakers of Tables C.2 and C.3, whereas for the JBL 4 Charge speaker of Table C.5, they improve the stability of the offset of T_{AL} compared to $L_B = L_{B,ref}$. As a result, any of these three options could be chosen as the optimal buffer size, but a buffer size of $\frac{L_{B,ref}}{2}$ has been considered in this dissertation.

Bibliography

[1] A. Bertrand, S. Doclo, S. Gannot, N. Ono, and T. van Waterschoot, "Special issue on wireless acoustic sensor networks and ad hoc microphone arrays," *Signal Processing*, vol. 107, no. C, pp. 1–3, 2015.

[2] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and Classification of Acoustic Scenes and Events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, oct 2015.

[3] S. Markovich-Golan, A. Bertrand, M. Moonen, and S. Gannot, "Optimal distributed minimum-variance beamforming approaches for speech enhancement in wireless acoustic sensor networks," *Signal Processing*, vol. 107, pp. 4–20, 2015.

[4] M. Ferrer, M. De Diego, G. Piñero, and A. Gonzalez, "Active noise control over adaptive distributed networks," *Signal Processing*, vol. 107, pp. 82–95, 2015.

[5] V. Molés-Cases, G. Piñero, M. de Diego, and A. Gonzalez, "Personal sound zones by subband filtering and time domain optimization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2684–2696, 2020.

[6] H. Fastl and E. Zwicker, *Psychoacoustics: Facts and models*, 3rd ed. Berlin, Heidelberg: Springer Science & Business Media, 2007.

[7] J. Rämö, V. Välimäki, M. Alanko, and M. Tikander, "Perceptual frequency response simulator for music in noisy environments," in *Audio Engineering Society Conference: 45th International Conference: Applications of Time-Frequency Processing in Audio*. Helsinki, Finland: Audio Engineering Society, 1-4 Mar. 2012, pp. 216–225.

[8] H. Kim, S. J. Lee, J. W. Choi, H. Bae, J. Lee, J. Song, and I. Shin, "Mobile maestro: Enabling immersive multi-speaker audio applications on commodity mobile devices," in *UbiComp 2014 - Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, Seattle, WA, US, 13-17 Sep. 2014, pp. 277–288.

[9] M. Cobos, J. J. Perez-Solano, Ó. Belmonte, G. Ramos, and A. M. Torres, "Simultaneous ranging and self-positioning in unsynchronized wireless acoustic sensor networks," *IEEE Transactions on Signal Processing*, vol. 64, no. 22, pp. 5993–6004, 2016.

[10] A. Bertrand, "Applications and trends in wireless acoustic sensor networks: A signal processing perspective," in *2011 18th IEEE Symposium on Communications and Vehicular Technology in the Benelux, SCVT 2011*, Ghent, Belgium, 22-23 Nov. 2011, pp. 1–6.

[11] G.-Z. Yang and G. Yang, *Body sensor networks*, 2nd ed. Heidelberg, UK: Springer, 2006, vol. 1.

[12] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–105, 2002.

[13] C. Y. Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, aug 2003.

[14] M. Tubaishat and S. Madria, "Sensor networks: An overview," *IEEE Potentials*, vol. 22, no. 2, pp. 20–23, apr 2003.

[15] L. Yu, N. Wang, and X. Meng, "Real-time forest fire detection with wireless sensor networks," in *Proceedings - 2005 International Conference on Wireless Communications, Networking and Mobile Computing, WCNM 2005*, vol. 2, Wuhan, China, 26 Sep. 2005, pp. 1214–1217.

[16] V. Boonsawat, J. Ekchamanonta, K. Bumrungkhet, and S. Kittipiyakul, "Xbee wireless sensor networks for temperature monitoring," in *The second conference on application research and development (ECTI-CARD 2010)*, Chon Buri, Thailand, 10-12 May 2010, pp. 221–226.

[17] C. C. Poon, Y. T. Zhang, and S. D. Bao, "A novel biometrics method to secure wireless body area sensor networks for telemedicine and M-health," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 73–81, 2006.

[18] J. M. Caldeira, J. J. Rodrigues, and P. Lorenz, "Toward ubiquitous mobility solutions for body sensor networks on healthcare," *IEEE Communications Magazine*, vol. 50, no. 5, pp. 108–115, 2012.

[19] S. H. Lee, S. Lee, H. Song, and H. S. Lee, "Wireless sensor network design for tactical military applications: Remote large-scale environments," in *Proceedings - IEEE Military Communications Conference MILCOM*, Boston, MA, USA, 19-21 Oct. 2009, pp. 1–7.

[20] S. M. Diamond and M. G. Ceruti, "Application of wireless sensor network to military information integration," in *IEEE International Conference on Industrial Informatics (INDIN)*, vol. 1, Vienna, Austria, 23-27 Jun. 2007, pp. 316–322.

[21] K. Martinez, J. K. Hart, and R. Ong, "Environmental sensor networks," *Computer*, vol. 37, no. 8, pp. 50–56, 2004.

[22] D. Puccinelli and M. Haenggi, "Wireless sensor networks: Applications and challenges of ubiquitous sensing," *IEEE Circuits and systems magazine*, vol. 5, no. 3, pp. 19–29, 2005.

[23] K. Chintalapudi, T. Fu, J. Paek, N. Kothari, S. Rangwala, J. Caffrey, R. Govindan, E. Johnson, and S. Masri, "Monitoring civil structures with a wireless sensor network," *IEEE Internet Computing*, vol. 10, no. 2, pp. 26–34, mar 2006.

[24] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[25] P. A. Naylor and N. D. Gaubitch, *Speech Dereverberation*. London, UK: Springer Science & Business Media, 2010.

[26] M. Maroti, G. Simon, A. Ledeczi, and J. Sztipanovits, "Shooter localization in urban terrain," *Computer*, vol. 37, no. 8, pp. 60–61, aug 2004.

[27] W. P. Chen, J. C. Hou, and L. Sha, "Dynamic clustering for acoustic target tracking in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 258–271, jul 2004.

[28] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh, "Monitoring volcanic eruptions with a wireless sensor network," in *Proceedings of the Second European Workshop on Wireless Sensor Networks, EWSN 2005*, vol. 2005, Istanbul, Turkey, 2 Feb. 2005, pp. 108–120.

