

# ALGORITMO DE REVERBERACIÓN ESPACIALIZADA DENTRO DE UN RECINTO SIMULADO

**ALGORITMO DE REVERBERACIÓN ESPACIALIZADA DENTRO DE UN  
RECINTO SIMULADO**

**MIGUEL ÁNGEL AMADO DEVIA  
DIRECTOR DEL PROYECTO: RICARDO QUINTANA SOLER**

**PONTIFICIA UNIVERSIDAD JAVERIANA  
FACULTAD DE ARTES  
CARRERA DE ESTUDIOS MUSICALES  
ÉNFASIS EN INGENIERÍA DE SONIDO  
BOGOTÁ D.C  
2018**

## **Agradecimientos**

*A los compañeros de la carrera de Estudios Musicales que participaron en el proyecto, en especial a Danna Botero, María Paula Contreras y Felipe Warren.*

*A mi familia, en especial a mi Madre, Abuelos y Tíos, por su apoyo incondicional, dar consejos de vida y ser un ejemplo que seguir.*

*Al Ingeniero y Profesor Ricardo Quintana por todo el apoyo, la motivación y la enseñanza durante todo el transcurso de desarrollo del proyecto. Al Ingeniero y Profesor Ricardo Escallón por su asesoría, supervisión y apoyo.*

*A los profesores tanto de música como del énfasis por todo el carisma y conocimiento que transmiten por medio de sus enseñanzas.*

## Contenido

<b>1). Introducción</b> .....	5
<b>2). Objetivos</b> .....	6
<b>2.1). Objetivo General</b> .....	6
<b>2.2). Objetivos Específicos</b> .....	6
<b>3). Marco Teórico</b> .....	7
<b>3.1). Fuente Sonora</b> .....	7
<b>3.1.1). Directividad</b> .....	7
<b>3.1.2). Ley del Inverso de la distancia</b> .....	10
<b>3.1.3). Retraso debido a la distancia</b> .....	12
<b>3.2). Características Acústicas de un Recinto</b> .....	13
<b>3.2.1). Reflexión</b> .....	13
<b>3.2.2). Respuesta al Impulso</b> .....	16
<b>3.3). Reverberación a partir de la convolución</b> .....	17
<b>3.4). Obtención teórica de la Respuesta al Impulso con parámetros de un recinto</b> .....	17
<b>3.4.1). Caso de Estudio: Método Fuente-Imagen</b> .....	17
<b>3.5). Reverberación a partir de red de retrasos</b> .....	21
<b>3.5.1). <i>Feedback Delay Network</i></b> .....	21
<b>3.6). Reverberación a partir de red de retrasos y parámetros de un recinto</b> .....	22
<b>3.6.1). Caso de estudio: <i>Scattering Delay Network</i></b> .....	23
<b>4). Desarrollo</b> .....	26
<b>4.1). Implementación de los algoritmos sin espacialización</b> .....	27
<b>4.1.1). <i>Feedback Delay Network</i></b> .....	27
<b>4.1.2). Fuente Imagen</b> .....	28
<b>4.1.3). <i>Scattering Delay Network</i></b> .....	30
<b>4.2). Implementación de los algoritmos con espacialización</b> .....	33
<b>4.2.1). <i>Feedback Delay Network</i></b> .....	33
<b>4.2.2). <i>Scattering Delay Network</i></b> .....	34
<b>4.2.3). Fuente Imagen</b> .....	35
<b>4.3). Validación subjetiva</b> .....	39
<b>4.3.1). Procedimiento previo</b> .....	39
<b>4.3.2). Proceso de evaluación</b> .....	42
<b>4.3.3). Resultados</b> .....	43
<b>4.4). Aplicación artística del algoritmo escogido</b> .....	45
<b>4.4.1). Grabación del video y los audios</b> .....	45
<b>4.4.2). Procesamiento de los audios</b> .....	47
<b>5). Conclusiones</b> .....	50
<b>6). Bibliografía</b> .....	52
<b>Anexos</b> .....	53

## 1). INTRODUCCIÓN

A lo largo del desarrollo de las herramientas tecnológicas para la ingeniería de sonido en pro del arte, uno de los objetivos que ha prevalecido al momento de usarlas es presentarle al oyente una realidad diferente. Durante la presente década, uno de los recursos que se ha incorporado en estas herramientas es la espacialización. Esta se deriva de la posibilidad que los seres humanos (y otros seres vivos) tenemos de localizar el sonido proveniente de cualquier dirección, incluso infiriendo la distancia a la que se encuentra. Otro recurso que se complementa con la localización del sonido es la reverberación, al cual estamos acostumbrados de forma natural cuando estamos en algún lugar determinado (iglesia, cuarto, estudio, etc.). Ambos permiten construir una realidad sonora tal que para nuestra percepción auditiva es información espaciotemporal de la trayectoria del sonido.

La información del párrafo anterior se puede sintetizar en lo que sería una posible mejora de la famosa estereofonía. Este elemento ha ido evolucionando con el paso del tiempo, a través de diferentes maneras: el audio multicanal<sup>1</sup> y el Audio 3D<sup>2</sup> principalmente. Gracias a estos campos de desarrollo del audio, se tiene la posibilidad de apreciar arte sonoro cada vez más atractivo para los oyentes. Teniendo en cuenta esto, la meta de algún formato o propuesta algorítmica de audio inmersivo, no siendo este proyecto una excepción, es presentar al oyente una experiencia sonora tal que se sienta como parte de esta.

En este proyecto se trabajará la simulación del comportamiento del sonido emitido por una fuente hacia un receptor, dentro de un recinto rectangular. Con base en esto, el problema principal que se busca resolver en este proyecto es combinar la espacialización sonora con la reverberación modelada a partir de tal simulación.

---

<sup>1</sup> CCM. "Multichannel Audio (5.1, 6.1, 7.1)". < <https://ccm.net/contents/50-multichannel-audio-5-1-6-1-7-1>>. [Consultado el 14-11-18].

<sup>2</sup> Vox Media. The Verge. "Surrounded by sound: how 3D Audio hacks your brain". © 2018 Vox Media, Inc. All Rights Reserved. < <https://www.theverge.com/2015/2/12/8021733/3d-audio-3dio-binaural-immersive-vr-sound-times-square-new-york>>. [Consultado el 14-11-18].

## **2). OBJETIVOS**

### **2.1). Objetivo General**

Solucionar el problema planteado, a través de un algoritmo que le permita al observador de un video sentir el audio con movimiento dentro de un recinto.

### **2.2). Objetivos Específicos**

- 1).** Obtener un algoritmo que procese de manera digital un archivo de audio ingresado y cuya salida sea el archivo modelado sonoramente en un recinto simulado
- 2).** Validar de manera subjetiva (acorde con la percepción auditiva) el algoritmo en cuanto a su funcionamiento
- 3).** Adaptar el algoritmo obtenido a parámetros de un recinto real.
- 4).** Aplicar el algoritmo realizado a varios archivos de audio, basados en un video que será capturado dentro del recinto

### 3). MARCO TEÓRICO

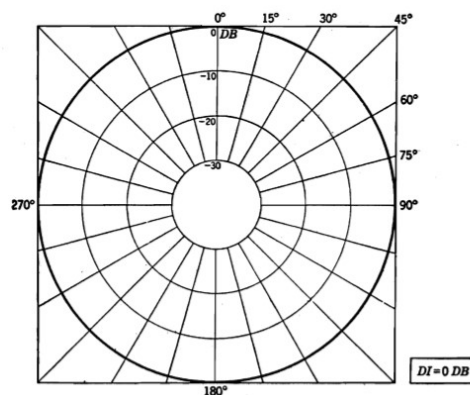
En este capítulo se abordará la explicación del comportamiento de una señal emitida por una fuente sonora dentro de un recinto, con respecto a un punto de referencia (receptor). Esto es, las modificaciones que sufre la señal desde que es emitida hasta que llega al receptor, en términos tanto conceptuales como de procesamiento de señales. Con base en estas definiciones se abordarán algunos modelos matemáticos implementados en la construcción de reverberaciones algorítmicas y en el cálculo de respuestas al impulso.

#### 3.1). Fuente Sonora

En términos de este proyecto, es un elemento que irradia energía acústica a una zona determinada del espacio en que se encuentre (p.ej. recinto). Se requiere este elemento para poder emitir un sonido cualquiera y cuantificarlo con respecto a una posición de referencia (receptor). En esta sección se abordará cómo se comporta una señal emitida por una fuente cuando hay cambios de posición del receptor o de la fuente misma.

##### 3.1.1). Directividad

La directividad es la medida de energía acústica irradiada en un ángulo determinado<sup>3</sup>. Es una medida adimensional, la cual se obtiene a partir del factor de directividad y el índice de directividad [3]. Está representada por medio del diagrama direccional (también conocido como patrón polar). En las figuras 1 y 2 se observan ejemplos de diferentes patrones de directividad de una fuente, los cuales constituyen una descripción gráfica de la respuesta del transductor en función de la dirección de la onda transmitida (o incidente, en el caso de micrófonos) en un plano y frecuencia específicos.



**Figura 1:** Diagrama direccional de una fuente omnidireccional (tomado de [3]).

<sup>3</sup> Hacıhabibog˘lu, H; G˘nel, B; Kondo˘z, A. "Source Directivity Simulation in Digital Waveguide Mesh-based Room Acoustics Models. 2007. AES 30th International Conference.

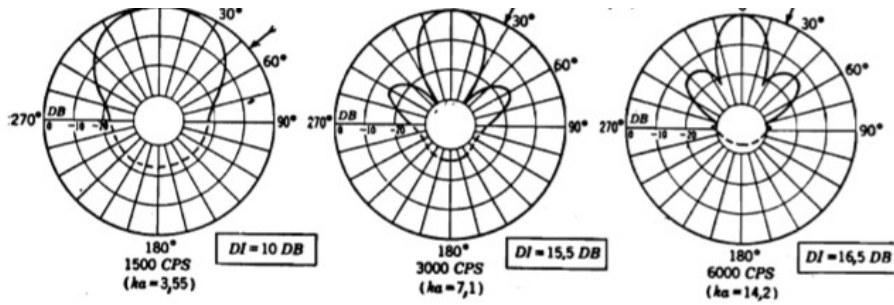


Figura 2: Diagramas direccionales de una fuente para algunas frecuencias (tomado de [3]).

Teniendo en cuenta estos diagramas direccionales, es importante aclarar que cada círculo dentro de cada diagrama representa un valor de atenuación en  $dBs$   $SPL$  (*Sound Pressure Level* – Nivel de Presión Sonora) distinto. Entonces, para calcular el valor de la atenuación de una frecuencia determinada de acuerdo al ángulo  $\theta$ , se verifica la siguiente operación:

$$\begin{aligned}
 SPL_t(\theta) &= SPL_e(\theta) + SPL_a(\theta) \\
 20 \log(P_t) &= 20(\log(P_e \cdot A_{ten}))^{4\ 5} \\
 P_t &= P_e \cdot A_{ten}
 \end{aligned}
 \tag{1}$$

De donde  $SPL_t$  es el nivel de presión sonora total,  $SPL_e$  es el nivel de presión sonora emitido por la fuente y  $SPL_a$  es el valor de atenuación en  $dBs$  (de acuerdo al diagrama direccional/patrón polar, usualmente dado por los fabricantes en los *datasheet*).  $P_t, P_e, A_{ten}$  son los valores de la presión total, emitida y atenuada respectivamente. Para determinar el valor de la presión sonora atenuada  $A_{ten}$  se acude a la definición de nivel de presión sonora (como se realizó en la ecuación (1)):

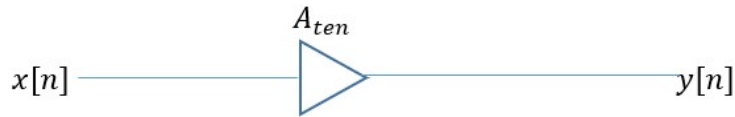
$$\begin{aligned}
 SPL_a &= 20\log(A_{ten}) \\
 A_{ten} &= 10^{\frac{SPL_a}{20}}
 \end{aligned}
 \tag{2}$$

Básicamente lo que se realiza es convertir el valor de  $SPL_a$  en  $dB$  a una escala lineal. Siendo  $y[n]$  la señal en el receptor y  $x[n]$  la señal emitida por la fuente, la representación en procesamiento se puede ver en la figura 3.

