

**UNIVERSIDAD POLITÉCNICA DE MADRID**

**ESCUELA TÉCNICA SUPERIOR  
DE INGENIEROS DE TELECOMUNICACIÓN**



**MÁSTER UNIVERSITARIO EN INGENIERÍA  
DE TELECOMUNICACIÓN  
TRABAJO FIN DE MÁSTER**

**DISEÑO E IMPLEMENTACIÓN DE UNA  
CADENA DE GRABACIÓN -  
REPRODUCCIÓN BIAURAL DE AUDIO  
ESPACIAL PARA NORMA MPEG-H**

**LOURDES COBO-REYES LENDÍNEZ  
2021**



# MÁSTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIÓN

## TRABAJO FIN DE MÁSTER

**Título:** Diseño e implementación de una cadena de grabación - reproducción binaural de audio espacial para norma MPEG-H.

**Autor:** D<sup>a</sup> Lourdes Cobo-Reyes Lendínez.

**Tutor:** D. Javier Casajús Quirós.

**Ponente:** D. ....

**Departamento:** Señales, Sistemas y Radiocomunicaciones.

## MIEMBROS DEL TRIBUNAL

**Presidente:** D. ....

**Vocal:** D. ....

**Secretario:** D. ....

**Suplente:** D. ....

Los miembros del tribunal arriba nombrados acuerdan otorgar la calificación de: .....

Madrid, a                      de                      de 20...

**UNIVERSIDAD POLITÉCNICA DE MADRID**

**ESCUELA TÉCNICA SUPERIOR  
DE INGENIEROS DE TELECOMUNICACIÓN**



**MÁSTER UNIVERSITARIO EN INGENIERÍA  
DE TELECOMUNICACIÓN  
TRABAJO FIN DE MÁSTER**

**DISEÑO E IMPLEMENTACIÓN DE UNA  
CADENA DE GRABACIÓN-REPRODUCCIÓN  
BIAURAL DE AUDIO ESPACIAL PARA  
NORMA MPEG-H**

**LOURDES COBO-REYES LENDÍNEZ  
2021**



## RESUMEN

El continuo desarrollo e investigación por parte de instituciones como la ITU o el grupo MPEG ha dado lugar a un cambio en la forma en la que los usuarios disfrutamos del contenido multimedia. Desde 1930, con el nacimiento de las grabaciones estéreo, se ha tratado de perfeccionar la experiencia auditiva de los usuarios, pasando de configuraciones en las que dos canales incluían la misma información a estructuras conformadas por múltiples señales, de manera que el oyente es capaz de percibir el origen del sonido e incluso la sensación espacial de su entorno. Este paradigma ha motivado la elección de la elaboración de este Trabajo de Fin de Máster, en el que se integran múltiples conceptos relacionados con el tratamiento de audio mediante la implementación de distintos sistemas.

La elaboración de la investigación llevada a cabo a lo largo de este trabajo se organiza, principalmente, alrededor de tres ejes: el estudio de la base matemática que sustenta la Física de las ondas sonoras en el espacio tridimensional, el tratamiento de dichas señales para su transformación y su procesamiento para presentarlas de manera correcta al oyente. Esto constituirá el punto de partida para la construcción de dos sistemas en los que podremos verificar dicha base teórica.

Una señal sonora de 19 canales generada por un micrófono de tipo matriz dará comienzo a la cadena de procesamiento de audio que culminará mediante su entrega de naturaleza binaural. Para ello, se ha transformado la muestra inicial a una en el formato – B de Ambisonics, que consiste en un sistema multicanal que ofrece un sonido envolvente y tridimensional. De entre todas las posibilidades que ofrece este sistema en cuanto al número de canales, se eligen un total de 16, los cuales contendrán toda la información necesaria de la toma original. Tras esto, se computa la rotación espacial de cada una de las fuentes sonoras, de manera que el oyente las perciba en la posición deseada. A continuación, se elabora una decodificación en la que las fuentes pasarán a formar parte de un sistema de 12 canales, el 7.1.4. Llegamos a la etapa final del procesado, en la que se decodifica la señal anterior a un sistema binaural, es decir, a una configuración de dos canales que cuentan con la información necesaria del espacio sonoro y a través de la cual el oyente puede obtener una experiencia 3D sin necesidad de un gran número de altavoces físicos, sino que solo requiere el uso de auriculares.

La cadena se ha desarrollado en forma de dos sistemas. El primero, al que hemos denominado “piloto”, se ha programado en Python y ha servido como la base y verificación del resultado del segundo, un Plug – In importable en una Estación de Audio Digital y operable en tiempo real. Además, se ha utilizado un hardware específico para la correcta presentación del sistema sonoro mediante la generación de señales que permitirán el emplazamiento del sonido estereofónico.

De esta manera, hemos comprobado que la teoría se puede llevar a cabo en la práctica gracias a los medios y herramientas adecuados.

## SUMMARY

The way users enjoy and consume multimedia content has been affected by the development of new formats and systems by institutions or groups such as the ITU or MPEG. From the birth of stereo recordings back in 1930, where the two channels included the same information, a tremendous change has been experienced concerning the audio experience. Nowadays, the listener can perceive the incoming sound from different points in space, together with the environmental information, thanks to the new channels set – ups. This is the motive that has encouraged the topic of this Master Thesis, where many audio processing concepts are studied and put into practice with the implementation of different systems.

The investigation carried out has been organised around three main topics: the Mathematics and Physics behind sound waves travelling through tridimensional space, the treatment of those waves for their transformation and their processing in order to render them to the listener properly. All this information will be the starting point for building the systems up, so the theoretical basis can be verified.

A 19-channel sound signal obtained by an array microphone will be the first element of the audio processing chain that will conclude with its binaural representation. For that purpose, an Ambisonics B – format of 3<sup>rd</sup> order signal has been obtained from the initial audio file, which refers to a multichannel system capable of delivering a surrounding sound. The 16 channels given by this format will be spatially rotated, therefore, the user can perceive them where desired. Afterwards, a decoding process will take place to obtain 12 channels, set by the 7.1.4 recommendation by the ITU, out of the 16 previous ones. In the final step of this processing chain there will be a new decoding from the 12 signals to binaural, where two channels are enough for the listener to fulfil the 3D experience only by means of a pair of headphones and no physical speakers.

This chain has been presented in two different systems. The first one, named “pilot” and programmed by Python, has been the trial and verification technique of the second one: an audio Plug – In that can be inserted into a Digital Audio Workstation. What is more, the stereophonic sound created in the binaural configuration has been possible thanks to the use of a hardware capable of collecting the surrounding information.

As a result, we can confirm that theory is correctly put into practice when the means and tools are the proper ones.

## **PALABRAS CLAVE**

Audio, codificación, Moving Picture Experts Group, sistema, estéreo, multicanal, binaural, imagen sonora, Unión Internacional de Telecomunicaciones, armónicos esféricos, Ambisonics, Fourier, rotación, decodificación, Plug – In, Python, C++.

## **KEYWORDS**

Audio, coding, Moving Picture Experts Group, system, stereo, multichannel, binaural, sound image, International Telecommunications Union, spherical harmonics, Ambisonics, Fourier, rotation, decoding, Plug – In, Python, C++.

# ÍNDICE DEL CONTENIDO

|                                                                                        |           |
|----------------------------------------------------------------------------------------|-----------|
| <b>1. INTRODUCCIÓN Y OBJETIVOS.....</b>                                                | <b>1</b>  |
| 1.1. Introducción .....                                                                | 1         |
| 1.2. Objetivos.....                                                                    | 2         |
| <b>2. TEORÍA Y ESTADO DEL ARTE.....</b>                                                | <b>3</b>  |
| 2.1. Introducción a la codificación de audio.....                                      | 3         |
| 2.2. Codificación de Audio en MPEG .....                                               | 3         |
| 2.2.1. MPEG-1 Parte 3: Audio .....                                                     | 3         |
| 2.2.2. MPEG-2 Partes 3 y 7: Audio y <i>Advanced Audio Coding</i> .....                 | 5         |
| 2.2.3. MPEG-4 Parte 3: Audio .....                                                     | 5         |
| 2.2.4. MPEG-D Partes 1 y 2: <i>Surround</i> y <i>Spatial Audio Object Coding</i> ..... | 7         |
| 2.2.5. MPEG-H: 3D Audio.....                                                           | 12        |
| 2.3. Fundamentos Analíticos del Procesado Para Arrays Esféricos. ....                  | 14        |
| 2.3.1. Funciones sobre una Esfera .....                                                | 15        |
| 2.3.2. Armónicos Esféricos .....                                                       | 16        |
| 2.3.3. Transformada Esférica de Fourier.....                                           | 19        |
| 2.3.4. Rotación de Funciones .....                                                     | 20        |
| 2.4. Configuraciones de Sonido .....                                                   | 23        |
| 2.4.1. Sonido Monofónico .....                                                         | 23        |
| 2.4.2. Sonido Estéreo .....                                                            | 24        |
| 2.4.3. Sonido Cuadrafónico .....                                                       | 25        |
| 2.4.4. Multicanal .....                                                                | 26        |
| 2.5. Tecnología Ambisonics.....                                                        | 30        |
| 2.5.1. Higher Order Ambisonics (HOA).....                                              | 32        |
| 2.5.2. Relaciones entre Formato – B de Ambisonics y HOA .....                          | 33        |
| 2.6. Funciones de Transferencia Relacionadas con la Cabeza (HRTF) .....                | 35        |
| 2.7. Desarrollo de Plug - Ins de Audio.....                                            | 37        |
| 2.7.1. Tipos de Plug – Ins de Audio .....                                              | 37        |
| 2.7.2. JUCE .....                                                                      | 38        |
| 2.7.3. Microsoft Visual Studio .....                                                   | 41        |
| 2.8. Estaciones de Trabajo de Audio Digital.....                                       | 41        |
| 2.8.1. Reaper.....                                                                     | 42        |
| 2.8.2. Ableton .....                                                                   | 42        |
| 2.8.1. Adobe Audition .....                                                            | 43        |
| <b>3. HARDWARE .....</b>                                                               | <b>44</b> |
| 3.1. Micrófono Zylia .....                                                             | 44        |
| 3.2. Sennheiser Ambeo Smart Headset.....                                               | 45        |
| <b>4. DESARROLLO .....</b>                                                             | <b>46</b> |

|                                                                           |                                                                       |            |
|---------------------------------------------------------------------------|-----------------------------------------------------------------------|------------|
| 4.1.                                                                      | Arquitectura del sistema.....                                         | 46         |
| 4.1.                                                                      | Procedimiento de generación de la HRTF .....                          | 46         |
| 4.2.                                                                      | Desarrollo del sistema en Python.....                                 | 48         |
| 4.2.1.                                                                    | Preparación del Entorno .....                                         | 49         |
| 4.2.2.                                                                    | Generador de señales y Codificador Ambisonics .....                   | 49         |
| 4.2.3.                                                                    | Rotador Virtual de Canales .....                                      | 50         |
| 4.2.4.                                                                    | Decodificación a ITU 7.1.4 .....                                      | 51         |
| 4.2.5.                                                                    | Decodificación a un sistema Binarial .....                            | 56         |
| 4.3.                                                                      | Desarrollo del Plug - In .....                                        | 58         |
| 4.3.1.                                                                    | Elaboración de la Interfaz .....                                      | 58         |
| 4.3.2.                                                                    | Rotador Virtual de Canales .....                                      | 60         |
| 4.3.3.                                                                    | Decodificación a ITU 7.1.4.....                                       | 61         |
| 4.3.4.                                                                    | Decodificación a un sistema Binarial .....                            | 62         |
| <b>5.</b>                                                                 | <b>IMPLEMENTACIÓN .....</b>                                           | <b>64</b>  |
| 5.1.                                                                      | Preparación de archivos .....                                         | 64         |
| 5.2.                                                                      | Implementación en Reaper .....                                        | 64         |
| 5.3.                                                                      | Implementación en Adobe Audition.....                                 | 69         |
| <b>6.</b>                                                                 | <b>RESULTADOS.....</b>                                                | <b>73</b>  |
| 6.1.                                                                      | Resultados Numéricos .....                                            | 73         |
| 6.2.                                                                      | Resultados Perceptuales .....                                         | 75         |
| 6.2.1.                                                                    | Desarrollo de la prueba.....                                          | 76         |
| 6.2.2.                                                                    | Visualización de Resultados .....                                     | 76         |
| <b>7.</b>                                                                 | <b>APLICACIONES.....</b>                                              | <b>101</b> |
| <b>8.</b>                                                                 | <b>CONCLUSIONES Y LÍNEAS FUTURAS .....</b>                            | <b>102</b> |
| 8.1.                                                                      | Conclusiones .....                                                    | 102        |
| 8.2.                                                                      | Líneas futuras .....                                                  | 103        |
| <b>9.</b>                                                                 | <b>BIBLIOGRAFÍA.....</b>                                              | <b>104</b> |
| <b>ANEXO A: ASPECTOS ÉTICOS, ECONÓMICOS, SOCIALES Y AMBIENTALES .....</b> |                                                                       | <b>108</b> |
| A.1                                                                       | Introducción .....                                                    | 108        |
| A.2                                                                       | Descripción de impactos relevantes relacionados con el proyecto ..... | 108        |
| a.2.1.                                                                    | Impacto Económico .....                                               | 108        |
| A.2.2.                                                                    | Impacto Sanitario .....                                               | 109        |
| A.3                                                                       | Análisis detallado de alguno de los principales impactos .....        | 109        |
| A.3.1.                                                                    | Impacto Sanitario .....                                               | 109        |
| A.4                                                                       | Conclusiones .....                                                    | 110        |
| <b>ANEXO B: PRESUPUESTO ECONÓMICO.....</b>                                |                                                                       | <b>111</b> |

## ÍNDICE DE ILUSTRACIONES

|                                                                                                          |    |
|----------------------------------------------------------------------------------------------------------|----|
| Figura 1. Codificador de audio de las Capas I y II de MPEG-1, Parte 3 [2].                               | 4  |
| Figura 2. Codificador de audio de la Capa III de MPEG-1, Parte 3 [2].                                    | 4  |
| Figura 3. Codificador y decodificador de MPEG-4 Spectral Band Replication [5].                           | 6  |
| Figura 4. Representación del ángulo lateral en función de las Diferencias Interaurales de Intensidad.    | 8  |
| Figura 5. Sistema de coordenadas con los ángulos acimut y elevación                                      | 9  |
| Figura 6. Representación del ángulo lateral en función de las Diferencias Inter-aurales de Tiempo [8].   | 10 |
| Figura 7. Esquema de representación de la $ILT$ [EP].                                                    | 10 |
| Figura 8. Ilustración de los fenómenos ILD, ITD e IC [EP].                                               | 11 |
| Figura 9. Logo de MPEG-H Audio.                                                                          | 12 |
| Figura 10. Arquitectura básica del decodificador de audio MPEG-H [17].                                   | 13 |
| Figura 11. Sistema de coordenadas Esférico definido en el sistema de coordenadas Cartesiano.             | 15 |
| Figura 12. Representaciones gráficas de las funciones definidas sobre la esfera [19].                    | 16 |
| Figura 13. Gráficas de balón de los armónicos esféricos desde $n = 0$ hasta $n = 3$ .                    | 18 |
| Figura 14. Ángulos de Euler.                                                                             | 20 |
| Figura 15. Rotación de un gráfico de globo [25].                                                         | 23 |
| Figura 16. Diagrama de sonido monoaural [EP].                                                            | 23 |
| Figura 17. Diagrama de sonido estéreo [EP].                                                              | 24 |
| Figura 18. Configuración de micrófonos <i>par coincidente</i> [EP].                                      | 25 |
| Figura 19. Diagrama del sonido cuadrafónico [EP].                                                        | 25 |
| Figura 20. Estructura de referencia del sistema de sonido 5.1 recomendado por la ITU.                    | 26 |
| Figura 21. Vista lateral de 5.1.2 [29].                                                                  | 28 |
| Figura 22. Vista lateral de 5.1.4 [30].                                                                  | 28 |
| Figura 23. Emplazamiento de altavoces del sistema 7.1.4 desde el plano cenital.                          | 29 |
| Figura 24. Vista lateral de 7.1.4.                                                                       | 30 |
| Figura 25. Primer micrófono Soundfield [34].                                                             | 31 |
| Figura 26. Logo de Ambisonics.                                                                           | 31 |
| Figura 27. Funciones esféricas de Bessel $j_m(kr)$ [36].                                                 | 33 |
| Figura 28. <i>Dummy-head</i> del fabricante Neumann.                                                     | 36 |
| Figura 29. El físico acústico Vern Oliver Knudsen en la cámara anecoica del MIT.                         | 37 |
| Figura 30. Logo de JUCE.                                                                                 | 38 |
| Figura 31. Captura del entorno Projucer.                                                                 | 40 |
| Figura 32. Logo de Reaper.                                                                               | 42 |
| Figura 33. Logo de Ableton.                                                                              | 43 |
| Figura 34. Logo de Adobe Audition (versión 14).                                                          | 43 |
| Figura 35. Micrófono de array esférico Zylia Pro.                                                        | 44 |
| Figura 36. Auriculares Sennheiser Ambeo Smart Headset.                                                   | 45 |
| Figura 37. Esquema de la arquitectura del sistema [EP].                                                  | 46 |
| Figura 38. Esquema de identificación de $\mathbf{h}(\mathbf{n})$ mediante $\mathbf{x}(\mathbf{n})$ .     | 47 |
| Figura 39. Gráficas de forma de onda, espectro y espectrograma del barrido lineal.                       | 47 |
| Figura 40. <i>HRTF</i> medida para una fuente sonora situada a la derecha del oyente en el oído derecho. | 48 |
| Figura 41. Representaciones de un punto sobre una esfera                                                 | 50 |
| Figura 42. Resultado de la Matriz de Wigner – D para valores concretos de $(\alpha, \beta, \gamma)$ .    | 51 |
| Figura 43. Esquema general de un sistema Ambisonics                                                      | 52 |
| Figura 44. Diagramas de directividad de $N = 7$ micrófonos.                                              | 53 |
| Figura 45. Gráfica de las directividades de $N = 7$ micrófonos.                                          | 53 |
| Figura 46. Método <i>Overlap and Add</i> de manera gráfica [EP].                                         | 57 |
| Figura 47. Interfaz del Plug – In desarrollado.                                                          | 59 |
| Figura 48. Captura de asociación entre procesos <i>Visual Studio</i> y <i>Adobe Audition</i> .           | 63 |
| Figura 49. Inserción de instrumentos virtuales en Reaper.                                                | 64 |

|                                                                                                                                        |    |
|----------------------------------------------------------------------------------------------------------------------------------------|----|
| Figura 50. Matriz de señal Ambisonics para $\theta, \phi = (90^\circ, 0^\circ)$ .                                                      | 65 |
| Figura 51. Visualizador de onda de la señal Ambisonics en Reaper.                                                                      | 65 |
| Figura 52. Forma de onda de una señal Ambisonics rotada.                                                                               | 66 |
| Figura 53. Forma de onda de una señal Ambisonics decodificada a 7.1.4 en <i>Reaper</i> .                                               | 66 |
| Figura 54. Forma de onda de una señal Ambisonics rotada y decodificada a 7.1.4 en <i>Reaper</i> .                                      | 66 |
| Figura 55. Forma de onda de una señal binaural.                                                                                        | 67 |
| Figura 56. Gráfico de las primeras dos mil muestras del canal izquierdo de una señal binaural con transitorio.                         | 67 |
| Figura 57. Formas de onda de una señal creciente Ambisonics de tercer orden y su binauralización.                                      | 68 |
| Figura 58. Gráfico de las primeras mil quinientas muestras del canal izquierdo de una señal binaural sin transitorio.                  | 68 |
| Figura 59. Captura de pantalla de visualizador de onda de <i>Adobe Audition</i> .                                                      | 69 |
| Figura 60. Visualizador de forma de onda de una señal Ambisonics rotada en <i>Audition</i> .                                           | 70 |
| Figura 61. Formas de onda de una señal rotada y decodificada a 7.1.4 (izquierda) y decodificada a 7.1.4 (derecha) en <i>Audition</i> . | 70 |
| Figura 62. Señal binaural con transitorio.                                                                                             | 71 |
| Figura 63. Señal binaural sin transitorio.                                                                                             | 71 |
| Figura 64. Resultado numérico de una señal Ambisonics rotada mediante el Plug – In.                                                    | 73 |
| Figura 65. Resultado numérico de una señal Ambisonics rotada mediante el sistema Python.                                               | 73 |
| Figura 66. Gráfica de comparación de las rotaciones mediante la división de sus valores.                                               | 74 |
| Figura 67. Resultado numérico de una señal 7.1.4 a partir de una Ambisonics obtenida mediante el Plug – In.                            | 74 |
| Figura 68. Resultado numérico de una señal 7.1.4 a partir de una Ambisonics obtenida mediante el sistema Python.                       | 74 |
| Figura 69. Resultado numérico de una señal 7.1.4 a partir de una Ambisonics rotada mediante el Plug – In.                              | 75 |
| Figura 70. Resultado numérico de una señal 7.1.4 a partir de una Ambisonics rotada mediante el sistema Python.                         | 75 |
| Figura 71. Ilustración de la percepción de un sistema cuadrafónico con posiciones cardinales.                                          | 78 |
| Figura 72. Forma de onda del clip de voz original.                                                                                     | 78 |
| Figura 73. Formas de onda de una versión binaural del clip de voz.                                                                     | 79 |
| Figura 74. Ilustración de la percepción de un sistema cuadrafónico con posiciones cardinales para una señal de voz.                    | 79 |
| Figura 75. Forma de onda del clip de música original.                                                                                  | 80 |
| Figura 76. Formas de onda de una versión binaural de la señal musical.                                                                 | 80 |
| Figura 77. Ilustración de la percepción de un sistema cuadrafónico con posiciones cardinales para una señal musical.                   | 81 |
| Figura 78. Ilustración de la percepción de un sistema cuadrafónico.                                                                    | 82 |
| Figura 79. Ilustración de la percepción de un sistema cuadrafónico, señal de voz.                                                      | 82 |
| Figura 80. Ilustración de la percepción de un sistema cuadrafónico, señal de música.                                                   | 83 |
| Figura 81. Ilustración de la percepción de un sistema 7.1.4 en el plano horizontal.                                                    | 84 |
| Figura 82. Ilustración de la percepción de un sistema 7.1.4 en el plano vertical.                                                      | 85 |
| Figura 83. Ilustración de la percepción de un sistema 7.1.4 en el plano horizontal, señal de voz.                                      | 86 |
| Figura 84. Ilustración de la percepción de un sistema 7.1.4 en el plano vertical, señal de voz.                                        | 86 |
| Figura 85. Ilustración de la percepción de un sistema 7.1.4 en el plano horizontal, señal musical.                                     | 87 |
| Figura 86. Ilustración de la percepción de un sistema 7.1.4 en el plano vertical, señal musical.                                       | 87 |

|                                                                                                                                                                                                                                                                                                             |     |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Figura 87. Ilustración de la percepción de un sistema cuadrafónico con posiciones cardinales, señal de voz.....                                                                                                                                                                                             | 88  |
| Figura 88. Ilustración de la percepción de un sistema cuadrafónico con posiciones cardinales, señal musical.....                                                                                                                                                                                            | 89  |
| Figura 89. Ilustración de la percepción de un sistema cuadrafónico, señal de voz.....                                                                                                                                                                                                                       | 90  |
| Figura 90. Ilustración de la percepción de un sistema cuadrafónico, señal musical.....                                                                                                                                                                                                                      | 91  |
| Figura 91. Ilustración de la percepción de un sistema 7.1.4 en el plano horizontal, señal de voz.....                                                                                                                                                                                                       | 92  |
| Figura 92. Ilustración de la percepción de un sistema 7.1.4 en el plano vertical, señal de voz.....                                                                                                                                                                                                         | 92  |
| Figura 93. Ilustración de la percepción de un sistema 7.1.4 en el plano horizontal, señal musical.....                                                                                                                                                                                                      | 93  |
| Figura 94. Ilustración de la percepción de un sistema 7.1.4 en el plano vertical, señal musical.....                                                                                                                                                                                                        | 94  |
| Figura 95. Formas de onda de dos señales binaurales. A izquierda, la señal original y a la derecha, la señal rotada. La señal original tiene una posición definida por $\theta, \phi = (90^\circ, 0^\circ)$ , la cual se ha rotado mediante $\alpha, \beta, \gamma = (0^\circ, 0^\circ, 180^\circ)$ .....   | 95  |
| Figura 96. Formas de onda de dos señales binaurales. A izquierda, la señal original y a la derecha, la señal rotada. La señal original tiene una posición definida por $\theta, \phi = (45^\circ, 225^\circ)$ , la rotada tiene los valores $\alpha, \beta, \gamma = (90^\circ, 90^\circ, 180^\circ)$ ..... | 95  |
| Figura 97. Forma de onda de una señal rotada hacia delante.....                                                                                                                                                                                                                                             | 97  |
| Figura 98. Gráfico de comparativa entre valores acústicos en dB y situaciones reales [56]. .....                                                                                                                                                                                                            | 109 |

## ÍNDICE DE ECUACIONES

|                                                                                                                     |    |
|---------------------------------------------------------------------------------------------------------------------|----|
| Ecuación 1. Relación entre frecuencia y longitud de onda [EP].                                                      | 8  |
| Ecuación 2. Relación entre diferencias temporales y de fase para señales senoidales.                                | 9  |
| Ecuación 3. Cálculo de la Diferencia Inter-aural de Tiempo.                                                         | 10 |
| Ecuación 4. Sistema de coordenadas Cartesianas.                                                                     | 15 |
| Ecuación 5. Denotación de un vector de posición en función de su radio y ángulos elevación y acimut.                | 15 |
| Ecuación 6. Relación entre las coordenadas cartesianas y la posición $r$ .                                          | 15 |
| Ecuación 7. Notación de representación de la superficie esférica en coordenadas Cartesianas.                        | 16 |
| Ecuación 8. Definición de armónicos esféricos.                                                                      | 17 |
| Ecuación 9. Definición de polinomios de Legendre.                                                                   | 18 |
| Ecuación 10. Definición de polinomios de Legendre para valores negativos de $m$ .                                   | 18 |
| Ecuación 11. Aproximación de la definición de los polinomios de Legendre                                            | 19 |
| Ecuación 12. Definición de función en el espacio de Hilbert.                                                        | 19 |
| Ecuación 13. Coeficientes, o pesos, de Fourier.                                                                     | 19 |
| Ecuación 14. Definición de punto sobre la esfera en notación algebraica.                                            | 20 |
| Ecuación 15. Definición del punto $x'$ mediante rotación del punto $x$ .                                            | 20 |
| Ecuación 16. Definición de rotación inversa.                                                                        | 21 |
| Ecuación 17. Matrices de rotación de Euler.                                                                         | 21 |
| Ecuación 18. Definición de operador rotación $\Lambda$ .                                                            | 21 |
| Ecuación 19. Definición de rotación de armónicos esféricos.                                                         | 21 |
| Ecuación 20. Función de Wigner-D.                                                                                   | 22 |
| Ecuación 21. Función de Wigner-d.                                                                                   | 22 |
| Ecuación 22. Definición de rotación en el dominio de armónicos esféricos.                                           | 22 |
| Ecuación 23. Rotación en el dominio de armónicos esféricos mediante coeficientes de Fourier.                        | 23 |
| Ecuación 24. Ecuación de onda para la descomposición de un campo de presión en armónicos esféricos.                 | 32 |
| Ecuación 25. Descomposición del campo de presión acústica en la serie de Fourier – Bessel.                          | 32 |
| Ecuación 26. Armónicos esféricos en formato B.                                                                      | 33 |
| Ecuación 27. Normalización SN3D.                                                                                    | 34 |
| Ecuación 28. Normalización N3D.                                                                                     | 34 |
| Ecuación 29. Definición de armónico esférico conjugado.                                                             | 34 |
| Ecuación 30. Relación entre normalizaciones de armónicos esféricos.                                                 | 34 |
| Ecuación 31. Definición de las directividades de $N = 7$ micrófonos.                                                | 53 |
| Ecuación 32. Definición de los $N = 7$ canales.                                                                     | 53 |
| Ecuación 33. Canales <i>High Order Ambisonics</i> para $N = 7$ .                                                    | 53 |
| Ecuación 34. Definición de las componentes direccionales <i>HOA</i> .                                               | 54 |
| Ecuación 35. Definición de muestra sonora para un altavoz enésimo en una configuración equiespaciada de los mismos. | 54 |
| Ecuación 36. Definición matricial de codificación Ambisonics.                                                       | 54 |
| Ecuación 37. Definición de matriz de decodificación como la pseudoinversa de la matriz de codificación.             | 55 |
| Ecuación 38. Definición de decodificación de Ambisonics.                                                            | 55 |
| Ecuación 39. Definición del método <i>Overlap and Add</i> .                                                         | 56 |

## ÍNDICE DE TABLAS

|                                                                                                                             |     |
|-----------------------------------------------------------------------------------------------------------------------------|-----|
| Tabla 1. Armónicos esféricos $Y_n^m \theta, \phi$ para $n = 0, \dots, 4$ [EP].                                              | 17  |
| Tabla 2. Ilustración de las funciones representadas en la Figura 13                                                         | 18  |
| Tabla 3. Descripción de los altavoces en la configuración 5.1 de la ITU.                                                    | 27  |
| Tabla 4. Definición de Ambisonics formato B en relación con los armónicos esféricos [EP].                                   | 35  |
| Tabla 5. Relaciones entre orden de Ambisonics y Wigner – D.                                                                 | 50  |
| Tabla 6. Matriz de confusión obtenida para los resultados de una medición de altavoces situados en un sistema cuadrafónico. | 77  |
| Tabla 7. Matriz de confusión para un sistema cuadrafónico con posiciones cardinales.                                        | 78  |
| Tabla 8. Matriz de confusión para un sistema cuadrafónico con posiciones cardinales, señal de voz.                          | 79  |
| Tabla 9. Matriz de confusión para un sistema cuadrafónico con posiciones cardinales, señal musical.                         | 80  |
| Tabla 10. Matriz de confusión para un sistema cuadrafónico, señal sinusoidal.                                               | 81  |
| Tabla 11. Matriz de confusión para un sistema cuadrafónico, señal de voz.                                                   | 82  |
| Tabla 12. Matriz de confusión para un sistema cuadrafónico, señal musical.                                                  | 83  |
| Tabla 13. Matriz de confusión para un sistema 7.1.4, señal sinusoidal.                                                      | 84  |
| Tabla 14. Matriz de confusión para un sistema 7.1.4, señal de voz.                                                          | 85  |
| Tabla 15. Matriz de confusión para un sistema 7.1.4, señal musical.                                                         | 87  |
| Tabla 16. Matriz de confusión para un sistema cuadrafónico con posiciones cardinales, señal de voz.                         | 88  |
| Tabla 17. Matriz de confusión para un sistema cuadrafónico con posiciones cardinales, señal musical.                        | 89  |
| Tabla 18. Matriz de confusión para un sistema cuadrafónico.                                                                 | 89  |
| Tabla 19. Matriz de confusión para un sistema cuadrafónico, señal musical.                                                  | 90  |
| Tabla 20. Matriz de confusión para un sistema 7.1.4, señal de voz.                                                          | 92  |
| Tabla 21. Matriz de confusión para un sistema 7.1.4, señal musical.                                                         | 93  |
| Tabla 22. Resultados de la rotación de todos los canales 7.1.4 hacia el plano frontal.                                      | 96  |
| Tabla 23. Resultados de la rotación de todos los canales 7.1.4 hacia el plano frontal, grabación de voz.                    | 97  |
| Tabla 24. Resultados de la rotación de todos los canales 7.1.4 hacia el plano frontal, clip musical.                        | 98  |
| Tabla 25. Resultados de la rotación de todos los canales 7.1.4 hacia el plano frontal, grabación de voz.                    | 99  |
| Tabla 26. Resultados de la rotación de todos los canales 7.1.4 hacia el plano frontal, clip musical.                        | 99  |
| Tabla 27. Presupuesto económico del Proyecto.                                                                               | 111 |

# 1. INTRODUCCIÓN Y OBJETIVOS

## 1.1. INTRODUCCIÓN

El auge del consumo de contenido multimedia, unido al continuo desarrollo de tecnologías por parte de distintas instituciones y entidades, ha impulsado la aparición de nuevos conceptos y elementos que forman parte de la experiencia de usuario. Es por ello por lo que el sector de audio ha contribuido activamente en el avance de sus tecnologías para generar nuevos sistemas que permitan potenciar la vivencia audiovisual. Además, el fácil acceso por parte de los consumidores a este tipo de sistemas hace que su exigencia se incremente a medida que aparecen nuevas aplicaciones en el mercado.

Uno de los principales responsables del avance de la Ingeniería de Audio es el grupo *MPEG*, que con sus múltiples aportaciones se ha conseguido un Estado del Arte muy competente en lo referido a los sistemas de sonido, los cuales se enfocan tanto en el procesamiento como en el almacenamiento, transmisión, compresión, codificación.... Así surgió, por ejemplo, el formato de audio digital que revolucionó la manera de consumir, principalmente, música: el *mp3*. Este sistema, al igual que ocurre con muchos otros que llegarían después, constituye una de las capas de un esquema global que define distintas partes dentro del tratamiento de señales audiovisuales. Desde esta extensión de audio hasta *MPEG – H*, en la cual nos hemos basado en este Trabajo, se han desarrollado alrededor de diez sistemas con diversas aplicaciones, lo cual confirma la importancia del Grupo y de su implicación en el sector.

El estudio de las ondas sonoras en el espacio abierto llevado a cabo por el Boaz Rafaely [1] ha influido positivamente en el establecimiento de las bases matemáticas que definen su naturaleza y comportamiento. En su libro, *Fundamentos del Procesado de Arrays Esféricos*, el Ingeniero se centra en las funciones definidas sobre la esfera, que se traducen en fuentes sonoras emplazadas en el espacio. Así, podremos localizar y parametrizar las posiciones fijas y móviles (a través de la rotación) de un conjunto de altavoces mediante los armónicos esféricos, una definición matemática de funciones expresadas en coordenadas esféricas. Esto será de gran utilidad, ya que nos permitirá caracterizar las fuentes en un formato rompedor denominado *Ambisonics*.

Existen múltiples configuraciones de audio en el mercado, desde aquellas que implican el uso de un solo canal (ya apenas utilizadas), hasta otras con tantas señales como altavoces posea el usuario. La evolución de todos los sistemas ha estado siempre motivada por la mejora de la experiencia auditiva y del realismo con respecto al espacio sonoro. Desde esquemas recomendados por la *ITU* y perfeccionados por fabricantes como Dolby, como en el caso del conocido 5.1, hasta el invento de un conjunto de Ingenieros y Matemáticos británicos, se ha buscado la inmersividad y la realidad virtual. Gracias a los últimos nace *Ambisonics*, el sistema de sonido envolvente y tridimensional adaptable a distintas configuraciones comenzando por cuatro altavoces en su primer orden. En el caso del tercero, se obtienen un total de 16 altavoces, los cuales se pueden transformar en un sistema estandarizado que incluya un menor número de fuentes.

Con tal de encontrar una funcionalidad práctica y demostrable, el sonido generado mediante *Ambisonics* y transformado a las posiciones de altavoces, que en nuestro Trabajo serán las definidas en 7.1.4, se volverá a transformar a otra configuración innovadora: el sistema *biaural*. Este consiste en la recreación de un sonido tridimensional, de forma que el oyente perciba el entorno físico de la grabación de origen mediante el uso de auriculares. Para conseguir este efecto junto con la transformación de las fuentes sonoras, es necesario obtener las respuestas al impulso del oyente, las denominadas *HRTF*. Mediante la grabación tridimensional de impulsos sonoros se puede conseguir no sólo la respuesta en relación con la cabeza, sino el impulso de la sala, ofreciendo así la sensación de espacialidad comentada.

Toda esta teoría resulta muy interesante si, además, tenemos en cuenta que su aplicación es posible gracias a la elaboración de sistemas mediante entornos de programación. Así, se trata de plasmar de manera perceptible el conjunto de efectos sonoros mencionados mediante el uso de un módulo conectable en una estación de trabajo. Una vez analizados los resultados obtenidos, se concluirá si, efectivamente, el efecto generado es el esperado.

## 1.2.OBJETIVOS

La principal meta de este Trabajo es el estudio, desarrollo e implementación de una cadena procesado de audio multicanal y envolvente que culmine con la presentación de una señal de audio de dos canales, denominada binaural. Para alcanzar dicho propósito, se deberán alcanzar los siguientes objetivos:

- El estudio y conocimiento de la base matemática que sustenta la definición de ondas sonoras, de funciones armónicas esféricas y de las operaciones que pueden aplicarse; del paradigma de los sistemas de audio y de las tecnologías empleadas para la generación del sonido tridimensional.
- El desarrollo de un sistema que implique un alto grado de procesamiento de audio en el lenguaje de programación más competente hoy en día: Python.
- La construcción de un módulo conectable, *Plug – In*, que incluya todas las funcionalidades de manera que pueda ser utilizado por cualquier usuario en tiempo real.
- La generación de respuestas al impulso mediante un hardware adquirido para tal uso, de forma que el resultado binaural sea competente y funcione correctamente.
- La comprobación del funcionamiento del sistema en un entorno real de trabajo, más concretamente, en una Estación Digital de Audio.

## 2. TEORÍA Y ESTADO DEL ARTE

La forma en la que nos comunicamos hoy en día, unido al cambio en la manera en la que consumimos contenido multimedia y la aparición de sistemas inmersivos, ha dado lugar a una gran evolución en el desarrollo de los diferentes sistemas de sonido. Para comprender correctamente la tecnología utilizada en este trabajo de investigación, es necesario en primer lugar situar el Estado del Arte de aquellas precedentes y las que actualmente son más utilizadas.

### 2.1. INTRODUCCIÓN A LA CODIFICACIÓN DE AUDIO

No podríamos hablar de codificación de audio sin la principal organización dedicada a la estandarización de tecnologías de audio y vídeo, *Moving Picture Experts Group (MPEG)*, la cual pertenece a la *ISO/IEC*<sup>1</sup>. Desde su fundación en 1988 por Hiroshi Yasuda y Leonardo Chiariglione, el Grupo ha desarrollado numerosos sistemas de compresión, procesado y codificación para la transmisión de señales de naturaleza multimedia, tanto de manera conjunta como independiente. Si bien no es el propósito de este trabajo explicar todas las tecnologías desarrolladas, se considera necesario establecer una base conceptual acerca de las más importantes para poder así indagar en más profundidad en aquella en la que implementación se basa nuestra implementación: MPEG-H.

En primer lugar, la organización de cada uno de los estándares se basa en *Partes*, de manera que cada una de ellas se refiere, como su nombre indica, a las distintas secciones del desarrollo: sistema general, vídeo, audio, testado, simulación, extensiones... Así, las señales se tratan y procesan por separado, pues su transmisión se lleva a cabo de igual manera. El Grupo de expertos ha conseguido construir más de veinte tecnologías, cada una de ellas adaptada al momento y a los recursos disponibles entonces. En relación con el audio, no todos los estándares aportan nuevas implementaciones, sino que aprovechan las ya presentes en el resto y las integran si cumplen los requisitos técnicos. Es el caso de MPEG-1, MPEG-2, MPEG-4, MPEG-7, MPEG-D, y MPEG-H.

### 2.2. CODIFICACIÓN DE AUDIO EN MPEG

#### 2.2.1. MPEG-1 PARTE 3: AUDIO

El primer codificador de audio digital nace en 1993, en la Parte 3 del estándar MPEG-1 (ISO/IEC 11172) y en tres diferentes Capas, o *Layers*. Las Capas I y II se utilizaban principalmente para la transmisión de vídeo y audio y, en ocasiones, para su almacenamiento y aplicaciones profesionales. Los valores de *bit rate* están en el rango entre 32 y 448 kbits/s, y las frecuencias de muestreo posibles son 32, 44.1 y 48 kHz. El sistema detrás de estas dos capas se basa en una codificación subbandas, de forma que las señales entrantes se dividen en 32 subbandas distribuidas a lo largo de la frecuencia. Este proceso es posible gracias a un banco de filtros que utiliza QMF (Quadrature Mirror Filter), tras el cual se lleva a cabo la división. Previo al proceso de decodificación mediante QMF inverso, se generan los cuadros de asignación o *allocation frames*, y se cuantifican y codifican mediante Adaptive Pulse Code Modulation. La Figura 1 muestra el esquema del sistema explicado.

<sup>1</sup> ISO/IEC: International Organization for Standardization / International Electrotechnical Commission

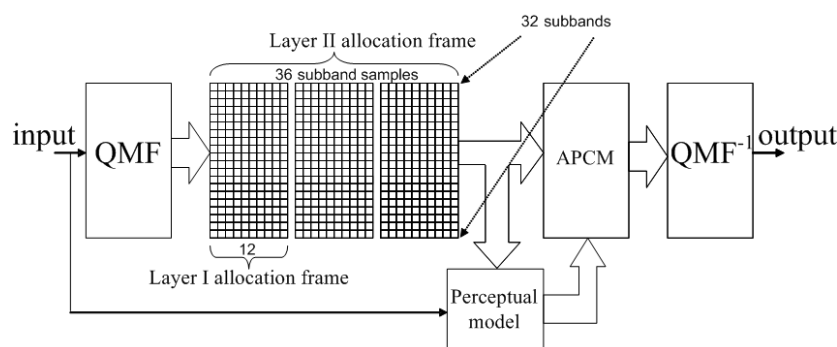


**Figura 1.** Codificador de audio de las Capas I y II de MPEG-1, Parte 3 [2].

MPEG-1 también desarrolló una tercera la capa, más conocida como *mp3*. Su importancia viene dada por el hecho de que este formato ofrece una gran capacidad de compresión, hasta de un factor 12, lo que hace que sea óptimo para el almacenamiento y transmisión de señales musicales tanto mono como estéreo. La Capa III cambió la manera en la que consumimos música y su invención llevó consigo otras como el reproductor mp3 y la transmisión vía satélite y en redes celulares. Las frecuencias de muestreo de esta capa son las mismas que las utilizadas en las capas I y II: 32, 44.1 y 48 kHz. También comparte con las capas inferiores el banco de filtros, de forma que el tamaño de *frames* sea el mismo para las tres.

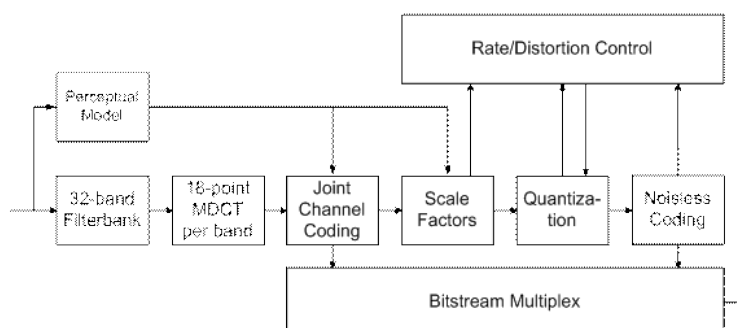


**Figura 2.** Codificador de audio de la Capa III de MPEG-1, Parte 3 [2].

En la Figura 2, vemos que el esquema introduce conceptos como “Modelo Perceptual” y “Codificación de canal conjunta”. El primero utiliza las propiedades del sistema auditivo humano y codifica la señal de audio en función del rango de frecuencia sensible al oído. Este rango se encuentra entre los 20 Hz y los 20 kHz, pero principalmente existe una mayor percepción en medias frecuencias. Para las altas, la imagen estéreo depende de la diferencia en los niveles de intensidad. Así, la salida de este bloque entrega valores de enmascaramiento umbral (o *threshold*) para determinar la cantidad de ruido soportable para que la compresión sea eficiente. El segundo, “*Joint Channel Coding*”, explota las características de las señales de audio estéreo, es decir, que ambos canales contienen la misma información y las redundancias pueden ser eliminadas para reducir el *bitrate* [2]. Veremos más adelante cómo la tecnología de audio ha avanzado hasta sistemas donde los pares estéreo no contienen necesariamente la misma información, sino que es diferente para crear sonidos envolventes, 3D, *surround*...

## 2.2.2. MPEG-2 PARTES 3 Y 7: AUDIO Y ADVANCED AUDIO CODING

El estándar MPEG-2 (ISO/IEC 13818) nace en 1995 con el propósito de establecer un sistema de transmisión de señales de alta definición sobre redes de cable, satélite y terrestres, así como el almacenamiento en discos ópticos (*Digital Versatile Disc, DVD*). La principal diferencia con respecto a MPEG-1 es la variedad de aplicaciones en las que MPEG-2 se puede utilizar, ya que se adapta a distintas resoluciones, calidades y velocidades. En lo que al audio se refiere, aparecen dos especificaciones: parte 3 y parte 7. La parte 3 corresponde al esquema desarrollado en MPEG-1, e introduce un sistema de compatibilidad hacia atrás. Esto significa que un decodificador MPEG-2 es capaz de decodificar una señal de audio de MPEG-1, pero no al revés. También es importante apuntar que permite menores frecuencias de muestreo para señales mono y estéreo, sin embargo, soporta audio multicanal para dichas frecuencias (32, 44.1 y 48 kHz) [3].

La novedad aparece en la parte 7, o más conocida como *Advanced Audio Coding, AAC*. Esta implementación surge de la necesidad de romper con la compatibilidad del estándar anterior, ya que los avances en la compresión de audio no permitían seguir usando dicho esquema. La razón de tener que elaborar un estándar de audio con nuevas características reside en que se requería una tecnología capaz de transmitir señales de cinco canales a un *bit rate* de 320 kbps en la Unión Europea de Radiodifusión (*European Broadcast Union, EBU*). Así, la principal característica de AAC es que ofrece una alta compresión y calidad de la señal a una tasa binaria muy inferior. Además, introduce la capacidad de transmitir y almacenar señales de audio no solo mono o estéreo, sino multicanal.

La codificación se basa en tres tácticas principales: eliminación de redundancia, reducción de información y correspondencia de entropía entre los valores de la señal y el *bitstream*<sup>2</sup>. La primera se consigue mediante el análisis estadístico de la señal; la segunda utiliza un modelo basado en el sistema auditivo para establecer un límite en el ruido de cuantificación y la tercera se consigue mediante la codificación de entropía. Al igual que en MPEG-1, incluye herramientas de codificación conjunta para señales de cualquier naturaleza.

El lanzamiento de la Codificación Avanzada de Audio, AAC, tuvo una gran acogida por parte de la industria para su aplicación a distintos fines. Como ya se ha mencionado, se utiliza para la transmisión de señales terrestres, así como en el *streaming* de contenido multimedia comprimido tanto en Internet como en telefonía móvil. También es la base para el desarrollo de una tecnología posterior que se utiliza en emisión por satélite y móvil: *HE-AAC*<sup>3</sup> [4].

## 2.2.3. MPEG-4 PARTE 3: AUDIO

Este estándar nace en 1998 y en él aparecen modificaciones con respecto al anterior, aportando así una mayor eficiencia en la codificación y la aparición de diversas aplicaciones. Se introducen nuevos conceptos y tecnologías como las codificaciones sinusoidal y paramétrica, la replicación de bandas espectrales, la codificación sin pérdidas y escalable y la MPEG-4 *Structured Audio*.