[29] B. Barbagli, G. Manes, R. Facchini, S. Marta, and A. Manes, "Acoustic Sensor Network for Vehicle Traffic Monitoring," in *VEHICULAR 2012 : The First International Conference on Advances in Vehicular Systems, Technologies and Applications*, no. May, Venice, Italy, 24-29Jun. 2012, pp. 1–6.

[30] J. E. Noriega-Linares and J. M. Navarro Ruiz, "On the application of the raspberry pi as an advanced acoustic sensor network for noise monitoring," *Electronics*, vol. 5, no. 4, p. 74, oct 2016.

[31] K. S. Fisher, K. Baxter, and S. Manderville, "Highly portable system for acoustic event detection," U.S. Patent 7,750,814, Jul. 6, 2010.

[32] C. V. Cotton and D. P. Ellis, "Spectral vs. spectro-temporal features for acoustic event detection," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, USA, 16-19Oct. 2011, pp. 69–72.

[33] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real life recordings," in *18th European Signal*

Processing Conference, Aalborg, Denmark, 23-27Aug. 2010, pp. 1267–1271.

[34] S. Jiang, "On securing underwater acoustic networks: A survey," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 1, pp. 729–752, jan 2019.

[35] J. Plata-Chaves, A. Bertrand, and M. Moonen, "Incremental multiple error filtered-X LMS for node-specific active noise control over wireless acoustic sensor networks," in *Proceedings of the IEEE Sensor Array and Multichannel Signal Processing Workshop*, Rio de Janeiro, Brazil, 10-13Jul. 2016.

[36] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. Shanghai, China: Institute of Electrical and Electronics Engineers Inc., 20-25 Mar. 2016, pp. 196–200.

[37] G. Piñero, C. Botella, M. De Diego, M. Ferrer, and A. González, "On the feasibility of personal audio systems over a network of distributed loudspeakers," in *25th European Signal Processing Conference, EUSIPCO 2017*. Kos, Greece: IEEE, 28-2 Aug./Sep. 2017, pp. 2729–2733.

[38] V. Molés-Cases, G. Piñero, A. Gonzalez, and M. De Diego, "Providing spatial control in personal sound zones using graph signal processing," in *European Signal Processing Conference*. A Coruna, Spain: European Signal Processing Conference, EUSIPCO, 2-6 Sep. 2019, pp. 1–5.

[39] A. Flammini, P. Ferrari, D. Marioli, E. Sisinni, and A. Taroni, "Wired and wireless sensor networks for industrial applications," *Microelectronics Journal*, vol. 40, no. 9, pp. 1322–1336, sep 2009.

[40] J. Segura-Garcia, S. Felici-Castell, J. J. Perez-Solano, M. Cobos, and J. M. Navarro, "Low-cost alternatives for urban noise nuisance monitoring using wireless sensor networks," *IEEE Sensors Journal*, vol. 15, no. 2, pp. 836–844, feb 2015.

[41] J. C. Szurley, "Distributed signal processing algorithms for acoustic sensor networks," Ph.D. dissertation, Katholieke Universiteit (KU) Leuven, Oude Markt 13, 3000, Leuven, Belgium, June 2015.

[42] C. Antoñanzas Manuel, "Distributed and Collaborative Processing of Audio Signals: Algorithms, Tools and Applications," Ph.D. dissertation, Universitat Politècnica de València, Camí de Vera, s/n, 46022 València, Valencia, July 2020.

[43] J. Chen and A. H. Sayed, "On the learning behavior of adaptive networks - Part I: Transient analysis," *IEEE Transactions on Information Theory*, vol. 61, no. 6, pp. 3487–3517, jun 2015.

[44] P. Di Lorenzo and S. Barbarossa, "Distributed estimation and control of algebraic connectivity over random graphs," *IEEE Transactions on Signal Processing*, vol. 62, no. 21, pp. 5615–5628, nov 2014.

[45] N. Kitakoga and T. Ohtsuki, "Distributed em algorithms for acoustic source localization in sensor networks," in *IEEE Vehicular Technology Conference*, Montreal, QC, Canada, 25-28 Sep. 2006, pp. 2494–2498.

[46] N. Bogdanović, J. Plata-Chaves, and K. Berberidis, "Distributed diffusion-based LMS for node-specific parameter estimation over adaptive networks," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, Florence, Italy, 4-9 May 2014, pp. 7223–7227.

[47] Y. Zhai, M. B. Yeary, J. P. Havlicek, and G. Fan, "A new centralized sensor fusion-tracking methodology based on particle filtering for power-aware systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 10, pp. 2377–2387, 2008.

[48] A. A. Syed and J. Heidemann, "Time synchronization for high latency acoustic networks," in *Proceedings IEEE INFOCOM. 25TH IEEE International Conference on Computer Communications*, vol. 6, Barcelona, Spain, 23-29 Apr. 2006, pp. 1–12.

[49] J. Schmalenstroeer, P. Jebramcik, and R. Haeb-Umbach, "A combined hardware-software approach for acoustic sensor network synchronization," *Signal Processing*, vol. 107, pp. 171–184, 2015.

[50] C. Jaynes, "Multi-Room Wireless Display and Collaboration with Solstice," *Mersive Technologies*, pp. 1–5, 2017.

[51] Gear Patrol. The Complete Sonos Buying Guide: Every Speaker, Soundbar and Amp Explained. [Online]. Available: https://www.gearpatrol.com/tech/g37771425/complete-sonos-buying-guide-1632836322/. Accessed: 2022-02-11

[52] P. Giannoulis, G. Potamianos, and P. Maragos, "Room-localized speech activity detection in multi-microphone smart homes," *Eurasip Journal on Audio, Speech, and Music Processing*, no. 1, dec 2019.

[53] Y. Korber, M. Feld, and T. Schwartz, "Pervasive audio playback in cyber-physical environments," in *2017 Intelligent Systems Conference, IntelliSys 2017*, London, UK, 7-8 Sep. 2018, pp. 531–541.

[54] Statista. Cell phone sales worldwide 2007-2020. [Online]. Available: https://www.statista.com/statistics/263437/global-smartphone-sales-to-end-users-since-2007/. Accessed: 2022-02-11

[55] E. Murphy and E. A. King, "Testing the accuracy of smartphones and sound level meter applications for measuring environmental noise," *Applied Acoustics*, vol. 106, pp. 16–22, may 2016.

