



PONTIFICIA  
UNIVERSIDAD  
CATÓLICA DE  
VALPARAÍSO



**Nelson Manuel Jara Donoso**

# Implementación y Simulación de Audio 3D en Raspberry Pi 2

Informe Proyecto de Título de Ingeniero Civil Electrónico



**Escuela de Ingeniería Eléctrica  
Facultad de Ingeniería**

Valparaíso, 28 de mayo de 2018



# Implementación y Simulación de Audio 3D en Raspberry Pi 2

Nelson Manuel Jara Donoso

Informe Final para optar al título de Ingeniero Civil Electrónico,  
aprobada por la comisión de la  
Escuela de Ingeniería Eléctrica de la  
Facultad de Ingeniería de la  
Pontificia Universidad Católica de Valparaíso  
conformada por

Sr. Sebastián Fingerhuth Massmann

Profesor Guía

Sr. Francisco Pizarro Torres

Segundo Revisor

Sr. Jorge Mendoza Baeza

Director de Escuela

Valparaíso, 28 de mayo de 2018

# Agradecimientos

En primer lugar, quiero dar las gracias a mis Padres y Hermana. Nancy, Nelson y Paula, por todo el apoyo, esfuerzo y amor incondicional que me han entregado durante toda mi vida y en especial en este proceso universitario.

También agradecer a Valeria, mi polola, por su reconfortante compañía en esta etapa y el apoyo en estos últimos semestres.

Un especial agradecimiento a mis compañeros y amigos de universidad, hogar y Rancagua, por todos los momentos vividos, su hermandad y el apoyo en este tiempo lejos de casa.

A mis profesores guías Sebastián Fingerhuth y Francisco Pizarro, quienes me aconsejaron y otorgaron las herramientas necesarias durante la realización del presente proyecto y, además, darme la oportunidad de aprender aún más.

*Valparaíso, 28 de mayo de 2018*

Nelson Jara Donoso

# Resumen

El siguiente trabajo consiste en la implementación y simulación de un sistema de audio 3D. Se desarrollan conceptos, metodología y sistemas que permitan conocer el sonido tridimensional y sus fundamentos. Esto permitirá comprender de qué manera nuestro sistema auditivo capta las señales acústicas, las calibra y localiza en el plano tridimensional real.

La problemática consiste básicamente en implementar y simular un sistema de Audio 3D en un dispositivo de bajo costo, desarrollando un programa en un software libre.

Se desarrolla el trabajo, tomando como referencia conceptos que son usados en la reproducción de audio 3D, tales como, la percepción del sonido, sistemas espaciales y funciones de transferencia relacionadas con la cabeza (Head Related Transfer Function).

Se complementa con los métodos de reproducción más esenciales en audio 3D, sus ventajas e inconvenientes. Se define la problemática del tema de proyecto y las posibles soluciones.

El hardware utilizado para la simulación es la Raspberry Pi 2, montada con una tarjeta de audio Cirrus Logic y el software de programación Pure Data. Se presenta un estudio general y previo de los elementos que la componen, junto con los detalles de cada proceso de instalación y configuración. También se presenta un acercamiento con la plataforma Pure Data, dando a conocer todos sus elementos básicos.

Se muestran los programas en Pure Data, en la versión 0.47 con sus respectivos programas, objetos externos y la adecuada explicación para comprender su funcionamiento. Se desarrolla un sistema de audio 3D, los conceptos que son claves en el proceso y que son parte del sistema.

Con el desarrollo de los diagramas del sistema, se logrará el entendimiento de los programas que son parte del software, estos resultan fundamentales en el funcionamiento del objeto principal. Se exponen los detalles del procedimiento que permitirá llevar a cabo las pruebas realizadas y respaldar los resultados que se mostraran gráficamente.

Palabras claves: Raspberry Pi 2, Pure Data, Convulación, Sistema de auralización espacial, HRTF, Transformada Discreta de Fourier, Sonido binaural.

# Abstract

The following work consists of the implementation and simulation of a 3D audio system. Concepts, methodology and systems are developed that allow to know the three-dimensional sound and its foundations. This will allow us to understand how our auditory system captures the acoustic signals, calibrates them and locates them in the real three-dimensional plane.

The problem consists basically of implementing and simulating a 3D Audio system in a low cost device, developing a program in a free software.

The work is developed, taking as reference concepts that are used in the reproduction of 3D audio, such as, the perception of sound, spatial systems and functions of transfer related to the head (Head Related Transfer Function).

It is complemented with the most essential reproduction methods in 3D audio, its advantages and disadvantages. The problem of the project topic and the possible solutions are defined.

The hardware used for the simulation is the Raspberry Pi 2, mounted with a Cirrus Logic audio card and the Pure Data programming software. A general and previous study of the elements that compose it is presented, together with the details of each installation and configuration process. It also presents an approach with the Pure Data platform, revealing all its basic elements.

The programs are shown in Pure Data, in version 0.47 with their respective programs, external objects and the adequate explanation to understand their operation. A 3D audio system is developed, the concepts that are key in the process and that are part of the system.

With the development of the diagrams of the system, the understanding of the programs that are part of the software will be achieved, these are fundamental in the operation of the main object. The details of the procedure that will allow carrying out the tests carried out and supporting the results that are shown graphically are exposed.

Keywords: Raspberry Pi 2, Pure Data, Convolution, Spatial auralization system, HRTF, Discrete Fourier Transform, Binaural sound.

# Índice general

Introducción .....	1
Objetivo General .....	4
Objetivos Especificos .....	4
<b>1 Definición de la Problemática .....</b>	<b>5</b>
1.1 Contexto de trabajo.....	5
1.1.1 Sistema auditivo periférico .....	5
1.1.2 Localización .....	6
1.1.3 Función de Transferencia de la Cabeza (HRTF) .....	11
1.1.4 Acústica del entorno – reverberación .....	12
1.2 Antecedentes generales .....	13
1.2.1 Sistema espacial estereofónico .....	13
1.2.2 Reconstrucción Binaural.....	14
1.2.3 Reconstrucción del campo de onda.....	17
1.3 Método y teoría.....	18
1.3.1 Transformada Discreta de Fourier .....	18
1.3.2 Convolución por medio de la DFT .....	19
1.3.3 Convolución circular .....	20
<b>2 Solución y marco teórico.....</b>	<b>22</b>
2.1 Raspberry Pi 2 Model B.....	22
2.2 Tarjeta de Audio Cirrus Logic.....	23
2.3 Pure Data .....	24
2.3.1 Configuración y elementos de Pure Data.....	26
2.4 Sistema de auralización .....	27
2.5 Sistema de convolución.....	28
2.5.1 Convolución de dos archivos de sonido pregrabados .....	29
2.5.2 Convolución con entrada en tiempo real .....	29
2.6 Base de datos de mediciones HRTF .....	30
<b>3 Desarrollo .....</b>	<b>32</b>
3.1 Estructura del Programa .....	32
3.2 Construcción del Programa.....	33
<b>4 Resultados .....</b>	<b>37</b>
4.1 Pruebas.....	37
4.2 Resultados obtenidos.....	40
Discusión y conclusiones .....	44
Bibliografía.....	47

# Introducción

Los humanos en general poseen distintas capacidades para percibir y asimilar su entorno, estas capacidades se denominan mecanismos fisiológicos, que usualmente se les conoce como sentidos. El ser humano tiene una gran capacidad para percibir el entorno a través de la audición. Aunque la visión aporta mucha más información que la audición, está limitada a una dirección frontal. Es el sentido del oído (audición) el que es capaz de detectar, localizar e identificar las fuentes sonoras en cualquier dirección alrededor del oyente. También es capaz de estimar las propiedades acústicas del ambiente físico en el que estemos situados, como por ejemplo saber si hay una pared cerca o estimar el tamaño de algún recinto.

Lo anteriormente mencionado es muy importante para el desarrollo de nuevas tecnologías, como la realidad virtual. Al pensar en una realidad virtual, cuesta imaginar que el ser humano pueda llegar a sumergirse en ella sintiendo lo mismo que en el mundo real, sin contar con una tecnología que reproduzca, lo mejor posible, un entorno acústico en el que sea posible localizar fuentes sonoras en todo el espacio con el que se interactúa. También, está el hecho de conseguir que el oyente identifique la posición de las fuentes sonoras, pudiendo realizar movimientos de desplazamiento como de cabeza, mientras se conserva el mismo entorno acústico. Este concepto de realizar movimientos conservando el entorno acústico se conoce como sonido direccional. En el año 1930 se realizaron los primeros trabajos sobre la reproducción de sonido direccional, en los laboratorios Bell. Desde este acontecimiento, los ingenieros e investigadores de sonido han estado interesados en una reproducción que sea capaz de recrear un campo físico, que represente la distribución escalar del nivel de presión sonora en cada punto del espacio tridimensional, lo que se conoce como campo sonoro.

Uno de los primeros avances importantes en la recreación de un campo sonoro, es la reproducción de sonido por medio de dos canales independientes (izquierda y derecha); esto se conoce como formato estéreo. Otro descubrimiento importante, que vino después del desarrollo del formato estéreo, fue la síntesis de campo de onda explicada por Berkhout [6], a fines de los años 80. Esta consiste en un método de reproducción de sonido basado en los fundamentos básicos de propagación de ondas.

Para comprender la variedad de descubrimientos que se puedan encontrar con respecto al tema, es necesario dominar muy bien los conceptos y terminologías relacionados con el sonido. Se puede inferir que, para implementar una correcta aplicación de un sonido tridimensional, se debe optimizar la eficacia de los recursos, debido a que cada persona posee distintas características físicas y por ende escucha distinto. Suponiendo que una persona se encuentra en medio de una sala con varias personas más interactuando y conversando alrededor, el oyente podría identificar, con los ojos cerrados, de

donde provienen las voces y también percibir de que dirección viene el ruido ocasionado por el cierre de una puerta o cualquier otra fuente sonora. El objetivo de un sistema de audio 3D es poder recrear esas fuentes de sonido de manera virtual, generando la sensación de que los sonidos llegan desde una posición donde realmente no existe ninguna fuente acústica.

En los últimos años se ha desarrollado un gran número de sistemas de audio multicanal, siempre con el objetivo de conseguir una mayor espacialización del sonido. Con el pasar del tiempo estos sistemas han ido evolucionando, a medida que evoluciona la tecnología utilizada en estos. Con la aparición del sonido inmersivo (sistema de audio para crear la percepción real de que los sonidos provienen de distintos puntos del espacio), se han podido utilizar un gran número de altavoces distribuidos en un espacio tridimensional en el que cada uno de ellos conlleva una señal independiente, pudiendo posicionar un sonido en casi cualquier parte del espacio. Todo esto es posible gracias a la tecnología que permite el procesamiento digital de señales (DSP – Digital Signal Processing) a una mayor velocidad de operación.

Al igual que el sonido inmersivo, la simulación de audio 3D trata de ubicar una fuente sonora en cualquier punto del espacio, por ende, la realización de un simulador de audio 3D en tiempo real implica una tecnología que permita realizar el procesamiento a gran velocidad [9]. Es fundamental en este proyecto investigar y estudiar a fondo el proceso de simulación de audio binaural. El audio binaural consiste en crear para el oyente una sensación de sonido 3D, similar a la de estar físicamente en el lugar donde se producen los sonidos, siendo grabado mediante el uso de dos micrófonos en una cabeza artificial. De esta forma, se puede saber que método resulta mejor para luego implementar un sistema de audio tridimensional.

El siguiente trabajo es de tipo investigativo y de simulación. A medida que se desarrolla con claridad la temática, se incorpora la construcción de un sistema de audio binaural que consiste en la implementación de un programa basado en el software Pure Data, todo esto montado en una Raspberry Pi 2. Para realizar esto, es fundamental el estudio previo, para así tener una comprensión clara y precisa acerca del fundamento teórico para desarrollar este sistema. Se abordarán algunos conceptos y propiedades esenciales con respecto al sonido y su percepción.

Se puede entender la percepción como el primer conocimiento de una cosa por medio de las impresiones que comunican los sentidos. Así, la percepción es la organización y reconocimiento de una o más experiencias sensoriales. Para entender estas sensaciones, los procesos perceptibles tienen que ser, por una parte, capaces de componer mentalmente un esquema espacial por medio de la extracción de información de varios estímulos. Por otro lado, también deben ser capaces de reagrupar y ordenar estos estímulos, tanto en el espacio como en el tiempo. Estos procesos de la percepción ocurren en nuestro cerebro, por lo tanto, es bastante más complejo describirlos fisiológicamente que los órganos sensoriales.

Por otro lado, se puede definir la atención como el proceso de discriminación en la elección del cerebro, entre todos los estímulos que deben pasar a una etapa superior, que consiste en el procesamiento de información. De esta manera, se puede centrar nuestra atención en un estímulo e ignorar otros. El reconocimiento de objetos y la atención están relacionados con la comprensión de los estímulos y la percepción tridimensional.

Con respecto a la sensación del oyente frente a una experiencia subjetiva de un suceso sonoro, se consideran los siguientes factores: naturaleza de la fuente sonora, comportamiento del espacio y de las reflexiones acústicas; influencia de la oreja, cabeza y torso (hombros); finalmente la experiencia previa del oyente en la localización de fuentes de sonido.

Al considerar y comprender todo lo anterior, surge la siguiente pregunta: ¿Cómo funcionan los sistemas de audio 3D? Para responder a esta pregunta, se parte por examinar como los humanos pueden localizar sonidos utilizando solo dos canales para su recepción (los oídos). Una perturbación acústica generada en el espacio crea una onda de sonido que se propaga a los oídos del oyente. Ésta, al interactuar con el sistema auditivo, hará que se intuya el entorno acústico del espacio tridimensional en el que se encuentra el oyente.

En este proyecto, se comienza por estudiar el proceso de simulación de audio binaural, utilizando para ello una Raspberry Pi 2. La Raspberry Pi 2 es un ordenador de tamaño reducido y de bajo costo, que actúa como un DSP (Digital Signal Processor). Este permite el procesamiento en tiempo real de la señal de audio. Para optimizar la entrada y salida de audio de la Raspberry Pi, se incorpora una tarjeta de audio de alta calidad en el dispositivo mencionado, en este caso una Cirrus Logic Audio Card. Este conjunto de elementos conforma el hardware en el que se realiza la implementación. Para realizar la simulación, se utiliza el software de programación Pure Data; este consiste en un lenguaje de programación gráfico que permite ejecución en tiempo real, actuando como un DSP en el que se recoge la entrada de audio, se procesa y se envía a la salida, todo en tiempo real.

La Raspberry Pi posee una capacidad de cálculo relativamente buena, por menos dinero y utilizando un espacio muy reducido, debido a su pequeño tamaño. Estos tres factores hacen que sea interesante y atractivo su uso como alternativa para el procesamiento digital de señales. Se ha demostrado que, hasta cierto punto, y de una forma relativamente simple, es posible realizar tareas de DSP en la Raspberry Pi.

La implementación de un sistema de sonido 3D, en tiempo real, implica una tecnología que permita realizar el procesamiento digital a gran velocidad. El siguiente proyecto consiste en implementar un procesamiento digital de señales en tiempo real utilizando, principalmente, el software Pure Data instalado en la Raspberry Pi 2. El sistema realiza la convolución de dos señales de audio, con el fin de generar un efecto de espacialización, requerido para localizar una fuente sonora en el espacio.

Para generar el efecto auditivo de una fuente sonora localizada en un punto del espacio tridimensional, se deben realizar dos convoluciones: una entre la respuesta al impulso captada por el oído izquierdo, que proviene de dicho punto a localizar, con una señal de audio seco (sin reverberación); otra entre la respuesta impulso captada por el oído derecho con la misma señal de audio seco. Para esto, se utiliza la base de datos de mediciones del maniquí KEMAR [8].

En este informe, se detalla el proceso de investigación referido al estado del arte de la temática. Además, se explica la construcción del sistema, implementado en la Raspberry Pi 2, utilizando para ello la tarjeta Cirrus Logic Audio Card y el software Pure Data. También, se presentan las pruebas realizadas, la forma en que estas fueron llevadas a cabo las y sus resultados correspondientes, con sus respectivos inconvenientes. Finalmente se muestran las conclusiones de este proyecto.

## Objetivo General

- El siguiente trabajo es parte del proyecto “Método híbrido para la obtención de funciones de transferencia relacionadas con la cabeza (HRTF): moldeo de oreja y reconstrucción 3D fotogramétrica de la cabeza y torso”, tiene como objetivo estudiar el funcionamiento de distintos métodos y/o protocolos de comunicación de audio en tiempo real, para uso en simulaciones acústicas de audio 3D y realidad virtual.

## Objetivos Específicos

- Estudiar que son las HRTF y las características que deben tener para poder usarlas en reproducción de audio 3D.
- Estudiar y comparar las tarjetas de audio del laboratorio desde un punto de vista de la capacidad de convolución, filtro en tiempo real y con coeficientes dinámicos.
- Implementar una interfaz de usuario.
- Realizar experimentos, pruebas de funcionamiento y convolución en tiempo real para la caracterización.

# 1 Definición de la Problemática

Para comenzar, se presenta un estudio referido al contexto del trabajo a desarrollar, basado en el entendimiento de la percepción del sonido y cómo funciona el sistema auditivo periférico. Se presenta también la definición de la problemática, que consiste en la revisión de los sistemas implementados hasta el momento. También, los distintos métodos utilizados para implementar el sistema y la teoría acerca de la función convolución. Un buen dominio de este tema proporcionará una base sólida para poder desarrollar un sistema de audio 3D.

## 1.1 Contexto de trabajo

Para poder construir un sistema de audio binaural, resulta fundamental realizar una revisión bibliográfica que sirva de base para un estado del arte sobre el sonido tridimensional, permitiendo así una comprensión más clara y precisa acerca del contenido que se abordará al desarrollar el sistema. A continuación, se tratarán algunos conceptos y propiedades esenciales respecto al sonido.

### 1.1.1 Sistema auditivo periférico

La mayoría de los seres humanos posee la capacidad natural de percibir sonidos en 3D, esto permite reconocer la posición de las fuentes de sonido, siendo esta una cualidad potente para ubicarse en el espacio y sobrevivir. En la percepción acústica del ser humano, existe un elemento muy importante: el pabellón auricular. Este se ubica en el borde externo del conducto auditivo, actuando como un receptor de señales acústicas.

Las cavidades del pabellón auricular amplifican algunas frecuencias y, al mismo tiempo, su estructura física produce interferencias, las cuales pueden ser destructivas y constructivas entre sí. Como las características físicas de la oreja son diferentes en todas las personas, la experiencia auditiva es distinta para cada individuo.

Como se mencionó anteriormente, los procesos fisiológicos relacionados con el sonido son causados por el sistema auditivo periférico. Estos procesos permiten captar el sonido y transformarlo en impulsos eléctricos, que son enviados al cerebro a través de nervios auditivos. Este sistema biológico de la oreja está compuesto por el oído externo, el oído medio y el oído interno.