<sup>4</sup> Por propiedades de los logaritmos. “Logarithmic Properties”. Copyright ©1999–2005, Keith G. Calkins <https://www.andrews.edu/~calkins/math/webtexts/numb17.htm>. [Consultado el 23/01/19].

<sup>5</sup> Definición matemática de  $dB$   $SPL$ . “ $dB$ : What is a decibel?”. © *School of Physics - UNSW 2052* <http://www.animations.physics.unsw.edu.au/jw/dB.htm#definition>. [Consultado el 23/01/19].





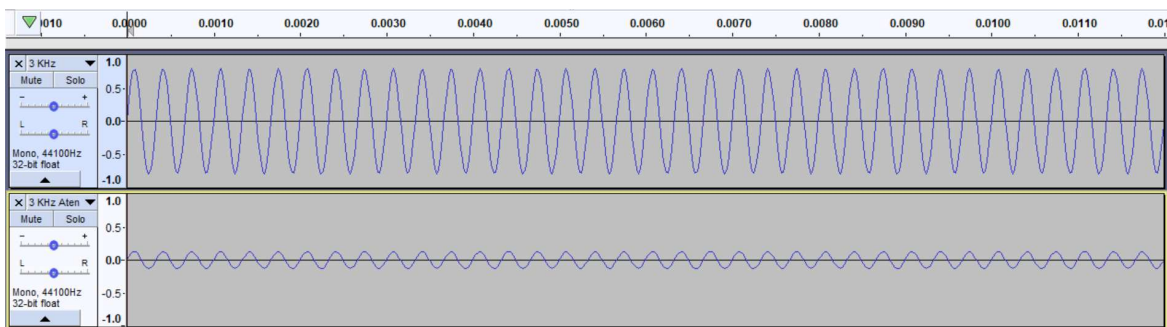
**Figura 3:** Representación en procesamiento de señales de la atenuación de la fuente de acuerdo al ángulo entre fuente y receptor.

La señal en el receptor sería:

$$y[n] = A_{ten} \cdot x[n] \quad (3)$$

Tomando como ejemplo el segundo patrón polar de la figura 2, en donde la frecuencia es 3kHz, la señal original en una *DAW* (Estación de trabajo de audio digital)<sup>6</sup> se puede apreciar en la figura 4 (arriba). Suponiendo que el receptor está a 60°, la atenuación en dBs según este patrón polar sería de -16 *dB SPL* aproximadamente (este valor corresponde a  $SPL_a$ ). La misma señal atenuada se puede apreciar también en la figura 4 (abajo). La atenuación sería:

$$A_{ten} = 10^{\frac{-16}{20}} \approx 0.16$$



**Figura 4:** Señal senoidal (arriba) de 3 KHz atenuada (abajo) en una sesión de *Audacity*

Esta atenuación se analiza para una señal senoidal ya que, como se mencionó, la directividad depende de la frecuencia. Así, en caso de que la señal emitida no sea una senoidal, la atenuación no es homogénea en todo su espectro. Con el fin de obtenerla se puede usar un filtro.

### 3.1.2). Ley del inverso de la distancia

Es importante considerar la distancia entre fuente y receptor, la cual, además del patrón de directividad definido en la sección 3.1.1, también va a causar atenuación sobre la señal emitida. La relación entre distancia e intensidad sonora está dada por

<sup>6</sup> Sistema digital diseñado para la grabación, edición y mezcla de audio digital. Kadis, J. *CCRMA. "The Digital Audio Workstations"*. ©1999-2018 Jay Kadis. All Rights Reserved. [Consultado el 16 de Septiembre de 2018]

medio de la ley del inverso cuadrado (Siendo  $I$ , la intensidad sonora,  $W$  la potencia sonora y  $A$  el área de radiación sonora):

$$I = \frac{W}{A} = \frac{W}{4\pi r^2} \quad (4)$$

Teniendo en cuenta la definición de intensidad sonora en campo libre<sup>7</sup>, se puede relacionar la presión sonora con la distancia, usando dicha definición y la ecuación (4):

$$\frac{P^2}{\rho C} = \frac{W}{4\pi r^2} \quad (5)$$

Teniendo una fuente sonora con potencia  $W$ , al relacionar la presión  $P_1$  medida a  $r_1 = 1m$  de la fuente, con  $P_2$  medida a  $r_2$  de la fuente, se tiene:

$$W = \frac{P_1^2}{\rho C} \cdot 4\pi r_1^2 = \frac{P_2^2}{\rho C} \cdot 4\pi r_2^2$$

$$\frac{P_1^2}{r_2^2} = \frac{P_2^2}{r_1^2}$$

$$\frac{P_1}{r_2} = \frac{P_2}{r_1} \quad (6)$$

Teniendo esto en cuenta, la atenuación que sufre una señal, sin pasar por ningún obstáculo (p.ej. superficie) es de  $\frac{1}{r^2}$ , siendo  $r_2$  la distancia fuente-receptor. Recopilando la directividad de la fuente sonora y la distancia entre fuente y receptor, siendo  $x(n)$  la señal emitida por la fuente en la muestra  $n$ ,  $A_{ten}$  la atenuación debida a la función de directividad de la fuente para el ángulo entre fuente y receptor; y  $g_{xy}$ , la atenuación debida a la distancia entre fuente y receptor la representación en procesamiento de señales se puede apreciar en la figura 5.



<sup>7</sup> Campo libre se refiere a que el sonido no sufre reflexiones.

$$I = \frac{P^2}{\rho C}$$

En donde  $P$  es la presión sonora,  $\rho$ , la densidad del aire y  $C$ , la velocidad del sonido. Brüel & Kjær. "Sound Intensity". 1993. Copyright © Brüel & Kjær. All Rights Reserved. [Consultado el 17 de Septiembre de 2018].

**Figura 5:** Representación en procesamiento de señales de la atenuación por la función de directividad de la fuente y distancia entre fuente y receptor.

Siendo  $pos_x$  la posición de la fuente (coordenadas en 3 dimensiones) y  $pos_y$  la posición del receptor, la señal en el receptor para la muestra  $n$  sería:

$$\begin{aligned}
 y(n) &= A_{ten} \cdot g_{xy} \cdot x(n) = A_{ten} \cdot \frac{1}{d_{xy}} \cdot x(n) \\
 &= A_{ten} \cdot \frac{1}{\|pos_x - pos_y\|} \cdot x(n) \quad ^8
 \end{aligned}
 \tag{7}$$

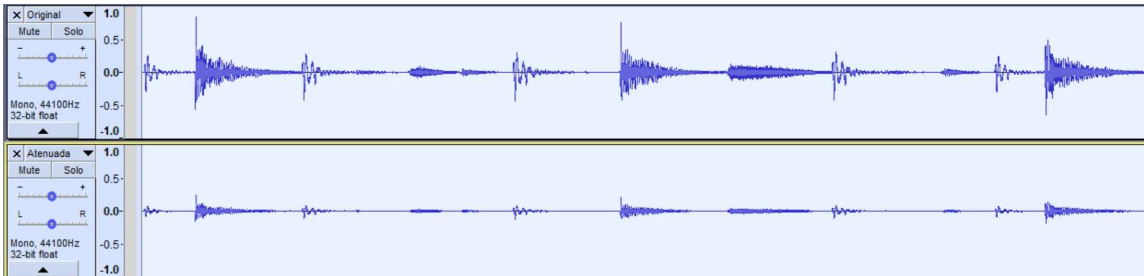
A manera de ejemplo para calcular la atenuación por distancia, suponiendo que la fuente está ubicada en las coordenadas  $[2, 3, 6]$  y el receptor está ubicado en las coordenadas  $[3, -1, 4]$ , la distancia (en metros) entre estos dos sería:

$$d_{xy} = \|[2, 3, 6] - [3, -1, 4]\| = \|-1, 4, 2\| = \sqrt{-1^2 + 4^2 + 2^2} \approx 4.582$$

La atenuación correspondiente sería:

$$g_{xy} = \frac{1}{4.582} \approx 0.206$$

Teniendo una señal en una *DAW* en la parte superior de la figura 6, se puede ver la atenuación debida a la distancia del ejemplo en la parte inferior de la misma figura.



**Figura 6:** Señal original (arriba) y señal atenuada (abajo) por inverso de la distancia en una sesión de *Audacity*

### 3.1.3). Retraso debido a la distancia

El retraso sobre una señal es un corrimiento de la misma en tiempo, que, en el contexto de este proyecto, es causado por la distancia entre dos puntos. Este se

<sup>8</sup> El operador  $\| \cdot \|$  es la norma de un vector de 3 dimensiones. Este se define:

$$\|v_{3D}\| = \sqrt{v_1^2 + v_2^2 + v_3^2}$$

obtiene calculando el tiempo que tarda la señal en viajar desde un punto al otro (fuente a receptor):

$$t_{d_{xy}} = \frac{d_{xy}}{c} \quad (8)$$

En el contexto de audio digital, es importante aclarar que el tiempo se representa a través de muestras; siendo 44100 muestras, el equivalente a 1 segundo (si la frecuencia de muestreo del sistema es de 44100Hz). Por lo tanto, para obtener el número de muestras  $m_{xy}$  que equivalen al tiempo de retraso, se utiliza la siguiente fórmula:

$$m_{xy} \cdot t_s = t_{d_{xy}}$$

$$m_{xy} = \frac{t_{d_{xy}}}{t_s} = f_s \cdot t_{d_{xy}} \quad (9)$$

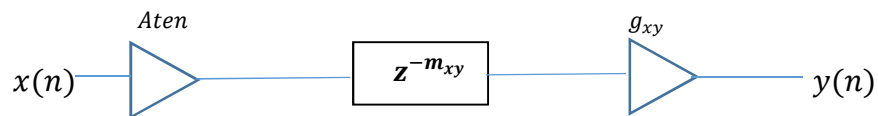
En donde  $t_s$  es el período de muestreo, el cual es el inverso de la frecuencia de muestreo:

$$t_s = \frac{1}{f_s}, \quad f_s = \frac{1}{t_s} \quad (10)$$

Es importante aclarar que no se consideran números decimales de muestras, sino enteros<sup>9</sup>. Por lo tanto, en caso de que el resultado de la división sea un número decimal, se redondea su valor al entero más cercano:

$$m_{xy} = \text{round}(f_s \cdot t_{d_{xy}}) \quad (11)$$

Añadiendo el retraso a la cadena de procesamiento, su representación se puede apreciar en la figura 7.



**Figura 7:** Representación en procesamiento de señales de la atenuación por la función de directividad de la fuente, el retraso de la señal debido a la distancia y la atenuación por la distancia entre fuente y receptor.

<sup>9</sup> Está ligado al hecho de que las señales digitales de audio se representan como secuencias; una secuencia es una función cuyo dominio es el conjunto de los números enteros.  $x = \{x[n]\}$ ,  $-\infty < n < \infty$ . Una secuencia se representa como una variable tipo vector en *Matlab*, cuyos índices (posiciones) son números enteros, los cuales representan las muestras.