La SSC, *Sinusoidal Coding*, se basa en la asunción de que una señal de audio se puede describir mediante tres componentes principales: una señal transitoria (suma de eventos definidos), una determinista (suma de componentes sinusoidales) y una señal ruidosa. Con esta nueva tecnología, se deja atrás la codificación de formas de onda, la cual había llegado a su máxima explotación técnica. La representación paramétrica ofrece unas mayores calidades de escalado del *tempo* y del tono en el decodificador sin coste adicional. Esta

<sup>2</sup> Bitstream: flujo de bits.

<sup>3</sup> High Efficiency – Advanced Audio Coding.

tecnología tiene diversas aplicaciones: telefonía móvil, descarga de archivos musicales y videojuegos. [5]

MPEG-4 SBR, *Spectral Band Replication*, es una herramienta utilizada adicionalmente en AAC y es muy eficiente, ya que reduce la tasa de bit y aumenta la calidad del audio. Su utilización da lugar a dos nuevos estándares: *High Efficiency AAC* y *High Efficiency AAC v2*. En rangos generales, el procedimiento de la replicación de bandas espectrales se basa en replicar las bajas frecuencias del espectro de la señal, codificar una representación paramétrica de dicha banda y utilizarla en el proceso de decodificación. La razón por la que se desestima la banda de frecuencias superiores es que se reduce el *bitrate* y, con ello, el ruido espectral añadido no es percibido y el sonido se escucha de manera más limpia. Para obtener una representación válida de las altas frecuencias a partir de las filtradas mediante paso-bajo, el codificador de Replicación de Bandas Espectrales utiliza datos de control y extrae información de la forma de onda a partir de la señal original. Además del codificador SBR, en paralelo se codifica la señal mediante AAC y a la mitad de la tasa de bit. En la parte del decodificador se utiliza QMF y la salida se obtiene gracias a la combinación ya empleada en el codificador. En la Figura 3 se muestran los esquemas de codificación y decodificación.

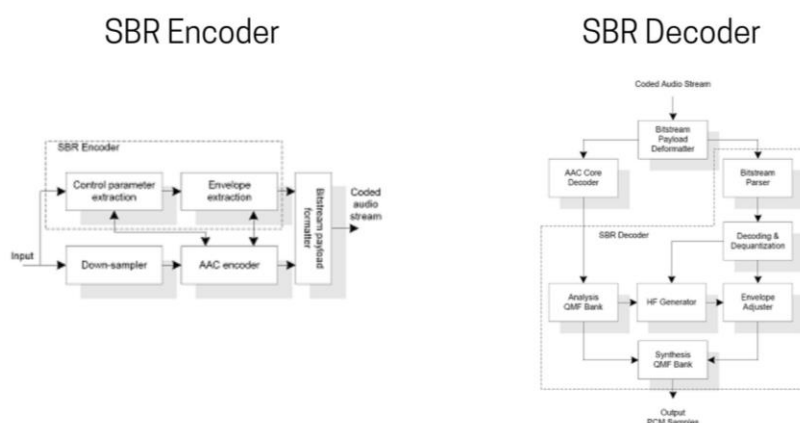


**Figura 3.** Codificador y decodificador de MPEG-4 Spectral Band Replication [5].

Gracias a esta técnica, se encuentra una solución a aquellas aplicaciones que no tienen capacidad para codificar correctamente toda la banda de señal, como las que implican transmisión en tiempo real.

Otra de las numerosas novedades que aportó MPEG-4 fue la codificación sin pérdidas, o *Audio Lossless Coding (ALC)*, la cual ha facilitado la interoperabilidad entre distintos entornos y plataformas y su uso en almacenamiento y producción. El algoritmo utiliza Codificación Lineal Predictiva (*LPC*) hacia delante, de forma que una predicción de la señal basada en muestras anteriores se transmite junto con coeficientes cuantificados del filtro y el proceso inverso es llevado a cabo en la recepción de la señal para su presentación. Una versión avanzada de ALC es la codificación sin pérdidas escalable, *Scalable Lossless Coding*, utilizada en transmisión de señales de audio además de en su almacenamiento. Lo importante a señalar en la aplicación para *broadcasting* es que se requiere que el *bitrate* se adapte a las condiciones de la red, ya que puede variar según la transmisión y la aplicación. La codificación sin pérdidas consigue reducir a la mitad la tasa de bit para señales estéreo de calidad estándar (768 kbps a 48 Hz y 16 bits) y para señales con mayor resolución la tasa de bit aumenta (hasta 4000 kbps a 192 kHz y 24 bits). Sin embargo, observamos que hay un amplio intervalo de tasas de bits para las cuales no se ha definido una compresión válida. MPEG-4 SLS ofrece un mayor rango de posibilidades entre la codificación transparente (o con

pérdidas) y la codificación sin pérdidas. Por último, es considerable recalcar el hecho de que esta codificación aporta una reducción de tasa de bit y una compresión de un ratio 2:1 sin pérdidas, donde otras tecnologías no pueden asegurar dichos valores.

Por último, MPEG-4 introduce el concepto de *Structured Audio*. Este se refiere a un modelo de esquema de codificación creado a partir de información de alto nivel de los sonidos a representar, por ejemplo, la interfaz MIDI<sup>4</sup> y los modelos de predicción lineal de la voz. MPEG-4 SA se utiliza, sobre todo, para la transmisión de contenido de audio a una tasa de bit muy baja, en la síntesis mediante *wavetable*<sup>5</sup> (que incluye componentes de audio para páginas web, entre otros), en el procesado de efectos de sonido y en la codificación generalizada [5].

#### 2.2.4. MPEG-D PARTES 1 Y 2: *SURROUND Y SPATIAL AUDIO OBJECT CODING*

El estándar MPEG-D, *ISO/IEC 23003*, también especificado como MPEG Audio Technologies, define en sus partes el audio espacial y su codificación y las codificaciones conjuntas de voz y audio, principalmente. Así, el Grupo lleva desde 2005 desarrollando las tecnologías de codificación de audio hasta la actualidad. Aunque nuestro trabajo se basa en la parte de audio del estándar MPEG-H, consideramos muy necesario detallar previamente el dedicado exclusivamente a la codificación de audio, MPEG-D.

En la Parte 1 se centran en el formato de compresión de audio espacial, más comúnmente conocido como audio *surround*. Los motivos que dieron lugar al desarrollo de dicho estándar son principalmente dos: la codificación para aplicaciones de baja tasa de bit y la aparición de sistemas de reproducción multicanal. Aunque ya existían métodos de almacenamiento de información de más de dos canales en sistemas estéreo, como los DVD y los CD, estos podían permitirse un gran ancho de banda y unas condiciones de tasa y profundidad de bit muy favorables. Sin embargo, MPEG-D *Surround* se centra en la transmisión en entornos con limitaciones en la capacidad de red, como ya se ha mencionado previamente. Utiliza y combina distintas técnicas, como la codificación paramétrica, la codificación binaural y tecnologías Dolby. El resultado es la presentación mono o estéreo de señales multicanal sin necesidad de manipular valores de diferencias de fase entre canales para obtener información espacial.

La diferencia entre otras codificaciones, como MP3, y MPEG *Surround* es que la primera explota la capacidad de cada canal de enmascarar el ruido de cuantificación asociado, mientras que la segunda saca provecho de la capacidad del sistema auditivo humano de percibir el sonido de manera dimensional. Es aquí donde profundizaremos en conceptos muy importantes para la construcción de sonido envolvente o espacial: las diferencias inter-aurales de nivel y de tiempo y la coherencia inter-aural. La forma de la cabeza y de las orejas, así como la localización de la fuente sonora son los principales elementos que dan lugar a los términos siguientes.

<sup>4</sup> MIDI: Musical-Instrument Digital Interface.

<sup>5</sup> Wavetable: es una técnica para producir música de manera digital.

## PARÁMETROS DE LOCALIZACIÓN DEL SONIDO EN EL OÍDO HUMANO

- Diferencia inter-aural de intensidad, DII (*Inter-aural Level Difference, ILD*): es la variación en la intensidad y la frecuencia sonoras entre los dos oídos. Esto ocurre porque la cabeza actúa como una sombra acústica, término que define el área de la cabeza opuesta a la fuente de sonido y que modifica el volumen debido al bloqueo de ondas sonoras, es decir, produce una difracción de la onda. [6] Las altas frecuencias son más afectadas por la sombra acústica que las bajas. Recordemos que la frecuencia y la longitud de onda son conceptos relacionados entre sí mediante

$$f = \frac{c}{\lambda} \text{ Hz},$$

**Ecuación 1.** Relación entre frecuencia y longitud de onda <sup>6</sup> [EP]<sup>7</sup>.

de forma que se cumple que, a bajas frecuencias (por debajo de 1000 Hz), las longitudes de onda son relativamente grandes con respecto al tamaño de la cabeza. Esto hace que la onda se difracte fácilmente y que la sombra acústica generada sea pequeña, al contrario que a altas frecuencias (o bajas longitudes de onda), donde la difracción es menor y por tanto la onda que llega al oído contrario a la posición de la fuente ve disminuida su intensidad. Concluimos entonces que la cabeza funciona como un filtro paso bajo [7]. En la Figura 4 [8] se relaciona la frecuencia y las Diferencias Inter-aurales de Intensidad con el ángulo acimut que forma la fuente y la posición de la cabeza del oyente.

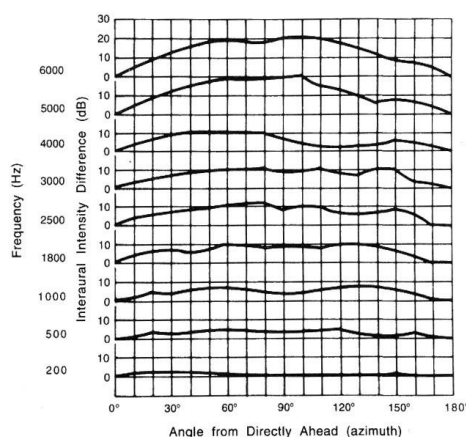


**Figura 4.** Representación del ángulo lateral en función de las Diferencias Interaurales de Intensidad.

El oído humano utiliza la intensidad para detectar sonidos cuyas frecuencias están comprendidas entre 300 Hz y 5 kHz.

<sup>6</sup> En la ecuación,  $c$  es la velocidad del sonido (343 m/s) y  $\lambda$  es la longitud de onda, medida en metros.

<sup>7</sup> Todas las ecuaciones incluidas en esta Memoria se han introducido manualmente por la autora de este Trabajo de Fin de Máster.

- Diferencia inter-aural de tiempo (*Inter-aural Time Difference, ITD*): es el intervalo de tiempo transcurrido desde que el sonido llega a cada oído. El sistema auditivo humano se caracteriza por tener una alta sensibilidad, ya que es capaz de detectar diferencias de milisegundos. Gracias a dicha precisión, somos capaces de situar un objeto en el espacio (más concretamente, en el ángulo acimut). [9]

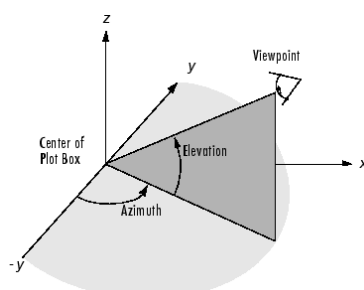


**Figura 5.** Sistema de coordenadas con los ángulos acimut y elevación <sup>8</sup>.

Existe una equivalencia entre la diferencia de tiempo y de fase entre las ondas sonoras que llegan a cada oído cuando éstas son tonos puros:

$$\Delta t = \frac{\Delta \Phi}{2\pi f}$$

**Ecuación 2.** Relación entre diferencias temporales y de fase para señales senoidales.<sup>9</sup>

Nos encontramos de nuevo con una relación frecuencia – longitud de onda. A bajas frecuencias, la diferencia de fase es tan baja que no es posible percibirla. A frecuencias más altas, la longitud de onda disminuye y la diferencia de fase aumenta y, a partir de 800 Hz, el sistema auditivo se encuentra ante cierta ambigüedad. Para ese valor, la longitud de onda es aproximadamente dos veces la medida de la cabeza y las diferencias de fase son de 180°, pero no es posible detectar qué onda está desfasada con respecto a otra. Si bien este mecanismo es útil para localizar una fuente para frecuencias entre 200 y 800 Hz, deja de ser útil a partir de frecuencias de 1.5 kHz [10]. En la Figura 6, vemos que se alcanza el mayor valor de diferencia de tiempo (0.7 ms) para un ángulo de 90°. Si la fuente está situada a la derecha del oyente, donde el valor del ángulo acimut es 0°, la *ITD* marca 0 segundos.

<sup>8</sup> Los ángulos elevación y acimut se denominarán theta ( $\Theta$ ) y phi ( $\Phi$ ), respectivamente, durante el desarrollo de este Trabajo.

<sup>9</sup> Nótese que, en este caso,  $\Delta \Phi$  no se refiere al ángulo elevación, sino a la fase de la señal.

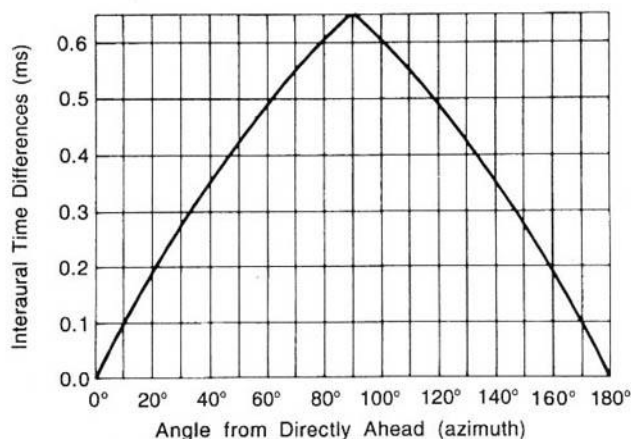


**Figura 6.** Representación del ángulo lateral en función de las Diferencias Inter-aurales de Tiempo [8].

Gracias a Lord Rayleigh, quien contribuyó significativamente al estudio de las ondas y su comportamiento sobre superficies y a través de ellas, podemos calcular de manera sencilla la diferencia de tiempo inter-aural mediante la Ecuación 3. Cálculo de la Diferencia Inter-aural de Tiempo. En la figura situada a la derecha se pueden observar las correspondencias en cuanto a distancia a la fuente, acimut y separación entre oídos. Aplicando la fórmula y asignando los valores precisos, observamos que se cumple que, para una fuente situada a  $\theta = 0^\circ$ , la diferencia de tiempo en llegar en un oído y otro es nula [7].

$$ITD = \frac{a}{c} (\theta + \sin \theta), \quad -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}$$

**Ecuación 3.** Cálculo de la Diferencia Inter-aural de Tiempo.

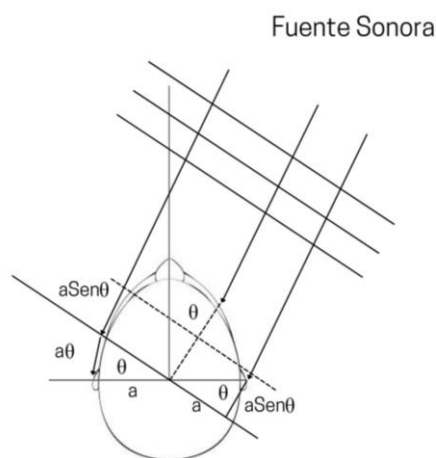


**Figura 7.** Esquema de representación de la ILT [EP]<sup>10</sup>.

- Coherencia inter-aural (*Inter-aural Coherence, IC*): este término nace de la naturaleza acústica del espacio sonoro y del principal efecto causado por el mismo: la reverberación. Un entorno es reverberante si las ondas experimentan un cambio de sentido y orientación en su trayectoria por encontrarse con obstáculos (objetos, paredes, ...), de manera que el oyente percibe la onda original junto con una copia coherente, retardada, y atenuada. El concepto de Coherencia Inter-aural recoge la medida de la similitud entre las señales reverberantes que llegan a cada oído. Se

<sup>10</sup> Todas las figuras y tablas designadas mediante [EP] se han elaborado de manera propia por la autora de este Trabajo de Fin de Máster.

considera muy importante en el audio baural, ya que éste debe incluir la acústica de la sala para crear una sensación espacial donde las fuentes están correctamente posicionadas [11].

En la siguiente imagen se pueden observar los tres fenómenos previamente explicados:

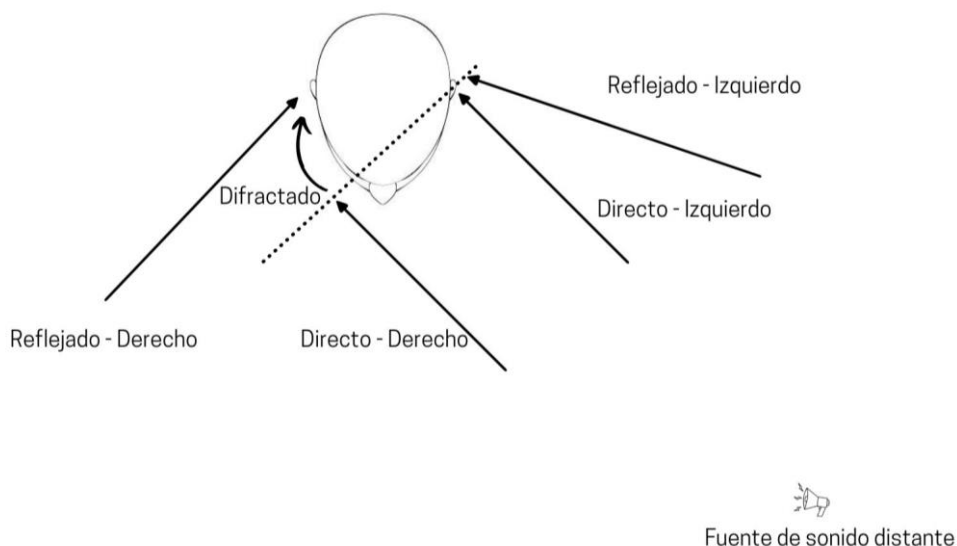


**Figura 8.** Ilustración de los fenómenos ILD, ITD e IC [EP].

Una vez introducidos los conceptos de *ILD*, *ITD* e *IC*, continuamos con la tecnología *MPEG-D Surround*. El aspecto clave de esta tecnología es la explotación de las diferencias de nivel, fase y coherencia entre los canales para obtener una “imagen” espacial. Este proceso de extracción de parámetros tiene lugar en el codificador. A su salida se obtienen dos señales, también denominadas mezcla o *downmix*, resultado de un proceso de codificación de las señales de entrada utilizando los mecanismos ya expuestos previamente (como *MPEG-1 Capa III*, *AAC* o *HEAAC*). Estos dos canales se transmiten, previamente comprimidos, y se decodifican en el receptor. La tecnología *Surround* permite una presentación tanto multicanal como estéreo, siendo más aconsejable la segunda debido al gran número de aplicaciones en las que se usan auriculares en la reproducción. Sin embargo, también se utiliza esta tecnología mediante salida multicanal, como en *Dolby Surround*.

En cuanto a los parámetros técnicos, es importante mencionar que gran parte del éxito de este mecanismo se debe a la alta calidad que ofrece a tasas de bits muy bajas, hasta 48 kbps. Otro factor es la gran demanda de aplicaciones en las que se utiliza, como teleconferencias, servicios de *streaming* de música (donde el ancho de banda está muy acotado), retransmisión de televisión digital, almacenamiento de archivos de audio, videojuegos y, en los últimos años, entornos de realidad inmersiva y virtual [12].

En la parte 2 de *MPEG-D* se detalla la Codificación de Objetos de Audio Espacial, *Spatial Audio Object Coding (SAOC)*. El concepto de Codificación de Objetos se refiere a la codificación independiente de elementos en el espacio sonoro de manera que pueden ser ajustados y controlados en función de las necesidades del usuario. El estándar llama a los objetos pistas, o *tracks*, y busca elaborar una codificación paramétrica de cada una de ellas. Además, aporta la funcionalidad de manipular los objetos para presentarlos en una escena sonora basada en distintos sistemas de reproducción (multicanal para altavoces y baural para auriculares).

Al igual que en *Surround*, SAOC también utiliza parámetros de percepción espacial para llevar a cabo su transformación y codificación. En este caso, se utilizan los conceptos siguientes: Diferencias de Intensidad de Objetos (*Object Level Differences, OLD*), Coherencias Cruzadas Entre Objetos (*Inter-Object Cross Coherences, IOC*) y Diferencias de Intensidad en los Canales de la Mezcla (*Downmix Channel Level Differences, DCLD*). Esta información se extrae de los *tracks* y se codifica en un flujo de bits que se envían al decodificador SAOC y el cual genera una representación MPEG-D *Surround*. Esta es dirigida a un decodificador *Surround* y es entonces cuando se genera la escena de audio deseada.

Es una tecnología muy potente, ya que tiene tantas aplicaciones como el usuario quiera implementar. Se utiliza con frecuencia en mesas de mezclas, en entornos donde se requiere un bajo coste computacional (como videojuegos), en producción musical y cinematográfica y en infraestructuras de telecomunicaciones [13].

Por último, MPEG-D incluye un estándar denominado *Unified Speech and Audio Coding*, lanzado en 2011. Es importante ya que surgió como necesidad de adaptarse a un nuevo paradigma en el que el consumo de contenido sonoro empezaba a tener lugar en un solo dispositivo, tendencia que continúa y sigue creciendo en nuestros días. Hasta entonces, el contenido, voz y música, se solía codificar por separado por tratarse de señales independientes. Con esta nueva tendencia se consigue una mayor calidad en la reproducción incluso a tasas de bit tan bajas como 8 kbps para señales mono y 32 kbps para estéreo. Además, en aquellos casos en los que el *bitrate* es mayor, de 64 kbps, se consiguen niveles de compresión altos. Esta innovación en la codificación conjunta de señales de voz y música se ha conseguido unificando técnicas previas desarrolladas también por MPEG en anteriores *códecs* de audio con técnicas perceptuales basadas en el modelo de generación de voz humano [14].

Las principales aplicaciones son aquellas que, una vez más, implican una tasa de bit muy baja ya sea en la transmisión o en el almacenamiento. Ejemplos de ello son la descarga o consumo en tiempo real de contenido en dispositivos móviles (música, cine, audiolibros...). Otra aplicación importante, aunque no sea destinada para el usuario final, será su uso en el estándar de MPEG para audio inmersivo: MPEG-H.

#### 2.2.5. MPEG-H: 3D AUDIO

Tras una detallada explicación de las principales codificaciones de audio MPEG, llegamos al estándar que ha motivado la elección y elaboración de este Trabajo de Fin de Máster: MPEG-H (ISO/IEC 23008), *High Efficiency Coding and Media Delivery in Heterogeneous Environments*, nacido en 2015. En sus partes I y II se definen, respectivamente, el Transporte de Medios de MPEG (*MPEG Media Transport*) y la Codificación de Vídeo de Alta Eficiencia (*High Efficiency Video Coding*). Nos centraremos en la parte dedicada al audio, la parte III, *3D Audio*.

Según define su entidad creadora, el Instituto Fraunhofer IIS, MPEG-H Audio es “una tecnología de audio de nueva generación capaz de aportar más realismo alrededor del oyente en todas las direcciones” [15].



### MPEG-H AUDIO

Figura 9. Logo de MPEG-H Audio.

La principal característica de este estándar es la generación de audio inmersivo, también conocido como 3D o binaural, que aporta dimensionalidad en el plano vertical. Además, la escucha mediante audífonos o auriculares es también posible para percibir la dimensionalidad del sonido. Es muy importante no confundir el sonido inmersivo con el envolvente, ya que el segundo trata de recrear la direccionalidad de las señales mediante altavoces dispuestos en distintas posiciones, únicamente plano horizontal. Es decir, mediante la tecnología de audio 3D percibimos el sonido como si existieran fuentes en cada punto del espacio, mientras que en la tecnología de audio envolvente las fuentes están físicamente posicionadas. Con la creación de MPEG-H, se consigue la posibilidad de ofrecer distintas maneras de presentación obteniendo una experiencia de usuario igualmente óptima para cada una de ellas, ya sea mediante altavoces o auriculares.

El sistema de audio MPEG-H se basa en la codificación de canales, objetos y Ambisonics de Mayor Orden (HOA)<sup>11</sup>. A diferencia de MPEG-D SOAC, donde los objetos podían ser los distintos elementos del espacio sonoro, MPEG-H se refiere a señales que incluyen información sobre su posición. En cuanto a los canales, son las posiciones específicas de los altavoces para la configuración a utilizar y la información HOA contiene las señales del campo sonoro. En el desarrollo de este Trabajo, nos centraremos en dichos elementos, ya que resultan muy potentes a la hora de establecer la información espacial y su simulación para la presentación binaural [16]. En la Figura 10, se detalla el esquema de decodificación con los elementos de presentación o *rendering*. Además, incluye el mezclador (*Mixer*), el Control de Rango Dinámico (*Dynamic Range Control, DRC*) y la presentación binaural hacia auriculares (*Headphones, HP*) y 3D hacia altavoces (*Loudspeakers, LP*).

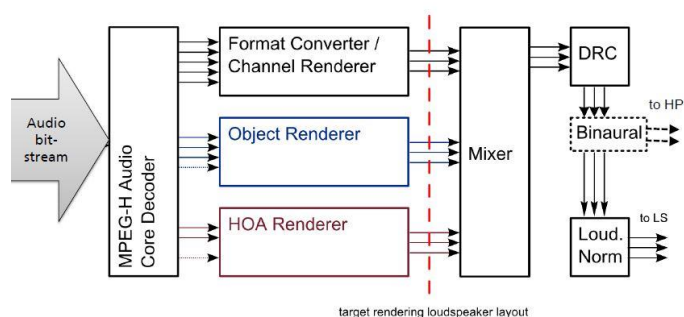


**Figura 10.** Arquitectura básica del decodificador de audio MPEG-H [17].

El codificador principal de MPEG-H se basa en el estándar MPEG-D USAC, visto en el apartado 2.2.4, y mejora su eficiencia mediante herramientas que explotan la naturaleza de distintos parámetros como la “imagen” sonora multicanal, la adaptación automática de *bitrate* y flujo de contenido y la señalización para las configuraciones binaurales. Como se ha mencionado en la introducción, así como en el resumen, nuestro cometido será realizar una presentación binaural basado en una configuración ITU 7.1.4, para la cual se establece una tasa de bit de 504 kbps [17]. Los valores para otras configuraciones descienden a 288 y 180 kbps para ITU 5.1 y 2.0 (estéreo), respectivamente.

El conversor de formato o presentador de canales (*Format Converter/Channel Rendering*) incluye una configuración de reproducción flexible y adaptada al diseño de altavoces y posibilita la inmersión sonora, así como la percepción espacial también en altura. Al igual que

<sup>11</sup> Ambisonics de Mayor Orden: *Higher Order Ambisonics*. Se explicará en la sección 2.5.1.

en MPEG-D, es capaz de generar una mezcla adaptada al entorno y consigue efectos de cancelación de sonido [17].

En cuanto al presentador de objetos (o señales), funciona mediante la combinación de altavoces situados en posiciones cercanas a las deseadas. En nuestro caso, utilizaremos altavoces virtuales mediante la herramienta de rotación virtual, ya que no contamos con el número deseado de dichos dispositivos [17].

Llegamos a una sección importante: el presentador de Ambisonics. Trata de representar el campo sonoro de manera universal y con independencia de los altavoces utilizados. Hace uso de micrófonos multi cápsula, esto es, dispositivos que incluyen un array de transductores. Un ejemplo es el micrófono Zylia, utilizado en este TFM y explicado en la sección 3.1. Por último, sintetiza las señales HOA y son directamente presentadas en los altavoces [17].

El Control de Rango Dinámico se utiliza para adaptar dicho parámetro al dispositivo usado y las condiciones de escucha. Se crean así distintas configuraciones programables al tipo de contenido sonoro y, una vez más, a los altavoces o auriculares [17].

Los presentadores o *renderers* y su elaboración se detallan en la parte de Implementación en la sección 4.

## 2.3.FUNDAMENTOS ANALÍTICOS DEL PROCESADO PARA ARRAYS ESFÉRICOS.

Para desarrollar nuestro sistema de codificación, rotación, decodificación y presentación, se ha utilizado un conjunto de potentes modelos matemáticos empleados en el procesado de sonido y en herramientas de grabación, esto es, en una matriz de micrófonos, o *array microphone*. Es por ello por lo que se considera importante explicar las bases que han dado lugar al resultado completo del Proyecto, siguiendo la guía del investigador Boaz Rafaely, miembro senior del *Institute of Electrical and Electronics Engineers, IEEE*. En su libro, Fundamentos del Procesado de Arrays Esféricos [1], se centra en la generación de campos de sonido tridimensionales y en su tratamiento mediante los ya mencionados *arrays* de micrófonos. El motivo de su investigación es el auge en la demanda de aplicaciones que requieren un tratamiento especial de estas señales: análisis de la acústica de salas, control de ruido, seguridad y defensa, entretenimiento, comunicación de voz y, con especial interés, audio espacial.

Siguiendo la recomendación de Rafaely y la norma MPEG-H 3D Audio, utilizamos para nuestra grabación tridimensional una matriz esférica de micrófonos. El autor establece que los transductores dispuestos en una esfera resultan más ventajosos frente a configuraciones lineales, rectangulares o incluso circulares a la hora de capturar sonido en entornos cerrados como salas de conciertos o de grabación, donde las ondas viajan y generan el campo sonoro espacial.

- Gracias a la naturaleza de la esfera, que ofrece una simetría rotacional completa, se puede elaborar un filtrado espacial y de forma de haz. Esto se traduce en la posibilidad de intensificar o disminuir las fuentes de cada dirección, lo cual se utilizará para la generación de señales *Ambisonics* sintéticas.
- El modelado acústico es más sencillo en el dominio de Fourier esférico, que se puede utilizar gracias a que el procesado es posible en el dominio de armónicos esféricos.
- Se puede implementar la forma de haz (*beamforming*) a partir del patrón de haz completo y centralizado. Así, se consigue un método sencillo y flexible en la realización de matrices [18].

Los *arrays* esféricos muestrean funciones sobre una esfera, como el campo sonoro. Es por ello por lo que, en las siguientes subsecciones, se van a detallar tanto los conceptos teóricos como su utilización en la implementación del sistema de grabación-reproducción.

### 2.3.1. FUNCIONES SOBRE UNA ESFERA

El sistema de coordenadas Cartesiano es aquel definido mediante los ejes  $(x, y, z)$  en el espacio tridimensional de números reales  $\mathbb{R}^3$ . Siendo  $\mathbf{x}$  un vector, decimos que pertenece a dicho sistema mediante:

$$\mathbf{x} \equiv (x, y, z) \in \mathbb{R}^3$$

**Ecuación 4.** Sistema de coordenadas Cartesianas.

Una superficie esférica de radio unidad, denotada mediante  $S^2$ , se puede relacionar con el sistema de coordenadas cartesiano mediante el vector  $\mathbf{x}$  y la norma Euclídea  $\|\cdot\|$ . Si, además, se quieren situar distintas posiciones  $\mathbf{r}$  sobre la esfera en el sistema, contamos con el radio  $r$  y los ángulos elevación y acimut (ya mencionados previamente en este documento),  $\theta$  y  $\phi$ , respectivamente. En la siguiente figura vemos el sistema de coordenadas esférico relacionado con el cartesiano mediante la posición  $\mathbf{r}$ .

$$\mathbf{r} \equiv (r, \theta, \phi)$$

**Ecuación 5.** Denotación de un vector de posición en función de su radio y ángulos elevación y acimut.

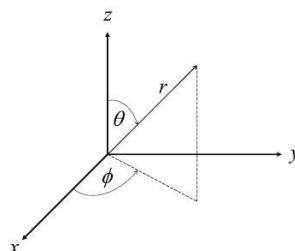


**Figura 11.** Sistema de coordenadas Esférico definido en el sistema de coordenadas Cartesiano.

$$\begin{aligned} x &= r \sin \theta \cos \phi \\ y &= r \sin \theta \sin \phi \\ z &= r \cos \theta \end{aligned}$$

**Ecuación 6.** Relación entre las coordenadas cartesianas y la posición  $\mathbf{r}$ .

Como vemos, el ángulo elevación, también denominado inclinación,  $\theta$  es aquel definido desde el eje  $z$  hacia el plano  $xy$ , mientras que el ángulo acimut  $\phi$  se mide desde el eje  $x$  hacia el  $y$ . Este par de ángulos será de gran utilidad en la implementación del generador sintético de fuentes ya que, asignando la posición de un solo altavoz mediante  $(\theta, \phi)$ , podemos generar un escenario de directividades. Aunque profundizaremos en ello en los apartados de armónicos esféricos y de implementación en *Python* (sección 4.2.2), se ha considerado recomendable establecer un apunte previo a este respecto.

Rafaely se centra en las funciones definidas sobre la esfera y propone una variedad de representaciones gráficas, mostradas a continuación: mapa de colores sobre la superficie de la esfera, mapa de contorno de color sobre el plano  $\theta\phi$  y gráfico de globo.

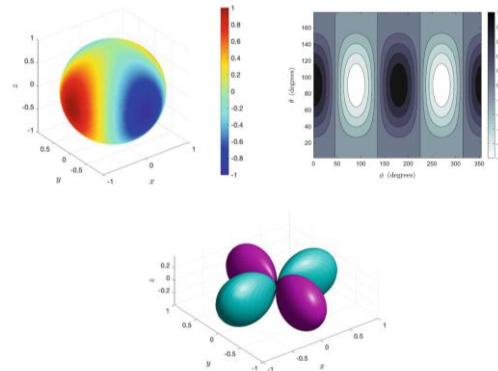


**Figura 12.** Representaciones gráficas de las funciones definidas sobre la esfera [19].

Además, detalla las notaciones utilizadas para representar las mencionadas funciones:  $f(\theta, \phi)$  o  $f(\mu)$ , siendo  $\mu \equiv \mu(\theta, \phi) \in S^2$ . Por último, también se puede representar la superficie esférica en coordenadas Cartesianas, haciendo uso de las relaciones de la Ecuación 6, mediante la siguiente notación:

$$f(x), \quad x = (\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta) \in S^2$$

**Ecuación 7.** Notación de representación de la superficie esférica en coordenadas Cartesianas.

### 2.3.2. ARMÓNICOS ESFÉRICOS

Durante esta sección, se detallarán los aspectos principales de los armónicos esféricos, ya que serán de gran utilidad de cara a la definición de los distintos formatos de Ambisonics. Es por ello que, previamente a la definición de Ambisonics, es necesario determinar las bases matemáticas que la sustentan. Recordemos que las funciones armónicas son aquellas cuyas segundas derivadas parciales satisfacen la ecuación de Laplace ( $\Delta^2 u(x, y) = 0$ ).

Los armónicos esféricos son un conjunto de funciones base que, presentadas mediante su suma ponderada, caracterizan las funciones sobre la esfera de radio unidad. También pueden utilizarse las funciones de Fourier para dicho fin [20].

$$Y_n^m(\theta, \phi) \equiv \sqrt{\frac{2n+1}{4\pi} \frac{(n-m)!}{(n+m)!}} P_n^m(\cos \theta) e^{im\phi}$$

**Ecuación 8.** Definición de armónicos esféricos.

- $Y_n^m(\theta, \phi)$ , armónico esférico.
- $m \in \mathbb{Z}$ , número entero que denota el grado de la función.
- $n \in \mathbb{N}$ , número natural que denota el orden de la función.
- $(\cdot)!$ , función factorial.
- $P_n^m(\cdot)$ , funciones de Legendre asociadas.

Observamos que los ángulos elevación y acimut son determinantes para la generación de estas funciones. Si se desarrolla la definición en la Ecuación 8, se llega a la conclusión de que los armónicos esféricos tienen un exponente complejo dependiente de  $\phi$ , de forma que  $|Y_n^m(\theta, \phi)|$  será constante a lo largo de  $\phi$ . Es por ello por lo que, habitualmente, se representa la parte real e imaginaria de los armónicos esféricos en las gráficas, y no la magnitud y fase. Por otro lado, el orden  $n$  determina la máxima potencia de los términos  $\cos \theta$  y  $\sin \theta$ , controlando así la dependencia de los armónicos esféricos sobre  $\theta$ ; mientras que el orden  $m$  determina la dependencia sobre  $\phi$  en el término exponencial  $e^{im\phi}$ . En la siguiente tabla se desarrollan los armónicos esféricos en para órdenes  $n = 0, 1, 2, 3$ . Veremos la relación y la adaptación de estos resultados a los obtenidos en Ambisonics.

| $y$ | $m$ | $Y_n^m(\theta, \phi)$                                                               |
|-----|-----|-------------------------------------------------------------------------------------|
| 0   | 0   | $Y_0^0(\theta, \phi) = \sqrt{1/4\pi}$                                               |
| 1   | -1  | $Y_1^{-1}(\theta, \phi) = \sqrt{3/8\pi} \sin \theta e^{-i\phi}$                     |
|     | 0   | $Y_1^0(\theta, \phi) = \sqrt{3/4\pi} \cos \theta$                                   |
|     | 1   | $Y_1^1(\theta, \phi) = -\sqrt{3/8\pi} \sin \theta e^{i\phi}$                        |
| 2   | -2  | $Y_2^{-2}(\theta, \phi) = \sqrt{15/32\pi} \sin^2 \theta e^{-2i\phi}$                |
|     | -1  | $Y_2^{-1}(\theta, \phi) = \sqrt{15/8\pi} \sin \theta \cos \theta e^{-i\phi}$        |
|     | 0   | $Y_2^0(\theta, \phi) = \sqrt{5/16\pi} (3\cos^2 \theta - 1)$                         |
|     | 1   | $Y_2^1(\theta, \phi) = -\sqrt{1/8\pi} \sin \theta \cos \theta e^{-i\phi}$           |
|     | 2   | $Y_2^2(\theta, \phi) = \sqrt{15/32\pi} \sin^2 \theta e^{2i\phi}$                    |
| 3   | -3  | $Y_3^{-3}(\theta, \phi) = \sqrt{35/64\pi} \sin^3 \theta e^{-3i\phi}$                |
|     | -2  | $Y_3^{-2}(\theta, \phi) = \sqrt{105/32\pi} \cos \theta \sin^2 \theta e^{-i\phi}$    |
|     | 0   | $Y_3^0(\theta, \phi) = \sqrt{7/16\pi} (5\cos^3 \theta - 3\cos \theta)$              |
|     | 1   | $Y_3^1(\theta, \phi) = -\sqrt{21/64\pi} (5\cos^2 \theta - 1) \sin \theta e^{i\phi}$ |
|     | 2   | $Y_3^2(\theta, \phi) = \sqrt{105/32\pi} \cos \theta \sin^2 \theta e^{2i\phi}$       |
|     | 3   | $Y_3^3(\theta, \phi) = -\sqrt{35/64\pi} \sin^3 \theta e^{3i\phi}$                   |

**Tabla 1.** Armónicos esféricos  $Y_n^m(\theta, \phi)$  para  $n = 0, \dots, 4$  [EP].

En la siguiente figura se muestran geoméricamente las funciones de la Tabla 1 desde la perspectiva de los ejes  $(x, y, z)$  mediante un ángulo de visión de  $(\theta, \phi) = (60^\circ, -127, 5^\circ)$ . Para su generación, se ha utilizado el código de Matlab facilitado por Rafaely en su libro [1] .

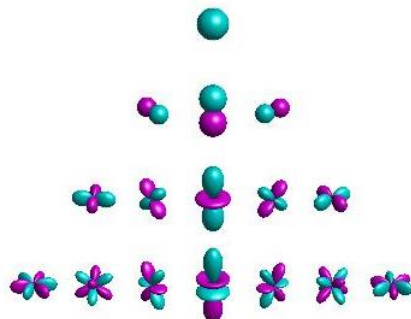


**Figura 13.** Gráficas de balón de los armónicos esféricos desde  $n = 0$  hasta  $n = 3$ .

|                                                                                                                     |
|---------------------------------------------------------------------------------------------------------------------|
| $Y_0^0$                                                                                                             |
| $Im\{Y_1^{-1}\} Y_1^0 Re\{Y_1^1\}$                                                                                  |
| $Im\{Y_2^{-2}\} Im\{Y_2^{-1}\} Y_2^0 Re\{Y_2^1\} Re\{Y_2^2\}$                                                       |
| $Im\{Y_3^{-3}\} Im\{Y_3^{-2}\} Im\{Y_3^{-1}\} Y_3^0 Re\{Y_3^1\} Re\{Y_3^2\} Re\{Y_3^3\}$                            |
| $Im\{Y_4^{-4}\} Im\{Y_4^{-3}\} Im\{Y_4^{-2}\} Im\{Y_4^{-1}\} Y_4^0 Re\{Y_4^1\} Re\{Y_4^2\} Re\{Y_4^3\} Re\{Y_4^4\}$ |

**Tabla 2.** Ilustración de las funciones representadas en la Figura 13

En el plano vertical se representan los gráficos en función del orden  $n$ , siendo  $n = 0$  la superior y  $n = 3$  la inferior. En cuanto al plano horizontal, a la izquierda se sitúa la parte imaginaria,  $Im\{Y_n^m(\theta, \phi)\}$ , mientras que a la derecha se encuentra la parte real,  $Re\{Y_n^m(\theta, \phi)\}$ , con valores  $m = -n$  y  $m = n$  respectivamente. La distribución se muestra en la para una comprensión más clara e ilustrada. Observamos que el armónico esférico de orden 0 y grado 0  $Y_0^0$ , situado en la parte superior, es constante sobre la esfera. Así, asemeja a un monopolio, mientras que la segunda fila es similar a un dipolo. En las siguientes gráficas, donde hay una mayor cantidad de valores  $m$  y  $n$ , las formas geométricas son más complejas [19].

Con la definición de armónicos esféricos en la Ecuación 8. Definición de armónicos esféricos., aparecía el término “funciones de Legendre”. Estas funciones, si bien no muy utilizadas en procesamiento de señales, son imprescindibles a la hora de ofrecer la propiedad de la ortogonalidad en  $Y_n^m(\theta, \phi)$ . Además, son responsables de su comportamiento en función del ángulo acimut y de su naturaleza compleja [21].

La función de Legendre se deriva de los polinomios de Legendre:

$$P_n^m(x) = (-1)^m (1-x^2)^{\frac{m}{2}} \frac{d^m}{dx^m} P_n(x), \quad x \in [-1, 1]$$

**Ecuación 9.** Definición de polinomios de Legendre.

Para valores negativos de  $m$ , las funciones de Legendre asociadas son proporcionales a aquellas con valores positivos, y se definen como

$$P_n^m(x) = (-1)^m \frac{(n-m)!}{(n+m)!} P_n^m(x), \quad x \in [-1, 1]$$

**Ecuación 10.** Definición de polinomios de Legendre para valores negativos de  $m$ .

Esta definición puede aproximarse mediante la siguiente fórmula:

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} (x^2 - 1)^n$$

**Ecuación 11.** Aproximación de la definición de los polinomios de Legendre

El desarrollo matemático de las funciones de Legendre no se ha considerado relevante para la elaboración de nuestro sistema baural, por lo que no se entrará en detalle en esta sección.

### 2.3.3. TRANSFORMADA ESFÉRICA DE FOURIER

Los armónicos esféricos forman una base en el espacio de Hilbert,  $L_2(S^2)$ , que está compuesto por el conjunto de funciones cuadrado-integrables sobre la esfera unidad. Las funciones cuadrado-integrables son aquellas en las que el resultado de  $\int_{-\infty}^{\infty} |f(x)|^2 dx$  es finito y la norma  $L_2$  implica que los armónicos esféricos pueden componer cualquiera de estas funciones con un error decreciente [22].

Una función definida sobre una esfera y perteneciente al espacio de Hilbert,  $f(\theta, \phi) \in L_2(S^2)$ , se puede representar mediante la suma ponderada, o pesos, de armónicos esféricos. Los pesos,  $f_{nm}$ , forman la transformada esférica de Fourier y se obtienen mediante la Ecuación 13.

$$f(\theta, \phi) = \sum_{n=0}^{\infty} \sum_{m=-n}^n f_{nm} Y_n^m(\theta, \phi)$$

**Ecuación 12.** Definición de función en el espacio de Hilbert.

$$f_{nm} = \int_0^{2\pi} \int_0^{\pi} f(\theta, \phi) [Y_n^m(\theta, \phi)]^* \sin \theta d\theta d\phi$$

**Ecuación 13.** Coeficientes, o pesos, de Fourier.

Estas definiciones serán de gran utilidad para la herramienta matemática presentada en la siguiente sección: la rotación de funciones.

### 2.3.4. ROTACIÓN DE FUNCIONES

Al igual que ocurre con las funciones definidas sobre una línea o círculo de radio unidad, las especificadas sobre la esfera pueden modificar su orientación. En el caso de las funciones lineales y circulares, se consigue mediante un parámetro denominado desplazamiento (por ejemplo,  $x_0$  o  $\phi_0$ ). Con respecto a las esféricas, el desplazamiento no se consigue en el mismo dominio que el parámetro, el par  $(\theta, \phi)$ . Para llevar a cabo una operación de rotación o modificación espacial, se debe recurrir a una operación tridimensional pudiendo así cambiar la orientación en función de cada uno de los ejes [23].

El conjunto de parámetros definido para realizar una operación de rotación espacial de funciones definidas sobre una esfera son los ángulos  $(\alpha, \beta, \gamma)$ , y se formulan utilizando los ángulos de Euler<sup>12</sup>. El matemático postuló que cualquier rotación puede ser descrita mediante tres rotaciones en sentido antihorario sucesivas sobre ejes linealmente independientes y, por supuesto, mediante tres ángulos de rotación. El proceso es el siguiente: en primer lugar, se ejecuta una rotación sobre el eje  $z$  mediante el ángulo  $\gamma$  (precesión); se continúa con una rotación sobre el nuevo eje  $x$  por medio de  $\beta$  (nutación) y se finaliza con una rotación sobre el nuevo eje  $z$  gracias a  $\alpha$ .

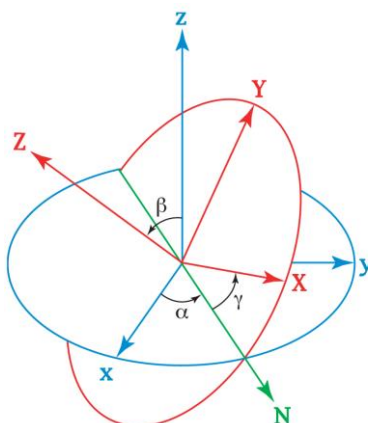


**Figura 14.** Ángulos de Euler.

En primer lugar, se define una posición sobre la esfera de radio unidad en coordenadas Cartesianas mediante la Ecuación 14 y se rota dicha posición, obteniendo  $x'$  utilizando las matrices de ángulos de Euler en la Ecuación 15.

$$x = [x, y, z]^T = [\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta]^T$$

**Ecuación 14.** Definición de punto sobre la esfera en notación algebraica.

$$x' = R_z(\alpha) R_y(\beta) R_z(\gamma) x$$

**Ecuación 15.** Definición del punto  $x'$  mediante rotación del punto  $x$ .

<sup>12</sup> A los ángulos  $(\alpha, \beta, \gamma)$  también se les denomina  $(\psi, \theta, \phi)$ .

