



Universidad de Oviedo

Memoria del Trabajo Fin de Máster realizado por

IGNACIO MARQUÉS FERNÁNDEZ

para la obtención del título de

Máster en Ingeniería de Automatización e Informática Industrial

**SIMULACIÓN DE ESPACIOS SONOROS PARA EL
MANTENIMIENTO Y SUPERVISIÓN DE
INSTALACIONES INDUSTRIALES Y MÁQUINAS**

JULIO 2019



RESUMEN

El trabajo presente describe la aplicación de una técnica basada en la utilización de la respuesta a impulso de la cabeza (HRIR, del inglés Head Related Impulse Response) para conseguir la recreación de ambientes sonoros mediante fuentes de sonido virtuales. El objetivo es estudiar y determinar en la medida de lo posible su utilidad de cara a una posible aplicación en entornos industriales, en los que podría usarse como una herramienta de supervisión a distancia para la detección de fallas en máquinas.

Índice

1. Introducción.....	5
2. Fundamentos teóricos	8
2.1.- Conceptos matemáticos.....	8
2.1.1.- Respuesta a impulso y convolución.....	8
2.1.2.- Coordenadas esféricas.....	10
2.2.- La audición humana	12
2.2.1.- Diferencia de tiempo interaural (ITD).....	13
2.2.2.- Diferencia de volumen interaural (ILD)	13
2.2.3.- Función de transferencia relacionada con la cabeza (HRTF).....	13
3. Objetivos y alcance.....	15
4. Metodología.....	16
4.1.- Tareas y herramientas.....	17
4.2.- Base de datos	19
4.3.- Técnica de interpolación tetraédrica	20
4.4.- Organización de clases, métodos y atributos	24
4.4.1.- Clase Emisor	25
4.4.2.- Clase Receptor	26
4.4.3.- Clase Física	26
4.4.4.- Clase GestionHRIR.....	28
4.4.5.- Clase GestionAudio	31
4.4.6.- Clase Procesamiento	32
4.4.7.- Otras clases y funciones.....	34
5. Resultados y experimentos	35
5.1.- Análisis objetivo de los resultados	35
5.2.- Análisis subjetivo de los resultados, encuesta.....	45
5.2.1.- Primera pregunta, senoidal por el oído derecho	45
5.2.2.- Segunda pregunta, ruido mecánico por detrás	46
5.2.3.- Tercera pregunta, ruido de taladro por delante	47
5.2.4.- Cuarta pregunta, senoidal por el oído izquierdo	48

5.2.5.-	Quinta pregunta, taladro por la izquierda y ruido mecánico por la derecha 49	
5.2.6.-	Sexta pregunta, taladro por detrás y ruido mecánico por delante.....	50
5.2.7.-	Séptima pregunta, taladro por debajo de la línea de los oídos y ruido mecánico por encima.....	51
5.2.8.-	Octava pregunta, opiniones	52
6.	Conclusión y trabajos futuros.....	54
7.	Referencias	57

1. Introducción

“Cuando tenía diez años presencié un accidente entre dos coches. El choque ocurrió detrás de mí, muy cerca, y me salvé porque pude localizar el ruido [1]. Esa fue la primera vez que me interesé por la procedencia del sonido. Empecé a preguntarme cómo era posible que el ser humano posicionase el sonido en cualquier parte con sus dos oídos”

Estas palabras pertenecen a Hugo Zucarelli, el padre de la holofonía, una técnica de audio 3D que impulsó el estudio de la espacialización del sonido allá por los años 70. El objetivo de Zucarelli no era otro que el de conseguir un formato de audio que, escuchado a través de auriculares o altavoces, permitiera al oyente diferenciar de dónde proceden los diferentes sonidos.

Sin embargo, la tecnología desarrollada por Zucarelli [2], a pesar de haber sido utilizada en la industria musical por artistas y grupos como Pink Floyd, quedó olvidada principalmente por dos motivos: la aparición de otras tecnologías de audio como dolby surround y la falta de rigor científico de sus explicaciones, ya que solo se apoyó en experiencias subjetivas para defender la validez de su trabajo.

No obstante, Zucarelli sentó las bases y propuso un método de medición que serviría poco después para tomar datos del sistema auditivo humano de manera muy precisa. Utilizó un maniquí con micrófonos en los oídos y un tímpano artificial para recoger la información que necesitaba para desarrollar su tecnología. Esta técnica de medición se conoce como “Dummy Head”.

Gracias al Dummy Head y otros métodos científicos, a día de hoy es relativamente fácil encontrar bases de datos [3][4][5] con la información necesaria para procesar audio en 3D, lo que ha impulsado el desarrollo del sonido binaural. Sin embargo, las técnicas de audio tridimensional han pasado desapercibidas, eclipsadas por otras técnicas como el sonido surround o el estéreo.

El sonido surround consiste en utilizar una serie de altavoces dispuestos alrededor del oyente, de tal forma que se intenta imitar la posición de la que vienen diferentes

sonidos en el mundo físico, gracias a la ubicación de los altavoces en el espacio. Esta tecnología se encuentra presente, por ejemplo, en las salas de cine.

El sonido estéreo, por su parte, emplea dos canales (izquierdo y derecho), para enviar distinta información a cada oreja. Ha sido y sigue siendo ampliamente utilizado en la industria musical. Los álbumes de los Beatles son un gran ejemplo de este tipo de tecnología, ya que, si se escuchan con cascos o a través de un buen equipo de altavoces, se puede discernir claramente de qué canal, izquierdo o derecho, procede el sonido de cada instrumento.

Es importante aclarar que ni el surround ni el estéreo son técnicas de audio 3D, ya que la ubicación de la que procede el sonido está limitada por la posición real del altavoz o auricular. Además, estas técnicas no tienen en cuenta otros factores que se tratarán más adelante, como la distancia entre las dos orejas ni la distancia desde la fuente de sonido al oyente.

La marginación del audio 3D durante décadas, puede haberse visto condicionada por la dificultad de la grabación de audio en 3D, la limitación de los equipos de audio (sobre todo entre los consumidores particulares), o la apuesta de grandes multinacionales por tecnologías como el sonido surround.

Sin embargo, recientemente el audio 3D ha recobrado fuerza y tomado un nuevo impulso. Esta vez bajo el nombre de audio 8D (en este caso parece que la “D” se refiere a direcciones y no dimensiones), que se ha popularizado gracias a fenómenos virales en la red como la canción “Dilbar” de Satvameva Jayate o los videos del youtuber Jaime Altozano [6].

Por tanto, la diferencia entre audio 3D, 8D o binaural no parece clara. Y en realidad no parece clara porque, si existe, no es mucha. Algunos expertos señalan que el audio 3D o binaural se origina por la grabación in situ de los sonidos, es decir, siguiendo un proceso de grabación en 3D con equipos específicos en un ambiente controlado y tratado acústicamente. Por otro lado, el audio 8D resulta del procesamiento a posteriori de una señal de audio cualquiera de una manera diversa para cada oído, permitiendo generar la sensación de que se ubica en un sitio determinado. En este documento, se referirá a audio 3D a cualquier tipo de audio utilizado para crear una sensación de audio espacial.

A día de hoy, el audio 3D es utilizado en videojuegos, música, cine, realidad virtual y otros sectores. Pero, ¿podría aplicarse en otros sectores? ¿podría ser útil también, como una herramienta de supervisión a distancia en plantas industriales? ¿Mejoraría acaso la detección de errores en diversas máquinas? ¿Es viable su utilidad con este propósito? Para resolver estas dudas, el trabajo presente describe las partes más importante del desarrollo de una aplicación de escucha en 3D orientada a la supervisión de máquinas industriales. Se pretende proyectar una solución a este problema y descubrir si los resultados son indicadores de la viabilidad de esta herramienta.

2. Fundamentos teóricos

2.1.- CONCEPTOS MATEMÁTICOS

Antes de explicar los fundamentos de la audición humana, conviene tratar algunos conceptos que serán de utilidad en el desarrollo de este trabajo. Se explicará brevemente qué es la respuesta a impulso y la convolución, así como la representación espacial mediante coordenadas esféricas.

2.1.1.- Respuesta a impulso y convolución

En el procesamiento digital de señal (DSP), un impulso es una señal que solo contiene un punto que no sea cero, mientras que el resto de puntos son ceros [7]. Si la muestra número 0 es dicho punto de valor distinto de cero, la señal resultante se llamará función delta.

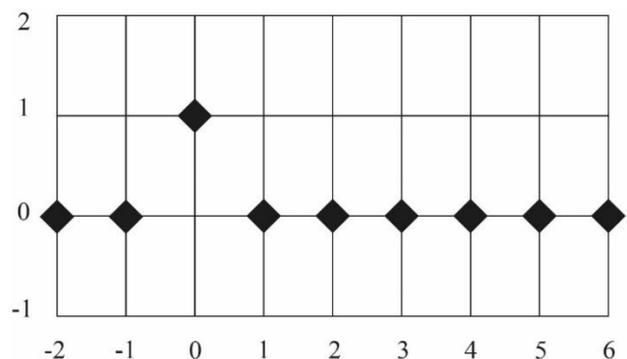


Figura 2.1.- Función Delta.

La respuesta a impulso de un sistema es la señal de salida obtenida como consecuencia de una señal de entrada de la función delta (figura 2.2).

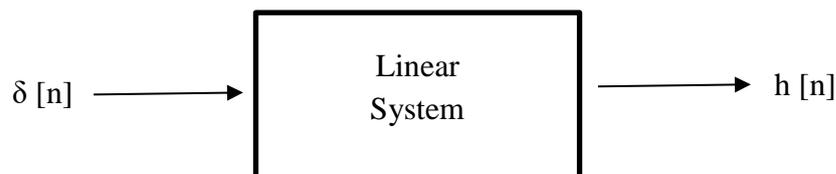


Figura 2.2.- La letra griega δ representa la función delta, mientras que la letra h es usada para representar la respuesta a impulso.

La importancia de la respuesta a impulso viene dada por su capacidad de caracterizar completamente un sistema lineal. Por lo tanto, si se conoce la respuesta a impulso de un sistema se puede calcular la señal de salida ante cualquier entrada. Para calcular dicha señal de salida, es necesario realizar una operación matemática llamada convolución.

La convolución [7] es la operación más importante en DSP. Toma dos señales y las combina de manera que produce una tercera. En notación DSP, una señal de entrada se representa normalmente por la letra x , mientras que la señal de salida se representa por y . (La h representa la respuesta a impulso.) Por tanto, en forma de ecuación, la convolución se expresa como:

$$x[n]*h[n]=y[n] \quad (2.1)$$

El operador de convolución es el asterisco, "*", así como "+" representa la suma o "×" representa la multiplicación. También puede ser representado por medio de un círculo con una cruz en el interior, "⊗".

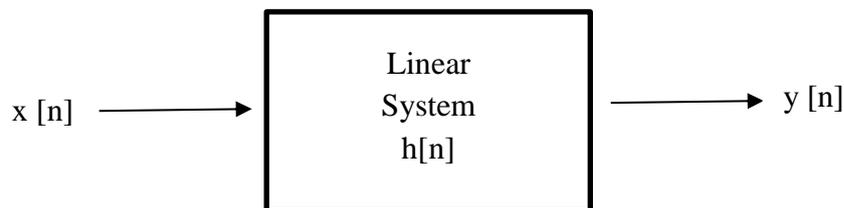


Figura 2.3.- A la señal $x[n]$, se le aplica una convolución con la respuesta a impulso característica del sistema, $h[n]$, con el fin de obtener una señal de salida, $y[n]$.

Matemáticamente, la convolución puede expresarse mediante la ecuación 2.2. Esta ecuación se denomina suma de convolución y muestra los cálculos necesarios para obtener cada punto de la señal de salida a partir de la utilización de la respuesta a impulso y las muestras correspondientes de la señal de entrada.

$$y[i] = \sum_{j=0}^{M-1} h[j]x[i - j] \quad (2.2)$$

Otra ventaja de la convolución es la propiedad algebraica de la asociatividad. En la práctica, esto significa que dos o más respuestas de impulso diferentes se podrían juntar mediante una convolución en una nueva respuesta cuya convolución con una señal de entrada daría lugar a una señal de salida igual a la misma señal lograda al realizar las convoluciones por separado (figura 2.4). Esto permite el cálculo de la señal de salida en menos tiempo, lo que es un factor crítico en aplicaciones en tiempo real.

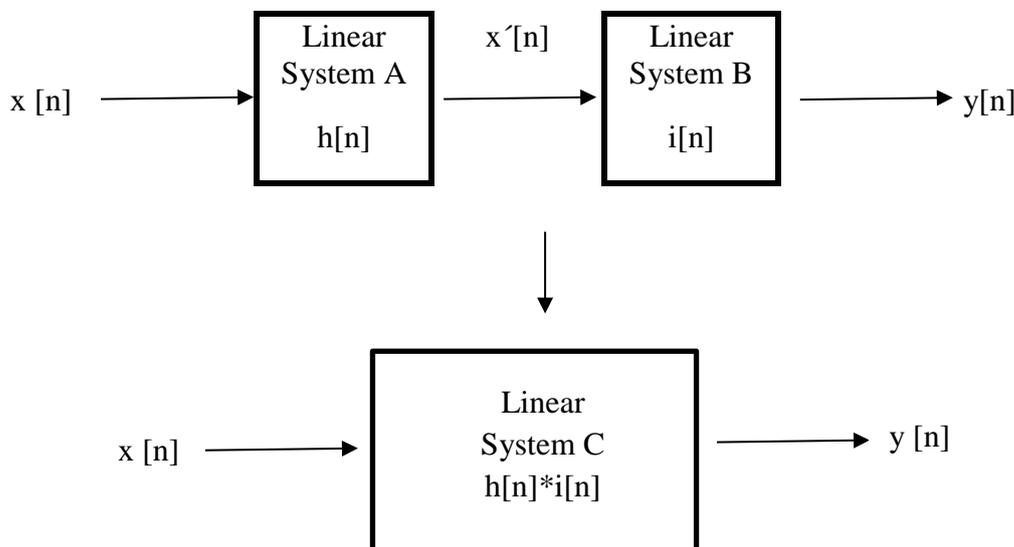


Figura 2.4.- Representación de la propiedad asociativa de la convolución. Un sistema lineal ficticio C es creado a partir de la convolución de $h[n]$ y $i[n]$. Esto permite una velocidad de cómputo superior.

2.1.2.- Coordenadas esféricas

Las coordenadas esféricas constituyen un sistema de representación que permite definir puntos en el espacio gracias a una distancia y dos ángulos [8]. Por tanto, un punto P puede representarse por medio de un radio r , el ángulo polar θ y el azimutal ϕ .

- El radio es la magnitud de la distancia desde el origen de coordenadas al punto P.

- El ángulo polar tiene un rango de 180° . Se puede definir de 0° a 180° , o bien de -90° a 90° dependiendo del autor. La convención internacional lo define de 0° a 180° pero en todos los casos define la rotación en torno al eje x.