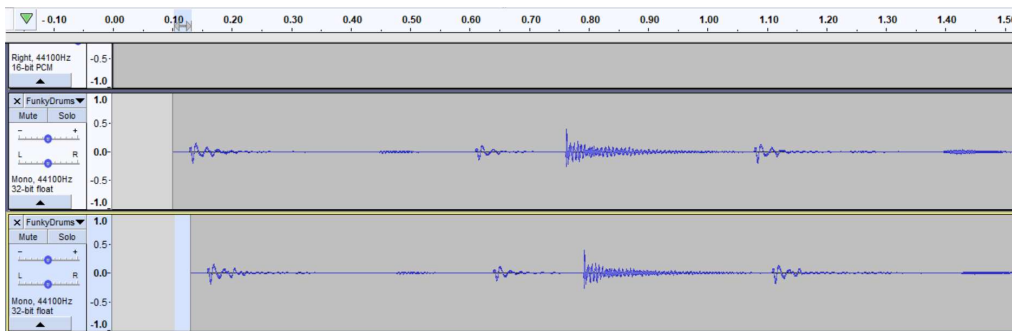
En la anterior figura, la expresión  $z^{-m_{xy}}$  representa el retraso de  $m_{xy}$  muestras que se aplica a la señal. La señal en el receptor sería:

$$y[n] = A_{ten} \cdot g_{xy} \cdot x[n - m_{xy}] \quad (12)$$

A manera de ejemplo, si la distancia entre fuente y receptor fuera de 10 metros, el retraso usando la ecuación (11) sería: (siendo  $f_s = 44100$ )

$$m_{xy} = \text{round}\left(f_s \cdot \frac{10}{340}\right) = 1297 \text{ muestras}$$

El valor en segundos  $t_{dxy}$  sería de 0.029s, o 29 ms. El retraso de la señal se puede apreciar en la figura 8.



**Figura 8:** Señal original (arriba) y señal retrasada (abajo) en una sesión de *Audacity*. La franja en azul claro es el tiempo correspondiente al retraso.

### 3.2). Características Acústicas de un recinto

El estudio del sonido dentro de un recinto cubre no sólo la búsqueda de cómo los sonidos se reflejan, sino también cómo medir el sonido bajo esas condiciones y el efecto (absorción) que los materiales causan sobre este. Para este proyecto, el recinto a trabajar es el de forma rectangular. Los parámetros con los que se trabajarán son las dimensiones, los coeficientes de reflexión de cada superficie y la posición tanto de la fuente sonora como del receptor.

#### 3.2.1). Reflexión

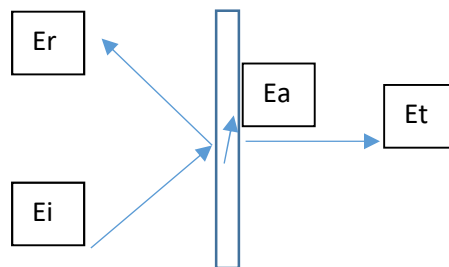
La interacción del sonido con superficies sólidas puede ser considerado como el inicio de la acústica arquitectónica<sup>10</sup>. Se dan tres tipos de interacción al momento de su encuentro con un obstáculo: reflexión, absorción y transmisión. Esto quiere decir que cuando las ondas interactúan con los materiales, la energía que contienen estas ondas que inciden es reflejada, transmitida a través del material y absorbida por el mismo.

<sup>10</sup> Long, M. *Architectural Acoustics. Second Edition*. Editorial Academic Press.

La distribución de la energía se da de la siguiente manera ( $E_i$  es la energía de la señal emitida,  $E_r$  es la energía de la señal reflejada en la superficie,  $E_t$  es la energía de la señal transmitida por la superficie y  $E_a$  es la energía de la señal absorbida por la superficie):

$$E_i = E_r + E_t + E_a \quad (13)$$

Con base a la ecuación (13) se puede considerar una representación gráfica de la distribución de energía en la figura 9.



**Figura 9:** Distribución de la energía en la interacción sonido-superficie.

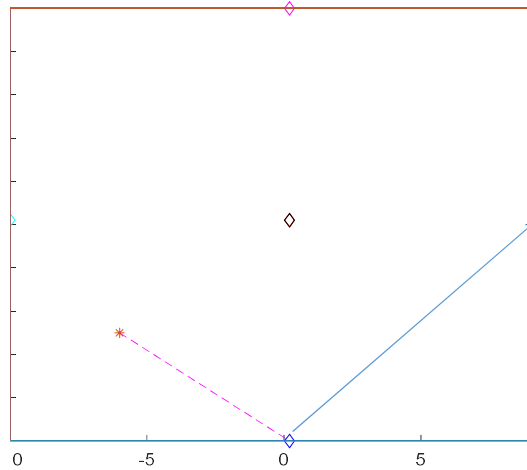
Teniendo en cuenta esta distribución de energía, es posible abordar el comportamiento de la señal dentro de un recinto, para este proyecto, por medio de los coeficientes de absorción y reflexión. Se puede definir el coeficiente de absorción  $\alpha$  como la relación entre intensidad sonora absorbida por la superficie ( $I_a$ ) y la intensidad sonora que incide sobre la superficie ( $I_i$ )<sup>11</sup>:

$$\alpha = \frac{I_a}{I_i} \quad (14)$$

Ahora se analiza el comportamiento de la señal, considerando una superficie (una reflexión) antes de llegar al receptor. Con base en las ecuaciones (3), (7) y (12), se hace el mismo análisis incluyendo una reflexión antes de llegar al receptor en la trayectoria de la señal. Esta vez se consideran las distancias fuente a superficie y superficie a receptor. La posición de un punto en la superficie (figura 10) es  $pos_{sf}$ . Teniendo en cuenta esto y las ecuaciones mencionadas, ahora la atenuación sería:

<sup>11</sup> Kuczmariski; M; Johnston, J. NASA. "Acoustic Absorption in Porous Materials". 2011. <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20110011143.pdf>. [Consultado el 20-09-18]

$$g_{xy} = \frac{1}{\|pos_x - pos_{sf}\| + \|pos_{sf} - pos_y\|}$$

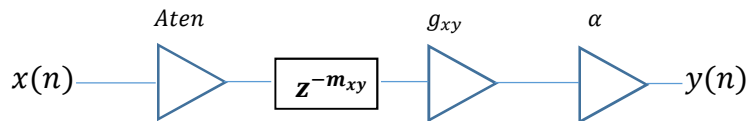


**Figura 10:** Esquema del recinto simulado en 2D (vista de planta). La distancia entre fuente (\*) y una de las superficies es la línea segmentada (- -). La distancia entre superficie y receptor (+) es la línea continua (\_\_\_).

Siendo  $d_{xy} = \|pos_x - pos_{sf}\| + \|pos_{sf} - pos_y\|$  la distancia entre fuente y receptor considerando una superficie, el retraso debido a esta distancia sería:

$$m_{xy} = \text{round}\left(f_s \cdot \frac{d_{xy}}{c}\right)$$

Añadiendo el coeficiente de absorción  $\alpha$ , la cadena de procesamiento total desde fuente a receptor (pasando por una superficie) se puede apreciar en la figura 11.



**Figura 11:** Representación en procesamiento de señales de la trayectoria de la señal desde la fuente, pasando por una superficie, hasta el receptor.

Finalmente, la señal definitiva sobre el receptor con una reflexión sería:

$$y[n] = A_{ten} \cdot g_{xy} \cdot \alpha \cdot x[n - m_{xy}] \quad (15)$$

En una DAW, se puede visualizar un impulso (figura 12) y una señal senoidal (figura 13) como señales de entrada, pasando por una reflexión.



Figura 12: Impulso y una reflexión en Audacity.

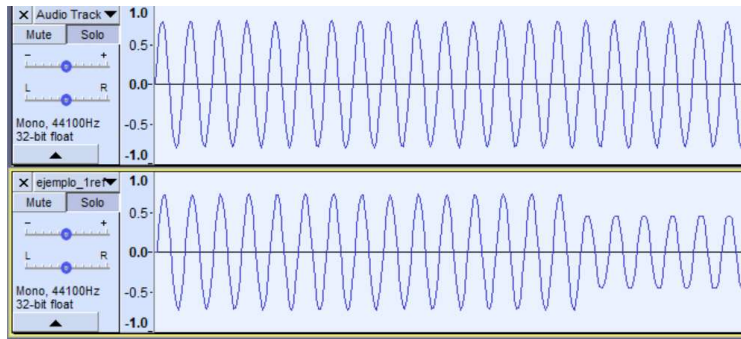


Figura 13: Señal senoidal original (arriba) y señal senoidal luego de una reflexión (abajo) en Audacity.

Se puede observar en la figura 13 que cuando una señal continua sufre una reflexión posee variaciones en amplitud y fase. En este caso, hay atenuación de la señal debida a la diferencia de fase entre señal original y señal reflejada.

### 3.2.2). Respuesta al Impulso de un recinto

Desde el punto de vista acústico, con base en el análisis de la señal luego de una reflexión en la sección 3.2.1, y en la figura 12, la respuesta al impulso es el equivalente a tener muchas reflexiones (Figura 14). La respuesta al impulso caracteriza la transformación hecha al sonido emitido. En términos de procesamiento de señales, esto es posible gracias al operador convolución, el cual permite obtener el efecto de reverberación cuando se operan la señal original y la respuesta al impulso del lugar en el que se quiera escuchar dicha señal (sección 3.3).

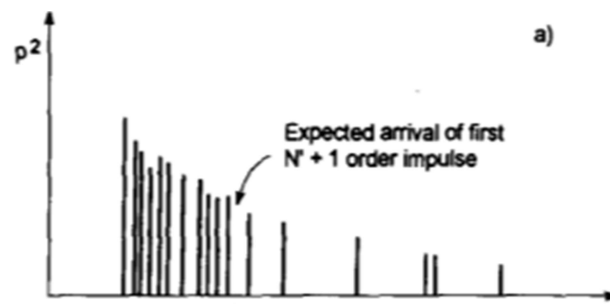


Figura 14: Respuesta al impulso de un recinto para un sonido corto. Tomada de: *Extending the Image Method to Higher-Order Reflections*, Kristiansen, U; Krokstad, A; Follestad, T.



### 3.3). Reverberación a partir de la convolución

La convolución es la operación entre dos señales de tal manera que el resultado muestra cómo una afecta a la otra. Esta operación obtiene la señal resultante al pasar por las reflexiones de la respuesta al impulso deseada cuando se conoce la señal emitida por la fuente. Como se mencionó en la sección 3.2.2, el operador convolución permite operar la respuesta al impulso de un sistema (recinto) determinado con cualquier señal de entrada para obtener la salida respecto a esta. En otras palabras, a través de este operador se puede analizar cómo una señal (entrada) es afectada por otra (respuesta al impulso). El operador convolución se define así:

$$(h * x)[n] = \int_0^n h[k] \cdot x[n - k]$$
$$y[n] = \sum_{k=1}^{M-1} h[k] \cdot x[n - k] \quad 12 \tag{16}$$

Siendo  $y[n]$  la señal resultante,  $h[k]$  la respuesta al impulso,  $M$  el número de muestras (tamaño/longitud) de la respuesta al impulso y  $x[n]$  la señal emitida por la fuente, evaluados en la muestra  $n$ . El tamaño de la salida  $y$  va a ser de  $N + M - 1$ , siendo  $N$  el tamaño de la entrada  $x$ .

### 3.4). Obtención de la Respuesta al Impulso con parámetros de un recinto

Una vez explicado el concepto de convolución, en esta sección se explicará un método para calcular la respuesta al impulso de forma teórica, sin necesidad de una medición.

#### 3.4.1). Caso de estudio: Método Fuente-Imagen

El principio en el cual se basa el método Fuente Imagen es en el de la geometría de modelamiento acústico de recintos, en el que el concepto de onda sonora se interpreta como un rayo<sup>13</sup>: trayectoria recta entre puntos en el espacio. Sin embargo, los parámetros vistos hasta el momento (reflexión, absorción, distancia) son los que determinan el comportamiento adecuado del rayo.

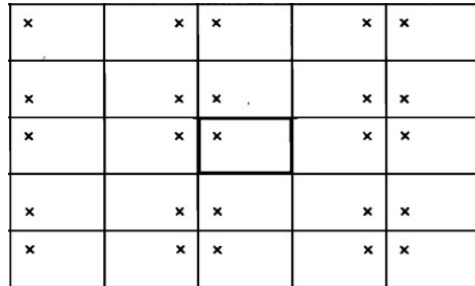
El método de imágenes es comúnmente usado para analizar las propiedades

---

<sup>12</sup> La sumatoria en este caso empieza en  $k = 1$ , ya que en *Matlab* los vectores inician en la posición 1, a diferencia de otros lenguajes en donde los vectores inician en la posición 0.