[56] C. A. Kardous and P. B. Shaw, "Evaluation of smartphone sound measurement applications," *The Journal of the Acoustical Society of America*, vol. 135, no. 4, pp. EL186–EL192, apr 2014.

[57] M. Mielke and R. Brück, "Smartphone application for automatic classification of environmental sound," in *Proceedings of the 20th International Conference on Mixed Design of Integrated Circuits and Systems, MIXDES 2013*, Gdynia, Poland, 20-22 Jun. 2013, pp. 512–515.

[58] P. Bihler, P. Imhoff, and A. B. Cremers, "SmartGuide - A smartphone museum guide with ultrasound control," *Procedia Computer Science*, vol. 5, pp. 586–592, 2011.

[59] StatCounter. Mobile Operating System Market Share Worldwide. [Online]. Available: https://gs.statcounter.com/os-market-share/mobile/worldwide#yearly-2009-2021. Accessed: 2021-12-06

[60] A. A. Sheikh, P. T. Ganai, N. A. Malik, and K. A. Dar, "Smartphone: Android Vs IOS," *The SIJ Transactions on Computer Science Engineering & its Applications (CSEA)*, vol. 01, no. 04, pp. 31–38, 2013.

[61] M. S. Ahmad, N. E. Musa, R. Nadarajah, R. Hassan, and N. E. Othman, "Comparison between android and iOS Operating System in terms of security," in *2013 8th International Conference on Information Technology in Asia - Smart Devices Trend: Technologising Future Lifestyle, Proceedings of CITA 2013.* Kota Samarahan, Malaysia: IEEE Computer Society, 1-4 Jul. 2013.

[62] Z. Benenson, F. Gassmann, and L. Reinfelder, "Android and iOS Users' Differences concerning Security and Privacy," in *Conference on Human Factors in Computing Systems (CHI)*, Paris, France, 27-2 Apr./May 2013, pp. 817–822.

[63] Android Developers. Platform Architecture. [Online]. Available: https://developer.android.com/guide/platform. Accessed: 2022-02-11

[64] S. Lee and J. W. Jeon, "Evaluating performance of android platform using native C for embedded systems," in *ICCAS 2010 - International Conference on Control, Automation and Systems*, Gyeonggi-do, Korea (South), 27-30 Oct. 2010, pp. 1160–1163.

[65] M. Linares-Vasquez, C. Vendome, Q. Luo, and D. Poshyvanyk, "How developers detect and fix performance bottlenecks in Android apps," in *2015 IEEE 31st International Conference on Software Maintenance and Evolution, ICSME 2015 - Proceedings*, Bremen, Germany, 29-1 Sep./Oct. 2015, pp. 352–361.

[66] SocialCompare. Android versions comparison — Comparison tables. [Online]. Available: http://socialcompare.com/en/comparison/android-versions-comparison. Accessed: 2022-02-11

[67] D. Zigunovs, J. Smirnova, G. Vitols, and G. Stonys, "Solution for Sound Playback Delay on Android Devices," *Procedia Computer Science*, vol. 104, pp. 413–420, 2016.

[68] T. Mantoro, M. A. Ayu, and D. Jatikusumo, "Live video streaming for mobile devices: An application on android platform," in *Proceeding of 2012 International Conference on Uncertainty Reasoning and Knowledge Engineering, URKE 2012*, Jalarta, Indonesia, 14-15 Aug. 2012, pp. 119–122.

[69] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai, "IEEE 802.11 wireless local area networks," *IEEE Communications Magazine*, vol. 35, no. 9, pp. 116–126, sep 1997.

[70] P. Committee, "IEEE Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN - Specific Requirements - Part 15: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPANs)," *IEEE Std 802.15.1-2002*, pp. 1–473, June 2002.

[71] N. Chhabra, "Comparative Analysis of Different Wireless Technologies," *International Journal Of Scientific Research In Network Security & Communication*, vol. 1, no. 5, pp. 13–17, 2013.

[72] A. P. Bilan, "Streaming audio over Bluetooth ACL links," in *Proceedings ITCC 2003, International Conference on Information Technology: Computers and Communications*, Las Vegas, NV, USA, 28-30 Apr. 2003, pp. 287–291.

[73] D. Jakubisin, M. Davis, C. Roberts, and I. Howitt, "Real-time audio transceiver utilizing 802.11b wireless technology," in *Conference Proceedings - IEEE SOUTHEASTCON*, Richmond, VA, USA, 22-25 Mar. 2007, pp. 692–697.

[74] Bluetooth Technology Website. Traditional Profile Specifications. [Online]. Available: https://www.bluetooth.com/Specifications/Profiles-Overview. Accessed: 2022-02-11

[75] H. Fletcher, "Auditory patterns," *Reviews of Modern Physics*, vol. 12, no. 1, pp. 47–65, 1940.

[76] G. S. Wong, "Microphones and Their Calibration," in *Springer Handbook of Acoustics*, 2nd ed. New York, NY: Springer, 2014, pp. 1061–1091.

[77] Y. Lin and W. H. Abdulla, "Principles of Psychoacoustics," in *Audio Watermark*. Cham: Springer International Publishing, 2015, pp. 15–49.

[78] E. G. Walsh, "Experiments in hearing," *Quarterly Journal of Experimental Physiology and Cognate Medical Sciences: Translation and Integration*, vol. 45, no. 3, pp. 324–325, 1960.

[79] J. A. Bierer, S. M. Bierer, H. A. Kreft, and A. J. Oxenham, "A fast method for measuring psychophysical thresholds across the cochlear implant array," *Trends in Hearing*, vol. 19, p. 233121651556979, dec 2015.

[80] H. Fletcher and W. A. Munson, "Loudness, Its Definition, Measurement and Calculation," *Bell System Technical Journal*, vol. 12, no. 4, pp. 377–430, oct 1933.

[81] E. Zwicker, G. Flottorp, and S. S. Stevens, "Critical Band Width in Loudness Summation," *The Journal of the Acoustical Society of America*, vol. 29, no. 5, pp. 548–557, may 1957.

[82] E. Zwicker and I. E. Terhardt, "Analytical expressions for critical-band rate and critical bandwidth as a function of frequency," *Journal of the Acoustical Society of America*, vol. 68, no. 5, pp. 1523–1525, nov 1980.