Las matrices de rotación de Euler,  $R_z(\alpha) R_y(\beta) R_z(\gamma)$ , están definidas en el grupo Especial Ortogonal de  $3 \times 3^{13}$  matrices ortogonales, de manera que se satisface que su determinante es la unidad. Esta propiedad permite la definición de la rotación inversa mediante

$$[R_z(\alpha)R_y(\beta)R_z(\gamma)]^{-1} = R_z^T(\gamma)R_y^T(\beta)R_z^T(\alpha) = R_z(-\gamma)R_y(-\beta)R_z(-\alpha)$$

**Ecuación 16.** Definición de rotación inversa

donde,

$$R_z(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \quad R_z(\gamma) = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

**Ecuación 17.** Matrices de rotación de Euler.

Estas matrices operan en vectores de posición en coordenadas Cartesianas, por lo que se puede definir la operación rotación en términos de la Ecuación 7, donde  $f(x)$ ,  $x \in S^2$ . La siguiente ecuación, donde  $\Lambda$  es el operador de rotación, denota en su lado izquierdo la rotación de los valores de la función mientras que el sistema de coordenadas permanece fijo; lo cual es equivalente a mantener los valores de la función fijos mientras que se rota el sistema de coordenadas, que en este caso se representa en el lado derecho de la ecuación.

$$\Lambda(\alpha, \beta, \gamma) f(x) = f([R_z(\alpha)R_y(\beta)R_z(\gamma)]^{-1} x)$$

**Ecuación 18.** Definición de operador rotación  $\Lambda$ .

Todas las definiciones introducidas hasta el momento hacen referencia a la operación de rotación de valores y/o sistemas de coordenadas de forma generalizada. Nuestro desarrollo se basa en los armónicos esféricos, así que detallaremos su procedimiento de rotación para obtener los altavoces virtuales en las posiciones deseadas. Esta herramienta será muy potente durante la implementación tanto del sistema piloto mediante *Python*, así como del resultado final: el Plug – In *VST3*.

A continuación, se presentan un conjunto de definiciones que forman parte de la rotación de funciones de armónicos esféricos,  $Y_n^m(\theta, \phi)$ . El resultado será una función sobre la esfera que se puede representar como la suma de armónicos esféricos del mismo orden y grado. En la siguiente definición se relacionan los conceptos: rotación, armónicos esféricos y función Wigner-D.

$$\Lambda(\alpha, \beta, \gamma) Y_n^m(\theta, \phi) = \sum_{m'=-n}^n D_{m'm}^n(\alpha, \beta, \gamma) Y_n^{m'}(\theta, \phi)$$

**Ecuación 19.** Definición de rotación de armónicos esféricos.

<sup>13</sup> Este grupo se denomina  $SO(3)$ , *Spherical Orthogonal*. Es el grupo de rotación de elementos en el espacio  $\mathbb{R}^3$ .

La función de Wigner-D,  $D_{m'm}^n(\alpha, \beta, \gamma)$ , se define como “la matriz de elementos del operador de rotación” [24] y forma la base de la transformada de Fourier rotacional, la cual se aplica en funciones definidas sobre el grupo de rotación  $SO(3)$ . Utiliza, a su vez, la función real Wigner-d, que puede escribirse en términos del polinomio de Jacobi,  $P_s^{(\mu, \nu)}(x)$ .

$$D_{m'm}^n(\alpha, \beta, \gamma) = e^{-im'\alpha} d_{m'm}^n(\beta) e^{-im\gamma}$$

**Ecuación 20.** Función de Wigner-D.

$$d_{m'm}^n(\beta) = \zeta_{m'm} \sqrt{\frac{s!(s+\mu+\nu)!}{(s+\mu)!(s+\nu)!}} \sin(\beta/2)^\mu \cos(\beta/2)^\nu P_s^{(\mu, \nu)}(\cos(\beta))$$

**Ecuación 21.** Función de Wigner-d.

Los términos involucrados en la ecuación son:

- $\zeta_{m'm} = \begin{cases} 1 & , m \geq m' \\ (-1)^{m-m'} & , m < m' \end{cases}$
- $\mu = |m' - m|$
- $\nu = |m' + m|$
- $s = n - (\mu + \nu)/2$

Para obtener la rotación de una función en el dominio de los armónicos esféricos,  $g(\theta, \phi)$ , se utiliza la suma ponderada de los coeficientes de Fourier de la función original,  $f(\theta, \phi)$ , y la Ecuación 18, resultando así:

$$\begin{aligned} g(\theta, \phi) &= \Lambda(\alpha, \beta, \gamma) f(\theta, \phi) \\ &= \sum_{n=0}^{\infty} \sum_{m=-n}^n f_{nm} \Lambda(\alpha, \beta, \gamma) Y_n^m(\theta, \phi) \\ &= \sum_{n=0}^{\infty} \sum_{m=-n}^n f_{nm} \sum_{m'=-n}^n D_{m'm}^n(\alpha, \beta, \gamma) Y_n^m(\theta, \phi) \\ &= \sum_{n=0}^{\infty} \sum_{m=-n}^n \left[ \sum_{m'=-n}^n f_{nm} D_{m'm}^n(\alpha, \beta, \gamma) \right] Y_n^{m'}(\theta, \phi) \end{aligned}$$

**Ecuación 22.** Definición de rotación en el dominio de armónicos esféricos.

Llegamos entonces a la simplificación máxima de la expresión de la ecuación anterior combinando la última igualdad con la definición descrita en la Ecuación 12Ecuación 13. Así,

los coeficientes de Fourier de la función rotada se pueden formular como la suma ponderada de los coeficientes de Fourier de la función original,  $f_{nm}$ .

$$g_{nm'} = \sum_{m=-n}^n f_{nm} D_{m'm}^n(\alpha, \beta, \gamma)$$

**Ecuación 23.** Rotación en el dominio de armónicos esféricos mediante coeficientes de Fourier.

Finalmente, Rafaely nos muestra de manera gráfica el resultado de la rotación  $\Lambda(\alpha, \beta, \gamma)$ .

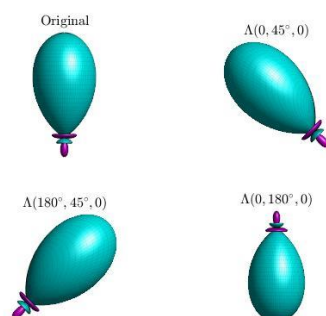


**Figura 15.** Rotación de un gráfico de globo [25].

## 2.4. CONFIGURACIONES DE SONIDO

### 2.4.1. SONIDO MONOFÓNICO

El audio monofónico, también llamado monoaural o 1.0, es aquel que se presenta al oyente desde una posición determinada, de forma que no existe direccionalidad ni espacialidad en la percepción sonora. Este sistema de grabación – reproducción, el primero de la Historia, nació a finales del siglo XIX con la invención del fonógrafo de Thomas Alva Edison y el gramófono de Emile Berliner, en el que sólo había un elemento de grabación (cilindro y disco, respectivamente) y de reproducción (bocina). El dispositivo de Edison permaneció en el mercado hasta 1929, año en el que Berliner vio su gramófono ocupar el primer puesto, sobre todo, por la durabilidad y resistencia del disco con respecto al cilindro. No fue hasta mediados del siglo XX que el sistema 1.0 se viera desplazado por el 2.0, o estéreo.

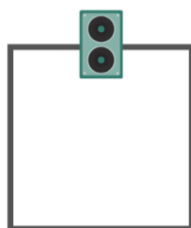


**Figura 16.** Diagrama de sonido monoaural [EP].

A pesar de que, teóricamente, solo sea necesario un micrófono para la grabación y un altavoz para la reproducción, se llevan a cabo técnicas de mezclado y masterización para presentar la misma señal en aquellos casos en los que haya más de un medio. En el caso de utilización de auriculares, se presenta la misma señal para cada uno de los canales. Esta señal está orientada de manera que se percibe como si se reprodujera desde el centro, como se observa en la Figura 16.

Debido a su simplicidad, la configuración 1.0 ha sido eclipsada por otras más complejas y potentes en cuanto a calidad, dimensionalidad y espacialidad. Sin embargo, sigue siendo el estándar en comunicaciones radio por ocupar poco ancho de banda.

#### 2.4.2. SONIDO ESTÉREO

El sonido estéreo, abreviatura de estereofónico, es aquel cuya grabación y reproducción se lleva a cabo mediante dos canales situados en distintas posiciones, de manera que el oyente lo percibe con cierta dimensionalidad y espacialidad. Generalmente, el sistema de reproducción se basa en dos altavoces que están situados formando alrededor de  $60^\circ$  con respecto al oyente, como se puede observar en la Figura 17 [26].

Este sistema nace en la década de 1930 en los laboratorios *Bell Labs* en Estados Unidos y en los *EMI* en Inglaterra, de la mano del ingeniero Alan Blumlein. Cada uno introdujo una técnica distinta para crear sonido 2.0 con el propósito de elaborar una “imagen” tridimensional.

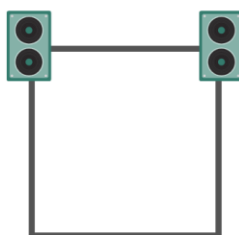


**Figura 17.** Diagrama de sonido estéreo [EP].

La idea llevada a cabo por los laboratorios estadounidenses se basa en la utilización de dos o más micrófonos situados en diferentes posiciones del espacio sonoro de la izquierda a la derecha. En cuanto a los europeos, se usa un micrófono estéreo o un *array*, en el que un par de transductores se encuentran muy cerca el uno del otro y que apuntan a distintas direcciones formando un ángulo determinado [26]. En esta disposición, que recibe el nombre de par coincidente, o técnica de Blumlein, cada micrófono tiene un diagrama polar en forma de cardioide y el ángulo suele ser de  $90^\circ$ , como se muestra en la siguiente figura.

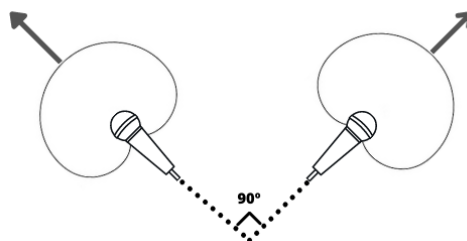


**Figura 18.** Configuración de micrófonos *par coincidente* [EP].

Gracias a las técnicas introducidas por ambos laboratorios se consigue un realismo sonoro bastante satisfactorio y una direccionalidad suficientemente clara. Sin embargo, existe una limitación en la amplitud sonora ya que, en ocasiones, desplazar las posiciones de los altavoces genera un vacío sonoro en ciertas ubicaciones y una pérdida en la sensación de profundidad.

#### 2.4.3. SONIDO CUADRAFÓNICO

La necesidad de añadir profundidad sonora a la experiencia auditiva fue el principal factor que incentivó la investigación de una configuración sonora más completa y compleja que el par estereofónico. Así, a principios de los 70, se llevó a cabo una investigación para alcanzar este propósito, dando lugar al sistema cuadrafónico, o 4.0. En él, se añaden un par estéreo en la parte trasera al oyente [26].

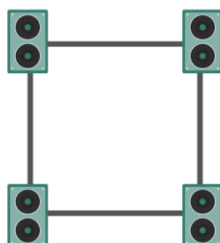


**Figura 19.** Diagrama del sonido cuadrafónico [EP].

Pese a ser una configuración aparentemente eficiente y válida para conseguir el propósito para el que fue inventada, este sistema tuvo muy poca acogida en el mercado por distintos factores. Ya que el espacio de grabación consistía en el emplazamiento de cuatro altavoces, se requería una reproducción mediante cuatro canales para conseguir percibir la mejora en la experiencia sonora. Desgraciadamente, en la década de 1970 no existían medios de distribución de cuatro canales excepto por la cinta, la cual no era muy utilizada. Por lo tanto, se desarrollaron otras técnicas para poder transmitir las cuatro señales en medios de dos canales, como el disco de vinilo, lo cual requería una precisión en el mezclado de los dos canales para la transmisión y la decodificación de los cuatro canales. Además, se debía asegurar que el medio de dos canales fuera reproducible en un equipo estéreo convencional. [26] Otra de las razones que propiciaron el fracaso comercial fue el alto precio de producción del sistema.

Sin embargo, no podríamos dar paso a las siguientes configuraciones sin el sonido cuadrafónico. Los Laboratorios Dolby lo modificaron para adaptarlo a las necesidades del mundo del cine y del teatro, dando lugar a los sistemas detallados en la sección 2.4.4.

#### 2.4.4. MULTICANAL

Como ya se ha introducido, el sistema 4.0 fue responsable del desarrollo de otros sistemas muy utilizados en la actualidad. La Unión Internacional de Telecomunicaciones, UIT o ITU en inglés, redactó una Recomendación denominada ITU-R BS.775 para “Sistemas de sonido estereofónico multicanal con y sin acompañamiento de imagen”<sup>14</sup> [27]. Las siglas BS se refieren a los servicios de transmisión de sonido, *Broadcasting Service (Sound)*. Esta recomendación forma parte de la ITU-R BS.2159-8 para la “Tecnología de sonido multicanal en aplicaciones de transmisión y de hogar”<sup>15</sup> [28]. Se considera necesario mencionar que la ITU estableció las bases de los sistemas multicanal con su investigación desde principios de los años 90, pero que fueron diversos los laboratorios que implementaron sus recomendaciones para dar lugar a sistemas adaptados a las necesidades del consumidor, como los Laboratorios Dolby. En esta sección nos centraremos en las Recomendaciones desarrolladas por la Unión.

#### SISTEMA 5.1

La UIT-R desarrolló la Recomendación BS.775 para el sistema de sonido multicanal estereofónico 3/2, comúnmente conocido como 5.1. Esta Recomendación surge como necesidad de especificar una jerarquía de sistemas multicanales compatibles y capaces de aumentar la estabilidad de la imagen de sonido frontal y la sensación ambiental, exigiendo una posición de altavoces a la misma altura que los oídos del oyente. Este sistema se caracteriza por su utilización en servicios de transmisión digital, cines y en aplicaciones personales como reproducción en salas de estar.

En la siguiente figura se muestra la estructura de referencia, con sus elementos explicados a continuación [27].

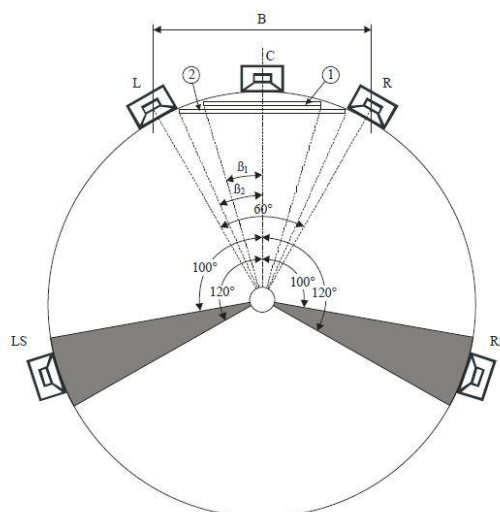


**Figura 20.** Estructura de referencia del sistema de sonido 5.1 recomendado por la ITU.

<sup>14</sup> “Multichannel stereophonic sound system with and without accompanying picture”.

<sup>15</sup> “Multichannel sound technology in home and broadcasting applications”.

| Altavoz                       | Ángulo horizontal (°) | Altura (m) | Inclinación (°)       |
|-------------------------------|-----------------------|------------|-----------------------|
| C – Center                    | 0                     | 1.2        | 0                     |
| L, R – Left, Right            | 30                    | 1.2        | 0                     |
| LS, RS – Left, Right Surround | 100 ... 120           | $\geq 1.2$ | 0 ... 15 hacia abajo. |

**Tabla 3.** Descripción de los altavoces en la configuración 5.1 de la ITU.

El elemento designado mediante ① se refiere a la pantalla de televisión de alta definición (*HDTV*), que debe cumplir que: su posición menos la distancia de referencia debe ser igual a tres veces la altura de la pantalla ( $H$ ) y debe cumplir  $2\beta_1 = 33^\circ$ . El elemento ② es otra pantalla (no necesariamente *HDTV*) cuya posición debe ser el doble de su altura, y debe cumplir  $2\beta_2 = 48^\circ$ . Por último, la abreviatura  $B$  se refiere a la distancia entre las posiciones de los altavoces base,  $L$  y  $R$ .

Observamos que en la Figura 20 vemos representados cinco altavoces y que, no obstante, el sistema se denomina 5.1. El 1 hace referencia a un altavoz de bajas frecuencias denominado *LFE*, *Low Frequency Effects* o *subwoofer*, y que aparecerá en cada uno de los sistemas multicanal que presenten “0.1” en su descripción. Este altavoz se utiliza para mejorar las bajas frecuencias, ya que incluir este rango correctamente en los demás es muy complicado y los resultados no son adecuados. Los *LFE* se caracterizan por tener una banda de frecuencia es muy estrecha, de tan solo 120 Hz, y por aplicar 10 dB de ganancia a la reproducción [27].

En la Recomendación BS.2159-8 [28] se especifican los requisitos básicos para crear sistemas multicanal que vayan más allá del 5.1 y se mencionan a continuación:

1. Requisitos sobre la “imagen” sonora: “se debe mantener la estabilidad direccional de la imagen de sonido frontal en todo el área de mayor resolución” y “la imagen sonora debe reproducirse en todas las direcciones alrededor del oyente, incluyendo la elevación”. El primero surge como necesidad de mantener la posición entre las imágenes sonoras y de vídeo, mientras que el segundo se indica tanto para mantener la direccionalidad desde las posiciones horizontales como para añadir la vertical.
2. Requisito sobre la sensación de impresión espacial: “intensificar significativamente la sensación espacial tridimensional para añadir realismo perceptual mediante el uso de altavoces detrás y/o a los lados y arriba y/o abajo”. Como ya se ha explicado en esta sección, el sistema 5.1 establece que todos los altavoces deben situarse a la altura de los oídos del oyente, por lo que la sensación espacial se limita únicamente al plano horizontal. Con este requisito se consigue incluir una sensación de elevación, así como de reverberación de la sala y de ambiente sonoro. Observamos que complementa o define lo establecido en el requisito 1.
3. Requisito básico del área de escucha: “se debe mantener la calidad de sonido alrededor de una superficie más amplia que la ofrecida por el sistema 5.1”. Se espera una mejora en zona sonora que la proporcionada por el sistema estereofónico 3/2.
4. Requisito básico de compatibilidad con sistemas de sonido existentes: “se debe asegurar un grado aceptable de compatibilidad con el sistema de sonido 5.1 especificado en la Recomendación ITU-R BS.775”. Se establece porque se debe asegurar una coexistencia con el mezclado hacia configuraciones por debajo, como

la estereofónica o la monofónica. Por lo tanto, cualquier sistema debe ser compatible tanto con el 5.1 como con los situados por debajo. De esta manera, se puede adaptar cualquier sistema de sonido a las necesidades de la aplicación y del ingeniero/a al cargo.

5. Requisito básico de transmisión en vivo: “debe ser posible la grabación en vivo, el mezclado y la transmisión”. Este es un requerimiento ya especificado en ITU-R BS.775 para 5.1 y es necesario su mantenimiento para sistemas más avanzados.

Dentro de 5.1, se han implementado mejoras que añaden dos y cuatro canales en altura, 5.1.2 y 5.1.4, respectivamente. Para el primero, los ángulos de posición de los altavoces deben estar entre  $60^\circ$  y  $100^\circ$  medidos desde el plano de visión del espectador. Para el segundo, el ángulo para los altavoces delanteros superiores debe estar entre  $30^\circ$  y  $55^\circ$  y para los altavoces traseros superiores, entre  $125^\circ$  y  $150^\circ$ . En las siguientes figuras se puede visualizar la disposición en el plano vertical para las dos configuraciones citadas.

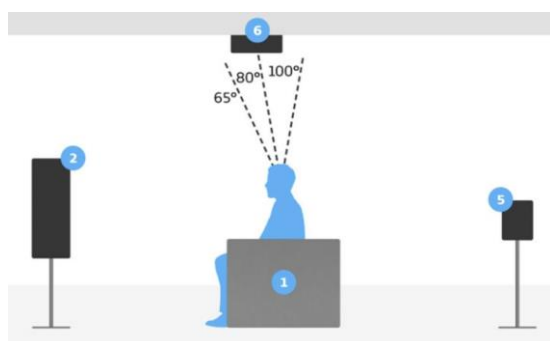


**Figura 21.** Vista lateral de 5.1.2 [29].

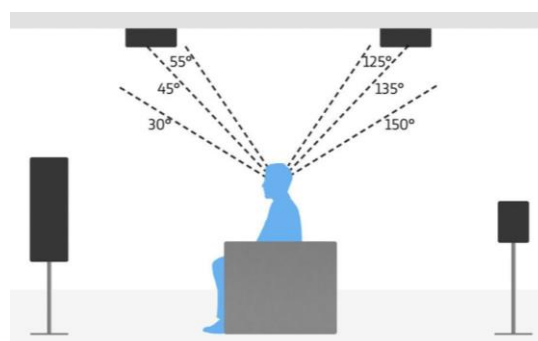


**Figura 22.** Vista lateral de 5.1.4 [30].

Por último, y aunque en la Recomendación BS.2159-8 se indican sistemas para aplicaciones de transmisión como el 10.2 y el 22.2, no se consideran relevantes ya que se encuentran bajo desarrollo y su implementación no se ha tenido en cuenta para la realización de este Trabajo.

## SISTEMA 6.1

La única novedad que introduce esta configuración es el altavoz situado en el centro en la posición trasera al oyente entre los canales *Ls* y *Rs*, en el plano horizontal y, al igual que en 5.1, a la altura de los oídos. Se incluye debido a que la localización desde la posición posterior es menos precisa por el sistema auditivo humano debido a la forma de la oreja. El 6.1 no se considera un sistema clave ni determinante a la hora de ofrecer una experiencia de usuario envolvente.

## SISTEMA 7.1

Esta configuración, al igual que la anterior, incorpora novedades a la 5.1 para añadir realismo y nitidez a la imagen sonora. A diferencia de 6.1, el sistema 7.1 prescinde del altavoz trasero central debido a su poca eficiencia perceptual. En este caso, se sitúan dos altavoces en las posiciones laterales al oyente aumentando así el realismo sonoro tanto para acompañamiento cinematográfico como para otras experiencias que solo implican sonido, como la reproducción musical. Así, se consigue obtener una mayor precisión en la ubicación

espacial sonora y no sólo para el oyente situado en la posición central, sino para cualquier espectador y posición [31].

La mayor característica del sistema 7.1 es la posibilidad de ofrecer una ubicación determinada de cada detalle sonoro, aunque no se consigue una percepción tridimensional debido a la ausencia de altavoces situados en altura. Por ello, la modificación del 7.1 mediante el 7.1.4 resulta muy beneficiosa para conseguir el Requisito 2 de la ITU-R.

#### SISTEMA 7.1.4

Esta es la configuración de disposición de altavoces elegida para construir el sonido envolvente que se transformará en un sistema binaural para su presentación mediante auriculares. Se ha decidido utilizar porque proporciona una imagen sonora muy completa, tanto en el plano vertical como horizontal, así como en la percepción de profundidad y direccionalidad.

Consiste en la misma disposición de altavoces que 7.1 pero, además, añade cuatro en el techo, cuya posición forma desde  $125^\circ$  hasta  $150^\circ$  ante el plano horizontal a la altura de los ojos del espectador. En la Figura 23. Emplazamiento de altavoces del sistema 7.1.4 desde el plano cenital. vemos la posición de los altavoces desde el plano cenital y en la Figura 24 se observan los grados que han de formar los situados en la parte superior trasera con respecto al oyente. La misma configuración se aplica para los altavoces situados en la parte superior delantera [32].

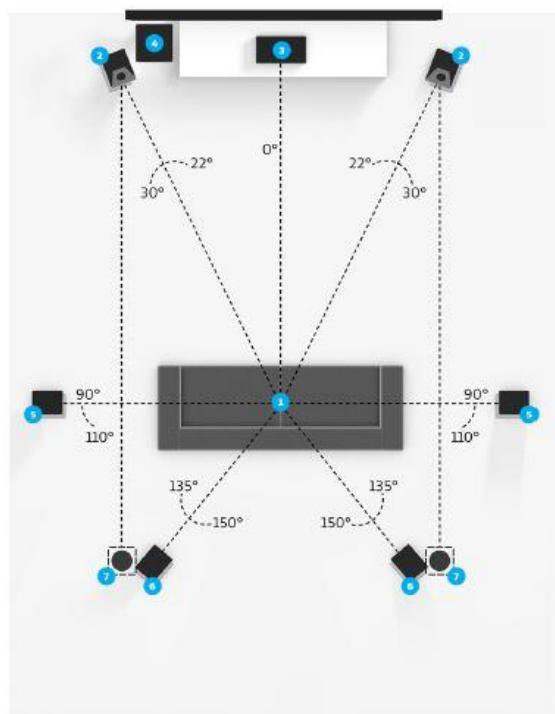


**Figura 23.** Emplazamiento de altavoces del sistema 7.1.4 desde el plano cenital.

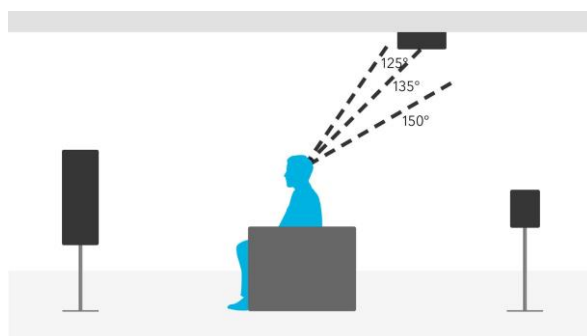


**Figura 24.** Vista lateral de 7.1.4.

Este sistema de sonido envolvente consta de 12 altavoces, 7 en el plano horizontal: central delantero, derecho e izquierdo delanteros, derecho e izquierdo *surround* y derecho e izquierdo traseros *surround*; 1 altavoz de bajas frecuencias (LFE o *subwoofer*): situado junto al central; y 4 altavoces en el plano vertical: derecho e izquierdo delantero superior y derecho e izquierdo trasero superior. Para nuestro desarrollo, no utilizaremos el *subwoofer*, ya que lo que se pretende es obtener la sensación de espacialidad y dimensionalidad una vez transformado el sistema en uno de dos canales para presentarlo mediante los audífonos, y este dispositivo no otorga dichas características, sino que realza las bajas frecuencias.

Aunque existen más configuraciones de altavoces, como 9.1, se considera suficiente el análisis que se ha hecho de los sistemas a lo largo de esta sección.

## 2.5. TECNOLOGÍA AMBISONICS

Los sistemas de sonido multicanal explicados en la sección anterior se caracterizan por su buen funcionamiento en cines y teatros, ofreciendo un nivel de efectos de sonido satisfactorio. Además, se creía que las configuraciones como la 5.1 iban a proporcionar un resultado similar en cuanto a experiencias musicales. Sin embargo, estos sistemas no se crearon para crear un sonido musical envolvente, por lo que se tuvo que desarrollar una tecnología distinta para conseguir este propósito en todos los entornos digitales.

Esta tecnología se denomina *Ambisonics*, y fue creada por los estudiantes de Matemáticas de la Universidad de Oxford Michael Gerzon, Peter Craven, Stephen Thornton y Philip Allen durante los años 70, creando la *Oxford University Tape Recording Society*, *OUTRS*. El equipo, junto al Profesor Peter Felget de la Universidad de Reading, trabajó para desarrollar un sistema de sonido envolvente para conseguir recrear, en la medida de lo posible, el ambiente acústico de la grabación original [10]. Gerzon consideró que el sistema cuadrafónico, si bien muy popular durante principios de los años 70, ofrecía una excesiva sensación de reverberación y una gran falta de información direccional desde los lados izquierdo y derecho del oyente, así como desde el plano vertical [33]. Así, el sistema se diseñó desde el inicio para grabar mediante un micrófono envolvente, el *Soundfield Microphone*, basado en la transducción completa del campo sonoro mediante cuatro cápsulas de media pulgada organizadas en un arreglo tetraédrico. En 1978 se comercializa el primer micrófono *Soundfield*, creado en colaboración con *Calrec*, el Mark IIIB, el cual creaba cuatro salidas que contribuían hacia una matriz que creaba, a su vez, cuatro salidas sintéticas (tres se correspondían a componentes de velocidad y una a la componente de presión) que se recombinaban para permitir que cualquier micrófono de característica de primer orden

apuntara en la dirección deseada. Estas cuatro salidas se denominaron formato B, *Ambisonics B-format*.



**Figura 25.** Primer micrófono Soundfield [34].

Las cuatro señales *B-Format*, embebidas en una sola señal, se organizan en información derecha-izquierda, delante-atrás y arriba-abajo, además de una señal de referencial monoaural. Además, se incluyó la posibilidad de reproducir información de altura que, como ya hemos comentado, ofrece más realismo a los sistemas envolventes. Para la reproducción, la señal se introduce en un decodificador que genera un mínimo de cuatro alimentaciones para altavoces que, en un sistema Ambisonics, contienen toda la información de la grabación, pero con distintas relaciones. Los altavoces están alineados para recrear la acústica y el ambiente de la pista original.

Una de las características principales de estos sistemas es la utilización de los métodos de localización de fuentes sonoras llevados a cabo por el cerebro y el sistema auditivo humano. Se lleva a cabo una reconstrucción de frente de onda superpuesta a los valores de nivel, fase y coherencia explicados en la sección 2.2.4. Esto se traduce en varios beneficios notables. En primer lugar, se pueden situar los altavoces aproximadamente en cualquier posición, ya que se puede indicar al decodificador dónde están. En segundo, el oyente puede situarse fuera de la configuración de altavoces y podrá seguir experimentando la “imagen” sonora gracias a que el efecto envolvente creado por Ambisonics es estable alrededor de un área de escucha muy amplio. Por último, se ofrece aquello que hasta la fecha era inimaginable: la capacidad de reproducir información de altura, a lo que se le denominó Perifonía, que en griego significa “sonido alrededor”. En el logo comercial de Ambisonics, en la Figura 26, se indica los componentes de la señal *B-Format*. El círculo exterior representa la alimentación omnidireccional, denominada *W*; la figura vertical representa el frente menos el posterior, *X*; la figura horizontal es el plano izquierdo menos el derecho, *Y*; y el círculo central es el plano superior menos el inferior visto desde arriba, *Z* [10].

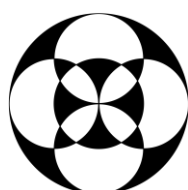


**Figura 26.** Logo de Ambisonics.

Las componentes  $W, X, Y$  y  $Z$  son denominadas de primer orden y, al tener baja resolución espacial, esto implica algunas limitaciones, especialmente el punto dulce o *sweet spot*. Este punto es aquel en el que se puede escuchar la grabación tal y como fue creada, con toda la información espacial. Más allá de este área la percepción especial se pierde y la imagen se percibe desde el altavoz más cercano [35]. Debido a la imposibilidad de ofrecer una reproducción del campo sonoro a gran escala, se creó una extensión del formato-B hacia resoluciones espaciales mayores, *Higher Order Ambisonics (HOA)*, las cuales han demostrado aumentar el punto dulce de escucha.

### 2.5.1. HIGHER ORDER AMBISONICS (HOA)

En esta subsección se explica cómo Ambisonics de Mayor Orden representa el campo sonoro. Las componentes espaciales obtenidas mediante la aplicación de la Transformada de Fourier esférica sobre cada onda son señales armónicas esféricas y se denominan armónicos en formato B,  $B_n^m(\theta, \phi)$ , y veremos que tendrán una clara relación con los armónicos esféricos  $Y_n^m(\theta, \phi)$ . La descomposición en armónicos esféricos de un campo acústico de presión  $p$  viene dada por la ecuación de onda siguiente en el sistema de coordenadas esférico, donde ya sabemos que cualquier punto situado en una esfera se describe mediante los ángulos elevación y acimut y el radio.

$$\Delta p + k^2 p = 0$$

**Ecuación 24.** Ecuación de onda para la descomposición de un campo de presión en armónicos esféricos.

Desarrollando esta ecuación, se obtiene la descomposición de Fourier – Bessel y, por lo tanto, los componentes Ambisonics se relacionan con los armónicos esféricos y con las funciones radiales  $j_m(kr)$  o funciones esféricas de Bessel. Las gráficas en la Figura 27 muestran cómo estas componentes contribuyen al campo sonoro en función de la distancia desde el centro de la esfera. Cuanto mayor sea el orden ( $m$ ), mayor es la contribución que el grupo de componentes de orden  $m^{th}$  sobre el campo sonoro en comparación con la longitud de onda. Este comportamiento influye en los componentes Ambisonics al relacionarlos con las derivadas del campo sonoro espacial, ya que las derivadas de mayor orden ayudan a aproximar el campo sonoro sobre una mayor vecindad con referencia en el centro de la esfera.

$$p(\vec{r}) = \sum_{m=0}^{\infty} j_m(kr) \sum_{0 \leq n \leq m} B_n^m(\theta, \phi) Y_n^m(\theta, \phi)$$

**Ecuación 25.** Descomposición del campo de presión acústica en la serie de Fourier – Bessel.

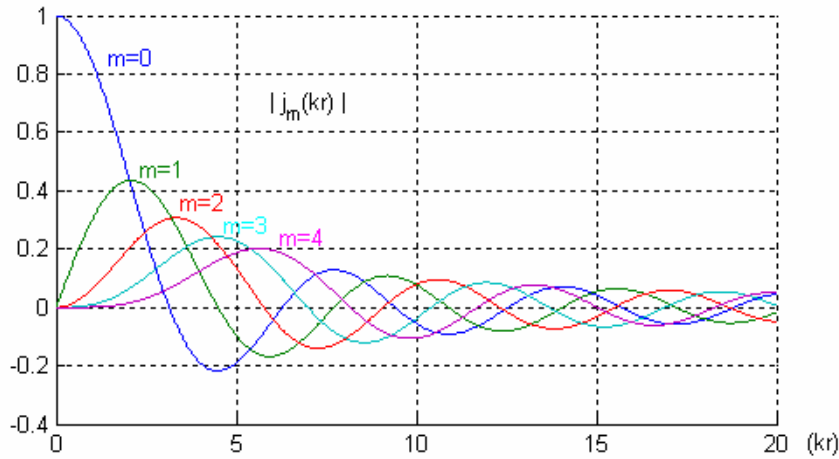


**Figura 27.** Funciones esféricas de Bessel  $j_m(kr)$  [36].

Los sistemas Ambisonics se basan en un orden  $M$  finito, de forma que  $m \leq M$ , así que se obtienen un número de componentes Ambisonics tridimensionales  $K_{3D} = (M + 1)^2$ . En el caso en el que solo se cuente con fuentes en el plano horizontal, el orden es igual al grado y el número de componentes es igual a  $K_{2D} = 2M + 1$ . Este trabajo se desarrolla en el espacio tridimensional, por lo que utilizaremos la componente  $K_{3D}$ . [36]

En Ambisonics B-format de primer orden,  $M = 1$ ,  $K_{3D} = 4$ , los componentes  $W, X, Y, Z$  se denominan  $B_0^0, B_1^1, B_1^{-1}, B_1^0$ , respectivamente. En el segundo orden,  $M = 2$  y  $K_{3D} = 9$ , y las siguientes componentes en Ambisonics se denominan  $B_2^{-2}, B_2^{-1}, B_2^0, B_2^1, B_2^2$ . En el caso del tercer orden, el número de componentes es igual a  $K_{3D} = 16$ , siendo las siguientes componentes  $B_3^{-3}, B_3^{-2}, B_3^{-1}, B_3^0, B_3^1, B_3^2, B_3^3$ . Existen hasta 6 órdenes de Ambisonics, con sus consiguientes  $B_m^n$ , pero no se consideran dentro del ámbito de estudio de este Trabajo.

### 2.5.2. RELACIONES ENTRE FORMATO – B DE AMBISONICS Y HOA

Como ya indicamos en la sección dedicada al estudio de armónicos esféricos en la sección 2.3.2, existe una relación entre sus definiciones y las del Formato B de Ambisonics. Estas serán de gran utilidad, ya que las utilizaremos para crear el generador de fuentes virtuales en las posiciones deseadas.

La normalización de armónicos esféricos en Formato B viene dada por la siguiente expresión:

$$B_n^m(\theta, \phi) = N_n^{|m|} P_n^{|m|}(\cos \theta) \begin{cases} \sin -m\phi, & m < 0 \\ \cos m\phi, & m \geq 0 \end{cases}$$

**Ecuación 26.** Armónicos esféricos en formato B.

En la ecuación, el término  $N_n^{|m|}$  se refiere a la normalización para el Formato B, el cual tiene una definición dependiendo de si se trata del tipo  $SN3D$  o  $N3D$ . Estos se refieren a esquemas de normalización de las amplitudes relativas usadas en cada canal de la señal de audio Ambisonics, de manera que se otorgue el mejor uso de estos para cualquier campo sonoro en coordenadas esféricas [37].

El convenio  $SN3D$  se refiere a la semi – normalización de Schmidt, y es utilizado por Jérôme Daniel debido a la propiedad de cálculo recursivo de los coeficientes de codificación, siendo los componentes de orden 1 los del vector de incidencia unitaria. Por otro lado,  $N3D$  se refiere a la base ortonormal para la descomposición tridimensional y asegura una potencia igual de las componentes codificadas en un campo 3D perfectamente difuso. A continuación, se muestran las definiciones para ambas normalizaciones [38].

$$N_n^m = \sqrt{|2 - \delta(m)| \frac{(n - |m|)!}{(n + |m|)!}}$$

**Ecuación 27.** Normalización SN3D.

$$N_n^m = \sqrt{|2 - \delta(m)|(2n + 1) \frac{(n - |m|)!}{(n + |m|)!}}$$

**Ecuación 28.** Normalización N3D.

Recordamos que en la Ecuación 10 se presentaba la definición de polinomios de Legendre para valores negativos de  $m$ , de forma que, relacionándolos con la definición de armónicos esféricos en la Ecuación 8, también para valores negativos de  $m$ , se obtiene la definición de armónicos esféricos conjugados:

$$Y_n^{-m} = (-1)^m (Y_n^m)^* \rightarrow (Y_n^m)^* = (-1)^m Y_n^{-m}$$

**Ecuación 29.** Definición de armónico esférico conjugado.

Así, para la normalización  $N3D$ , se concluye que las partes reales e imaginarias de los armónicos esféricos estándar conforman los armónicos en formato B, cuyas definiciones están condicionadas por el valor de  $m$ . En el caso de  $SN3D$ , mantienen una relación con los anteriores mediante el término  $\frac{1}{\sqrt{2n+1}}$ , de forma que:

$$B_n^{m(SN3D)} = \frac{1}{\sqrt{2n+1}} B_n^{m(N3D)}$$

**Ecuación 30.** Relación entre normalizaciones de armónicos esféricos.

Llegamos entonces a nuestro objetivo: relacionar el formato B de Ambisonics con las definiciones de armónicos esféricos. En la siguiente tabla se detallan los valores para un máximo de orden 3, que será el formato que se utilizará para crear el generador de señales Ambisonics, y que también servirán como parte del decodificador.

| $n$ | $m$ | $B_n^m$                                                                        |
|-----|-----|--------------------------------------------------------------------------------|
| 0   | 0   | $B_0^0 = 1$                                                                    |
| 1   | -1  | $B_1^{-1} = -\sin \theta \sin \phi$                                            |
|     | 0   | $B_1^0 = \cos \theta$                                                          |
|     | 1   | $B_1^1 = \sin \theta \cos \phi$                                                |
| 2   | -2  | $B_2^{-2} = \sqrt{3/4} \sin^2 \theta \sin 2\phi$                               |
|     | -1  | $B_2^{-1} = -\sqrt{3} \cos \theta \sin \theta \sin \phi$                       |
|     | 0   | $B_2^0 = 1/2 (3\cos^2 \theta - 1)$                                             |
|     | 1   | $B_2^1 = -\sqrt{3} \cos \theta \sin \theta \cos \phi$                          |
|     | 2   | $B_2^2 = \sqrt{3/4} \sin^2 \theta \cos 2\phi$                                  |
| 3   | -3  | $B_3^{-3} = -\sqrt{5/8} \sin^3 \theta \sin 3\phi$                              |
|     | -2  | $B_3^{-2} = \sqrt{15/4} \cos \theta \sin^2 \theta \sin 2\phi$                  |
|     | -1  | $B_3^{-1} = -\sqrt{3/8} (5\cos^3 \theta - 3\cos \theta) \sin \theta \sin \phi$ |
|     | 0   | $B_3^0 = 1/2 (5\cos^3 \theta - 3\cos \theta)$                                  |
|     | 1   | $B_3^1 = -\sqrt{3/8} (5\cos^3 \theta - 3\cos \theta) \sin \theta \cos \phi$    |
|     | 2   | $B_3^2 = \sqrt{15/4} \cos \theta \sin^2 \theta \cos 2\phi$                     |
|     | 3   | $B_3^3 = -\sqrt{5/8} \sin^3 \theta \cos 3\phi$                                 |

**Tabla 4.** Definición de Ambisonics formato B en relación con los armónicos esféricos [EP].

## 2.6. FUNCIONES DE TRANSFERENCIA RELACIONADAS CON LA CABEZA (HRTF)

En esta sección, se explica el significado y funcionamiento de un concepto muy importante en acústica y en tecnologías de grabación – reproducción envolvente, 3D o binaural: las Funciones de Transferencia Relacionadas con la Cabeza, más conocidas mediante sus siglas *HRTF* (*Head Related Transfer Functions*). Las *HRTF* son el resultado de aplicar la Transformada de Fourier a una respuesta al impulso que, en el caso de análisis sonoros y acústicos, es especial debido a que tiene en cuenta no solo el espacio donde se encuentra el oyente, sino que también considera la cabeza del oyente. Este concepto se denomina Respuesta al Impulso Relacionada con la Cabeza, *Head-Related Impulse Response (HRIR)*. En la sección 2.2.4 explicábamos las definiciones de *ITD*, *ILD* y *IC*, las cuales son de gran utilidad a la hora de aproximar matemáticamente el funcionamiento del sistema auditivo humano y del cerebro a la hora de percibir y ubicar sonidos. La gran característica de la *HRTF* es que permite sintetizar un sonido binaural, ya que se trata de una función que permite encontrar la presión sonora desde una fuente hasta los oídos del oyente.

Para llevar a cabo la medición de estas funciones se deben tener en cuenta tres elementos: el espacio donde se encuentra situado el oyente, su cabeza y la frecuencia del sonido. Habitualmente, la forma de obtener las medidas de la *HRTF* se basa en la utilización de un micrófono insertado en un busto con forma de cabeza humana, comúnmente

denominado *dummy head*. En un estudio llevado a cabo por el Instituto Tecnológico de Massachussets, se utiliza el dispositivo de la marca KEMAR [39], aunque con gran frecuencia se utiliza también el creado por el fabricante alemán Neumann.



**Figura 28.** *Dummy-head* del fabricante Neumann.

En la investigación se miden las respuestas al impulso en los oídos izquierdo y derecho emitidas desde un altavoz *Optimus Pro 7* (ya en desuso, véase que el estudio data de 1994), situado a una distancia de 1.4 metros desde el oyente y en un total de 710 posiciones distintas, desde un grado  $\theta$  de elevación de  $-40^\circ$  hasta  $+90^\circ$ . El clip de audio utilizado fue una secuencia pseudo aleatoria de bits muestreados a 44.1 kHz. El procedimiento tuvo lugar en la cámara anecoica del MIT.

Una cámara anecoica es una sala acústica conformada con materiales absorbentes en las paredes, suelo y techo; instalados con el propósito de absorber la mayor cantidad de energía sonora posible de forma que no se produzcan reflexiones de la señal original. Con esto se consiguen unas condiciones muy aproximadas a las de campo libre, en el que no existen obstáculos que originen réplicas de la señal. El coeficiente de absorción del material instalado en las paredes y el techo, denotado por la letra griega  $\alpha$ , debería tener un valor de 0.99. En cuanto al material utilizado para el suelo, suele consistir en una rejilla de acero inoxidable. Este tipo de salas cuentan con un valor de frecuencia de corte, que es aquél a partir del cual la cámara se comporta de manera anecoica, de forma que se considera que las condiciones simulan, como se buscaba, las del campo libre. Para frecuencias mayores que la de corte, las ondas sonoras son absorbidas en su totalidad. Otro concepto importante es la efectividad, medida en decibelios de rechazo (dB), definidos como la relación entre el sonido directo y el reflejado dentro de un recinto. El valor esperado debe ser mayor o igual a 80 dB en el aproximadamente todo el rango de frecuencias auditivo, desde 80 Hz hasta 20 kHz<sup>16</sup>. Las cámaras anecoicas se utilizan para realizar mediciones acústicas, modelos de cines y teatros, obtención de diagramas de directividad de altavoces o, como ocurre en nuestro caso, obtener respuestas al impulso de la cabeza del oyente [40].

<sup>16</sup> El rango total de frecuencias audibles comprende desde 20 Hz hasta 20 kHz.



**Figura 29.** El físico acústico Vern Oliver Knudsen en la cámara anecoica del MIT.

A pesar de que la Universidad Politécnica de Madrid cuenta con una cámara anecoica, debido a la pandemia de la Covid-19, no ha sido posible realizar las medidas para obtener las HRTF, por lo que las medidas se han llevado a cabo en una sala doméstica con los auriculares *Sennheiser Ambeo Smart*. Los resultados de las medidas obtenidas para la posición de cada altavoz se deben convolucionar con la señal original para conseguir experimentar la direccionalidad y espacialidad requeridas. Este procedimiento se explica detalladamente en la sección 4.1.

## 2.7. DESARROLLO DE PLUG - INS DE AUDIO

Un Plug – Ines un software que, incluido en un programa, añade características complementarias a este. En el caso de los Plug – Inde audio, se suelen agregar en Estaciones de Audio Digital, *Digital Audio Workstation (DAW)* para ampliar sus funcionalidades. Existen múltiples entornos de trabajo para que el programador lleve a cabo el desarrollo de su Plugin, entre los cuales se encuentran: IPlug, VST.NET, Delphi ASIO o JUCE. Todos utilizan el lenguaje de programación C++ debido a que, a pesar de ser de bajo nivel, su tiempo de compilación es muy corto. Esto resulta muy satisfactorio, ya que los Plug – Ins de audio suelen utilizarse en tiempo real.

Para el desarrollo de este trabajo se ha utilizado JUCE, un *framework* muy potente que incluye librerías y soporte entre plataformas.

### 2.7.1. TIPOS DE PLUG – INS DE AUDIO

A la hora de implementar o utilizar un Plug – Inde audio, debemos tener en cuenta tanto su formato como el soportado por la estación de trabajo con la que estemos trabajando. Son tres las principales extensiones para elaborar este software: VST/VST3, AU y AAX [41].