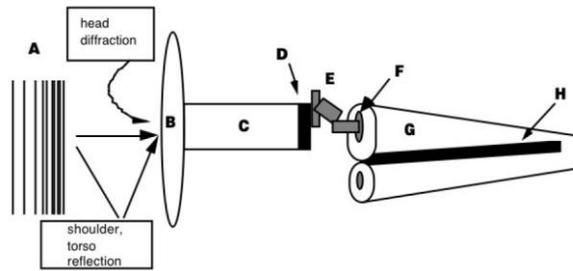


Figura 1-1: Esquema del sistema auditivo [11].

El oído externo se compone por el pabellón (letra B en Figura 1-1), que concentra las ondas sonoras en el conducto y el conducto auditivo externo (C) que termina en el tímpano (D). La no linealidad de las funciones de transferencia del oído (respuesta de frecuencia) comienza en el pabellón, que, debido a sus características, tiene una frecuencia de resonancia entre los 4.5 [kHz] y los 5 [kHz]. El canal auditivo externo tiene en promedio unos 2.7 [cm] (de longitud y un diámetro promedio de 0,7 [cm]), este se considera como un tubo cerrado en el que oscila una columna de aire, cuya frecuencia de resonancia del canal es de 3.2 [kHz] aproximadamente.

El oído medio está compuesto por el tímpano (D), los osículos (martillo, yunque, estribo y pequeños huesos) y la trompa de Eustaquio (E). La función de los osículos es transmitir el movimiento del tímpano al oído interno a través de la membrana conocida como ventana oval (F). El oído externo canaliza la energía acústica, mientras que el oído medio la transforma en energía mecánica transmitiéndola y amplificándola hasta el oído interno. En este se realiza la definitiva transformación en impulsos eléctricos.

El laberinto óseo es una cavidad que contiene a la cóclea o caracol (G). Dentro del laberinto se encuentra el laberinto membranoso, los ductos semicirculares y el ducto coclear (H), este último es el único que cumple una función en la audición, los otros se desempeñan en nuestro sentido de equilibrio.

### 1.1.2 Localización

La localización del sonido se basa en el procesamiento neuronal de las señales acústicas. El cerebro aprende y calibra estas señales acústicas, utilizando los datos de precisión espacial de los otros sistemas sensoriales; de esta manera, le es posible determinar la posición de un sonido.

Para demostrar la existencia de la calibración espacial, se realizaron investigaciones modificando el oído externo de personas, mediante moldes. Aunque la localización espacial de la elevación del sonido fue degradada inmediatamente después de la modificación, al tiempo se vuelve a adquirir. El aprendizaje de las nuevas señales espectrales no interfirió con la representación neuronal de las señales originales. De esta forma, las personas podían localizar sonidos con los dos oídos: modificado y normal [2]. En la figura 1-2 se ilustra el sistema de referencia.

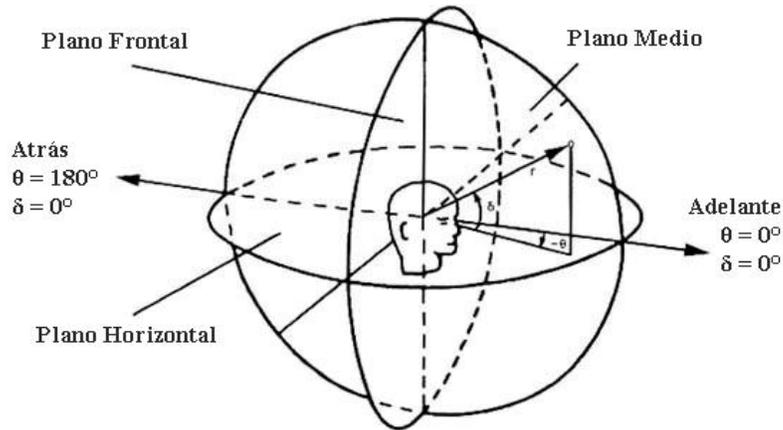


Figura 1-2: Sistema de referencia [10].

### Lateralizar (Teoría Dúplex)

La lateralización consiste en el proceso de convertir en lateral un sonido que no lo es; es decir, al escucharlo con una oreja, este se perciba distinto en comparación a escucharlo con la otra, con el fin de poder diferenciar si la fuente de lo que se oye, con ambos oídos, está más orientada al lado derecho o al izquierdo.

Para la lateralización se utilizó la teoría dúplex, que consiste en un modelo para estimar la ubicación espacial mediante dos señales binaurales, estas son: ITD (Diferencia Interaural de Tiempo) y ILD (Diferencia Interaural de nivel sonoro).

La ITD es la diferencia que toman los distintos tiempos de llegada del frente de sonido a cada uno de nuestros oídos. Esto es posible ya que la distancia entre los dos oídos es de aproximadamente 20 [cm].

La ILD corresponde a la diferencia que toman las distintas intensidades captadas por ambos oídos. La cabeza tiende a oponerse al paso de la onda de sonido; se trata de una característica conocida como "efecto de sombra" de la cabeza. Así pues, se presenta una diferencia interaural de nivel sonoro, que refleja la diferencia en la intensidad de la onda que llega a cada oído. Sin embargo, este índice tiene una dependencia importante de la frecuencia del sonido emitido. Para tonos por debajo de 1500 [Hz], la ILD prácticamente no existe. No obstante, para las frecuencias superiores a 1500 [Hz], el ILD se convierte en un índice eficaz.

Estos dos parámetros sirven para la localización de un sonido en la dirección 'izquierda-derecha'. El sonido es percibido hacia un lado dependiendo de la onda que llegue primero, mayor ITD significa mayor desplazamiento lateral. Cabe señalar que, desde 1500 [Hz] hacia abajo, la longitud de onda es comparable al diámetro de la cabeza, por lo tanto, la ITD empieza a ser confusa. La cabeza hace sombra en el oído más alejado de la fuente de sonido, esto significa que menos energía llega a este oído. Las ILD son esenciales para localizar sonidos con frecuencias por encima de los 1500 [Hz]. En la figura 1-3 se presenta un esquema de las diferencias interaurales.

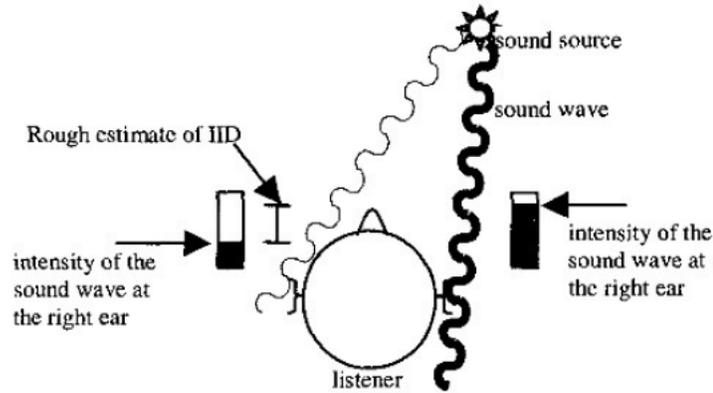


Figura 1-3: Esquema de las ILD [4]

### Efecto Haas

Cuando dos o más fuentes están emitiendo prácticamente el mismo contenido auditivo, un concepto clave para comprender el mecanismo de localización de un sonido, por parte del cerebro humano, es el efecto precedencia o efecto Haas. El efecto Haas consiste en un fenómeno psicoacústico descubierto en 1949 por Helmut Haas. Este indica las consecuencias que tiene en la percepción el hecho de que un sonido llegue con retraso.

Si se tienen dos fuentes sonoras, situadas de forma simétrica respecto al plano medio del oyente, emitiendo un mismo sonido, con la misma intensidad, en la cabeza de este se forma una imagen sonora que sitúa el sonido en el punto medio entre ambas fuentes. No obstante, si uno de los dos sonidos llega con un retardo respecto al otro, incluso aunque estos tengan la misma intensidad, el cerebro lo interpretará como que la fuente que lo provoca está más cercana al oído que capta primero el sonido. Este efecto también puede conseguirse reduciendo en unos cuantos decibelios el nivel de una de las dos fuentes.

Sin embargo, tampoco vale cualquier retardo. Si este es demasiado grande, se escucharán dos sonidos claramente distintos; si es excesivamente pequeño, seremos incapaces de percibir la diferencia. Es recomendable que el retardo esté situado entre 5 y 30 [ms] para poder apreciarlo bien [9]. La relación de retardo e intensidad se puede ver de forma gráfica en la figura 1-4.

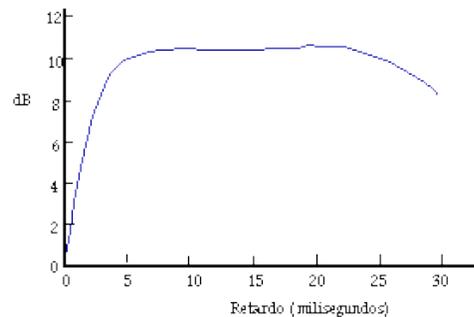


Figura 1-4: Diferencia en amplitud de la percepción entre dos sonidos iguales reproducidos a diferentes tiempos entre ellos [12].

En la figura 1-4, se observa que, aproximadamente después de los 5 [ms] de retraso, la magnitud alcanza los 10 [dB] y se mantiene relativamente constante. Antes de los 5 [ms], la magnitud tiene un valor pequeño, lo cual dificulta la percepción del sonido. Se concluyó que, para que el sonido retardado aparente tener la misma fidelidad que el no retardado, el primero debe superar al segundo en unos cuantos decibelios de amplitud, para compensar la ventaja que supone el haber llegado primero.

La teoría dúplex es sencilla y efectiva, pero solo explica la percepción en el plano azimutal. La localización del sonido está comprendida también en la elevación y la distancia, las ITD y ILD no especifican una única posición. Existen infinitas posiciones a igual distancia del oyente, que tienen asociadas las mismas ITD y/o ILD. Esa serie de puntos infinitos son llamados “cono de confusión”, el cual se ilustra en la figura 1-5.

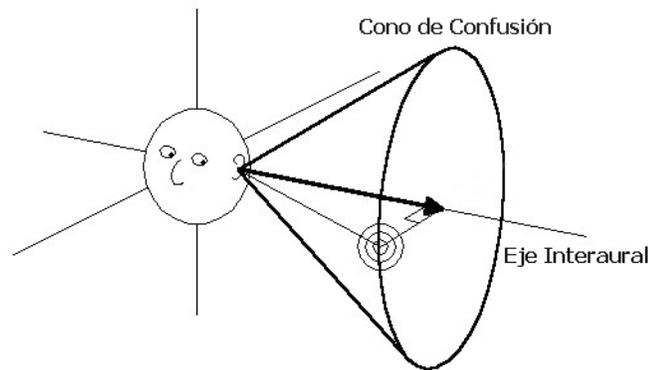


Figura 1-5: Cono de confusión [10].

Este inconveniente se hace más notorio en el plano medio, donde las ILD y ITD de cualquier fuente de sonido originada en cualquier punto de este plano son cero. Por lo tanto, la información de las diferencias es mínima.

### **Elevación**

La determinación del ángulo lateral es binaural, es decir, se realiza simultáneamente con los dos oídos, pero la determinación de la elevación es monoaural. Podemos percibir la elevación con tan solo un oído, ya que nuestra oreja actúa como un resonador acústico. Es más, su respuesta en frecuencia depende de la dirección, y para la correcta localización de elevación en el plano medio, el sonido debe contener frecuencias de 6 [kHz] o mayores, como se observa en la figura 1-6.

Además de las señales de elevación estática, proporcionadas por el pabellón auricular, la cabeza y el torso, los movimientos de cabeza también dan señales de elevación. El movimiento rotatorio de la cabeza provoca diferencias interaurales, fundamentales para determinar si una fuente de sonido se encuentra en frente o detrás. Una rotación de la cabeza provoca un índice máximo de cambio en la ITD y las fuentes de ILD en el plano horizontal, además de solucionar el problema de la percepción auditiva frontal y trasera.

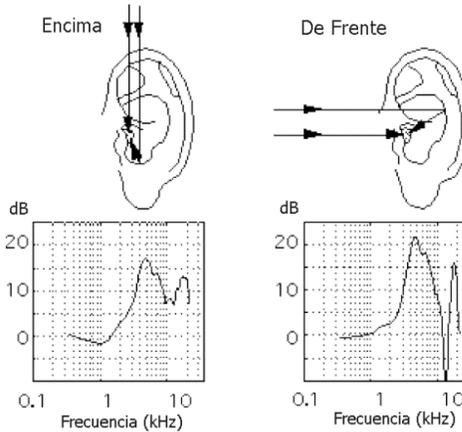


Figura 1-6: Esquema de la diferencia de respuesta de frecuencia variando la localización de la fuente [9].

## Distancia

Al intentar localizar una fuente sonora, se comprenden muy bien los mecanismos para determinar el ángulo lateral de la dirección donde proviene el sonido, no así los que estiman la elevación. Es más difícil aún entender el mecanismo que nos dice a qué distancia está la fuente. Para determinar la distancia se tienen en cuenta los siguientes mecanismos:

- Intensidad (Loudness)
- Movimiento de la cabeza (Head-Tracking)
- Exceso en las diferencias interaurales de intensidad (ILD)
- La razón entre el sonido directo y el sonido reverberante

El mecanismo de detección por intensidad se debe a que la energía sonora capturada, proveniente directamente de la fuente, decae inversamente con el cuadrado de la distancia, dependiendo del tipo de fuente. Es evidente que la energía recibida es proporcional a la energía emitida por la fuente, y que no puede haber una relación uno-a-uno entre la intensidad y la distancia. Reproducir un sonido a un volumen bajo, no produce por sí mismo la sensación de que la fuente esté muy lejos. En el caso de la voz humana, cada uno de nosotros sabemos la diferencia entre la característica del sonido asociada a un susurro, a una conversación normal o a un grito sin tener en cuenta el nivel sonoro, debido a que en esto cambia mucho el contenido espectral.

El movimiento de la cabeza se refiere a que, si el oyente la mueve, el cambio de azimut puede depender de la distancia. Para fuentes cercanas, un movimiento mínimo puede producir un gran cambio en el ángulo lateral, mientras que, para fuentes lejanas, esencialmente no hay cambios de azimut.

Las ILD pueden aumentar cuando una fuente se aproxima mucho a la cabeza, este aumento se hace patente para distancias menores a un metro, aproximadamente. Un caso extremo, por ejemplo, se da cuando alguien susurra en un oído.

El último mecanismo es la razón entre el sonido directo y el reverberante. El sonido es reflejado y dispersado muchas veces por las superficies del entorno, y la energía reverberante que llega a los oídos no cambia mucho con la distancia entre la fuente y el oyente. Esta razón es el mejor mecanismo para

determinar la distancia. Para distancias cortas, esta razón es muy grande, mientras que para distancias largas es pequeña.

### 1.1.3 Función de Transferencia de la Cabeza (HRTF)

Formalmente, la función de transferencia de la cabeza (HRTF: Head-Related Transfer Function) se define como la respuesta de frecuencia del oído izquierdo o derecho de un individuo, medida desde un punto específico en el campo libre, a un punto específico en el canal auditivo. Consideremos una fuente de ondas sonoras que es emitida desde una posición, esta es 'filtrada' por fenómenos de difracción y reflexión con nuestra cabeza, el pabellón de la oreja y el torso antes que llegue a nuestro tímpano. Este filtrado ayuda a nuestro sistema neuronal a percibir la posición de la fuente. El sistema define la función de transferencia como el cociente entre el espectro de la señal emitida y el espectro de la señal recibida.

Un método para obtener la función de transferencia de la cabeza (HRTF), de la localización de una fuente, es medir la respuesta al impulso relacionada con la cabeza (HRIR: Head Related Impulse Response) en el tímpano. La HRTF es una función bastante compleja, en la cual interviene la frecuencia y tres variables espaciales. El método más eficiente es medir el HRIR a un maniquí dummy o en nuestros propios oídos.

Típicamente, la HRTF es medida en una sala anecoica para reducir al mínimo la influencia de reflexiones tempranas y reverberación en la respuesta. Esta se mide en incrementos pequeños de 15° aproximadamente, tanto en el plano horizontal como el vertical, usando interpolación para posiciones arbitrarias [8].

Para comprender las diferencias espectrales, existen diferentes tipos de investigaciones, modelos físicos, mediciones empíricas y recientes simulaciones informáticas [13]. Esto, para conseguir una relación directa entre la dirección de la fuente de sonido, y la respuesta de frecuencia que recibe el oído.

Los Filtros de Respuesta de Impulso Finitos (FIR) son un tipo de filtros digitales en el que, si la entrada es una señal impulso, la salida tendrá un número finito de términos no nulos. La HRTF también se puede describir como filtros FIR de fase mínima, y también asume ambas diferencias interaurales (ITD y ILD). Los tiempos de retardo son codificados en los filtros de fase espectrales y la información de la ILD se relaciona como la potencia general de la imagen.

Empíricamente, se puede simplificar un filtro FIR en dos ramas:

- La hipótesis de fase mínima nos permite especificar una fase de la HRTF con una respuesta de magnitud.
- La hipótesis de fase mínima nos permite separar la información de las ITD de la especificación FIR de la HRTF.

La caracterización completa de las señales auditivas, asociadas con una localización espacial, supone la medición de las tres variables: las respuestas de la magnitud del oído izquierdo y derecho, y el ITD.

Los efectos de la elevación de la HRTF son, básicamente, producidos por la oreja. Las frecuencias que toman importancia para esta decodificación espacial son las comprendidas entre 6 y 8 [kHz] [14]. En la figura 1-7 se ilustra el esquema de la estructura de la oreja.

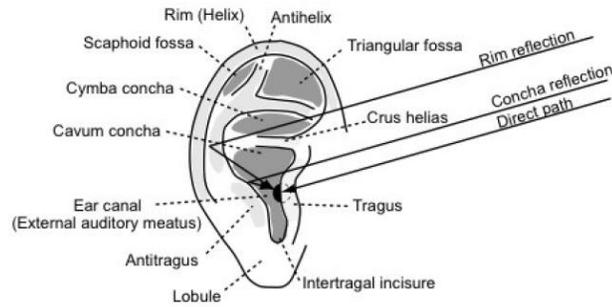


Figura 1-7: Esquema de la Oreja [14].

El relacionar una estructura anatómica particular y las características HRTF, forma parte de uno de los métodos para evitar perder tiempo y recursos al medir funciones de transferencia. Para simular un modelo antropométrico de nuestro sistema auditivo, se debe separar por partes el efecto producido por nuestro cuerpo. La función de transferencia de la oreja (PRTF) devuelve la señal intervenida por las reflexiones, producidas por las cavidades del pabellón. Los llamados modos resonantes son los principales elementos que modificarán la PRTF, por ende, a la HRTF. Gracias a estos modos se percibe la elevación. La investigación entre la relación de las medidas antropométrica de la oreja y el PRTF es realmente interesante, es la parte del sistema auditivo que más difiere de persona a persona.