[83] M. D. Diego, A. González, G. Piñero, M. Ferrer, and J. J. García-Bonito, "Subjective evaluation of actively controlled interior car noise," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 5, Salt Lake City, UT, USA, 7-11 May 2001, pp. 3225–3228.

[84] U. Widmann, "Aurally adequate evaluation of sounds," in *Proceedings of Euro-noise*, vol. 98, 1998, pp. 29–46.

[85] G. Piñero, L. Fuster, A. Gonzalez, M. De Diego, M. Ferrer, K. Pernias, and M. Romero, "Sound quality evaluation of powered riding toy noises by children," in *41st International Congress and Exposition on Noise Control Engineering 2012, INTER-NOISE 2012*, vol. 12, New York, NY, USA, 19-22 Aug. 2012, pp. 10 488–10 499.

[86] M. D. Swanson, B. Zhu, A. H. Tewfik, and L. Boney, "Robust audio watermarking using perceptual masking," *Signal Processing*, vol. 66, no. 3, pp. 337–355, may 1998.

[87] T. S. Gunawan, E. Ambikairajah, and J. Epps, "Perceptual speech enhancement exploiting temporal masking properties of human auditory system," *Speech Communication*, vol. 52, no. 5, pp. 381–393, may 2010.

[88] U. Zölzer, *Digital Audio Signal Processing*, 2nd ed. Chichester, UK: John Wiley & Sons, Ltd, 2008.

[89] T. Painter and A. Spanias, "Perceptual coding of digital audio," *Proceedings of the IEEE*, vol. 88, no. 4, pp. 451–512, 2000.

[90] D. Pan, "A Tutorial on MPEG/Audio Compression," *IEEE Multimedia*, vol. 2, no. 2, pp. 60–74, 1995.

[91] N. Virag, "Speech enhancement based on masking properties of the auditory system," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 1. Detroit, MI, USA: IEEE, 9-12 May 1995, pp. 796–799.

[92] T. Zhou, Y. Zeng, and R. Wang, "Single-channel speech enhancement based on psychoacoustic masking," *AES: Journal of the Audio Engineering Society*, vol. 65, no. 4, pp. 272–284, apr 2017.

[93] M. Christoph, "Noise dependent equalization control," in *Audio Engineering Society Conference: 48th International Conference: Automotive Audio*. Munich, Germany: Audio Engineering Society, 21-23 Sep. 2012, pp. 77–86.

[94] J. Ramo, V. Valimaki, and M. Tikander, "Perceptual headphone equalization for mitigation of ambient noise," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, Vancouver, BC, Canada, 26-31 May 2013, pp. 724–728.

[95] H. Fletcher and W. A. Munson, "Relation Between Loudness and Masking," *Journal of the Acoustical Society of America*, vol. 9, no. 1, pp. 1–10, 1937.

[96] ISO Central Secretary, "Acoustics- Normal equal-loudness-level contours," International Organization for Standardization, Geneva, CH, Standard ISO 226:2003, August 2003.

[97] ISO Central Secretary, "Acoustics — Method for calculating loudness level — Part 1: Zwicker method," International Organization for Standardization, Geneva, CH, Standard ISO 532-1:2017, June 2017.

[98] R. Sottek, "Progress in calculating tonality of technical sounds," in *INTERNOISE 2014 - 43rd International Congress on Noise Control Engineering: Improving the World Through Noise Control*, vol. 249, no. 4, Melbourne, Australia, 16-19 Nov. 2014, pp. 3319–3327.

[99] J. D. Johnston, "Estimation of perceptual entropy using noise masking criteria," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, New York, NY, USA, 11-14 Apr. 1988, pp. 2524–2527.

[100] G. Q. Di, X. W. Chen, K. Song, B. Zhou, and C. M. Pei, "Improvement of Zwicker's psychoacoustic annoyance model aiming at tonal noises," *Applied Acoustics*, vol. 105, pp. 164–170, apr 2016.

[101] J. Becker, R. Sottek, and T. Lobato, "Progress in Tonality Calculation," in *Proceedings of the 23rd International Congress on Acoustics (ICA)*, Aachen, Germany, 9-13 Sep. 2019.

[102] Application Note, Psychoacoustics, II, "Calculating psychoacoustic parameters in ArtemiS SUITE," *HEAD acoustics GmbH*, pp. 1–9, 2016.

[103] M. Bosi and R. E. Goldberg, *Introduction to Digital Audio Coding and Standards.* Norwell, USA: Kluwer Academic Publishers, 2002, vol. 721.

[104] R. P. Hellman, "Effect of Noise Bandwidth on the Loudness of a 1000-Hz Tone," *The Journal of the Acoustical Society of America*, vol. 48, no. 2B, pp. 500–504, aug 1970.

[105] R. P. Hellman, "Asymmetry of masking between noise and tone," *Perception & Psychophysics*, vol. 11, no. 3, pp. 241–246, 1972.

[106] S. Van De Par, A. Kohlrausch, G. Charestan, and R. Heusdens, "A new psychoacoustical masking model for audio coding applications," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2, Orlando, FL, USA, 13-17 May 2002.

[107] B. S. Atal and J. L. Hall, "Optimizing digital speech coders by exploiting masking properties of the human ear," *Journal of the Acoustical Society of America*, vol. 66, no. 6, pp. 1647–1652, dec 1979.

[108] T. E. Miller and J. Barish, "Optimizing Sound for Listening in the Presence of Road Noise," in *Audio Engineering Society Convention 95*. New York, NY, USA: Audio Engineering Society, 7-10 Oct. 1993.

[109] P. Balazs, B. Laback, G. Eckel, and W. A. Deutsch, "Time-frequency sparsity by removing perceptually irrelevant components using a simple model of simultaneous masking," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 1, pp. 34–49, jan 2010.

[110] J. A. Mosquera-Sánchez, M. Sarrazin, K. Janssens, L. P. de Oliveira, and W. Desmet, "Multiple target sound quality balance for hybrid electric powertrain noise," *Mechanical Systems and Signal Processing*, vol. 99, pp. 478–503, jan 2018.

[111] J. D. Johnston, "Transform Coding of Audio Signals Using Perceptual Noise Criteria," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 2, pp. 314–323, 1988.

[112] P. D. Welch, "The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodograms," *IEEE Transactions on Audio and Electroacoustics*, vol. 15, no. 2, pp. 70–73, 1967.