El formato *VST*, cuyas siglas significan *Virtual Studio Technology*, fue el primero que se creó para su uso libre. Fue en 1996 cuando se introdujo por primera vez esta extensión, gracias a Steinberg, creador de la Estación de Audio Digital *Cubase*. Al ser *opensource*, es soportado en casi todas las *DAW* existentes, excepto para las desarrolladas por Apple, que cuentan con su formato único. *VST3* indica que la última versión disponible es la 3.x.

Apple creó para esta herramienta un formato exclusivo para ser utilizado en sus equipos Mac, el *AU*, *Audio Unit*. Se asemejan a los *VST* en que son los más acogidos en las *DAW*, pero en este caso deben ser soportadas por el sistema operativo macOS. Existen adaptadores disponibles para ambos formatos y, habitualmente, se encuentran las dos versiones para el mismo Plug – In.

Para explicar el último formato, *AAX*, debemos tener en cuenta sus antecesores, los *RTAS* y *TDM*, *Real-Time AudioSuite* y *Time-Division Multiplexing*, respectivamente. Tanto los últimos como el primero fueron creados por el fabricante Avid, creador de *Protools*. La diferencia principal es el uso que el Plug – In hace del procesador. En el caso de *RTAS*, se utiliza el procesador del ordenador, mientras que *TDM* libera la *CPU* del ordenador utilizando el procesador del Procesamiento Digital de Señales, el módulo *DSP*. A pesar de ser muy potentes, tuvieron poca proyección debido a que su uso era únicamente posible en su “entorno madre”, *Protools*. No era de esperar que desaparecieran, de manera que en 2013 fueron sustituidos por un nuevo formato, el *AAX* (*Avid Audio eXtension*). Este formato funciona tanto en *CPUs* como en *DSPs* y se utiliza, principalmente, en proyectos de gran tamaño de 64 bits.

En conclusión, se ha decidido utilizar el formato *VST3* debido a su software y uso libres y porque el sistema operativo utilizado para desarrollar este Proyecto es Windows.

### 2.7.2. JUCE

JUCE, siglas de *Jules’ Utility Class Extensions*, es un entorno de trabajo que se utiliza para el desarrollo de aplicaciones móviles y de escritorio. Es un programa muy potente gracias a las librerías para generar *Plug – Ins* e interfaces gráficas de usuario, *GUI*.



**Figura 30.** Logo de JUCE.

JUCE ofrece una herramienta de gestión de proyectos, Projucer, que cuenta con el motor *ICE*, *Instant Compilation Environment*, de forma que se pueden visualizar los cambios en el código de manera instantánea. Este programa permite crear proyectos de distinta índole, todos ellos relacionados con el procesado de audio. Se detallan a continuación [42].

- *GUI*: crea el esqueleto de una aplicación JUCE con una ventana vacía, de manera que el usuario puede añadir nuevas funcionalidades o componentes gracias a las numerosas clases que se proporcionan.
- Aplicación animada: genera una interfaz gráfica animada, normalmente utilizada en aplicaciones móviles.

- OpenGL: permite el desarrollo de una aplicación con modelos 3D a partir de un esqueleto formado por una GUI y OpenGL para poner diseñar distintas características.
- Aplicación de consola: desarrollo de programas que no cuentan con una interfaz de usuario y que funcionan mediante líneas de comandos.
- Aplicación de audio: genera una aplicación JUCE vacía con una GUI y una configuración de entrada y salida de audio.
- Plug – Inde Audio: genera el esqueleto de un Plug – In y añade un código automáticamente que soporta los formatos más utilizados (VST, *AudioUnit* y AAX).
- Librerías estáticas y dinámicas: crea librerías de software reutilizables que se construyen en la capa más alta de JUCE.

Durante la creación del proyecto de Plug – Inde audio, se deben comprobar sus formatos y características dependiendo del tipo de software que se desee generar. Además, se pueden seleccionar los módulos necesarios para su correcto funcionamiento. En el caso que nos acontece, se incluirán aquellos que JUCE ofrece de manera libre para la gestión del audio, entre los que se encuentran: *juce\_audio\_basics*, *juce\_audio\_formats*, *juce\_audio\_core*, *juce\_audio\_plugin\_client*, .... Por otro lado, también se debe elegir el Entorno de Desarrollo Integrado, *IDE*, que será utilizado por el usuario para construir el Plug – In. JUCE es capaz de soportar múltiples entornos, dependiendo del sistema operativo: Xcode (macOS e iOS), Visual Studio (Windows), Android y Linux Makefile, entre otros. En nuestro caso, se ha utilizado Visual Studio, detallado en la siguiente sección, ya que el sistema operativo utilizado ha sido Windows.

Una vez creado el proyecto, se generan en JUCE cuatro documentos necesarios para el correcto funcionamiento y compilación del software. Dos de ellos tienen una extensión .h, que indica que son cabeceras (*headers*) que incluirán la definición de funciones, variables y elementos para un correcto funcionamiento de los otros dos documentos, cuya extensión es .cpp (archivos con formato C++) [43].

- PluginEditor.cpp: incluye la implementación de las funciones necesarias para generar la *GUI*, de manera que se manejen sus controles o visualizaciones.
- PluginEditor.h: contiene la declaración de las clases del PluginEditor.cpp. Se incluye la definición de ámbito público de las cabeceras de funciones, elementos que se utilizarán en la interfaz de usuario. En el ámbito privado, se encuentra la referencia al procesador de audio mediante `NewProjectAudioProcessor& audioProcessor;`.
- PluginProcessor.cpp: es el cuerpo del Plug – In. Está formado por funciones autogeneradas que permiten el correcto funcionamiento del programa. Las más importantes son el constructor, la función `prepareToPlay()`, `releaseResources()` y `processBlock()`.
  - Constructor: `AudioProcessor::AudioProcessor()`. Creará los *buses* de entrada y salida principales, los cuales estarán desactivados por defecto. Aquí se indican el número de canales de entrada y salida soportados por el Plug – In.
  - `void ProjectAudioProcessor::prepareToPlay(double sampleRate, int samplesPerBlock)`: esta función es llamada antes de que comience la reproducción, de forma que el procesador esté preparado para su correcto funcionamiento. En ella, se inicializan las variables utilizadas en el bloque de proceso. El parámetro `sampleRate` es la tasa de muestreo del *host*, en nuestro caso el *DAW*. El parámetro `samplesPerBlock` es el número de muestras de cada uno de los bloques. Este valor es clave para nuestro desarrollo, ya que veremos que en determinados procedimientos el tratamiento se realiza mediante bloques.

- void AudioProcessor::releaseResources(): se llama a esta función cada vez que la reproducción<sup>17</sup> se detenga. De esta forma, se liberan los recursos que no sean necesarios.
- void ProjectAudioProcessor::processBlock(juce::AudioBuffer<float>& buffer, juce::MidiBuffer& midiMessages): es la función donde se lleva a cabo el procesamiento de cada uno de los bloques. El parámetro buffer es de gran utilidad, debido a que contiene información relevante como el número de canales de entrada y salida. Es necesario mencionar que existe un único buffer tanto para el *bus* de entrada como para el de salida. Así, se deben definir variables intermedias que no modifiquen los valores de cada uno de los canales. Esto se detallará en la sección de Desarrollo. El parámetro midiMessages almacena los mensajes *MIDI (Musical Instrument Digital Interface)* en el caso de que se reciba una entrada de este dispositivo (ya sea real o virtual). En nuestro caso, no se utilizará, por lo que dicho parámetro no se utilizará. Todo el procedimiento y desarrollo matemático llevado a cabo en esta función se explica más adelante, ya que no se considera parte del Estado del Arte.
- PluginProcessor.h: contiene la declaración de las funciones y elementos utilizados en el archivo anterior. Aquí, al igual que en el editor, se cuenta con declaraciones públicas y privadas. Las públicas serán de gran utilidad, ya que mediante su creación en la cabecera se ahorra tiempo de computación y de procesamiento.

Vemos que las clases están relacionadas entre sí, por lo que es necesario establecer un vínculo entre ellas. El lenguaje C++ ofrece una directiva, `#include`, que se declara al inicio de las clases .cpp y que son leídas por el preprocesador, ordenándole que incluya el contenido de las cabeceras. En nuestro caso, tanto el archivo PluginEditor.cpp como PluginProcessor.h incluye las cabeceras del Editor y del Procesador. Además, las cabeceras pueden incluir a su vez otras librerías, como en el caso de PluginEditor.h, que incorpora preparada `<JuceHeader.h>` y el archivo PluginProcessor.h.

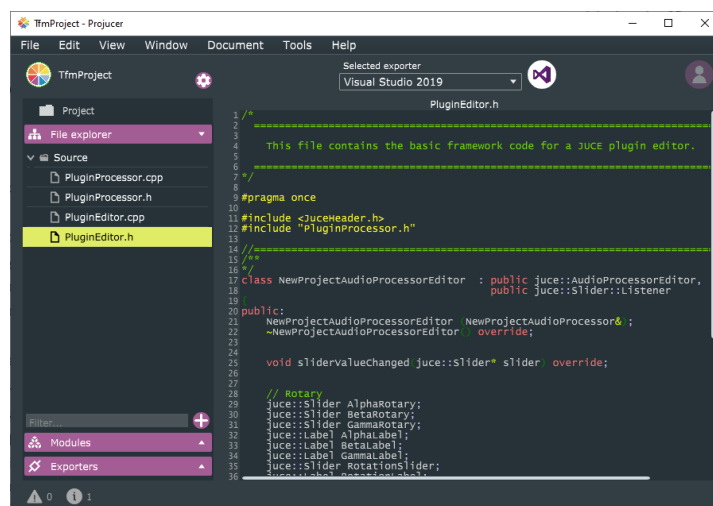


Figura 31. Captura del entorno Projucer.

<sup>17</sup> A la reproducción se le llamará también *playback*.

Observamos en la parte superior de la pantalla el símbolo de Visual Studio, . Cuando se pulsa sobre este icono, Projucer guarda el proyecto y lo ejecuta en el *IDE* especificado. Cualquier cambio guardado en el código modificado en Visual Studio se verá automáticamente reflejado en JUCE.

### 2.7.3. MICROSOFT VISUAL STUDIO

El entorno de desarrollo integrado creado por Microsoft, *Visual Studio*, es muy utilizado por desarrolladores para la implementación de programas, aplicaciones web, .... Permite realizar no solo el desarrollo del software, sino su análisis, depuración y ejecución de pruebas. Además, es soportado tanto en Windows como en macOS y es compatible con múltiples lenguajes de programación: C++, Java, Python y C#, entre otros. El análisis y descripción de este *IDE* no se considera necesario en esta Memoria, sin embargo, sí que se deben explicar y tener en cuenta dos acciones que ofrece este entorno: la compilación y la depuración del proyecto.

Por un lado, cuando se compila el proyecto entero, esto es, los cuatro archivos explicados en la sección anterior, se genera una carpeta denominada *Build*. En esta “construcción” se incluyen los ficheros necesarios para un correcto funcionamiento del programa, en este caso Plug – In. Uno de los elementos creados tiene la extensión .vst3, y será el utilizado por la estación de audio digital.

Por otro lado, Microsoft *Visual Studio* ofrece una herramienta muy potente para el análisis del código y variables: la depuración. En la mayoría de los entornos de programación se permite hacer un análisis exhaustivo del programa mediante el uso de *breakpoints* y comandos intermedios. El problema a la hora de depurar el código de un Plug – In es, principalmente, que su uso implica un segundo participante: la *DAW*. Por ello, no podrían visualizarse los valores de las variables y los estados del programa si no fuera por la opción que ofrece VS: la asociación a procesos externos. Así, se puede ejecutar el programa de manera externa y comprobar su funcionamiento en la ventana de variables locales. En la sección 4.3 se mencionan algunos de los valores obtenidos mediante esta herramienta, explicando por qué han resultado tan beneficiosos para la creación del Plug – In.

En definitiva, *Visual Studio* se considera un entorno muy competente para la implementación de nuestro software de audio.

## 2.8. ESTACIONES DE TRABAJO DE AUDIO DIGITAL

Una estación de trabajo de audio digital, *Digital Audio Workstation (DAW)*, es una interfaz digital de usuario que permite la grabación y/o edición de audio mediante el uso conjunto de *software* y *hardware*. Las primeras *DAW* nacieron en los años 70, y estaban orientadas a un uso mediante *hardware*, ya que los ordenadores carecían de potencia de procesamiento entonces. Sin embargo, ahora estos entornos están más enfocados a su uso *software*. Existen múltiples en el mercado con distintas características: comerciales, de código abierto, de uso libre, de pago.... Entre las más utilizadas se encuentran: Reaper, Ableton Live, Adobe Audition, Cubase, GarageBand o Pro Tools. Se detallan las características principales de los tres mencionados al inicio, ya que los últimos no han sido de gran uso debido a sus limitaciones en el soporte de canales de entrada y salida.

En este trabajo se ha considerado el uso de diversas estaciones, las cuales se explican en las siguientes secciones y se termina con la elección definitiva para el desarrollo.

### 2.8.1. REAPER

En 2004, el Ingeniero de Software Justin Frankel creó una compañía denominada *Cockos* con el propósito de desarrollar un programa para crear música y manipular audio. Es así cómo, tras numerosos inventos que lo situaron en la lista del *MIT* titulada “Inventores menores de 35”, creó la estación de audio digital *REAPER*, acrónimo que viene de *Rapid Environment for Audio Production, Engineering and Recording*). El programa sigue siendo hoy de gran acogida, siendo su versión actual la número 6.26, lanzada en 2021.



Figura 32. Logo de Reaper.

Las principales características de Reaper son [44]:

- El soporte en los sistemas operativos Windows, macOS y Linux.
- La lectura de múltiples formatos como AVI, MOV, MP3, WAV y MPEG, entre otros.
- La escritura en distintas extensiones como AVI, MOV, OGG VORBIS, MP3 y WAV, entre otras.
- La instalación e incorporación de Plug – Ins de 64 *bits*, de formatos como VST, VST3 AU.
- La grabación en distintas pistas y con múltiples configuraciones de canales desde cualquier hardware, también *MIDI*.
- La producción y el mezclado de audio y vídeo mediante una interfaz sencilla y cómoda, pero no por ello básica.

Además, incluye una versión de prueba gratuita por una duración de 60 días, razón que, unida a las propiedades definidas, ha motivado la elección para su uso en este Trabajo. A pesar de ser un entorno muy potente, se tuvo que dejar de utilizar debido a un mal funcionamiento del Plug – In durante su testeo, lo cual se explica en la sección Implementación en Reaper.

### 2.8.2. ABLETON

Ableton nació en 1999 con el propósito de crear un producto que fuera usado por músicos, ingenieros de sonido y artistas. La última versión lanzada se denomina *Live 11* y cuenta con potentes características para sus tres planes: *Intro Limited*, *Standard Full*, *Suite Full*. Se mencionan los principales elementos para las tres modalidades, excepto para el valor del número de canales de entrada y salida, que varía en función del plan.



**Figura 33.** Logo de Ableton.

- Soporta los sistemas operativos Windows y macOS, pero no Linux.
- Lee y escribe los formatos de audio WAV, AIFF, MP3, OGG VORBIS y FLAC.
- Soporta las extensiones de Plug – In VST, VST3 y AU.
- El manejo y tratamiento de múltiples pistas de audio.
- Entradas y salidas de audio soportadas:
  - 256 canales para los planes *Full*.
  - 8 canales para el plan *Limited*.

A pesar de ser un potente software, vemos que en la versión de prueba (*Intro Limited*), la estación soporta hasta 8 canales, cuando en nuestro Proyecto trabajamos con una señal de entrada Ambisonics de formato-B de tercer orden, es decir, de 16 canales. La utilización de las DAW ha sido requerida desde el principio de la implementación del Plug – In, por lo que no se consideró adecuada para la comprobación del funcionamiento del Plug – In.

#### 2.8.1. ADOBE AUDITION

Este software, previamente denominado *Cool Edit Pro*, nace en 2003 y se considera una de las mejores estaciones de trabajo del mercado. De la mano del gigante *Adobe*, *Audition* cuenta con un gran número de funcionalidades, efectos y opciones de grabación, tratamiento y procesado de audio. En esta DAW se encuentran las mismas características que en las mencionadas en *Ableton*, excepto que en este caso el número de canales soportado es mayor que 8, por lo que funciona correctamente con la señal de entrada.



**Figura 34.** Logo de Adobe Audition (versión 14).

Son casi infinitas las posibilidades que ofrece esta estación de audio y, aunque sea muy similar a las ya explicadas, se decidió usar debido a que el Plug – In desarrollado funcionaba correctamente sin ningún inconveniente. Es por ello por lo que, tras utilizar Reaper, se realizó una suscripción al programa tras su versión gratuita, la cual sólo dura 7 días.

En cuanto al uso de Plug – Ins en Audition, es posible importar aquellos con formato VST o VST3, siempre y cuando el archivo se encuentre en el directorio C:\Program Files\Common Files\VST3.

### 3. HARDWARE

En este apartado se presentan los dos elementos hardware utilizados para la realización de este Trabajo de Fin de Máster.

#### 3.1. MICRÓFONO ZYLIA

En la sección **¡Error! No se encuentra el origen de la referencia.** se presentaba la base matemática para la definición de *arrays* esféricos, y se establecía que, para realizar una grabación en 3D, es aconsejable el uso de una matriz de micrófonos dispuestos en una esfera. Como sabemos, el objetivo de este trabajo es la transformación de un archivo de audio con una configuración multicanal a una presentación binaural; por lo que necesitamos en primer lugar una grabación con dichas características. Para ello, se ha utilizado el dispositivo *Zylia Pro*, disponible en la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad Politécnica de Madrid.



**Figura 35.** Micrófono de array esférico Zylia Pro.

La finalidad de este micrófono es la grabación 360° tanto en estudio como en otros ambientes, en un formato – B de Ambisonics de tercer orden. Con la tecnología de altavoces virtuales, se pueden seleccionar secciones de sonido en el post – procesado para la combinación con vídeo o para obtener mezclas de audio sorprendentes. Además del hardware, *Zylia* incluye dos softwares y Plug – Ins, cada uno de ellos de estudio y de conversión de Ambisonics.

Entre las especificaciones técnicas principales se encuentran [45]:

- Un total de 19 micrófonos conformados por el mismo número de sistemas microelectromecánicos digitales (*Microelectromechanical Systems, MEMS*). Se trata de dispositivos de tamaño entre  $1\mu m$  y  $1mm$  formados por componentes activos y pasivos para la percepción, procesamiento de datos, comunicaciones y actuación sobre el entorno. En este caso, se trata de sensores que miden cambios de presión sonora en el ambiente [46].
- Sensibilidad al rango de frecuencias audibles: 20 Hz – 20 kHz.
- Rango dinámico de 105 dB.
- Relación señal a ruido:  $SNR (dB) = 69$ .
- Distorsión armónica total:  $THD = < 1\%$  para 128 dB SPL.
- Diagrama de directividad omnidireccional.

- Conversión Analógico/Digital con una resolución de bit de 24 bits y una tasa de muestreo de 48 kHz.

El archivo de audio generado mediante este micrófono se ha grabado en la Escuela y consiste en un cuarteto de cuerda dispuesto alrededor de Zylia.

### 3.2. SENNHEISER AMBEO SMART HEADSET

Para la presentación binaural del sistema envolvente Ambisonics es necesaria la producción de las respuestas al impulso de cada altavoz en sus distintas posiciones, midiendo así la Función de Transferencia Relacionada con la Cabeza (*HRTF*), explicada en la sección 2.6. Como se detalla en dicho apartado, la manera más óptima de obtener las funciones se lleva a cabo en una cámara anecoica. Sin embargo, al no tener posibilidad de utilizarla, se ha recurrido a una alternativa menos profesional pero no por ello insatisfactoria: la grabación en tres dimensiones mediante el hardware *Sennheiser Ambeo Smart Headset* [47].



**Figura 36.** Auriculares Sennheiser Ambeo Smart Headset.

Este dispositivo es el primero del mercado capaz de grabar sonido 3D. Los audífonos cuentan con unos micrófonos omnidireccionales que capturan todo aquello alrededor del oyente, de igual manera que lo hacen los oídos. Además, cuenta con un preamplificador que permite la cancelación de ruido, la grabación de sonido de manera transparente y un reductor de intensidad para entornos con altos niveles de sonoridad. Por otro lado, contiene un micrófono “de usuario” para su uso en conversaciones o grabaciones de audio. En nuestro caso, utilizaremos los micrófonos omnidireccionales de los audífonos. Es importante mencionar que sólo puede ser utilizado en dispositivos Apple, ya que el conector integrado es de tipo *Lightning*.

Las especificaciones técnicas de *Ambeo Smart HeadSet* son:

- Respuesta en frecuencia entre 15 Hz y 22 kHz.
- Nivel de presión sonora máxima: 112 dB.
- Preamplificador de la marca Apogee.
- Tecnología binaural.

El uso e integración de estos auriculares a lo largo del Proyecto se detallará en la sección Procedimiento de generación de la *HRTF*.

## 4. DESARROLLO

Llegamos a la parte de desarrollo y elaboración del Trabajo propiamente dicho. A lo largo de este apartado, se explicará cómo se han integrado los conceptos definidos hasta ahora para generar una cadena de grabación – reproducción desde multicanal hasta presentación binaural.

### 4.1. ARQUITECTURA DEL SISTEMA

Nuestro sistema, en su totalidad, cuenta con la grabación de un escenario sonoro mediante el micrófono Zylia, el cual genera 19 canales de audio que serán introducidos en el codificador de Ambisonics para su transformación a los 16 canales del tercer orden del formato – B. Una vez generados, se aplica una rotación de funciones de armónicos esféricos de forma que se presenten delante del oyente. Tras esta operación, los canales pasan por un decodificador que realizará otra conversión de canales, en este caso de los 16 de Ambisonics a los 12 del formato 7.1.4 recomendado por la ITU. Finalmente, se llega a la presentación binaural mediante auriculares a través del presentador o *renderer*, el cual generará los dos canales de salida después de una elaborada cadena de procesamiento de audio. En la Figura 37 vemos cada uno de los elementos que conforman el sistema y las relaciones entre ellos.

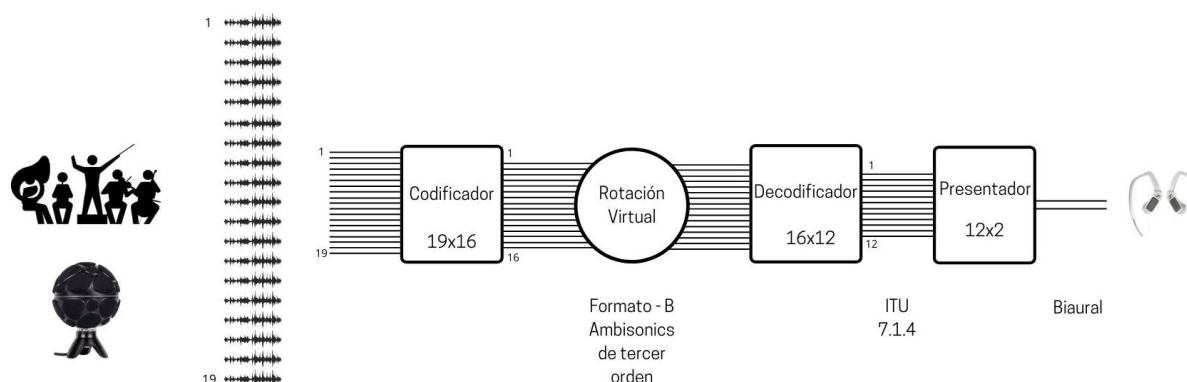


Figura 37. Esquema de la arquitectura del sistema [EP].

Para la elaboración, se ha generado en primer lugar un proyecto “piloto” mediante Python, el cual ha permitido comprobar el correcto funcionamiento de cada uno de los bloques. En segundo lugar, se ha decidido crear un Plug – In para ser utilizado en una Estación de Audio Digital y poder así percibir el resultado binaural en tiempo real. Previamente, se explica cómo se han obtenido las Funciones de Transferencia Relacionadas con la Cabeza, ya que serán cruciales para la fase final de ambos desarrollos.

### 4.1. PROCEDIMIENTO DE GENERACIÓN DE LA HRTF

En esta sección, se explica el procedimiento de generación de las *HRTF*, las cuales deberán ser procesadas para obtener todas las respuestas al impulso con un mismo origen

de tiempos y número de muestras. Se ha utilizado Matlab para esta tarea, ya que sus herramientas de visualización de gráficos y de procesamiento de señales son muy potentes, lo cual ayudará a la correcta elección de los valores de las señales grabadas mediante *Sennheiser Ambeo Smart Headset*.

Para la obtención de las respuestas al impulso,  $h(n)$ , se ha llevado a cabo un proceso de identificación mediante una secuencia de longitud máxima, a la cual llamaremos  $x(n)$ . En nuestro caso, la secuencia será una señal generada y lo que se obtendrá serán las *HRTF*. Para obtener cada una de las funciones de transferencia, utilizaremos la autocorrelación de la señal para cada una de las posiciones. Para obtenerla, llevamos a cabo la convolución de la señal original invertida (en la siguiente figura será  $x(-n)$ ) y la enviada por los altavoces en cada posición (que será la respuesta al impulso  $h(n)$ ).

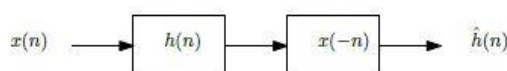


**Figura 38.** Esquema de identificación de  $h(n)$  mediante  $x(n)$ .

Comenzamos con la generación de dicha secuencia. Esta consiste en un barrido lineal de frecuencias de 5 segundos de duración, muestreada a 44.1 kHz y cuyo ancho de banda comprende entre 100 Hz y 15 kHz, ya que no se ha percibido respuesta sonora para valores inferiores al mínimo o superiores al máximo. Esta señal se consigue gracias a la función *chirp*. En las siguientes figuras observamos la forma de onda, el espectro y el espectrograma del barrido lineal. El espectrograma de una señal representa la distribución de la energía de la señal en función de la frecuencia.

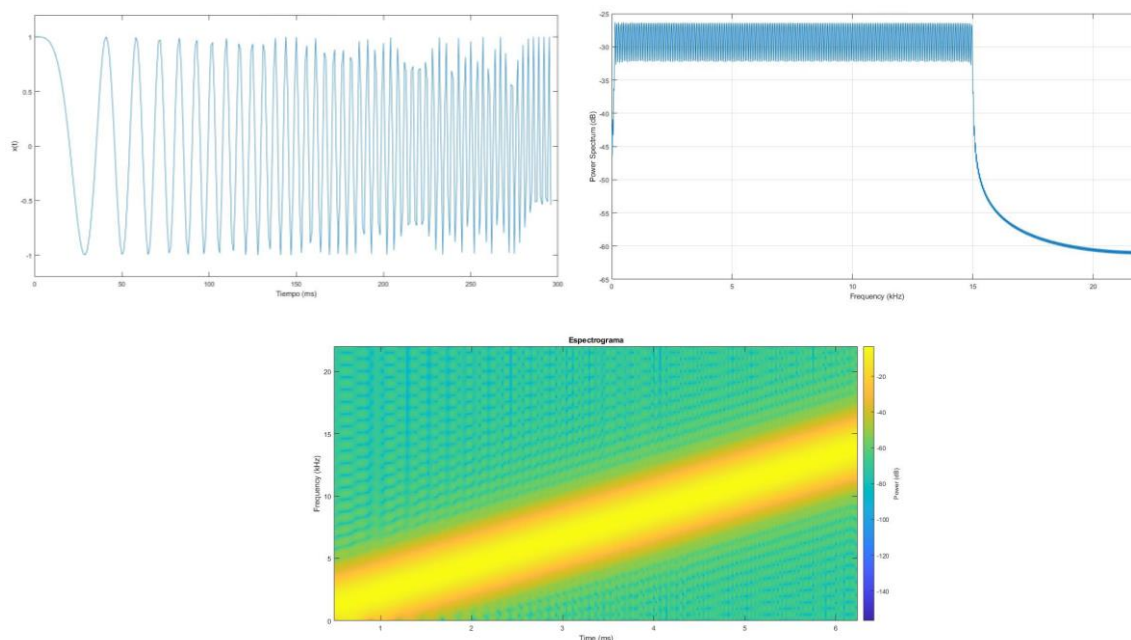


**Figura 39.** Gráficas de forma de onda, espectro y espectrograma del barrido lineal.

Esta señal se ha emitido por un altavoz situado en las posiciones que indica la Recomendación para una configuración 7.1.4 y se ha grabado mediante los auriculares *Sennheiser*, con cancelación de ruido para la obtención de una señal más limpia. Se

consiguen entonces 11 señales de dos canales, izquierdo y derecho, que se importan y muestrean a una tasa igual que la de la señal original.

Para cada respuesta al impulso, se crea un array destinado a cada canal, de forma que posteriormente nos permita generar el sistema binaural. Se lleva a cabo un proceso de normalización en el que se modifica el tamaño de las señales mediante la localización de su valor máximo y las 500 muestras anteriores y posteriores, de forma que, finalmente, se obtienen 22 respuestas al impulso (11 de cada oído) con un tamaño igual a mil muestras. Según el esquema de la Figura 38, necesitamos la señal  $x(-n)$ , que se construye simplemente volteándola, por lo que se obtiene una señal del mismo tamaño, pero con sus elementos invertidos.

La siguiente gráfica muestra la *HRTF* obtenida para una fuente sonora situada a la derecha del oyente, en las posiciones  $(\theta, \phi) = (90^\circ, 0^\circ)$ , y medida para el oído derecho.

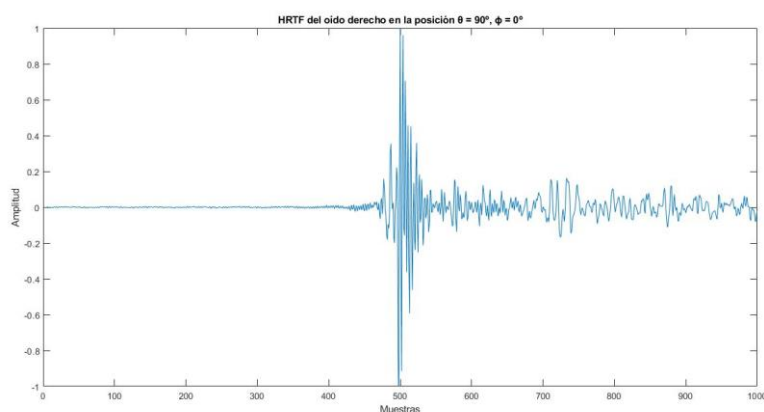


**Figura 40.** *HRTF* medida para una fuente sonora situada a la derecha del oyente en el oído derecho.

Finalmente, se obtienen las 22 *HRTF* mediante la convolución de cada una de las señales grabadas por el barrido original y las que serán imprescindibles para la generación del sistema binaural. Es conveniente mencionar que todas ellas se han normalizado para que sus valores se encuentren entre  $\pm 1$ , pues se ha comprobado durante la implementación a través del Plug – In que, para los valores originales, la señal resultante se satura y no se reproduce correctamente.

## 4.2. DESARROLLO DEL SISTEMA EN PYTHON

Para la programación del sistema en Python se ha utilizado el software *Anaconda*, una plataforma de código abierto que cuenta con más de 25 millones usuarios en el mundo. Permite la elaboración de programas mediante más de 7500 paquetes de aprendizaje máquina (*machine learning*) y ciencias de datos (*data science*). Cuenta con numerosas aplicaciones, entre las que se encuentran: *Spyder*, *Jupyter* y *RStudio*. Se ha elegido *Spyder* (*Scientific Python Development EnviRonment*) porque consiste en un Entorno de Desarrollo Integrado muy potente con edición avanzada y testeado y depuración interactivas.

#### 4.2.1. PREPARACIÓN DEL ENTORNO

*Anaconda* cuenta con numerosas y potentes librerías para el tratamiento de datos, las cuales deben ser importadas si se quieren utilizar sus funciones y recursos. Para nuestro desarrollo, se han necesitado las siguientes:

- NumPy (*Numerical Python*): se utiliza cuando se quiere trabajar con conjuntos de datos (*arrays*), matrices, álgebra lineal, transformada de Fourier.... Fue creado en 2005 por Travis Oliphant y su origen viene de la necesidad de manejar datos de manera rápida.
- Matplotlib: es la librería de visualización de *Python*. Permite la creación de gráficos estáticos, dinámicos e interactivos.
- Math: permite el uso de funciones matemáticas.
- Pandas: es una herramienta de manejo y análisis de datos muy potente, así como rápida y fácil de usar.
- Scipy: es un ecosistema de software de código abierto creado para con fines matemáticos, científicos e ingenieriles. Incluye todas las bibliotecas anteriores, además de la librería *SciPy* para la computación científica. Por ejemplo, se ha utilizado para la lectura de archivos de audio con formato *.wav*.

#### 4.2.2. GENERADOR DE SEÑALES Y CODIFICADOR AMBISONICS

Con tal de probar nuestro sistema, necesitamos una señal que esté compuesta por los 16 canales Ambisonics pertenecientes al tercer orden del formato – B. Para ello, se ha generado un pulso de una frecuencia dada, 300 Hz, que se enviará a un altavoz virtual situado en una posición determinada por los ángulos  $\theta$  y  $\phi$ , mencionados en múltiples ocasiones a lo largo de esta memoria. Así, se han podido probar fuentes en las posiciones de la configuración 7.1.4.

Utilizando la definición de armónicos esféricos de la Tabla 4, los valores de elevación y acimut elegidos y la señal senoidal generada, se obtiene una matriz con 16 filas (los canales Ambisonics) y tantas columnas como muestras tenga la señal. A esta variable la denominaremos *AB\_matrix*, y será muy utilizada en los bloques siguientes.

Por ejemplo, si queremos obtener los armónicos esféricos de una fuente a la derecha del oyente, solo hay que introducir los valores  $(\theta, \phi) = (90^\circ, 0^\circ)$  y se generará la matriz mencionada junto con la figura de una esfera con un punto en la posición deseada, de forma que se pueda comprobar que la ubicación es correcta. A continuación, se muestran cuatro gráficas para puntos situados en distintas posiciones sobre la esfera de diámetro unidad.

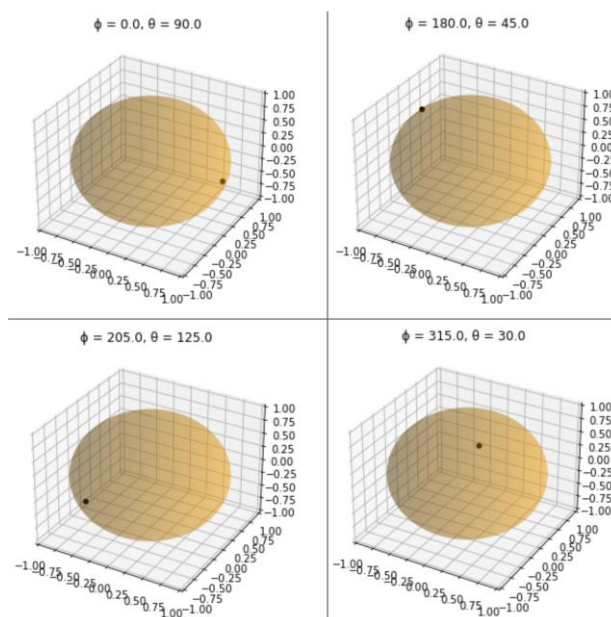


**Figura 41.** Representaciones de un punto sobre una esfera

#### 4.2.3. ROTADOR VIRTUAL DE CANALES

El siguiente paso para la presentación binaural es la traslación de los canales de audio, o posiciones de altavoces, de manera que se sitúen en frente del oyente. Para ello, utilizamos la rotación de funciones explicadas en la sección 2.3.4, en concreto las definiciones de la Ecuación 19. Definición de rotación de armónicos esféricos, la Ecuación 20. Función de Wigner-D y Ecuación 21. Función de Wigner-d. Recordamos que la función de Wigner – D depende de la Wigner – d y ésta, a su vez, de los polinomios de Jacobi. Gracias a la biblioteca *Scipy*, que incluye una función denominada `eval_jacobi` dentro de su módulo `special`, la cual resulta de gran utilidad ya que permite evaluar el polinomio de Jacobi en un punto determinado. El resto de las funciones matemáticas necesarias (potencias, valores absolutos, funciones seno y coseno, exponencias, factoriales y raíces) se obtienen de las librerías *Math* y *NumPy*.

La rotación implementada admite como parámetros los ángulos de rotación de Euler,  $(\alpha, \beta, \gamma)$ , y el orden de Ambisonics  $N$  y devuelve las funciones de Wigner – D almacenadas en una matriz llamada  $D$ , cuyo tamaño depende del orden elegido.

| $N$ | Tamaño de la matriz $D$ |
|-----|-------------------------|
| 1   | (4 x 4)                 |
| 2   | (9 x 9)                 |
| 3   | (16 x 16)               |

**Tabla 5.** Relaciones entre orden de Ambisonics y Wigner – D.

Curiosamente, la matriz resultante presenta sus valores mediante una disposición fija en filas y columnas. Cuando el valor de  $N = 1$ , se obtiene un valor en la posición  $(0, 0)$  y una

“submatriz” de tamaño  $(3 \times 3)$  cuyo valor izquierdo superior se sitúa en la posición  $(1, 1)$  de  $D(4 \times 4)$ . Si el orden aumenta en una unidad,  $N = 2$ , aparece una nueva *submatriz* de  $(5 \times 5)$  situada en  $(4, 4)$ , dando lugar al tamaño  $(9 \times 9)$  de  $D$ . Por último, si  $N = 3$ , el valor  $(0, 0)$  de la *submatriz*, que en este caso es de  $(7 \times 7)$ , se encuentra en la casilla  $(9, 9)$ . Un apunte importante a destacar es que, para cualquier orden, el resultado de la función de Wigner –  $D$  para la posición  $(0, 0)$  es 1. Esto ocurre debido a que la primera iteración cuenta con un único valor distinto de cero, el del polinomio de Jacobi (cuyo resultado es la unidad independiente del valor del ángulo  $\beta$ ).

Con tal de clarificar lo explicado en el párrafo anterior, se presenta en la siguiente figura la disposición de los valores de una matriz  $D$  para  $N = 3$  y  $(\alpha, \beta, \gamma) = (30^\circ, 30^\circ, 30^\circ)$ .

|    | 0 | 1     | 2     | 3     | 4        | 5       | 6     | 7        | 8       | 9        | 10       | 11      | 12       | 13       | 14      | 15      |
|----|---|-------|-------|-------|----------|---------|-------|----------|---------|----------|----------|---------|----------|----------|---------|---------|
| 0  | 1 | 0     | 0     | 0     | 0        | 0       | 0     | 0        | 0       | 0        | 0        | 0       | 0        | 0        | 0       | 0       |
| 1  | 0 | 0.47  | 0.31  | 0.067 | 0        | 0       | 0     | 0        | 0       | 0        | 0        | 0       | 0        | 0        | 0       | 0       |
| 2  | 0 | -0.31 | 0.87  | 0.31  | 0        | 0       | 0     | 0        | 0       | 0        | 0        | 0       | 0        | 0        | 0       | 0       |
| 3  | 0 | 0.067 | -0.31 | 0.47  | 0        | 0       | 0     | 0        | 0       | 0        | 0        | 0       | 0        | 0        | 0       | 0       |
| 4  | 0 | 0     | 0     | 0     | -0.44    | 8.3e-17 | 0.077 | 0.029    | 0.0045  | 0        | 0        | 0       | 0        | 0        | 0       | 0       |
| 5  | 0 | 0     | 0     | 0     | -1.1e-16 | 0.34    | 0.46  | 0.18     | 0.029   | 0        | 0        | 0       | 0        | 0        | 0       | 0       |
| 6  | 0 | 0     | 0     | 0     | 0.077    | -0.46   | 0.63  | 0.46     | 0.077   | 0        | 0        | 0       | 0        | 0        | 0       | 0       |
| 7  | 0 | 0     | 0     | 0     | -0.029   | 0.18    | -0.46 | 0.34     | 1.1e-16 | 0        | 0        | 0       | 0        | 0        | 0       | 0       |
| 8  | 0 | 0     | 0     | 0     | 0.0045   | -0.029  | 0.077 | -8.3e-17 | -0.44   | 0        | 0        | 0       | 0        | 0        | 0       | 0       |
| 9  | 0 | 0     | 0     | 0     | 0        | 0       | 0     | 0        | 0       | -0.81    | -0.46    | -0.11   | 4.3e-18  | 0.0081   | 0.0024  | 0.0003  |
| 10 | 0 | 0     | 0     | 0     | 0        | 0       | 0     | 0        | 0       | 0.46     | -0.26    | 1.1e-16 | 0.15     | 0.083    | 0.021   | 0.0024  |
| 11 | 0 | 0     | 0     | 0     | 0        | 0       | 0     | 0        | 0       | -0.11    | -1.7e-16 | 0.19    | 0.52     | 0.32     | 0.083   | 0.0081  |
| 12 | 0 | 0     | 0     | 0     | 0        | 0       | 0     | 0        | 0       | -4.3e-18 | 0.15     | -0.52   | 0.32     | 0.52     | 0.15    | 4.3e-18 |
| 13 | 0 | 0     | 0     | 0     | 0        | 0       | 0     | 0        | 0       | 0.0081   | -0.083   | 0.32    | -0.52    | 0.19     | 1.7e-16 | -0.11   |
| 14 | 0 | 0     | 0     | 0     | 0        | 0       | 0     | 0        | 0       | -0.0024  | 0.021    | -0.083  | 0.15     | -1.1e-16 | -0.26   | -0.46   |
| 15 | 0 | 0     | 0     | 0     | 0        | 0       | 0     | 0        | 0       | 0.0003   | -0.0024  | 0.0081  | -4.3e-18 | -0.11    | 0.46    | -0.81   |

**Figura 42.** Resultado de la Matriz de Wigner –  $D$  para valores concretos de  $(\alpha, \beta, \gamma)$ .

Finalmente, esta matriz se multiplica por la señal Ambisonics,  $AB\_matrix$ , dando como resultado la señal rotada en una variable de 16 canales, cada uno de ellos con el número de muestras de la señal. Esta señal se denomina  $B\_rot$ .

#### 4.2.4. DECODIFICACIÓN A ITU 7.1.4

Siguiendo en la cadena de procesamiento de la señal Ambisonics, hemos llegado al punto en el que se desea transformar los 16 canales a una configuración 7.1.4, en la que ya sabemos que contamos con 12 altavoces. No obstante, uno de ellos, el *subwoofer*, no será tenido en cuenta en la decodificación, ya que éste se encarga de la banda de bajas frecuencias de la señal y no se considera objetivo de este trabajo. Aun así, veremos que la señal de salida sí que cuenta con 12 canales, resultado del proceso de decodificación.

El Ingeniero acústico Jérôme Daniel detalla el proceso de decodificación de una señal Ambisonics a cualquier configuración de altavoces en su Tesis titulada “Representación de campos acústicos, aplicación a la transmisión y reproducción de escenas sonoras complejas en un contexto multimedia”, concretamente en el capítulo 3: “Extensión del formalismo y de las soluciones: de la codificación a la decodificación” [48]. La siguiente muestra los elementos y relaciones que forman un sistema Ambisonics.

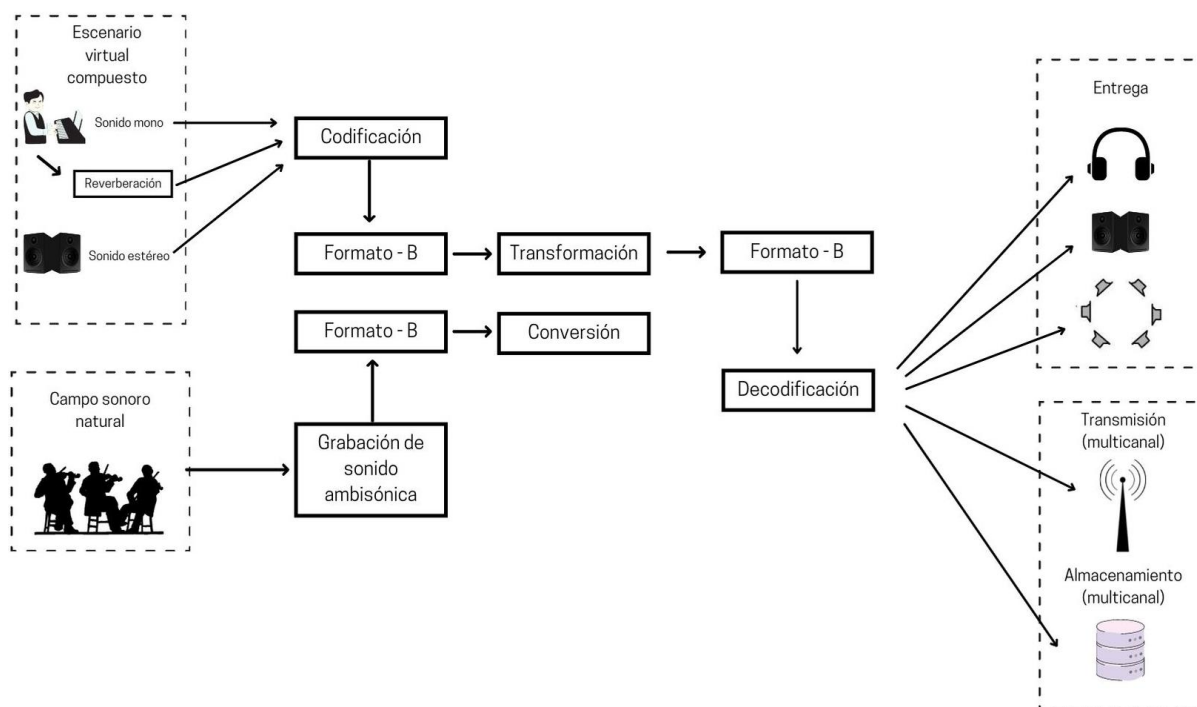


**Figura 43.** Esquema general de un sistema Ambisonics <sup>18</sup>.

Para el tratamiento de estas señales se necesitan operaciones lineales simples junto con un filtrado final, las cuales pueden ser descritas de manera matricial, donde  $M_e$  y  $M_d$  serán las matrices de codificación y decodificación, respectivamente. Con tal de poder explicar y entender correctamente la transformación de Ambisonics de tercer orden a la configuración de altavoces deseada, se requiere en primer lugar conocer el procedimiento básico de la codificación.