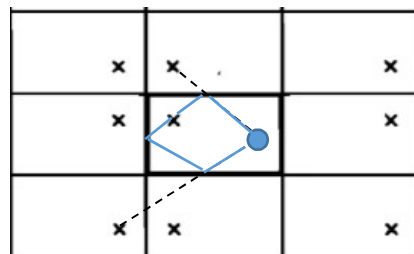
<sup>13</sup> *School of Engineering and Applied Science, Princeton University. Survey of Methods for Modeling Sound Propagation in Interactive Virtual Environment Systems. © 2018 The Trustees of Princeton University. [Consultado el 27/08/18].*

acústicas de recintos<sup>14</sup>. En este método, se enumeran los pasos de propagación del sonido entre fuente y receptor; en otras palabras, se enumera la cantidad de reflexiones. A medida que se consideran más pasos de propagación, más fuentes imagen se consideran en el método<sup>15</sup>. Básicamente estas fuentes imagen son copias de la señal original, a una distancia determinada, como si proviniera de afuera del recinto (Figura 15).



**Figura 15:** Esquema de fuente real y fuentes imagen (o virtuales). El rectángulo resaltado es el recinto y la fuente real se encuentra dentro de este. Tomada de *Extension of the image model to arbitrary polyhedra*, Borish, J.

Se puede apreciar que las fuentes virtuales están distribuidas organizadamente respecto a la fuente real, dado que el recinto es rectangular<sup>16</sup>. La forma en que se distribuyen espacialmente las fuentes virtuales es a manera de espejo, como si este se colocara en cada borde del recinto. Y así con todas las demás fuentes imagen, dependiendo de cuántas reflexiones especulares se consideren para determinar la longitud de la respuesta al impulso. En la figura 16 se puede apreciar la trayectoria de la señal para una reflexión de primer orden y para una de segundo orden. Geométricamente, la trayectoria de las reflexiones en el recinto real (color azul) es equivalente (en términos de distancia) a la trayectoria recta entre cada fuente imagen y el receptor (color negro, punteada). Esto se debe a que la proyección de la fuente imagen es a manera de espejo.



<sup>14</sup> Allen, J; Berkeley, D. *Image Method for efficiently simulating small-room acoustics*. 1979. ©1979 Acoustical Society of America.

<sup>15</sup> Kristiansen, U; Krokstad, A; Follestad, T. *Extending the Image Method to Higher-Order Reflections*. 1993. © 1993 Elsevier Science Publishers Ltd, England. Printed in Great Britain

<sup>16</sup> Borish, J. *CCRMA. Extension of the Image model to arbitrary polyhedra*. 1984. ©1984 Acoustical Society of America

**Figura 16:** Distancia entre fuente imagen y receptor (fuente (x), receptor (O)). Tomada de *Extension of the image model to arbitrary polyhedra*, Borish, J.

El método fuente imagen estudiado en este caso es el de Viveiros y Gibbs<sup>17</sup>. Para empezar el análisis, es importante definir la ubicación de las fuentes imagen. Primero, se define el concepto de “celda imagen”, el cual es básicamente un recinto virtual en el cual se encuentra una fuente imagen. Hay tantos recintos virtuales como fuentes imagen. El número del recinto virtual en donde se encuentra la fuente imagen está dada por las coordenadas  $[l, m, n]$ ; la del receptor, está dada por  $[x_r, y_r, z_r]$ ; la de la fuente, está dada por  $[x_f, y_f, z_f]$  y las dimensiones del recinto son  $\alpha, \beta, \gamma$  (largo, ancho y alto). La ubicación (coordenadas) de la fuente imagen  $[d_x, d_y, d_z]$  esta dada por las ecuaciones (17), (18) y (19). La distancia entre fuente imagen y receptor, está dada por la ecuación (20).

$$d_x = (\alpha l + (-1)^l x_f) - x_r \quad (17)$$

$$d_y = (\beta m + (-1)^m y_f) - y_r \quad (18)$$

$$d_z = (\gamma n + (-1)^n z_f) - z_r \quad (19)$$

$$d_{l,m,n} = \sqrt{(d_x)^2 + (d_y)^2 + (d_z)^2} \quad (20)$$

Relacionando esta distancia  $d_{l,m,n}$  con la distancia  $d_{xy}$  definida en la sección 3.2.1, se puede decir que es la misma distancia, pero desde la fuente imagen ubicada en la celda imagen (“recinto virtual”, figura 16). Teniendo esta distancia, se puede definir el factor de amplificación debido a la distancia  $F_{amp\_d}$  de la presión sonora medida en la posición del receptor respecto a la fuente imagen en las coordenadas  $[l, m, n]$  (asumiendo que la potencia sonora de la fuente es constante):

$$F_{amp\_d} = \sqrt{\frac{1}{4\pi d_{l,m,n}^2}} \quad (21)$$

Se toma la raíz cuadrada ya que la intensidad es directamente proporcional al cuadrado de la presión sonora; así, es coherente con la atenuación debida a la distancia descrita en la sección 3.1.2. Por otro lado, teniendo en cuenta el coeficiente de absorción  $\alpha_{abs}$  de las superficies<sup>18</sup> y la cantidad de estas, el factor de

<sup>17</sup> Viveiros, E; Gibbs, B. *An Image model for predicting the field performance of acoustic louvres from impulse measurements*. 2003. © 2003 Elsevier Science Ltd. All rights reserved.

<sup>18</sup> En este proyecto, para este algoritmo el coeficiente de absorción funciona más como un indicador de qué tan absorbente/reflejante es el recinto, no tanto como un coeficiente de absorción dependiente de la frecuencia.

amplificación debido al coeficiente de absorción se ve afectado de la siguiente forma:

$$F_{amp\_abs} = \sqrt{(1 - \alpha_{abs})^{N_b}} \quad (22)$$

Siendo  $N_b$  el número de reflexiones especulares:

$$N_b = |l| + |m| + |n| \quad (23)$$

Posteriormente, teniendo en cuenta la absorción del aire, en donde  $m_a$  es la constante de atenuación de energía, el factor  $F_{amp\_air}$  se redefiniría así:

$$F_{amp\_air} = \sqrt{e^{(-m_a \cdot d_{l,m,n})}} \quad (24)$$

Considerando todas las contribuciones de energía de fuentes imagen se puede obtener el factor de amplificación general sobre el receptor, multiplicando todos los factores de atenuación definidos en las ecuaciones (21), (22) y (24):

$$\begin{aligned} F_{amp} &= \sum_{-l}^l \sum_{-m}^m \sum_{-n}^n F_{amp\_d} \cdot F_{amp\_abs} \cdot F_{amp\_air} \\ &= \sum_{-l}^l \sum_{-m}^m \sum_{-n}^n \sqrt{\frac{1}{4\pi d_{l,m,n}^2} \cdot (1 - \alpha_{abs})^{N_b} \cdot e^{(-m_a \cdot d_{l,m,n})}} \end{aligned} \quad (25)$$

En la expresión (25), es importante notar que, para determinar el factor de atenuación total, se requiere de un proceso iterativo<sup>19</sup> (sumatoria), ya que se obtiene un valor de  $F_{amp}$  por cada trayectoria que puede tomar el sonido entre fuente y receptor. Aprovechando esto se calcula la atenuación por cada valor de  $[l, m, n]$  y no el factor de amplificación total. En otras palabras, se obtiene la amplitud de la señal por cada reflexión especular. Por otro lado, al cambiar  $l, m$  o  $n$  se tiene una distancia determinada entre fuente imagen y receptor, por ende, un retraso determinado. Teniendo en cuenta la distancia definida en la ecuación (20), se puede calcular el retraso, usando las ecuaciones (8) y (11). Habiendo definido los elementos fundamentales para llevar a cabo su desarrollo, la implementación de este método puede apreciarse en la sección 4.2.1.

### Ventajas:

<sup>19</sup> Mathworks. "Iteration". <https://www.mathworks.com/content/dam/mathworks/mathworks-dot-com/moler/exm/chapters/iteration.pdf>.

- Permite calcular una respuesta al impulso teórica de un recinto rectangular sin necesidad de realizar una medición.
- Se puede determinar la cantidad de reflexiones deseada en la respuesta al impulso (primer orden, segundo orden, etc.) con precisión.

### 3.5). Reverberación a partir de Red de Retrasos

Este tipo de reverberación es de los primeros que se ha utilizado en los últimos 53 años<sup>20</sup>. Básicamente consiste en generar múltiples copias de la señal, a las que se les aplica un retraso (*Delay*) y se les multiplica por un coeficiente que disminuye su amplitud. Este tipo de configuraciones se puede apreciar fundamentalmente en filtros (*IIR, FIR, Comb filters, allpass filters*, entre otros<sup>21</sup>).

#### 3.5.1). Feedback Delay Network

Este algoritmo de reverberación consiste en sumar retrasos atenuados de una señal de entrada. La configuración básica de este algoritmo (Stautner & Puckette, 1982, [10]) consta de:

- Coeficiente  $d$  que atenúa la señal original.
- Coeficientes  $b_l$  que atenúan los retrasos de la señal de entrada<sup>22</sup>.
- Retrasos  $D_l$  en número entero de muestras, aplicado a los retrasos de la señal.
- Coeficientes  $c_l$  que atenúan los retrasos la señal a la salida.
- Coeficientes  $A_{1x1}, A_{1x2}, A_{1x3}, \dots, A_{4x4}$  que atenúan los retrasos en la realimentación.

Siguiendo el esquema de la figura 17, se calcula el resultado  $x_s$  de cada sumatoria  $s$  ( $s = 1, 2, \dots, l$ ) encerradas en círculo azul:

$$x_s[n] = (b_s \cdot x[n]) + (A_{s,1} \cdot x_1[n - D_1]) + (A_{s,2} \cdot x_2[n - D_2]) + (A_{s,3} \cdot x_3[n - D_3]) + (A_{s,4} \cdot x_4[n - D_4]) \quad (26)$$

Luego, se calcula el resultado de la sumatoria encerrada en círculo marrón:

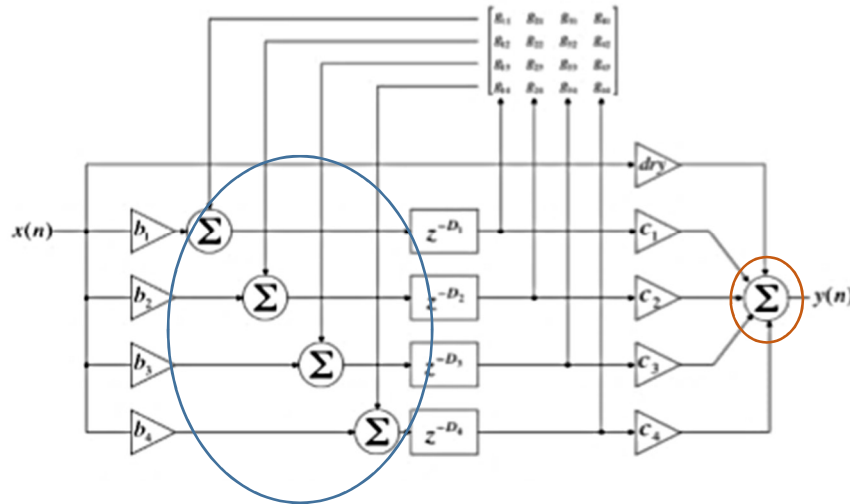
<sup>20</sup> De Sena, E. *Efficient Synthesis of Room Acoustics via Scattering Delay Networks*. 2015. 2329-9290 © 2015 IEEE.

<sup>21</sup> Coggin, J. University of Miami. "Automatic design of Feedback Delay Network Reverb Parameters for perceptual room impulse response matching". 2015.

<sup>22</sup> 1 → Longitud del reverberador FDN.

$$y[n] = c_1 \cdot x_1[n - D_1] + c_2 \cdot x_2[n - D_2] + c_3 \cdot x_3[n - D_3] + c_4 \cdot x_4[n - D_4] + d \cdot x[n] \quad (27)$$

La implementación de este algoritmo se puede apreciar en la sección 4.1.1.