[113] N. Virag, "Single channel speech enhancement based on masking properties of the human auditory system," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 2, pp. 126–137, 1999.

[114] F. Jabloun and B. Champagne, "Incorporating the Human Hearing Properties in the Signal Subspace Approach for Speech

Enhancement," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 700–708, 2003.

[115] D. M. Green, "Additivity of Masking," *The Journal of the Acoustical Society of America*, vol. 41, no. 6, pp. 1517–1525, jun 1967.

[116] R. A. Lutfi, "Additivity of simultaneous masking," *The Journal of the Acoustical Society of America*, vol. 73, no. 1, pp. 262–267, jan 1983.

[117] B. C. Moore, "Additivity of simultaneous masking, revisited," *Journal of the Acoustical Society of America*, vol. 78, no. 2, pp. 488–494, aug 1985.

[118] L. E. Humes and W. Jesteadt, "Models of the additivity of masking," *Journal of the Acoustical Society of America*, vol. 85, no. 3, pp. 1285–1294, mar 1989.

[119] K. Brandenburg, G. Stoll, Y. F. Dehery, J. D. Johnston, L. v.d. Kerkhof, and E. F. Schroder, "ISO-MPEG-1 Audio: A generic standard for coding of high-quality digital audio," *AES: Journal of the Audio Engineering Society*, vol. 42, no. 10, pp. 780–792, oct 1994.

[120] H. Gockel, B. C. J. Moore, and R. D. Patterson, "Asymmetry of masking between complex tones and noise: The role of temporal structure and peripheral compression," *The Journal of the Acoustical Society of America*, vol. 111, no. 6, pp. 2759–2770, jun 2002.

[121] H. Gockel, B. C. J. Moore, and R. D. Patterson, "Asymmetry of masking between complex tones and noise: Partial loudness," *The Journal of the Acoustical Society of America*, vol. 114, no. 1, pp. 349–360, jul 2003.

[122] K. Brandenburg, C. Faller, J. Herre, J. D. Johnston, and W. B. Kleijn, "Perceptual coding of high-quality digital audio," *Proceedings of the IEEE*, vol. 101, no. 9, pp. 1905–1919, 2013.

[123] K. H. Arehart, J. H. Hansen, S. Gallant, and L. Kalstein, "Evaluation of an auditory masked threshold noise suppression

algorithm in normal-hearing and hearing-impaired listeners," *Speech Communication*, vol. 40, no. 4, pp. 575–592, jun 2003.

[124] A. Taghipour, B. C. J. Moore, and B. Edler, "Masked threshold for noise bands masked by narrower bands of noise: Effects of masker bandwidth and center frequency," *The Journal of the Acoustical Society of America*, vol. 139, no. 5, pp. 2403–2406, may 2016.

[125] K. Brandenburg and J. D. Johnston, "Second Generation Perceptual Audio Coding: The Hybrid Coder," in *Audio Engineering Society Convention 88*. Montreux, Switzerland: Audio Engineering Society, 13-16 Mar. 1990.

[126] A. Taghipour, "Psychoacoustics of detection of tonality and asymmetry of masking: implementation of tonality estimation methods in a psychoacoustic model for perceptual audio coding," Ph.D. dissertation, Faculty of Engineering of Friedrich-Alexander Universität Erlangen-Nürnberg, Schloßplatz 4, 91054 Erlangen, Germany, April 2016.

[127] W. Aures, "Procedure for calculating the sensory pleasantness. of various sounds," *Acustica*, vol. 59, no. vol. 59, pp. 130–141, 1985.

[128] A. H. Gray and J. D. Markel, "A Spectral-Flatness Measure for Studying the Autocorrelation Method of Linear Prediction of Speech Analysis," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 22, no. 3, pp. 207–217, 1974.

[129] M. Shrestha and Z. Zhong, "Sound quality user-defined cursor reading control-tonality metric," Master's thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, March 2003.

[130] A. Hastings, K. H. Lee, P. Davies, and A. M. Surprenant, "Measurement of the attributes of complex tonal components commonly found in product sound," *Noise Control Engineering Journal*, vol. 51, no. 4, pp. 195–209, 2003.

[131] E. Terhardt, "Calculating virtual pitch," *Hearing Research*, vol. 1, no. 2, pp. 155–182, mar 1979.

[132] E. Terhardt, G. Stoll, and M. Seewann, "Algorithm for extraction of pitch and pitch salience from complex tonal signals," *Journal of the Acoustical Society of America*, vol. 71, no. 3, pp. 679–688, mar 1982.

[133] A. Hastings and P. Davies, "An examination of aures's model of tonality," in *INTERNOISE 2002 -The 2002 International Congress and Exposition on Noise Control Engineering*, vol. 2, Dearborn, MI, USA, 19-21 Aug. 2002, pp. 4–9.

[134] G. Laboratory. Audio and communication signal processing group (GTAC) website. [Online]. Available: https://gtac.webs.upv.es/. Accessed: 2022-02-11

[135] Mathworks. MATLAB & Simulink. [Online]. Available: https://mathworks.com/products/matlab.html. Accessed: 2022-02-11

[136] M. Nilsson, S. D. Soli, and J. A. Sullivan, "Development of the Hearing In Noise Test for the measurement of speech reception thresholds in quiet and in noise," *The Journal of the Acoustical Society of America*, vol. 95, no. 2, pp. 1085–1099, 1994.

[137] V. Välimäki and J. D. Reiss, "All about audio equalization: Solutions and frontiers," *Applied Sciences*, vol. 6, no. 5, p. 129, may 2016.

[138] J. S. Abel and D. P. Berners, "Filter Design Using Second-Order Peaking and Shelving Sections," in *30th Annual International Computer Music Conference Proceedings (ICMC)*, Miami, FL, USA, 1-6 Nov. 2004.

[139] R. J. Oliver and J. M. Jot, "Efficient multi-band digital audio graphic equalizer with accurate frequency response control," in *Audio Engineering Society Convention 139*. New York, NY, USA: Audio Engineering Society, 29-1 Oct./Nov. 2015.

[140] B. Bank, "Perceptually motivated audio equalization using fixed-pole parallel second-order filters," *IEEE Signal Processing Letters*, vol. 15, pp. 477–480, 2008.