### CODIFICACIÓN AMBISONICS DE TERCER ORDEN

En la sección 2.5.1 veíamos que, para orden 1, contábamos con las componentes  $W$ ,  $X$ ,  $Y$  y  $V$  de Ambisonics. También se mostraba que cada una de ellas hacía referencia a las distintas directividades de los canales de HOA, siendo  $W$  la omnidireccional y el resto las bidireccionales. En el caso de orden 3,  $M = 3$ , los canales se interpretan como el conjunto de  $N = 2M + 1$  micrófonos con distintos diagramas polares especificados mediante  $D_W, D_X, D_Y, D_U, D_V, D_P$  y  $D_Q$ , donde los subíndices indican las componentes o canales. En este caso, no sólo se obtienen omni y bidireccionalidad, sino que aparecen nuevos diagramas más complejos y sofisticados (tetradireccionales y hexadireccionales). Recordamos que los diagramas polares o de directividad describen la ganancia de un dispositivo receptor, en este un micrófono, en función de una dirección determinada. Las funciones que definen los diagramas para  $M = 3$  son las mostradas en el conjunto de ecuaciones a la izquierda y sus directividades en las gráficas de la derecha [49]. El ángulo  $\varphi$  indica la posición de la fuente.

<sup>18</sup> Esquema elaborado por la autora de este TFM a partir de la documentación de la tesis de Jérôme Daniels, capítulo 3. [48]

$$\begin{aligned} D_W(\varphi) &= 1 \\ D_X(\varphi) &= \cos \varphi & D_Y(\varphi) &= \sin \varphi \\ D_U(\varphi) &= \cos 2\varphi & D_V(\varphi) &= \sin 2\varphi \\ D_P(\varphi) &= \cos 3\varphi & D_Q(\varphi) &= \sin 3\varphi \end{aligned}$$

**Ecuación 31.** Definición de las directividades de  $N = 7$  micrófonos.

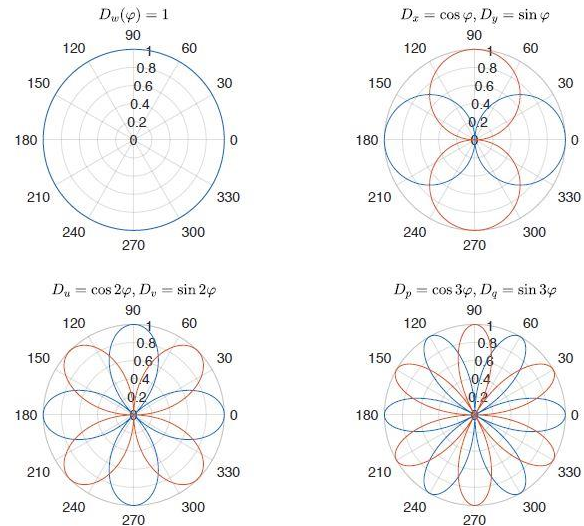


**Figura 45.** Gráfica de las directividades de  $N = 7$  micrófonos.

Los 7 canales (obtenidos cada uno de ellos por un micrófono) pueden definirse también en función de sus directividades y, además, de las amplitudes  $a_\psi$  de las señales que vienen del ángulo  $\psi$ . De esta forma, la información sobre la dirección de llegada de un sonido está contenida en los canales codificados. Si, por ejemplo, la señal se envía desde una posición que forma un ángulo  $\psi = 90^\circ$ , tendremos información en el canal omnidireccional ( $W$ ) y en  $Y, V$  y  $Q$  con un valor  $a_\psi$ , mientras que en  $X, U$  y  $P$  no contarán con ningún registro de la señal.

$$\begin{aligned} W(\psi) &= a_\psi \\ X(\psi) &= a_\psi \cos \psi & Y(\psi) &= a_\psi \sin \psi \\ U(\psi) &= a_\psi \cos 2\psi & V(\psi) &= a_\psi \sin 2\psi \\ P(\psi) &= a_\psi \cos 3\psi & Q(\psi) &= a_\psi \sin 3\psi \end{aligned}$$

**Ecuación 32.** Definición de los  $N = 7$  canales.

## DECODIFICACIÓN AMBISONICS DE TERCER ORDEN

Un altavoz se puede representar mediante un modelo Ambisonics, por lo que, al igual que los micrófonos en codificación, se consigue especificarlos mediante canales de HOA, de forma que se organicen en función de sus direccionalidades. De tal forma, se puede codificar la onda sonora  $a_n$  producida por un altavoz enésimo situado en una posición determinada  $\varphi_n$  obteniendo los canales siguientes:

$$\begin{aligned} W_n &= a_n \\ X_n &= a_n \cos \varphi_n & Y_n &= a_n \sin \varphi_n \\ U_n &= a_n \cos 2\varphi_n & V_n &= a_n \sin 2\varphi_n \\ P_n &= a_n \cos 3\varphi_n & Q_n &= a_n \sin 3\varphi_n \end{aligned}$$

**Ecuación 33.** Canales High Order Ambisonics para  $N = 7$ .

Así, por ejemplo, el canal omnidireccional estará formado por la suma de la componente omnidireccional de todos los altavoces. Lo mismo ocurre para el resto de las directividades para un total de  $N_l$  altavoces.

$$\begin{aligned}
 W &= \sum_{n=1}^{N_l} W_n = \sum_{n=1}^{N_l} a_n \\
 X &= \sum_{n=1}^{N_l} X_n = \sum_{n=1}^{N_l} a_n \cos \varphi_n & Y &= \sum_{n=1}^{N_l} Y_n = \sum_{n=1}^{N_l} a_n \sin \varphi_n \\
 U &= \sum_{n=1}^{N_l} U_n = \sum_{n=1}^{N_l} a_n \sin 2\varphi_n & V &= \sum_{n=1}^{N_l} V_n = \sum_{n=1}^{N_l} a_n \cos 2\varphi_n \\
 P &= \sum_{n=1}^{N_l} P_n = \sum_{n=1}^{N_l} a_n \cos 3\varphi_n & Q &= \sum_{n=1}^{N_l} Q_n = \sum_{n=1}^{N_l} a_n \sin 3\varphi_n
 \end{aligned}$$

**Ecuación 34.** Definición de las componentes direccionales HOA.

Estas ecuaciones tienen forma de transformada de Fourier  $\mathcal{F}$ , por lo que puede aplicarse la transformada inversa  $\mathcal{F}^{-1}$  para obtener las muestras de la señal sonora del altavoz  $n$  mediante la siguiente relación, siempre y cuando los altavoces estén equiespaciados:

$$a_n = \frac{1}{N_l} (W + 2X \cos \varphi_n + 2Y \cos \varphi_n + 2U \cos 2\varphi_n + 2V \sin 2\varphi_n + P \cos 3\varphi_n + Q \sin 3\varphi_n)$$

**Ecuación 35.** Definición de muestra sonora para un altavoz  $n$ ésimo en una configuración equiespaciada de los mismos.

Llegados a este punto podemos describir las operaciones de codificación y decodificación de manera matricial a través de la siguiente relación, donde a la izquierda de la igualdad tenemos los componentes Ambisonics y a la derecha la matriz de codificación  $M_e$  y las fuentes (o altavoces),  $P_N$ .

$$\begin{pmatrix} W \\ X \\ Y \\ U \\ V \\ P \\ Q \end{pmatrix} = M_e \cdot \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ \vdots \\ P_N \end{pmatrix}, \quad \text{donde } M_e = \begin{pmatrix} 1 & 1 & 1 \\ a_n \cos \varphi_1 & a_n \cos \varphi_2 & a_n \cos \varphi_N \\ a_n \sin \varphi_1 & a_n \sin \varphi_2 & a_n \sin \varphi_N \\ a_n \cos 2\varphi_1 & a_n \cos 2\varphi_2 & \dots & a_n \cos 2\varphi_N \\ a_n \sin 2\varphi_1 & a_n \sin 2\varphi_2 & & a_n \sin 2\varphi_N \\ a_n \cos 3\varphi_1 & a_n \cos 3\varphi_2 & & a_n \cos 3\varphi_N \\ a_n \sin 3\varphi_1 & a_n \sin 3\varphi_2 & & a_n \sin 3\varphi_N \end{pmatrix}$$

**Ecuación 36.** Definición matricial de codificación Ambisonics.

Se podría deducir que, para conseguir los valores de las fuentes  $P_N$  mediante la reconstrucción de las ondas planas<sup>19</sup> correspondientes, la matriz de decodificación es la inversa que la de codificación. Sin embargo, no se cumplen las condiciones para la invertibilidad (la matriz no es cuadrada y su determinante puede no ser distinto de cero). Por lo tanto, Daniel define que la matriz de decodificación  $M_d$  es la pseudoinversa de  $M_e$ . La matriz pseudoinversa de Moore – Penrose surge como una generalización de la matriz inversa y se define como:

$$M_d = M_e^t \cdot (M_e \cdot M_e^t)^{-1}$$

**Ecuación 37.** Definición de matriz de decodificación como la pseudoinversa de la matriz de codificación.

Con lo que, finalmente, llegamos a la generalización de la obtención de las fuentes mediante los componentes *High Order Ambisonics*.

$$\begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ \vdots \\ P_N \end{pmatrix} = M_d \cdot \begin{pmatrix} W \\ X \\ Y \\ U \\ V \\ P \\ Q \end{pmatrix}$$

**Ecuación 38.** Definición de decodificación de Ambisonics.

Observamos que es necesario conocer el valor de  $M_e$  para la decodificación y que, hasta ahora, se ha definido la codificación para un valor de fuentes  $N = 7$ . Sabemos que la configuración de nuestro sistema de altavoces es ITU 7.1.4, por lo que se requieren 11 altavoces. Además, debemos conocer los valores de las directividades para cada altavoz. Como hemos expuesto a lo largo de este trabajo, existe una relación entre los valores de las directividades de los armónicos esféricos con la representación de Ambisonics en el formato – B para cualquier orden. Así, la matriz  $M_e$  se caracterizará por los siguientes elementos:

- Tendrá un tamaño de (12 x 12).
- La primera fila, tal y como ocurre en la definición de la Ecuación 36, estará formada por unos, los cuales representan el diagrama de directividad omnidireccional.
- En las siguientes 11 filas se tendrán los valores de la Tabla 4. Definición de Ambisonics formato B en relación con los armónicos esféricos [EP]. especificados para cada par de ángulos. Es decir, cada fila representará cada posición de los altavoces mediante su valor de Ambisonics de formato – B y su par de ángulos  $(\theta, \phi)$ , valor que aparece recogido en la Recomendación ITU BS-775.3, y en la especificación de Dolby (ver Figura 23 y Figura 24).

Finalmente, una vez obtenida la matriz de codificación y su pseudoinversa, ésta se multiplica a la señal rotada,  $B\_rot$ , o a la señal sin rotar,  $AB\_matrix$ , obteniendo la señal formada por los canales que irían a los altavoces,  $P\_louds$ . Estará formada por 12 filas y

<sup>19</sup> Se considera que las ondas sonoras que llegan al oyente son planas, ya que se asume que se emiten a una distancia tal que sus frentes de onda se consideran planos cerca del punto de escucha.

tantas columnas como muestras tenga la señal. El siguiente paso será la decodificación al sistema biaural.

#### 4.2.5. DECODIFICACIÓN A UN SISTEMA BIARUAL

Para conseguir elaborar un sistema de presentación biaural a partir de una configuración específica de altavoces, es absolutamente necesario obtener las distintas respuestas al impulso de la sala con respecto al oyente para las posiciones de dichos altavoces. En la sección 4.1 se detalla cómo se han generado las respuestas al impulso para todas las posiciones de los altavoces en la sala y para cada oído. La señal biaural estará formada por dos canales, izquierdo y derecho, cada uno de ellos compuesto por la suma de las señales de cada oído filtradas por su respuesta al impulso.

Como sabemos, el filtrado en el dominio del tiempo se lleva a cabo mediante la convolución, mientras que en el dominio de la frecuencia se computa mediante la multiplicación. Es por ello por lo que se ha elegido el dominio de Fourier para un manejo más cómodo de las operaciones en el desarrollo mediante Python. Para convolucionar o multiplicar una señal con una respuesta al impulso, es necesario poseer toda la información acerca de la señal. En nuestro caso, la pista de audio se recibirá en bloques, por lo que no contamos con dicha información. Además, al tratarse de señales de larga duración, no se debe realizar la convolución de manera directa, sino que es conveniente establecer una división de la señal en bloques de longitud fija. Este proceso se denomina *Overlap and Add*, cuya definición matemática es la siguiente:

$$y[n] \triangleq x[n] * h[n] \triangleq \sum_{m=-\infty}^{\infty} h[m] \cdot x[n-m] = \sum_{m=1}^M h[m] \cdot x[n-m]$$

**Ecuación 39.** Definición del método *Overlap and Add*.

La idea es simplificar el problema mediante la convolución de segmentos de la señal original con la respuesta al impulso, que tendrá un número de muestras  $M$  (nuestros filtros tienen  $M = 1000$ ). El tamaño de los segmentos de la señal,  $L$ , será definido por el usuario y, junto con el filtro, condicionará el tamaño de la Transformada de Fourier,  $N = L + M - 1$ . En la figura siguiente, se divide la señal en bloques de 100 muestras y se realiza la convolución con la respuesta del filtro. Lo que ocurre es que el valor de  $N$  es mayor que  $L$ , de manera que aparece una “cola”, la cual se solapará al inicio del bloque siguiente. Así, se conseguirá una reconstrucción correcta de la señal, sin efectos de *aliasing* ni pérdida de información.

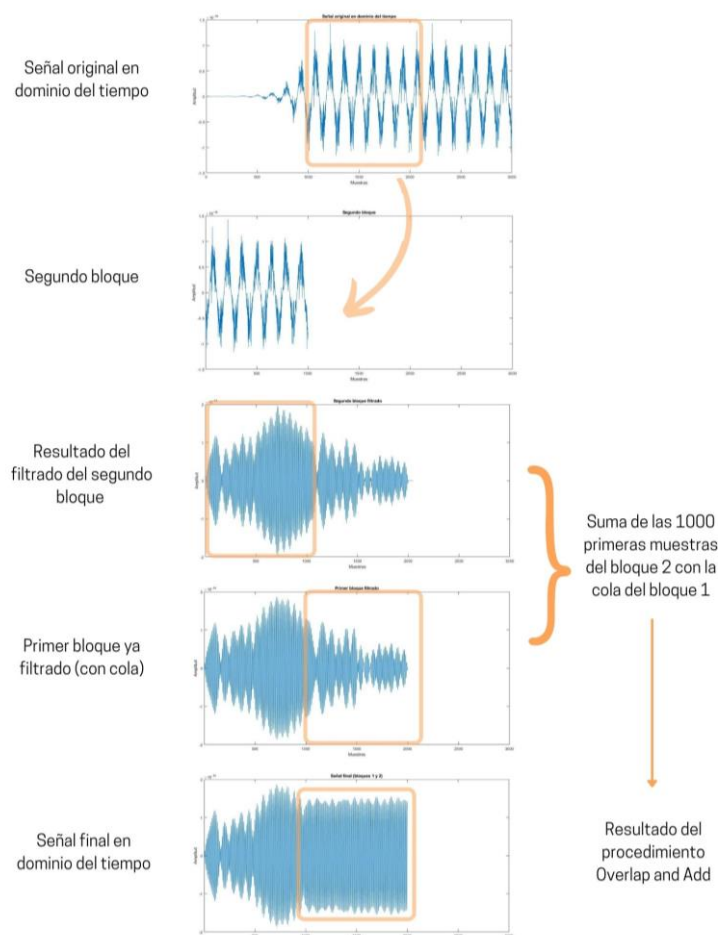


**Figura 46.** Método *Overlap and Add* de manera gráfica [EP]<sup>20</sup>.

En nuestro caso, la señal se dividirá en bloques de mil muestras, a los que se le aplicará la Transformada de Fourier de tamaño  $N = 2048$ . Este valor viene dado por la aproximación a la potencia de dos de  $N = L + M - 1$ , que se utiliza para mejorar la eficiencia de computación. Por otro lado, también se aplicará dicha Transformada, con el mismo tamaño, a cada una de las respuestas al impulso. Contaremos entonces con dos señales en el dominio de la frecuencia de tamaño 2048 muestras, pero que únicamente contarán con información en las mil primeras. Tras multiplicar, se aplica la transformada de Fourier inversa al resultado, y se obtienen 2048 muestras de las cuales:

- Las mil primeras serán el resultado de la multiplicación del bloque en cuestión con el filtro.
- Las mil segundas serán la cola.
- Las 48 muestras restantes serán ceros.

Para el siguiente bloque se lleva a cabo el mismo proceso, pero, una vez se obtienen las mil primeras muestras, éstas se suman a la cola del bloque anterior dando como resultado al segundo bloque de la señal resultante. Las siguientes mil muestras, serán la cola del segundo bloque y se sumarán al resultado de la multiplicación del tercero con la respuesta al impulso del filtro. Finalmente, se obtiene una señal en el dominio del tiempo (ya que se ha ido

<sup>20</sup> La representación gráfica del método OLA se ha obtenido de manera manual mediante el código elaborado para implementar dicho algoritmo.

computando la transformada de Fourier inversa tras cada multiplicación) correctamente filtrada y sin efectos de distorsión por submuestreo.

Recordamos que el resultado de la decodificación a 7.1.4 es la matriz  $P_{\text{louds}}$  y que cada una de sus filas contiene la señal del altavoz correspondiente. Se computa entonces el método *OLA* de forma que la señal coincida con el filtro de la fuente adecuada. Por ejemplo, para obtener el resultado de la señal proveniente del altavoz situado en frente del oyente, se ha utilizado la fila de la matriz correspondiente y la respuesta al impulso medida desde dicha posición para el oído derecho, por un lado, y para el izquierdo por otro. Se consiguen dos resultados independientes que formarán parte, respectivamente, de los canales binaurales. Realizando esta operación a todas las señales, se consiguen las 22 resultantes, 11 para cada canal.

Finalmente, se suman las soluciones y se almacenan en dos *arrays*, denominados  $\text{Bin\_R}$  y  $\text{Bin\_L}$ , que serán cada uno de los canales de la señal binaural  $\text{Bin}$ , llegando al objetivo de este Proyecto.

### 4.3. DESARROLLO DEL PLUG - IN

La implementación del sistema mediante Python ha permitido comprobar el correcto funcionamiento de cada uno de los bloques y, por lo tanto, podemos confirmar que el proyecto es satisfactorio. Aun así, se ha querido ir un paso más allá, de manera que esta cadena de procesamiento pueda ser utilizada en un entorno real, esto es, una Estación Digital de Audio. Como ya se ha especificado, este Plug – In se ha implementado mediante *JUCE* y *Microsoft Visual Studio* en el lenguaje de programación C++.

A continuación, se explica la construcción de la *GUI* y de los bloques de procesamiento de audio.

#### 4.3.1. ELABORACIÓN DE LA INTERFAZ

La Interfaz Gráfica de Usuario, *GUI*, se define en los archivos de edición, tanto en la cabecera como en el C++. Este elemento es muy importante debido a que será la única manera mediante la cual el usuario podrá hacer uso del Plug – In. Por tanto, debe ser intuitiva, atractiva y eficiente. Recordemos que en este programa se ofrecen las siguientes funcionalidades:

- Elección de los ángulos de rotación.
- Activación de la operación de rotación.
- Decodificación a un sistema de altavoces 7.1.4.
- Decodificación a un sistema binaural.

En primer lugar, hablaremos de los elementos que formarán parte de la Interfaz. Para ello, su declaración debe instanciarse en la cabecera *PluginEditor.h*, en su sección de atributos públicos. *JUCE* cuenta con una clase denominada *Slider*, que consiste en una función de cambio de atributos para un determinado elemento. De este modo, cuando el usuario interactúe con estos valores, el programa lo reconocerá y activará las funcionalidades correspondientes mediante una conexión realizada internamente. Se crean entonces 3 elementos de este tipo para la elección de los valores  $(\alpha, \beta, \gamma)$  y otros 3 para la activación de las funciones de Rotación, Decodificación a 7.1.4 y Decodificación a Binaural. En este fichero, también se crean las cabeceras de las funciones para renderizar los elementos de la *GUI*.

En PluginEditor.cpp elaboraremos la distribución y apariencia de los elementos mediante funciones públicas ofrecidas por la clase Slider y su declaración en el constructor. Entre las más importantes se encuentran:

- void Slider::addListener (Listener \* listener): permite que el objeto reconozca si su valor ha cambiado.
- void Slider::setRange (double newMinimum, double newMaximum, double newInterval = 0): establece el rango de valores modificable mediante ese “deslizador”. En el caso de los ángulos de rotación, los valores irán entre 0° y 360°, mientras que en los elementos de activación de las funciones sólo tendrán dos posibles opciones: 0 (off) y 1 (on).
- Funciones de estilización: tipo de deslizador (rotacional o lineal), color, texto, etc.

Para que estos elementos se muestren en pantalla, es obligatorio llamar a la función addAndMakeVisible(SliderName).

Por otro lado, existe una función predefinida denominada paint, la cual determina la apariencia del elemento de la clase Component, ofrecida por *JUCE* para albergar todos objetos de la interfaz de usuario. En ella se instancian los atributos gráficos y la disposición de los elementos mencionados.

Finalmente, se llama a la función sliderValueChanged(Slider\* slider) para asociar a un parámetro el valor seleccionado por el usuario mediante el Slider correspondiente. Dicho parámetro se crea en la cabecera del procesador del Plug – In, lo cual permitirá elaborar una conexión entre la *GUI* y el bloque de proceso.

En la siguiente figura se muestra la interfaz del Plugin:

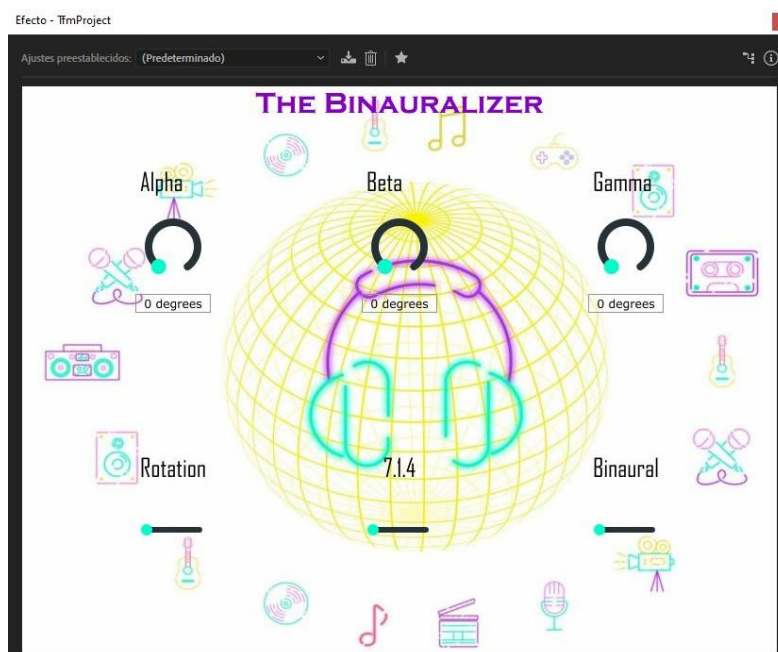


**Figura 47.** Interfaz del Plug – In desarrollado.

Previamente a comenzar la explicación de los bloques, señalamos que la generación de señales no es necesaria en nuestro Plug – In, ya que se introducirá directamente desde la *DAW*. Para que el programa sea capaz de leer y escribir el número de canales necesarios, se debe especificar en el constructor mediante el indicador ambisonic(3), es decir, 16 canales para la entrada y salida. Se establece también a la salida porque el usuario puede simplemente rotar los canales (obteniendo los mismos que a la entrada), decodificarlos a

7.1.4 (resultando en este caso en doce canales) o decodificarlos a binaural. Se debe ofrecer la posibilidad de generar el mayor número de señales necesario.

Para asociar cada canal a una variable y así poder trabajar con ella, la biblioteca de *JUCE* ofrece la función `getWritePointer(int channelNumber)`, y se instancia mientras la llamada al puntero del canal indicado en el valor pasado como parámetro. Como mencionábamos en la sección del Estado del Arte dedicada a *JUCE*, existe un solo *buffer* para entrada y salida, por lo que debemos tener cuidado a la hora de trabajar con cada uno de los canales.

#### 4.3.2. ROTADOR VIRTUAL DE CANALES

Para llevar a cabo el proceso de rotación, al igual que en Python, se ha obtenido la matriz de rotación *D* que, como sabemos, depende de los ángulos de rotación. En este caso, el usuario selecciona el valor para cada uno de ellos y activa el *Slider* de rotación. En ese momento, se realiza una conexión con el bloque de proceso, *processBlock*, y se comienza la construcción de la matriz y de la aplicación a la señal de entrada.

Comenzamos con la generación de la matriz *D*. El problema es que, al tratarse de un lenguaje de bajo nivel, la implementación se ha tenido que llevar a cabo de manera diferente con respecto al sistema anterior. La computación de los valores que implican un uso de recursos matemáticos básicos no ha supuesto ninguna complicación adicional. Sin embargo, recordamos que las funciones de rotación hacen uso de los polinomios de Jacobi. Pues bien, en C++ no existe una función predefinida para la obtención de dichos polinomios, de manera que se ha recurrido a un estudio exhaustivo de los elementos que forman las matrices de rotación. Para ello y, gracias a la ayuda de Francisco Javier, se ha analizado la naturaleza de los valores y su relación con su posición.

Resaltamos la propiedad de la matriz de rotación que mencionábamos en el apartado 4.3.2, que indica que dicha matriz está formada por un número determinado de “submatrices” en función del orden del formato Ambisonics y que el resto de las posiciones tienen el valor 0. Esto ayuda a simplificar el problema de obtención de polinomios de Jacobi. Así, se ha conseguido obtener un patrón entre los valores de dichos polinomios, que resultan ser funciones de  $\cos \beta$ , y su posición en la matriz. Mediante una consecución de secuencias condicionales en las que se comprueba la posición del elemento a calcular, se ha conseguido obtener el valor correcto de cada casilla de la matriz. Para comprobar la veracidad del resultado, se ha comparado insertando los mismos valores de los ángulos de Euler en el sistema de Python.

Una vez generada la matriz, el siguiente paso es su multiplicación por la señal de entrada. Este proceso también debe llevarse a cabo de manera distinta ya que no contamos con la señal completa ni en forma de matriz, sino que las muestras de cada canal se van leyendo conforme la *DAW* las envía. Para cada una de ellas, se lee el búfer de cada canal y se guardan sus valores en una variable intermedia de tamaño 16. Por ejemplo, se comienza leyendo los 16 canales para la muestra inicial, por lo que se obtendrán 16 muestras, cada una de ellas almacenada en una posición del *array* intermedio denominado *InBuffer*. También se crea un *array* intermedio de salida *OutBuffer*. Con tal de conseguir los canales de salida, se vuelve a aprovechar la propiedad de la matriz de rotación y se multiplican los valores según el número de canal que se esté computando, de tal forma que la multiplicación de matrices se reduce a multiplicaciones término a término de la siguiente manera:

- Para el canal 0, el valor resultante es simplemente la multiplicación de la posición  $[0,0]$  de la matriz por el valor almacenado en la primera posición de *InBuffer*.
- Para el canal 1, se multiplica la fila 1 de la matriz de rotación, que únicamente cuenta con valores en sus posiciones  $[1,1]$ ,  $[1,2]$  y  $[1,3]$ , término a término con las

posiciones 1, 2 y 3 de InBuffer, respectivamente, y el resultado del canal de salida, OutBuffer[1], estará constituido por la suma de los resultados obtenidos. Lo mismo ocurre para los canales 2 y 3, solo que en este caso se multiplican las filas 2 y 3 de la matriz con las posiciones 1, 2 y 3 de InBuffer.

- Para los canales 4, 5, 6, 7 y 8, también se lleva a cabo una combinación lineal entre los valores de sus respectivas filas de D y los búferes. Así, por ejemplo, para obtener el canal de salida 5, se multiplica la fila 5 de la matriz de rotación, que sólo tendrá valores distintos de cero en [5, 4], [5, 5], [5, 6], [5, 7] y [5, 8], por InBuffer[4], InBuffer[5], InBuffer[6], InBuffer[7] y InBuffer[8], respectivamente. El resultado se suma y se almacena en OutBuffer[5].
- El mismo comportamiento se mantiene para los canales que complementan el tercer orden, es decir, del número 9 al 15.

Una vez se hayan computado los valores para la muestra en cuestión, la cual llamaremos *muestras* a partir de ahora, y se hayan almacenado en el array intermedio OutBuffer, se requiere su asociación al *buffer* real de salida. El resultado será la señal rotada para todas las muestras de los 16 canales.

Un apunte importante que destacar es que la operación de rotación siempre se realizará sobre la señal de entrada Ambisonics. Veremos en las siguientes secciones que las operaciones pueden realizarse tanto para la señal de entrada como para la señal rotada.

#### 4.3.3. DECODIFICACIÓN A ITU 7.1.4

Como se ha explicado en la implementación en Python, para la decodificación de un sistema Ambisonics a cualquier configuración de altavoces, es necesaria la generación de la matriz de decodificación mediante la computación de la pseudoinversa de la matriz de codificación. Sus valores son fijos para las posiciones de elevación y acimut de los altavoces, por lo que se ha decidido crear una variable en lugar de computar la matriz cada vez que se haga uso de ella durante el proceso, lo que supondrá un ahorro de coste computacional.

Para llevar a cabo este proceso de decodificación, se ha recurrido al *array* de datos intermedio para poder manejar los valores de las muestras para cada uno de los canales de entrada. También se ha creado otro similar para los datos de salida, que estará compuesto por 12 elementos, cada uno de ellos correspondiente a un altavoz; que posteriormente serán asociados a los búferes de salida.

La manera de computar la multiplicación de matrices ha estado determinada, como en el caso anterior, por el hecho de que no contamos con la matriz de entrada completa, sino que vamos recibiendo muestras de la señal de audio progresivamente. Para ello, para cada canal del *array* de salida, se ha llevado a cabo una combinación lineal entre la fila de decodificación correspondiente, que como sabemos se corresponde con un altavoz concreto, y cada uno de los valores de la entrada para la muestra *s*. Por ejemplo: para obtener la muestra correspondiente al búfer de salida asociado al canal 0, se ha multiplicado el búfer de entrada asociado al canal 0 por la posición [0, 0] de la matriz de decodificación y se le ha sumado el resultado de la misma operación para el canal 1 y la posición [0, 1] de D, así hasta el último canal, el número 15. En definitiva, se trata de una operación de filas por columnas, donde las filas son aquellas de la matriz de decodificación y la columna contiene 16 muestras, cada una de ellas correspondiente a un canal de entrada distinto.

Por último, se asocian los valores obtenidos en la variable intermedia a los búferes de salida, que en este caso son 12, correspondientes a los 11 altavoces de 7.1.4 más la componente omnidireccional.

El usuario puede activar la funcionalidad de decodificación para la señal de entrada “limpia”, o para su versión rotada mediante los elementos de la interfaz destinados a ello. El programa se encargará de identificar a qué muestras debe realizar la transformación en función de la elección.

#### 4.3.4. DECODIFICACIÓN A UN SISTEMA BIAURAL

Hasta ahora, los esquemas de elaboración de cada uno de los bloques se han llevado a cabo de manera similar al sistema en Python. Si bien ha sido necesario tener en cuenta las diferencias en los lenguajes de programación, la base teórica detrás de la construcción es la misma. Para la bauralización ocurre lo mismo, sin embargo, en lugar de trabajar en el dominio de la frecuencia, como ocurría en el sistema piloto, trabajaremos en el dominio del tiempo. Es decir, computaremos la convolución entre señales en lugar de la multiplicación.

Las razones que han motivado esta decisión son las siguientes:

- La computación de la Transformada de Fourier para cada una de las señales de entrada es muy compleja. Se ha intentado elaborarla, pero ha conllevado dificultades en cuanto a su manejo y tratamiento.
- Ocurre de manera similar para la Transformada de Fourier Inversa, ya que supone una gran capacidad computacional y no resulta eficiente para la aplicación en tiempo real.

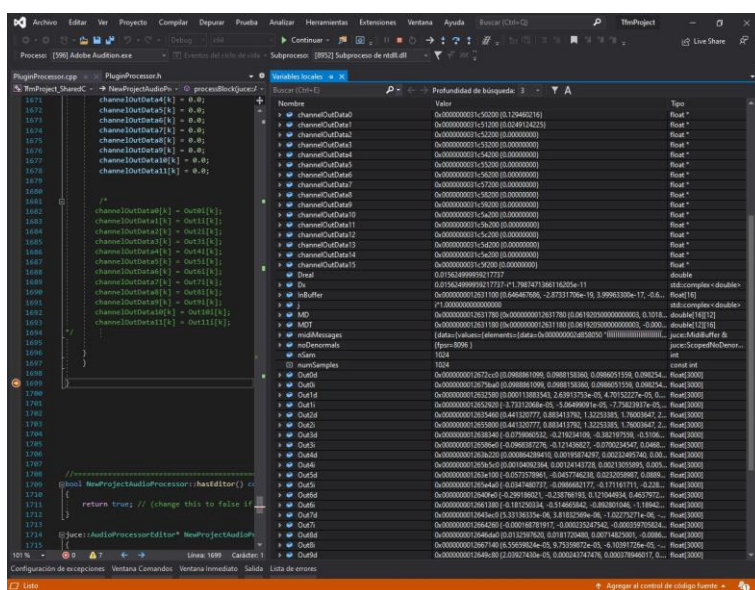
A pesar de que se ha tratado de encontrar la manera de conseguir trabajar en frecuencia, ninguna de las soluciones parecía apropiada ya que la elaboración de Fourier mediante *JUCE* no es sencilla ni cómoda. Por ello, se ha recurrido al dominio del tiempo mediante la definición formal de convolución y de la computación de *Overlap and Add*.

En primer lugar, explicamos las variables necesarias para el correcto funcionamiento de lo que se considera el bloque más importante del Plug – In, ya que será el encargado de ofrecer el resultado final mediante la presentación de la señal envolvente en formato baural. Al igual que en los bloques anteriores, se requieren *arrays* intermedios para no sobrescribir los búferes de entrada y salida. Estas variables son: los *arrays* de entrada, que almacenarán bloques de la señal para cada uno de los canales; 22 *arrays* de cola, 11 para cada oído, que servirán para la computación de *OLA*; y 2 *arrays* de salida, que estarán formados por la suma de los valores computados para el oído izquierdo y derecho. Por otro lado, también se han almacenado las respuestas al impulso elaboradas en *arrays*. La creación y declaración de todas las variables se ha hecho en el fichero de cabecera, ya que se ha comprobado que mejora la velocidad de computación de los bloques.

Para la decodificación baural, se debe accionar previamente el *slider* de decodificación 7.1.4, ya que la bauralización se lleva a cabo mediante la transformación del sonido para altavoces (físicos o virtuales) a auriculares de dos canales. Así, una vez computada (de manera transparente para el usuario), se activa el *slider* de decodificación baural y ocurre lo siguiente:

1. Se inicializan a cero los *arrays* intermedios de salida para las  $N$  muestras de lo que, en el caso de Python, era la Transformada de Fourier. En este sistema, dicho valor condiciona el tamaño de los bloques de convolución. Se considera necesario mencionar que *JUCE* trabaja con un número dinámico de muestras por bloques, por lo que el valor de  $N$  será: el número de muestras leído en el bloque, sumado al tamaño de la cola (1000) menos una unidad.
2. A continuación, se realiza la convolución entre cada una de las señales de entrada y las respuestas al impulso correspondientes.

- El entorno *Microsoft Visual Studio* tiene una función de depuración que ha sido crucial para la elaboración de este bloque: la asociación a procesos. Así, se pueden conectar el *IDE* con la *DAW* (*Adobe Audition*). Gracias a esto, se ha podido comprobar el número de las muestras de los bloques, los valores de la convolución, de la matriz de rotación, etc. A continuación, se adjunta una captura donde se observan los valores de múltiples variables involucradas en el proceso.



Tras comparar los resultados obtenidos mediante el Plug – In y el sistema construido en Python para una misma señal de entrada, llegamos a la conclusión de que ambos programas funcionan correctamente. Esto se explicará con más detalle en el apartado Resultados.

## 5. IMPLEMENTACIÓN

En esta sección se explica cómo se ha integrado el VST3 generado en dos Estaciones de Audio Digital diferentes: Reaper y Adobe Audition.

### 5.1. PREPARACIÓN DE ARCHIVOS

Nuestro Plug – In necesita, en el inicio de comprobación de los bloques, una señal de entrada de 16 canales, es decir, una señal Ambisonic de formato – B de tercer orden. Gracias a nuestro bloque “Generador de Señales”, hemos podido crear una señal sinusoidal a la cual se le han aplicado los armónicos esféricos para conseguir los canales deseados. La señal consta de 16 canales de cincuenta mil muestras y una frecuencia de 300 Hz. Posteriormente, se elige una posición mediante los valores elevación y acimut para generar el efecto Ambisonics. El resultado se guarda en una variable de extensión .wav con la forma AB\_elevation\_acimut.wav, donde se especifica el valor elegido para cada ángulo. Para comprobar el funcionamiento del bloque de rotación, se introduce esta señal en la DAW.

Con tal de examinar el resultado de la decodificación de manera independiente, se exportan las señales tanto rotada como sin rotar también desde el sistema de Python. En el caso de la biaurolización, se pueden introducir la señal que va a los altavoces sin interactuar con el Plug – In. Con esto, lo que se pretende es cerciorarnos de que cada uno de los bloques funciona adecuadamente de manera independiente. Por supuesto, se comprobará la interacción entre los mismos partiendo de la señal Ambisonics creada por el generador de señales.

### 5.2. IMPLEMENTACIÓN EN REAPER

Durante los primeros pasos de la elaboración del Plug – In se hizo uso del programa *Reaper* para comprobar su funcionamiento y resultados, de forma que pudiéramos visualizar correctamente el comportamiento del programa.

En primer lugar, es necesario importar el VST3 a la estación. Para ello, hay que añadir la ubicación de nuestro Plug – In en las preferencias de configuración de *Reaper* mediante la selección: *Options -> Reaper Preferences -> VST Plug – Ins settings*. Una vez actualizados, añadimos el programa insertándolo desde *Track -> Insert virtual instrumento on new track*, donde lo podremos seleccionar en la pestaña VST3. En la siguiente imagen se muestran los instrumentos de este formato existentes en el equipo. El desarrollado para este trabajo se denomina *TfmProject*, donde se observa que es compatible con los 16 canales deseados, mientras que el resto no los especifica al tratarse de entradas y salidas estéreo.

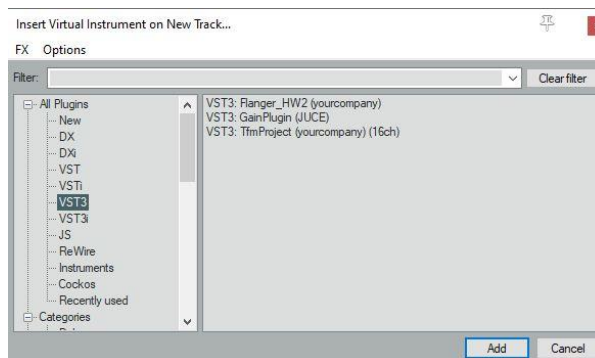


Figura 49. Inserción de instrumentos virtuales en Reaper.

Una vez seleccionado, importamos la señal de audio. Gracias a la potente herramienta de visualización de forma de onda de Reaper, podemos observar la totalidad de los canales, así como la naturaleza de cada uno de ellos, es decir, si existe señal en ese canal, cómo es la senoide, etc. Para el siguiente ejemplo se han utilizado los valores  $(\theta, \phi) = (90^\circ, 0^\circ)$  donde la fuente está situada a la derecha del oyente. Fijándonos en los valores de la matriz generada mediante Python, vemos que sólo existe información de la señal en los canales 0, 3, 6, 8, 13 y 15. Para el resto, los valores son 0 (Python, en ocasiones, no asocia el 0 a resultados infinitesimales (potencias de 10 del orden de más de -10 de exponente). Se ha elaborado unas líneas de código para que, en esas situaciones, los valores sean 0. A continuación se muestran las primeras 18 muestras para cada uno de los 16 canales<sup>21</sup>.

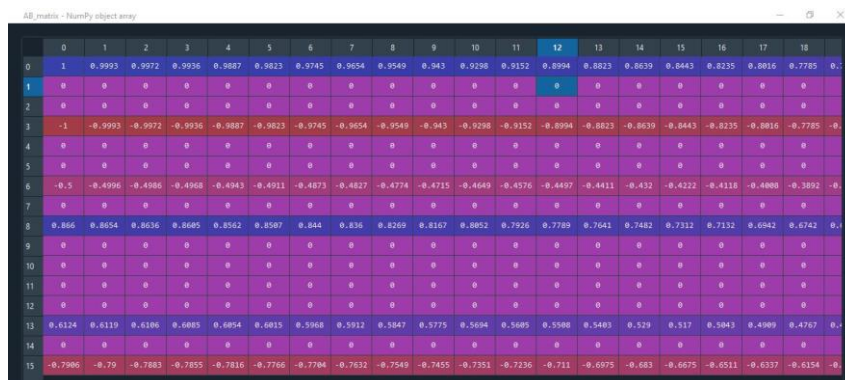


Figura 50. Matriz de señal Ambisonics para  $(\theta, \phi) = (90^\circ, 0^\circ)$ .

De manera correspondiente se visualizan los canales en *Reaper*.



Figura 51. Visualizador de onda de la señal Ambisonics en *Reaper*.

## ROTACIÓN DE CANALES

Cuando ejecutamos la rotación en los canales, se percibe un cambio en la experiencia auditiva (que analizaremos en los resultados) pero no en la visualización de la forma de onda. Para ver la forma de onda de la señal rotada, se exporta el archivo mediante la opción *Render*, y se añade al proyecto. Observamos que la información de los canales ha cambiado, apareciendo ahora señales en aquellos en los que previamente no había o viceversa.

<sup>21</sup> Los canales son cada una de las filas de la matriz.



**Figura 52.** Forma de onda de una señal Ambisonics rotada.

#### DECODIFICACIÓN A 7.1.4

Como se ha explicado antes, el usuario puede decidir realizar la decodificación de la señal de entrada o de su versión rotada. Para ambos casos, hay una variación en la señal de audio y, una vez exportadas, observamos que la decodificación presenta una configuración de canales distinta para las distintas elecciones, como se esperaba. En la primera figura a continuación, se adjunta la forma de onda de la señal original Ambisonics decodificada y en la segunda se puede observar el resultado de la señal rotada decodificada donde, curiosamente, se perciben diversas amplitudes en función del canal. Esto favorecerá el efecto envolvente cuando se decodifique a binaural.

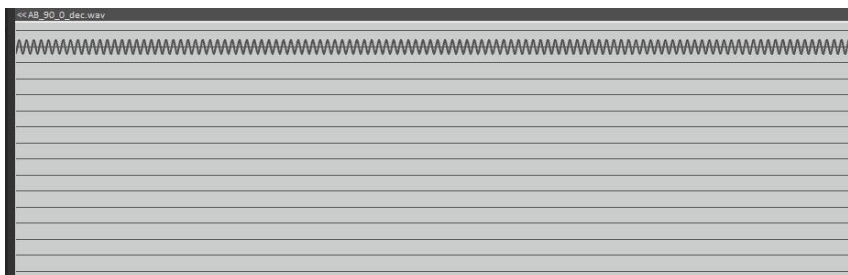


**Figura 53.** Forma de onda de una señal Ambisonics decodificada a 7.1.4 en Reaper.



**Figura 54.** Forma de onda de una señal Ambisonics rotada y decodificada a 7.1.4 en Reaper.

#### DECODIFICACIÓN BIAURAL

Esta configuración no permite su computación directamente desde la señal de entrada, ya que para crear una señal binaural se requiere una simulación de altavoces situados en distintos puntos del espacio. Por lo tanto, el usuario puede binauralizar una señal rotada y decodificada o simplemente decodificada, si bien se aconseja computar la rotación de cara al oyente para una mejor experiencia auditiva. En el caso de la señal de la Figura 54, donde la

señal se ha rotado  $45^\circ$  sobre el eje  $z$  ( $\gamma$ ), su acimut se mueve  $45^\circ$  hacia la izquierda. También se traslada  $30^\circ$  sobre el eje  $y$  ( $\beta$ ) y  $90^\circ$  sobre el nuevo eje  $z$  ( $\alpha$ ). Es de esperar que la mayor parte de la señal se concentre en el hemisferio derecho del oyente, pero que cuente con cierta información en la sección izquierda. Al exportar la señal y visualizar su forma de onda, comprobamos que la derecha, correspondiente al canal 1 y situada en la parte inferior, contiene una amplitud mayor (y por lo tanto más presencia sonora).

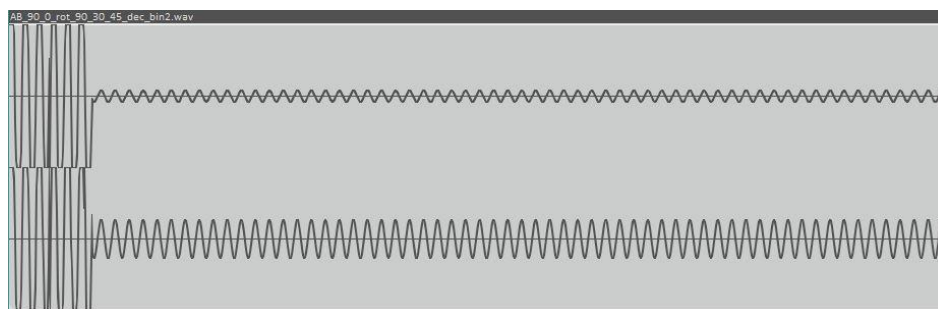


**Figura 55.** Forma de onda de una señal binaural.

Desgraciadamente, observamos que aparece una sección no uniforme y saturada en las primeras muestras de la señal. Este fenómeno, que recibe el nombre de transitorio y se percibe como un chasquido, aparece cuando existe un cambio brusco en la amplitud de la señal, por lo que el filtrado no tiene las muestras suficientes para llevar a cabo el proceso. Analizamos que este comportamiento ocurre durante las mil primeras muestras (véase la Figura 56), lo cual tiene sentido ya que este es el tamaño del filtro.

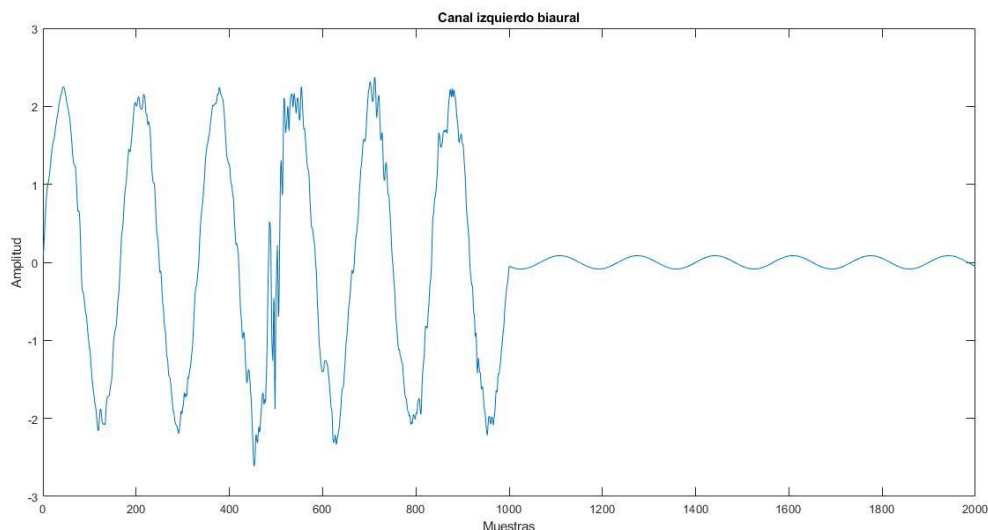


**Figura 56.** Gráfico de las primeras dos mil muestras del canal izquierdo de una señal binaural con transitorio.

Para evitar este efecto, se recomienda generar una señal creciente en amplitud. Vemos que se consigue evitar el efecto transitorio cuando se trabaja con una señal que al inicio tiene muestras cercanas a cero y van ganando cada vez más amplitud de manera que se acerca a la muestra número 1000.