#### 1.1.4 Acústica del entorno – reverberación

La reverberación es esencial para establecer el contexto espacial de una escena auditiva en una sala o espacio cerrado, es importante saber de qué manera percibimos el entorno donde se transmite una fuente. Da información sobre el tamaño y el carácter del espacio envolvente, que es muy útil para la percepción correcta de las distancias. En un espacio reverberante, cuando la distancia entre la fuente y el oyente aumenta, el nivel del sonido directo disminuye en una relación de 3 [dB] por metro, pero el nivel de reverberación se mantiene prácticamente constante.

En la figura 1-8 se puede observar, de forma gráfica, la reverberación producida por la Respuesta de impulso de una sala.

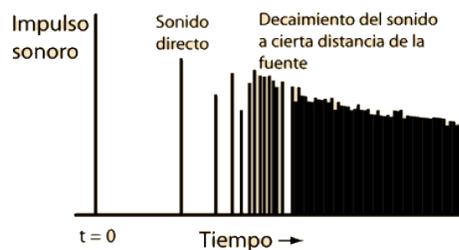


Figura 1-8: Respuesta de impulso de una sala (RIR) [17].

La distribución en tiempo y amplitud de las señales reflejadas en una sala, junto con su decaimiento en amplitud corresponden a la respuesta de impulso de una sala. También existe la distribución de las respuestas de impulso en frecuencia, llamada respuesta de frecuencia (Frequency Response Function o FRF), que se consigue haciendo la transformación de Fourier con la RIR.

Una onda sonora se expande desde la fuente llegando a las paredes y otros objetos, donde la energía del sonido es, a la vez, absorbida y reflejada. Solo con el sonido sin resonancia la fuente acústica, en general, es percibida como cercana, y al añadirle reverberación se percibe lejanía.

## **1.2 Antecedentes generales**

En la actualidad, la mayoría de los investigadores interesados en el sonido tridimensional tienen como punto de mira la reproducción fidedigna de este. Existen hasta la fecha varios métodos de reproducción de sistemas espaciales; el problema radica en la complejidad de estos sistemas, y lo difícil que resulta para una simulación de audio 3D generar un entorno virtual óptimo, sin mayores inconvenientes.

Al tener el oído humano una estructura física distinta para cada individuo, resulta muy difícil imaginar una HRTF única que funcione para todas las personas por igual, percibiendo las mismas localizaciones sonoras. A esto se le puede atribuir que la diferencia entre una HRTF con otra puede ser tal, que pueda funcionar perfectamente con una persona, pero completamente diferente con otra, incluso para algunos llegando a ser imperceptible, imposibilitando la percepción de las localizaciones de las fuentes sonoras emitidas por la reproducción. ¿Qué tanto puede variar la efectividad de una simulación de audio 3D, con dos o más funciones de transferencia de distintas características?

Básicamente se pueden distinguir entre tres sistemas de audio tridimensional principales, de los cuales derivaran todos los demás sistemas y métodos: Altavoces Estereofónicos, Reconstrucción Binaural en los oídos del oyente y Síntesis del Campo Sonoro alrededor del oyente. A continuación, se revisa de forma concisa cada uno de estos sistemas para conocer sus ventajas e inconvenientes.

### **1.2.1 Sistema espacial estereofónico**

Este término hace referencia a cualquier sistema sonoro que convierte o maneja sonidos tridimensionales. Aun así, hay muchas técnicas estéreo cuya reproducción se basa en altavoces que reflejan tan solo algunas de las referencias espaciales existentes en el sonido original. El grado de acierto de estas técnicas depende de distintos factores, sin embargo, son razonablemente sencillas de implementar desde el punto de vista de grabación y reproducción, y por tener una calidad sonora subjetiva bastante buena.

## **2 Canales**

Esta corresponde a una tecnología de larga tradición y experiencia. Para la reproducción se acepta, de forma casi universal, que la configuración óptima para un sistema estéreo 2-0 es un triángulo equilátero entre los altavoces y el oyente, como muestra la figura 1-9.

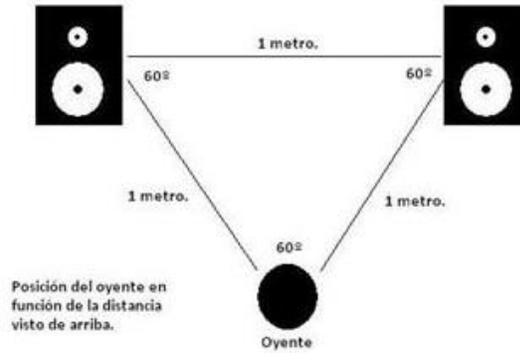


Figura 1-9: Posición óptima de escucha para un sistema estéreo 2-0.

Para la grabación, la técnica se basa en la utilización de dos micrófonos con diferentes posiciones de montaje y de respuesta direccional. Los parámetros a tener en cuenta son la localización y la profundidad. Estos parámetros nos dan información sobre diferencia de nivel y de tiempo entre las dos señales microfónicas recibidas.

### Multicanal – Surround

Los sistemas o formatos estéreo multicanal, conocidos también como de sonido envolvente, tienen muchas variaciones, tales como el formato 3-0, 3-1, 3-2, el popular estándar 5.1, 3-4, entre otros. En la figura 1-10, se ilustra la posición de los altavoces del multicanal estándar 5.1.

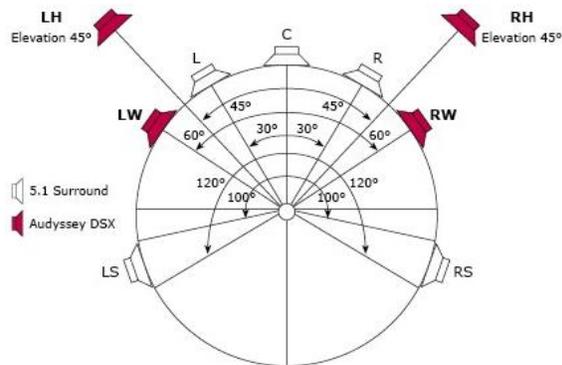


Figura 1-10: Multicanal estándar 5.1 [15].

### 1.2.2 Reconstrucción Binaural

La reproducción de señales binaurales, a través de auriculares, es una de las distintas maneras con las cuales se intentan recrear escenas originales para el oyente. La grabación se puede realizar mediante la ubicación de una cabeza de maniquí en un campo sonoro, pero también puede ser sintetizado en un equipo mediante la síntesis binaural. Se considera que la señal binaural es aquella que ya ha sido transformada, por lo tanto, incluye la HRTF.

## Grabación Binaural

Para conseguir una señal binaural, se necesita grabar la señal sonora una vez que haya sufrido la transformación producida por nuestro sistema auditivo. El sistema más utilizado para esto, es la simulación de una cabeza humana mediante una cabeza artificial, llamada "dummy head". A esta se le instala, en cada oído, un micrófono, los cuales reciben las ondas sonoras y las registran. En la figura 1-11, se ilustra la grabación binaural con un maniquí dummy head.

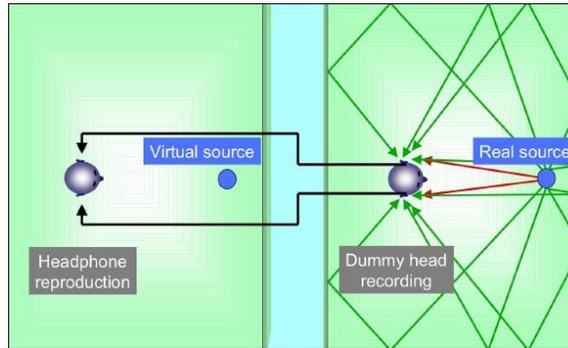


Figura 1-11: Grabación binaural con un maniquí dummy head [9].

Uno de los principales inconvenientes mencionado anteriormente radica en que la función HRTF cambia con cada individuo, de manera que las sensaciones que se pueden percibir con una HRTF distinta a la nuestra pueden ser confusas.

En la actualidad, la tendencia parece ser disponer de un conjunto de HRTF relativamente amplio, e implementar el más adecuado al oyente que va a utilizar el sistema, en función de algún aspecto físico sencillo de su anatomía, tales como la distancia interaural, el tamaño de la cabeza, el tamaño de la oreja, entre otros.

## Síntesis Binaural

Aunque las estructuras de la percepción de las HRTF no han sido comprendidas en su totalidad, estas ya han sido utilizadas ampliamente para sintetizar los sonidos espacializados a través de auriculares. Las HRTF y la ITD se pueden usar para filtrar un sonido monoaural en un sonido binaural; este sonará como si se originara en ese lugar.

Una señal monoaural se puede convertir en binaural usando un sistema de convolución, esto consiste en aplicar una función matemática a la señal entrante, con una señal que contenga la HRTF medida en el punto donde se quiere localizar la muestra.

La síntesis binaural del entorno (Binaural Room Synthesis o BRS) es un método para añadir el modelado acústico de un espacio sonoro. Esto resulta análogo a la reverberación producida en una sala cuando se emite un sonido; este es modelado físicamente por la estructura de la sala. El método mencionado es eficaz, desde un punto de vista de las teorías acústicas, pero es costoso computacionalmente. Los métodos alternativos que deben ser utilizados para generar esta reverberación suelen conseguir el efecto con el uso de filtros recursivos (filtros con feedback). De esta manera, vemos como un concepto tan tratado durante muchos años por la ingeniería acústica es la base para la síntesis binaural de entornos.

## Reproducción Binaural

Las señales binaurales han sido creadas para ser reproducidas directamente a los oídos del oyente, puesto que la señal ya sufre la transformación que añadiría el sistema auditivo. Entonces al incluir la HRTF se debe intentar no volver a añadir otra HRTF contraproducente. El sistema más fidedigno por el momento es mediante auriculares. Aun así, se crean irregularidades en las frecuencias debido al acoplamiento de los auriculares a nuestro pabellón auditivo.

Un problema en la reproducción mediante auriculares es el llamado efecto de localización dentro de la cabeza. Este efecto hace que las imágenes perceptivas de los sonidos se ubiquen distorsionadas en los ángulos azimutales y de elevación, resultando para el oyente más cercanas de lo que en realidad están. Además, si el oyente mueve la cabeza, al llevar auriculares las señales binaurales se moverán con ella, esto significa que estará anclado a una posición fija y dejará de parecer un efecto real. La implementación del seguimiento de la cabeza trae dos beneficios: uno, es agrandar el área de posición óptima del oyente. El otro, es que, si el seguimiento es suficientemente rápido, el oyente podrá tener el beneficio de las señales de localización dinámicas. Estas son de elevada importancia para la percepción espacial, sobre todo para solucionar la confusión típica delante-detrás que provocan los sistemas binaurales.

En un sistema de auriculares virtuales, se utiliza un sistema de reproducción estéreo de dos canales, usando altavoces. Pero existe un detalle. Al recibir señales en cada oído, procedentes de ambos altavoces, se crea el efecto llamado Crosstalk. Al mezclarse la señal de ambos altavoces en el mismo oído (llamado también diafonía), hay cancelaciones de ondas debido a las fases de estas, esto produce un espectro de frecuencia distinto, que es procesado por el cerebro. En otras palabras, se entiende por crosstalk a la presencia de una señal de audio no deseada a través de un acoplamiento accidental. En la figura 1-12, se observa un esquema de las señales referidas al efecto crosstalk.

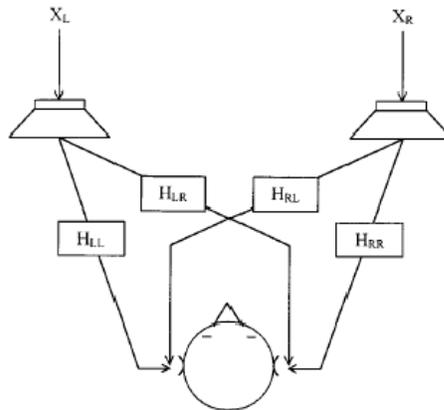


Figura 1-12: Efecto Crosstalk [10].

La HRTF es una función que, en cierto ángulo de incidencia, describe un frente de ondas del campo sonoro libre, el cual se dirige a un punto del oído del oyente. Esto produce que la imagen espacial, contenida en la señal que es emitida por unos altavoces, se escuche distorsionada. También degrada la imagen virtual sonora, limitando el campo a la distancia de separación de los altavoces, mediante el efecto Haas comentado anteriormente. Sin embargo, se pueden diseñar filtros canceladores que permitan al oyente tener presente, en cada oído, solo la señal deseada.

El sistema se modela con cuatro filtros, de forma que la señal de cada altavoz pasa por dos de ellos. Uno de estos filtros dará la respuesta directa al oído correspondiente a ese altavoz, mientras que el otro filtro proporcionará la señal al otro oído. Estos filtros son llamados XTC.

Los auriculares han tenido grandes dificultades para reproducir las imágenes frontales, utilizando HRTF no individualizadas. Además, a través de auriculares, las imágenes tienden a ser localizadas muy cerca de la cabeza, especialmente para los objetivos de distancia media [9].

### 1.2.3 Reconstrucción del campo de onda

Como se ha revisado hasta ahora, los sistemas de reproducción espacial convencionales están basados básicamente en las HRTF y la técnica de paneo, mediante la intensidad. Estos ajustan las respuestas de los diferentes canales de altavoces, para que al superponerse se produzca la direccionalidad deseada en los oídos del oyente.

El sistema de Síntesis de Campo de Onda (WFS: Wave Field Synthesis), utiliza una formación de altavoces para reproducir un campo sonoro sin la limitación de una posición fija, tal como muestra la figura 1-14. Ha sido identificado como el equivalente acústico de la Holografía, por esta razón a veces es referido como Holofonía.

#### Principios y Características

La WFS es un método de reproducción de sonido, basado en los fundamentos básicos de propagación de ondas y en el principio de Huygens. Según este principio, la propagación de un frente de ondas puede ser descrita como la contribución de un número de fuentes secundarias puntuales distribuidas a lo largo del frente de ondas.

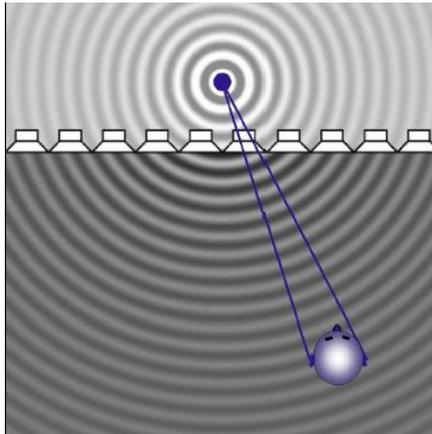


Figura 1-13: Representación del frente de ondas simulado mediante WFS [6].

La WFS permite la generación de campos sonoros con características temporales y espaciales, dentro del volumen o área encerrada por las fuentes secundarias, como se ilustra en la figura 1-13. Este método ofrece un área de escucha con alta calidad. La alimentación de cada altavoz dependerá de la posición de la fuente virtual, la cantidad de altavoces y del área de escucha. Se debe tener un algoritmo de síntesis para cada altavoz, con el fin de que estos proporcionen la señal de excitación. De esta manera, el campo reproducido, dentro del frente sonoro, estará compuesto por la suma de las contribuciones de cada altavoz que forma el frente de altavoces.

### 1.3 Método y teoría

En el ámbito de la acústica, un eco se puede relacionar con la convolución del sonido emitido por una fuente, con una función matemática que representa la estructura física del lugar y los diversos objetos que la reflejan. A modo de ejemplo, se considera que el gritar, aplaudir, o dar un golpe en una sala, genera efectos audibles producidos por la reflexión de la onda sonora con el entorno. En el área de la ingeniería, se sabe que la salida de un sistema lineal (estacionario, tiempo-invariante o espacio-invariante) es la convolución de la entrada con la respuesta del sistema a un impulso.

El objetivo del proyecto es realizar una correcta implementación de un sistema de audio 3D, esto consiste básicamente en convolucionar una señal de audio cualquiera, con otra señal que cumpla la función de filtro y que proviene de la HRTF. Todo esto, con el objetivo de que, al escuchar el sonido resultante, este genere la sensación de que una señal de audio cualquiera, sea emitida de una fuente localizada en la dirección que especifique dicha función de transferencia.

La forma de realizar esta convolución, consiste en aplicar la Transformada de Fourier a cada una de las señales entrantes en el dominio del tiempo, para obtener el espectro en el dominio de frecuencia antes de realizar la multiplicación. Al resultado del producto, se le aplica la transformada inversa de Fourier, para volver a obtener la señal en el dominio del tiempo y esta quede lista para su reproducción.

A continuación, se revisarán los conceptos previos para la comprensión correcta del sistema a implementar.

#### 1.3.1 Transformada Discreta de Fourier

La transformada discreta en el tiempo de Fourier (DTFT), entrega una representación en el dominio de la frecuencia de una secuencia  $x(n)$ , y está dada por:

$$X(\omega) = DTFT[x(n)] = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} \quad (1-1)$$

Cuando se intenta calcular la DTFT en forma numérica con un computador se presentan dos problemas:

- La secuencia  $x(n)$  puede ser arbitrariamente larga.
- El espectro  $X(\omega)$  es, en general, una función continua de  $\omega$ .

La Transformada Discreta de Fourier (DFT) es una transformación tiempo – frecuencia similar a DTFT, pero es discreta en ambos dominios y opera sobre un número finito de datos. En la figura 1-15, se observa una comparación gráfica entre dominio de tiempo y dominio de frecuencia. La DFT está dada por:

$$X(k) = DFT[x(n)] = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} \quad , \quad 0 \leq k \leq N - 1 \quad (1-2)$$

Esta transformada opera sobre un bloque de N muestras de la secuencia  $x(n)$ , y entrega N componentes de frecuencia, indexadas mediante la variable entera k.

La DFT es siempre calculable en un computador. Cabe señalar que las dos características mencionadas anteriormente hacen que el resultado obtenido no sea siempre similar al entregado por la DTFT. Sin embargo, si se usa un valor de  $N$  adecuado (de algunos centenares o miles de muestras), se puede obtener una buena aproximación al espectro calculado por la DTFT.