- El azimuth, por otro lado, tiene un rango de 0° a 360° y puede definirse tanto de -180° a 180° como de 0° a 360° , como es el caso de la convención internacional. Define la rotación en torno al eje z.

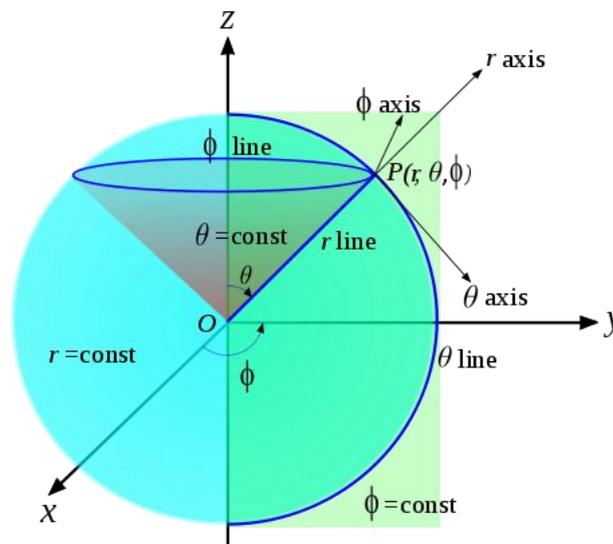


Figura 2.5.- Relación de las coordenadas esféricas y las cartesianas.

Es posible cambiar de coordenadas esféricas a cartesianas:

$$x = r \sin \theta \cos \varphi \quad (2.3)$$

$$y = r \sin \theta \sin \varphi$$

$$z = r \cos \theta$$

Así como de cartesianas a esféricas

$$r = \sqrt{x^2 + y^2 + z^2} \quad (2.4)$$

$$\theta = \arctan \left(\frac{\sqrt{x^2 + y^2}}{z} \right) \text{ si } z > 0$$

$$\theta = \pi + \arctan\left(\frac{\sqrt{x^2 + y^2}}{z}\right) \text{ si } z < 0$$

$$\theta = \frac{\pi}{2} \text{ si } z = 0$$

$$\varphi = \arctan\left(\frac{y}{x}\right) \text{ si } x > 0 \text{ e } y > 0$$

$$\varphi = 2\pi + \arctan\left(\frac{y}{x}\right) \text{ si } x > 0 \text{ e } y < 0$$

$$\varphi = \pi + \arctan\left(\frac{y}{x}\right) \text{ si } x < 0$$

De ahora en Adelante, se dará por hecho que las posiciones de las HRIR están definidas en coordenadas esféricas. Además, para mantener la concordancia de la convención propuesta por los autores de la base de datos de HRIR que se usará, y a pesar de la convención internacional, las posiciones constarán de las tres coordenadas citadas en el siguiente orden: (*azimuth, elevación, distancia*).

2.2.- LA AUDICIÓN HUMANA

Tras miles de años de evolución, el ser humano ha adquirido un sistema de percepción sonora que le permite ubicar y detectar frecuencias con una precisión bastante acertada. La supervivencia de la especie ha siempre estado supeditada al desarrollo de los sentidos, y junto con la vista, el oído se erige como uno de los principales contactos con la realidad exterior. Aunque en muchos casos prevalece la información visual, el ser humano es capaz de saber con bastante precisión si una fuente sonora se encuentra cerca, a qué oreja se aproxima más, o cuál es su intensidad.

Existen diversos fenómenos físicos que afectan a la audición humana y su capacidad de ubicar los sonidos en el espacio. De conocerlos, ¿sería posible crear fuentes de sonido virtuales capaces de engañar al oído? La respuesta es sí, ya que al igual que muchos otros fenómenos, la audición humana se puede modelar de manera matemática como un sistema definido por una función de transferencia o una respuesta a impulso. A continuación, se clasifican y explican estos factores [9].

2.2.1.- Diferencia de tiempo interaural (ITD)

Se trata de la diferencia temporal que existe entre la llegada del sonido a un oído respecto al otro. Esta diferencia es del orden de milisegundos, pero es una referencia importante que permite al cerebro facilitar la tarea de ubicar el sonido en el espacio. Dado que el sonido viaja a 340 m/s, y la distancia de oreja a oreja es de unos 20 cm, para una fuente sonora ubicada justo en frente a una de las orejas, la diferencia de tiempo interaural será de unos 0,59 ms. Sin embargo, esta cifra no tiene en cuenta la propagación del sonido a través de la cabeza ni las difracciones que en ella se pueden producir. La diferencia de tiempo interaural va disminuyendo conforme la fuente de sonido se desplaza hacia cualquier punto situado a la misma distancia de las dos orejas, donde toma el valor 0.

2.2.2.- Diferencia de volumen interaural (ILD)

Se trata simplemente de la diferencia de intensidad percibida en un oído respecto al otro. Por ejemplo, un susurro en uno de los oídos no se percibe de la misma forma en ese mismo oído que en el opuesto, ya que el primero detectará una amplitud mucho mayor. Se puede decir de manera generalista que la diferencia de volumen interaural se reduce conforme la fuente de sonido se desplaza hacia cualquier punto situado a la misma distancia de las dos orejas, aunque los puntos de ILD cero dependen más bien de las condiciones auditivas del individuo y no tanto de la distancia a la fuente de sonido.

2.2.3.- Función de transferencia relacionada con la cabeza (HRTF)

A pesar de su nombre, la función de transferencia relacionada con la cabeza (de ahora en adelante se denominará HRTF, proveniente de las siglas en inglés, *Head Related Transfer Function*) recoge la información de cómo el torso superior, cabeza, cuello y orejas responden a la información sonora. Se trata de un factor clave en la audición que permite facilitar la ubicación del sonido ya sea delante o detrás del individuo y su elevación. Gracias a la HRTF se puede adivinar la posición de la que procede un sonido a pesar de que su ITD y su ILD sea cero, ya que la respuesta en frecuencia de nuestro sistema auditivo varía en base a la ubicación de la fuente sonora y estas diferencias se hacen perceptibles gracias a los aspectos de nuestro físico ya citados, especialmente gracias a la forma de nuestro pabellón auricular. La transformada inversa de Fourier de la HRTF da lugar a la respuesta impulsional relacionada con la cabeza, que de ahora en

adelante se denominará HRIR (debió a sus siglas en inglés de *Head Related Impulse Response*) y será de fundamental importancia en este trabajo.

3. Objetivos y alcance

El objetivo de este trabajo es, como ya se ha mencionado, desarrollar una aplicación capaz de crear audio 3D a partir de algoritmos que utilicen la respuesta a impulso relacionada con la cabeza. Esta técnica, es conocida como audio 8D por ciertos autores como se ha visto. La aplicación debería de ser capaz de ubicar las fuentes de sonido en un espacio cartesiano y asociarles el sonido captado por un micrófono que capte el ruido de una máquina en tiempo real. Como no existe la posibilidad de hacer esto último y se trabaja en una fase de desarrollo, se asignará un archivo de audio o señal sintética a cada emisor (por ejemplo, una onda senoidal generada a partir de las funciones matemáticas del lenguaje de programación utilizado o un archivo .mp3). A continuación, el programa debe calcular la posición relativa de estas fuentes de audio respecto a la posición del receptor, y devolverlas en forma de coordenadas esféricas con centro en la cabeza del oyente. Una vez obtenidas las posiciones respecto al oyente, se deben conseguir las respuestas a impulso HRIR asociadas a dichas posiciones para procesar el audio. Finalmente, los datos obtenidos se someterán a un análisis, tanto objetivo como subjetivo. Este último a través de una encuesta.

Se considerará que los resultados son satisfactorios, si el análisis objetivo responde a grandes rasgos a los criterios esperados. (Para este proyecto, no se ha dispuesto de las herramientas adecuadas para hacer una valoración experimental, ya que sería necesaria una cámara anecoica y demás equipo especializado.) Por otro lado, el cuestionario puede servir de ayuda para dilucidar hasta qué punto es útil y/o rentable aplicar el software resultante en la industria, como herramienta de supervisión.

4. Metodología

Tras un estudio en profundidad de los requisitos técnicos necesarios para realizar este trabajo, se definió el siguiente plan de acción mostrado en el diagrama de la figura:

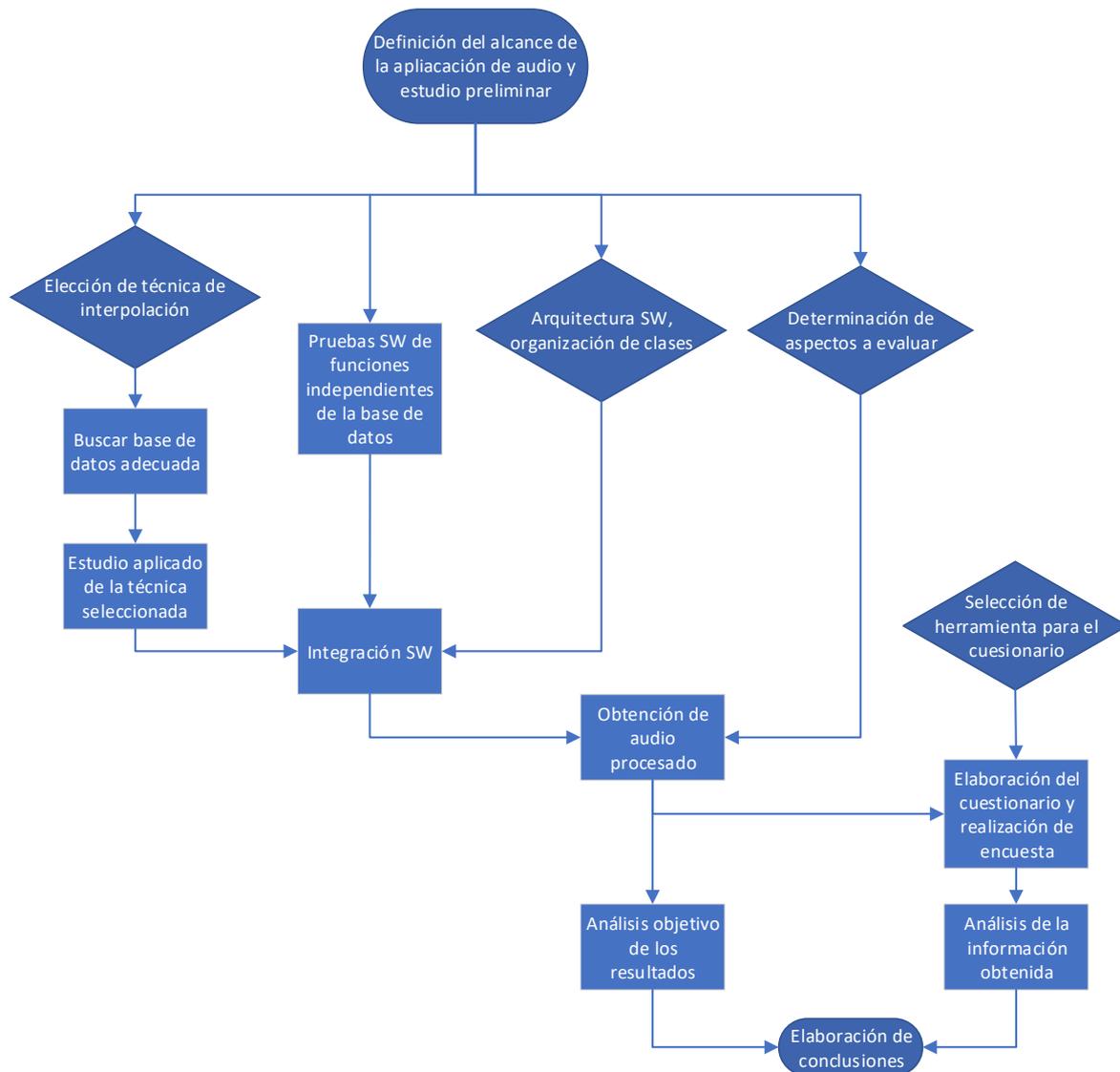


Figure 4.1.- Relación de las tareas planificadas.

El diagrama muestra las diferentes tareas realizadas y su dependencia entre ellas. Así mismo, los contenedores representan, dependiendo de su forma, si la tarea comprende un proceso (contenedor rectangular), una decisión (contenedor con forma de rombo) o si es el fin o inicio del proyecto (contenedor redondeado). Los contenedores a los que no les llega

ninguna flecha representan tareas independientes, mientras que los contenedores a los que les llega una o más flechas, representan tareas que dependen de una o más de una tarea respectivamente. Por ejemplo, la tarea *obtención de audio procesado* depende de que se haya completado la tarea *integración de SW* y la tarea de *determinación de aspectos a evaluar*. Por otro lado, la tarea *selección de herramienta para el cuestionario* se considera independiente de todas las demás, pudiendo llevarse a cabo en cualquier momento, a pesar del estado del desarrollo del software.

En el siguiente apartado, se detallan todas las tareas y se relacionan con la herramienta utilizada para llevarla a cabo.

4.1.- TAREAS Y HERRAMIENTAS

Definición del alcance de la aplicación de audio y estudio preliminar: Para realizar esta tarea se emplearon diferentes artículos científicos con el objetivo de conocer los detalles técnicos del desarrollo del proyecto [3][4][5][6][10] y se llevó a cabo un estudio para determinar qué lenguaje de programación y qué librerías eran las más adecuadas. Esta tarea asienta las bases de lo que se hizo a continuación.

Elección de técnica de interpolación: El estudio preliminar dio lugar a esta tarea, que supuso el estudio de diferentes técnicas de interpolación de la HRIR para dar con la más adecuada y aplicarla. La técnica de interpolación seleccionada se argumenta y explica en el apartado 4.3.

Búsqueda de base de datos adecuada: La selección de la técnica de interpolación determinó el uso de la base de datos, ya que, para implementar una técnica de interpolación tetraédrica, no sirve una base de datos con respuestas a impulso en un único plano, sino que hacen falta muchos puntos susceptibles de ser triangulados dando lugar a trapezoides. La información contenida en la base de datos se explica en detalle en la sección 4.2.

Estudio aplicado de la técnica seleccionada: Se desarrollaron las funciones software que conforman el grueso de la técnica de interpolación. Para ello, se utilizó como base una librería de funciones de interpolación tetraédrica en lenguaje Matlab [13], que fue adaptada a la sintaxis de Python. Se utilizó el pack “anaconda”, en el que está incluido la plataforma

Spyder para el desarrollo en lenguaje Python. Se verificó que el resultado de las funciones desarrolladas fuera coincidente con el de las funciones de la librería de Matlab.

Pruebas software de funciones independientes de la base de datos: A parte de las funciones relacionadas con la base de datos y la técnica de interpolación tetraédrica, fue posible llevar a cabo en paralelo otro tipo de pruebas, especialmente orientadas al procesamiento de señales. Todas estas pruebas se llevaron a cabo en Python, aunque se verificó su correcto funcionamiento cotejando los resultados con los obtenidos por funciones “built-in” de Matlab.