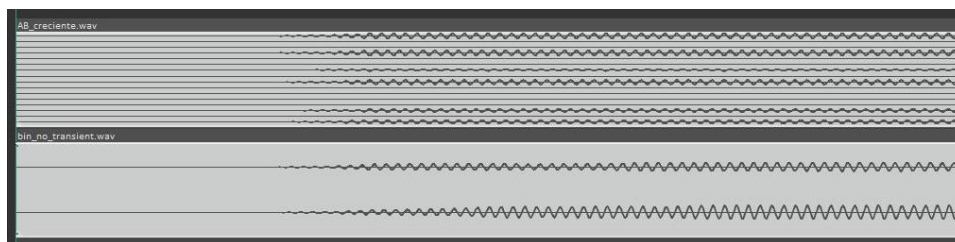


**Figura 57.** Formas de onda de una señal creciente Ambisonics de tercer orden y su biauralización.

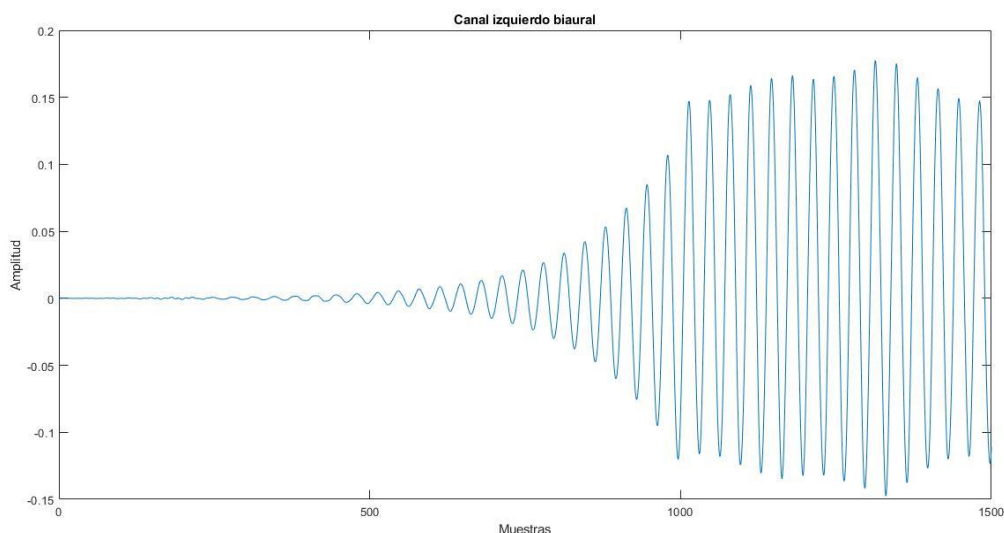


**Figura 58.** Gráfico de las primeras mil quinientas muestras del canal izquierdo de una señal biaural sin transitorio.

Antes de concluir, nos gustaría indicar que esta *DAW* ha sido de gran ayuda a lo largo de la primera etapa de elaboración del Trabajo. Sin embargo, se tuvo que cambiar el Estación de Trabajo, ya que en un momento dado y sin haber realizado ningún cambio tanto en el Plug – In como en el ordenador, *Reaper* no reconocía correctamente el *VST3* y el programa se colapsaba cada vez que se intentaba ejecutar nuestro sistema. Esto tuvo lugar en el paso previo a la comprobación del bloque de biauralización, el más importante del Trabajo. Por ello, se tuvo que recurrir a la utilización de una *DAW* más potente: *Adobe Audition*.

Finalizamos la sección de implementación en *Reaper* destacando sus ventajas e inconvenientes:

- Ventajas:
  - Período de prueba largo.
  - Interfaz cómoda e intuitiva.
  - Compatibilidad con *VST3*.
  - Visualizador de forma de onda multicanal.
  - Fácil sistema de exportación de archivos.
- Inconvenientes:
  - Interrupción del programa de forma inesperada.
  - No permite ver cambios en la forma de onda.
  - Baja resolución de visualización.

### 5.3.IMPLEMENTACIÓN EN ADOBE AUDITION

Con tal de comprobar el último elemento del desarrollo de nuestro Plug – In tras el problema con el que nos enfrentamos con la Estación de Audio *Reaper*, se ha decidido utilizar *Adobe Audition*. A pesar de haber ya verificado los bloques anteriores a la decodificación binaural, se presenta en esta sección la confirmación del funcionamiento del programa completo.

Antes de comenzar, se considera importante comentar que la importación de los VST3 en *Audition* no ha sido tan sencilla como ocurrió con la anterior DAW. Tras probar de diferentes maneras, se concluyó que los archivos de dicha extensión deben estar en la carpeta C:\Program Files\Common Files\VST3, por lo que se deben mover manualmente desde la carpeta de compilación.

A continuación, se introduce el mismo archivo que hemos utilizado en *Reaper*. Observamos que el visualizador de forma de onda es más completo: nos permite ver cada uno de los canales con claridad, ampliar y reducir la base de tiempos, aumentar o reducir el nivel de la señal, añadir efectos....

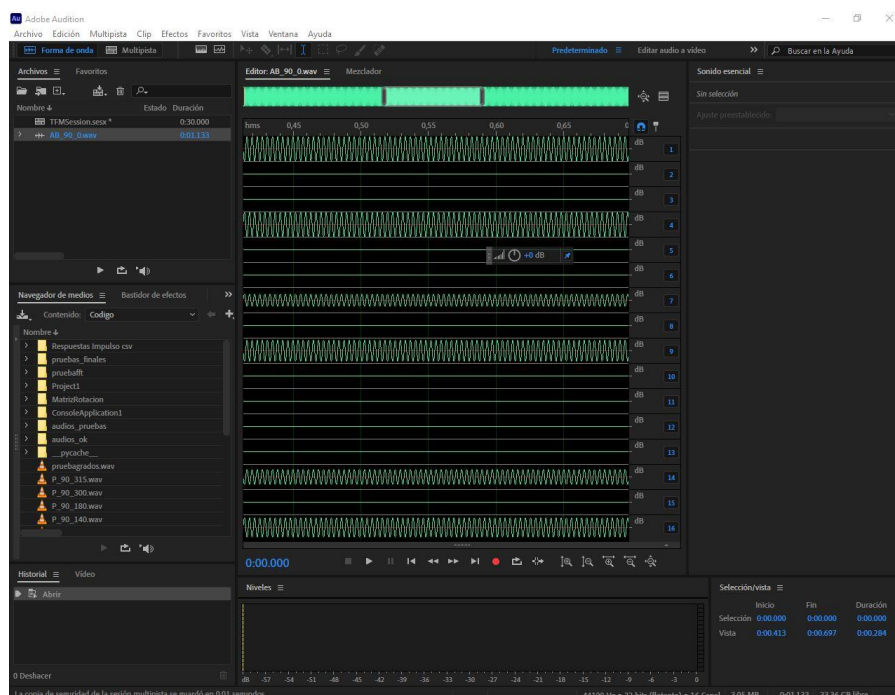


**Figura 59.** Captura de pantalla de visualizador de onda de *Adobe Audition*.

La gran ventaja que hemos comprobado que esta DAW tiene sobre *Reaper* es que, cuando aplicamos el efecto deseado desde nuestro Plug – In, se genera automáticamente el cambio en la forma de onda de la señal resultante. Lo vemos en la implementación de los bloques:

#### ROTADOR DE CANALES

Con tal de comprobar el comportamiento de *Audition*, insertamos la misma señal y la rotamos con los ángulos de rotación de Euler utilizados anteriormente:  $(\alpha, \beta, \gamma) = (90^\circ, 30^\circ, 45^\circ)$ . Cuando activamos el botón “Rotación” y reproducimos la señal, percibimos el cambio en la respuesta sonora. Además, una vez pulsamos el botón “Aplicar”, notamos que la forma de onda de los canales ha cambiado, y ahora se corresponde a la de la señal rotada, lo cual lo podemos comprobar con los resultados obtenidos y reflejados en la Figura 52,

Figura 64 y Figura 65. Podemos exportar la señal de audio desde *Archivo -> Exportar archivo*, donde se pueden elegir distintos formatos, velocidad de muestreo, extensión de la muestra en bits...

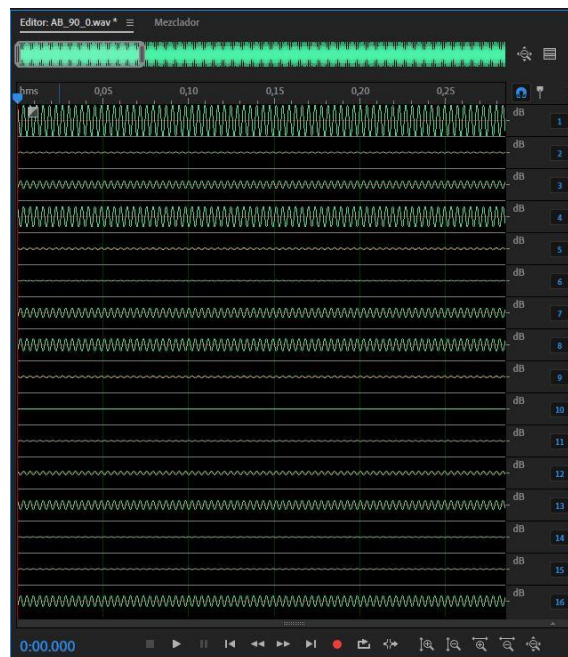


**Figura 60.** Visualizador de forma de onda de una señal Ambisonics rotada en *Audition*.

#### DECODIFICADOR A 7.1.4

Ejecutamos la decodificación a las posiciones virtuales de los altavoces tanto para la señal rotada como para la señal original Ambisonics y observamos que el cambio sonoro y visual se corresponde con lo obtenido mediante *Reaper*, así como con los valores numéricos de las matrices generadas sintéticamente (mediante *Python*).

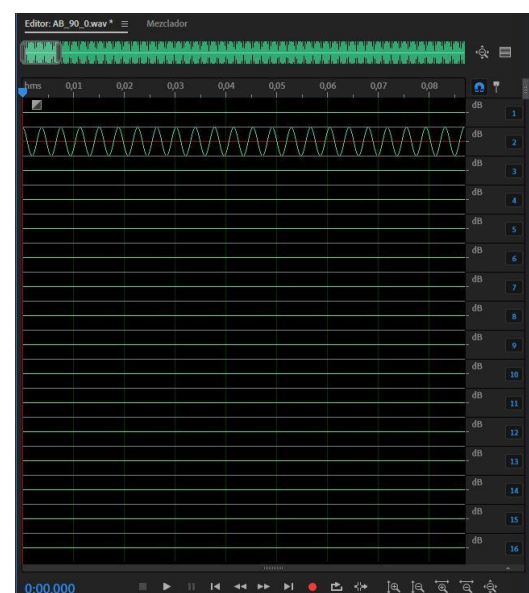
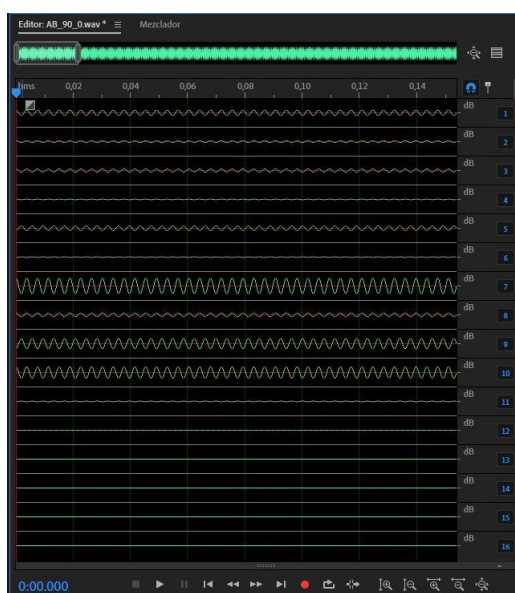


**Figura 61.** Formas de onda de una señal rotada y decodificada a 7.1.4 (izquierda) y decodificada a 7.1.4 (derecha) en *Audition*.

## DECODIFICADOR BIAURAL

Finalizamos con la transformación al sistema binaural. En este caso, la experiencia auditiva mediante la reproducción previa a la aplicación del efecto no es satisfactoria, ya que, como hemos explicado, el procesado se lleva a cabo a través de bloques, efecto que puede percibirse manifestado mediante el sonido interrumpido en lugar de continuado. Sin embargo, al aplicar el efecto y reproducirlo, se comprueba que la experiencia binaural es satisfactoria. En las siguientes imágenes observamos el resultado con el transitorio, explicado previamente, y la solución con el efecto eliminado gracias a la utilización de una señal con amplitud creciente. Ya que hemos configurado el Plug – In de manera que la salida ofrezca un máximo de 16 canales, los binaurales se encuentran en las posiciones 1 y 2, oído izquierdo y derecho, respectivamente. Para exportar el archivo correspondiente, podemos seleccionar la salida de canales de tipo estéreo.

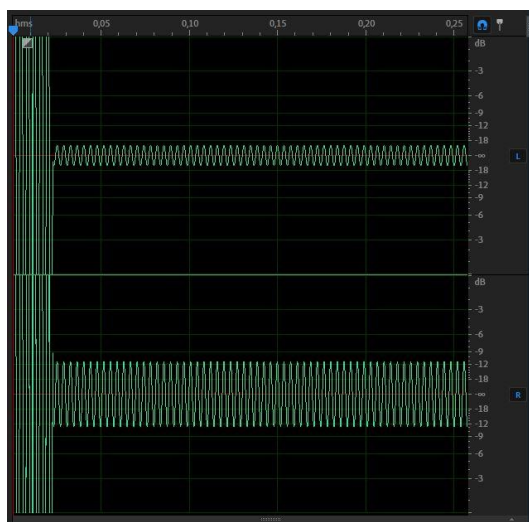


**Figura 62.** Señal binaural con transitorio.

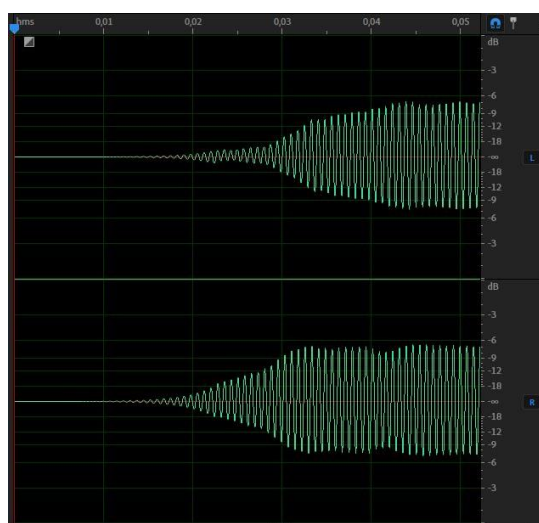


**Figura 63.** Señal binaural sin transitorio.

Gracias a este software, hemos podido realizar todas las comprobaciones necesarias para asegurarnos de que el Plug – In desarrollado para ese Trabajo de Fin de Máster funciona correctamente y como se esperaba.

Finalizamos esta sección con los aspectos positivos y negativos de *Adobe Audition* experimentados durante el desarrollo de la implementación del Plug – In.

- Ventajas:
  - Gran capacidad de procesamiento.
  - Visualización de forma de onda profesional.
  - Cambios visibles sin necesidad de exportar la señal.
  - Múltiples formatos de generación de archivos.
  - Compatibilidad con VST3.
- Inconvenientes:
  - Período de prueba muy corto (una semana).
  - Software de pago.
  - Dificultad de exportación de Plug – Ins.

## 6.RESULTADOS

### 6.1.RESULTADOS NUMÉRICOS

#### ROTACIÓN

Se comparan los resultados obtenidos mediante el sistema elaborado en Python y el Plug – In mediante las matrices obtenidas a través de ambos procesos. Se elige la señal AB\_90\_0.wav y se computa la rotación para valores de  $(\alpha, \beta, \gamma) = (90, 30, 45)$ . Tras exportar el archivo generado tanto en *Reaper* como en *Audition*, se crea una variable en *Spyder* para almacenar los valores obtenidos. Para comparar que los resultados son similares, se realiza la división muestra a muestra de cada canal y debe obtenerse una matriz donde todas sus posiciones estén ocupadas por unos. Mediante la herramienta de visualización de Matlab, vemos que el resultado de la división es la unidad para las cincuenta mil muestras. Se incluye también una captura de las primeras 15 muestras para cada uno de los 16 canales, donde comprobamos que los valores son los mismos.

AB\_rot\_reaper - NumPy object array (read only)

|    | 0       | 1       | 2       | 3       | 4       | 5       | 6       | 7       | 8       | 9       | 10      | 11      | 12      | 13      | 14      | 15      | 16      |
|----|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| 0  | 1       | 0.999   | 0.997   | 0.994   | 0.989   | 0.982   | 0.975   | 0.965   | 0.955   | 0.943   | 0.93    | 0.915   | 0.899   | 0.882   | 0.864   | 0.844   | 0.824   |
| 1  | -0.0474 | -0.0473 | -0.0472 | -0.0471 | -0.0468 | -0.0465 | -0.0462 | -0.0457 | -0.0452 | -0.0447 | -0.0444 | -0.0434 | -0.0426 | -0.0418 | -0.0409 | -0.04   | -0.039  |
| 2  | -0.25   | -0.25   | -0.249  | -0.248  | -0.247  | -0.246  | -0.244  | -0.241  | -0.239  | -0.236  | -0.232  | -0.229  | -0.225  | -0.221  | -0.216  | -0.211  | -0.206  |
| 3  | 0.66    | 0.659   | 0.658   | 0.656   | 0.652   | 0.648   | 0.643   | 0.637   | 0.63    | 0.622   | 0.613   | 0.604   | 0.593   | 0.582   | 0.57    | 0.557   | 0.543   |
| 4  | 0.0765  | 0.0765  | 0.0763  | 0.0761  | 0.0757  | 0.0752  | 0.0746  | 0.0739  | 0.0731  | 0.0722  | 0.0712  | 0.0701  | 0.0688  | 0.0675  | 0.0661  | 0.0646  | 0.063   |
| 5  | 0.029   | 0.029   | 0.0289  | 0.0288  | 0.0287  | 0.0285  | 0.0283  | 0.028   | 0.0277  | 0.0274  | 0.027   | 0.0265  | 0.0261  | 0.0256  | 0.0251  | 0.0245  | 0.0239  |
| 6  | -0.312  | -0.312  | -0.312  | -0.311  | -0.309  | -0.307  | -0.305  | -0.302  | -0.298  | -0.295  | -0.291  | -0.286  | -0.281  | -0.276  | -0.27   | -0.264  | -0.257  |
| 7  | -0.404  | -0.404  | -0.403  | -0.401  | -0.399  | -0.397  | -0.394  | -0.39   | -0.386  | -0.381  | -0.376  | -0.37   | -0.363  | -0.356  | -0.349  | -0.341  | -0.333  |
| 8  | 0.0765  | 0.0765  | 0.0763  | 0.0761  | 0.0757  | 0.0752  | 0.0746  | 0.0739  | 0.0731  | 0.0722  | 0.0712  | 0.0701  | 0.0688  | 0.0675  | 0.0661  | 0.0646  | 0.063   |
| 9  | -0.0003 | -0.0003 | -0.0003 | -0.0003 | -0.0003 | -0.0003 | -0.0003 | -0.0003 | -0.0003 | -0.0003 | -0.0003 | -0.0003 | -0.0003 | -0.0003 | -0.0003 | -0.0003 | -0.0003 |
| 10 | 0.0428  | 0.0428  | 0.0427  | 0.0425  | 0.0423  | 0.042   | 0.0417  | 0.0413  | 0.0409  | 0.0404  | 0.0398  | 0.0392  | 0.0385  | 0.0378  | 0.037   | 0.0363  | 0.0352  |
| 11 | 0.128   | 0.128   | 0.128   | 0.127   | 0.127   | 0.126   | 0.125   | 0.124   | 0.122   | 0.121   | 0.119   | 0.117   | 0.115   | 0.113   | 0.111   | 0.108   | 0.105   |
| 12 | 0.297   | 0.297   | 0.296   | 0.295   | 0.294   | 0.292   | 0.289   | 0.287   | 0.283   | 0.28    | 0.276   | 0.272   | 0.267   | 0.262   | 0.256   | 0.251   | 0.244   |
| 13 | -0.0343 | -0.0343 | -0.0342 | -0.0341 | -0.0339 | -0.0337 | -0.0334 | -0.0331 | -0.0328 | -0.0324 | -0.0319 | -0.0314 | -0.0309 | -0.0303 | -0.0296 | -0.029  | -0.0283 |
| 14 | 0.0428  | 0.0428  | 0.0427  | 0.0425  | 0.0423  | 0.042   | 0.0417  | 0.0413  | 0.0409  | 0.0404  | 0.0398  | 0.0392  | 0.0385  | 0.0378  | 0.037   | 0.0363  | 0.0352  |
| 15 | -0.356  | -0.356  | -0.355  | -0.354  | -0.352  | -0.35   | -0.347  | -0.344  | -0.34   | -0.336  | -0.331  | -0.326  | -0.32   | -0.314  | -0.308  | -0.301  | -0.293  |

Figura 64. Resultado numérico de una señal Ambisonics rotada mediante el Plug – In.

B\_rot - NumPy object array

|    | 0       | 1       | 2       | 3       | 4       | 5       | 6       | 7       | 8       | 9       | 10      | 11      | 12      | 13      | 14      | 15      |
|----|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| 0  | 1       | 0.999   | 0.997   | 0.994   | 0.989   | 0.982   | 0.975   | 0.965   | 0.955   | 0.943   | 0.93    | 0.915   | 0.899   | 0.882   | 0.864   | 0.844   |
| 1  | -0.0474 | -0.0473 | -0.0472 | -0.0471 | -0.0468 | -0.0465 | -0.0462 | -0.0457 | -0.0452 | -0.0447 | -0.0444 | -0.0434 | -0.0426 | -0.0418 | -0.0409 | -0.04   |
| 2  | -0.25   | -0.25   | -0.249  | -0.248  | -0.247  | -0.246  | -0.244  | -0.241  | -0.239  | -0.236  | -0.232  | -0.229  | -0.225  | -0.221  | -0.216  | -0.211  |
| 3  | 0.66    | 0.659   | 0.658   | 0.656   | 0.652   | 0.648   | 0.643   | 0.637   | 0.63    | 0.622   | 0.613   | 0.604   | 0.593   | 0.582   | 0.57    | 0.557   |
| 4  | 0.0765  | 0.0765  | 0.0763  | 0.0761  | 0.0757  | 0.0752  | 0.0746  | 0.0739  | 0.0731  | 0.0722  | 0.0712  | 0.0701  | 0.0688  | 0.0675  | 0.0661  | 0.0646  |
| 5  | 0.029   | 0.029   | 0.0289  | 0.0288  | 0.0287  | 0.0285  | 0.0283  | 0.028   | 0.0277  | 0.0274  | 0.027   | 0.0265  | 0.0261  | 0.0256  | 0.0251  | 0.0245  |
| 6  | -0.312  | -0.312  | -0.312  | -0.311  | -0.309  | -0.307  | -0.305  | -0.302  | -0.298  | -0.295  | -0.291  | -0.286  | -0.281  | -0.276  | -0.27   | -0.264  |
| 7  | -0.404  | -0.404  | -0.403  | -0.401  | -0.399  | -0.397  | -0.394  | -0.39   | -0.386  | -0.381  | -0.376  | -0.37   | -0.363  | -0.356  | -0.349  | -0.341  |
| 8  | 0.0765  | 0.0765  | 0.0763  | 0.0761  | 0.0757  | 0.0752  | 0.0746  | 0.0739  | 0.0731  | 0.0722  | 0.0712  | 0.0701  | 0.0688  | 0.0675  | 0.0661  | 0.0646  |
| 9  | -0.0003 | -0.0003 | -0.0003 | -0.0003 | -0.0003 | -0.0003 | -0.0003 | -0.0003 | -0.0003 | -0.0003 | -0.0003 | -0.0003 | -0.0003 | -0.0003 | -0.0003 | -0.0003 |
| 10 | 0.0428  | 0.0428  | 0.0427  | 0.0425  | 0.0423  | 0.042   | 0.0417  | 0.0413  | 0.0409  | 0.0404  | 0.0398  | 0.0392  | 0.0385  | 0.0378  | 0.037   | 0.0363  |
| 11 | 0.128   | 0.128   | 0.128   | 0.127   | 0.127   | 0.126   | 0.125   | 0.124   | 0.122   | 0.121   | 0.119   | 0.117   | 0.115   | 0.113   | 0.111   | 0.108   |
| 12 | 0.297   | 0.297   | 0.296   | 0.295   | 0.294   | 0.292   | 0.289   | 0.287   | 0.283   | 0.28    | 0.276   | 0.272   | 0.267   | 0.262   | 0.256   | 0.251   |
| 13 | -0.0343 | -0.0343 | -0.0342 | -0.0341 | -0.0339 | -0.0337 | -0.0334 | -0.0331 | -0.0328 | -0.0324 | -0.0319 | -0.0314 | -0.0309 | -0.0303 | -0.0296 | -0.029  |
| 14 | 0.0428  | 0.0428  | 0.0427  | 0.0425  | 0.0423  | 0.042   | 0.0417  | 0.0413  | 0.0409  | 0.0404  | 0.0398  | 0.0392  | 0.0385  | 0.0378  | 0.037   | 0.0363  |
| 15 | -0.356  | -0.356  | -0.355  | -0.354  | -0.352  | -0.35   | -0.347  | -0.344  | -0.34   | -0.336  | -0.331  | -0.326  | -0.32   | -0.314  | -0.308  | -0.301  |

Figura 65. Resultado numérico de una señal Ambisonics rotada mediante el sistema Python.

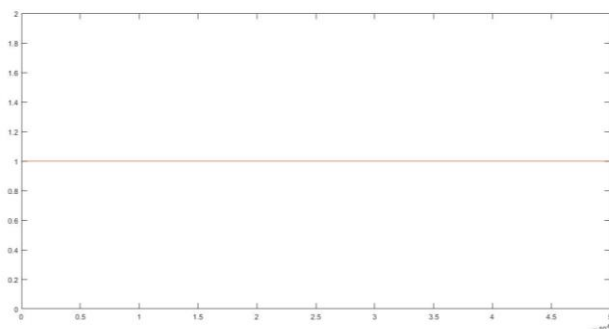


**Figura 66.** Gráfica de comparación de las rotaciones mediante la división de sus valores.

#### DECODIFICACIÓN 7.1.4

Llevamos a cabo el mismo procedimiento que en el caso anterior y comprobamos que los resultados son los esperados: mismos valores iniciales en las matrices y variable de comparación correcta.

En las siguientes figuras, se contrastan los resultados para la decodificación de la señal Ambisonics original, sin rotar. Observamos que sólo hay componente de la señal en el canal 1, que corresponde al posicionado en los ángulos  $(\theta, \phi) = (90^\circ, 0^\circ)$ , es decir, a la derecha del oyente. Comprobaremos en la sección experimental que, una vez transformado al sistema binaural, se escucha desde dicha dirección.

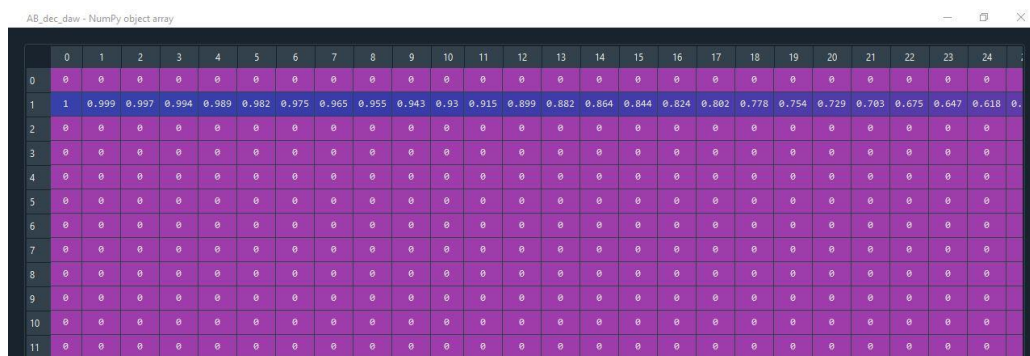


**Figura 67.** Resultado numérico de una señal 7.1.4 a partir de una Ambisonics obtenida mediante el Plug-In.

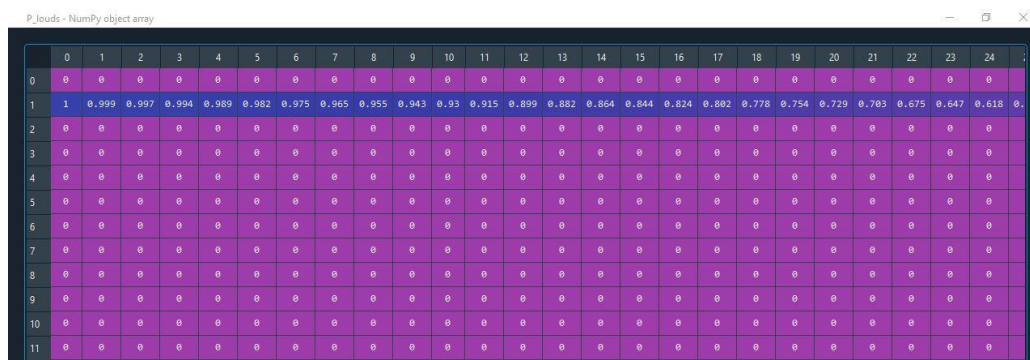


**Figura 68.** Resultado numérico de una señal 7.1.4 a partir de una Ambisonics obtenida mediante el sistema Python.

A continuación, observamos el resultado para la señal Ambisonics decodificada a 7.1.4 con previa rotación mediante los mismos ángulos de Euler.

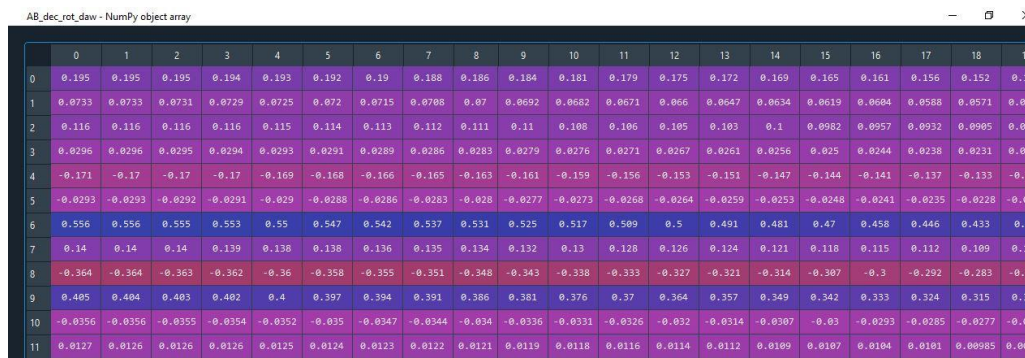


**Figura 69.** Resultado numérico de una señal 7.1.4 a partir de una Ambisonics rotada mediante el Plug – In.

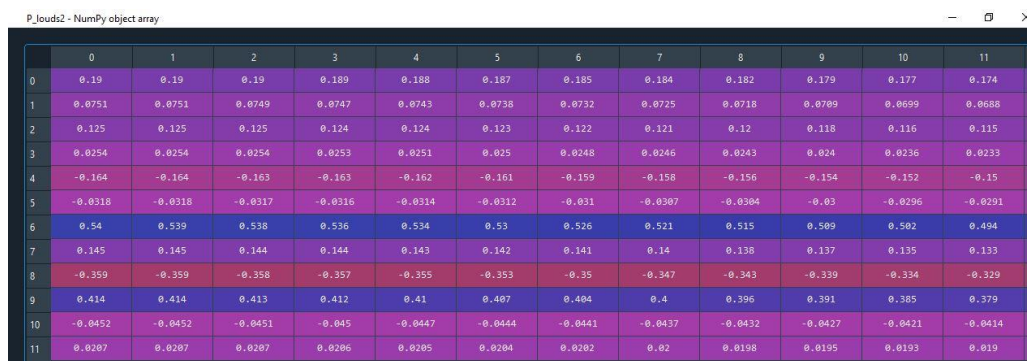


**Figura 70.** Resultado numérico de una señal 7.1.4 a partir de una Ambisonics rotada mediante el sistema Python.

## DECODIFICACIÓN BIAURAL

Los resultados de la codificación binaural se ilustrarán en la siguiente sección. Esto se debe a que el desarrollo mediante Python no incluye la señal completa, sino una versión reducida de la misma para comprobar el correcto funcionamiento del filtrado. Por otro lado, la solución binaural se puede comprobar de manera experimental.

## 6.2.RESULTADOS PERCEPTUALES

A continuación, se demostrará de manera experimental el verdadero efecto de nuestro sistema binaural. Para ello, se ha utilizado para la reproducción la misma persona que para la grabación, ya que las *Head Related Transfer Functions* son dependientes de la cabeza humana.

### 6.2.1. DESARROLLO DE LA PRUEBA

Para elaborar las pruebas, se han preparado distintos tipos de archivos de audio: un pulso de frecuencia constante, una grabación de voz y una muestra musical. Las muestras se han transformado a los 16 canales de Ambisonics mediante nuestro “Generador de señales”. Lo que se pretende comprobar es el funcionamiento del sistema en un escenario trivial y en condiciones más complejas. Para completar el ensayo, se han probado tres configuraciones de altavoces distintas y dos métodos de binauralización (sin rotación y con ella).

El primer sistema de altavoces consiste en la disposición en un sistema de coordenadas cardinales, donde las posiciones están determinadas por los ángulos  $(90^\circ, 0^\circ)$ ,  $(90^\circ, 90^\circ)$ ,  $(90^\circ, 180^\circ)$ ,  $(90^\circ, 270^\circ)$ . El propósito de usar esta configuración no estandarizada ni recomendada por la ITU es la comprobación básica del sistema, ya que, si la binauralización funciona correctamente, percibiremos el sonido proveniente de la derecha e izquierda, así como de la parte delantera. Como ya se ha discutido en esta memoria, la parte trasera es más complicada de percibir debido a la forma de la oreja, por lo que no se espera un resultado excelente.

La segunda estructura será un sistema cuadrafónico situado en las posiciones recomendadas por la ITU, es decir, en las cuatro esquinas de un cuadrado imaginario formando entre sí  $90^\circ$ . Se utiliza por ser un sistema estandarizado y muy utilizado, y porque ayuda a seguir comprobando algunas direccionalidades básicas antes de pasar al siguiente nivel.

La última configuración y la más importante es la multicanal 7.1.4 recomendada por la ITU, objetivo de nuestro Proyecto. Además, será la responsable de ofrecer un sistema binaural más preciso y realista, por lo que su correcto funcionamiento determinará el grado de éxito del desarrollo del programa tanto en Python como en C++.

Por otro lado, se ha llevado a cabo la decodificación binaural tanto con señales originales como con las rotadas. El motivo de no rotar las señales es verificar que las Ambisonics generadas se decodifican a 7.1.4 correctamente y que su presentación binaural es la esperada. Es decir, si hemos elegido una posición sobre la esfera unidad de  $(\theta, \phi) = (90^\circ, 180^\circ)$  y hemos generado los armónicos esféricos correspondientes, el diagrama de directividad será mayor para dicha dirección, por lo que el sonido debe de percibirse mayormente desde la izquierda. En cuanto a la transformación espacial de los altavoces, se pretende situarlos de cara al oyente para evitar los efectos negativos de los canales posteriores.

Los resultados se mostrarán a continuación mediante matrices de confusión y también mediante figuras con las posiciones originales y las percibidas.

### 6.2.2. VISUALIZACIÓN DE RESULTADOS

Una matriz de confusión es una herramienta que se utiliza normalmente en estadística para ofrecer una visualización del rendimiento de un programa de aprendizaje supervisado. Aunque nuestro sistema no tenga dichas características, ha decidido utilizar esta matriz gracias a su sencillez y claridad a la hora de representar resultados.

Los valores situados a la izquierda de la matriz (filas) se corresponden con la posición desde la que se espera recibir el sonido, mientras que los posicionados en la parte superior (columnas) son las verdaderas posiciones. El valor de cada casilla indica el porcentaje de precisión estimado para cada una de las posiciones. Para ofrecer una visualización clara e ilustrativa, se adjunta a continuación una matriz de confusión ya obtenida en la cual se explica el código de colores y los valores de cada casilla. Utilizamos la siguiente tabla (correspondiente a la Tabla 10) porque incluye diversos valores, por lo que es muy ilustrativa.

|               |                         | Posición percibida |                        |                         |                         |                         |
|---------------|-------------------------|--------------------|------------------------|-------------------------|-------------------------|-------------------------|
|               |                         | $(\theta, \phi)$   | $(90^\circ, 45^\circ)$ | $(90^\circ, 135^\circ)$ | $(90^\circ, 225^\circ)$ | $(90^\circ, 315^\circ)$ |
| Posición real | $(90^\circ, 45^\circ)$  | 80%                | 20%                    | 0%                      | 0%                      |                         |
|               | $(90^\circ, 135^\circ)$ | 10%                | 90%                    | 0%                      | 0%                      |                         |
|               | $(90^\circ, 225^\circ)$ | 0%                 | 0%                     | 60%                     | 40%                     |                         |
|               | $(90^\circ, 315^\circ)$ | 0%                 | 0%                     | 50%                     | 50%                     |                         |

**Tabla 6.** Matriz de confusión obtenida para los resultados de una medición de altavoces situados en un sistema cuadrafónico.

- El color verde se utiliza en aquellas posiciones en las que el grado de coincidencia sea entre 80% y 100%, es decir, el oyente percibe de manera correcta el sonido desde la posición esperada.
- El color azul se utiliza para un grado de coincidencia medio, entre el 50% y 80%.
- Se ha considerado un nivel de coincidencia bajo a los porcentajes situados entre los valores 0 y 50, para el cual se ha utilizado el color violeta. Véase que este valor no siempre indica una mala observación. Por ejemplo, en la segunda fila, vemos los valores de 10% y 0% marcados con violeta. La posición real de la fuente está en  $(90^\circ, 135^\circ)$ , y no se recibe información sonora desde  $(90^\circ, 225^\circ)$  y  $(90^\circ, 315^\circ)$ , por lo tanto, el resultado es correcto. El 10% marcado en  $(90^\circ, 45^\circ)$  nos indica que cierta información viene desde el lado frontal derecho, en lugar del izquierdo, pero se percibe de manera muy ligera.

En cuanto a las figuras, los altavoces representan la posición de la fuente virtual y los círculos con una onda en su interior indican desde dónde, aproximadamente, se percibe el sonido generado en el sistema binaural.

## DECODIFICACIÓN DE SEÑALES AMBISONICS SIN TRASLACIÓN ESPACIAL

Se realizarán los experimentos sonoros a dos sujetos distintos. El primero será la persona para la cual se han elaborado las *HRTF* y el segundo utilizará las mismas respuestas al impulso para observar los resultados.

### Pruebas sobre el sujeto 1

- Comenzamos con los resultados de las posiciones en puntos cardinales. Observamos mediante la matriz de confusión de la Tabla 7 que lo percibido por el oyente se corresponde al 100% en dos posiciones claramente diferenciadas (a la derecha y a la izquierda del oyente) y casi perfectamente cuando el sonido viene de la parte delantera. Cuando la fuente se encuentra en la parte posterior, la percepción no es tan clara, sino que se confunde con la parte delantera. En la Figura 71 se ofrece una visualización más clara de las posiciones desde donde se perciben las señales sonoras.

- ♪ Muestra de audio: senoide de frecuencia y amplitud constantes, duración 1 segundo.

|               |                         | Posición percibida |                       |                        |                         |                         |
|---------------|-------------------------|--------------------|-----------------------|------------------------|-------------------------|-------------------------|
|               |                         | $(\theta, \phi)$   | $(90^\circ, 0^\circ)$ | $(90^\circ, 90^\circ)$ | $(90^\circ, 180^\circ)$ | $(90^\circ, 270^\circ)$ |
| Posición real | $(90^\circ, 0^\circ)$   | 100%               | 0%                    | 0%                     | 0%                      |                         |
|               | $(90^\circ, 90^\circ)$  | 10%                | 90%                   | 0%                     | 0%                      |                         |
|               | $(90^\circ, 180^\circ)$ | 0%                 | 0%                    | 100%                   | 0%                      |                         |
|               | $(90^\circ, 270^\circ)$ | 0%                 | 30%                   | 0%                     | 70%                     |                         |

**Tabla 7.** Matriz de confusión para un sistema cuadrafónico con posiciones cardinales.

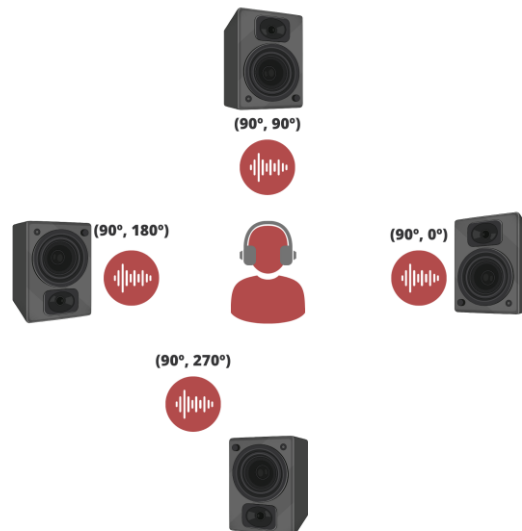


**Figura 71.** Ilustración de la percepción de un sistema cuadrafónico con posiciones cardinales.

- ♪ Muestra de audio: grabación de voz de duración 9 segundos. Percibimos que el procesamiento del sonido no ha distorsionado en ningún caso la muestra original. En las siguientes figuras observamos la forma de onda de la señal de partida y su versión baural para una posición en frente del oyente.

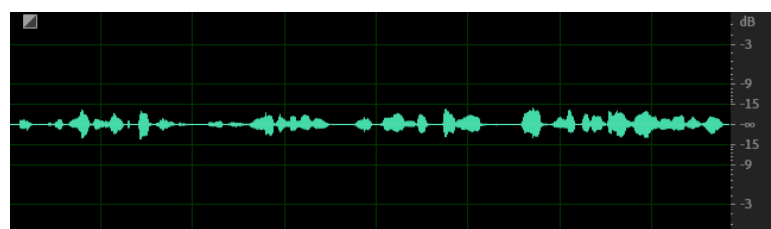


**Figura 72.** Forma de onda del clip de voz original.

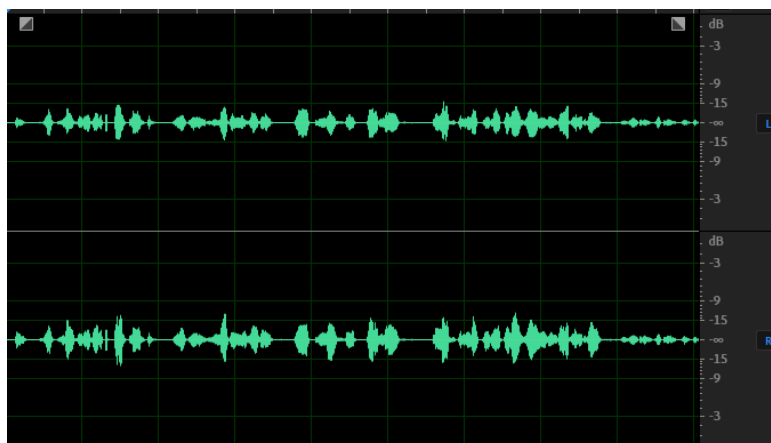


**Figura 73.** Formas de onda de una versión binaural del clip de voz.

|               |                         | Posición percibida |                       |                        |                         |                         |
|---------------|-------------------------|--------------------|-----------------------|------------------------|-------------------------|-------------------------|
|               |                         | $(\theta, \phi)$   | $(90^\circ, 0^\circ)$ | $(90^\circ, 90^\circ)$ | $(90^\circ, 180^\circ)$ | $(90^\circ, 270^\circ)$ |
| Posición real | $(90^\circ, 0^\circ)$   | 100%               | 0%                    | 0%                     | 0%                      |                         |
|               | $(90^\circ, 90^\circ)$  | 0%                 | 100%                  | 0%                     | 0%                      |                         |
|               | $(90^\circ, 180^\circ)$ | 0%                 | 0%                    | 100%                   | 0%                      |                         |
|               | $(90^\circ, 270^\circ)$ | 30%                | 0%                    | 0%                     | 70%                     |                         |

**Tabla 8.** Matriz de confusión para un sistema cuadrafónico con posiciones cardinales, señal de voz.

La siguiente composición especifica de manera más concreta desde donde se percibe la conversación entre los dos hablantes.

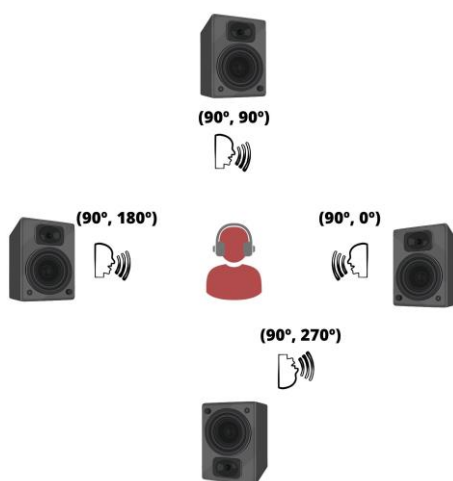


**Figura 74.** Ilustración de la percepción de un sistema cuadrafónico con posiciones cardinales para una señal de voz.

- ♪ Muestra de audio: clip musical de 16 segundos de duración. Este archivo de audio está formado por una melodía interpretada por un piano y que culmina con la aparición de una guitarra, una batería y la voz del cantante (por ello cambia la forma de onda y la amplitud de la señal). Comparamos, al igual que en el caso de la señal de voz, las formas de onda de la señal original y su versión binaural situada en  $(90^\circ, 90^\circ)$ .