En la DFT, como la señal es discreta en el tiempo, el espectro es periódico. Además, como el espectro es discreto en la frecuencia, la secuencia es periódica. Es decir, para la DFT, tanto la

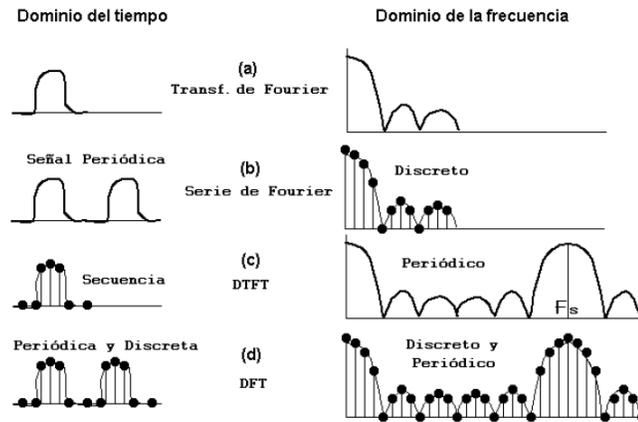


Figura 1-15: Comparación dominio de tiempo - frecuencia [3].

secuencia como su espectro son periódicos, y se repiten fuera del bloque analizado de largo  $N$ , aunque en la realidad no sea así.

### 1.3.2 Convolución por medio de la DFT

La convolución es una operación importante en la práctica, ya que, para filtrar una secuencia con un filtro digital tipo FIR, se debe convolver la secuencia con la respuesta impulso  $h(n)$  del filtro. Sin embargo, la convolución es una operación exigente para los procesadores: para calcular cada muestra de la salida de un filtro, con una  $h(n)$  de largo  $N$ , es necesario sumar  $N$  productos de las muestras de la secuencia, con las muestras de  $h(n)$ . Por ejemplo, si la secuencia está compuesta por un millón de muestras, y  $N=1000$  [-], se deben efectuar mil millones de multiplicaciones y sumas.

Como existen algoritmos muy eficientes para calcular la DFT y la IDFT (FFT e IFFT), y como multiplicar en el dominio de la frecuencia es equivalente a convolver en el dominio del tiempo (pero mucho más eficiente desde el punto de vista computacional), parece conveniente usar un “atajo” a través del dominio de la frecuencia para convolver.

Sin embargo, cuando se multiplican las DFT de 2 secuencias y se calcula la IDFT del producto, no se obtiene la convolución normal o “lineal”, sino otro tipo de convolución, denominada “convolución circular”. Esto se debe a que la DFT asume que la secuencia es periódica, fuera del bloque analizado.

Resulta posible alterar los datos antes de calcular la convolución circular, de modo que el resultado obtenido sea idéntico a la convolución lineal, logrando para valores grandes de  $N$  un ahorro significativo en el tiempo de cálculo. En esta sección se desarrolla y justifica este procedimiento, que lleva a lo que se denomina normalmente “fast convolution”, o “convolución rápida”.

### 1.3.3 Convolución circular

A diferencia de la convolución lineal, la convolución circular se aplica sobre un número  $N$  de muestras (no entre  $-\infty$  y  $+\infty$ ), y utiliza desplazamiento circular (en lugar de desplazamiento lineal). La fórmula es:

$$y(n) = \sum_{p=0}^{N-1} x_1(p)x_2(n-p) \quad (1-3)$$

Como la sumatoria se realiza sobre un período de la secuencia, la equivalencia en el dominio de la frecuencia se expresa como:

$$DFT [x(n - n_0)] = X(k)e^{-j2\pi kn_0/N} \quad (1-4)$$

Se ha demostrado que cuando se multiplican las DFT de 2 secuencias y se calcula la IDFT del producto, se obtiene la convolución circular de las secuencias.

$$IDFT[X_1(k)X_2(k)] = \sum_{p=0}^{N-1} x_1(p)x_2(n-p) \quad (1-5)$$

La convolución lineal de 2 secuencias de largo  $N_1$  y  $N_2$  produce una secuencia de largo  $N_1 + N_2 - 1$ . Si las 2 secuencias tienen el mismo largo  $N$ , el resultado tiene un largo  $2N-1$ . En cambio, la convolución circular produce una secuencia de largo  $N$ . Esto es consistente con el hecho de que el paso por el dominio de la frecuencia vía la DFT y la IDFT se realiza en base a espectros de largo  $N$ .

Para que la convolución circular entregue el mismo resultado que la convolución lineal, basta con aumentar la duración de las secuencias originales a  $N_1 + N_2 - 1$ , agregándoles muestras de valor cero. De esta forma, cuando las muestras desplazadas hacia la derecha reingresan por la izquierda, no alteran el resultado, porque valen cero.

Como se mencionó anteriormente, la FFT y la IFFT son algoritmos que permiten calcular la DFT y la IDFT con un menor número de operaciones aritméticas, en comparación al que demanda la aplicación directa de las fórmulas de la DFT y de la IDFT.

Se deben seguir los siguientes pasos:

1. Agregar muestras de valor cero a ambos vectores hasta completar una longitud de al menos  $N_1 + N_2 - 1$  muestras, para que la convolución circular entregue el mismo resultado que la convolución lineal. Como la FFT y la IFFT logran su máxima eficiencia cuando el número de muestras  $N$  es una potencia de 2, se deben agregar muestras de valor cero a los vectores hasta llegar a la potencia de 2 más cercana.
2. Calcular las FFT de los vectores.
3. Multiplicar los espectros obtenidos.
4. Calcular la IFFT del producto.

La FFT no es una transformada, sino un algoritmo que permite calcular la DFT con menor número de operaciones aritméticas. Mientras la DFT requiere  $N^2$  multiplicaciones complejas, la FFT requiere  $N \cdot \log_2(N)$ , siempre que  $N$  sea una potencia de 2. En la tabla siguiente se muestran algunos ejemplos.

Tabla 1-1: Número de multiplicaciones complejas requeridas por la DFT y por la FFT

$N$	$N^2$ [DFT]	$N * \log_2(N)$ [FFT]	[%]
4	16	8	50
128	16384	896	5.5
1024	1048576	10240	0.98

La reducción del tiempo de cálculo, combinado con el avance en la tecnología de semiconductores, generó una serie de aplicaciones para la FFT, tales como análisis espectral, convolución rápida, radar doppler, módems, detección de señales inmersas en ruido, entre otros.

Para este proyecto utilizaremos la Raspberry Pi 2 Model B como DSP, esta se encargará de efectuar los cálculos a través del software Pure Data, para llevar a cabo la convolución entre dos señales y así, posteriormente, poder generar el efecto de auralización espacial.

Todos estos antecedentes teóricos acerca de la investigación sobre la auralización espacial y, la implementación de la función de convolución, nos servirán para formar una sólida base de conocimiento previo, y así comenzar con el desarrollo de la simulación de audio tridimensional. En este capítulo se desarrolla el contexto de trabajo basado en la percepción del sonido y sus propiedades, se define la problemática de los distintos sistemas de reproducción binaural investigados y se presenta el marco teórico basado en la función de convolución, cuyo entendimiento resulta de gran importancia para este trabajo.

## 2 Solución y marco teórico

En este capítulo, se detallan las características de los elementos y software a utilizar para la implementación del sistema de Audio 3D. Se presentan las especificaciones técnicas de la Raspberry Pi 2, la tarjeta de audio Cirrus Logic y, los elementos básicos y esenciales del software Pure Data para construir el programa para la simulación. También se presenta el sistema de auralización, un diagrama para facilitar su entendimiento y la base de datos de las HRTF utilizadas para realizar la convolución.

### 2.1 Raspberry Pi 2 Model B

Raspberry Pi es un ordenador de una sola placa, de bajo costo y tamaño reducido (Single Board Computer – SBC), creado por la fundación Raspberry Pi en Reino Unido (UK) y lanzada al mercado en el año 2012. Principalmente este ordenador fue diseñado para que pueda ser utilizado por diversas personas en el ámbito computacional, de cualquier edad y con fines pedagógicos. En la web se puede encontrar una gran variedad de proyectos realizados con Raspberry Pi debido a su pequeño tamaño y la amplia gama de posibilidades que ofrece, pudiendo interactuar con pulsadores, sensores, Leds, tarjetas de sonido, pantalla táctil y cámara de video, entre otros, a través de sus pines GPIO.

En esta ocasión, la Raspberry Pi 2 Model B será utilizada como un procesador digital de señales (DSP) en tiempo real que permita la simulación de audio 3D para auriculares, por lo que tendrá incorporada una interfaz de audio, la tarjeta Cirrus Logic. En la Figura 2-1, se muestra una imagen de la Raspberry Pi 2 Model B.



Figura 2-1: Raspberry Pi 2 Model B [16].

Las características de la Raspberry Pi 2 Model B más interesantes para este proyecto son [16]:

- **Procesamiento:** Chipset; Broadcom BCM2836 32 Bit, 900 [MHz] Quad-Core ARM Cortex-A7.
- **RAM:** 1 [GB]
- **Almacenamiento:** tarjeta MicroSD
- **Fuente de alimentación:** 1,8 [A] / 5 [V]
- **Puertos USB:** 4
- **GPIO 40 Pines**
- **Puerto HDMI**
- **Puerto Ethernet**

El ordenador Raspberry Pi 2 Model B permite la instalación de cualquier sistema operativo (SO) ARM GNU/Linux debido a su procesador ARMv7. También permite instalar Windows 10.

Para este proyecto se utilizará el sistema operativo Debian (Linux) optimizado para Raspberry Pi, este es el sistema oficial de la fundación. La instalación de este se realiza a través de la tarjeta microSD, en la cual se carga mediante NOOBS (instalador de Sistemas Operativos) o una imagen de disco del sistema operativo deseado.

## 2.2 Tarjeta de Audio Cirrus Logic

Al estar la Raspberry Pi 2 Model B limitada en cuanto a entradas, salidas y calidad de audio, se ha optado por utilizar la interfaz de audio Cirrus Logic, la cual permite realizar distintos tipos de entrada/salida de alta calidad. En la figura 2-3, se presentan las conexiones de la tarjeta de audio Cirrus Logic.

Esta tarjeta de audio está basada en la interfaz de audio Wolfson, de Wolfson Microelectronics, adquirida por Cirrus Logic en 2014. La tarjeta Cirrus Logic se conecta a la Raspberry Pi 2 mediante los 40 pines GPIO. En la Figura 2-2, se ilustra la tarjeta de audio Cirrus Logic.



Figura 2-2: Cirrus Logic Audio Card [16].

Las características generales que ofrece esta tarjeta de sonido son las siguientes:

- Entradas y salidas analógicas de nivel de línea (hasta 192 [kHz], 24 [bits]).
- Entrada y salida de audio digital (SPDIF)
- Salida de auriculares de alta calidad, con posibilidad de auriculares con micrófono.
- Amplificador incorporado para alimentar altavoces (requiere alimentación).
- Micrófonos digitales incorporados basados en la tecnología MEMS.
- Encabezado de expansión de pines
- Puede ser alimentada por la alimentación de la interfaz de audio (AUX Power in).

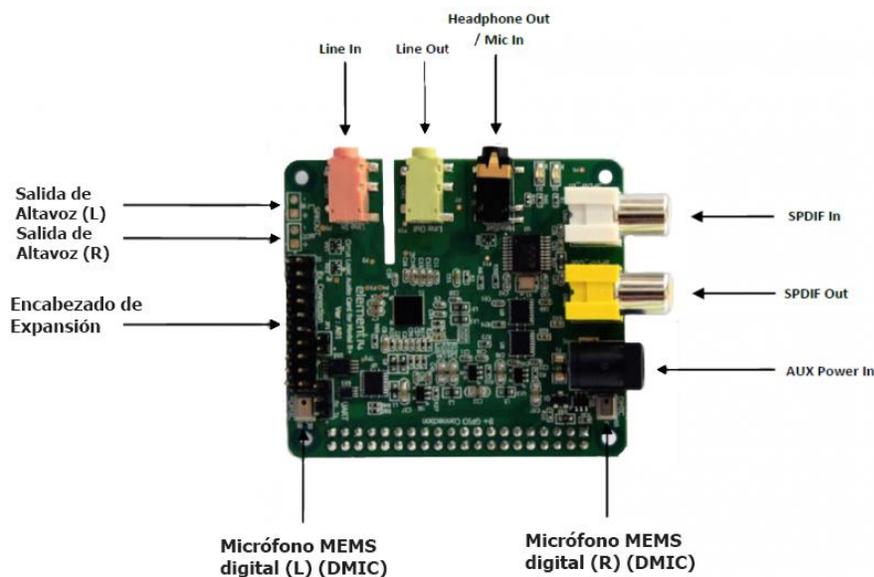


Figura 2-3: Conexiones de la tarjeta de audio Cirrus Logic.

Se realizó la instalación de la tarjeta de sonido. Todo este proceso junto con la selección de la Cirrus Logic, como tarjeta de audio predeterminada y su configuración, se detalla en el apéndice.

## 2.3 Pure Data

Pure Data (Pd) es un software libre que se basa en un lenguaje de programación gráfico en tiempo real, también denominado como lenguaje de programación de flujo de datos, desarrollado por Miller Puckette en el IRCAM (Institut de Recherche et Coordination Acoustique/Musique) de Paris. Es un software de multiplataforma (Linux, Windows y OS) que permite el procesamiento de audio, video, gráficos, y la interacción con otros dispositivos como por ejemplo MIDI.

Pd permite la implementación a nivel gráfico, basándose en objetos como funciones que realizan su cometido dentro de un flujo de datos y es capaz de realizar el procesamiento en tiempo real, actuando como DSP. Este sistema permite crear un programa sin necesidad de escribir líneas de código, sin embargo, también es posible crear estos objetos en otros lenguajes de programación, como por ejemplo C, desarrollando nuevos códigos que puedan ser utilizados como objetos. A estos objetos, Pd los denomina externals y librerías.

La programación en Pd se basa en un ambiente gráfico que modela y crea un flujo de datos que utiliza objetos, mensajes, sliders, comentarios, entre otros, que modifican dicho flujo de datos, siendo estos de audio o de control, quedando representado por líneas que actúan como envíos de un objeto a otro. El flujo de datos de audio es representado por una línea gruesa, mientras que el flujo de datos de control queda representado por una línea fina.

A continuación, en la Figura 2-4, se muestra un ejemplo de un programa básico implementado en Windows. Este ejemplo tiene un control de volumen mediante un Vslider, y un Hslider que controla la frecuencia a generar. Ambos elementos están conectados a sus correspondientes interpoladores temporales, para alcanzar el valor deseado en un cierto intervalo de tiempo, con el objetivo de que los cambios no sean bruscos.

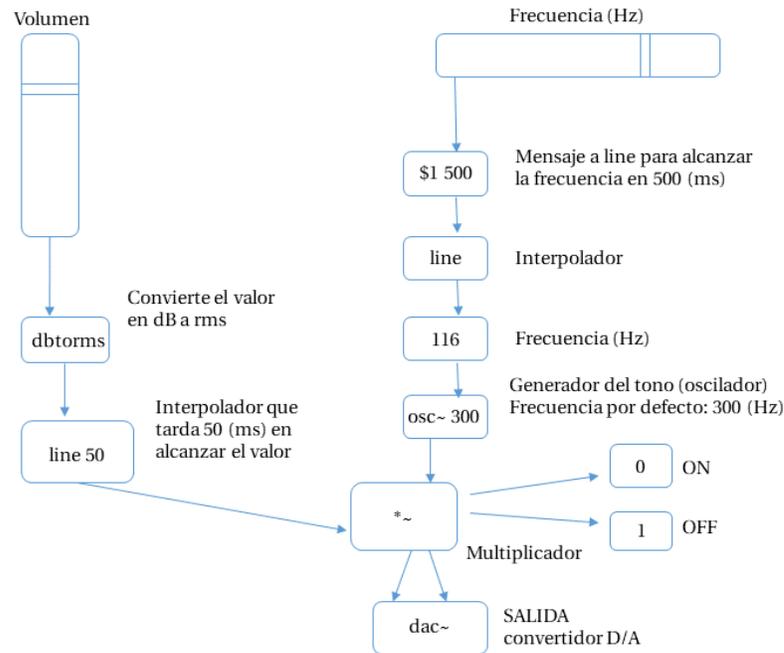


Figura 2-4: Ejemplo básico de un Programa en Pd.

Con el valor de la frecuencia se genera un tono mediante un oscilador, cuya salida corresponde a la entrada del convertidor Digital/Analógico, y esta, previamente multiplicada por el valor establecido en el control de volumen. Este programa básico consiste en un barrido de frecuencia moviendo el *Hslider*. Como se puede apreciar, las líneas conectoras que no contienen audio son más finas, a diferencia de las más gruesas, que representan el flujo de una señal de audio.

Pure Data pertenece a la familia de los lenguajes de programación basados en programas (MAX), entre ellos MAX/FTS, ISPW MAX. Estos son muy utilizados por compositores, diseñadores de software, investigadores, artistas, etc... Existen dos distribuciones principales de este programa, ellas son Pd-vanilla y Pd-extended. Pd-vanilla es la versión más básica de Pure Data, esta versión está principalmente orientada a la señal de audio y el procesamiento MIDI. En cambio, Pd-extended es una versión más completa de Pd-vanilla que contiene una amplia variedad de librerías, estas permiten una mayor flexibilidad y funcionalidad. Además de estas dos distribuciones existen muchas otras versiones,

entre ellas Pd-L2Ork, una versión centrada en Linux, basada en Pd-extended. Esta contiene una amplia gama de librerías y mejoras, en comparación con Pd-extended.

Para este trabajo, se considera instalar en la Raspberry Pi 2 la versión 0.47, debido a sus mejoras, variedad y control de objetos gráficos.

### 2.3.1 Configuración y elementos de Pure Data

Antes de utilizar Pd, se debe configurar para determinar la interfaz de audio, los canales, la frecuencia de muestreo, el retardo de salida y el tamaño de bloque de muestras de audio a utilizar.

La frecuencia de muestreo a configurar, debe ser la misma que la frecuencia a la que están muestreadas las HRTF, que se utilizarán en este proyecto. El retardo, para que el procesamiento sea considerado en tiempo real, debe tener como valor máximo recomendable 20 [ms]. El tamaño de bloque indica el número de muestras de la señal de audio, esto quiere decir que obtiene el audio en bloques de 64 muestras. Los dispositivos de entrada y de salida corresponden a los del Notebook, al implementar Pd en la Raspberry Pi 2 y configurar la tarjeta de audio se tendrá como dispositivo “*snd\_rpi\_wsp*”, este corresponde a la interfaz de audio Cirrus Logic, utilizando dos canales de entrada y dos canales de salida (izquierdo y derecho).

Como introducción básica a Pd, se presentan a continuación los elementos a utilizar dentro de un programa:

- **Objeto:** Indica el objeto externo o internal (objetos ya incluidos por defecto en Pd) a utilizar. Los objetos que manipulan audio se deben indicar con el símbolo “~” tras el nombre del objeto.



Figura 2-5: Ejemplo visual del bloque “objeto”.

- **Mensaje:** Envía mensajes a objetos pudiendo contener valores numéricos y letras o palabras. Algunos objetos contienen métodos que son ejecutados cuando reciben un mensaje con el nombre del método deseado.



Figura 2-6: Ejemplo visual del bloque “mensaje”.

- **Número:** Representa el número que obtiene a su entrada y lo devuelve a su salida, o incluso se puede modificar su contenido con el ratón, enviándose a su salida.