**Figura 17:** Esquema del reverberador *FDN* de longitud 4. Tomado de [10]. Se encerraron en círculo sumatorias de la figura.

### Ventajas:

- Es recursivo y requiere de pocos parámetros para poder utilizarlo y obtener reverberación.
- Se puede hacer más dispersa la matriz para generar más difusión en la reverberación.

### Desventaja:

- No se definen parámetros de un recinto como tal. Para ello, se pueden colocar filtros sobre las líneas de *feedback* y así definir los coeficientes de absorción de las superficies de un recinto determinado. También se podrían calcular los tiempos de retraso de tal forma que las dimensiones del recinto estén implícitas.

### 3.6). Reverberación a partir de red de retrasos y parámetros de un recinto

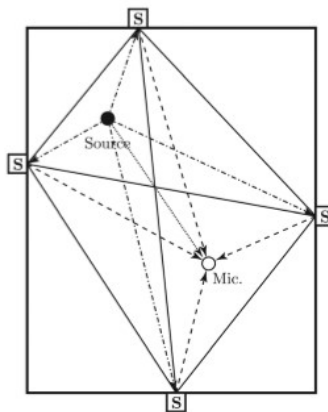
Teniendo en cuenta los parámetros definidos en las secciones 3.1 y 3.2, es posible combinar todo en un solo modelo de reverberación. Al igual que en un reverberador *FDN* (sección 3.5.1), la señal entra directamente a ser procesada (no se obtiene

reverberación por convolución), pero a diferencia de este, se consideran parámetros de un recinto rectangular y se define una trayectoria del sonido dentro de este.

### 3.6.1). Caso de estudio: *Scattering Delay Network*

El objetivo del reverberador *SDN* es simular el comportamiento acústico de un recinto, usando una topología simple. Su funcionamiento general se basa en sintetizar la trayectoria del sonido a lo largo de un recinto rectangular de  $K$  superficies, en puntos específicos: nodos. El sonido parte desde la fuente sonora y se esparce/dispersa (*scatters*) hacia un punto específico de cada superficie (conocido como nodo de dispersión), y hacia el receptor. En resumen, hay 4 tipos de trayectorias (figura 18):

- De fuente sonora a nodos de dispersión
- Entre nodos de dispersión
- De nodos de dispersión a receptor
- De fuente sonora a receptor



**Figura 18:** Definición gráfica del reverberador *SDN*. Los nodos en cada superficie son los bloques S. El nodo del receptor es el círculo blanco (*Mic.*). El nodo de la fuente es el círculo negro (*Source*). Las rayas continuas en negrilla son las trayectorias entre los nodos. Las rayas punteadas discontinuas son las trayectorias entre los nodos en las superficies y el receptor. Las rayas discontinuas son las trayectorias entre la fuente y los nodos en las superficies.

Teniendo en cuenta las trayectorias que sigue el sonido en este tipo de reverberador, se puede empezar a analizar su funcionamiento en términos de procesamiento de señales. Los dos elementos fundamentales de cada trayectoria son el retraso y la atenuación debidos a la distancia. En el primer tipo de trayectoria, el retraso  $m_{xk}$ , considerando la fuente sonora  $x$  ubicada en  $pos_x$  y un nodo  $k$  ubicado en  $pos_k$ , se define (con base en la ecuación (11)):

$$m_{xk} = \text{round}\left(f_s \cdot \frac{\|pos_x - pos_k\|}{c}\right) \quad (28)$$

Y la atenuación  $g_{xk}$ , así (con base en la ecuación (6)):

$$g_{xk} = \frac{1}{\|pos_x - pos_k\|} \quad (29)$$

En el segundo tipo de trayectoria, considerando un segundo nodo  $l$  ubicado en  $pos_l$ , el retraso entre estos dos se considera así:

$$\begin{aligned} m_{lk} = m_{kl} &= \text{round}\left(f_s \cdot \frac{\|pos_l - pos_k\|}{c}\right) \\ &= \text{round}\left(f_s \cdot \frac{\|pos_k - pos_l\|}{c}\right) \end{aligned} \quad (30)$$

Este tipo de trayectoria es bidireccional; es decir, la señal va de un nodo a otro y se devuelve. Por ende, hay realimentación de la señal en cada nodo de dispersión y es importante controlarla para prevenir su exceso, que puede ocasionar inestabilidad en la señal de salida. Esto se realiza fundamentalmente mediante unos coeficientes de atenuación  $S_{K \times K}$ .

Por otro lado, como se mencionó en el primer párrafo de esta sección, cada nodo de dispersión está ubicado en un punto específico de cada superficie. Por lo tanto, se define un coeficiente de absorción  $\alpha$  para cada uno.

Teniendo esto en cuenta, la señal que va del nodo  $l$  al nodo  $k$  se define así:

$$\begin{aligned} nod_{lk}[n] &= S_{l,k} \cdot \alpha_l \cdot nod_l[n - m_{lk}] \\ k &= 1, 2 \dots, K; l = 1, 2 \dots, K \end{aligned} \quad (31)$$

En el tercer tipo de trayectoria, considerando el receptor  $y$  ubicado en  $pos_y$ , el retraso  $m_{ky}$  se define así:

$$m_{ky} = \text{round}\left(f_s \cdot \frac{\|pos_k - pos_y\|}{c}\right) \quad (32)$$

Y la atenuación  $g_{ky}$ , así:



$$g_{ky} = \frac{1}{\|pos_k - pos_y\|} \quad (33)$$

Finalmente, en la cuarta trayectoria, el retraso  $m_{xy}$  se define así:

$$m_{xy} = \text{round}\left(f_s \cdot \frac{\|pos_x - pos_y\|}{c}\right) \quad (34)$$

Y la atenuación  $g_{xy}$  así:

$$g_{xy} = \frac{1}{\|pos_x - pos_y\|} \quad (35)$$

Con base en las ecuaciones (28) a (35), la señal  $nod_k$  que entra al nodo  $k$  se define así:

$$nod_k[n] = \left(\frac{1}{2} \cdot g_{xk} \cdot A_{ten-x}(\theta_{xk}) \cdot x[n - m_{xk}]\right) + \sum_l \alpha_l \cdot nod_{lk}[n] \quad ^{23} \quad (36)$$

$$k = 1,2,3 \dots, K, \quad l = 1,2,3, \dots, K, \quad l \neq k$$

Y la señal total que sale del nodo  $k$  al receptor:

$$y_{nod_k}[n] = A_{ten-y}(\theta_{ky}) \cdot g_{ky} \cdot nod_k[n - m_{ky}], \quad (37)$$

$$k = 1,2,3 \dots, K$$

Finalmente, la señal total en el receptor sería:

$$y[n] = \left(\sum_k y_{nod_k}[n]\right) + (g_{xy} \cdot A_{ten-y}(\theta_{xy}) \cdot x[n - m_{xy}]) \quad (38)$$

$$k = 1,2,3 \dots, K$$

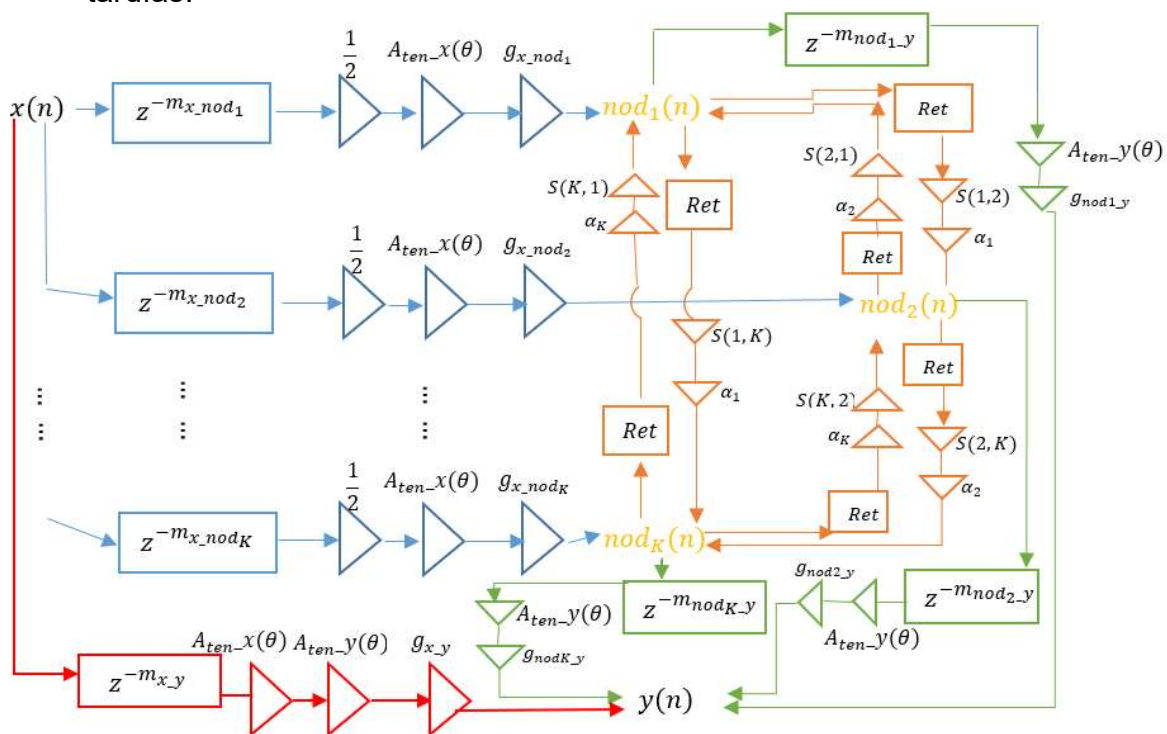
---

<sup>23</sup> Se multiplica por  $\frac{1}{2}$  ya que la trayectoria entre nodos en las superficies es bidireccional.

En el esquema de la figura 19 se puede apreciar la representación en procesamiento de señales del reverberador SDN, y su implementación se puede apreciar en la sección 4.1.3.

### Ventajas:

- La posición de las reflexiones en las superficies está en un punto específico (nodos).
- Los ángulos entre receptor-nodos y entre fuente-nodos siempre son los mismos; esto implica que las reflexiones van al receptor siempre con la misma dirección.
- Se pueden obtener las primeras reflexiones y una aproximación a reflexiones tardías.



**Figura 19:** Representación general del reverberador SDN en procesamiento de señales. Los elementos en color azul son las trayectorias fuente a nodos; los de naranja, las de nodo a nodo (las cajas Ret son los retardos para cada trayectoria entre nodos); los de verde, las de nodo a receptor; y los de rojo, las de fuente a receptor.

## 4). DESARROLLO

Teniendo en cuenta el capítulo anterior, en este capítulo se lleva a cabo el uso de los parámetros vistos en la implementación de los tres algoritmos estudiados: *Feedback Delay Network*, Fuente Imagen y *Scattering Delay Network*. Inicialmente, se abordarán sin espacialización (es decir, monofónicos). Luego se explicará cómo se llevó a cabo la espacialización en cada uno. Finalmente, se seleccionarán dos

algoritmos de estos tres los cuales se evaluarán subjetivamente y sólo uno de estos dos se utilizará para la aplicación artística (objetivo final).

#### 4.1). Implementación de los algoritmos sin espacialización

La implementación se llevó a cabo en *Matlab*<sup>24</sup>. En cada uno se explica la forma en que se abordó la programación con base a lo estudiado en el capítulo 3. Los códigos se encuentran en el capítulo de Anexos.

##### 4.1.1). *Feedback Delay Network*

Con base a la sección 3.5.1, este algoritmo se abordó de la siguiente forma:

- Se establecieron los parámetros necesarios en el reverberador.
  - Coeficientes:
    - $d, b_1, b_2, b_3, b_4, c_1, c_2, c_3, c_4$  y los coeficientes  $A$ , organizados en una matriz  $A_{4 \times 4}$ , todos menores a 1.
  - Retrasos
    - $D_1, D_2, D_3, D_4$ , todos números enteros de muestras.
- Con la ecuación (26) se calcularon las cuatro copias retrasadas de la señal de entrada.
  - Se utilizó un ciclo *for* para controlar la variable  $n$ , que corresponde al número de la muestra (tiempo discreto).
- Luego, se obtuvo la salida  $y$  utilizando la ecuación (27).
  - Se utilizó nuevamente un ciclo *for* para controlar la variable  $n$ .
- Finalmente, se exportó  $y$  como archivo de audio para escuchar el resultado.
  - `audiowrite('Señal_resultante.wav', y, f_s)`, siendo  $f_s$  la frecuencia de muestreo del archivo de audio de entrada.