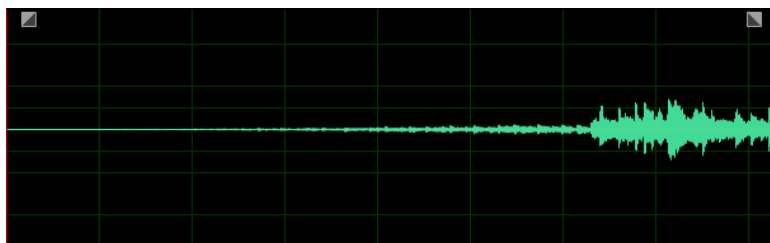


**Figura 75.** Forma de onda del clip de música original.

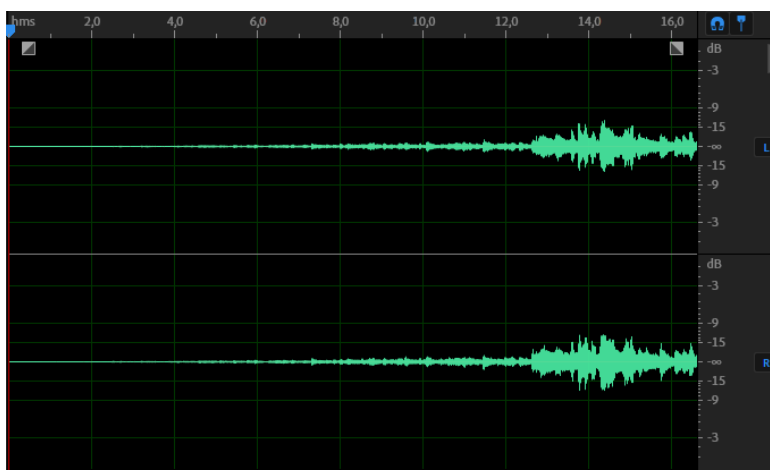


**Figura 76.** Formas de onda de una versión binaural de la señal musical.

|               |                         | Posición percibida |                       |                        |                         |                         |
|---------------|-------------------------|--------------------|-----------------------|------------------------|-------------------------|-------------------------|
|               |                         | $(\theta, \phi)$   | $(90^\circ, 0^\circ)$ | $(90^\circ, 90^\circ)$ | $(90^\circ, 180^\circ)$ | $(90^\circ, 270^\circ)$ |
| Posición real | $(90^\circ, 0^\circ)$   | 100%               | 0%                    | 0%                     | 0%                      |                         |
|               | $(90^\circ, 90^\circ)$  | 0%                 | 90%                   | 10%                    | 0%                      |                         |
|               | $(90^\circ, 180^\circ)$ | 0%                 | 0%                    | 100%                   | 0%                      |                         |
|               | $(90^\circ, 270^\circ)$ | 0%                 | 0%                    | 0%                     | 100%                    |                         |

**Tabla 9.** Matriz de confusión para un sistema cuadrafónico con posiciones cardinales, señal musical.

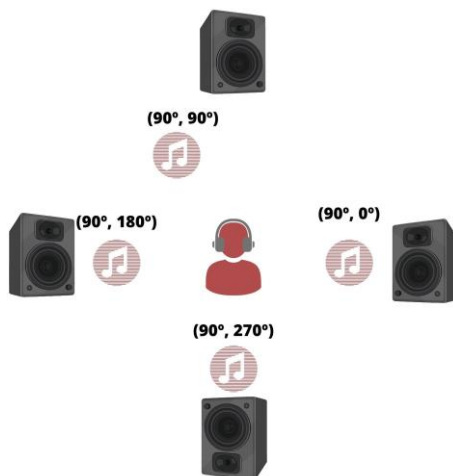


**Figura 77.** Ilustración de la percepción de un sistema cuadrafónico con posiciones cardinales para una señal musical.

- A continuación, se presenta el sistema cuadrafónico formal. Vemos que las posiciones involucradas, que forman  $45^\circ$  con el oyente desde sus planos horizontal y transversal, no son tan sencillas de percibir. Esto ocurre debido a que, aunque el oído humano cuenta con sistemas de localización de sonidos bastante potentes, puede no tener completamente claro el origen de la señal. Aun así, los resultados son muy prometedores, sobre todo en la sección frente al oyente.

♪ Muestra de audio: senoide de frecuencia y amplitud constantes, duración 1 segundo.

|               |                         | Posición percibida |                        |                         |                         |                         |
|---------------|-------------------------|--------------------|------------------------|-------------------------|-------------------------|-------------------------|
|               |                         | $(\theta, \phi)$   | $(90^\circ, 45^\circ)$ | $(90^\circ, 135^\circ)$ | $(90^\circ, 225^\circ)$ | $(90^\circ, 315^\circ)$ |
| Posición real | $(90^\circ, 45^\circ)$  | 80%                | 20%                    | 0%                      | 0%                      |                         |
|               | $(90^\circ, 135^\circ)$ | 10%                | 90%                    | 0%                      | 0%                      |                         |
|               | $(90^\circ, 225^\circ)$ | 0%                 | 0%                     | 60%                     | 40%                     |                         |
|               | $(90^\circ, 315^\circ)$ | 0%                 | 0%                     | 50%                     | 50%                     |                         |

**Tabla 10.** Matriz de confusión para un sistema cuadrafónico, señal sinusoidal.

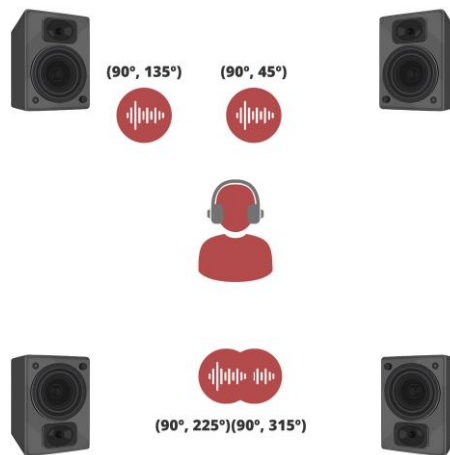


**Figura 78.** Ilustración de la percepción de un sistema cuadrafónico.

♪ Muestra de audio: grabación de voz de duración 9 segundos.

|               |                         | Posición percibida |                        |                         |                         |                         |
|---------------|-------------------------|--------------------|------------------------|-------------------------|-------------------------|-------------------------|
|               |                         | $(\theta, \phi)$   | $(90^\circ, 45^\circ)$ | $(90^\circ, 135^\circ)$ | $(90^\circ, 225^\circ)$ | $(90^\circ, 315^\circ)$ |
| Posición real | $(90^\circ, 45^\circ)$  | 75%                | 15%                    | 0%                      | 0%                      |                         |
|               | $(90^\circ, 135^\circ)$ | 0%                 | 90%                    | 10%                     | 0%                      |                         |
|               | $(90^\circ, 225^\circ)$ | 0%                 | 0%                     | 80%                     | 20%                     |                         |
|               | $(90^\circ, 315^\circ)$ | 0%                 | 0%                     | 50%                     | 50%                     |                         |

**Tabla 11.** Matriz de confusión para un sistema cuadrafónico, señal de voz.

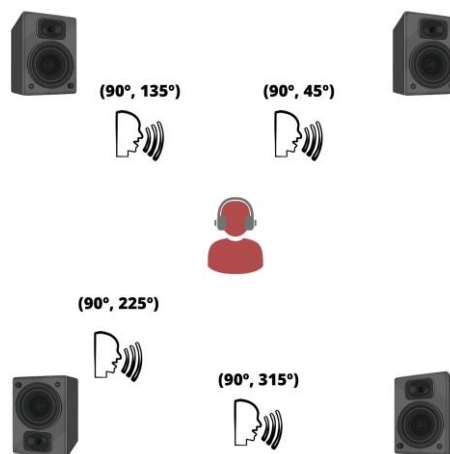


**Figura 79.** Ilustración de la percepción de un sistema cuadrafónico, señal de voz.

- ♪ Muestra de audio: clip musical de duración 16 segundos. Visualizando los resultados, llegamos a la conclusión de que la percepción depende también del tipo de señal utilizada. En este caso, hay una mayor precisión en situar las fuentes virtuales delanteras y la trasera izquierda, mientras que la trasera derecha se percibe desde la parte posterior central.

|               |                         | Posición percibida |                        |                         |                         |                         |
|---------------|-------------------------|--------------------|------------------------|-------------------------|-------------------------|-------------------------|
|               |                         | $(\theta, \phi)$   | $(90^\circ, 45^\circ)$ | $(90^\circ, 135^\circ)$ | $(90^\circ, 225^\circ)$ | $(90^\circ, 315^\circ)$ |
| Posición real | $(90^\circ, 45^\circ)$  | 90%                | 10%                    | 0%                      | 0%                      |                         |
|               | $(90^\circ, 135^\circ)$ | 10%                | 90%                    | 0%                      | 0%                      |                         |
|               | $(90^\circ, 225^\circ)$ | 0%                 | 0%                     | 80%                     | 20%                     |                         |
|               | $(90^\circ, 315^\circ)$ | 0%                 | 0%                     | 50%                     | 50%                     |                         |

**Tabla 12.** Matriz de confusión para un sistema cuadrafónico, señal musical.

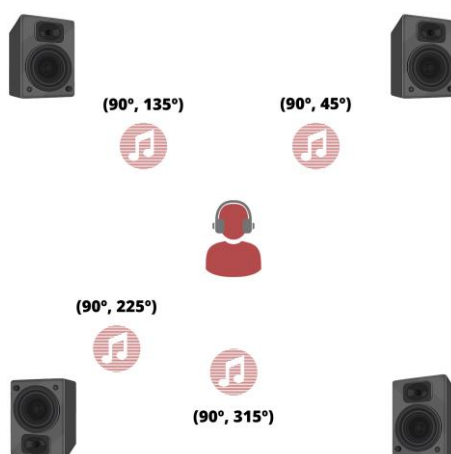


**Figura 80.** Ilustración de la percepción de un sistema cuadrafónico, señal de música

- Por último, llegamos a la decodificación binaural para un sistema 7.1.4. Al igual que en los resultados anteriores, el área situada delante del oyente es la que mejor se percibe. El ángulo de elevación de  $90^\circ$  se corresponde con los altavoces situados en el plano horizontal, a la altura de los oídos del espectador, mientras que el valor  $45^\circ$  es aquel que indica la posición de los altavoces emplazados por encima del oyente. Observamos mediante los resultados de la matriz de confusión<sup>22</sup> que incluso se detectan los cambios en el plano vertical para los altavoces situados delante, mientras que los oídos encuentran más dificultades en la discriminación de sonidos traseros.

<sup>22</sup> Se ha dividido la tabla para una mejor lectura de los datos.

♪ Muestra de audio: senoide de frecuencia y amplitud constantes, duración 1 segundo.

|               |                         | Posición percibida |                       |                        |                        |                        |                         |                         |
|---------------|-------------------------|--------------------|-----------------------|------------------------|------------------------|------------------------|-------------------------|-------------------------|
|               |                         | $(\theta, \phi)$   | $(90^\circ, 0^\circ)$ | $(90^\circ, 60^\circ)$ | $(45^\circ, 68^\circ)$ | $(90^\circ, 90^\circ)$ | $(45^\circ, 112^\circ)$ | $(90^\circ, 120^\circ)$ |
| Posición real | $(90^\circ, 0^\circ)$   | 100%               | 0%                    | 0%                     | 0%                     | 0%                     | 0%                      | 0%                      |
|               | $(90^\circ, 60^\circ)$  | 0%                 | 80%                   | 10%                    | 10%                    | 0%                     | 0%                      | 0%                      |
|               | $(45^\circ, 68^\circ)$  | 0%                 | 10%                   | 90%                    | 0%                     | 0%                     | 0%                      | 0%                      |
|               | $(90^\circ, 90^\circ)$  | 0%                 | 10%                   | 0%                     | 90%                    | 0%                     | 0%                      | 0%                      |
|               | $(45^\circ, 112^\circ)$ | 0%                 | 0%                    | 0%                     | 0%                     | 80%                    | 20%                     |                         |
|               | $(90^\circ, 120^\circ)$ | 0%                 | 0%                    | 0%                     | 0%                     | 20%                    | 80%                     |                         |

|               |                         | Posición percibida |                         |                         |                         |                         |                         |
|---------------|-------------------------|--------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
|               |                         | $(\theta, \phi)$   | $(90^\circ, 180^\circ)$ | $(45^\circ, 225^\circ)$ | $(90^\circ, 240^\circ)$ | $(90^\circ, 300^\circ)$ | $(45^\circ, 315^\circ)$ |
| Posición real | $(90^\circ, 180^\circ)$ | 100%               | 0%                      | 0%                      | 0%                      | 0%                      | 0%                      |
|               | $(45^\circ, 225^\circ)$ | 10%                | 70%                     | 20%                     | 0%                      | 0%                      | 0%                      |
|               | $(90^\circ, 240^\circ)$ | 0%                 | 10%                     | 70%                     | 10%                     | 10%                     | 10%                     |
|               | $(90^\circ, 300^\circ)$ | 0%                 | 0%                      | 10%                     | 80%                     | 10%                     | 10%                     |
|               | $(45^\circ, 315^\circ)$ | 0%                 | 0%                      | 10%                     | 20%                     | 70%                     | 70%                     |

**Tabla 13.** Matriz de confusión para un sistema 7.1.4, señal sinusoidal.

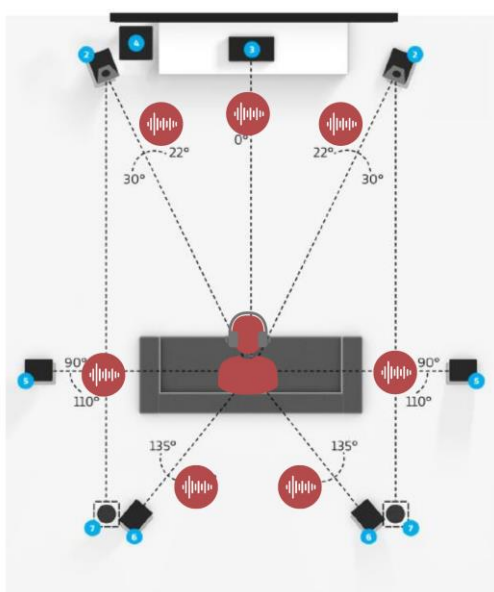


**Figura 81.** Ilustración de la percepción de un sistema 7.1.4 en el plano horizontal.

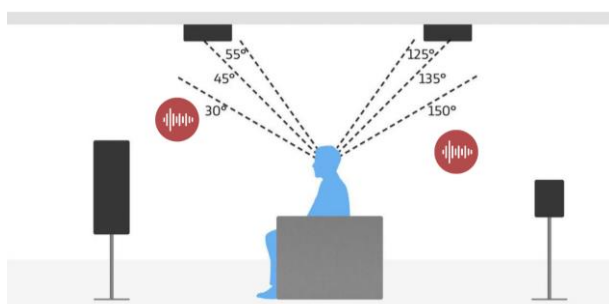


**Figura 82.** Ilustración de la percepción de un sistema 7.1.4 en el plano vertical.

- ♪ Muestra de audio: grabación de voz de duración 9 segundos. Verificamos que los resultados obtenidos una pista de audio más larga permite una mejor identificación de la dirección sonora por parte del sujeto. Comprobamos los valores mediante los porcentajes de acierto en las tablas a continuación. Volvemos a confirmar que los sonidos provenientes de la parte posterior de la cabeza son los más complicados de ubicar.

|               |                         | Posición percibida |                       |                        |                        |                        |                         |                         |
|---------------|-------------------------|--------------------|-----------------------|------------------------|------------------------|------------------------|-------------------------|-------------------------|
|               |                         | $(\theta, \phi)$   | $(90^\circ, 0^\circ)$ | $(90^\circ, 60^\circ)$ | $(45^\circ, 68^\circ)$ | $(90^\circ, 90^\circ)$ | $(45^\circ, 112^\circ)$ | $(90^\circ, 120^\circ)$ |
| Posición real | $(90^\circ, 0^\circ)$   | 100%               | 0%                    | 0%                     | 0%                     | 0%                     | 0%                      | 0%                      |
|               | $(90^\circ, 60^\circ)$  | 0%                 | 80%                   | 10%                    | 10%                    | 0%                     | 0%                      | 0%                      |
|               | $(45^\circ, 68^\circ)$  | 0%                 | 10%                   | 90%                    | 0%                     | 0%                     | 0%                      | 0%                      |
|               | $(90^\circ, 90^\circ)$  | 0%                 | 0%                    | 0%                     | 100%                   | 0%                     | 0%                      | 0%                      |
|               | $(45^\circ, 112^\circ)$ | 0%                 | 0%                    | 0%                     | 0%                     | 70%                    | 30%                     |                         |
|               | $(90^\circ, 120^\circ)$ | 0%                 | 0%                    | 0%                     | 0%                     | 0%                     | 100%                    |                         |
|               |                         |                    |                       |                        |                        |                        |                         |                         |

|               |                         | Posición percibida |                         |                         |                         |                         |                         |
|---------------|-------------------------|--------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
|               |                         | $(\theta, \phi)$   | $(90^\circ, 180^\circ)$ | $(45^\circ, 225^\circ)$ | $(90^\circ, 240^\circ)$ | $(90^\circ, 300^\circ)$ | $(45^\circ, 315^\circ)$ |
| Posición real | $(90^\circ, 180^\circ)$ | 100%               | 0%                      | 0%                      | 0%                      | 0%                      | 0%                      |
|               | $(45^\circ, 225^\circ)$ | 0%                 | 70%                     | 30%                     | 0%                      | 0%                      | 0%                      |
|               | $(90^\circ, 240^\circ)$ | 0%                 | 0%                      | 80%                     | 10%                     | 10%                     | 10%                     |
|               | $(90^\circ, 300^\circ)$ | 0%                 | 0%                      | 50%                     | 50%                     | 0%                      | 0%                      |
|               | $(45^\circ, 315^\circ)$ | 0%                 | 0%                      | 0%                      | 10%                     | 90%                     | 0%                      |

**Tabla 14.** Matriz de confusión para un sistema 7.1.4, señal de voz.

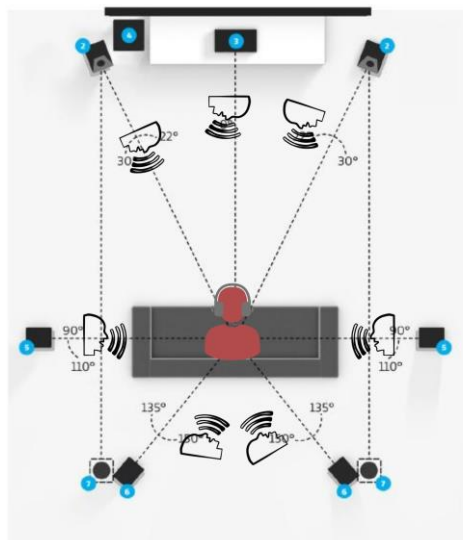


**Figura 83.** Ilustración de la percepción de un sistema 7.1.4 en el plano horizontal, señal de voz.

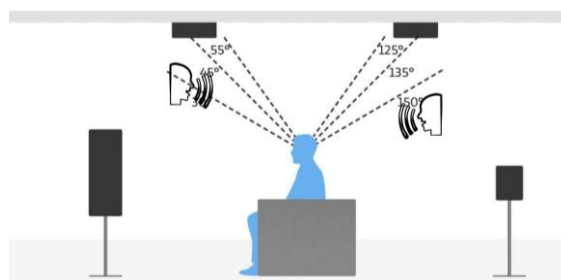


**Figura 84.** Ilustración de la percepción de un sistema 7.1.4 en el plano vertical, señal de voz.

- ♪ Muestra de audio: clip musical de duración 16 segundos. En la segunda fila, donde  $(\theta, \phi) = (45^\circ, 68^\circ)$ , se indica el 100% de precisión porque se percibe la señal desde una altura mayor que la posición de los oídos. Visualizando los resultados, volvemos a comprobar que los resultados obtenidos mediante la escucha de una señal musical son muy satisfactorios.

|               |                         | Posición percibida |                       |                        |                        |                        |                         |                         |
|---------------|-------------------------|--------------------|-----------------------|------------------------|------------------------|------------------------|-------------------------|-------------------------|
|               |                         | $(\theta, \phi)$   | $(90^\circ, 0^\circ)$ | $(90^\circ, 60^\circ)$ | $(45^\circ, 68^\circ)$ | $(90^\circ, 90^\circ)$ | $(45^\circ, 112^\circ)$ | $(90^\circ, 120^\circ)$ |
| Posición real | $(90^\circ, 0^\circ)$   | 100%               | 0%                    | 0%                     | 0%                     | 0%                     | 0%                      | 0%                      |
|               | $(90^\circ, 60^\circ)$  | 0%                 | 90%                   | 10%                    | 0%                     | 0%                     | 0%                      | 0%                      |
|               | $(45^\circ, 68^\circ)$  | 0%                 | 0%                    | 100%                   | 0%                     | 0%                     | 0%                      | 0%                      |
|               | $(90^\circ, 90^\circ)$  | 0%                 | 0%                    | 0%                     | 100%                   | 0%                     | 0%                      | 0%                      |
|               | $(45^\circ, 112^\circ)$ | 0%                 | 0%                    | 0%                     | 0%                     | 100%                   | 0%                      | 0%                      |
|               | $(90^\circ, 120^\circ)$ | 0%                 | 0%                    | 0%                     | 0%                     | 10%                    | 90%                     |                         |

|               |                         | Posición percibida |                         |                         |                         |                         |                         |
|---------------|-------------------------|--------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
|               |                         | $(\theta, \phi)$   | $(90^\circ, 180^\circ)$ | $(45^\circ, 225^\circ)$ | $(90^\circ, 240^\circ)$ | $(90^\circ, 300^\circ)$ | $(45^\circ, 315^\circ)$ |
| Posición real | $(90^\circ, 180^\circ)$ | 100%               | 0%                      | 0%                      | 0%                      | 0%                      | 0%                      |
|               | $(45^\circ, 225^\circ)$ | 0%                 | 90%                     | 10%                     | 0%                      | 0%                      | 0%                      |
|               | $(90^\circ, 240^\circ)$ | 0%                 | 0%                      | 100%                    | 0%                      | 0%                      | 0%                      |
|               | $(90^\circ, 300^\circ)$ | 0%                 | 0%                      | 0%                      | 100%                    | 0%                      | 0%                      |
|               | $(45^\circ, 315^\circ)$ | 0%                 | 0%                      | 0%                      | 0%                      | 0%                      | 100%                    |

**Tabla 15.** Matriz de confusión para un sistema 7.1.4, señal musical.

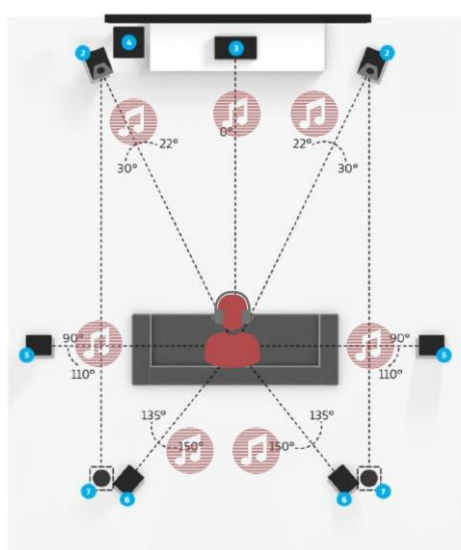


**Figura 85.** Ilustración de la percepción de un sistema 7.1.4 en el plano horizontal, señal musical.

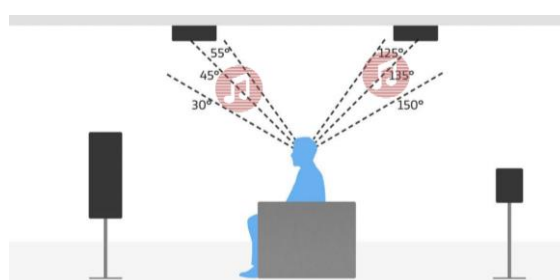


**Figura 86.** Ilustración de la percepción de un sistema 7.1.4 en el plano vertical, señal musical.

### Pruebas sobre sujeto 2

En este caso, se han probado únicamente los archivos de audio más complejos, ya que mediante el pulso el sujeto (no involucrado en el desarrollo de este TFM) no percibe correctamente la posición de las fuentes. Mediante la aplicación de *tracks* más largos y atractivos, el oyente puede situar con más precisión los altavoces virtuales.

- Fuentes situadas en los puntos cardinales.

- ♪ Muestra de audio: grabación de voz de 9 segundos. El sujeto percibe correctamente la posición de los altavoces virtuales. En el caso del altavoz situado en la parte posterior, reconoce que el sonido lo escucha desde la zona trasera de su cabeza.

|               |                         | Posición percibida |                       |                        |                         |                         |
|---------------|-------------------------|--------------------|-----------------------|------------------------|-------------------------|-------------------------|
|               |                         | $(\theta, \phi)$   | $(90^\circ, 0^\circ)$ | $(90^\circ, 90^\circ)$ | $(90^\circ, 180^\circ)$ | $(90^\circ, 270^\circ)$ |
| Posición real | $(90^\circ, 0^\circ)$   | 100%               | 0%                    | 0%                     | 0%                      |                         |
|               | $(90^\circ, 90^\circ)$  | 0%                 | 100%                  | 0%                     | 0%                      |                         |
|               | $(90^\circ, 180^\circ)$ | 0%                 | 0%                    | 100%                   | 0%                      |                         |
|               | $(90^\circ, 270^\circ)$ | 0%                 | 0%                    | 0%                     | 100%                    |                         |

**Tabla 16.** Matriz de confusión para un sistema cuadrafónico con posiciones cardinales, señal de voz.

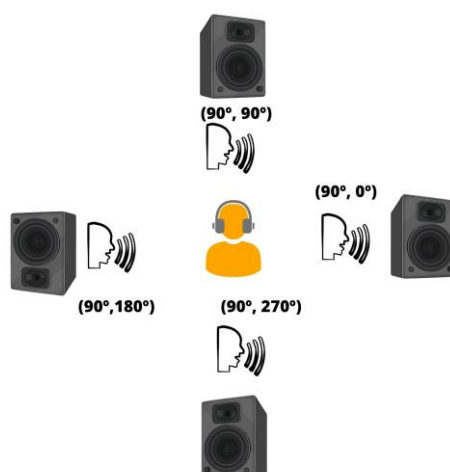


**Figura 87.** Ilustración de la percepción de un sistema cuadrafónico con posiciones cardinales, señal de voz.

- ♪ Muestra de audio: clip musical de 16 segundos de duración. En este caso, el sujeto detecta como origen el cuadrante delantero izquierdo cuando el sonido del altavoz virtual está realmente posicionado en frente. El resto los sitúa correctamente.

|               |                         | Posición percibida |                       |                        |                         |                         |
|---------------|-------------------------|--------------------|-----------------------|------------------------|-------------------------|-------------------------|
|               |                         | $(\theta, \phi)$   | $(90^\circ, 0^\circ)$ | $(90^\circ, 90^\circ)$ | $(90^\circ, 180^\circ)$ | $(90^\circ, 270^\circ)$ |
| Posición real | $(90^\circ, 0^\circ)$   | 100%               | 0%                    | 0%                     | 0%                      |                         |
|               | $(90^\circ, 90^\circ)$  | 0%                 | 60%                   | 40%                    | 0%                      |                         |
|               | $(90^\circ, 180^\circ)$ | 0%                 | 0%                    | 100%                   | 0%                      |                         |
|               | $(90^\circ, 270^\circ)$ | 0%                 | 0%                    | 0%                     | 100%                    |                         |

**Tabla 17.** Matriz de confusión para un sistema cuadrafónico con posiciones cardinales, señal musical.

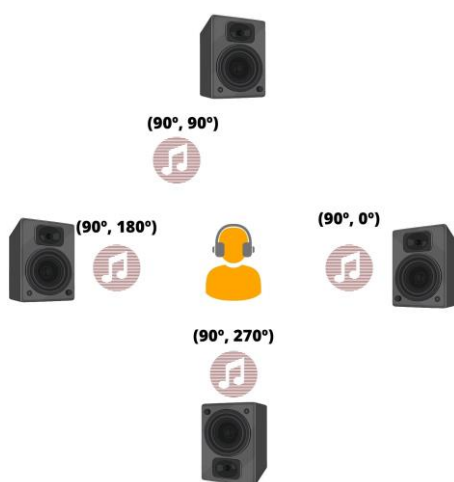


**Figura 88.** Ilustración de la percepción de un sistema cuadrafónico con posiciones cardinales, señal musical.

- Fuentes situadas en un sistema cuadrafónico.
  - ♪ Muestra de audio: grabación de voz de 9 segundos. El sujeto afirma que las posiciones percibidas son las descritas en la Figura 89, de forma que la tendencia de la escucha es hacia un valor del ángulo acimut  $90^\circ$  para el plano frontal y  $270^\circ$  para el trasero.

|               |                         | Posición percibida |                        |                         |                         |                         |
|---------------|-------------------------|--------------------|------------------------|-------------------------|-------------------------|-------------------------|
|               |                         | $(\theta, \phi)$   | $(90^\circ, 45^\circ)$ | $(90^\circ, 135^\circ)$ | $(90^\circ, 225^\circ)$ | $(90^\circ, 315^\circ)$ |
| Posición real | $(90^\circ, 45^\circ)$  | 70%                | 30%                    | 0%                      | 0%                      |                         |
|               | $(90^\circ, 135^\circ)$ | 40%                | 60%                    | 0%                      | 0%                      |                         |
|               | $(90^\circ, 225^\circ)$ | 0%                 | 0%                     | 80%                     | 20%                     |                         |
|               | $(90^\circ, 315^\circ)$ | 0%                 | 0%                     | 50%                     | 50%                     |                         |

**Tabla 18.** Matriz de confusión para un sistema cuadrafónico.

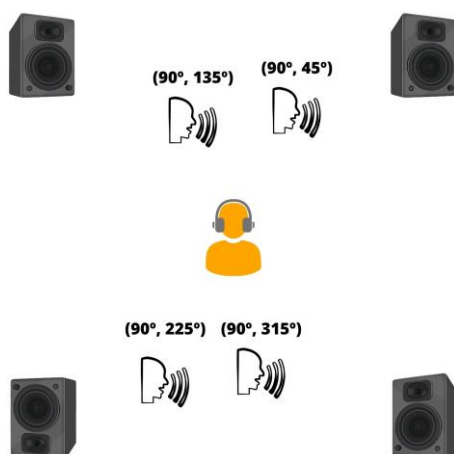


**Figura 89.** Ilustración de la percepción de un sistema cuadrafónico, señal de voz.

- ♪ Muestra de audio: clip musical de 16 segundos. La naturaleza de esta señal condiciona la sensación percibida. Sólo la posición ocupada por los ángulos  $(90^\circ, 135^\circ)$  se sitúan en un lugar alejado de la fuente virtual.

|               |                         | Posición percibida |                        |                         |                         |                         |
|---------------|-------------------------|--------------------|------------------------|-------------------------|-------------------------|-------------------------|
|               |                         | $(\theta, \phi)$   | $(90^\circ, 45^\circ)$ | $(90^\circ, 135^\circ)$ | $(90^\circ, 225^\circ)$ | $(90^\circ, 315^\circ)$ |
| Posición real | $(90^\circ, 45^\circ)$  | 90%                | 10%                    | 0%                      | 0%                      |                         |
|               | $(90^\circ, 135^\circ)$ | 0%                 | 60%                    | 40%                     | 0%                      |                         |
|               | $(90^\circ, 225^\circ)$ | 0%                 | 0%                     | 100%                    | 0%                      |                         |
|               | $(90^\circ, 315^\circ)$ | 0%                 | 0%                     | 0%                      | 100%                    |                         |

**Tabla 19.** Matriz de confusión para un sistema cuadrafónico, señal musical.

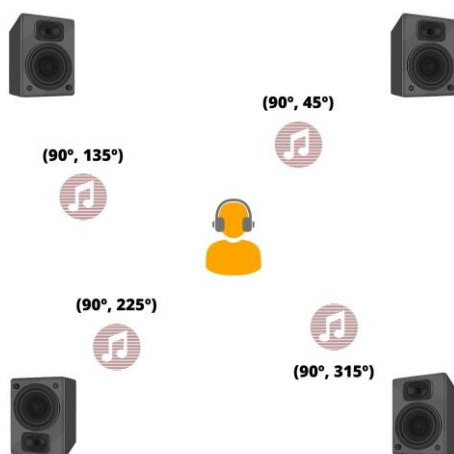


**Figura 90.** Ilustración de la percepción de un sistema cuadrafónico, señal musical.

- Fuentes situadas en un sistema 7.1.4.
  - ♪ Muestra de audio: grabación de voz de 9 segundos. El sujeto sitúa perfectamente las posiciones correspondientes a los ángulos  $(90^\circ, 0^\circ)$ ,  $(90^\circ, 90^\circ)$  y  $(90^\circ, 180^\circ)$ , y se acerca considerablemente a la ubicación de aquellas situadas en las esquinas derecha frontal e izquierda posterior. En la Figura 91 se muestran los lugares señalados por el oyente. Observamos que, la posición posterior derecha es la peor percibida.

|               |                         | Posición percibida |                       |                        |                        |                        |                         |                         |
|---------------|-------------------------|--------------------|-----------------------|------------------------|------------------------|------------------------|-------------------------|-------------------------|
|               |                         | $(\theta, \phi)$   | $(90^\circ, 0^\circ)$ | $(90^\circ, 60^\circ)$ | $(45^\circ, 68^\circ)$ | $(90^\circ, 90^\circ)$ | $(45^\circ, 112^\circ)$ | $(90^\circ, 120^\circ)$ |
| Posición real | $(90^\circ, 0^\circ)$   | 100%               | 0%                    | 0%                     | 0%                     | 0%                     | 0%                      | 0%                      |
|               | $(90^\circ, 60^\circ)$  | 0%                 | 80%                   | 0%                     | 20%                    | 0%                     | 0%                      | 0%                      |
|               | $(45^\circ, 68^\circ)$  | 0%                 | 20%                   | 80%                    | 0%                     | 0%                     | 0%                      | 0%                      |
|               | $(90^\circ, 90^\circ)$  | 0%                 | 0%                    | 0%                     | 100%                   | 0%                     | 0%                      | 0%                      |
|               | $(45^\circ, 112^\circ)$ | 0%                 | 0%                    | 0%                     | 0%                     | 70%                    | 30%                     |                         |
|               | $(90^\circ, 120^\circ)$ | 0%                 | 0%                    | 0%                     | 0%                     | 30%                    | 70%                     |                         |
|               |                         |                    |                       |                        |                        |                        |                         |                         |

|               |                         | Posición percibida |                         |                         |                         |                         |                         |
|---------------|-------------------------|--------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
|               |                         | $(\theta, \phi)$   | $(90^\circ, 180^\circ)$ | $(45^\circ, 225^\circ)$ | $(90^\circ, 240^\circ)$ | $(90^\circ, 300^\circ)$ | $(45^\circ, 315^\circ)$ |
| Posición real | $(90^\circ, 180^\circ)$ | 100%               | 0%                      | 0%                      | 0%                      | 0%                      | 0%                      |
|               | $(45^\circ, 225^\circ)$ | 0%                 | 60%                     | 40%                     | 0%                      | 0%                      | 0%                      |
|               | $(90^\circ, 240^\circ)$ | 0%                 | 10%                     | 70%                     | 20%                     | 0%                      | 0%                      |
|               | $(90^\circ, 300^\circ)$ | 0%                 | 0%                      | 30%                     | 70%                     | 0%                      | 0%                      |
|               | $(45^\circ, 315^\circ)$ | 0%                 | 0%                      | 10%                     | 20%                     | 70%                     | 0%                      |

**Tabla 20.**Matriz de confusión para un sistema 7.1.4, señal de voz.

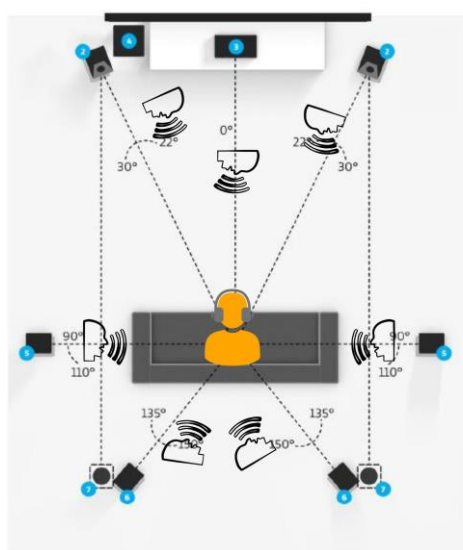


**Figura 91.** Ilustración de la percepción de un sistema 7.1.4 en el plano horizontal, señal de voz.

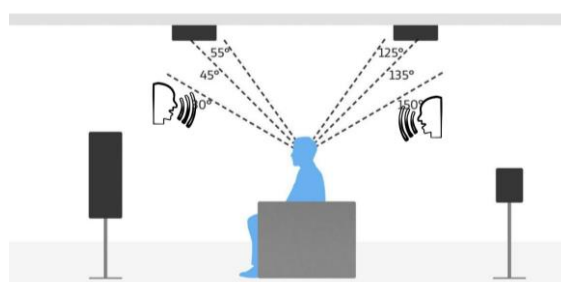


**Figura 92.** Ilustración de la percepción de un sistema 7.1.4 en el plano vertical, señal de voz.

- ♪ Muestra de audio: clip musical segundos. Los valores obtenidos en este caso son mejores que en la muestra vocal. El oyente es capaz de situar aproximadamente todas las fuentes virtuales correctamente.

|               |                         | Posición percibida |                       |                        |                        |                        |                         |                         |
|---------------|-------------------------|--------------------|-----------------------|------------------------|------------------------|------------------------|-------------------------|-------------------------|
|               |                         | $(\theta, \phi)$   | $(90^\circ, 0^\circ)$ | $(90^\circ, 60^\circ)$ | $(45^\circ, 68^\circ)$ | $(90^\circ, 90^\circ)$ | $(45^\circ, 112^\circ)$ | $(90^\circ, 120^\circ)$ |
| Posición real | $(90^\circ, 0^\circ)$   | 100%               | 0%                    | 0%                     | 0%                     | 0%                     | 0%                      | 0%                      |
|               | $(90^\circ, 60^\circ)$  | 0%                 | 80%                   | 0%                     | 20%                    | 0%                     | 0%                      | 0%                      |
|               | $(45^\circ, 68^\circ)$  | 0%                 | 20%                   | 80%                    | 0%                     | 0%                     | 0%                      | 0%                      |
|               | $(90^\circ, 90^\circ)$  | 0%                 | 0%                    | 0%                     | 100%                   | 0%                     | 0%                      | 0%                      |
|               | $(45^\circ, 112^\circ)$ | 0%                 | 0%                    | 0%                     | 0%                     | 80%                    | 20%                     | 0%                      |
|               | $(90^\circ, 120^\circ)$ | 0%                 | 0%                    | 0%                     | 0%                     | 20%                    | 80%                     | 0%                      |

|               |                         | Posición percibida |                         |                         |                         |                         |                         |
|---------------|-------------------------|--------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
|               |                         | $(\theta, \phi)$   | $(90^\circ, 180^\circ)$ | $(45^\circ, 225^\circ)$ | $(90^\circ, 240^\circ)$ | $(90^\circ, 300^\circ)$ | $(45^\circ, 315^\circ)$ |
| Posición real | $(90^\circ, 180^\circ)$ | 100%               | 0%                      | 0%                      | 0%                      | 0%                      | 0%                      |
|               | $(45^\circ, 225^\circ)$ | 0%                 | 70%                     | 20%                     | 10%                     | 0%                      | 0%                      |
|               | $(90^\circ, 240^\circ)$ | 0%                 | 10%                     | 70%                     | 20%                     | 0%                      | 0%                      |
|               | $(90^\circ, 300^\circ)$ | 0%                 | 0%                      | 30%                     | 60%                     | 10%                     | 0%                      |
|               | $(45^\circ, 315^\circ)$ | 0%                 | 0%                      | 10%                     | 20%                     | 70%                     | 0%                      |

**Tabla 21.**Matriz de confusión para un sistema 7.1.4, señal musical.

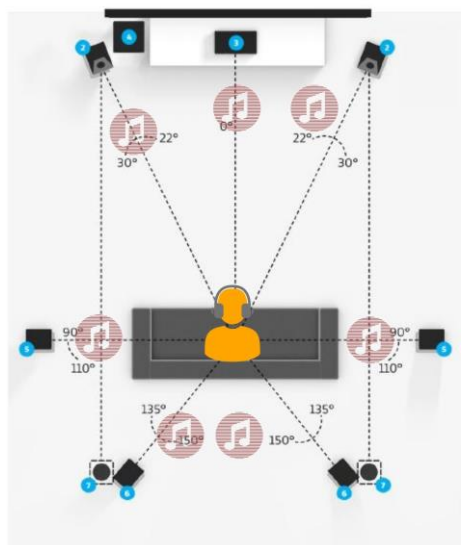


**Figura 93.** Ilustración de la percepción de un sistema 7.1.4 en el plano horizontal, señal musical.

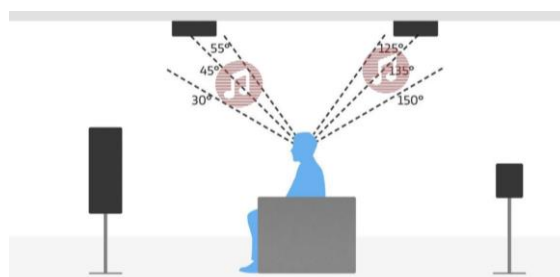


**Figura 94.** Ilustración de la percepción de un sistema 7.1.4 en el plano vertical, señal musical.

Con los resultados obtenidos durante todas las pruebas, mostrados en las tablas y en las ilustraciones gráficas, llegamos a la conclusión de dos comportamientos predominantes. El primero se refiere a que el oyente tiende a mezclar la proveniencia del sonido entre una fuente y la situada de manera contigua. Esto lo vemos, entre otras situaciones, en la Tabla 18, en la que se confunde la posición ( $90^\circ, 135^\circ$ ) con ( $90^\circ, 45^\circ$ ). El segundo comportamiento es que el oído humano no diferencia bien los sonidos cuyas fuentes se sitúan en el plano posterior al oyente debido a, principalmente, la forma de la oreja. Los resultados los podemos comprobar en la Tabla 21, en la que los resultados correspondientes a los ángulos indicados para posiciones posteriores son menos precisos.

## DECODIFICACIÓN DE SEÑALES AMBISONICS CON TRASLACIÓN ESPACIAL

En esta sección, llevaremos a cabo dos ejecuciones distintas de rotación. Primero, computaremos traslaciones presumiblemente perceptibles para confirmar que la experiencia auditiva es la esperada. Por ejemplo, elegiremos una señal cuya fuente está situada a la derecha del oyente (y que hemos comprobado que, efectivamente, se oye desde dicha posición) y la trasladaremos hacia la izquierda del oyente utilizando los ángulos de Euler. Segundo, giraremos la fuente de manera que se sitúe delante del oyente y de esta manera se consiga una experiencia binaural más real, pues hemos confirmado mediante las pruebas anteriores que el plano cuyo ángulo acimut  $\phi \in [0^\circ, 180^\circ]$  es el más fácilmente distinguible.

Seguidamente, se muestran dos señales de dos canales, izquierdo en la parte superior y derecho en la parte inferior, que representan dos fuentes situadas en posiciones distintas. Como se mencionaba en el párrafo anterior, se ha utilizado una fuente a la derecha del oyente y se ha rotado hacia la derecha, utilizando el ángulo  $\gamma$  y otorgándole el valor  $180^\circ$ . No ha sido necesario usar el resto de los ángulos, pues con la rotación sobre el eje  $z$  es suficiente, ya que no se ha pretendido realizar cambios en la posición vertical. En la figura de la izquierda, se contempla una mayor amplitud en la forma de onda inferior que en la superior. Esto se traduce en una percepción auditiva en la que el oído derecho es el que recibe la información más relevante y el cerebro lo traduce como la posición de la fuente. Por el contrario, en las formas de onda de la derecha se comprueba un mayor nivel en la señal del oído izquierdo. De esta manera, confirmamos que el resultado de la rotación es el esperado.

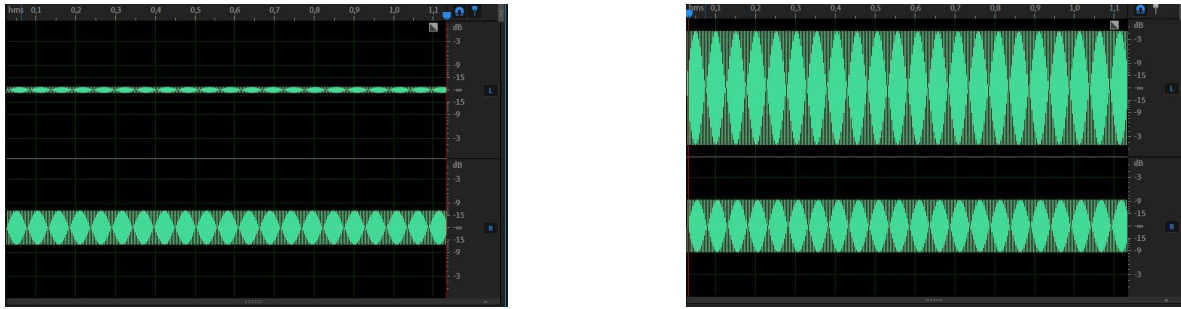


**Figura 95.** Formas de onda de dos señales biaurales. A izquierda, la señal original y a la derecha, la señal rotada. La señal original tiene una posición definida por  $(\theta, \phi) = (90^\circ, 0^\circ)$ , la cual se ha rotado mediante  $(\alpha, \beta, \gamma) = (0^\circ, 0^\circ, 180^\circ)$ .

Probamos ahora con una señal posicionada en la parte posterior izquierda, concretamente en  $(\theta, \phi) = (45^\circ, 225^\circ)$  y la trasladamos mediante los ángulos  $(\alpha, \beta, \gamma) = (90^\circ, 90^\circ, 180^\circ)$ , de forma que esperamos que el sonido se siga percibiendo desde la parte trasera, pero en la sección derecha. En este caso, la visualización de las formas de onda no refleja de manera clara la sensación auditiva. La señal de la izquierda, que se corresponde con la original, tiene más información de amplitud en su canal derecho, aunque al reproducirla se detecta que el origen viene de la zona izquierda trasera. Esto ocurre porque la direccionalidad percibida desde detrás es más compleja que en el caso anterior, donde en la señal original el oído derecho recibe más información debido a que la fuente se encuentra exactamente en esa posición, mientras que el izquierdo contiene menos información de la señal.