Figura 2-7: Ejemplo visual del bloque “número”.

- **Comentario:** Permite describir partes del programa para una mejor comprensión del mismo.
- **Bang:** Es equivalente a un mensaje con la palabra “bang” pero con apariencia gráfica. Cada vez que se pulsa envía a su salida la señal de realizar una acción (un 1).

- **Toggle:** Envía un “1” digital a la salida si ha sido pulsado, y un “0” si esta despulsado.
- **Slider:** Deslizadores que reciben y envían un número, posicionando la *slider* en la posición del número en cuestión.
- **Radio:** Representa un gráfico con distintas casillas. Al seleccionar una casilla devuelve el número correspondiente a la posición de la casilla marcada en el gráfico.
- **VU:** Representa el nivel de volumen de las señales de audio dB.
- **Graph:** Es un gráfico en el cual se pueden representar una curva de valores.
- **Array:** Es una gráfica XY que permite representar y manipular las señales en ella contenidas.

## 2.4 Sistema de auralización

Una vez solucionadas las complicaciones encontradas, debido a problemas por fallas de instalación del sistema operativo, se logró obtener en la Raspberry Pi la versión más reciente de Pure Data. Las primeras pruebas de funcionamiento fueron realizadas bajo la recomendación del profesor guía. Se aconsejó realizar la convolución de dos señales; un audio cualquiera y otro con distintos impulsos para escuchar la señal resultante. Esta debería reproducir el audio cualquiera (la voz de quien redacta), en el mismo instante de tiempo en el que se encuentran los distintos impulsos.

Una vez hechas estas pruebas, comprobando que el programa realiza una correcta convolución, se procedió a diseñar el sistema de auralización. Este consiste en dos convoluciones, una referida a la respuesta impulso del oído izquierdo y otra a la respuesta impulso del oído derecho. Ambas convolucionadas con la misma señal de audio. A continuación, en la figura 2-8 se presenta un diagrama de bloques general del sistema a diseñar.

En la figura 2-8, se puede apreciar un diagrama sencillo del sistema a implementar. Cabe señalar que los bloques HRTF, asociados a la oreja izquierda y derecha, representan todas las señales de las HRTF almacenadas en una base de datos. La señal de audio se convuelve por separado, tanto para la oreja izquierda como para la derecha, ya que cada una contiene una HRTF distinta asociada a la posición, en la cual se encuentra la fuente de sonido. Luego, la señal se envía a los auriculares y estos emiten el sonido a los oídos.

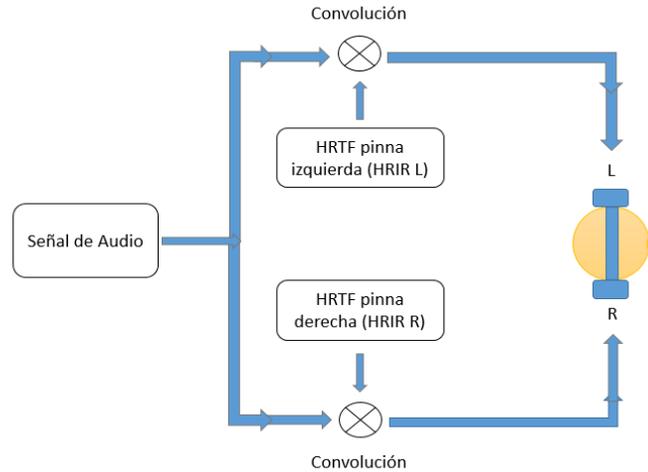


Figura 2-8: Diagrama general del sistema a diseñar.

En la figura 2-9, se observa un diagrama de bloques. Este describe el proceso realizado dentro del bloque “convolución”, mostrado en la figura 2-8. Cabe mencionar que este proceso es idéntico, tanto como para las HRTF relacionadas con la oreja izquierda, como para las de la derecha.

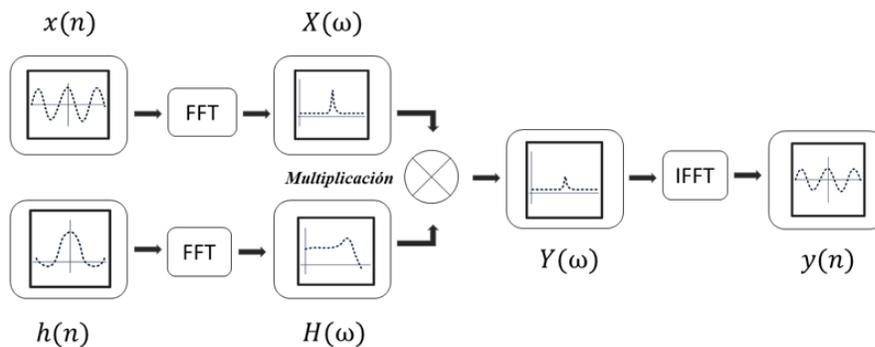


Figura 2-9: Diagrama en bloque del proceso de convolución.

Una vez ingresadas las señales  $x(n)$  y  $h(n)$ , ambas en dominio de tiempo, se les aplica la transformada rápida de Fourier. Estas dos señales resultantes se encuentran después de la transformada en dominio de frecuencia, luego se multiplica  $X(\omega)$  con  $H(\omega)$  para obtener  $Y(\omega)$ . Finalmente, se le aplica la transformada inversa rápida de Fourier, para obtener el producto de ambas señales en dominio de tiempo  $y(n)$  y, posteriormente, esta pueda ser reproducida por los auriculares.

## 2.5 Sistema de convolución

Como se ha visto, la convolución es una operación muy importante en DSP. Esta permite calcular la salida de un sistema para una excitación cualquiera, si se conoce la respuesta impulso del sistema. No solo establece una relación matemática entre la entrada y la salida, sino que, en muchos casos, corresponde a la forma en cómo se implementa el sistema. El diseño consiste en obtener la respuesta impulso, correspondiente a la respuesta de frecuencia deseada. El proceso de filtraje se realiza convolviendo la entrada con la respuesta impulso.

El principal objetivo del proyecto consiste en diseñar el sistema de convolución, en el software Pure Data. En esta oportunidad, se necesita realizar las operaciones descritas previamente para producir el efecto de auralización espacial, para esto utilizaremos el objeto “convolve~”.

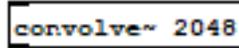


Figura 2-10: Objeto “convolve~” extraído de Pd.

Este objeto se obtiene directamente de Pd (versión 0.47 en adelante). El bloque “convolve~” proporciona la función de convolución con respuesta de impulso (IR). El único argumento de creación es el tamaño de partición (el cual se observa en la Figura 2-10), el cual determina la cantidad de retraso. Esta función opera regularmente bien al analizar diferentes IR’s (Respuestas de Impulso) en tiempo real, sólo se oye un clic momentáneo cuando se cambian las respuestas de impulso.

A continuación, se detallará el proceso de convolución con archivos de audio pregrabados y con entrada de audio en tiempo real.

### 2.5.1 Convolución de dos archivos de sonido pregrabados

Con dos archivos de sonido pregrabados, la convolución es bastante simple. Se determina qué archivo es más largo, se encuentra la próxima potencia más alta de 2 (denominada  $N$ ), y luego se rellena cada uno de los archivos con ceros adicionales hasta  $N$ . A continuación, se realiza una transformada de Fourier directa en cada uno de ellos, y se multiplican los conjuntos de datos transformados punto por punto. Debido a que los datos contienen valores imaginarios, se debe hacer una multiplicación compleja. Después de la multiplicación, todo lo que queda por hacer es una transformada de Fourier inversa al producto que está en dominio de la frecuencia. Dependiendo del algoritmo FFT específico que se esté utilizando, también se podría tener que escalar la amplitud por el recíproco de  $N$ .

Este es el método de dominio de frecuencia estándar, el cual resulta ser mucho más rápido que realizar convolución en el dominio de tiempo. En cualquier caso, el resultado es que una copia escalada y desplazada en el tiempo de un archivo, toma el lugar de cada muestra en el otro. Por lo tanto, si se tiene una respuesta de impulso (IR) de una habitación (es decir, un ruido violento de poca duración, seguido por los ecos naturales que produce el recinto), se puede insertar ese patrón de ecos en cada una de las muestras del archivo de entrada.

### 2.5.2 Convolución con entrada en tiempo real

Con la entrada en tiempo real, la situación se vuelve más complicada. Se puede tener una IR preparada de antemano, pero la otra fuente de sonido sólo aparece poco a poco. Normalmente, llegará en trozos de 64, 128, 256, o algún otro número de muestras. La convolución particionada resuelve este problema dividiendo la IR en un cierto número de particiones, que tiene un largo de  $M$  muestras.  $M$  debe ser del mismo tamaño que los trozos de entrada que llegan en tiempo real. Luego, cada una de las particiones es transformada por Fourier y almacenada como datos de dominio de frecuencia.

Cuando llega cada nueva parte extraída de la señal de entrada (las  $M$  muestras más recientes), a este se le aplica la Transformada de Fourier y lo convierte en datos de dominio de frecuencia complejos. El nuevo bloque de datos de dominio de frecuencia se multiplica por cada una de las particiones de IR pre-

preparadas que también están en el dominio de frecuencia. Se puede hacer inmediatamente una transformada de Fourier inversa en el fragmento más reciente disponible, para escuchar esa parte de la entrada procesada en el dominio de tiempo, pero los otros fragmentos esperarán en un búfer para ser emitidos en el futuro. El punto crucial a comprender es que a medida que nuevos trozos de entrada ingresan, se transforman y se multiplican en el dominio de la frecuencia, se suman estos nuevos datos con cualquier otro dato en dominio de frecuencia que estaba esperando en esa parte del búfer. Así, ahorramos numerosas llamadas FFT sumando en el dominio de la frecuencia.

El último punto a considerar, es que el tamaño de la transformada de Fourier debe ser  $2M$ . Es decir, cuando se reciben las últimas  $M$  muestras se rellena con ceros a  $2M$  muestras, luego se transforma. Todas las particiones IR se rellenan a este mismo tamaño. Cuando a cualquier trozo se le aplica la transformada de Fourier inversa, su contenido principal estará sobre las primeras  $M$  muestras, arrastradas por muestras de baja amplitud o ceros. A continuación, se realiza un proceso de superposición y adición de modo que, las primeras  $M$  muestras de las  $2M$  procesadas más recientemente se mezclan con las últimas  $M$  muestras, del trozo previo de  $2M$  procesadas.

## 2.6 Base de datos de mediciones HRTF

Para recrear un espacio sonoro tridimensional, se necesitan las respuestas impulso referidas a cada oreja, es decir, la señal que será auralizada se debe convolucionar con la respuesta impulso de la oreja izquierda, asociada a una cierta posición; también con la respuesta impulso de la oreja derecha, asociada a la misma posición (cada fuente sonora del espacio tiene asociado dos archivos de audio, uno para la oreja izquierda y otro para la derecha).

En el año 1994, se completó un extenso conjunto de mediciones de la función de transferencia de cabeza (HRTF), de un micrófono KEMAR. Las mediciones se basan en las respuestas de impulso de la oreja derecha e izquierda, de un altavoz Realistic Optimus Pro 7 montado a 1,4 metros del KEMAR. Se usaron secuencias binarias pseudoaleatorias para obtener las respuestas de impulso a una velocidad de muestreo de 44100 [Hz]. Se tomaron muestras de un total de 710 posiciones diferentes, a elevaciones de -40 grados a +90 grados. También, se midió la respuesta de impulso del altavoz en el campo libre, con distintos auriculares colocados en el maniquí, como se ilustra en la figura 2-11.



Figura 2-11: Maniquí KEMAR con el cual se realizaron las mediciones de las HRTF [8].

En este proyecto, se utilizarán las HRTF obtenidas en esta base de datos, ya que estas mediciones presentan con alta precisión las respuestas impulsos asociadas a la oreja y una gran variedad de posiciones para realizar las pruebas. Se recogerán alrededor de 20 señales, para realizar la prueba y sentir el efecto de que la fuente sonora va cambiando su posición.

El objeto “convolve~” proporciona las herramientas necesarias para implementar el sistema de simulación de audio en 3D, y así localizar la fuente auditiva. Se desarrolló un programa en Pd utilizando este objeto. Este realiza una convolución entre un archivo de audio a elección y distintas señales de audio, con diferentes respuestas al impulso, para escuchar y analizar si el resultado audible corresponde a la voz grabada, pero reproducida en una serie de posiciones en el espacio tridimensional, percibiendo de manera considerable el cambio entre una y otra.

Para llevar a cabo una correcta implementación del simulador, resulta necesario conocer las especificaciones técnicas del hardware a utilizar. Este se compone de una Raspberry Pi 2 montada con una interfaz de audio Cirrus Logic. Por otro lado, es importante ilustrar el sistema de auralización a simular y conocer el lenguaje de programación de Pure Data.

## 3 Desarrollo

Este capítulo detalla la estructura del programa construido en Pd. Se explican, paso a paso, las funciones y las distintas partes de este. Primero, se explica su diagrama en bloques para facilitar el entendimiento, luego, el programa hecho en Pd con sus correspondientes sub-programas. También, se presentan las posiciones de las HRTF utilizadas para la demostración de la simulación, de esta manera se puede tener una idea previa sobre la posición espacial en la cual se encuentran las fuentes que serán reproducidas.

### 3.1 Estructura del Programa

A modo de poder comprender de manera más sencilla los procesos del programa diseñado, se presenta, a continuación, un diagrama en bloques del sistema. Este explica cada proceso, desde que se llama al archivo de audio almacenado en el disco duro, hasta que este sale por el convertor Digital/Análogo para ser reproducido por los auriculares.

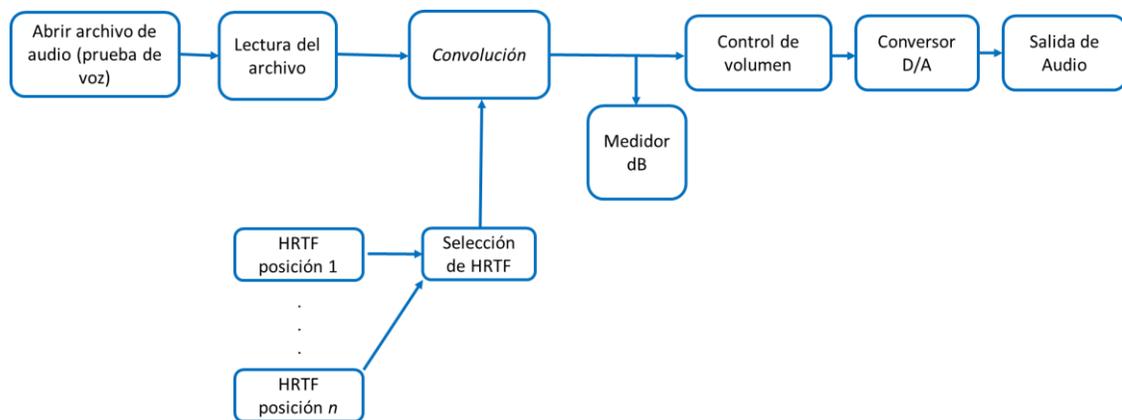


Figura 3-1: Diagrama en bloques del programa.

Se observa que, en la parte izquierda de la figura 3-1, se comienza por realizar un llamado al archivo de audio almacenado en nuestro computador (la prueba de voz a auralizar). Este se lee y se ingresa al bloque de convolución, el cual en el programa corresponde al objeto *convolve~*. Por otro lado, se debe seleccionar la señal impulso que queremos para realizar la convolución, en la figura se observa el bloque referido a cada posición de la fuente sonora para la reproducción. Estas señales (HRTF L y HRTF R; respuesta impulso de oreja izquierda y derecha), se envían a un objeto, el cual cumple la función de enlazar la señal seleccionada, enviándola al bloque “convolución”.

Este bloque se encarga de realizar todas las operaciones señaladas en los apartados anteriores. Posteriormente, esta señal se envía a un objeto que convierte los datos de audio en decibeles para realizar la medición. Luego, ambas ingresan a un controlador de volumen, manipulado por el usuario. Finalmente, después de salir del controlador de volumen, la señal se envía al conversor D/A y es emitida por los auriculares.

### 3.2 Construcción del Programa

A continuación, se mostrará el programa realizado en Pd, junto con los sub-programas. Se explica la manera en que estos operan. Se realiza una demostración auditiva de la correcta convolución de las señales y la experiencia sensorial de que la fuente sonora va cambiando de posición.

En un comienzo, el programa fue diseñado en Windows, con el objetivo de probar las funciones y señales de audio de forma previa. Luego se descargó e instaló el programa Pd en la Raspberry, finalmente se configuraron los drivers de la tarjeta de audio Cirrus Logic. Nada más bastó con guardar el programa hecho en Windows y abrirlo en Pd instalado en Debian. Una vez probado se procedió a construir el programa aplicando dos operaciones de convolución de forma simultánea, una para la función de transferencia procedente de la oreja izquierda y otra para la oreja derecha. En la figura 3-2, se muestra una imagen del programa construido.

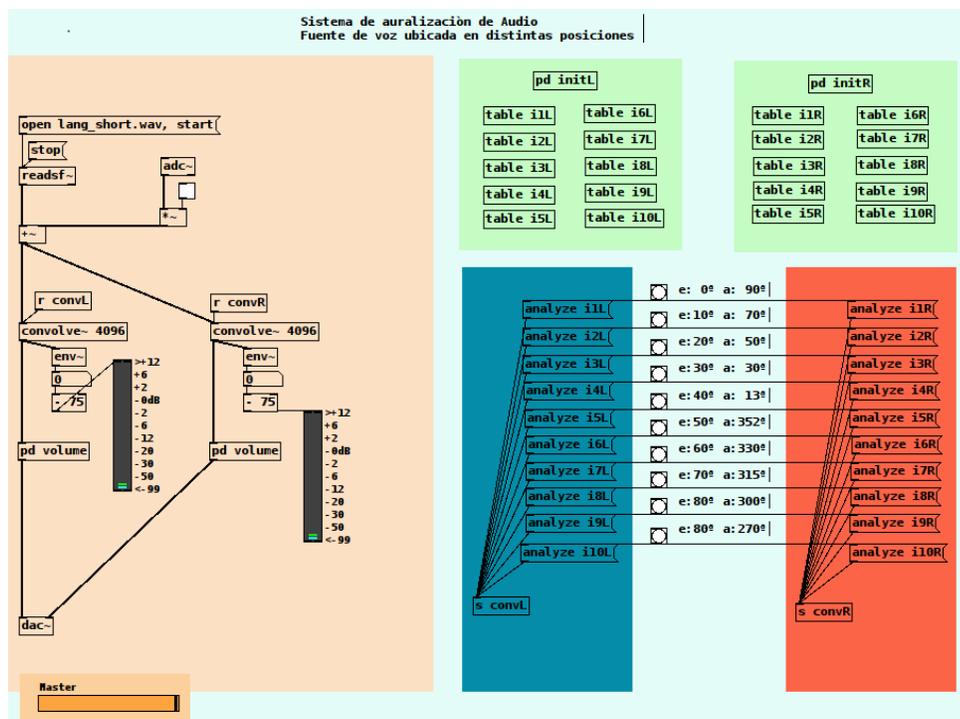


Figura 3-3: Programa implementado en Raspberry Pi con Pd.