[141] P. Hoffmann and B. Kostek, "A concept of signal equalization method based on music genre and the listener's room

characteristics," in *Signal Processing - Algorithms, Architectures, Arrangements, and Applications Conference Proceedings, SPA*. Poznan, Poland: IEEE Computer Society, 21-23 Sep. 2016, pp. 213–218.

[142] N. Westerlund, M. Dahl, and I. Claesson, "Speech enhancement using an adaptive gain equalizer with frequency dependent parameter settings," in *IEEE Vehicular Technology Conference*, vol. 60, no. 5, Los Angeles, CA, USA, 26-29 Sep. 2004, pp. 3718–3722.

[143] M. Ferrer, A. Gonzalez, M. De Diego, and G. Piñero, "Distributed Affine Projection Algorithm over Acoustically Coupled Sensor Networks," *IEEE Transactions on Signal Processing*, vol. 65, no. 24, pp. 6423–6434, dec 2017.

[144] G. Pinero and P. A. Naylor, "Channel estimation for crosstalk cancellation in wireless acoustic networks," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. New Orleans, LA, USA: IEEE, 5-9 Mar. 2017, pp. 586–590.

[145] S. Hansun, "A new approach of moving average method in time series analysis," in *2013 International Conference on New Media Studies, CoNMedia 2013*, Tangerang, Indonesia, 27-28 Nov. 2013, pp. 1–4.

[146] V. Valimaki and J. Liski, "Accurate cascade graphic equalizer," *IEEE Signal Processing Letters*, vol. 24, no. 2, pp. 176–180, feb 2017.

[147] H. A. David, *The Method of Paired Comparisons*, 2nd ed. London, UK ;New York, NY, USA: C. Griffin ;Oxford University Press, 1989.

[148] B. Widrow, C. S. Williams, J. R. Glover, J. M. McCool, R. H. Hearn, J. R. Zeidler, J. Kaunitz, E. Dong, and R. C. Goodlin, "Adaptive Noise Cancelling: Principles and Applications," *Proceedings of the IEEE*, vol. 63, no. 12, pp. 1692–1716, 1975.

[149] S. J. Elliott and P. A. Nelson, "The Active Control of Sound," *Electronics and Communication Engineering Journal*, vol. 2, no. 4, pp. 127–136, 1990.

[150] S. J. Elliott and P. A. Nelson, "Active Noise Control," *IEEE Signal Processing Magazine*, vol. 10, no. 4, pp. 12–35, 1993.

[151] M. J. Ji and S. M. Kuo, "An active harmonic noise equalizer," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1. Minneapolis, MN, USA: Publ by IEEE, 27-30 Apr. 1993, pp. 189–192.

[152] S. M. Kuo and M. J. Ji, "Development and Analysis of an Adaptive Noise Equalizer," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 3, pp. 217–222, 1995.

[153] A. Gonzalez, M. De Diego, M. Ferrer, and G. Piñero, "Multichannel active noise equalization of interior noise," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 1, pp. 110–122, jan 2006.

[154] J. C. Burgess, "Active adaptive sound control in a duct: A computer simulation," *Journal of the Acoustical Society of America*, vol. 70, no. 3, pp. 715–726, sep 1981.

[155] S. J. Elliott, I. M. Stothers, and P. A. Nelson, "A multiple error lms algorithm and its application to the active control of sound and vibration," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 10, pp. 1423–1434, 1987.

[156] J. Donley, C. Ritz, and W. B. Kleijn, "Multizone Soundfield Reproduction with Privacy- and Quality-Based Speech Masking Filters," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 26, no. 6, pp. 1037–1051, jun 2018.

[157] H. Khalilian, I. V. Bajić, and R. G. Vaughan, "A Simulation Study of a Three-Dimensional Sound Field Reproduction System for Immersive Communication," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 25, no. 5, pp. 980–995, may 2017.

[158] J. Zhang, Y. Li, and Y. Wei, "Using timestamp to realize audio-video synchronization in real-time streaming media transmission," in *ICALIP 2008 - 2008 International Conference on Audio, Language and Image Processing, Proceedings*, Shanghai, China, 7-9 Jul. 2008, pp. 1073–1076.

[159] B. Yonghwan, J. Han, L. Kyusang, J. Yoon, J. Jinoo, S. Yang, and K. R. June-Koo, "Wireless network synchronization for multichannel multimedia services," in *International Conference on Advanced Communication Technology, ICACT*, vol. 2, Gangwon, Korea (South), 15-18 Feb. 2009, pp. 1073–1077.

[160] M. Wright, R. J. Cassidy, and M. F. Zbyszy, "Audio and Gesture Latency Measurements on Linux and OSX Introduction and Prior Work," in *Proceedings of the 2004 International Computer Music Conference, ICMC*, Miami, Florida, USA, 1-6 Nov. 2004.

[161] Android Developers. Audio — Android Open Source Project. [Online]. Available: https://source.android.com/devices/audio. Accessed: 2022-02-11

[162] Robert Triggs. Android's Bluetooth latency needs a serious overhaul. [Online]. Available: https://www.soundguys.com/android-bluetooth-latency-22732/. Accessed: 2022-02-11

[163] Android Developers. Audio Latency. [Online]. Available: https://developer.android.com/ndk/guides/audio/audio-latency. Accessed: 2022-02-11

[164] N. Juillerat, S. M. Arisona, and S. Schubiger-Banz, "Real-time, low latency audio processing in Java," in *International Computer Music Conference, ICMC 2007*, Copenhagen, Denmark, 27-31 Aug. 2007, pp. 99–102.

[165] Android Developers. Sampling audio. [Online]. Available: https://developer.android.com/ndk/guides/audio/sampling-audio. Accessed: 2022-02-11

[166] D. Hermann, R. L. Brennan, H. Sheikhzadeh, and E. Cornu, "Low-power implementation of the Bluetooth Subband audio CODEC," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 5. Montreal, QC, Canada: IEEE, 17-21 May 2004.

[167] G. Spittle, "The Applications and Challenges of Processing Audio over Bluetooth," in *Audio Engineering Society Conference: UK 23rd*

Conference: Music Everywhere. Cambridge, UK: Audio Engineering Society, 9-10 Apr. 2008, pp. 1–10.

[168] H. Chen, H. Jin, K. Hu, and M. Yuan, "Adaptive audio-aware scheduling in Xen virtual environment," in *2010 ACS/IEEE International Conference on Computer Systems and Applications, AICCSA 2010*. Hammamet, Tunisia: IEEE Computer Society, 16-19 May 2010.