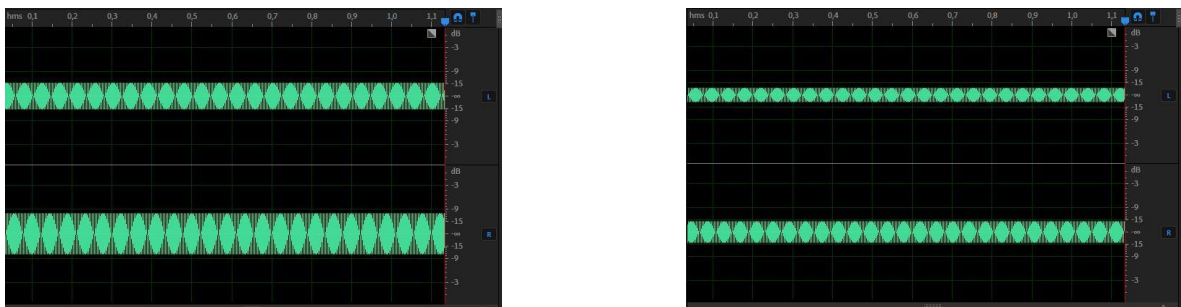


**Figura 96.** Formas de onda de dos señales biaurales. A izquierda, la señal original y a la derecha, la señal rotada. La señal original tiene una posición definida por  $(\theta, \phi) = (45^\circ, 225^\circ)$ , la rotada tiene los valores  $(\alpha, \beta, \gamma) = (90^\circ, 90^\circ, 180^\circ)$ .

A continuación, se va a llevar a cabo un proceso de rotación de cada una de las señales del sistema 7.1.4 en el que todas las fuentes se rotarán virtualmente al plano frontal del oyente. La información obtenida se recoge en la siguiente tabla donde las filas indican la posición inicial de la fuente, la primera columna incluye los valores de los ángulos de rotación de Euler utilizados, la segunda columna muestra el éxito en la traslación y la tercera concreta los resultados obtenidos.

Prueba sobre el sujeto 1:

- ♪ Muestra de audio: senoide de frecuencia y amplitud constantes, duración 1 segundo.

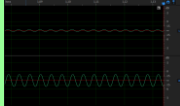
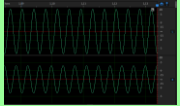
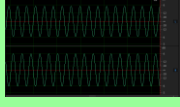
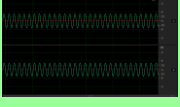
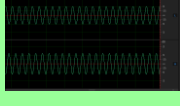
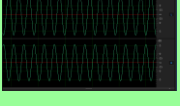
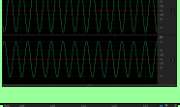
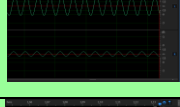
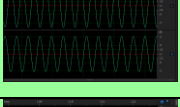
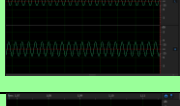
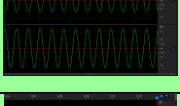
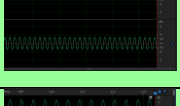
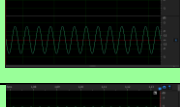
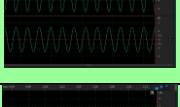
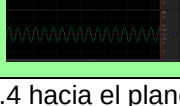
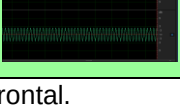
| $(\theta, \phi)$        | $(\alpha, \beta, \gamma)$         | Percepción frontal | Percepción real                 | Señal original                                                                        | Señal rotada                                                                          |
|-------------------------|-----------------------------------|--------------------|---------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| $(90^\circ, 0^\circ)$   | $(0^\circ, 0^\circ, 90^\circ)$    | ✓                  | Frontal                         |    |    |
| $(90^\circ, 60^\circ)$  | $(0^\circ, 0^\circ, 30^\circ)$    | ×                  | Frontal, izquierda              |    |    |
| $(45^\circ, 68^\circ)$  | $(0^\circ, 45^\circ, 20^\circ)$   | ✓                  | Frontal                         |    |    |
| $(45^\circ, 112^\circ)$ | $(22^\circ, 45^\circ, 270^\circ)$ | ✓                  | Frontal                         |    |    |
| $(90^\circ, 120^\circ)$ | $(90^\circ, 0^\circ, 220^\circ)$  | ✓                  | Frontal                         |   |   |
| $(90^\circ, 180^\circ)$ | $(90^\circ, 0^\circ, 180^\circ)$  | ✓                  | Frontal                         |  |  |
| $(45^\circ, 225^\circ)$ | $(270^\circ, 45^\circ, 0^\circ)$  | ✓                  | Frontal (ligeramente derecha)   |  |  |
| $(90^\circ, 240^\circ)$ | $(0^\circ, 0^\circ, 180^\circ)$   | ✓                  | Frontal                         |  |  |
| $(90^\circ, 300^\circ)$ | $(0^\circ, 0^\circ, 150^\circ)$   | ✓                  | Frontal (ligeramente izquierda) |  |  |
| $(45^\circ, 315^\circ)$ | $(45^\circ, 45^\circ, 90^\circ)$  | ✓                  | Frontal                         |  |  |

**Tabla 22.** Resultados de la rotación de todos los canales 7.1.4 hacia el plano frontal.

Estos resultados indican que el sistema funciona correctamente y que las soluciones obtenidas son muy satisfactorias. La elección de los ángulos permite una mayor flexibilidad en las posiciones generadas, por ejemplo: el canal 2, situado en los ángulos  $(90^\circ, 60^\circ)$ , no se rota correctamente para el único ángulo que necesita modificación, esto es,  $\gamma = 30^\circ$ , lo cual vemos en la tabla superior. Sin embargo, si modificamos no sólo gamma, sino también alfa (que rotará el mismo plano ya que no se ha introducido ningún valor de beta), la sensación sonora es la esperada: el sonido se percibe desde una fuente virtual situada enfrente del

oyente. Los valores empleados han sido  $(\alpha, \beta, \gamma) = (15^\circ, 0^\circ, 15^\circ)$  y se han conseguido dos formas de onda equiparadas para cada oído.

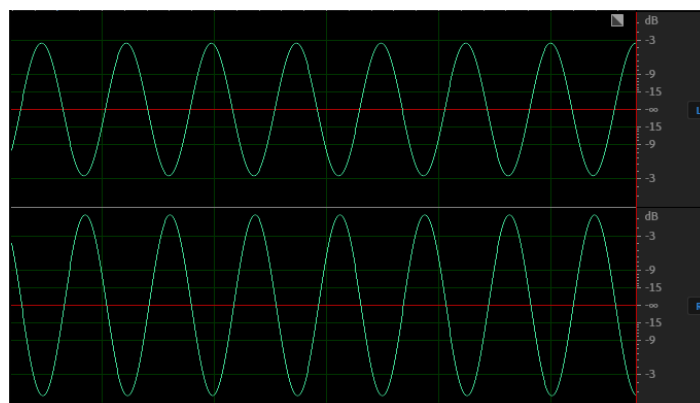


**Figura 97.** Forma de onda de una señal rotada hacia delante.

Probamos ahora con los resultados de las rotaciones de los archivos de voz y música. Comenzaremos con el sujeto 1 y proseguiremos con el sujeto 2.

- ♪ Muestra de audio: grabación de voz de 9 segundos. En este caso, no se han incorporado las formas de onda de la señal original y la rotada, ya que se ha considerado que no ofrecen una información relevante en cuanto a la percepción de la orientación de la fuente sonora. Se ha optado por ofrecer un poco más de información acerca del sonido recibido.

| $(\theta, \phi)$        | $(\alpha, \beta, \gamma)$         | Percepción frontal | Percepción real                                                                    |
|-------------------------|-----------------------------------|--------------------|------------------------------------------------------------------------------------|
| $(90^\circ, 0^\circ)$   | $(0^\circ, 0^\circ, 90^\circ)$    | ✓                  | Exclusivamente frontal.                                                            |
| $(90^\circ, 60^\circ)$  | $(0^\circ, 0^\circ, 30^\circ)$    | ×                  | Frontal, con un poco de influencia en el ángulo $\phi = 60^\circ$ .                |
| $(45^\circ, 68^\circ)$  | $(0^\circ, 45^\circ, 20^\circ)$   | ✓                  | Exclusivamente frontal.                                                            |
| $(45^\circ, 112^\circ)$ | $(22^\circ, 45^\circ, 270^\circ)$ | ✓                  | Exclusivamente frontal.                                                            |
| $(90^\circ, 120^\circ)$ | $(90^\circ, 0^\circ, 220^\circ)$  | ✓                  | Frontal. La voz del artista se percibe ligeramente en el ángulo $\phi = 120^\circ$ |
| $(90^\circ, 180^\circ)$ | $(90^\circ, 0^\circ, 180^\circ)$  | ✓                  | Exclusivamente frontal.                                                            |
| $(45^\circ, 225^\circ)$ | $(270^\circ, 45^\circ, 0^\circ)$  | ✓                  | Exclusivamente frontal.                                                            |
| $(90^\circ, 240^\circ)$ | $(0^\circ, 0^\circ, 180^\circ)$   | ✓                  | Exclusivamente frontal.                                                            |
| $(90^\circ, 300^\circ)$ | $(0^\circ, 0^\circ, 150^\circ)$   | ✓                  | Exclusivamente frontal.                                                            |
| $(45^\circ, 315^\circ)$ | $(45^\circ, 45^\circ, 90^\circ)$  | ✓                  | Exclusivamente frontal.                                                            |

**Tabla 23.** Resultados de la rotación de todos los canales 7.1.4 hacia el plano frontal, grabación de voz.

- ♪ Muestra de audio: clip musical de 16 segundos. Al igual que en la prueba anterior, tampoco se ha considerado necesario incorporar las formas de onda. Es importante mencionar que en esta prueba se ha notado de manera muy clara cómo las fuentes se han rotado hacia delante, por lo que el resultado es muy satisfactorio.

| $(\theta, \phi)$        | $(\alpha, \beta, \gamma)$         | Percepción frontal | Percepción real                                                                    |
|-------------------------|-----------------------------------|--------------------|------------------------------------------------------------------------------------|
| $(90^\circ, 0^\circ)$   | $(0^\circ, 0^\circ, 90^\circ)$    | ✓                  | Exclusivamente frontal.                                                            |
| $(90^\circ, 60^\circ)$  | $(0^\circ, 0^\circ, 30^\circ)$    | ✓                  | Frontal, con un poco de influencia en el ángulo $\phi = 60^\circ$ .                |
| $(45^\circ, 68^\circ)$  | $(0^\circ, 45^\circ, 20^\circ)$   | ✓                  | Exclusivamente frontal.                                                            |
| $(45^\circ, 112^\circ)$ | $(22^\circ, 45^\circ, 270^\circ)$ | ✓                  | Exclusivamente frontal.                                                            |
| $(90^\circ, 120^\circ)$ | $(90^\circ, 0^\circ, 220^\circ)$  | ✓                  | Frontal. La voz del artista se percibe ligeramente en el ángulo $\phi = 120^\circ$ |
| $(90^\circ, 180^\circ)$ | $(90^\circ, 0^\circ, 180^\circ)$  | ✓                  | Exclusivamente frontal.                                                            |
| $(45^\circ, 225^\circ)$ | $(270^\circ, 45^\circ, 0^\circ)$  | ✓                  | Exclusivamente frontal.                                                            |
| $(90^\circ, 240^\circ)$ | $(0^\circ, 0^\circ, 180^\circ)$   | ✓                  | Exclusivamente frontal.                                                            |
| $(90^\circ, 300^\circ)$ | $(0^\circ, 0^\circ, 150^\circ)$   | ✓                  | Exclusivamente frontal.                                                            |
| $(45^\circ, 315^\circ)$ | $(45^\circ, 45^\circ, 90^\circ)$  | ✓                  | Exclusivamente frontal.                                                            |

**Tabla 24.** Resultados de la rotación de todos los canales 7.1.4 hacia el plano frontal, clip musical.

### Pruebas sobre el sujeto 2

En este caso, se pide al sujeto que sitúe aproximadamente la posición desde donde emana el sonido. Al no poder señalar con exactitud los ángulos (tal y como se ha comentado en el caso del sujeto 1, que soy yo misma), se considera como negativo el resultado diferente de la posición delantera exacta.

♪ Muestra de audio: grabación de voz de 9 segundos.

| $(\theta, \phi)$        | $(\alpha, \beta, \gamma)$         | Percepción frontal | Percepción real                             |
|-------------------------|-----------------------------------|--------------------|---------------------------------------------|
| $(90^\circ, 0^\circ)$   | $(0^\circ, 0^\circ, 90^\circ)$    | ✓                  | Exclusivamente frontal.                     |
| $(90^\circ, 60^\circ)$  | $(0^\circ, 0^\circ, 30^\circ)$    | ×                  | Frontal. Perceptiblemente hacia la derecha. |
| $(45^\circ, 68^\circ)$  | $(0^\circ, 45^\circ, 20^\circ)$   | ×                  | Frontal, derecha.                           |
| $(45^\circ, 112^\circ)$ | $(22^\circ, 45^\circ, 270^\circ)$ | ✓                  | Exclusivamente frontal.                     |
| $(90^\circ, 120^\circ)$ | $(90^\circ, 0^\circ, 220^\circ)$  | ×                  | Frontal, derecha.                           |
| $(90^\circ, 180^\circ)$ | $(90^\circ, 0^\circ, 180^\circ)$  | ✓                  | Frontal con percepción de altura.           |
| $(45^\circ, 225^\circ)$ | $(270^\circ, 45^\circ, 0^\circ)$  | ✓                  | Exclusivamente frontal.                     |
| $(90^\circ, 240^\circ)$ | $(0^\circ, 0^\circ, 180^\circ)$   | ×                  | Frontal, izquierda.                         |
| $(90^\circ, 300^\circ)$ | $(0^\circ, 0^\circ, 150^\circ)$   | ✓                  | Exclusivamente frontal.                     |
| $(45^\circ, 315^\circ)$ | $(45^\circ, 45^\circ, 90^\circ)$  | ✓                  | Exclusivamente frontal.                     |

**Tabla 25.** Resultados de la rotación de todos los canales 7.1.4 hacia el plano frontal, grabación de voz.

♪ Muestra de audio: clip musical de 16 segundos.

| $(\theta, \phi)$        | $(\alpha, \beta, \gamma)$         | Percepción frontal | Percepción real                               |
|-------------------------|-----------------------------------|--------------------|-----------------------------------------------|
| $(90^\circ, 0^\circ)$   | $(0^\circ, 0^\circ, 90^\circ)$    | ✓                  | Exclusivamente frontal.                       |
| $(90^\circ, 60^\circ)$  | $(0^\circ, 0^\circ, 30^\circ)$    | ✓                  | Frontal, ligeramente hacia la derecha.        |
| $(45^\circ, 68^\circ)$  | $(0^\circ, 45^\circ, 20^\circ)$   | ✓                  | Exclusivamente frontal.                       |
| $(45^\circ, 112^\circ)$ | $(22^\circ, 45^\circ, 270^\circ)$ | ×                  | Frontal, izquierda.                           |
| $(90^\circ, 120^\circ)$ | $(90^\circ, 0^\circ, 220^\circ)$  | ✓                  | Exclusivamente frontal.                       |
| $(90^\circ, 180^\circ)$ | $(90^\circ, 0^\circ, 180^\circ)$  | ×                  | Frontal. Perceptiblemente desde la izquierda. |
| $(45^\circ, 225^\circ)$ | $(270^\circ, 45^\circ, 0^\circ)$  | ✓                  | Exclusivamente frontal.                       |
| $(90^\circ, 240^\circ)$ | $(0^\circ, 0^\circ, 180^\circ)$   | ✓                  | Exclusivamente frontal.                       |
| $(90^\circ, 300^\circ)$ | $(0^\circ, 0^\circ, 150^\circ)$   | ✓                  | Exclusivamente frontal.                       |
| $(45^\circ, 315^\circ)$ | $(45^\circ, 45^\circ, 90^\circ)$  | ✓                  | Exclusivamente frontal.                       |

**Tabla 26.** Resultados de la rotación de todos los canales 7.1.4 hacia el plano frontal, clip musical.

Tras estas pruebas, concluimos que los resultados son muy satisfactorios, teniendo en cuenta que la grabación y obtención de las *HRTF* no se ha llevado a cabo de manera estrictamente profesional. Tal y como se esperaba, la experiencia de usuario es mejor en el caso en el que el oyente es el mismo sujeto para el cual se elaboraron las Respuestas al Impulso Relacionadas con la Cabeza. Sin embargo, nos resulta muy satisfactorio que el sujeto 2 perciba de manera correcta numerosas posiciones y, también, que su sensación se aproxime a la que se pretendía conseguir.

## 7. APLICACIONES

Tal y como indicábamos en la introducción, los avances en la investigación de señales audiovisuales se deben principalmente en el auge de la demanda de contenido multimedia. La aparición de nuevos sistemas, dispositivos y soluciones ha contribuido muy positivamente al aumento de aplicaciones de diversa índole. A continuación, se enumeran los posibles usos de nuestra implementación:

- Integración en una Estación Digital de Audio: el propósito de este trabajo era generar una señal binaural desde una con formato Ambisonics y poder integrar el sistema en un entorno de trabajo real. Tras haberlo conseguido, consideramos que esta es una de las aplicaciones principales, la importación de un Plug – In en cualquier DAW para que los usuarios, ya sean ingenieros, músicos o aficionados, sean capaces de crear un entorno inmersivo de manera fácil y cómoda.
- Transmisión de Audio Digital: cabe esperar que, si existen sistemas capaces de generar señales multicanal ofreciendo una calidad sonora y sensación espacial óptimas, dichas señales puedan retransmitirse por medio de los canales terrestre, satélite o cable. Sin embargo, la mayoría de los sistemas de transmisión digital no pueden ofrecer canales de sonido multicanal debido al alto coste de los equipos de transmisión, el tamaño relativamente pequeño del ancho de banda en proporción a la demanda de contenido por parte de los usuarios, ... [15]. Así, la introducción de sistemas inmersivos mediante dos canales de naturaleza binaural favorecería la experiencia de los usuarios tanto en la transmisión digital como en radio (mediante modulación en frecuencia, FM).
- Cine en casa individualizado: cada vez más, las empresas y fabricantes dedicadas al mundo del espectáculo invierten más en la investigación de sistemas que permitan al espectador formar parte de la historia que están contemplando. Es así como, por ejemplo, se han desarrollado los conocidos sistemas de sonido *Dolby Surround*, basados en las recomendaciones de la ITU. Aunque ofrecen una calidad sonora impecable y son muy demandados, tienen dos inconvenientes principales: el espacio que ocupan y su precio. Por otro lado, cada vez son más los usuarios que disfrutan de cine, series o música de manera individual debido al aumento de plataformas de *streaming* como *Netflix*, *HBO*, *Spotify*, .... Por ello, se considera que el sistema 7.1.4 transformado a binaural tendría gran acogida entre aquellos espectadores que quieran utilizar unos auriculares durante el visionado y así prescindir de los altavoces.
- Servicios de *streaming* o radio mediante Internet: según datos de MPEG, las condiciones del ancho de banda para la transmisión de radio mediante Internet están muy restringidas, así como sus valores de tasa de bit. Gracias a una configuración conjunta mediante los sistemas *surround* y la codificación binaural, se podría conseguir enviar las señales sonoras con un *bitrate* no mayor de 48 kbps.
- Audio para videojuegos: el crecimiento de los usuarios en plataformas de *gaming online* y la exigencia de los mismos, promueve la inversión en sistemas de sonido envolvente. Muchos ordenadores y equipos llevan integrados los sistemas multicanal, mayoritariamente el 5.1. Sin embargo, el resultado no es tan bueno debido a su uso mediante un solo altavoz o auriculares. Así, consideramos que una aplicación interesante sería la introducción de una estructura binaural en este entorno, de manera que el jugador pueda percibir los sonidos (pisadas, naturaleza, voces, explosiones, ...) desde donde se sitúen en pantalla (o en el espacio).
- Realidad virtual e inmersiva: incorporación de auriculares en los sistemas de visión tridimensional para añadir el componente auditivo a la experiencia del usuario mediante la tecnología binaural.

## 8. CONCLUSIONES Y LÍNEAS FUTURAS

### 8.1. CONCLUSIONES

La principal conclusión que hemos obtenido tras la elaboración de este Proyecto es que los sistemas de sonido binaural ofrecen unos resultados excelentes, ya que realmente permiten al oyente situar el origen de la fuente sonora y ofrecen cierta sensación de espacialidad. Si bien es cierto que el resultado del trabajo ha sido muy satisfactorio, también nos hemos encontrado con algunas dificultades que han importunado su realización. Las conclusiones extraídas se explican a continuación en función del orden de implementación.

Inicialmente nos centramos en el desarrollo general de cada uno de los bloques. Gracias al desarrollo piloto mediante Python, corroboramos en primer lugar que los sistemas Ambisonics de formato – B ofrecen una gran cantidad de posibilidades: sonido envolvente, tridimensional, capacidad de transformación a otras configuraciones.... Además, su compatibilidad con las definiciones matemáticas de armónicos esféricos facilita su aplicación en entornos de programación. Por ejemplo, ha sido de gran ayuda en la sección del generador de señales, ya que simplemente con la multiplicación de la señal por cada una de las directividades se ha podido elaborar una señal de 16 canales muy útil para el resto de la implementación. Por otro lado, de rotación de funciones mediante los ángulos de Euler se ha puesto en práctica tanto en Python como en el Plug – In y se ha comprobado que la definición es correcta, pues se consigue un cambio perceptible en las posiciones de los altavoces. En cuanto a la decodificación tanto a 7.1.4 como a binaural, se ha entendido que los procedimientos deben ser distintos ya que el primero puede llevarse a cabo mediante la computación de una matriz de decodificación que contiene las directividades de las señales de cada uno de los altavoces, y el segundo requiere la información de respuesta tanto de la cabeza del oyente como de la sala para conseguir recrear el ambiente sonoro.

En segundo lugar, podemos afirmar que el lenguaje de programación Python, así como el entorno de desarrollo elegido, Spyder, tiene hoy en día el reconocimiento que merecen. Se llega a esta determinación debido a que ha resultado relativamente sencillo conseguir implementar cada uno de los bloques de procesamiento gracias a la existencia de librerías que contienen múltiples funciones predefinidas, lo cual ha ayudado a computar funciones complicadas como los polinomios de Jacobi. Además, no necesita gran capacidad de computación, por lo que la compilación y la depuración han sido fácilmente ejecutables. El único inconveniente del entorno Spyder es el entorno gráfico, que es más limitado y no permite observar los valores con tanta precisión como Matlab. Es por ello por lo que se ha utilizado este software para la visualización de gráficas como las respuestas al impulso o las formas de onda de las señales.

Seguidamente, hemos deducido que el módulo enchufable ha supuesto un gran desarrollo y esfuerzo y que ha resultado ser de gran utilidad, pues gracias a él hemos podido observar el efecto de sonido que queríamos implementar tal y como un usuario común lo utilizaría. A pesar de haber encontrado algunos problemas, sobre todo, relacionados con un lenguaje de bajo nivel como es C++, podemos decir que el Plug – In funciona correctamente y que ofrece una presentación binaural al oyente tal y como se esperaba. Por otro lado, confirmamos que el entorno JUCE es muy potente y que ofrece infinitas posibilidades en el desarrollo de audio, lo cual se cubrirá en el apartado siguiente.

Nuestro Plug – In, desarrollado en el formato VST3, ha resultado exitoso ya que su funcionamiento se ha comprobado en dos DAW, *Reaper* y *Adobe Audition*. Su capacidad de

trabajar en tiempo real tiene lugar en los módulos de rotación y de decodificación a 7.1.4, ya que, si se prueba el sonido, la sensación espacial varía cuando aplicamos dichas funciones. En el caso de la decodificación binaural, esta ejecución se puede aplicar, pero no se escucha con claridad debido a la convolución por bloques.

En cuanto al procedimiento de obtención de las *HRTF*, nos sentimos verdaderamente satisfechos con el resultado generado. El hardware adquirido, los auriculares *Sennheiser Ambeo Smart Headset*, han conseguido suplir la necesidad de generar las respuestas al impulso a pesar de no haber podido contar con la cámara anecoica. Aun así, tras analizar los resultados obtenidos en las pruebas realizadas a los distintos sujetos, verificamos que este parámetro es personal, ya que usándolo en otro oyente el efecto es distinto.

En definitiva, concluimos que el trabajo realizado ha cumplido con las expectativas exigidas y que, tanto de manera teórica como experimental, se ha conseguido llegar a la generación de un sistema que permite al usuario introducirse en una experiencia auditiva inmersiva y tridimensional.

## 8.2. LÍNEAS FUTURAS

Nuestro sistema, si bien realiza las funciones de procesamiento de audio de manera eficiente, puede escalarse de manera que se perfeccione su desarrollo y se lleven a cabo modificaciones que generen nuevas aplicaciones e implementaciones.

- Mejorar la interfaz: generar una *GUI* más elaborada donde la interacción usuario – Plug – In sea más cómoda. Por ejemplo, se puede crear un sistema de coordenadas donde se puedan modificar los ángulos de Euler o, directamente, se pueda elegir la nueva posición de la fuente sin necesidad de definir un valor concreto.
- Computar la decodificación binaural mediante la Transformada de Fourier en *JUCE*: como se apunta en las conclusiones, *ProJucer* es muy potente y ofrece infinitas posibilidades a la hora de procesar audio. Sin embargo, no ha sido posible explotar algunas de las funcionalidades debido a la complejidad del entorno y del manejo de sus señales. Por ello se propone como línea futura la posibilidad de programar la binauralización en el dominio de la frecuencia en lugar en el tiempo.
- Múltiple configuración de altavoces decodificable: ofrecer al usuario la posibilidad de generar el sistema inmersivo desde otros sistemas de sonido además del 7.1.4, por ejemplo: cuadrafónico, 5.1 o 9.1.
- Binauralización en tiempo real: optimizar el código de manera que no sea necesario aplicar el efecto para percibir el efecto sonoro.
- Integración en aplicaciones de *streaming*: en algunas plataformas, en la selección de audio aparece la opción de audio 5.1. Proponemos integrar nuestro sistema binaural en dichas opciones, de manera que el espectador pueda disfrutar el contenido de manera tridimensional.

## 9. BIBLIOGRAFÍA

- [1] B. Rafaely, *Fundamentals on Spherical Array Processing*, Springer, 2018.
- [2] S. Q. B. Grill, «MPEG-1 Audio,» [En línea]. Available: <https://mpeg.chiariglione.org/standards/mpeg-1/audio>.
- [3] S. Q. B. Grill, «MPEG-2 Audio,» [En línea]. Available: <https://mpeg.chiariglione.org/standards/mpeg-2/audio>.
- [4] S. Quackenbush, «MPEG-2 Audio AAC,» [En línea]. Available: <https://mpeg.chiariglione.org/standards/mpeg-2/advanced-audio-coding>.
- [5] S. Quackenbush, «MPEG-4 Audio,» [En línea]. Available: <https://mpeg.chiariglione.org/standards/mpeg-4/audio>.
- [6] J. H. K. Bennett L. Schwartz, «SAGE Edge. Chapter 11. Interaural Level Differences,» [En línea]. Available: <https://isle.hanover.edu/Ch11AudBrainLoc/Ch11InterLoud.html#:~:text=The%20interaural%20level%20difference%20is,will%20sound%20loud%20to%20you.&text=Amazingly%2C%20our%20ears%20can%20detect,the%20left%20and%20right%20ears..>
- [7] J. I. Arribas, «Universidad de Valladolid,» [En línea]. Available: [https://www.lpi.tel.uva.es/~nacho/docencia/ing\\_ond\\_1/trabajos\\_05\\_06/io5/public\\_html/2.2.htm](https://www.lpi.tel.uva.es/~nacho/docencia/ing_ond_1/trabajos_05_06/io5/public_html/2.2.htm).
- [8] B. Moore, *An Introduction to the Psychology of hearing*, Academic Press, 2003.
- [9] J. H. K. Bennett L. Schwartz, «SAGE Edge. Chapter 11. Interaural Time Differences,» [En línea]. Available: <https://isle.hanover.edu/Ch11AudBrainLoc/Ch11InterTime.html>.
- [10] R. Ellen, «Ambisonics, the Surround alternative,» 2001.
- [11] O. R. Claudia Arias, «Audición Espacial en Ambientes Reverberantes: Aspectos Teóricos Relevantes,» *Revista Interamericana de Psicología/Interamerican Journal of Psychology*, 2003.
- [12] S. Q. B. Grill, «MPEG-D Surround,» [En línea]. Available: <https://mpeg.chiariglione.org/standards/mpeg-d/mpeg-surround>.
- [13] S. Q. B. Grill, «MPEG-D Spatial Audio Object Coding,» [En línea]. Available: <https://mpeg.chiariglione.org/standards/mpeg-d/spatial-audio-object-coding>.
- [14] S. Q. B. Grill, «MPEG-D Unified Speech and Audio Coding,» 2011. [En línea]. Available: <https://mpeg.chiariglione.org/standards/mpeg-d/unified-speech-and-audio-coding>.
- [15] F. I. f. I. Circuits, «MPEG-H Audio,» [En línea]. Available: <https://www.iis.fraunhofer.de/en/ff/amm/broadcast-streaming/mpegh.html>.
- [16] A. T. S. Committee, «ATSC A/342:2021 Part 3, MPEG-H System,» 2021.
- [17] J. Plogsties, «International Telecommunications Union,» 2015. [En línea]. Available: <https://www.itu.int/en/ITU-R/study-groups/workshops/2015-TFAB/PublishingImages/Pages/default/Presentation%202-1.pdf>.
- [18] B. Rafaely, «Preface,» de *Fundamentals on Spherical Array Processing*, Springer, 2018.

- [19] B. Rafaely, «Spherical Harmonics,» de *Fundamentals on Spherical Array Processing*, Springer, 2018, pp. 4-13.
- [20] E. Williams, *Fourier Acoustics: Sound Radiation and Nearfield Acoustical Holography*, New York: Academic Press, 1999.
- [21] B. Rafaely, «Exponential and Legendre Functions,» de *Fundamentals of Spherical Array Processing*, Springer, 2018, pp. 13-17.
- [22] B. Rafaely, «Spherical Fourier Transform,» de *Fundamentals of Spherical Array Processing*, Springer, 2018, p. 17.
- [23] B. Rafaely, «Rotation of Functions,» de *Fundamentals of Spherical Array Processing*, Springer, 2018, pp. 24-25.
- [24] A. N. M. D. A. V. V K Khersonskii, «Chapter 4: Wigner Functions,» de *Quantum Theory Of Angular Momentum*, World Scientific, 1988, p. 72.
- [25] B. Rafaely, «Rotation of Functions,» de *Fundamentals of Spherical Array Processing*, Springer, 2018, p. 28.
- [26] R. Ellen, «Surround sound: A brief history and Introduction,» 2000. [En línea]. Available: <https://www.ambisonic.net/about5.1.html>.
- [27] I. T. Union, «UIT-R BS.775-3. Multichannel stereophonic sound system with and without accompanying picture",» 2012.
- [28] I. T. Union, «Report ITU-R BS.2159-7. Multichannel sound technology in home and broadcasting applications. BS Series,» 2015.
- [29] Dolby, «Dolby 5.1.2 Overhead system setup,» [En línea]. Available: [https://www.dolby.com/siteassets/about/support/guide/setup-guides/5.1.2-overhead-speaker-placement/5\\_1\\_2\\_overhead\\_speakers\\_setup.pdf](https://www.dolby.com/siteassets/about/support/guide/setup-guides/5.1.2-overhead-speaker-placement/5_1_2_overhead_speakers_setup.pdf).
- [30] Dolby, «Dolby 5.1.4 Overhead system setup,» [En línea]. Available: <https://www.dolby.com/siteassets/about/support/guide/setup-guides/5.1.4-overhead-speaker-placement/sell-sheet-5.1.4-mounted.pdf>.
- [31] «Dolby Surround 7.1,» [En línea]. Available: <https://www.dolby.com/xl/es/technologies/dolby-surround-7-1.html>.
- [32] Dolby, «Hybrid Dolby Atmos Enabled Speaker Overhead,» [En línea]. Available: [https://www.dolby.com/siteassets/about/support/guide/setup-guides/7.1.4-hybrid-dolby-atmos-enabled--overhead-speaker-placement/7\\_1\\_4\\_hybrid\\_dolby\\_atmos\\_enabled\\_overhead\\_speaker\\_placement.pdf](https://www.dolby.com/siteassets/about/support/guide/setup-guides/7.1.4-hybrid-dolby-atmos-enabled--overhead-speaker-placement/7_1_4_hybrid_dolby_atmos_enabled_overhead_speaker_placement.pdf).
- [33] «Soundfield: Our Story, 1971.,» [En línea]. Available: <https://www.soundfield.com/#!/timeline/1971a>.
- [34] S. G. Sebastian Merchel, «Analysis and Implementation of a Stereophonic Play Back System for Adjusting the "Sweet Spot" to the Listener's Position,» Munich, 2009.
- [35] J. Daniel, «Further Study of Sound Field Coding with Higher Order Ambisonics,» Berlin, 2004.
- [36] J. Daniel, «Spatial Sound Encoding Including Near Field Effect: Introducing Distance Coding Filters and a Viable, New Ambisonic Format,» Copenhagen, 2003.
- [37] J. Daniel, «Représentation de champs acoustiques, application à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimédia,» Paris, 2001.

- [38] K. M. Bill Gardner, «HRTF Measurements of a KEMAR dummy-head Microphone,» Cambridge, 1994.
- [39] O. B. Gené, «Píldora Informativa UPM: Cámara Anecoica,» Madrid, 2012.
- [40] M. O'Connor, «Electronic Drum Advisor,» 2020. [En línea]. Available: <https://www.electronicdrumadvisor.com/es/formatos-de-plugins-diferencias-entre-vst-vst3-aax-rta-tdm/>.
- [41] P. d. Milano, *JUCE framework for VST plugins*, Milan.
- [42] «JUCE: Class List,» [En línea]. Available: <https://docs.juce.com/master/index.html>.
- [3] «Reaper: about,» [En línea]. Available: <https://www.reaper.fm/about.php>.
- [44] «Zylia Pro,» [En línea]. Available: <https://shop.zylia.co/shop/product/zylia-pro-have-it-all-216#attr=>.
- [45] K. W. N. Maluf, *An Introduction to Microelectromechanical Systems Engineering*, Artech House, 2004.
- [46] «Sennheiser AMBEO Smart Headset,» [En línea]. Available: <https://es-mx.sennheiser.com/audifonos-intraaurales-3d-audio-ambeo-smart-headset>.
- [47] J. Daniel, «Représentation de champs acoustiques, application à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimédia,» 2000. [En línea]. Available: [https://www.researchgate.net/publication/242669363\\_Representation\\_de\\_champs\\_acoustiques\\_application\\_a\\_la\\_transmission\\_et\\_a\\_la\\_reproduction\\_de\\_scenes\\_sonores\\_complexes\\_dans\\_un\\_contexte\\_multimedia](https://www.researchgate.net/publication/242669363_Representation_de_champs_acoustiques_application_a_la_transmission_et_a_la_reproduction_de_scenes_sonores_complexes_dans_un_contexte_multimedia).
- [48] F. J. C. Quirós, «Tools for Audio Object Production».
- [49] «Wikipedia, Overlap-and-Add method,» [En línea]. Available: [https://en.wikipedia.org/wiki/Overlap%E2%80%93add\\_method](https://en.wikipedia.org/wiki/Overlap%E2%80%93add_method).
- [50] M. Mena, «Statista: Spotify alcanza los 155 millones de suscriptores de pago,» [En línea]. Available: <https://es.statista.com/grafico/19793/usuarios-activos-y-de-pago-de-spotify/#:~:text=Desde%20entonces%2C%20las%20cifras%20de,respectivamente%2C%20a%20finales%20de%202020..>
- [51] A. Orus, «Evolución del número de suscriptores de Netflix en el mundo del tercer trimestre de 2011 hasta el cuarto trimestre de 2020,» [En línea]. Available: <https://es.statista.com/estadisticas/598771/numero-de-suscriptores-netflix-en-streaming-en-todo-el-mundo/#:~:text=A%20pesar%20de%20ello%2C%20el,con%20alrededor%20de%20195%20millones..>
- [52] J. Morla, «Se dispara el consumo 'online' de videojuegos,» [En línea]. Available: <https://elpais.com/cultura/2020-05-03/el-videojuego-echa-su-partida-mas-incierta.html>.
- [53] M. Mena, «Statista: Zoom dispara sus ingresos durante el coronavirus,» [En línea]. Available: <https://es.statista.com/grafico/21912/los-ingresos-de-zoom-se-disparan-con-el-coronavirus/>.
- [54] [En línea]. Available: <https://www.amazon.com/-/es/Dolby-Atmos-7-1-4-Klipsch-RP-280F/dp/B015GB31TO>.
- [55] W. H. Organization, «WHO: 1.1 billion people at risk of hearing loss,» [En línea]. Available: <https://www.who.int/mediacentre/news/releases/2015/ear-care/en/>.

- [56] «ALLPE: ¿A QUÉ EQUIVALEN LOS DIFERENTES NIVELES DE DECIBELIOS?,» [En línea]. Available: <https://www.allpe.com/acustica/ingenieria-acustica/mediciones-acusticas/a-que-equivalen-los-diferentes-niveles-de-decibelios/>.
- [57] [En línea]. Available: <https://shop.zylia.co/shop/product/zylia-pro-ambisonics-215#attr=>.
- [58] «<https://novomusica.com/tienda/imagen-sonido-profesional/sennheiser/sennheiser-ambeo-smart-headset-10802>,» [En línea].
- [59] [En línea]. Available: <https://www.apple.com/es/shop/product/FQAC2QL/A/iPhone-X-de-64-GB-reacondicionado-Gris-espacial>.
- [60] [En línea]. Available: [https://www.energysistem.com/es/music-box-1-pear-44596?AF=087836&gclid=Cj0KCQjws-OEBhCkARIsAPhOkIZOag\\_UA7eDzomtUNkZ0Sz804VfbzzIN7SHtibqXjbGcR21hjOrCmwaAq1vEALw\\_wcB](https://www.energysistem.com/es/music-box-1-pear-44596?AF=087836&gclid=Cj0KCQjws-OEBhCkARIsAPhOkIZOag_UA7eDzomtUNkZ0Sz804VfbzzIN7SHtibqXjbGcR21hjOrCmwaAq1vEALw_wcB).
- [61] «Portal web de la ETSIT UPM,» [En línea]. Available: <http://www.etsit.upm.es/>.

## ANEXO A: ASPECTOS ÉTICOS, ECONÓMICOS, SOCIALES Y AMBIENTALES

### A.1 INTRODUCCIÓN

La elección del tema del proyecto ha estado motivada por la gran presencia de contenido multimedia en la vida social y laboral. Durante el año 2020, el año de la pandemia de la Covid – 19, el aumento del uso de plataformas audiovisuales ha crecido considerablemente. La plataforma de *streaming* de música, Spotify, ha incrementado su número de usuarios en 75 millones en el último año [50]; Netflix cuenta con 195 millones de subscriptores [51]; el consumo de videojuegos online experimentó un aumento del 75 % en nuestro país [52]. Además de plataformas de ocio, también han crecido otras aplicaciones utilizadas en el mercado laboral debido a la implementación del teletrabajo: Zoom pasó de ingresar 188 M€ en el último cuatrimestre de 2019 a 883 M€ en el mismo período del año 2020 [53]. Este escenario ha influido en la decisión de construir un sistema de audio que mejore la experiencia de los usuarios.

El objetivo principal es la elaboración de una cadena de procesamiento de audio que comienza con una grabación multicanal y termina con la presentación tridimensional mediante la tecnología binaural para ofrecer al oyente una experiencia auditiva excelente. Para llevar a cabo este sistema, no se han identificado ningún inconveniente social ni legal, ya que se trata de una aplicación disponible para cualquier usuario que quiera hacer uso de ella. Al haber desarrollado el Plug – In de forma autónoma, no se han quebrantado las leyes de Propiedad Intelectual.

El uso de la tecnología Ambisonics se ha basado en la adaptación matemática entre sus directividades y las de los armónicos esféricos definidos teóricamente. En cuanto a la Recomendación de la ITU, ITU-R BS.775-3, se trata de una configuración empleable de manera libre.

Se considera que este Trabajo no entra en conflicto con ningún tema ético, económico o medioambiental. El único factor que se ha tenido en cuenta es el sanitario, ya que el uso de auriculares puede generar problemas de audición. Este impacto se desarrollará en la próxima sección.

### A.2 DESCRIPCIÓN DE IMPACTOS RELEVANTES RELACIONADOS CON EL PROYECTO

#### A.2.1. IMPACTO ECONÓMICO

En el Anexo B: Presupuesto Económico se detalla el presupuesto necesario para llevar a cabo el desarrollo de la cadena de grabación – reproducción binaural. Observamos que el equipamiento tiene un coste asumible y que la mayor inversión recae sobre la mano de obra, que sería un Ingeniero/a Junior trabajando el tiempo estimado requerido para la realización de este TFM: 600 horas.

Si, además, tenemos en cuenta el ahorro de equipamiento por parte del usuario (un equipo de reproducción 7.1.4 con un coste de unos 5000 € [54]) y el gasto de consumo energético, la inversión económica se reduciría considerablemente. Por otro lado, nuestro Plug – In podría venderse a las Estaciones de Audio Digital mediante licencias para formar parte de los paquetes de módulos enchufables predefinidos.

Llegamos a la conclusión de que nuestro sistema es asumible económicamente y que la inversión podría recuperarse en un período corto de tiempo.

## A.2.2. IMPACTO SANITARIO

Según la Organización Mundial de la Salud, más de mil millones de personas se encuentran en riesgo de sufrir pérdida auditiva debido a múltiples factores, tales como el uso prolongado de dispositivos de reproducción de sonido, los altos niveles acústicos en conciertos, clubes y grandes eventos o la larga exposición a eventos sonoros mediante escucha individual a través de audífonos [55].

El software desarrollado en este Proyecto está directamente relacionado con la escucha mediante auriculares, por lo que se ha considerado que el impacto sanitario es el más relevante de los valorados en este Anexo. El análisis detallado se ofrece en la siguiente sección.

## A.3 ANÁLISIS DETALLADO DE ALGUNO DE LOS PRINCIPALES IMPACTOS

### A.3.1. IMPACTO SANITARIO

Como introducíamos previamente, la OMS advierte de los grandes riesgos a los que la sociedad se expone debido al uso inapropiado de equipos de reproducción y a los largos tiempos de escucha sonora a altos niveles de intensidad. En un estudio elaborado en países con valores medios – altos de Renta Per Cápita se ha llegado a la conclusión de que casi el 50% de los adolescentes y jóvenes entre 12 y 35 años están expuestos a niveles inapropiados debido al uso de dispositivos personales y, de manera más alarmante, cerca del 40% lo hacen en eventos de entretenimiento sometiéndose a niveles muy dañinos para el sistema auditivo. Estos niveles pueden ser: la escucha durante 8 horas a 85 dB o a 100 dB durante 15 minutos. Por lo tanto, se recomienda evitar las situaciones en las que los niveles de sonido alcanzan dichos valores. Además, La OMS recomienda que los jóvenes protejan su audición mediante un buen control en el volumen de sus dispositivos, utilizar auriculares con cancelación de ruido y limitar el tiempo de estancia en lugares muy sonoros.

En la siguiente figura se muestra la comparativa entre valores de intensidad sonora medida en decibelios y situaciones cotidianas.



**Figura 98.** Gráfico de comparativa entre valores acústicos en dB y situaciones reales [56].

En lo relativo a nuestro Plug – In, se han tomado una serie de medidas para reducir los daños en el sistema auditivo durante el desarrollo para evitar problemas en la solución final. En el procedimiento de obtención de *HRTF* mediante la grabación de las respuestas al impulso con los auriculares *Sennheiser*, se utilizó la opción de “Cancelación de Ruido” para

evitar señales acústicas parásitas. Por otro lado, se ha llevado a cabo un proceso de normalización de las respuestas al impulso obtenidas, con el objetivo de reducir el nivel de sonido resultante tras filtrar la señal original. Asimismo, se elaboran señales de amplitud creciente para reducir el efecto del transitorio, pues se manifiesta como un sonido chirriante y molesto.

Tras tener en cuenta el impacto negativo en la salud auditiva que podría causar nuestro Plug – In y tomar las medidas para evitar daños a corto y largo plazo, se considera que el riesgo asumido es bajo gracias a las soluciones adoptadas.

## A.4 CONCLUSIONES

Finalizamos este Anexo con la síntesis de los aspectos éticos, sociales, económicos, sanitarios y medioambientales involucrados en este proyecto. Tal y como se ha explicado, no se han considerado relevantes aquellos relacionados con el medio ambiente o con la ética, pues en ningún caso este Trabajo perjudica a la naturaleza mediante su implantación o utilización; así como tampoco quiebra ningún valor moral.

En lo social, somos conscientes de que puede tener un impacto positivo, pues mejorará la experiencia de los usuarios en los ámbitos lúdicos y profesionales. Económicamente, se detalla que la inversión es asequible y que puede verse recuperada en un período de tiempo bajo.

Por último, asumimos que el mayor impacto es aquél relacionado con el daño auditivo debido al uso de nuestro software. Sin embargo, se han tenido en cuenta los problemas que se pueden ocasionar y se han elaborado soluciones al respecto, de manera que se considera que se ha aportado un valor añadido al proyecto, anticipándonos a los inconvenientes que pueden causarse en el usuario.

Con todo ello, concluimos que este Trabajo de Fin de Máster no quiebra ningún valor ético, así como tampoco influye negativamente en el resto de los aspectos que se tienen en cuenta en este Anexo.

## ANEXO B: PRESUPUESTO ECONÓMICO

El presupuesto económico estimado para la realización de este Proyecto se adjunta en la siguiente tabla:

|                                                |                  |              |                  |          |
|------------------------------------------------|------------------|--------------|------------------|----------|
|                                                | Horas            | Precio/hora  | TOTAL            |          |
| COSTE DE MANO DE OBRA (coste directo)          | 600              | 30 €         | 18.000 €         |          |
| COSTE DE RECURSOS MATERIALES (coste directo)   |                  |              |                  |          |
|                                                | Precio de compra | Uso en meses | Amortiz. en años | TOTAL    |
| Ordenador personal                             | 700,00 €         | 6            | 5                | 70,00 €  |
| Software (Adobe Audition)                      | 228,00 €         | 6            | 5                | 22,80 €  |
| Hardware (Zylia Pro Ambisonics) [57]           | 950,00 €         | 6            | 5                | 95,00 €  |
| Hardware (Sennheiser Ambeo Smart Headset) [58] | 299,00 €         | 6            | 5                | 29,90 €  |
| iPhone X 64 GB [59]                            | 629,00 €         | 6            | 5                | 62,90 €  |
| Altavoces Energy Sistem Music Box 1+ [60]      | 20 €             | 6            | 5                | 2,00 €   |
| TOTAL                                          |                  |              |                  | 282,60 € |
| GASTOS GENERALES (costes indirectos)           | 15%              | sobre CD     | 2.742,39 €       |          |
| BENEFICIO INDUSTRIAL                           | 6%               | sobre CD+CI  | 1.261,50 €       |          |
| MATERIAL FUNGIBLE                              |                  |              |                  |          |
| Ratón                                          |                  |              | 20,00 €          |          |
| SUBTOTAL PRESUPUESTO                           |                  |              | 22.306,49 €      |          |
| IVA APLICABLE                                  |                  | 21%          | 4.684,36 €       |          |
| TOTAL PRESUPUESTO                              |                  |              | 26.990,85 €      |          |

Tabla 27. Presupuesto económico del Proyecto.