Arquitectura software, organización de clases: Esta tarea supuso el ordenamiento y estructuración del código desarrollado, en diferentes módulos lo más independientes unos de otros posible, dando lugar a diferentes clases en lenguaje Python. Esta tarea se llevó a cabo sobre el papel, sin programar. A pesar de la susceptibilidad de pequeñas modificaciones, modularizar el código sobre el papel supuso un muy buen punto de partida para la programación ulterior. La organización de clases es explicada en detalle en el apartado 4.4.

Integración software: Una vez las diferentes pruebas fueron realizadas, así como la organización de clases fue establecida, se comenzó a trabajar en un script en el que se reunió todo el código desarrollado. A partir de entonces, se determinaron los mecanismos de comunicación entre las clases y se desarrollaron las funciones faltantes para garantizar un procesamiento de audio adecuado y sencillo. Esta tarea se desarrolló en Spyder, que como ya se ha dicho, fue la plataforma elegida para realizar la programación en Python,

Determinación de aspectos a evaluar: Se estudiaron diferentes opciones para determinar el correcto funcionamiento del Software. Esta decisión determinó los resultados obtenidos, es decir, la manera de procesar el audio que resultaría susceptible de análisis.

Obtención de audio procesado: Se programaron las posiciones y se seleccionaron los sonidos a procesar de acuerdo con la tarea precedente, y se programaron diferentes escenarios con uno o más de un emisor en diferentes posiciones. En este caso, se utilizó también la plataforma de programación jupyter notebook, que permite evaluar secciones de código de manera independiente y resulta muy práctico para la obtención y visualización de resultados.

Selección de herramienta para el cuestionario: Se estudió la manera más cómoda para hacer llegar la encuesta de la manera más simple al mayor número de personas posible. Finalmente se eligió la herramienta de *formularios de Google*.

Elaboración de cuestionario y realización de encuesta: Tras seleccionar la plataforma del cuestionario más conveniente, se realizó un formulario integrando las preguntas relacionadas con los aspectos a evaluar junto con el audio procesado resultante.

Análisis objetivo de los resultados: Gracias a la visualización de datos, fue posible verificar de manera gráfica la forma de onda percibida por nuestros oídos, con el objetivo de no lanzar conclusiones equivocadas fruto de un posible “efecto placebo” basado en escuchar lo que se desea. El análisis fue posible gracias a la representación gráfica de los archivos de audio usando Matlab.

Análisis de la información obtenida del cuestionario: Una vez el cuestionario fue propuesto a una muestra poblacional de tamaño considerable, se realizaron gráficas para determinar los puntos fuertes y débiles de la aplicación y tomar en consideración aspectos subjetivos de la aplicación desarrollada.

Elaboración de conclusiones: Juntando los análisis realizados hasta el momento, se llegó a las conclusiones expuestas en el apartado 6 del presente documento.

4.2.- BASE DE DATOS

La base de datos utilizada conocida como PKU&IOA [3], contiene 6344 HRIR tomadas a diferentes distancias, de 20 cm a 160 cm; grados, de 0° a 360°; y elevaciones, de -40 a 90°. En el siguiente cuadro se precisan las coordenadas de los puntos usados en base a la elevación y el paso en el azimuth, es decir, los grados de separación entre las sucesivas mediciones.

Elevacion (grados)	-40	-30	-20	-10	0	10	20	30	40	50	60	70	80	90
Paso (grados)	5	5	5	5	5	5	5	5	5	5	10	15	30	360
Numero	72	72	72	72	72	72	72	72	72	72	36	24	12	1

Tabla 4.1.- Mediciones de la HRIR para cada distancia de 20 cm a 160 cm.

Para cada distancia al centro de la cabeza (20, 30, 40, 50, 75, 100, 130, 160 cm), se tomaron mediciones de acuerdo al cuadro superior. Por tanto, para cada distancia o radio, la HRIR fue obtenida para 793 ubicaciones. Se obtiene de esta forma una base de datos muy completa para la aplicación presente.

Es importante tener en cuenta la convención empleada por los autores de esta base de datos, ya que las ecuaciones de conversión de coordenadas esféricas a cartesianas y viceversa serán ligeramente diferentes a las descritas en el apartado de base teórica, en el que se mostraban las ecuaciones en caso de usar la convención internacional.

Además, la base de datos contiene información sobre la ITD y la ILD, que se haya presente en las mediciones de la HRIR de manera implícita. Por ejemplo, en la siguiente imagen de dos HRIR asociadas a posiciones diversas, se puede comprobar que el transitorio comienza en muestras diferentes, y que la intensidad es menor cuanto mayor es la distancia de la fuente al centro de la cabeza.

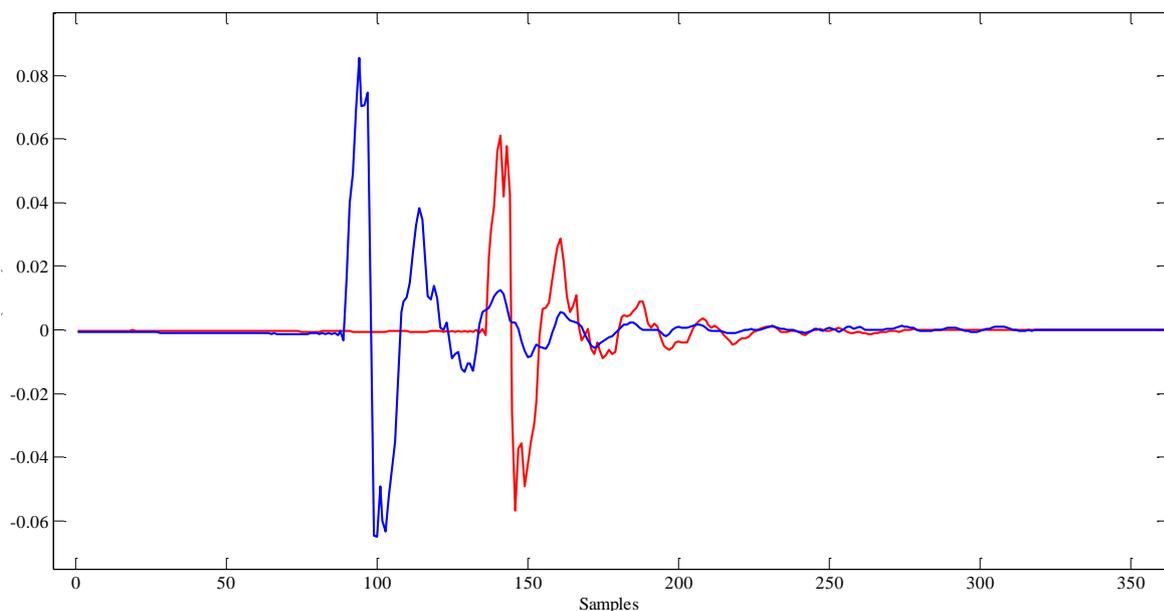


Figura 4.2.- En color azul, HRIR asociada a la posición (35, 40, 50); en color rojo, HRIR asociada a la posición (35, 40, 75).

4.3.- TÉCNICA DE INTERPOLACIÓN TETRAÉDRICA

Además de la base de datos, ha sido necesario utilizar una técnica de interpolación adecuada a las respuestas a impulso. Muchas veces, no se dispone de la respuesta impulsional exacta a un punto ubicado en una posición determinada en torno a la cabeza del oyente y es necesario inferirla de algún modo. La interpolación de este tipo de respuestas es objeto de estudio científico, ya que no siempre guardan una relación sencilla entre ellas.

Existen varios métodos propuestos, y tras un estudio de las diferentes posibilidades, se ha optado por la utilización de una técnica de interpolación tetraédrica. De acuerdo con [10], la comparación de los métodos de interpolación bilineal rectangular, interpolación bilineal triangular e interpolación tetraédrica dio lugar a los siguientes resultados en término de error MSE:

	Tipo de error	Rectangular	Triangular	Tetraédrica
Az. - Ele.	MSE HRIR	7,13%	7,26%	3,72%
Dist. - Ele.	MSE HRIR	7,69%	9,52%	3,72%
Dist. - Az.	MSE HRIR	4,34%	9,52%	3,72%

Tabla 4.2.- MSE de tres técnicas de interpolación de HRIR existentes.

El MSE es el mínimo error cuadrado, típicamente utilizado para la verificación de estos métodos de interpolación. Se calcula de la siguiente forma:

$$MSE = \frac{|h_o - h_i|^2}{h_0^2} \quad (4.1)$$

Donde h_0 es la HRIR real medida correspondiente a una posición, mientras que h_i es la respuesta interpolada para esa misma posición.

Teniendo en cuenta que el MSE proporcionado por la interpolación tetraédrica supone el menor valor, se ha optado por esta técnica [11], que tiene en cuenta la HRIR de hasta cuatro puntos para obtener la HRIR asociada a un punto ubicado en el interior del tetraedro formado por dichos puntos. Se ha utilizado una librería en Matlab [13] para conseguir tal efecto, aunque ha sido necesario traducir las funciones al lenguaje Python para

incorporarlo en la aplicación. Merece la pena describir en que consiste el algoritmo, ya que supone una parte fundamental de este trabajo.

El algoritmo parte del uso de una base de datos 3D. El primer paso consiste en triangular la ubicación de los puntos para los que se cuenta con una HRIR. La triangulación de Delaunay es adecuada para tal efecto.

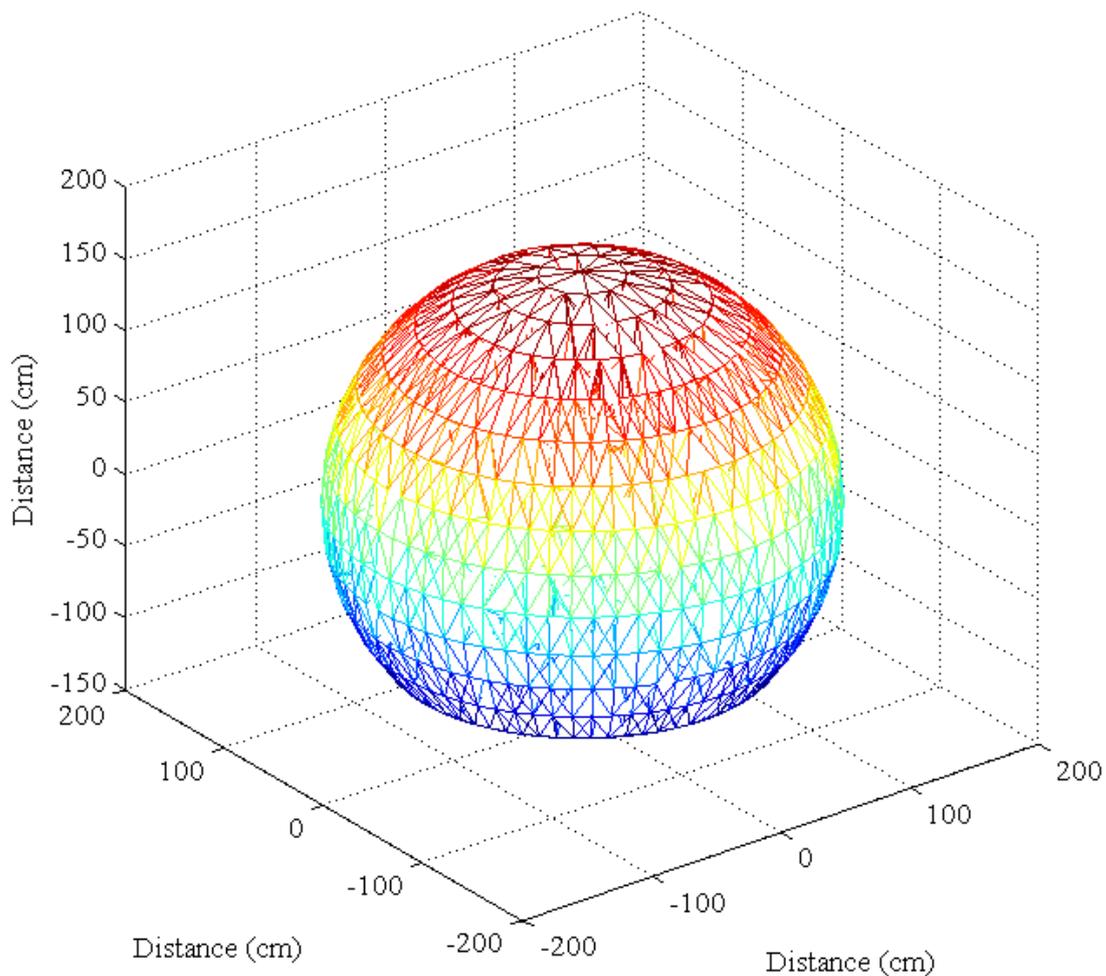


Figura 4.3.- Triangulación de Delaunay.

Se obtiene de esta forma una rejilla de puntos formada por tetraedros en la que cada punto solo puede estar contenido en un único tetraedro, excepto si el punto reside en un lado, vértice o cara compartida por algún tetraedro. Una vez la rejilla tetraédrica es calculada, la HRIR estimada de un punto X ubicado dentro de la rejilla, puede ser calculada gracias a la interpolación de las HRIR localizadas en los vértices de un tetraedro que contiene a X.

Considerando un tetraedro de vértices A, B, C y D, la HRIR de cualquier punto X ubicado en el interior del tetraedro puede ser representada como una combinación lineal de los vértices

$$X = g_1A + g_2B + g_3C + g_4D \quad (4.2)$$

Donde g_i son pesos escalares. Además, se debe cumplir que

$$\sum_{i=1}^4 g_i = 1 \quad (4.3)$$

Los pesos son equivalentes a las coordenadas bariocéntricas del tetraedro y pueden ser utilizados directamente para calcular la HRIR del punto X.

$$H_x = \sum_{i=1}^4 g_i H_i \quad (4.4)$$

Sustrayendo D de la ecuación 4.2

$$X - D = [g_1 \ g_2 \ g_3]T \quad (4.5)$$

Donde T:

$$T = \begin{bmatrix} A - D \\ B - D \\ C - D \end{bmatrix} \quad (4.6)$$

Por tanto, los pesos de interpolación se pueden obtener de la siguiente manera

$$[g_1 \ g_2 \ g_3] = (X - D)T^{-1} \quad (4.7)$$

Mientras que g_4 se obtiene a partir de la ecuación 4.4 cuando el resto de pesos ya han sido calculados:

$$g_4 = 1 - g_1 - g_2 - g_3 \quad (4.8)$$

Como la matriz T depende solamente de la geometría del tetraedro, la matriz T^{-1} puede ser calculada para cada tetraedro durante la inicialización, pudiendo ser almacenada en la memoria para reducir el tiempo de ejecución más adelante.

Por otra parte, si se cuenta con una base de datos de muchos puntos (como es el caso), estos se triangularán dando lugar a una gran cantidad de tetraedros. Esto hace que a la hora de buscar un tetraedro que contenga al punto deseado X , se emplee demasiado tiempo si se usara un algoritmo de fuerza bruta. El algoritmo de fuerza bruta calcularía las coordenadas baricéntricas del punto respecto a cada tetraedro hasta encontrar uno en que los pesos obtenidos (es decir, los valores de dichas coordenadas), sean positivos.