En la sección A1 (Anexo 1) se puede ver el código correspondiente. Por otro lado, este algoritmo se utilizó en la validación subjetiva (sección 4.4). En cuanto a la implementación:

Los valores de los parámetros fueron (teniendo en cuenta que el orden del reverberador es  $l = 4$ ): (se escogieron arbitrariamente)

---

\*\* *Mathworks. Matlab.* <https://la.mathworks.com/help/matlab/index.html>. © 1994-2018 The MathWorks, Inc. [Consultado el 28/09/18].

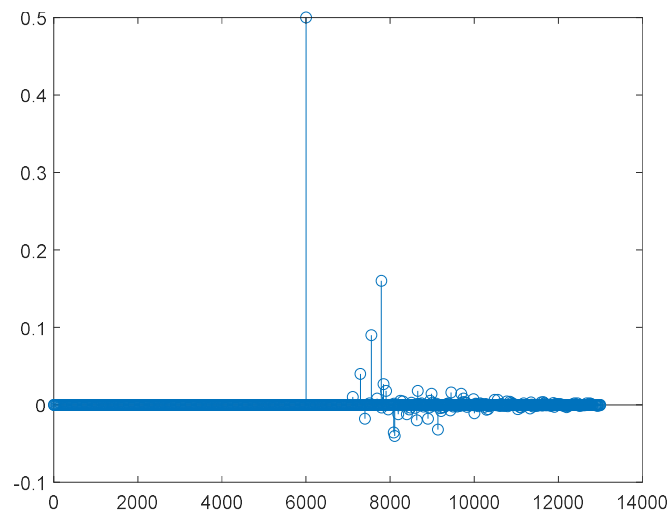
- $d = 0.5$
- $b_1 = 0.1, b_2 = 0.2, b_3 = 0.3, b_4 = 0.4$
- $c_1 = 0.1, c_2 = 0.2, c_3 = 0.3, c_4 = 0.4$
- $D_1 = 1009, D_2 = 591, D_3 = 853, D_4 = 793$

En esta implementación en particular, a través de prueba y error, se encontró que si se escogían valores  $> 0.5$  en la matriz  $A$ , la reverberación a la salida presentaba inestabilidad.

$A =$

0	-0.45	0	0.45
-0.45	0	0.45	0
0	0	-0.45	0.45
0	-0.45	-0.45	0

Como este es un algoritmo de reverberación que procesa directamente la señal, para obtener la respuesta al impulso (figura 20) es necesario que la señal de entrada sea un impulso (señal impulso unitario). Con base a la respuesta al impulso obtenida se puede observar que presenta un comportamiento estable (ya que la energía tiende a ser cero y no  $\pm \infty$ ). El comportamiento de la respuesta al impulso en cuanto a la manera en que decae la energía está relacionado principalmente con los coeficientes de la matriz  $A$ .



**Figura 20:** Respuesta al Impulso obtenida con el reverberador *FDN* programado en *Matlab*.

#### 4.1.2). Fuente imagen

Para llevar a cabo el método fuente imagen, se realizó el siguiente procedimiento:

- Se definieron los parámetros necesarios:

$l, m, n$	$\alpha, \beta, \gamma$	$[x_r, y_r, z_r]$	$[x_f, y_f, z_f]$	$\alpha_{abs}$	$m_a$
$l: -6 \text{ hasta } 6$	$\alpha = 20\text{mts}$	[0,0,0]	[3,2,0]	0.25	0.02
$m: -6 \text{ hasta } 6$	$\beta = 20\text{mts}$				
$n: -6 \text{ hasta } 6$	$\gamma = 20\text{mts}$				

- Retomando la ecuación (25), la forma en que se realizaron sumatorias en este proyecto fue a través de ciclos *for*. En este caso, al ser tres sumatorias, se implementaron tres ciclos *for* anidados (*Nested for loops*)<sup>25</sup>.
  - Un ciclo *for* controla la variable  $l$ ; otro, controla  $m$ ; y otro, controla  $n$ , recordando que estas variables determinan la posición de la celda imagen (o “recinto virtual”) en donde se encuentra la fuente imagen<sup>26</sup>.
  - Por cada valor de  $l, m, n$  se realizan las operaciones de las ecuaciones (17) a (25).
  - Teniendo el factor de atenuación  $F_{amp}$  calculado con la ecuación (25), se asigna a una posición determinada del vector respuesta al impulso  $h$ , que depende del tiempo de retraso en muestras ( $nmuestra$ ) debido a la distancia  $d_{l,m,n}$  calculada con la ecuación (20).
    - $nmuestra$  se calcula con la ecuación (11).
    - Luego de calcular  $nmuestra$ , se realiza la siguiente asignación:  

$$h(nmuestra) = F_{amp}^{27}$$
- Así, se obtuvo muestra a muestra la respuesta al impulso  $h$  (figura 21).
- Luego de haber calculado  $h$ , con base en la distancia fuente-receptor <sup>28</sup> se añadió la atenuación debida a esta (sección 3.1.2).
  - Se calculo el valor de atenuación  $aten_a$  con la ecuación (6).
- Posteriormente, se multiplicó el primer impulso de la respuesta al impulso (que corresponde a la intensidad del sonido directo en el receptor) por este valor para atenuarlo. Esto se hizo con el fin de dar la noción de profundidad del sonido directo dentro del recinto.

<sup>25</sup> Mathworks. Matlab. “Loop Control Statements”. © 1994-2018 The MathWorks, Inc. < [https://la.mathworks.com/help/matlab/matlab\\_prog/loop-control-statements.html](https://la.mathworks.com/help/matlab/matlab_prog/loop-control-statements.html) >.

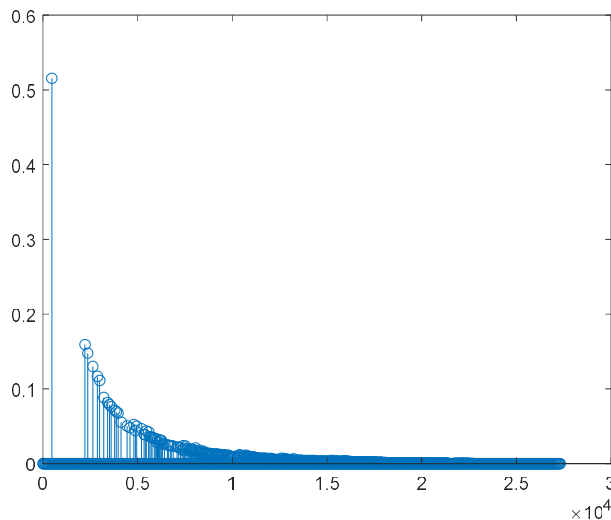
<sup>26</sup> En términos de trayectoria, determinan el orden de la reflexión (1er, 2do, 3er, ... orden).

<sup>27</sup> Dado el caso en que el valor de  $nmuestra$  sea el mismo para dos o más combinaciones de los valores de  $l, m$  o  $n$ , en el algoritmo se suman los valores de  $F_{amp}$  para esa muestra.

<sup>28</sup> Se calculó la distancia como en el ejemplo de la sección 3.1.2, página 14.

- $h(h == \max^{29}(h))^{30} = h(h == \max(h)) * atenu_d$
- Para obtener la salida  $y$ , se realizó convolución entre la señal de entrada  $x$  (cualquier archivo de audio monofónico con frecuencia de muestreo  $F_s$ ) y la respuesta al impulso obtenida  $h$ .
  - $y = conv(x, h)$
- Finalmente, se exportó la señal resultante.
  - `audiowrite('Señal_resultante.wav', y, F_s)`

Habiendo mencionado cada elemento y la manera en que se abordó el método, el código para obtener la respuesta al impulso mediante el método Fuente imagen se puede revisar en la sección A.2 (Anexo 2). En cuanto a la respuesta al impulso obtenida en la implementación<sup>31</sup> (figura 21), se puede notar que la energía decae a medida que avanza el tiempo de forma estable<sup>32</sup>. Esto está relacionado con el orden (número de rebotes en las superficies) de las reflexiones; a medida que el orden crece, la energía del sonido decae cada vez menos.



**Figura 21:** Respuesta al Impulso obtenida mediante el método fuente imagen de Gibbs y Viveiros, programado en *Matlab*.

### 4.1.3). *Scattering Delay Network*

<sup>29</sup> Mathworks. “max”. © 1994-2018 The MathWorks, Inc. <  
<https://la.mathworks.com/help/matlab/ref/max.html>>. [Consultado el 9-11-18]

<sup>30</sup> Mathworks. “Array Indexing”. © 1994-2018 The MathWorks, Inc. Revisar *logical indexing*  
 < [https://la.mathworks.com/help/matlab/math/array-indexing.html?searchHighlight=logical%20indexing&s\\_tid=doc\\_srchtile](https://la.mathworks.com/help/matlab/math/array-indexing.html?searchHighlight=logical%20indexing&s_tid=doc_srchtile)>. [Consultado el 9-11-18]

<sup>31</sup> El valor  $F_{amp}$  calcula una raíz cuadrada; por ende, los valores siempre dan positivos, en el caso de esta implementación en particular.

<sup>32</sup> En este caso, el primer impulso (intensidad del sonido directo) está atenuado ya que se utilizó la atenuación debida a la distancia.

Acorde con la sección 3.6.1 y el esquema que la simplifica en la figura 19, se realizó el siguiente procedimiento:

Se definieron los parámetros iniciales del recinto: cantidad de superficies ( $K = 6$ ), coeficiente de absorción de cada superficie ( $\alpha_1 = 0.15, \alpha_2 = 0.25, \alpha_3 = 0.15, \alpha_4 = 0.15, \alpha_5 = 0.05, \alpha_6 = 0.05$ ) y matriz de coeficientes:

$S_{6 \times 6} =$

0	-0.1	0	0.1	-0.1	0
-0.1	0	0.1	0	0	0.1
0	0	-0.1	0.1	0	-0.1
0	-0.1	-0.1	0	0	0.1
0.1	0	0.1	0	0	-0.1
0	0.1	0	0	-0.1	0

Luego se definió la ubicación de cada elemento: fuente  $[2 - 3 \ 0.2]$ , receptor  $[-10, 7, -15]$ , nodos:  $nod_1: [0, -12, 0]$ ,  $nod_2: [-12, 0, 0]$ ,  $nod_3: [2, 12, 0]$ ,  $nod_4: [12, 0, 0]$ ,  $nod_5: [0, 0, 12]$  y  $nod_6: [0, 0, -12]$ . Con base a la ubicación de cada elemento, se calcularon:

- Atenuaciones entre fuente sonora y nodos (ecuación (29)); entre nodos y receptor (ecuación (33)); y fuente sonora y receptor (ecuación (35)).
- Retrasos entre nodos (ecuación (30)); entre fuente sonora y nodos (ecuación (28)) entre nodos y receptor (ecuación (32)); y entre fuente sonora y receptor (ecuación (34)).

Posteriormente, se definió un patrón de directividad<sup>33</sup> para la fuente sonora ( $x$ ) y uno para el receptor ( $y$ ) (tabla 1), con base a dos *datasheets*: el de un *speaker JBL control 30*<sup>34</sup> para la fuente y el de un micrófono *Shure SM 81*<sup>35</sup> para el receptor. En cada uno se analizaron los diagramas de directividad por octava.

A manera de simplificación:

- Se consideraron los siguientes ángulos  $\theta$  horizontales:  $0^\circ, 30^\circ, 60^\circ, 90^\circ, 120^\circ, 150^\circ, 180^\circ, 210^\circ, 240^\circ, 270^\circ, 300^\circ, 330^\circ$  y  $360^\circ$ .
  - Para los valores intermedios, se utilizaron condicionales (*if-else*) con el fin de aproximar al valor más cercano.