Para aclarar y descubrir el funcionamiento del programa, se comienza por el bloque superior llamado "open lang\_short.wav, start". Este corresponde a un mensaje que "llama" al archivo de audio, para que este sea leído por el objeto "readsf~". Al clickearlo, inicia la reproducción del programa y el mensaje "stop" se encarga de detener la reproducción de este. El bloque "adc~" recibe la señal de audio

ingresada a la tarjeta de sonido montada en la Raspberry para ser convolucionada (esta corresponde a una convolución en tiempo real que por el momento no será utilizado en esta demostración). Este bloque ingresa a un multiplicador, al igual que el cuadrado blanco que se observa en la parte superior izquierda. Este cuadro simplemente funciona como un botón, para activar la entrada de audio de la tarjeta de sonido.

Como se puede observar, estas dos señales mencionadas anteriormente se dirigen a un objeto llamado “+~”. Este simplemente adiciona las señales permitiendo que, al activarlas, se seleccione cual se desea enviar al objeto “convolve~”. Como se observa en el programa, del objeto “+~” salen dos señales, ambas dirigidas a los objetos “convolve~” para ser convolucionadas. Los bloques llamados “r convL” y “r convR” cumplen la función del bloque “selección de HRTF”, mostrado en la Figura 3-1, la letra “r” viene de ‘receive’ o recibir. Análogamente, los bloques llamados “s convL” y “s convR” ubicados en la parte inferior derecha contiene la letra “s”, que viene de send o enviar. Como se puede notar, este objeto recibe los mensajes “analyze”, y los envía a “r convL” y “r convR”. Finalmente, este los ingresa al objeto “convolve~”.

En la parte superior derecha se pueden observar dos objetos, llamados “pd initL” y “pd initR”, seguidos hacia abajo por varios llamados “table”. Los objetos, cuyo argumento comienza con “pd”, se utilizan para crear subprogramas. Estos, al clickearlos, abren dos ventanas nuevas con los siguientes subprogramas:

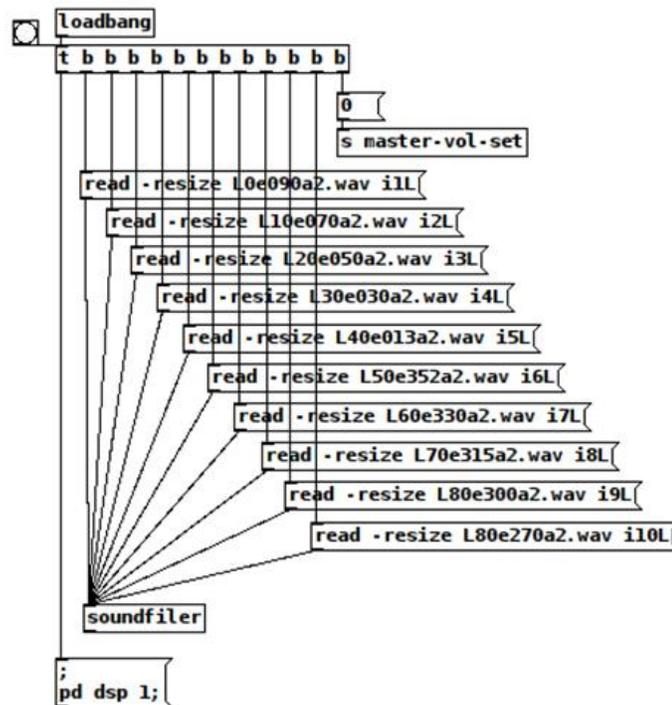


Figura 3-4: Programa dentro del objeto “pd initL”.

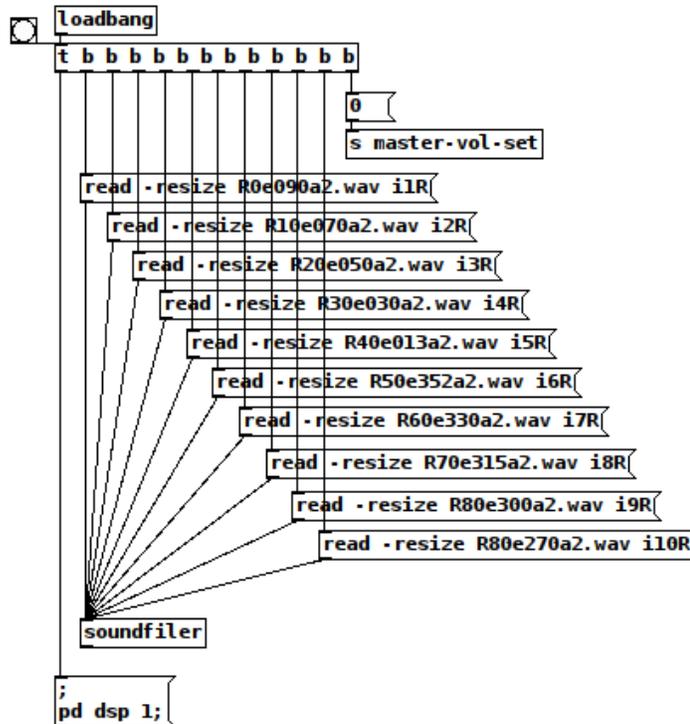


Figura 3-5: Programa dentro del objeto “pd initR”.

En los programas, mostrados en la figura 3-3 y la figura 3-4, se leen todos los archivos de las señales impulso HRTF que se quieren utilizar para la convolución. El objeto “loadbang” se utiliza cuando se desea mantener guardado varios valores para la próxima vez que se abra un programa, o cuando pueda haber un valor particular que desea recibir una orden de llamada al principio. El bloque “loadbang” envía una señal, o bang, tan pronto como se abre un sub-programa. El objeto “[t b ... b]” se utiliza para llamar a los distintos archivos de audio enlazado abajo, cargados con el mensaje que comienza con “read”. Luego, el nombre del archivo, y finalmente, los argumentos “i1L, i2L, i3L...”, los cuales serán utilizados para diferenciar un archivo con otro.

El objeto “soundfiler” se utiliza para transponer los datos del archivo de audio, a un arreglo en memoria denominado “i1L, i2L, ... i8L, i9L o i10L”. El objeto “soundfiler” es activado utilizando el mensaje [read -resize “nombrearchivo”.wav ix], que a la vez representa la forma de onda en los objetos “table ix” (notar que ix corresponde a “i1L, i2L, ... i8L, i9L o i10L”). El mensaje “0”, ubicado al final del objeto [t b ... b], seguido por “s master-vol-set”, se activa cuando se presiona el “bang” de la parte superior izquierda (cuadrado con un círculo dentro). Este simplemente baja el “slider” del volumen al momento de cargar los audios, para evitar cualquier salida de audio no deseada. Finalmente, el mensaje “pd dsp 1” activa el DSP de la consola, para poder procesar y manipular las señales de audio.

De esta forma, al clicar los mensajes “analyze ix”, se estará seleccionando la señal de impulsos que se quiera convolucionar con nuestro archivo “lang\_short.wav”. Luego, debemos clicar el mensaje “open lang\_short.wav, start” y acomodar el volumen con la slider “master”, para escuchar el resultado.

Se observa que “convolve~” cuenta también con un argumento, en este caso 4096. Este indica la cantidad de muestras de cada partición de las señales para realizar las operaciones dentro del objeto. La salida se envía a dos objetos, a “env~”, y antes de transformar los datos digitales en analógicos para

ser reproducidos, se envía a “pd volumen”. El objeto “env~” se encarga de transformar los datos en dB. El número “0” arroja los datos en dB al momento de comenzar la convolución, y el objeto “VUmeter” muestra la medición. El subprograma “pd volumen” se muestra en la figura 3-5.

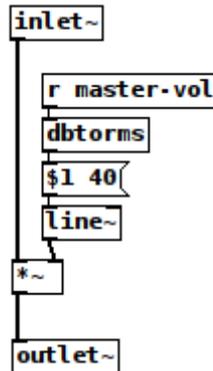


Figura 3-6: Programa dentro del objeto “pd volumen”.

Este programa consiste en implementar un “*slider*”, para regular el volumen de la salida. Transforma las unidades acústicas a decibeles. Cabe mencionar que la escala de ganancia consiste en un potenciómetro tipo *slider* que alcanza un valor nulo en la posición mínima, y un 100% en la posición máxima. Por lo cual, se tendrá que encontrar el volumen adecuado para su configuración en particular.

El programa consiste básicamente en realizar una muestra de 10 posiciones diferentes, cambiando su ángulo de elevación y azimut. Comienza con un ángulo de elevación de 0° y azimut de 90°, terminando con un ángulo de elevación de 80° y azimut 270°. El ángulo de elevación va incrementando cada 10°. Se escogieron estas posiciones de la base de datos para ejemplificar la diferencia en la recepción de la fuente de sonido, proveniente de una posición y otra, guardando una diferencia considerable para poder percibir de manera auditiva la diferencia.

## 4 Resultados

A continuación, se ilustran las pruebas realizadas y los resultados obtenidos. Se explican gráficamente la señal de prueba de voz y las respuestas al impulso de las señales referidas a las 3 posiciones mencionadas en el capítulo anterior. También se detalla la diferencia entre las respuestas al impulso de la oreja izquierda y derecha, referidas a la misma posición. Finalmente, se explica la señal resultante de la convolución implementada en el sistema de auralización.

### 4.1 Pruebas

Una vez implementado el sistema, se procedió a comprobar los resultados obtenidos. A continuación, con la ayuda del software MatLab, se mostrará gráficamente la señal de prueba con las señales de audio que contienen los impulsos correspondientes a las HRTF de la oreja izquierda y derecha, estas señales se muestran en dominio de tiempo.

La figura 4-1 muestra una imagen del archivo "lang\_short.wav" graficado en MatLab. Se puede apreciar claramente la forma de onda; en cuanto al eje horizontal, este nos arroja los valores en segundos.

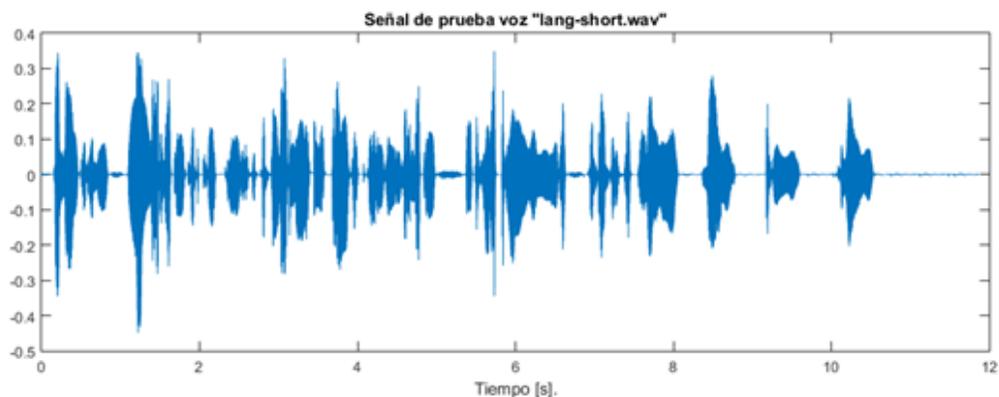


Figura 4-1: Forma de onda del archivo "lang\_short.wav".

Esta corresponde a la señal que será convolucionada con las distintas señales de impulsos. Se aprecia que la duración es de 10.68 [s].

A continuación, se presentará una serie de formas de onda. Estas corresponden a las señales de audio de los impulsos de las HRTF. Se compararán gráficamente las señales de 3 posiciones del espacio, una

con un ángulo de elevación de  $0^\circ$  y azimut  $90^\circ$ , otra con un ángulo de elevación de  $40^\circ$  y azimut  $13^\circ$ , la última con un ángulo de elevación de  $80^\circ$  y azimut  $270^\circ$ , todas en dominio de tiempo.

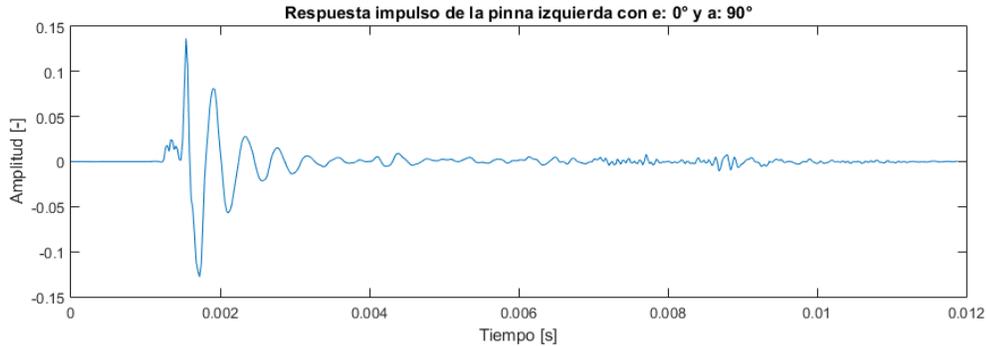


Figura 4-2: Respuesta Impulso de la oreja izquierda con elevación de  $0^\circ$  y azimut de  $90^\circ$ .

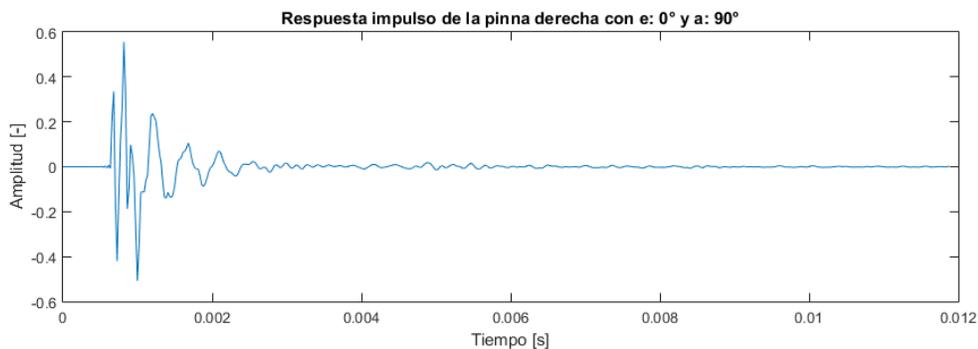


Figura 4-3: Respuesta Impulso de la oreja derecha con elevación de  $0^\circ$  y azimut de  $90^\circ$ .

En las figuras 4-2 y 4-3, se observa la respuesta impulso de la oreja izquierda y derecha, con un ángulo de elevación de  $0^\circ$  y azimut de  $90^\circ$ . Se aprecia que el impulso de la oreja izquierda aparece con un retardo temporal en comparación con el impulso de la derecha. Esto se debe a la diferencia interaural explicada anteriormente. Este impulso comienza aproximadamente en el instante de  $0,5669$  [ms], mientras que el impulso de la oreja izquierda comienza a los  $1,224$  [ms], existiendo un retardo temporal de  $0,6571$  [ms]. Esto se debe, a que la onda sonora viaja desde la fuente ubicada en el lado derecho del oyente (ya que el azimut es de  $90^\circ$ ), y llega primero a la oreja derecha, después se propaga para el lado izquierdo del oyente, existiendo una diferencia de distancia interaural entre una oreja y otra.

También, se puede apreciar una diferencia en la amplitud de ambas señales, ya que la onda se difracta con la textura de la cabeza, los hombros y el torso del oyente. De esta manera, la señal de audio llega con menor intensidad de amplitud a la oreja izquierda. La amplitud de la respuesta impulso relacionada con la oreja derecha, corresponde a  $0,556$  [-] (siendo el mínimo  $0$  y el máximo  $1$ ), y la relacionada con la oreja izquierda es de  $0,1368$  [-].

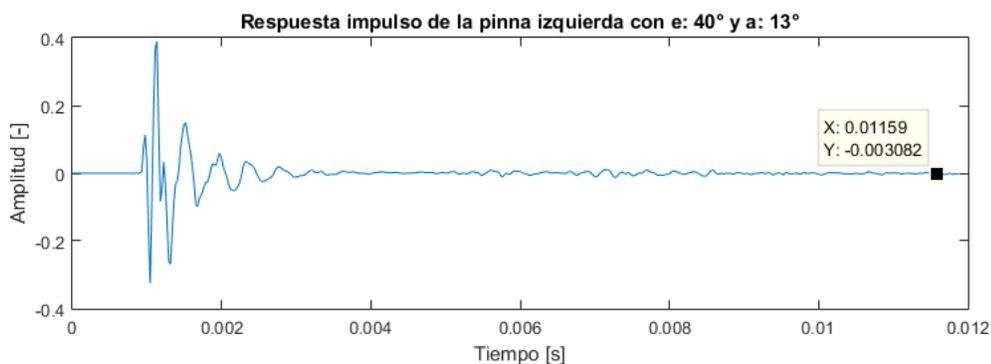


Figura 4-4: Respuesta Impulso de la oreja izquierda con elevación de 40° y azimut de 13°.

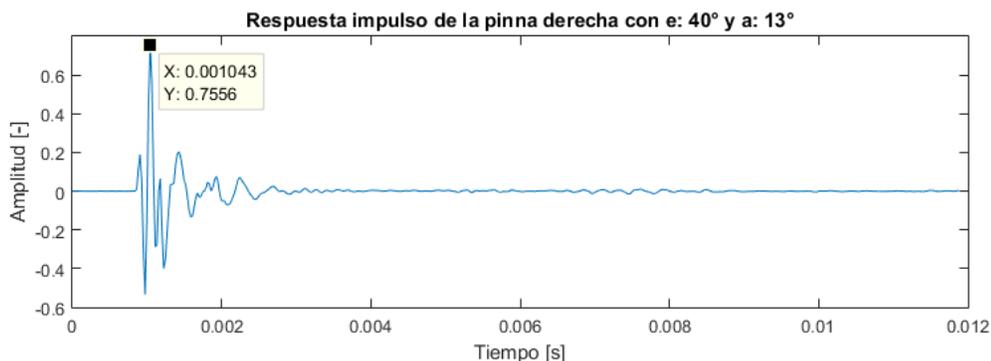


Figura 4-5: Respuesta Impulso de la oreja derecha con elevación de 40° y azimut de 13°.

Se observa, en las figuras 4-4 y 4-5, la respuesta impulso con un ángulo de elevación de 40° y azimut de 13°. En estos gráficos, resulta difícil apreciar que el impulso de la oreja izquierda aparece con un retardo temporal en comparación con el impulso de la derecha. Esto se debe a que el ángulo de azimut es bastante más pequeño, provocando que la señal llegue casi al mismo instante a la oreja izquierda y derecha. Este comienza aproximadamente en el instante de 0,839 [ms], mientras que el impulso de la oreja izquierda comienza a los 0,907 [ms], existiendo un retardo temporal de unos 0,068 [ms].