Una manera más eficiente de llevar a cabo la búsqueda del tetraedro deseado pasa por utilizar un camino de adyacencias partiendo de un tetraedro cercano a la posición del punto X . Para conocer la vecindad del punto deseado, se puede recurrir a la realización de un octree [12]. Se parte de un cubo que contiene todos los puntos para los que se cuenta con una HRIR. El primer cubo es conocido como nodo raíz. A partir del mismo, se divide de manera recursiva en 8 nuevos cubos hijos y se comprueba la cantidad de puntos contenidos en cada uno de los hijos. La subdivisión de los cuboides en 8 se detiene cuando estos contienen un máximo de N puntos fijado en la programación. Los cubos hijos que contienen este máximo de N puntos, se denominan nodos “leaf”. Para encontrar el tetraedro cercano a la posición deseada, se busca el nodo leaf contenedor de dicha posición. Un tetraedro contenido en dicho nodo, será adecuado para comenzar la búsqueda del tetraedro definitivo y unívoco contenedor de X en la mayoría de los casos.

4.4.- ORGANIZACIÓN DE CLASES, MÉTODOS Y ATRIBUTOS

La aplicación software fue dividida en diferentes clases para facilitar el orden, limpieza e interacción entre las diversas funciones del programa. Tras un estudio en profundidad de las necesidades software, se llegó a la implementación de las clases resultantes, presentadas en los siguientes subapartados.

4.4.1.- Clase Emisor

La clase **Emisor** permite crear instancias de fuentes sonoras con toda la información necesaria para definir las correctamente. Básicamente, esta información consiste en una posición (definida en coordenadas cartesianas respecto a un eje de referencia) y un array contenedor de una señal de audio estéreo, que representa el sonido del emisor. Por tanto, los **atributos** de la clase son:

- Posición: array de tres elementos representativos de una posición en coordenadas cartesianas,
- arrayAudio: Señal de audio estéreo que se quiere asociar a cada instancia del objeto Emisor. El array contiene dos columnas, una representativa del canal de audio izquierdo, y otra del derecho.
- sr: Frecuencia de muestreo de la señal de audio.

Mediante el constructor, denominado “__init__” (también para el resto de las clases del programa) se pueden definir los atributos inicialmente. Además, el constructor permite cargar audio desde un archivo (gracias a la clase “GestionAudio” y en caso de que se pase un directorio como parámetro), y directamente desde un array contenedor de una señal de audio, lo que permite utilizar ondas generadas sintéticamente.

Por otro lado, el resto de métodos permiten obtener (si el nombre del método comienza con “Get”) o modificar (si el nombre del método comienza con “Set”) los atributos de la clase:

- | | |
|-----------------|-----------------|
| - SetPosicion | - GetPosicion |
| - SetSound | - GetSound |
| - SetSampleRate | - GetSampleRate |
| - GetSoundLeft | - GetSoundRight |

4.4.2.- Clase Receptor

La clase receptor permite ubicar al oyente en el espacio respecto al mismo eje de coordenadas cartesianas que los emisores. Además, permite definir la orientación de la cabeza del mismo. Por tanto, sus atributos son:

- Posición: Array de tres elementos representativos de una posición en coordenadas cartesianas,
- Orientación: Valor en grados para definir la orientación de la cabeza del oyente respecto al eje de coordenadas cartesianas de referencia.

Tanto el atributo posición como el de orientación se inicializan con el constructor `__init__` de la clase. Además, las funciones “set” y “get” permiten modificar o leer los valores de los atributos respectivamente:

- SetPosicion
- GetPosicion
- SetOrientacion
- GetOrientacion

4.4.3.- Clase Física

La clase física rige la relación espacial de las instancias emisoras respecto al receptor. Para ello se sirve de todas las instancias del objeto emisor más la del receptor, que se convertirán en atributos de la clase física. Por tanto, los atributos son:

- Emisores: Se trata de un array de instancias del objeto emisor.
- Receptor: Contiene toda la instancia del objeto receptor.
- nEmisores: Se trata simplemente del número de emisores, coincidente con la longitud del array “Emisores”. Servirá para determinar el número de iteraciones en los métodos de la clase.

- Coordenadas: Se inicializa una matriz de ceros de acuerdo con el número de emisores. Esta matriz acabará por contener las posiciones de los emisores respecto al receptor en coordenadas esféricas.
- distMax: Máxima distancia para el mapeo. Se usará como referencia, ya que, mediante la técnica y la base de datos utilizada, se cuenta con un límite al partir del cual no es posible calcular correctamente las respuestas a impulso. Este límite viene definido por la aplicación de una técnica de interpolación válida para el *near-field*, es decir, las posiciones a menos de 2 m de distancia de la cabeza.

Los atributos son definidos en el constructor “*__init__*” y serán empleados en los siguientes métodos:

- CoordenadasRespectoReceptor: Esta función toma las coordenadas del receptor y las utiliza como referencia para obtener las coordenadas de los emisores respecto a este. Para ello, se constituye una matriz de rotación más traslación T que realizará el cambio de base: desde el eje cartesiano de referencia al eje cartesiano con origen en el centro de la cabeza del receptor:

$$T = \begin{pmatrix} \cos \theta & -\sin \theta & 0 & t_x \\ \sin \theta & \cos \theta & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (4.9)$$

Después, los puntos respecto a la cabeza del receptor se calcularán como:

$$P_r = T^{-1}P_o \quad (4.10)$$

Se realiza una iteración para obtener cada nueva posición respecto al receptor por cada emisor. Se actualiza la posición original del emisor y se emplea la ecuación 4.10 para obtener las coordenadas resultantes, que se almacenan en la matriz inicializada de coordenadas que se devuelve al final de la función.

Es cierto que se podría haber trabajado desde el principio utilizando el centro de la cabeza como eje de referencia y evitar así este cambio de base. Sin embargo,

es mucho más intuitivo posicionar todos los objetos al principio, tanto emisores como receptores, en un sistema cartesiano para cualquier posible usuario del software.

La rotación solo se ha tenido en cuenta alrededor del eje z, es decir, lo que corresponde a girar la cabeza o el cuerpo en sentidos horario o antihorario. No se tiene en cuenta, por tanto, el lado de la cabeza (lo cual implicaría tener más muestras de HRIR de base) ni la rotación para elevar o agachar la cabeza.

- `CoordsRespectoReceptorEsfMap`: Esta función mapea a una posición adecuada todas las coordenadas de los emisores respecto al receptor obtenidas utilizando la función previa (`CoordenadasRespectoReceptor`). Para ello, se pasan a coordenadas esféricas las posiciones con las que se cuenta. Esta operación se realiza por dos motivos: porque se quiere que el resultado sea en coordenadas esféricas y porque la tercera componente de las coordenadas es la distancia. Como se pretende adaptar la posición de todos los emisores en base a aquel más alejado, el mapeo se resolverá con una simple división.

Además, la clase física cuenta con las funciones encargadas del cambio de sistema de coordenadas, de esféricas a cartesianas (`Esf2Car`), y de cartesianas a esféricas (`Car2Esf`).

4.4.4.- Clase `GestionHRIR`

Una vez se obtienen las posiciones de los emisores respecto al receptor, se debe calcular la HRIR adecuada a esas posiciones. La clase `GestionHRIR` reúne todas las operaciones necesarias, basándose para ello en la técnica de interpolación tetraédrica ya descrita.

La clase es inicializada mediante un constructor en el que se definen los siguientes atributos:

- `Filepath`: Se trata del directorio en el que se haya la base de datos con las HRIR.
- `Points`: Es una matriz que contiene las posiciones en coordenadas esféricas respecto al receptor de los puntos que cuentan con una HRIR. Será usada de manera recurrente en los métodos de la clase.

- binCapacity: Es un parámetro que se toma como referencia para la división del espacio en octrees.
- T: Se trata de una matriz que contiene los índices de los puntos que forman los diferentes tetraedros resultantes de triangular los puntos para los que se cuenta con una HRIR. Esta matriz proviene de un fichero de tipo .csv obtenido con Matlab y la *built-in-function* relativa a la triangulación de Delaunay ya comentada. Cada fila contiene el índice de cada punto que conforma el mismo tetraedro, por tanto, es una matriz de 4 columnas. Es posible saber las coordenadas de cada punto remitiéndose a la matriz *Points* en la que cada fila representa el índice de un punto en el espacio.
- Nei: Se trata de otra matriz proveniente de un fichero .csv generado en Matlab. Esta matriz define la relación entre los diferentes tetraedros, es decir, las vecindades o adyacencias. Cada fila representa el índice de los tetraedros vecinos al tetraedro cuya identificación coincide con el número de fila. El índice de los tetraedros proviene del número de fila de la matriz T.
- OT: Se trata de una estructura que contiene todos los datos concernientes al octree. En Python no existe el tipo “struct” de la misma forma que en lenguaje C, aunque se puede conseguir un tipo parecido utilizando un “namedtuple”. El namedtuple constituido contiene los siguientes campos: *Points*, *PointBins*, *BinBoundaries*, *BinParents*, *BinDepths*, *BinCount*, *BinCentres*, *BinChildren*. Estos campos se explicarán de acuerdo vayan surgiendo según los métodos de la clase presente.

A continuación, se describen los métodos utilizados en la clase presente. Se ha tratado de presentarlos de la forma más ordenada posible de acuerdo con el algoritmo de interpolación tetraédrica ya explicado en la sección 4.3.

- get_HRTF_coords: Se trata del método creado para generar una matriz contenedora de las coordenadas de las posiciones para las que se cuenta con una HRIR. Esta función, básicamente utiliza la información de la tabla 4.1 para devolver la matriz que se asigna al atributo “points” ya mentado.

- getOT y getChildren: Estos métodos no se entienden el uno sin el otro, ya que su objetivo es originar el octree que servirá de referencia para encontrar el tetraedro deseado. La función getOT inicializa con matrices de valor cero de los campos de la “namedtuple” OT, salvo para *BinBoundaries*, que contendrá un array con las coordenadas de los puntos límite mínimo y máximo del octree. A continuación, se llama a la función *getChildren*, que se va a encargar de generar los nodos del octree. La generación del octree se basa en la utilización de la recursividad, técnica consistente en el llamamiento de la propia función. El algoritmo consiste en realizar 8 divisiones para cada nodo generado, que se seguirán parcelando en 8 nuevas divisiones hasta llegar a los nodos leaf (ver sección 4.3). Es por ello que la recursión se convierte en una técnica tan preciada en este caso, ya que permite generar las ramificaciones del octree sin perder el orden y con poco código. Al tiempo que estas ramificaciones son generadas, se completan los campos de la namedtuple OT cuya función es registrar y obtener un “mapa” de los nodos padre e hijo que se generan:
 - PointBins: Índices del nodo al que pertenece cada punto.
 - BinCount: Número total de nodos creados.
 - BinDepths: El número de subdivisiones para alcanzar cada nodo.
 - BinParents: Índices del nodo al que cada nodo hijo pertenece.
- getOTcentres: Este método busca un centro para cada nodo del octree generado y rellena el campo asociado a este parámetro de la namedtuple.
- queryOT: Este método se encarga de buscar un tetraedro próximo a la posición de la fuente sonora, para evitar hacer un gran número de iteraciones para encontrar el tetraedro adecuado. Devuelve el índice de dicho tetraedro.
- FindGains: Esta función implementa las ecuaciones 4.7 y 4.8 vistas en la sección 4.3 para finalmente obtener las coordenadas baricéntricas del punto en el que se haya un emisor de sonido cuya HRIR se desea obtener. Asimismo, devuelve los índices de los puntos que forman el tetraedro contenedor de la posición del emisor.

- InterpolarHRIR y ReadHRIR: Mediante estas dos funciones, se busca y se calcula la HRIR del punto deseado. Se itera para conseguir almacenar en sendas matrices, una para el canal izquierdo y otra para el derecho, las HRIR correspondiente a cada vértice del tetraedro. Para realizar esta función de manera eficiente, la función ReadHRIR, que permite navegar por el directorio de HRIRs, cuenta con un sistema de lectura basado en el uso del tipo “string” que permite buscar el fichero de respuestas a impulso cuyo nombre contiene los valores de las coordenadas esféricas deseadas. A continuación, basta multiplicar cada HRIR leída por el peso correspondiente obtenido en la función FindGains, y sumarlas para conseguir la HRIR interpolada. Finalmente se devuelve la respuesta a impulso asociada a cada oído.
- ComputeHRIR: Este método fue creado con el objetivo de evitar la repetición de cálculos, aumentando la eficiencia del programa. Básicamente, llama a los métodos que dependen de la posición de la fuente de sonido, permitiendo que una vez que sea creado correctamente un objeto de tipo “GestionAudio”, solo haga falta llamar externamente al método actual. Es por esto mismo que la llamada a otros métodos de la clase no dependientes de la posición del emisor (get_HRTF_coords, getOT, getOTcentres), se llevan a cabo en el mismo constructor.

4.4.5.- Clase GestionAudio

Esta clase se encarga de convertir el audio de un determinado formato (mp3, wav...) en un array de valores procesable de manera digital y viceversa. Contiene dos funciones principales: ReadAudio y WriteAudio.

- ReadAudio: Se encarga de la lectura del audio y de darle el formato adecuado si fuera necesario. Devuelve la frecuencia de muestreo del archivo de audio y la señal digitalizada en valores flotantes.
- WriteAudio: Permite exportar un array de audio procesable como un archivo con formato de audio, así como otorgarle nombre al mismo.

4.4.6.- Clase Procesamiento

Una vez se cuenta con la HRIR adecuada para la posición del emisor respecto al receptor, es hora de aplicarla a la señal de audio asociada al emisor. Para ello, se definen los siguientes atributos de la clase:

- F_s : Frecuencia de muestreo del audio que se desea procesar.
- x_{left} : Señal de audio a procesar asociada al canal izquierdo.
- x_{right} : Señal de audio a procesar asociada al canal derecho.
- h_l : Respuesta a impulso asociada al oído izquierdo.
- h_r : Respuesta a impulso asociada al oído derecho.
- $hSize$: Número de muestras de la HRIR. Se puede usar el mismo valor tanto para h_l , como para h_r , ya que tendrán la misma longitud.
- $xSize$: Número de muestras de la señal de entrada, que, de nuevo, será igual tanto para el canal izquierdo como para el derecho.