<sup>33</sup> Esto se incorporó en este algoritmo a diferencia de los anteriores 2 algoritmos, ya que en este se asume que las reflexiones vienen de 6 puntos específicos a lo largo del recinto. Se consideró más eficiente aplicarlo en este algoritmo que en los otros dos por cuestiones de simplificación.

<sup>34</sup> *JBL. Control 30. © Copyright 2007 JBL Professional.*

[http://www.jblpro.com/pages/install/Control\\_30/Ctrl30.pdf](http://www.jblpro.com/pages/install/Control_30/Ctrl30.pdf). [Consultado el 29-10-18]

<sup>35</sup> *Shure. Model SM81 Microphone.*

<https://www.shure.co.uk/productdocumentsfiles/default/products/microphones/sm81/sm81specsheel--english-a5813778fcf4d2276857c2e90f4a15da.pdf>. [Consultado el 29-10-18]

- Por cada uno se calculó una atenuación general en vez de una atenuación para cada frecuencia. Esta se obtuvo con un promedio de los valores de atenuación en  $dB SPL$ <sup>36</sup> de cada una.
  - Para cada valor de atenuación en  $dB SPL$ , se obtuvo  $A_{ten}(\theta)$  con la ecuación (3).

	$\theta = 0 - 15^\circ,$ $345 - 360^\circ$	$\theta = 15 - 30^\circ,$ $330 - 345^\circ$	$\theta = 30 - 45^\circ,$ $315 - 330^\circ$	$\theta = 45 - 60^\circ,$ $300 - 315^\circ$	$\theta = 60 - 75^\circ,$ $270 - 300^\circ$
$SPL_A(\theta)_x$	-0.125	-1.083		-5.735	
$SPL_A(\theta)_y$	0	-0.575		-3.052	
$A_{ten-x}(\theta)$	0.9857	0.8828		0.5167	
$A_{ten-y}(\theta)$	1	0.9359		0.7037	

	$\theta = 75 - 90^\circ,$ $255 - 270^\circ$	$\theta = 90 - 105^\circ,$ $240 - 255^\circ$	$\theta = 105 - 120^\circ,$ $225 - 240^\circ$	$\theta = 120 - 135^\circ,$ $210 - 225^\circ$
$SPL_A(\theta)_x$	-11.61		-15.07	
$SPL_A(\theta)_y$	-6.66		-12.20	
$A_{ten-x}(\theta)$	0.2627		0.1764	
$A_{ten-y}(\theta)$	0.4645		0.2455	

	$\theta = 135 - 150^\circ,$ $195 - 210^\circ$	$\theta = 150 - 165^\circ,$ $180 - 195^\circ$	$\theta = 165 - 180^\circ$
$SPL_A(\theta)_x$	-16.12		-15.2
$SPL_A(\theta)_y$	-16.92		-16.75
$A_{ten-x}(\theta)$	0.1563		0.1738
$A_{ten-y}(\theta)$	0.1426		0.1454

**Tabla 1:** Patrones de directividad (de fuente sonora ( $x$ ) y receptor ( $y$ )) definidos en la implementación del algoritmo de reverberación *SDN*.

Luego, se calculó el ángulo entre fuente y nodos; entre receptor y nodos; entre fuente y receptor. Para ello, fue necesario definir que tanto el frente de la fuente como el del receptor están apuntando a un punto específico. Se definió que el frente de la fuente estaba apuntando al nodo 1 ( $[0.2, -10]$ ), y el del receptor, al nodo 4 ( $[10, 0.2]$ ). Habiendo definido estos puntos, se utilizó una ecuación<sup>37</sup> para calcular el ángulo entre tres puntos:

<sup>36</sup> Recordando en la sección 3.1.1, este valor de atenuación corresponde a  $SPL_A(\theta)$ , definido en la ecuación (1) de dicha sección.

<sup>37</sup> Teniendo los puntos  $[x_1, y_1], [x_2, y_2], [x_3, y_3]$ , el ángulo entre estos puntos se obtuvo así:

$$x_{12} = x_1 - x_2; y_{12} = y_1 - y_2;$$

$$x_{32} = x_3 - x_2; y_{32} = y_3 - y_2;$$

$$\text{angulo} = \text{mod}(\text{atan2}(x_{12} * y_{32} - x_{32} * y_{12}, x_{12} * x_{32} + y_{12} * y_{32}), 2 * \pi) * (180/\pi)$$

Mathworks. Matlab Answers. "calculating angle between three points". 2017. © 1994-2018 The MathWorks, Inc. <<https://la.mathworks.com/matlabcentral/answers/331017-calculating-angle-between-three-points>>. [Consultado el 31-10-18].

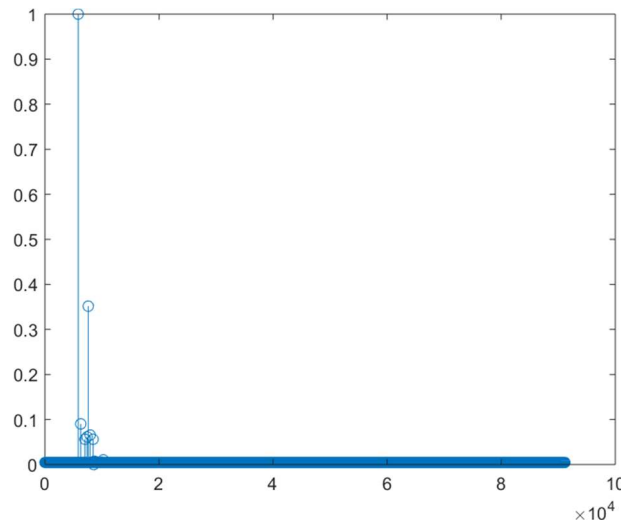


- El punto al que apunta la fuente/el receptor ( $[x_1, y_1]$ )
- El punto en donde está ubicada(o) la fuente/el receptor ( $[x_2, y_2]$ )
- El punto respecto al cual se desea calcular el ángulo ( $[x_3, y_3]$ )

Posteriormente, se procedió a realizar la programación con base al esquema de la figura 19:

- Se utilizó un ciclo *for* general, ya que los cálculos que se realizan son muestra a muestra. Con este ciclo se controla el valor de  $n$ . (muestra).
- En este ciclo, se realizan tres grupos de operaciones:
  - Cálculo de la señal de entrada a los 6 nodos
    - Usando la ecuación (36)
  - Cálculo de la señal que sale de los 6 nodos al receptor
    - Usando la ecuación (37).
  - Con la ecuación (38), se obtiene la salida  $y$ .
- Finalmente, se exporta la señal de salida.
- `audiowrite('Señal_resultante.wav', y, Fs)`

Al igual que en el algoritmo *FDN*, para obtener la respuesta al impulso (Figura 22) es necesario que un impulso sea la señal de entrada.



**Figura 22:** Respuesta al Impulso obtenida con el reverberador *SDN* programado en *Matlab*.

En la implementación se presentó un inconveniente. El sonido de la reverberación posee efecto de coloración (inestabilidad). Para evitarlo, en las fuentes bibliográficas que se consultaron de este algoritmo (hay muy pocas) se utilizan técnicas de control de sistemas que no son explicadas en detalle; por ello y por cuestiones de tiempo no se implementaron.

## 4.2). Implementación de los algoritmos con espacialización

En esta sección se explica cómo se abordó la espacialización en los tres algoritmos estudiados (*Feedback Delay Network*, *Scattering Delay Network*, Fuente Imagen). Acorde con las ventajas (mencionadas en cada sección correspondiente) y la implementación de cada uno, la forma en que se incorporó la espacialización es diferente en cada caso.

### 4.2.1). *Feedback Delay Network*

Para lograr la espacialización en este algoritmo, se implementó únicamente el parámetro psicoacústico *IID* (*Interaural Intensity Difference*), que básicamente hace referencia a una diferencia de nivel entre la señal que sale por el lado  $l$  y la que sale por el lado  $r$  (en otras palabras, paneo).

- Una vez obtenida la salida del reverberador, se duplicó, y una de las copias multiplicó por un coeficiente  $c$  menor a 1; es decir, se atenuó.
  - $y_l = y$ ,  $y_r = y \cdot c$ 
    - En este caso específico, la señal del lado derecho fue la que se atenuó.
- Luego, se juntaron las señales a una sola señal estéreo.
  - $y_{out} = [y_l \ y_r]$
- Finalmente, se exportó el archivo de audio de la salida  $y_{out}$  con frecuencia de muestreo  $F_s$  (la misma del archivo de entrada).
  - `audiowrite('Señal_resultante.wav',  $y_{out}$ ,  $F_s$ )`

### 4.2.2). *Scattering Delay Network*

De acuerdo con las ventajas del algoritmo mencionadas en la sección 3.1.6, luego de haber realizado todo el procedimiento explicado en dicha sección, se añadió la espacialización.

Inicialmente, se encontró un archivo en la página del laboratorio de la *CIPIC* (*Center for Image Processing and Integrated Computing*)<sup>38</sup> que contiene una base de datos de *HRFT*. Esta contiene, entre otros archivos, 2500 medidas de *HRIR* (*Head Related Impulse Response*) para 45 sujetos (cabezas) diferentes. Cada medida se realizó en 25 ángulos azimut (-80, -65, -55, -45 :5: 45 55 65 80 <sup>39</sup>) y en 50 ángulos de elevación (-45 + 5.625\*(0:49) <sup>40</sup>). Luego de haber descargado este archivo, se

---

<sup>38</sup> Ralph, V. *CIPIC Interface Laboratory*. "The CIPIC HRTF Database". 2001. Copyright (c) 2001 The Regents of the University of California. All Rights Reserved. < <https://www.ece.ucdavis.edu/cipic/spatial-sound/hrtf-data/>>. [Consultado el 31-10-18]

<sup>39</sup> El signo ':' indica en este caso que a partir de los -45°, los ángulos van de 5 en 5 hasta los 45°.

<sup>40</sup> Los ángulos van desde -45° hasta 230.625°, en pasos de 5.625°.

cargó en el código de *Matlab*. Como la base de datos contiene medidas para 45 cabezas, fue necesario escoger las medidas para un solo sujeto; teniendo esto en cuenta, se escogió arbitrariamente el sujeto número 9.

Luego se calcularon los ángulos tanto azimut como de elevación (utilizando la función para calcular los ángulos en el algoritmo (explicada en la sección 4.1.3)), entre nodos y receptor; y entre fuente y receptor. En total, fueron 14 ángulos calculados (7 azimut (*az*) y 7 de elevación (*elev*)). Con estos ángulos, se procedió a extraer las *HRIR* de la base de datos<sup>41</sup>.

Teniendo las 14 *HRIR* (nombradas  $h_{lado_{elem}}$ <sup>42</sup>) se utilizó la convolución para obtener la espacialización de la señal proveniente de los 6 nodos ( $nod_k, k = 1,2,3 \dots, 6$ ) y de la fuente sonora ( $x$ ). A manera de ejemplo, teniendo los ángulos azimut  $az_{nod_1}$  y de elevación  $elev_{nod}$  a los que se encuentra el nodo 1  $nod_1$  respecto al receptor  $y$ , para espacializar las reflexiones que provienen de dicho nodo, se realizó el siguiente procedimiento:

- $y_{l_{nod_1}} = conv(nod_1, h_{l_{nod_1}})$
- $y_{r_{nod_1}} = conv(nod_1, h_{r_{nod_1}})$

Teniendo todas las señales espacializadas tanto para el lado  $l$  como para el lado  $r$ , se sumaron por aparte:

- $y_{out\_l} = y_{l_{nod_1}} + y_{l_{nod_2}} + \dots + y_{l_{nod_6}} + y_{l_x}$
- $y_{out\_r} = y_{r_{nod_1}} + y_{r_{nod_2}} + \dots + y_{r_{nod_6}} + y_{r_x}$

Posteriormente, se juntaron ambas señales  $y_{out\_l}, y_{out\_r}$  a una sola señal estéreo  $y_{out}$ , que luego fue exportada.