También, se puede apreciar una diferencia en la amplitud de ambas señales. La señal correspondiente a la oreja izquierda llega a un punto máximo de amplitud de 0,3892 [-], mientras que la derecha llega a un 0,7556 [-]. Se observa que, a pesar de que el ángulo de azimut es bastante reducido, aun así, se percibe una diferencia de amplitud entre la señal que llega al oído derecho y la señal que llega al oído izquierdo.

A diferencia de las gráficas anteriores, en las figuras 4-6 y 4-7 se observa la respuesta impulso con un ángulo de elevación de 80° y azimut de 270°. Es decir que, la fuente en este caso se localiza en el lado izquierdo del oyente, pero con un ángulo de elevación de 80°. En estos gráficos, a pesar de que la fuente se localiza en el lado izquierdo, resulta más difícil aún apreciar las diferencias de las respuestas impulso de las señales debido al ángulo de elevación. Mientras más elevado sea este ángulo, más difusa será la percepción de la localización de la fuente auditiva. El impulso referido a la oreja izquierda llega al instante de 0,7483 [ms], mientras que la respuesta impulso de la oreja derecha llega a los 0,839 [ms].

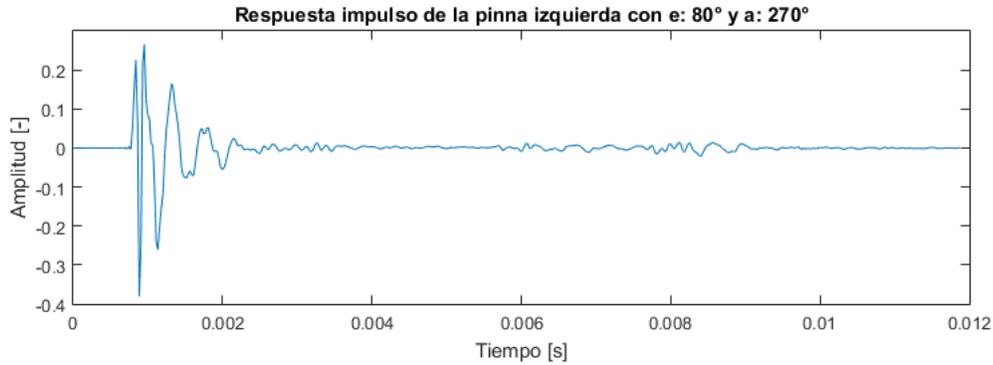


Figura 4-6: Respuesta Impulso de la oreja izquierda con elevación de 80° y azimut de 270°.

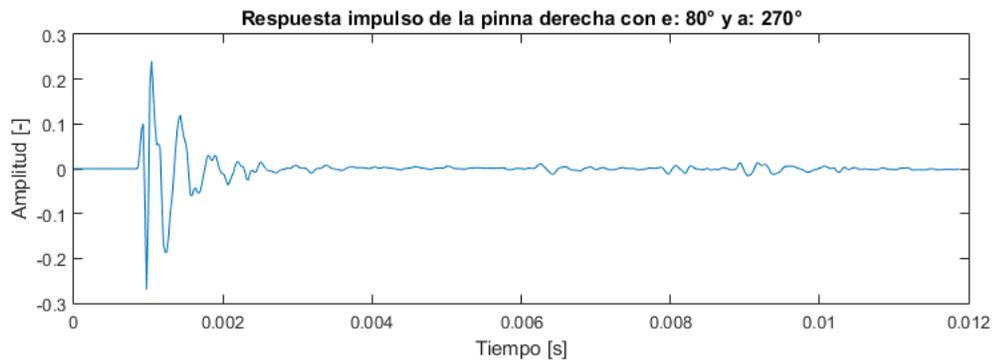


Figura 4-7: Respuesta Impulso de la oreja derecha con elevación de 80° y azimut de 270°.

Se puede apreciar que, aunque la diferencia sea mínima, la onda sonora llega antes a la oreja izquierda que a la derecha. También, se puede apreciar una diferencia en la amplitud de ambas señales. La señal correspondiente a la oreja izquierda llega a un punto máximo de amplitud de  $-0,3809$  [-], mientras que la derecha llega a un  $-0,2694$  [-]. Cabe mencionar que, debido a que el ángulo de elevación es bastante grande, la diferencia entre una respuesta impulso con otra es cada vez más pequeña, acercándose al punto de elevación de  $90^\circ$ , en el cual, independientemente del ángulo de azimut, la percepción de la fuente de sonido será la misma.

Todas estas señales presentadas anteriormente, fueron cargadas en Pd para ser procesadas. Al oír el resultado auditivo de la convolución, seleccionando las distintas posiciones presentadas en el programa, se puede estimar la ubicación de la fuente de sonido (generada por la convolución con las respuestas impulso HRTF). Al seleccionar una posición, se oye el audio como si este viniese de una fuente sonora ubicada en las coordenadas espaciales que especifican los ángulos.

## 4.2 Resultados obtenidos

Una vez cargadas las señales en la Raspberry Pi y montado todo el programa, se reprodujeron las distintas posiciones cargadas, para así sentir el efecto de una prueba de voz localizada en distintos puntos del espacio. Estas posiciones varían desde una elevación de  $0^\circ$  hasta  $80^\circ$ , con un azimut desde  $90^\circ$ , disminuyendo aproximadamente  $20^\circ$  cada muestra, llegando hasta el ángulo  $270^\circ$ .

La convolución se realiza de forma correcta, con el único detalle de que al ir cambiando y seleccionando las distintas posiciones se sienten unos pequeños “clicks” al reproducir el resultado. Esto se reduce,

significativamente, al modificar la cantidad de muestras de las particiones con el argumento del objeto “convolve~”. Es importante que este argumento sea una potencia de 2, para reducir el número de cálculos de la FFT y, de esta forma, no sobrecargar la capacidad computacional del procesador. En este caso, el programa divide la señal en particiones de 4096 muestras para realizar la convolución.

A continuación, se mostrarán unos objetos agregados al programa para grabar la salida del sistema. De esta forma, podremos abrir los archivos de audio y analizar, con ayuda de MatLab, gráficamente lo que está haciendo el programa al procesar las señales.

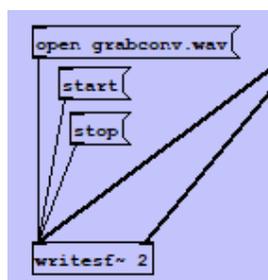


Figura 4-8: Programa con el objeto “writesf~” agregado para grabar la señal resultante en formato WAV.

Se observa en la parte inferior el objeto “writesf~”. Este se encarga de grabar la señal resultante en el momento en que se presiona el mensaje “start”, hasta que se detiene con “stop”. Este audio se graba en un archivo almacenado en la carpeta Pd, con el nombre que se desee. En este caso, se guardó con el nombre “grabconv.wav”.

En la Figura 4-9, se muestra la señal de prueba (Figura 4-1) después de ser procesada y convolucionada con sus respectivas respuestas impulso para cada posición. Se analizará la salida del sistema para las mismas posiciones de las respuestas impulsos vistas anteriormente.

Se observa, en la figura 4-9, que la señal tiene un punto máximo de amplitud cercano a 0,8 [-]. Además, se observa que la forma de onda se encuentra modificada en comparación con la voz de prueba limpia, sin convolucionar. Esto se debe a que las respuestas impulsos modifican la señal, actuando como un filtro y percibiendo auditiva y matemáticamente una diferencia, tanto interaural como en la amplitud.

Se aprecia en la figura 4-10 que la señal tiene un punto máximo de amplitud cerca de 1 [-] (máximo). También, se puede observar que la forma de onda se encuentra con un leve retardo en el tiempo, en comparación con la señal posicionada en  $e: 0^\circ$  y  $a: 90^\circ$ . Esto se debe a que las respuestas impulsos condicionan el momento, la amplitud y las frecuencias en las que será filtrada la señal a procesar.

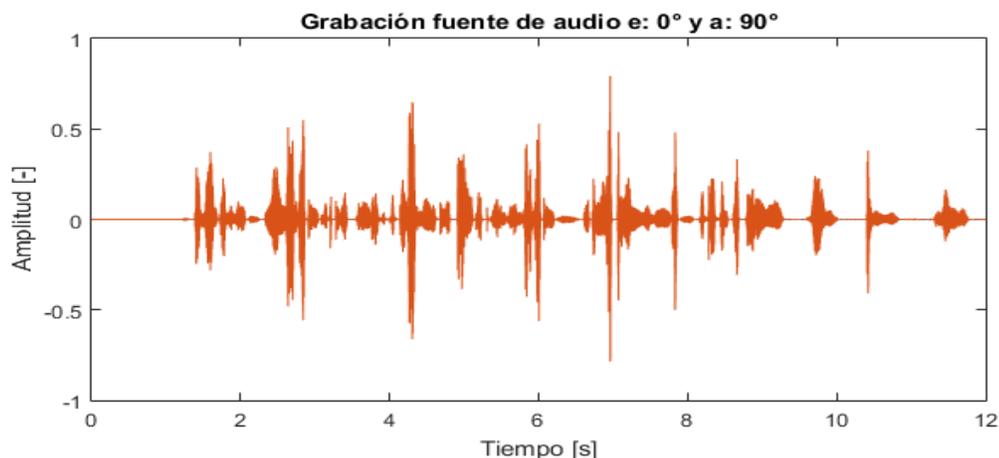


Figura 4-9: Grabación de prueba de voz auralizada con elevación de  $0^\circ$  y azimut de  $90^\circ$ .

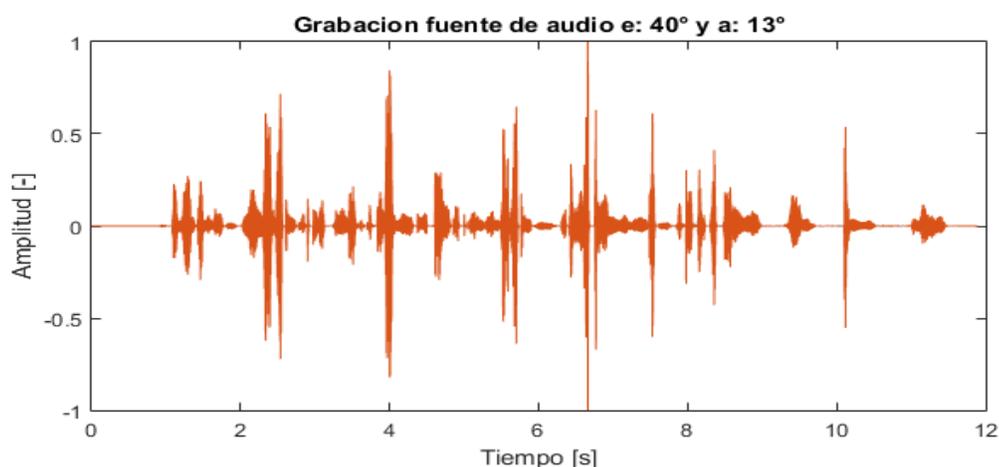


Figura 4-10: Grabación de prueba de voz auralizada con elevación de  $40^\circ$  y azimut de  $13^\circ$ .

En la figura 4-11, se observa que la señal tiene un punto máximo de amplitud cercano a 0,4 [-], bastante más reducido que las señales vistas anteriormente. Además, se puede notar una señal de color azul sobrepuesta a la roja, esta corresponde a la amplitud de la señal que recibe la oreja izquierda. Esta muestra una amplitud mayor en los picos de la onda, debido a que la fuente de sonido se encuentra más próxima a la oreja izquierda, a diferencia de los casos presentados anteriormente.

Se observa en la figura 4-11 que la señal tiene un punto máximo de amplitud cercano a 0,4 [-], bastante más reducido que las señales vistas anteriormente. Además, se puede notar una señal de color azul sobrepuesta a la roja, esta corresponde a la amplitud de la señal que recibe la oreja izquierda. Esta muestra una amplitud mayor en los picos de la onda, debido a que la fuente de sonido se encuentra más próxima a la oreja izquierda, a diferencia de los casos presentados anteriormente.

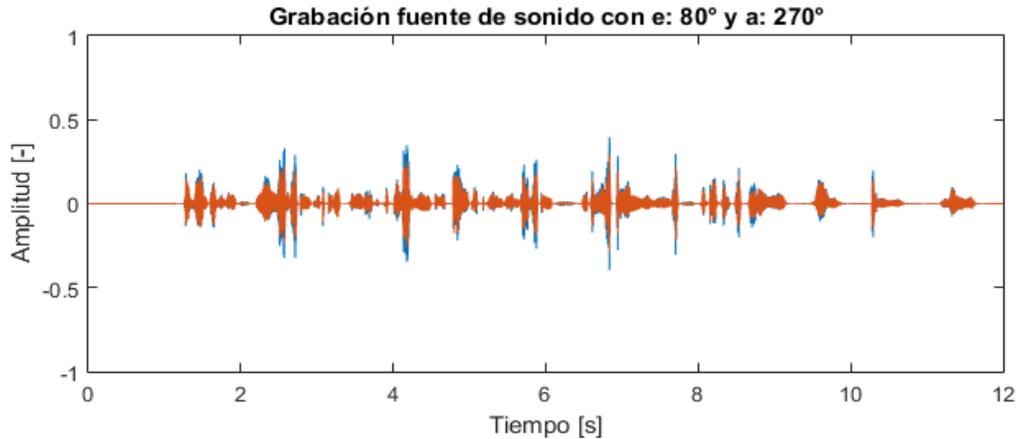


Figura 4-11: Grabación de prueba de voz auralizada con elevación de  $80^\circ$  y azimut de  $270^\circ$ .

También, se observa que la forma de onda de la figura 4-11 se aprecia diferente, en comparación a la voz de prueba limpia sin convolucionar, presentada en la figura 4-1. Esto se debe a que, el ángulo de elevación, el cual corresponde a  $80^\circ$ , hace que la onda sonora se difracte con la cabeza, provocando que las orejas perciban una especie de filtraje de la señal original. Esto modifica la amplitud haciendo que sea menor, en comparación a las señales analizadas anteriormente.

La demostración se realiza con auriculares. El oyente sentirá una fuente auditiva inicialmente en la posición de  $e: 0^\circ$  y  $a: 90^\circ$ , seguida por otra ubicada en  $e: 10^\circ$  y  $a: 70^\circ$ , luego otra en  $e: 20^\circ$  y  $a: 50^\circ$ , y así sucesivamente hasta llegar a  $e: 80^\circ$  y  $a: 270^\circ$ . Si bien, la percepción no es fluida debido a la inexistencia de interpolación entre un punto y otro, resulta perceptible el cambio de posición de la fuente. Esto mejora considerablemente con la calidad de los auriculares. Con el objetivo de facilitar la comprensión del sistema implementado, se optó por escoger un barrido de 10 posiciones diferentes. Estas siguen una trayectoria diagonal desde la derecha del oyente, con un ángulo de elevación  $0^\circ$ , hasta la izquierda, con un ángulo de elevación de  $80^\circ$ . Esta trayectoria se realiza en la zona frontal de la cabeza de quien oye.

## Discusión y conclusiones

La evidencia que se presenta corresponde a un estudio de la capacidad que tiene el ser humano para percibir el sonido, los diversos factores que están involucrados para localizar fuentes auditivas, y los distintos métodos para convolucionar señales. Esta información, necesaria para llevar a cabo los objetivos específicos del proyecto, condujo a una mayor comprensión de los fundamentos teóricos para poder implementar y simular un sistema de audio tridimensional, localizando una fuente de sonido en el espacio al ir variando su posición.

Los antecedentes teóricos sobre la auralización espacial y la implementación de la función de convolución, contribuyen a formar una base sólida para el desarrollo de este proyecto. Se desarrolla el contexto de trabajo basado en la percepción del sonido y sus propiedades, y se define la problemática de los distintos sistemas de reproducción binaural investigados. Además, se presenta el marco teórico basado en la función de convolución, cuyo entendimiento resulta de gran importancia para este trabajo.

Al estudiar en detalle la reconstrucción binaural y los métodos realizados anteriormente, junto con su funcionamiento, se logra diseñar un sistema que realiza la convolución entre 2 señales acústicas, entregando como resultado una señal filtrada. Esta señal resultante es el producto, en el dominio de la frecuencia, entre un audio cualquiera y una función de transferencia relacionada con la cabeza humana, asociada a una posición específica en el plano tridimensional. De esta manera, se confirma satisfactoriamente la hipótesis de que es posible construir un sistema de audio 3D, utilizando el método de convolver dos señales de audio.

En cuanto al hardware usado para la realización de este proyecto, se utiliza Raspberry Pi 2 en conjunto con la interfaz de audio Cirrus Logic. Estos permiten realizar un procesamiento de audio, de tamaño reducido, con una buena calidad de sonido, hecho comprobado mediante una serie de mediciones. También, se emplea el software Pure Data, que ofrece una amplia variedad de posibilidades. Este software es el encargado de tomar bloques de muestras de la entrada del conversor, y encaminarlas por el flujo de señal, diseñado gráficamente en la interfaz de usuario de Pd. En este proyecto, se consigue implementar correctamente el sistema en Pd, salvo el detalle de unos breves “clicks”, que se oyen al momento de cambiar la posición en el espacio de la señal que será auralizada. El conjunto de estos elementos (software y hardware) permite formar un sistema que se comporte como un auténtico dispositivo de procesamiento digital de señales, de reducido tamaño y con buenas prestaciones.

Por lo expuesto anteriormente, es que, para llevar a cabo una correcta implementación del simulador, resulta necesario conocer las especificaciones técnicas del hardware a utilizar para el desarrollo del

proyecto. También, es importante ilustrar el sistema de auralización y conocer el lenguaje de programación de Pure Data.

El programa expuesto en este informe realiza una muestra de 10 posiciones diferentes, cambiando su ángulo de elevación y azimut, partiendo con un ángulo de elevación de  $0^\circ$  y azimut de  $90^\circ$ , y terminando con un ángulo de elevación de  $80^\circ$  y azimut  $270^\circ$ . El ángulo de elevación se va incrementando, aproximadamente, cada  $10^\circ$ . Se escogieron estas posiciones de la base de datos obtenida de las mediciones hechas al maniquí "Kemar". Esto, para ejemplificar la diferencia en la recepción de la fuente de sonido proveniente de una posición y otra, guardando una distancia considerable para poder percibir de manera auditiva la diferencia.