Con todos los atributos definidos en el constructor, se pasa a los métodos que los emplearán:

- **Convolución:** Se trata de la operación del procesamiento digital de señales más importante. Esta función pretende aplicar la fórmula 2.2 de la manera más eficiente en el dominio del tiempo. El algoritmo de convolución puede describirse mediante los siguientes pasos:
 1. Inicializar un array xk de N elementos de valor cero, donde N es el número de coeficientes de la respuesta de impulso h utilizada.
 2. Guardar el valor actual de la señal de entrada en el primer componente del array xk e inicializar una variable auxiliar yk a cero.
 3. Para cada valor de la respuesta a impulso h_n , realizar una multiplicación componente por componente de las matrices de la respuesta a impulso y xk . Utilizar la variable auxiliar yk definida en el paso anterior como acumulador de las diferentes multiplicaciones. Una vez finalizadas las multiplicaciones, yk contendrá el valor de la señal de salida actual, que será el producto escalar de xk y h .

4. Desplazar todos los valores del array x_k una posición hacia la derecha.
5. Volver al paso número 2.

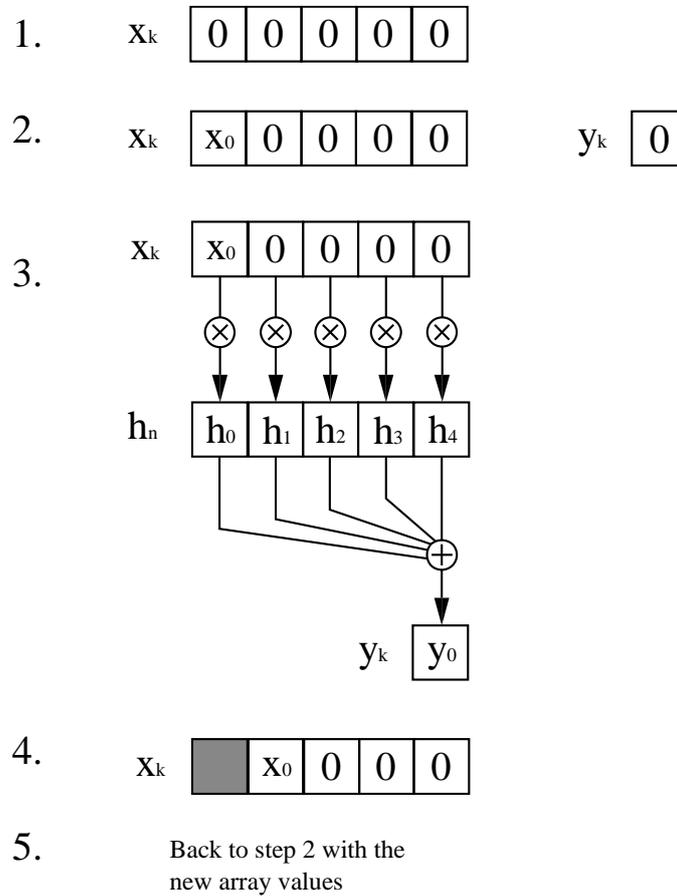


Figura 4.4.- Representación gráfica del algoritmo de convolución.

La mayor ventaja de este algoritmo es que no requiere ninguna estructura condicional. La creación del array x_k , que es inicializado con valores de cero, evita hacer un “padding” de ceros el comienzo de la señal de entrada y permite realizar el producto escalar de manera directa. Además, la implementación del algoritmo evitando “built in functions”, da la posibilidad de realizar modificaciones en el futuro, por ejemplo, para permitir el cambio de una HRIR a otra diferente durante el procesamiento.

4.4.7.- Otras clases y funciones

Otras clases desarrolladas son la clase Punto, y la clase Reproducción. La primera cuenta con tres atributos (x , y , z) que sirven para manejar los datos de coordenadas cartesianas. La segunda cuenta con un método para la reproducción del audio procesado, aunque su desarrollo completo se vio perjudicado por la falta de interacción de la aplicación software.

5. Resultados y experimentos

Al carecer del equipo necesario (cámara anecoica, equipo de grabación, dummy head etc...) para realizar una validación experimental completa de los resultados, se recurrió a una serie de análisis, tanto objetivos como subjetivos. Para el análisis objetivo, se estudió la representación gráfica de las señales de audio, verificando que coincidieran con lo esperado, Por otra parte, el análisis subjetivo se llevó a cabo gracias a una encuesta en la que se sometían a evaluación varios aspectos del procesamiento. En esta sección se detallan los resultados para cada una de las partes.

5.1.- ANÁLISIS OBJETIVO DE LOS RESULTADOS

Una forma de verificar los resultados del procesamiento a partir de las HRIR interpoladas, pasa por analizar la representación gráfica de la onda sonora generada a partir de la HRIR de algunos puntos clave. Estos puntos pueden ser posiciones básicas, como la de fuentes sonoras colocadas en frente de cada oreja o delante o detrás de la cabeza.

Como es de esperar, una fuente sonora colocada en frente de la oreja izquierda y perpendicular a ella dará origen a una respuesta a impulso de mayor magnitud para el oído izquierdo que para el derecho (ILD), y tardará más tiempo en llegar a este último (ITD). En la imagen inferior se muestra la HRIR interpolada para cada oído tomando como referencia un punto perpendicular al oído izquierdo. El punto en cuestión es el (270, 0, 73) en coordenadas esféricas, es decir, un punto a 73 centímetros en perpendicular del oído izquierdo.

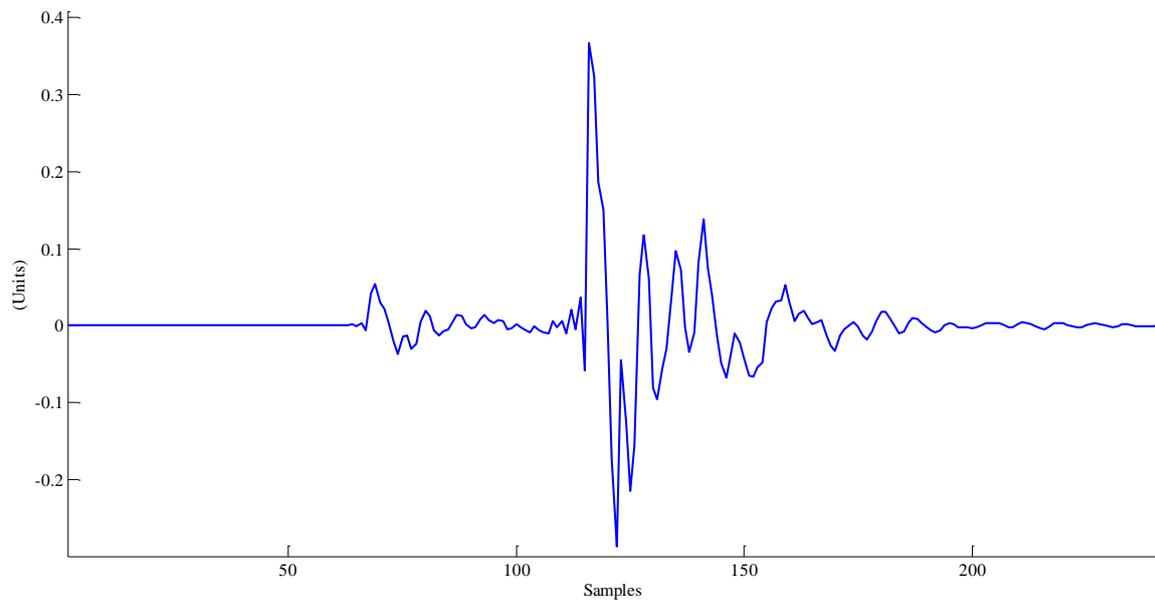


Figura 5.1.- HRIR oído izquierdo (270, 0, 73).

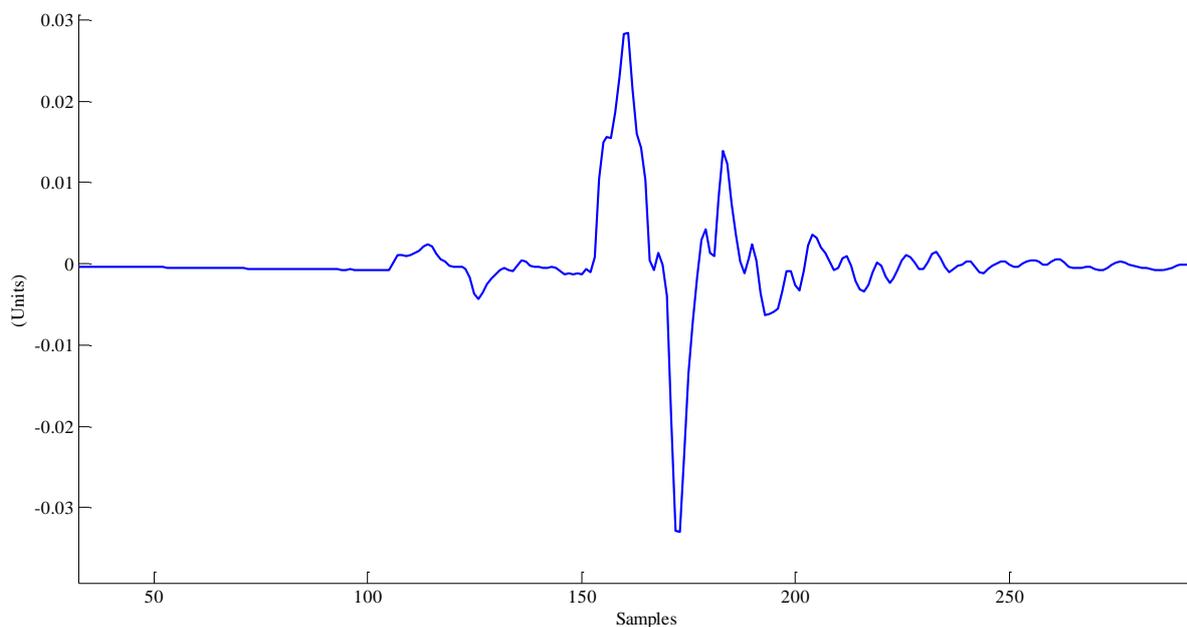


Figura 5.2.- HRIR oído derecho (270, 0, 73).

Para comprobar el valor de ILD, se pueden tomar los máximos o mínimos en valor absoluto de cada respuesta a impulso, y calcular la relación entre ellos. Por otra parte, la ITD puede ser determinada a partir del número de muestras de diferencia entre el transitorio de cada respuesta impulso. Es difícil llegar a una cifra exacta, ya que la forma

de las respuestas a impulso para ambos oídos es diferente, pero se puede afirmar que la diferencia entre ambos transitorios es de alrededor de 40 muestras. Con esta cifra, dividida entre la frecuencia de muestreo de 44100 Hz, se obtiene una ITD de 0,9 ms. Este valor concuerda en cierta manera con lo establecido en la sección 2.2.1. No se trata exactamente del mismo valor referido, pero, tal y como se había comentado, los 0,59 ms establecidos como ITD estándar, no tenían en cuenta la propagación del sonido a través de la cabeza.

Utilizando una senoidal de 200 Hz y las respuestas a impulso obtenidas, se realizó el procesamiento de la señal por medio de la convolución. Las características de la señal tras el procesamiento deberán cumplir con las restricciones provenientes de las HRIR ya comentadas:

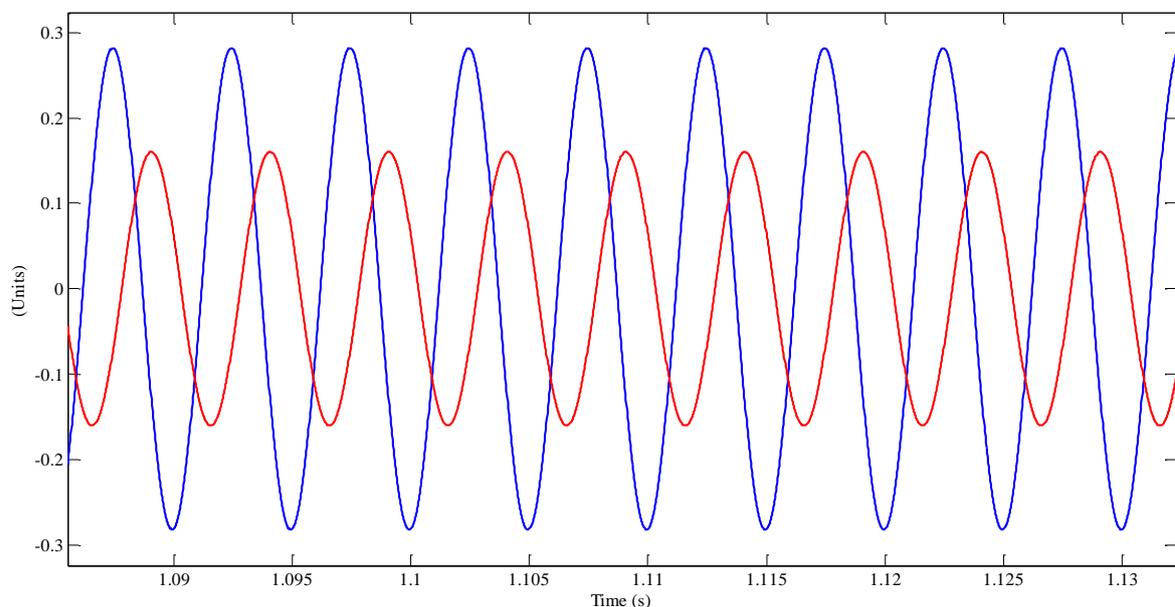


Figura 5.3.- Procesamiento de senoidal de 200 Hz ubicada en la posición (270, 0, 73). En color azul, señal asociada al oído izquierdo; en rojo, señal asociada al oído derecho.

Teniendo en cuenta que la señal azul es la senoidal asociada al oído izquierdo, y la roja la senoidal asociada al oído derecho, se puede afirmar que los resultados son coherentes. La señal izquierda tiene mayor intensidad que la derecha, ya que la fuente de sonido se halla más próxima al oído izquierdo. Por otro lado, hay un desfase derivado de la ITD, proveniente de la diferencia del origen del transitorio entre las HRIR.