[169] S. Glegg and W. Devenport, "Chapter 3 - Linear acoustics," in *Aeroacoustics of Low Mach Number Flows: fundamentals, analysis, and measurement*. Academic Press, jan 2017, pp. 49–72.

[170] M. Azaria and D. Hertz, "Time Delay Estimation by Generalized Cross Correlation Methods," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 280–285, 1984.

[171] J. Vanderkooy, "Aspects of MLS measuring systems," *AES: Journal of the Audio Engineering Society*, vol. 42, no. 4, pp. 219–231, apr 1994.

[172] A. Farina, "Simultaneous measurement of impulse response and distortion with a swept-sine technique," in *Audio Engineering Society Convention 108*, no. I, Paris, France, 19-22 Feb. 2000, pp. 1–15.

[173] T. Betlehem, P. D. Teal, and Y. Hioka, "Efficient crosstalk canceler design with impulse response shortening filters," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, Kyoto, Japan, 25-30 Mar. 2012, pp. 393–396.

[174] A. V. Oppenheim, *Discrete-time signal processing*, 2nd ed. Hoboken, NJ, USA: Prentice Hall Taylor and Francis, 1999.

[175] G. B. Stan, J. J. Embrechts, and D. Archambeau, "Comparison of different impulse response measurement techniques," *AES: Journal of the Audio Engineering Society*, vol. 50, no. 4, pp. 249–262, 2002.

[176] E. Mota-García and R. Hasimoto-Beltran, "A new model-based clock-offset approximation over ip networks," *Computer communications*, vol. 53, pp. 26–36, 2014.

[177] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *SenSys'03: Proceedings of the First International Conference on Embedded Networked Sensor Systems*. New York, NY, USA: ACM Press, 5-7 Nov. 2003, pp. 138–149.

[178] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *Operating Systems Review (ACM)*, vol. 36, no. Special Issue, pp. 147–163, dec 2002.

[179] C. Lenzen, P. Sommer, and R. Wattenhofer, "Optimal clock synchronization in networks," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, SenSys*. Berkeley, CA, USA: ACM Press, 4-6 Nov. 2009, pp. 225–238.

[180] W. H. Hsu, Y. S. Liu, W. Y. Lin, W. C. Tai, and J. S. Wu, "A measurement of time synchronization on mobile devices," in *2012 IEEE I2MTC - International Instrumentation and Measurement Technology Conference, Proceedings*, Graz, Austria, 13-16 May 2012, pp. 2692–2694.

[181] M. L. Sichitiu and C. Veerarittiphan, "Simple, accurate time synchronization for wireless sensor networks," in *IEEE Wireless Communications and Networking Conference, WCNC*, vol. 2. New Orleans, LA, USA: IEEE, 16-20 Mar. 2003, pp. 1266–1273.

[182] M. S. R. Danesh and A. Ghasemi, "Robust and energy efficient clock synchronization scheme for wireless sensor networks," in *2013 21st Iranian Conference on Electrical Engineering, ICEE 2013*, Mashhad, Iran, 14-16 May 2013, pp. 1–5.

[183] Q. M. Chaudhari, E. Serpedin, and Y. C. Wu, "Improved estimation of clock offset in sensor networks," in *IEEE International Conference on Communications*, Dresden, Germany, 14-18 Jun. 2009, pp. 1–4.

[184] D. L. Mills, "Internet Time Synchronization: The Network Time Protocol," *IEEE Transactions on Communications*, vol. 39, no. 10, pp. 1482–1493, 1991.

[185] S. M. Jun, D. H. Yu, Y. H. Kim, and S. Y. Seong, "A time synchronization method for NTP," in *Proceedings - 6th*

International Conference on Real-Time Computing Systems and Applications, RTCSA 1999. Hong Kong, China: Institute of Electrical and Electronics Engineers Inc., 13-15 Dec. 1999, pp. 466–473.

[186] L. Wang, J. Fernandez, J. Burgett, R. W. Conners, and Y. Liu, "An evaluation of network time protocol for clock synchronization in wide area measurements," in *IEEE Power and Energy Society 2008 General Meeting: Conversion and Delivery of Electrical Energy in the 21st Century, PES*, Pittsburgh, PA, USA, 20-24 Jul. 2008, pp. 1–5.

[187] S. K. Mani, R. Durairajan, P. Barford, and J. Sommers, "MNTP: Enhancing time synchronization for mobile devices," in *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC*. Santa Monica, CA, USA: Association for Computing Machinery, 14-16 Nov. 2016, pp. 335–348.

[188] L. Schenato and G. Gamba, "A distributed consensus protocol for clock synchronization in wireless sensor network," in *Proceedings of the IEEE Conference on Decision and Control*, New Orleans, LA, USA, 12-14 Dec. 2007, pp. 2289–2294.

[189] L. Ferrigno, V. Paciello, and A. Pietrosanto, "Experimental characterization of synchronization protocols for instrument wireless interface," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 3, pp. 1037–1046, mar 2011.

[190] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 3, no. 3, pp. 281–323, may 2005.

[191] R. Casas, H. J. Gracia, Á. Marco, and J. L. Falcó, "Synchronization in wireless sensor networks using Bluetooth," in *Proceedings of the Third International Workshop on Intelligent Solutions in Embedded Systems, WISES'05*, Hamburg, Germany, 20 May 2005, pp. 79–88.

[192] J. Schmalenstroeer, P. Jebramcik, and R. Haeb-Umbach, "A gossiping approach to sampling clock synchronization in wireless acoustic sensor networks," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing -*

Proceedings. Florence, Italy: Institute of Electrical and Electronics Engineers Inc., 4-9 May 2014, pp. 7575–7579.

[193] Q. M. Chaudhari, "A simple and robust clock synchronization scheme," *IEEE Transactions on Communications*, vol. 60, no. 2, pp. 328–332, feb 2012.

[194] J. Kannisto, T. Vanhatupa, M. Hännikäinen, and T. D. Hämäläinen, "Software and hardware prototypes of the IEEE 1588 precision time protocol on wireless LAN," in *14th IEEE Workshop on Local and Metropolitan Area Networks, LANMAN 2005*, Crete, Greece, 18 Sep. 2005.