Si bien la percepción no es fluida debido a la inexistencia de interpolación entre un punto y otro, resulta perceptible el cambio de posición de la fuente. Este aspecto mejora aumentando la calidad de los auriculares que se utilizan para el experimento. Con el objetivo de facilitar la comprensión del sistema implementado, se opta por escoger un barrido de 10 posiciones diferentes. Estas siguen una trayectoria diagonal desde la derecha del oyente con un ángulo de elevación  $0^\circ$ , hasta la izquierda, con un ángulo de elevación de  $80^\circ$ . Esta trayectoria se realiza en la zona frontal de la cabeza de quien escucha.

Al analizar el simulador diseñado, se percibe que la sensación de sonido inmersivo, producida por el proceso de convolución entre la señal de prueba con las señales de la base de datos de las HRTF, no es efectiva en su totalidad. Sin embargo, si se produce la espacialización del sonido en las posiciones presentadas, permitiendo localizar la fuente de sonido en el espacio.

Se cumple el objetivo de convolver las señales de audio para localizar una fuente de sonido en el espacio. Se implementó y construyó el programa de forma adecuada, y se realizaron pruebas, con resultados favorables. Aun así, existen detalles por mejorar, pero con esta base se pueden llevar a cabo los objetivos específicos y generales del proyecto final. El presente informe proporciona una base sólida respecto a la auralización espacial, presentando estudios tanto de los sistemas ya diseñados, como también de la teoría física involucrada en la acústica.

Al concluir este informe, es inevitable pensar en todas las funciones que contienen los softwares estudiados. Estas funciones brindan una ayuda considerable y proporcionan información valiosa para el objetivo final del proyecto. El trabajo en Pure Data se basa en la implementación de objetos, funciones y aplicaciones relacionados con el procesamiento digital de señales de audio. De esta forma, se llega a los resultados necesarios para llevar a cabo el desarrollo de una espacialización de fuente de sonido.

Como trabajo futuro, la obtención exacta de un sistema de simulación de audio 3D, para realizar pruebas de localización de fuentes sonoras en el laboratorio de acústica y vibraciones, resulta muy importante. Esto, debido a que se requiere un correcto entendimiento sobre la temática y los distintos sistemas de simulación existentes. Para lograr ese objetivo, este informe entrega los conocimientos necesarios para comprender e implementar un sistema de audio binaural. Presentará una gran ayuda para quien continúe con la elaboración del proyecto final en el laboratorio ("Método híbrido para la obtención de funciones de transferencia relacionadas con la cabeza (HRTF): moldeo de oreja y reconstrucción 3D fotogramétrica de la cabeza y torso").

Finalmente, resulta interesante poner atención al sistema de localización de audio del ser humano, el cual es tridimensional. Es increíble como este sistema auditivo, teniendo dos receptores, es capaz de

percibir y localizar fuentes de sonido en tres planos diferentes (tres dimensiones). También, es curioso cómo, teniendo cada persona un sistema auditivo diferente, se llega a tener una percepción muy similar de los sonidos, y cómo influye la capacidad de calibrar nuestro sentido auditivo para asignar las posiciones espaciales.

# Bibliografía

- [1] «Pd-Tutorial,» [En línea]. Available: <http://www.pd-tutorial.com>.
- [2] P. B. Botella, «Procesador de sonido y estudio de audio 3D en tiempo real,» Universidad Politecnica de Valencia , 2012.
- [3] J. Vignolo, Introducción al Procesamiento Digital de Señales, Valparaiso - Chile: Ediciones universitarias de Valparaiso - PUCV, 2008.
- [4] W. Cheng, «Representationsof HRTFs in time, frecuency, and space.».
- [5] L. Fuster, «Compensación del efecto de la sala de reproducción en Wave-Field Synthesis».
- [6] S. Moreau., «"Further Investigations of High Order Ambisonics and Wave Field Synthesis for Holophonic Sound Imaging.",» 2003.
- [7] P. Hofman, «Relearning sound localization with new ears,» 1998.
- [8] W. Gardner, «"mediciones HRTF de un KEMAR"».
- [9] À. L. Xaus, «"Sonido 3D: La percepcion acústica egocentrica y principios de sistemas espaciales",» 2009.
- [10] M. Vorländer, "Auralization: Fundamentals of Acoustics, Modelling, Simulation, Algorithms and Acoustic Virtual Reality.", 2008.
- [11] D. R. Begault, «"3-D Sound for Virtual Reality and Multimedia",» California, 2000.
- [12] J. Eargle, "Sound Recording", 1980.
- [13] Y. Kahana, «Numerical Modelling of the Transfer Functions of a Dummy-Head and of the External Ear",» 1999.

[14] P. Satarzadeh, «"A study of physical and circuit models of the human pinna"».

[15] «<https://audyssey.com/>,» [En línea].

[16] «<https://www.raspberrypi.org/>,» [En línea].

[17] H. P. - G. S. University, «<http://hyperphysics.phy-astr.gsu.edu/>,» [En línea].

# A Apéndice

En este apartado se detallan los pasos para llevar a cabo las instalaciones de la Raspberry Pi 2, la interfaz de audio Cirrus Logic y el software Pure Data.

## A.1 Instalación de Raspbian (Sistema Operativo)

El sistema operativo a instalar es Raspbian Debian (Linux) optimizado para Raspberry Pi. En concreto se instalará la versión Jessie (Raspbian Jessie). Esta instalación puede realizarse de dos maneras, una es montando una imagen de disco del SO en la tarjeta microSD, y la otra es mediante NOOBS, descomprimido y cargado en la tarjeta microSD, este es un instalador que contiene varios SO incorporados para poder elegir durante el proceso de instalación. Ambos pueden ser obtenidos en el apartado de descargas de la página oficial de Raspberry Pi, encontrando también una guía de configuración para NOOBS: “NOOBS setup guide”.

Previo a encender la Raspberry Pi 2 se debe conectar teclado, mouse, HDMI (pantalla o monitor), cable Ethernet (este no es estrictamente necesario), además de incorporar la propia tarjeta micro SD con el SO o NOOBS. Una vez puesta a punto se conecta el cable de alimentación, así como se muestra en la Figura A-1, de esta manera se encenderá la Raspberry Pi 2. Finalmente, si se ha utilizado NOOBS, se deberá seleccionar Raspbian como SO a instalar.

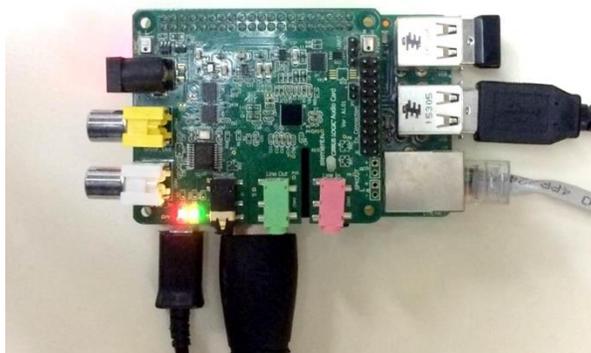


Figura A-1: *Raspberry Pi 2 conectada y encendida.*

Una vez que se instala el sistema operativo el equipo se reiniciara. En esta ocasión el inicio se realiza directamente sobre el escritorio de la interfaz gráfica, sin solicitar usuario y contraseña. Aunque cabe señalar que en algunas versiones es posible que, al iniciarse aparezca la línea de comandos solicitando usuario y contraseña, en tal caso, la configuración por defecto corresponde a:

- Username: pi
- Password: raspberry

## A.2 Configurar Raspi-config

Existe una herramienta de configuración básica de Raspbian la cual se denomina “raspi-config”, esta puede ser ejecutada mediante línea de comandos:

- `sudo raspi-config`

En la Figura A-2 se muestra la interfaz gráfica de la herramienta de configuración “Raspi-config”. Al iniciar la ventana de configuración, lo primero que se debe hacer es ejecutar “Expand Filesystem” para así asegurar que toda la capacidad de memoria de la tarjeta SD es utilizable. En el caso de utilizar NOOBS para la instalación del SO no es necesario, aun así, es aconsejable.

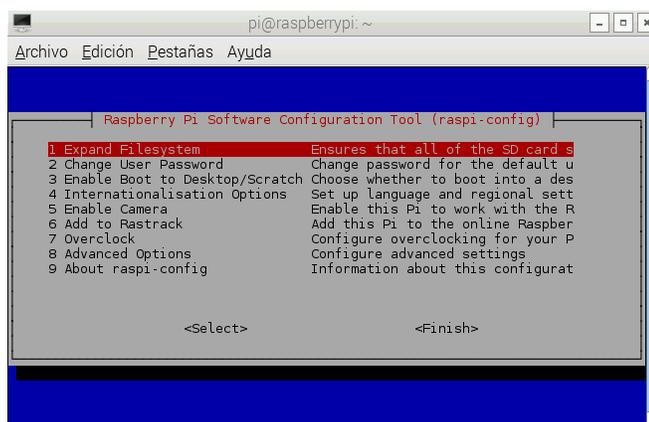


Figura A-2: Raspi-config

La segunda opción es Change User Password, esta permite cambiar la contraseña. La tercera opción, Boot Options, permite elegir entre cuatro opciones de inicio del sistema y la cuarta opción, Wait for Network al Boot, permite elegir entre inicio rápido sin esperar conexión de red o esperar a tener conexión de red antes de iniciar.

La quinta opción, Internationalization options, que se muestra en la imagen de la Figura A-3, permite cambiar el idioma y configuración regional, la zona horaria y el modelo de teclado.

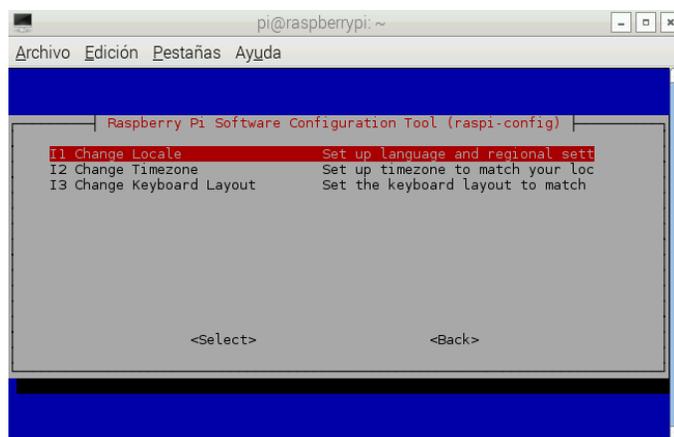


Figura A-3: Internationalization options

El proceso para conseguir la configuración de teclado a través de la opción de disposición del teclado, Change keyboard layout, se muestra en la Figura A-4, seleccionando las siguientes opciones:

- Keyboard Model: PC genérico 105 teclas (intl)
- Keyboard Layout: Otro
- Country of origin for the keyboard: español
- Keyboard Layout: español

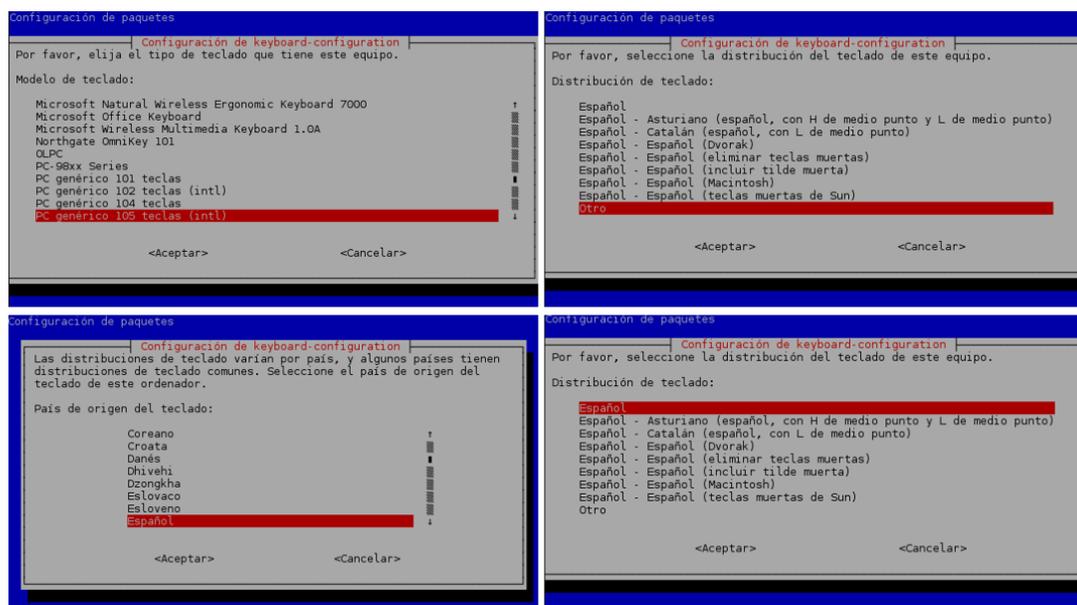


Figura A-4: Configuración de teclado

Una vez terminada la configuración se pulsa “finish” y se reinicia la Raspberry. Para reiniciar desde línea de comandos se puede ejecutar:

- ~\$ sudo reboot

### A.3 Instalación y configuración de la tarjeta de sonido

Para esta implementación es necesario utilizar una tarjeta de sonido de alta calidad que actúe como interfaz I/O (entrada/salida), con el objetivo de que el sistema funcione en tiempo real como DSP. Se ha optado por la "Cirrus Logic Audio Card", la cual cuenta principalmente con: entradas y salidas analógicas (24 [bits], 192 [kHz]) de nivel de línea; entrada y salida de audio digital (SPDIF); y una salida de auriculares de alta calidad que permite la utilización de micrófono. Para este proyecto se utilizaron las entradas y salidas: Line In/Out y Headphone Out.

Existen dos formas de instalar los drivers, una forma es instalando en la tarjeta microSD una imagen de disco que aporta Cirrus Logic con el sistema operativo y los drivers ya instalados. Es importante señalar que esta configuración de la tarjeta puede tener problemas con actualizaciones del SO.

La otra forma corresponde a instalar el conjunto de parches del Kernel (núcleo) en el SO. Esta es una opción más tediosa, sin embargo, es una actualización para el SO, por lo que no alterará cambios realizados anteriormente en el SO o en las aplicaciones.

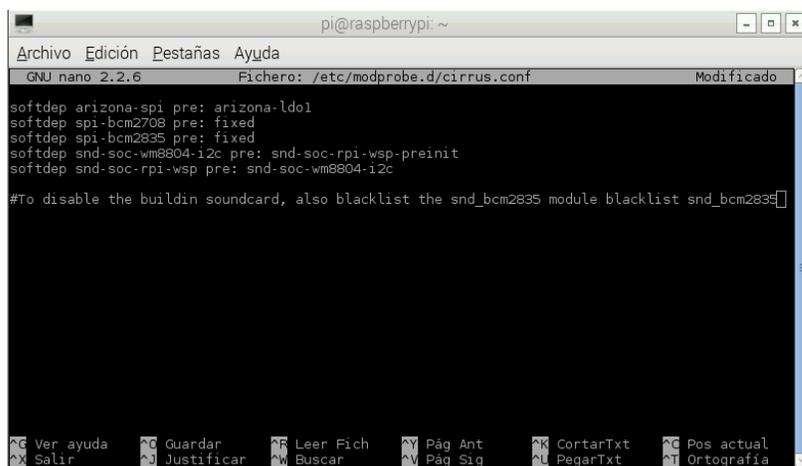
Al realizar la instalación, en primera instancia se optó por la segunda opción. Para ello, se debió seguir un tutorial que contenía una serie de comandos para ingresar. Una vez seguido paso a paso el tutorial se encontró un error al ingresar uno de los comandos, por lo que no se pudo continuar con la instalación. Este problema fue notificado al profesor, el cual junto con el estudiante reviso el error. Luego, se aconsejó realizar la instalación con la primera opción, esta no tuvo ningún inconveniente y una vez finalizada se debió editar y añadir las siguientes líneas a `"/etc/modprobe.d/cirrus.conf"`, tal y como se muestra en la Figura A-5.

```
• ~$ sudo nano /etc/modprobe.d/cirrus.conf
softdep arizona-spi pre: arizona-ldo1
softdep spi-bcm2708 pre: fixed
softdep spi-bcm2835 pre: fixed
softdep snd-soc-wm8804-i2c pre: snd-soc-rpi-wsp-preinit
softdep snd-soc-rpi-wsp pre: snd-soc-wm8804-i2c

# To disable the builtin soundcard, also blacklist the `snd_bcm2835` module
blacklist snd_bcm2835
```

Figura A-5: Comando tarjeta de sonido por defecto Cirrus Logic

El último comando permitirá que la tarjeta de sonido por defecto sea la Cirrus Logic, inhabilitando la tarjeta de sonido integrada de la Raspberry Pi 2.



```

pi@raspberrypi: ~
Archivo Edición Pestañas Ayuda
GNU nano 2.2.6 Archivo: /etc/modprobe.d/cirrus.conf Modificado
softdep arizona-spi pre: arizona-ldo1
softdep spi-bcm2708 pre: fixed
softdep spi-bcm2835 pre: fixed
softdep snd-soc-wm8904-i2c pre: snd-soc-rpi-wsp-preinit
softdep snd-soc-rpi-wsp pre: snd-soc-wm8904-i2c
#To disable the builtin soundcard, also blacklist the snd_bcm2835 module blacklist snd_bcm2835
Ver ayuda Guardar Leer Fich Páq Ant CortarTxt Pos actual
Salir Justificar Buscar Páq Sig PegarTxt Ortografía

```

Figura A-6: Edición del archivo “cirrus.conf”

Finalmente, se debe reiniciar la Raspberry Pi para que los cambios tengan efecto.

Para configurar las entradas y salidas de la tarjeta de sonido, Cirrus Logic aporta unos scripts en bash descargables desde “GitHub”. Una vez descargados y descomprimidos, en la carpeta “/wiki-content-master/scripts” se encuentran un conjunto de archivos .sh que al ejecutarse configuran la tarjeta de audio. En la siguiente figura se muestran estos ficheros.



Figura A-7: Scripts de configuración de las entradas y salidas de la tarjeta Cirrus Logic

Estos archivos .sh se pueden utilizar los siguientes comandos:

```

~$ sh <archivo>.sh
~$ bash <archivo>.sh
~$ ./<archivo>.sh

```

Para realizar una configuración partiendo de cero, se tiene el archivo Reset\_paths.sh el cual desconfigura todas las entradas y salidas, quedando inhabilitadas hasta que se ejecuten los archivos de configuración .sh de cada una de ellas. En este caso, por ejemplo, se quiere reproducir por Line Out y por HeadSet, así como grabar audio por la entrada Line In, luego desde la carpeta en la que se encuentran los scripts se debe ejecutar:

```
~$ sh Reset_paths.sh  
~$ sh Playback_to_Lineout.sh  
~$ sh Playback_to_Headset.sh  
~$ sh Record_from_lineIn.sh
```

Finalmente, una vez ejecutados estos comandos, las entradas y salidas correspondientes quedarán configuradas. Si se desea configurar de otra forma la tarjeta, basta con ejecutar los scripts necesarios, siempre ejecutando en primer lugar el archivo `Reset_paths..`