Para otra fuente, cercana al oído derecho en vez de al izquierdo, debe suceder algo muy similar, pero siendo el oído derecho el que percibe de manera más fuerte la intensidad que el izquierdo y al que el sonido le llega antes:

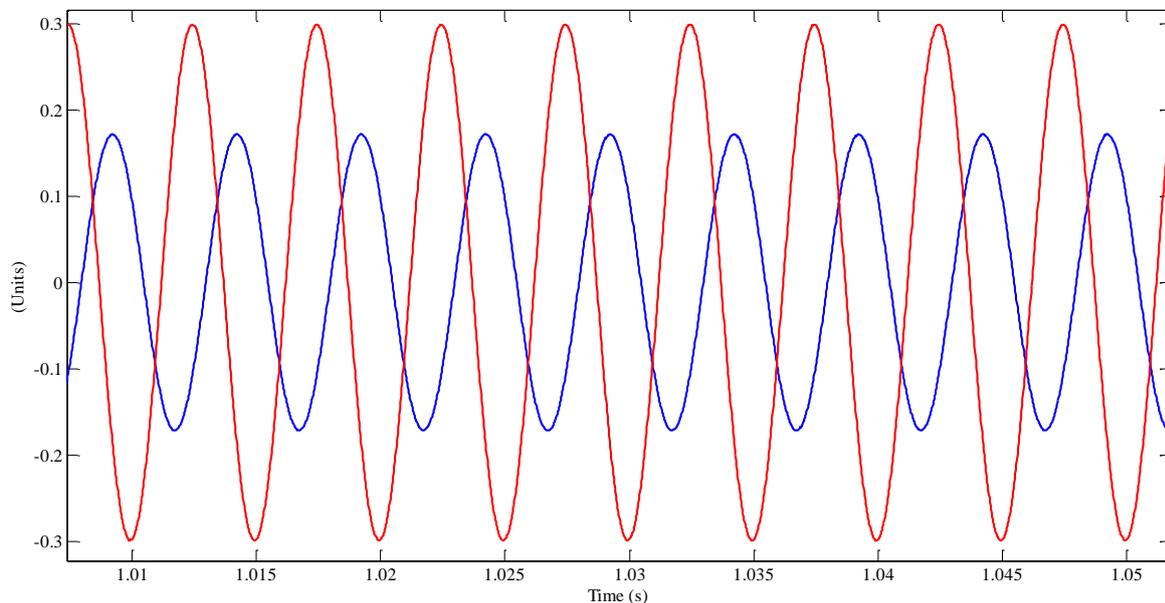


Figura 5.4.- Procesamiento de senoidal de 200 Hz ubicada en la posición (90, 0, 73). En color azul, señal asociada al oído izquierdo; en rojo, señal asociada al oído derecho.

De nuevo, los resultados, son consistentes con lo esperado. Se puede observar que la intensidad de las ondas no tiene exactamente los mismos valores que en el anterior caso, pero sí muy parecidos. Estas variaciones son determinadas por las diferencias de la capacidad auditiva del sujeto, o en este caso del dummy head que sirvió para hacer las mediciones de las HRIR.

A priori, las señales emitidas por una fuente ubicada en frente de cada oído deben ser discernidas correctamente de manera habitual. Pero ¿qué hay de las señales cuya ILD y cuya ITD es muy parecida? Por ejemplo, en los casos en los que la fuente de sonido se ubica justo en frente o detrás del oyente:

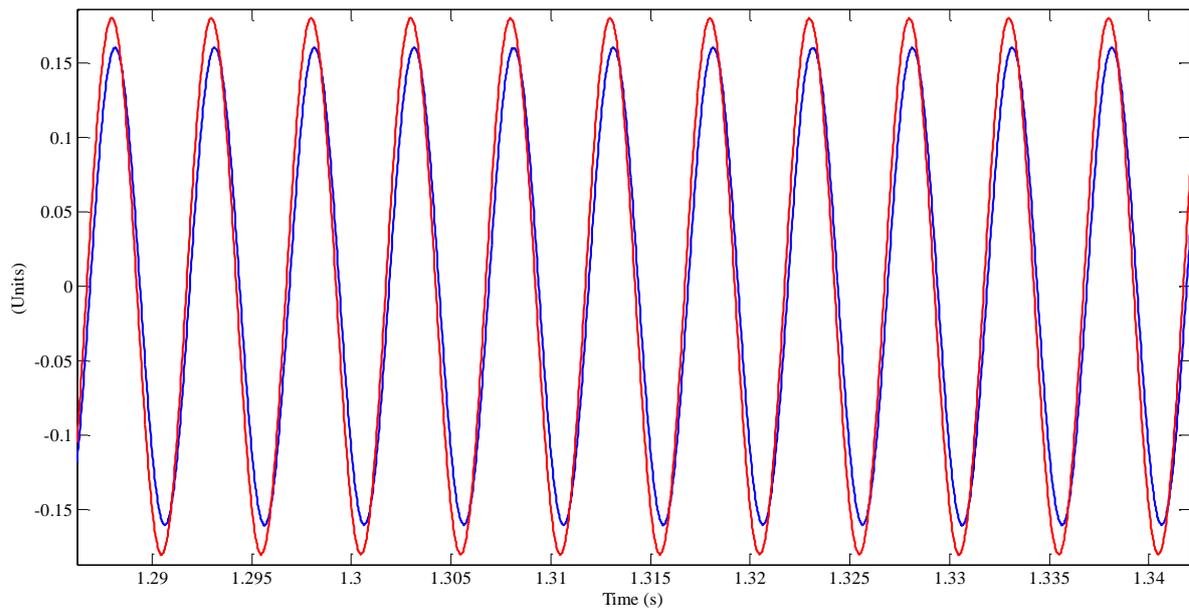


Figura 5.5.- Senoidal de 200 Hz procesada para la posición (0, 0, 73). En color azul, señal asociada al oído izquierdo; en rojo, señal asociada al oído derecho.

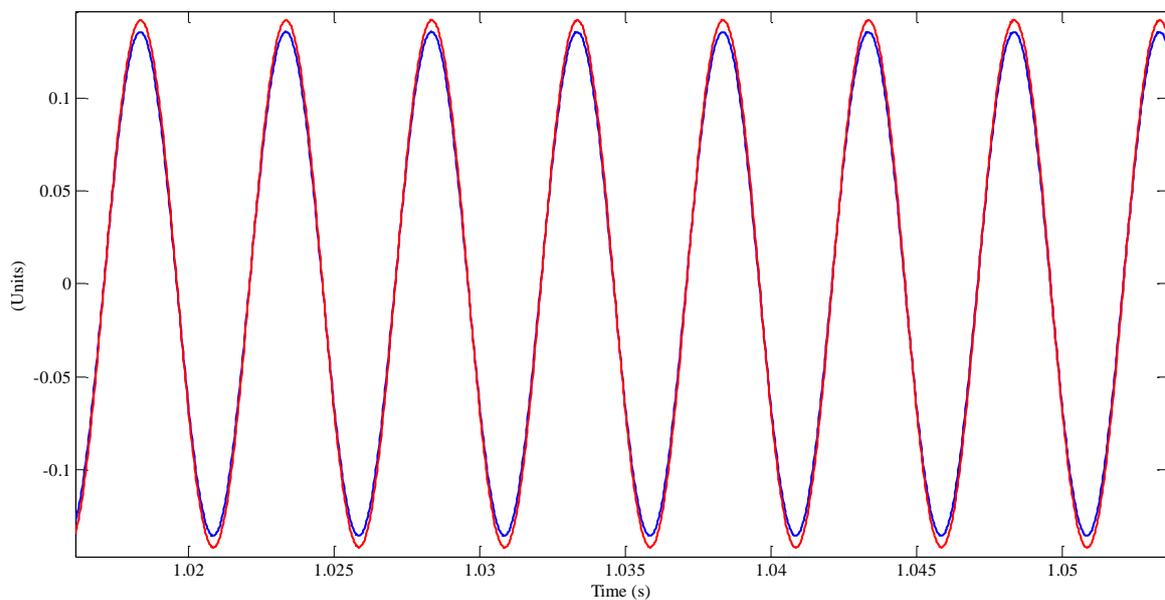


Figura 5.6.- Senoidal de 200 Hz procesada para la posición (180, 0, 73). En color azul, señal asociada al oído izquierdo; en rojo, señal asociada al oído derecho.

Como es de esperar, la intensidad de una fuente sonora proveniente de en frente, posición (0, 0, 73), es percibida con mayor intensidad que la de una proveniente de detrás, (180, 0, 73) a pesar de que ambas se encuentren a la misma distancia del centro de la cabeza. Esto se debe a que el pabellón auricular está mejor orientado hacia los sonidos provenientes de

delante. No obstante, en el caso de tener una señal con poco contenido armónico, como una senoidal de frecuencia 200 Hz, es muy difícil reconocer de dónde proviene el sonido. La única forma, sería tener muy claro cuál es la intensidad habitual del mismo para saber diferenciar si el emisor se ubica delante o detrás del oyente. Sin embargo, existe otro parámetro que puede ayudar a reconocer de dónde proviene un sonido. Se trata de la frecuencia, ya que un sonido determinado será “filtrado” por el sistema auditivo y el propio cuerpo de una forma u otra dependiendo de la posición de la que provenga:

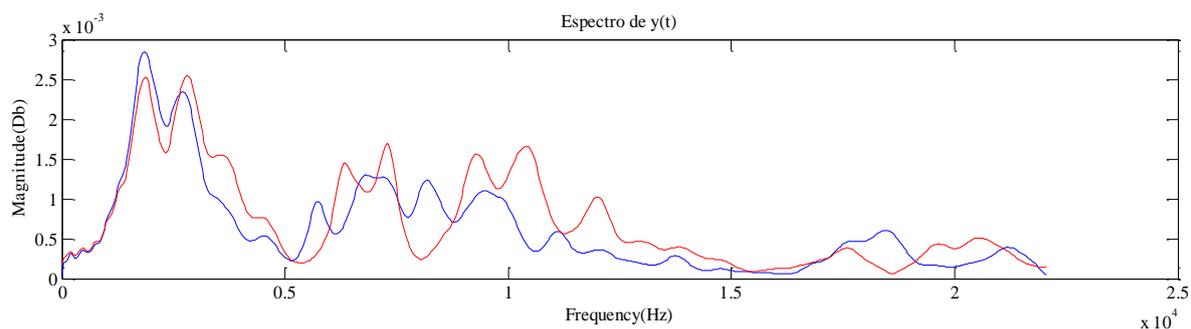


Figura 5.7.- Respuesta en frecuencia del oído izquierdo (color azul), y oído derecho (color rojo) para los sonidos ubicados en la posición (0, 0, 73).

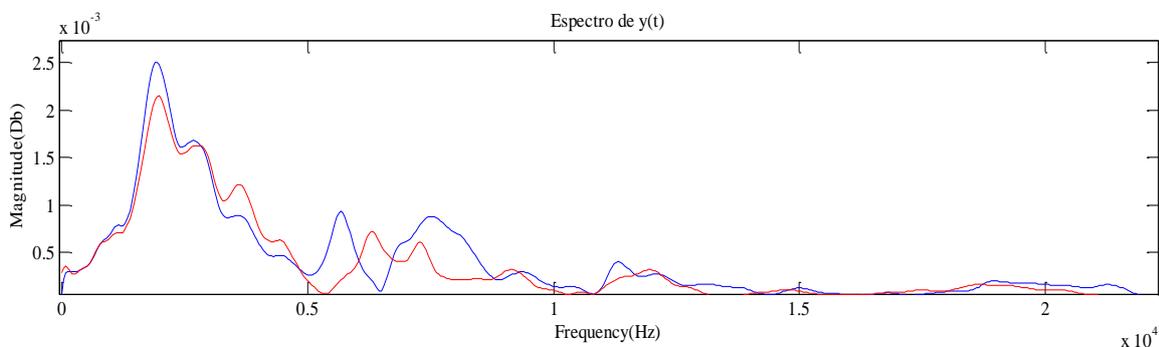


Figura 5.8.- Respuesta en frecuencia del oído izquierdo (color azul), y oído derecho (color rojo) para los sonidos ubicados en la posición (180, 0, 73).

En las figuras 5.7 y 5.8, se puede diferenciar este comportamiento hacia las diferentes frecuencias del espectro. Gracias a la representación en frecuencia de estas respuestas a impulso interpoladas (es decir, respuestas en frecuencia), se puede constatar que la percepción sonora es mucha más eficaz cuando el sonido viene desde delante. Para los sonidos inferiores a los 2000 Hz, no existe tanta diferencia (aunque la intensidad del sonido es siempre mejor percibida cuando la fuente está situada de frente), pero una vez se

pasa esta barrera, las frecuencias que vienen desde detrás de la cabeza se vuelven mucho más atenuadas a la percepción humana (figura 5.8). A continuación, se muestra mejor este comportamiento, tras procesar una señal compuesta por dos senoidales; una de 2780 Hz y otra de 6331 Hz.

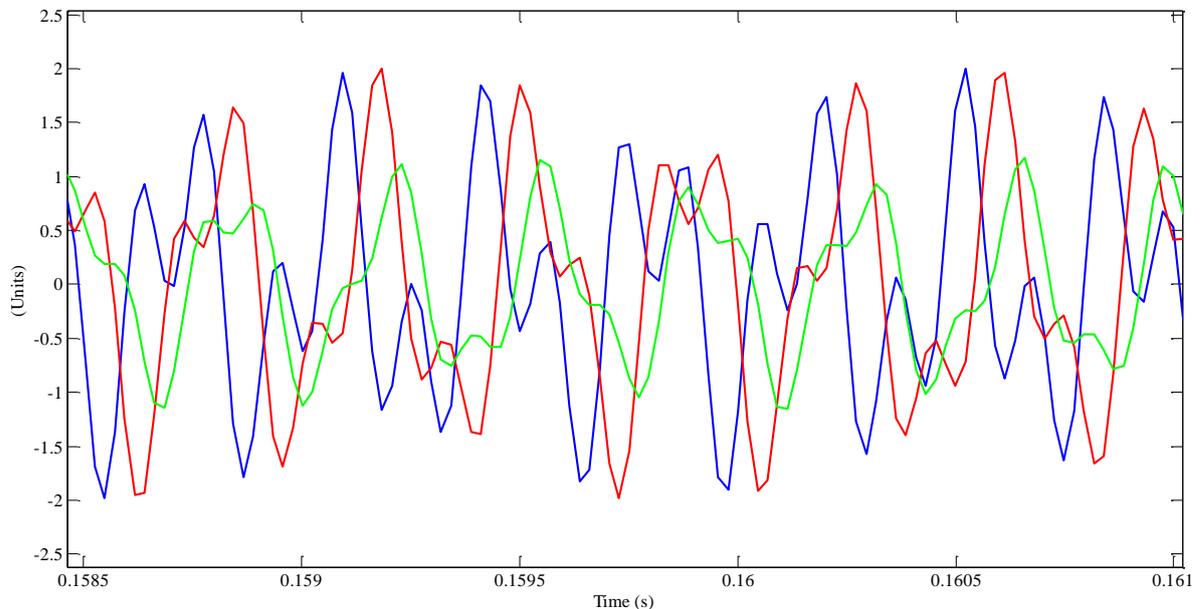


Figura 5.9.- En azul, se representa la señal original; en rojo la señal filtrada para el oído derecho con la HRIR asociada a la posición (0,0,73); en verde la señal filtrada para el oído derecho con la HRIR asociada a la posición (180,0,73).

Las frecuencias que componen la señal no fueran escogidas aleatoriamente, ya que coinciden con máximos locales de la respuesta en frecuencia asociada a la posición (0, 0, 73) que son muy atenuados para la respuesta asociada a la posición (180, 0, 73), pero que sirven para mostrar el efecto auditivo producido claramente.