[195] K. Correll, N. Barendt, and M. Branicky, "Design considerations for software only implementations of the IEEE 1588 precision time protocol," in *Conference on IEEE*, 2005, pp. 11–15.

[196] Z. W. Park, J. H. Lee, and M. K. Kim, "Design of an extended TCP for preventing DOS attacks," in *Proceedings - KORUS 2003: 7th Korea-Russia International Symposium on Science and Technology*, vol. 2, Ulsan, Korea (South), 28-6 Jun./Jul. 2003, pp. 385–389.

[197] R. Tjoa, K. L. Chee, P. K. Sivaprasad, S. V. Rao, and J. G. Lim, "Clock drift reduction for relative time slot TDMA-based sensor networks," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, vol. 2, Barcelona, Spain, 5-8 Sep. 2004, pp. 1042–1047.

[198] M. Benndorf and T. Haenselmann, "Time synchronization on android devices for mobile construction assessment," in *The Tenth International Conference on Sensor Technologies and Applications. Thinkmind*, Nice, France, 24-28 Jul. 2016.

[199] E. Y. Song and K. Lee, "An application framework for the IEEE 1588 standard," in *2008 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication, ISPCS 2008, Proceedings*, Ann Arbor, MI, USA, 22-26 Sep. 2008, pp. 23–28.

[200] Apache Commons Net Library - Overview. [Online]. Available: https://commons.apache.org/proper/commons-net/. Accessed: 2022-02-11

[201] H. Cho, J. Jung, B. Cho, Y. Jin, S.-W. Lee, and Y. Baek, "Precision time synchronization using ieee 1588 for wireless sensor networks," in *2009 International Conference on Computational Science and Engineering*, vol. 2. Vancouver, BC, Canada: IEEE, 29-31 Aug. 2009, pp. 579–586.

[202] J. Zhang and P. Wu, "Joint Sampling Synchronization and Source Localization for Wireless Acoustic Sensor Networks," *IEEE Communications Letters*, vol. 24, no. 5, pp. 1020–1023, may 2020.

[203] T. Betlehem, W. Zhang, M. A. Poletti, and T. D. Abhayapala, "Personal sound zones: Delivering interface-free audio to multiple listeners," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 81–91, mar 2015.

[204] J. Francombe, P. Coleman, M. Olik, K. Baykaner, P. J. Jackson, R. Mason, M. Dewhirst, S. Bech, and J. A. Pedersen, "Perceptually optimised loudspeaker selection for the creation of personal sound zones," in *Audio Engineering Society Conference: 52nd International Conference: Sound Field Control-Engineering and Perception*, Guildford, UK, 2-4 Sep. 2013, pp. 169–178.

[205] P. Coleman, P. Jackson, M. Olik, and J. A. Pedersen, "Optimizing the planarity of sound zones," in *Audio Engineering Society Conference: 52nd International Conference: Sound Field Control-Engineering and Perception*, Guildford, UK, 2-4 Sep. 2013, pp. 204–213.

[206] J.-Y. Park, J.-H. Chang, and Y.-H. Kim, "Generation of independent bright zones for a two-channel private audio system," *Journal of the Audio Engineering Society*, vol. 58, no. 5, pp. 382–393, 2010.

[207] D. B. Ward, "On the performance of acoustic crosstalk cancellation in a reverberant environment," *The Journal of the Acoustical Society of America*, vol. 110, no. 2, pp. 1195–1198, aug 2001.

[208] M. F. Simon Galvez, S. J. Elliott, and J. Cheer, "Time Domain Optimization of Filters Used in a Loudspeaker Array for Personal Audio," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 23, no. 11, pp. 1869–1878, 2015.

[209] P. A. Nelson, H. Hamada, S. J. Elliott *et al.*, "Adaptive inverse filters for stereophonic sound reproduction," *IEEE Transactions on Signal Processing*, vol. 40, no. 7, pp. 1621–1632, 1992.

[210] T. Kabzinski and P. Jax, "A causality-constrained frequency-domain least-squares filter design method for crosstalk cancellation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2942–2956, 2021.

[211] B. Masiero and M. Vorländer, "A framework for the calculation of dynamic crosstalk cancellation filters," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 22, no. 9, pp. 1345–1354, 2014.

[212] O. Kirkeby, P. A. Nelson, H. Hamada, and F. Orduna-Bustamante, "Fast deconvolution of multichannel systems using regularization," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 2, pp. 189–194, 1998.

[213] M. Miyoshi and Y. Kaneda, "Inverse filtering of room acoustics," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 36, no. 2, pp. 145–152, 1988.

[214] R. Van Rompaey and M. Moonen, "Distributed adaptive acoustic contrast control for node-specific sound zoning in a wireless acoustic sensor and actuator network," in *2020 28th European Signal Processing Conference (EUSIPCO)*. Amsterdam, Netherlands: IEEE, 18-21 Jan. 2021, pp. 481–485.

[215] M. Poletti, "An investigation of 2-d multizone surround sound systems," in *Audio Engineering Society Convention 125*, San Francisco, CA, USA, 2-5 Oct. 2008.

[216] ITU-T, "Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," International Telecommunication Union, Geneva, CH, Recommendation ITU-T P.862 Amendment 2, November 2005.

[217] M. Jakobsson and S. Wetzel, "Security weaknesses in bluetooth," in *Cryptographers' Track at the RSA Conference*. San Francisco, CA, USA: Springer, 8-12 Apr. 2001, pp. 176–191.

[218] Android Developers. Bluetooth overview. [Online]. Available: https://developer.android.com/guide/topics/connectivity/bluetooth. Accessed: 2022-02-11

[219] Android Developers. Wi-Fi scanning overview. [Online]. Available: https://developer.android.com/guide/topics/connectivity/wifi-scan. Accessed: 2022-02-11

[220] G. Sagers, B. Hosack, R. J. Rowley, D. Twitchell, and R. Nagaraj, "Where's the security in WiFi? An argument for industry awareness," in *Proceedings of the 48th Hawaii International Conference on System Sciences*. Kauai, HI, USA: IEEE Computer Society, 5-8 Jan. 2015, pp. 5453–5461.

[221] G. Gounaris, "WiFi security and testbed implementation for WEP/WPA cracking demonstration," Ph.D. dissertation, Kingston University London, Penrhyn Road, Kingston upon Thames, Surrey KT1 2EE, United Kingdom, January 2014.