Para la representación de las señales procesadas, se ha optado por solo incluir las relativas al oído derecho, para ambas posiciones, con el fin de no saturar la imagen. Se puede observar que la señal para la posición asociada a (0, 0, 73) en color rojo, apenas es atenuada respecto la señal original (color azul). Por otro lado, la señal verde asociada a la posición (180, 0, 73), sufre una atenuación mucho mayor para ambas frecuencias que conforman la señal. Atenuación que es perceptible en el dominio del tiempo, pero es cuantificable y se visualiza mejor en el dominio de la frecuencia:

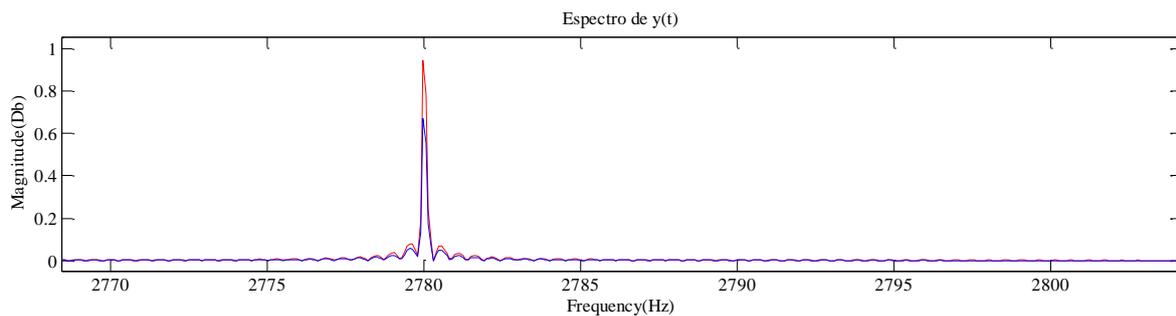


Figura 5.10.- Imagen del espectro centrado en 2780 Hz para la señal procesada en los casos de asociarla a las posiciones 0,0,73 (color rojo) y 180,0,73 (color azul).

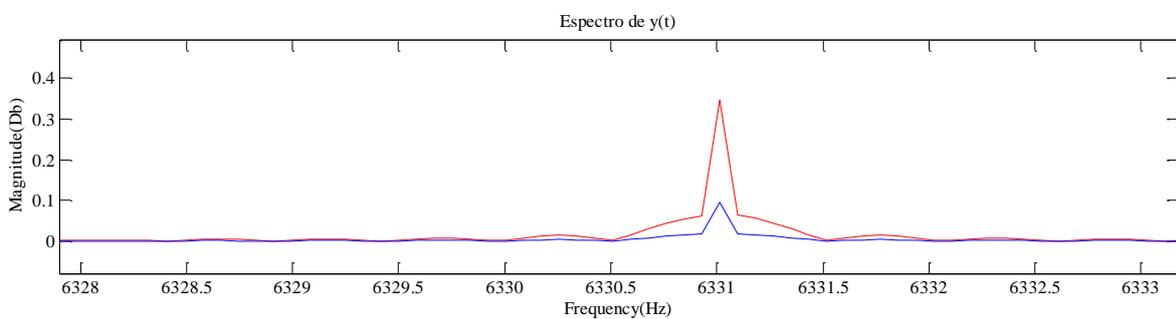


Figura 5.11.- Imagen del espectro centrado en 6331 Hz para la señal procesada en los casos de asociarla a las posiciones 0,0,73 (color rojo) y 180,0,73 (color azul).

La representación del espectro confirma lo que se sospechaba y es coherente con las respuestas a impulso interpoladas. Para ambas posiciones, la visualización del espectro responde con lo esperado, y para la posición (180, 0, 73) casi cancela la frecuencia de 6331 Hz.

Por otro lado, la percepción del sonido dependiendo de si este procede por encima o por debajo de la línea de los oídos, estará determinada por los mismos factores que ya se han comentado para las posiciones básicas. Por ejemplo, la respuesta a impulso del punto (32,55, 41,88, 74.89) dará lugar a la siguiente respuesta en frecuencia:

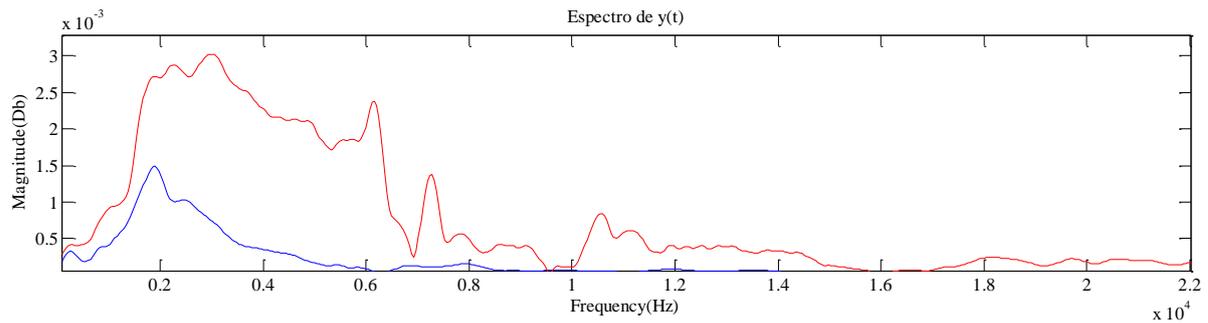


Figura 5.12.- Respuesta en frecuencia asociada a (32,55, 41,88, 74,89). En azul, respuesta asociada al oído izquierdo; en rojo, respuesta asociada al oído derecho.

Se puede observar que la recepción de un sonido proveniente de la posición (32,55, 41,88, 74,89) será más clara en el oído derecho (espectro de color rojo) que en el izquierdo (espectro de color azul). Se trata de un hecho coherente ya que el azimuth indica que la fuente de sonido está desplazada 32° en sentido horario del eje principal que parte del centro de la cabeza hacia el frente. Además, la forma del espectro no tiene nada que ver con la de las posiciones que están a la altura de los oídos, lo que es un hecho consistente con la posición referida en este caso.

A continuación, se utiliza esta posición para analizar la corrección del algoritmo de interpolación utilizado. En la siguiente imagen se aprecia en color azul y con círculos señalando cada punto la HRIR interpolada para la posición (32,55, 41,88, 74,89). El resto de señales representan las HRIR usadas en la interpolación.

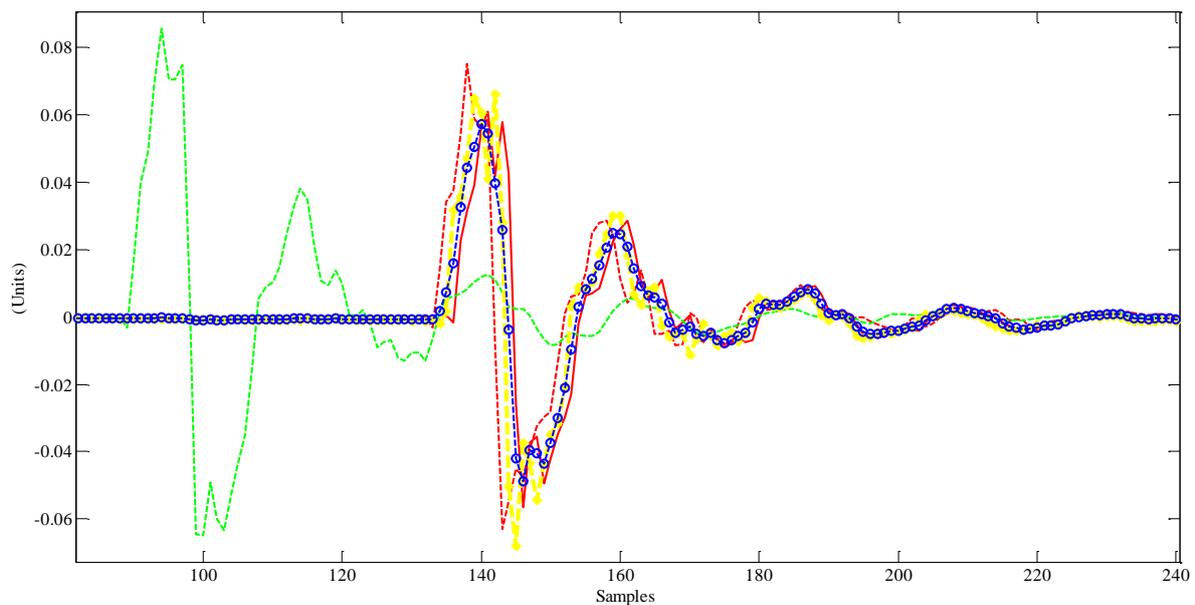


Figura 5.13.- Señalada con puntos la HRIR interpolada.

A priori, puede parecer que la respuesta interpolada no es del todo correcta, ya que la HRIR de color verde no parece estar teniendo toda la influencia que debería. Sin embargo, este hecho se puede verificar echando un vistazo a la posición de los puntos utilizados en la interpolación y a las coordenadas baricéntricas del receptor:

Posición	Color	Peso o coordenadas baricéntricas del receptor
35 40 75	Rojo	0.5057
30 50 75	Rojo, línea discontinua	0.1883
35 40 50	Verde	0.0043
30 40 75	Amarillo	0.3017

Tabla 5.1

De acuerdo con la tabla 5.1, la HRIR asociada a la posición (35, 40, 75) es la más relevante para la interpolación. Esto hace que la HRIR interpolada se parezca lo más posible a esta. En el otro extremo, se encuentra la señal asociada a la posición (35, 40, 50), de color verde,

que tiene una influencia ínfima, ya que este punto es el más alejado de la posición de la que se quiere obtener la HRIR.

5.2.- ANÁLISIS SUBJETIVO DE LOS RESULTADOS, ENCUESTA

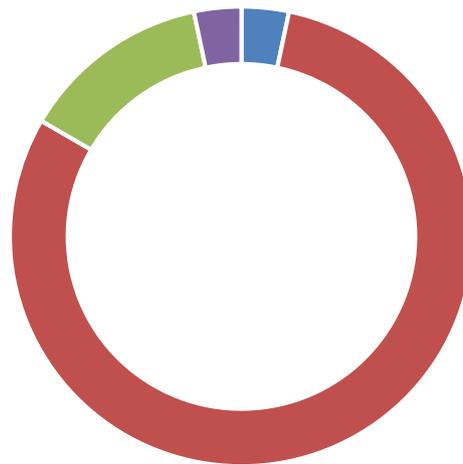
A pesar de los análisis de las representaciones gráficas del audio procesado vistas hasta el momento, la aplicación está diseñada para ser utilizada sencillamente escuchando el audio procesado, sin necesidad de recurrir a representaciones gráficas de ningún tipo. Por ello, y con el objetivo de evaluar su actuación y funcionamiento, se ha seleccionado a 30 personas, en su mayor parte de perfil técnico, para realizar un cuestionario al que debían responder a preguntas relacionadas con el audio procesado. Para agilizar el proceso, se realizó un cuestionario con varias opciones, aunque dando la posibilidad de escribir la respuesta deseada.

A continuación, se detallan los resultados para cada pregunta.

5.2.1.- Primera pregunta, senoidal por el oído derecho

Para esta pregunta, se pidió a los encuestados que escucharan una onda senoidal procesada para que parezca que proviene desde la **derecha** del oyente (90, 0, 73). Se preguntó de dónde parecía que provenía el sonido y se ofrecieron las siguientes opciones con los resultados siguientes:

1. De delante: 3,3 %
2. De la derecha: 80 %
3. De la izquierda: 13,3 %
4. De detrás: 3,3 %



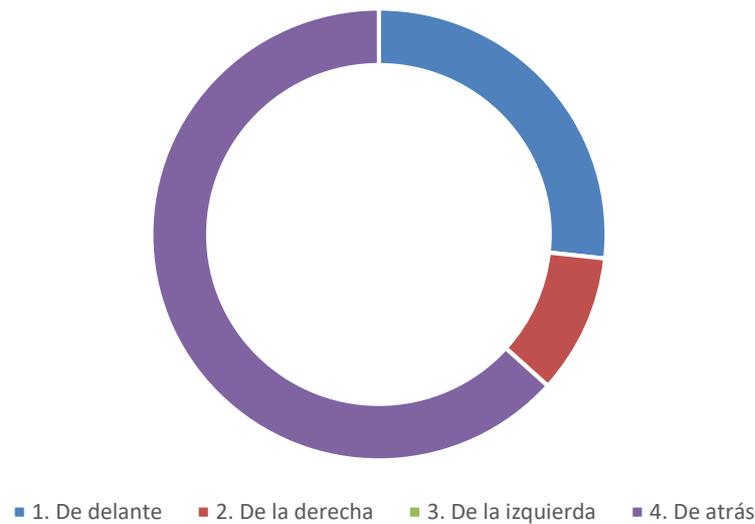
■ 1. De delante ■ 2. De la derecha ■ 3. De la izquierda ■ 4. De atrás

Por tanto, en este caso, la tasa de acierto fue del **80 %**.

5.2.2.- Segunda pregunta, ruido mecánico por detrás

Para esta pregunta, se pidió a los encuestados que escucharan un ruido mecánico procesado para que parezca que proviene desde **detrás** del oyente (180, 0, 73). Se preguntó de dónde parecía que provenía el sonido y se ofrecieron las siguientes opciones con los resultados siguientes:

1. De delante: 26.7 %
2. De la derecha: 10 %
3. De la izquierda: 0 %
4. De detrás: 63,3 %

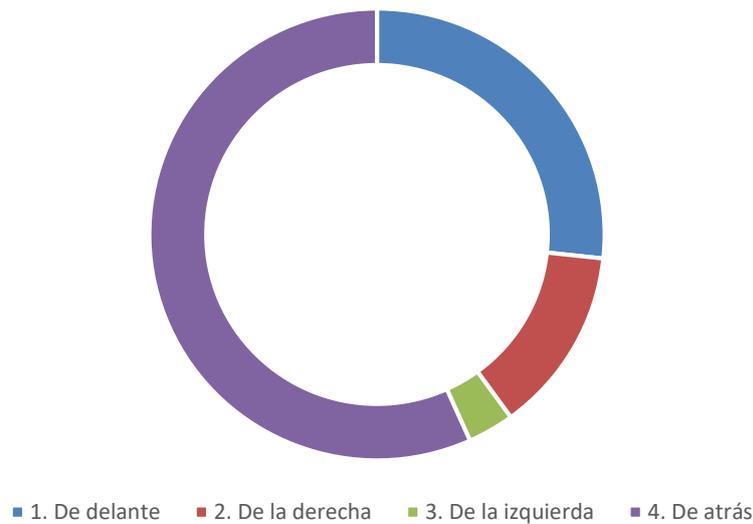


Por tanto, en este caso, la tasa de acierto fue del **63,3 %**.

5.2.3.- Tercera pregunta, ruido de taladro por delante

Para esta pregunta, se pidió a los encuestados que escucharan el sonido de un taladro procesado para que parezca que proviene desde **en frente** del oyente (0, 0, 66). Se preguntó de dónde parecía que provenía el sonido y se ofrecieron las siguientes opciones con los resultados siguientes:

1. De delante: 26,7 %
2. De la derecha: 13,3 %
3. De la izquierda: 3,3 %
4. De detrás: 56,7 %

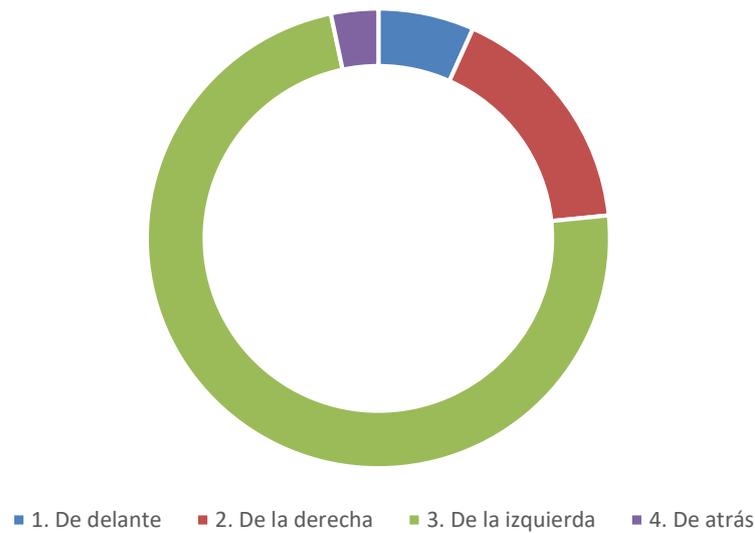


Por tanto, en este caso, la tasa de acierto fue del **26,7 %**.

5.2.4.- Cuarta pregunta, senoidal por el oído izquierdo

Para esta pregunta, se pidió a los encuestados que escucharan una onda senoidal procesada para que parezca que proviene desde la **izquierda** del oyente (270, 0, 73). Se preguntó de dónde parecía que provenía el sonido y se ofrecieron las siguientes opciones con los resultados siguientes:

1. De delante: 6,7 %
2. De la derecha: 16,7 %
3. De la izquierda: 73,3 %
4. De detrás: 3,3 %

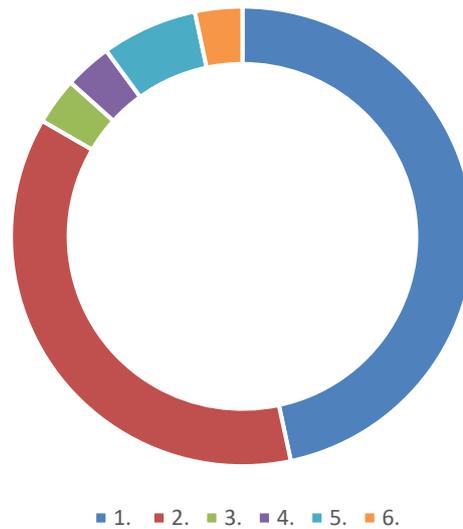


Por tanto, en este caso, la tasa de acierto fue del **73,3 %**.

5.2.5.- Quinta pregunta, taladro por la izquierda y ruido mecánico por la derecha

Para esta pregunta, se pidió a los encuestados que escucharan el sonido de un taladro procesado para que parezca que proviene desde la **izquierda** del oyente y un ruido mecánico por la **derecha**. Se preguntó de dónde parecía que provenían ambos sonidos y se ofrecieron las siguientes opciones con los resultados siguientes:

1. El ruido mecánico por la derecha y la taladradora por la izquierda: 46,6 %
2. El ruido mecánico por la izquierda y la taladradora por la derecha: 36,7 %
3. El ruido mecánico por delante y la taladradora por detrás: 3,3 %
4. El ruido mecánico por detrás y la taladradora por delante: 3,3 %
5. El ruido mecánico está ubicado por debajo de la línea de los oídos y la taladradora por encima: 6,7 %
6. El ruido mecánico está ubicado por encima de la línea de los oídos y la taladradora por debajo: 3,3%

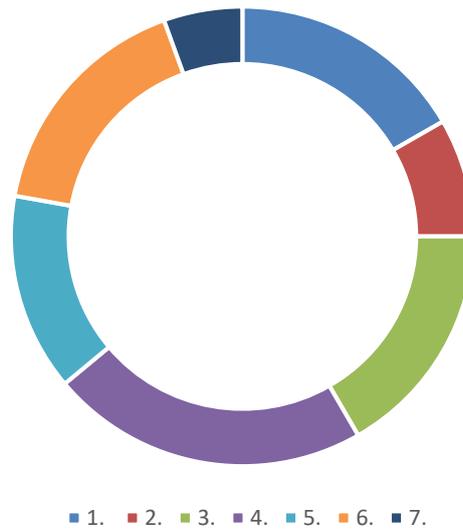


Por tanto, en este caso, la tasa de acierto fue del **46,6 %**.

5.2.6.- Sexta pregunta, taladro por detrás y ruido mecánico por delante

Para esta pregunta, se pidió a los encuestados que escucharan el sonido de un taladro procesado para que parezca que proviene desde **detrás** del oyente y un ruido mecánico por **delante**. Se preguntó de dónde parecía que provenían ambos sonidos y se ofrecieron las siguientes opciones con los resultados siguientes:

1. El ruido mecánico por la derecha y la taladradora por la izquierda: 20 %
2. El ruido mecánico por la izquierda y la taladradora por la derecha: 10 %
3. El ruido mecánico por delante y la taladradora por detrás: 20 %
4. El ruido mecánico por detrás y la taladradora por delante: 26,7 %
5. El ruido mecánico está ubicado por debajo de la línea de los oídos y la taladradora por encima: 16,7 %
6. El ruido mecánico está ubicado por encima de la línea de los oídos y la taladradora por debajo: 20%
7. Otras respuestas: 6,6%



Por tanto, en este caso, la tasa de acierto fue del **20 %**.

5.2.7.- Séptima pregunta, taladro por debajo de la línea de los oídos y ruido mecánico por encima

Para esta pregunta, se pidió a los encuestados que escucharan el sonido de un **taladro** procesado para que parezca que proviene de por **debajo de la línea de los oídos** del oyente y un **ruido mecánico por encima**. Se preguntó de dónde parecía que provenían ambos sonidos y se ofrecieron las siguientes opciones con los resultados siguientes:

1. El ruido mecánico por la derecha y la taladradora por la izquierda: 6,7 %
2. El ruido mecánico por la izquierda y la taladradora por la derecha: 6,7 %
3. El ruido mecánico por delante y la taladradora por detrás: 6,7 %
4. El ruido mecánico por detrás y la taladradora por delante: 10 %
5. El ruido mecánico está ubicado por debajo de la línea de los oídos y la taladradora por encima: 16,7 %
6. El ruido mecánico está ubicado por encima de la línea de los oídos y la taladradora por debajo: 23,3 %
7. Otras respuestas: 29,7 %

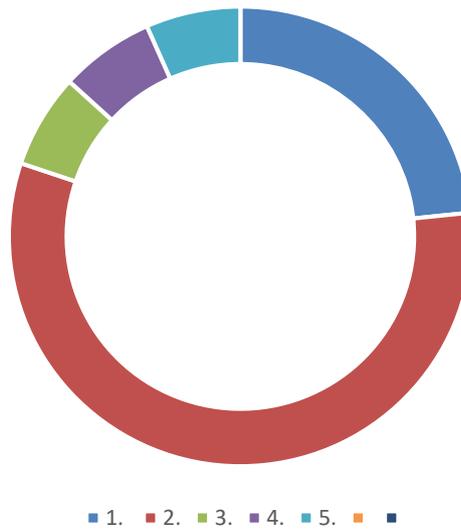


Por tanto, en este caso, la tasa de acierto fue del **23,3 %**.

5.2.8.- Octava pregunta, opiniones

En esta ocasión y para concluir la encuesta, se pidió a los encuestados que dieran su opinión de la aplicación del audio 3D en la industria respondiendo a la siguiente pregunta: “Después de haber contestado las preguntas, ¿le parece útil aplicar esta tecnología de audio en el ámbito industrial como una herramienta de inspección remota de máquinas? ¿Por qué?”

1. Sí, y puede constituir por ella misma una gran herramienta para la detección de problemas en las máquinas: 23,3 %
2. Sí, pero no siempre está claro de dónde provienen los sonidos, por lo que creo que contar con una referencia visual ayudaría: 56,7 %
3. No, me parece útil solo en casos muy concretos: 6,6 %
4. No me parece útil en absoluto: 6,6 %
5. No de una opinión: 6,6 %



Por tanto, al **56,7 %** les parece que esta aplicación puede ser útil en la industria, como método de supervisión a distancia, pero se vería reforzada por el uso de la imagen, ya que no siempre está claro de dónde procede el sonido.

6. Conclusión y trabajos futuros

Una vez expuestos los resultados, se puede concluir, gracias al análisis objetivo, que la aplicación funciona, que selecciona correctamente los tetraedros y proporciona una respuesta adecuada y coherente en base a las posiciones del tetraedro seleccionado. Sin embargo, el análisis de los resultados de la encuesta cuestiona el funcionamiento correcto de la aplicación para algunas posiciones. Se aprecia cierta confusión de los oyentes entre sonidos provenientes de delante y detrás, así como en la diferenciación de la procedencia de los sonidos cuando hay varios simultáneamente. En total, los encuestados fueron capaces de responder correctamente el 46,7 % de las veces, casi en la mitad de las ocasiones. Para las preguntas en las que los sonidos provenían de la izquierda o la derecha lo hicieron correctamente en el 66,63 % de las veces, mientras que cuando venían desde delante o detrás u otras posiciones, respondieron correctamente el 33,32 % de las veces. Este comportamiento se puede explicar con lo ya comentado en la sección 5.1; el hecho de no estar habituado a un determinado sonido, dificulta saber discernir entre las posiciones que no están claramente orientadas a uno de los oídos.

Por otra parte, el desplazamiento de la cabeza es un factor que ayuda a identificar el sonido. Por ejemplo, para determinar con claridad de dónde procede un sonido, un ser humano intentará orientarse de manera correcta hacía la fuente emisora del mismo, y en el proceso de hacerlo, recibirá información que habilitará al cerebro a calcular la procedencia del mismo. Esta capacidad, no está implementada en la aplicación, por lo que la tasa de aciertos puede verse disminuida por ello. Otro de los problemas para una correcta percepción proviene de la diferencia en la capacidad auditiva de los individuos. Por poner un ejemplo extremo, alguien que oiga mucho mejor de un oído que de otro, podría llegar a decir que un sonido proveniente del lado izquierdo, proviene a su parecer del derecho, ya que este oído es mucho más sensible que el izquierdo en su caso. Esto se traduce en la existencia de diferentes respuestas a impulso que modelan la percepción sonora para cada persona. Puede darse el caso de personas cuya percepción auditiva sea muy similar a la de la dummy head utilizada para crear la base de datos, que hayan respondido con facilidad a las cuestiones propuestas; como pudo haber casos en los que la percepción del individuo no tuviera nada que ver, habiendo fallado todas las cuestiones.

Es por ello, que como posibles mejoras de la aplicación se podrían incluir:

- Introducción de un método de control de la orientación de la cabeza del oyente en tiempo real. Esto podría conseguirse de diferentes formas, por ejemplo, mediante la asociación de la orientación de la cabeza al ratón o a alguna tecla; o bien mediante la utilización de un giroscopio que se incorporaría a los auriculares del usuario.
- Personalización de la HRIR. Existe material científico [14] que describe métodos de adaptación de la HRTF dependiendo del usuario.
- Teniendo en cuenta la primera ampliación del trabajo, esta conllevaría una optimización de software y una adaptación para su ejecución en tiempo real. Para ello, sería interesante la utilización de diversos hilos de procesamiento.
- Incorporación de una interfaz gráfica.
- Aplicación de técnicas de interpolación de respuestas a impulso para el far-field, es decir, las posiciones a mayor distancia de dos metros del centro de la cabeza del oyente.
- Control de la intensidad de las fuentes sonoras.

Tras el estudio de los resultados y de los análisis efectuados, se puede concluir que la inclusión de la tecnología de audio 3D para la supervisión de máquinas a distancia podría mejorar la percepción de la persona encargada de realizar esta operación. Se ha comprobado, que casi en la mitad de los casos, la percepción es correcta incluso en personas que no están familiarizadas con los sonidos presentados, siendo este un dato que podría mejorarse con la introducción de nuevas funciones software. Como se ha señalado, existen diversas opciones que harían que la aplicación mejorase, pero sería necesario verificar de nuevo su validez. Así mismo, sería indispensable realizar validaciones en plantas industriales, con los expertos pertinentes. Esto implicaría la obtención de datos en tiempo real del sonido de las máquinas a través del montaje de un equipo de sonido adecuado para recoger el ruido de las máquinas deseadas. No obstante, la utilización de audio 3D tiene un gran atractivo para un análisis veloz, sobre la marcha, siendo utilizada por un experto, por lo que su rentabilidad es discutible en el caso de la supervisión

industrial donde ya existen métodos basados en inteligencia artificial para la previsión de fallos que no dependen de la intervención humana.

7. Referencias

- [1] Pérez, D. (2014) “El inventor olvidado que pudo cambiar la historia de la música con la holofonía” Recuperado de: <https://elultimodiadesocrates.wordpress.com>
- [2] (2003) "Holofonía" Recuperado de: <https://es.wikipedia.org/wiki/Holofonía>
- [3] Tianshu Qu, Zheng Xiao, Mei Gong, Ying Huang, Xiaodong Li, and Xihong Wu. (2009) “Distance-Dependent Head-Related Transfer Functions Measured With High Spatial Resolution Using a Spark Gap”
- [4] Thibaut Carpentier, Hélène Bahu, Markus Noisternig, Olivier Warusfel. (2014) “Measurement of a head-related transfer function database with high spatial resolution.”
- [5] Wierstorf, H., Geier, M., Raake, A., Spors S., (2011) “A Free Database of Head-Related Impulse Response Measurements in the Horizontal Plane with Multiple Distances”
- [6] Altozano, J. (2018) “Qué es la Música 8D y por qué se ha hecho viral” Recuperado de: <https://www.youtube.com/watch?v=e6Ekz7ZDV-w>
- [7] W. Smith, Steven. “DSP Guide”
- [8] “Coordenadas esféricas”. Recuperado de https://es.wikipedia.org/wiki/Coordenadas_esféricas
- [9] “Efectos de audio 3D”. Recuperado de https://es.wikipedia.org/wiki/Efectos_de_audio_3D
- [10] Hugeng, H., Anggara J., Gunawan, D. (2016) “Implementation of 3d HRTF interpolation in synthesizing virtual 3d moving sound”
- [11] Gamper, H. (2013) “Head-related transfer function interpolation in azimuth, elevation, and distance”
- [12] Samet, H. (1989) “Implementing ray tracing with octrees and neighbor finding,”
- [13] Gamper, H. (Última revisión 31/10/ 2013). “3-D HRTF interpolation,” <http://www.mathworks.com/matlabcentral/fileexchange/43809>
- [14] Kazuhiro, I., Yohji, I. and Nishioka, S. (2013) “Personalization of head-related transfer functions in the median plane based on the anthropometry of the listener’s pinnae”