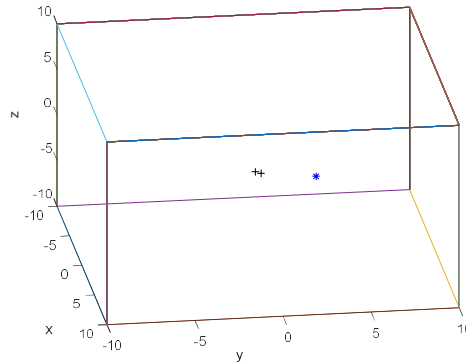
- $y_{out} = [y_{out\_l} \ y_{out\_r}]$ .
- $audiowrite('Señal\_resultante.wav', y_{out}, F_s)$

### 4.2.3). Fuente Imagen

Teniendo en cuenta la sección 4.1.2, con las mismas variables y procedimiento, se incorporó la espacialización, bajo la premisa de que ahora son dos puntos de referencia (dos “oídos”, figura 23).

<sup>41</sup> CIPIC Interface Laboratory. “Documentation for the UCD HRIR Files”. 1998.

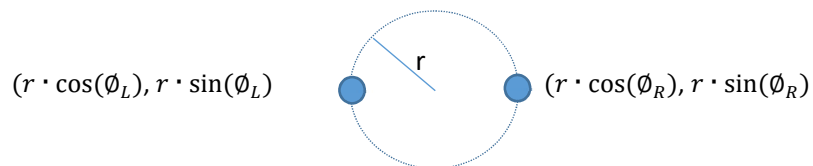
<sup>42</sup> La palabra *lado* hace referencia a  $l$  o  $r$ . La palabra *elem* hace referencia a alguno de los 6 nodos ( $nod_1, nod_2 \dots, nod_6$ ) o a la fuente sonora ( $x$ ).



**Figura 23:** Fuente (\*) y receptor (2 oídos, +) ubicados dentro del recinto simulado.

En otras palabras, se hace exactamente lo mismo que en la sección 4.1.2, pero ahora, con dos puntos de receptor a cambio de uno. Esto implica que se calculan 2 respuestas al impulso ( $h_l, h_r$ ), una para cada posición de oído. En cuanto a la posición de cada oído, se asumió que la cabeza es circular; por lo tanto, los oídos están ubicados por medio de coordenadas polares interaurales<sup>43</sup> (Figura 24).

- Cada oído se ubica en un ángulo determinado de la circunferencia. Las coordenadas se expresan mediante las funciones coseno (*cos*) y seno (*sin*).
  - Por ejemplo, si se quiere que el centro de la cabeza del receptor corresponda con el centro del recinto, el ángulo para el oído izquierdo sería  $\pi$ ; y el del derecho,  $2\pi$ . Dependiendo cual sea el radio  $r$  de la cabeza, las coordenadas (en  $[x, y]$ ), serían  $[r \cos \pi, r \sin \pi]$  para el oído izquierdo, que equivale a  $[-r, 0]$ ; y  $[r \cos 2\pi, r \sin 2\pi]$  para el derecho, equivalente a  $[r, 0]$ .



**Figura 24:** Ubicación del receptor (2 oídos) en el método fuente imagen con espacialización.  $r$  es el radio de la cabeza,  $\phi_L$  es el ángulo de ubicación del oído izquierdo; y  $\phi_R$ , el del oído derecho.

- Esta diferencia de posición va a generar una diferencia en las respuestas al impulso de cada uno; tanto en amplitud como en tiempo.

Teniendo ambas respuestas al impulso ( $h_l, h_r$ ), se realizó convolución entre la señal original ( $x$ ) y cada una de estas. Se asignó cada una a un lado del estéreo.

<sup>43</sup>Segura, M; Ramirez, N; Moreno, E. “Sonido Virtual o cómo mover una fuente monofónica en el espacio”. <https://upcommons.upc.edu/bitstream/handle/2099/9973/Article008.pdf?sequence=1&isAllowed=y>. [Consultado el 1-11-18].

- $y_l = conv(x, h_l), y_r = conv(x, h_r)$
- $y_{out} = [y_l \ y_r]$

Finalmente, se exportó la salida.

- `audiowrite('Señal_resultante.wav', y_out, F_s)`

A manera de verificación del algoritmo, se realizó una estimación de la respuesta al impulso dentro de un recinto real (estudio, imagen 1, sección 4.3.1), con el fin de comparar las respuestas al impulso programada y estimada. Para ello, dentro del recinto, se emitió ruido rosa a través de un parlante<sup>44</sup> durante 3 minutos seguidos, y se capturó con dos micrófonos de condensador<sup>45</sup>. El parlante se ubicó a dos metros de los micrófonos, todos a la misma altura (1.5m). Con base en la ecuación (21) se estableció la siguiente ecuación:

$$H(\omega) = \frac{Y(\omega)}{X(\omega)}$$

En donde  $\omega$  es la frecuencia,  $X(\omega)$  es la señal original,  $Y(\omega)$  es la señal con reverberación y  $H(\omega)$  es la respuesta al impulso, todas en dominio de frecuencia<sup>46</sup>. El proceso de estimación está basado en el método *ETFE (Empirical Transfer Function Estimate)*<sup>47</sup>. Teniendo la señal de entrada (señal emitida) y la señal con reverberación (señal capturada por los micrófonos), usando esta ecuación se puede obtener la respuesta al impulso estimada. En la figura 25 se pueden apreciar tanto las respuestas al impulso estimadas, como las respuestas al impulso simuladas para cada posición del receptor.

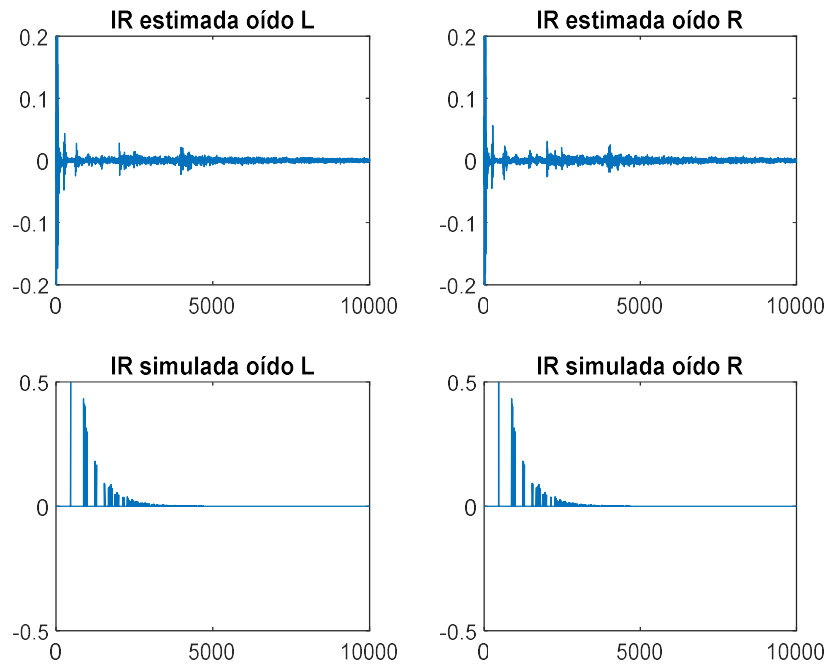
---

<sup>44</sup> *Electro-voice. ETX-12P*. © Bosch Sicherheitssysteme GmbH, 2018. < <https://www.electrovoice.com/product.php?id=2592>>. [Consultado el 19/10/18]

<sup>45</sup> *AKG. C430. Specs*. © 2018 Harman International Industries, Incorporated. All rights reserved. <https://www.ake.com/Microphones/Condenser%20Microphones/C430.html>. [Consultado el 19/10/18].

<sup>46</sup> *Mathworks. Matlab. "Practical Introduction to Frequency-Domain Analysis"*. © 1994-2018 The MathWorks, Inc. < <https://la.mathworks.com/help/signal/examples/practical-introduction-to-frequency-domain-analysis.html>>. [Consultado el 7 de octubre de 2018].

<sup>47</sup> Ljung, L. "System Identification: Theory for the user" (página 146). © 1987 by P T R Prentice Hall.



**Figura 25:** Respuestas al impulso estimadas y simuladas de uno de los recintos escogidos (estudio) con respecto a las dos posiciones de receptor.

De la gráfica anterior se puede destacar que tanto las respuestas estimadas como las simuladas describen el comportamiento decreciente de la energía en el recinto. Como la estimación se realizó para el receptor ubicado en el centro del mismo, las respuestas al impulso para ambos oídos son iguales. Por otro lado, es notable que la intensidad de las reflexiones en las respuestas estimadas es menor que la de las simuladas. También se observa que el signo de la intensidad de las reflexiones es completamente positivo en las respuestas simuladas.

Con base a la diferencia de signo entre las respuestas estimadas y simuladas, se optó por procesar las respuestas simuladas, con el fin de obtener un comportamiento similar al de las respuestas estimadas. Para ello, se introdujo cada respuesta al impulso simulada en el reverberador *FDN*, con los siguientes valores:

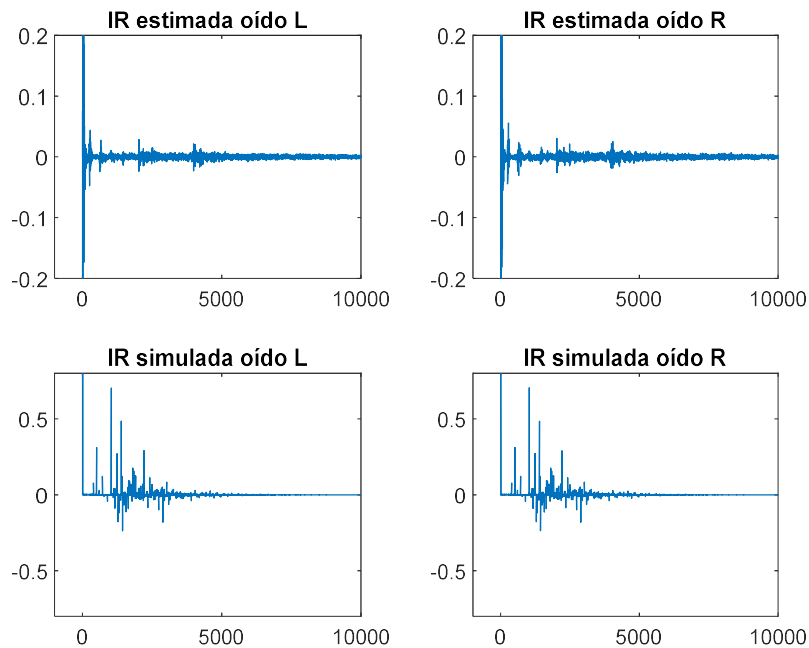
- $b = c = [0.1 \ 0.2 \ 0.3 \ 0.4]$
- $D = [133 \ 251 \ 773 \ 927]$
- $A =$

0	-0.38	0	-0.38
0.38	0	-0.38	-0.38
-0.38	0	0	-0.38
0	-0,38	0.38	0

Luego de haber realizado este proceso, se compararon nuevamente las respuestas simuladas con las estimadas. Se puede ver en la figura 26 que el comportamiento

de las reflexiones en las respuestas simuladas cambió considerablemente. Comparando las respuestas simuladas de la figura 25 con las de la figura 26, se aprecia que la longitud (duración) de las respuestas procesadas por el reverberador *FDN* es mayor. Esto está relacionado con los tiempos de retraso (en muestras)  $D$ . Al haber utilizado en su mayoría valores negativos en la matriz  $A$ , la intensidad de las reflexiones disminuyó con respecto a su valor original, y se puede ver algunas reflexiones cuya intensidad tiene signo negativo.

Con base a lo mencionado en el párrafo anterior, es importante resaltar que el reverberador *FDN* puede ser utilizado también como un algoritmo de difusión con el fin de dar complejidad a la reverberación. A pesar de que no son exactamente iguales las respuestas estimadas y simuladas en la figura 26, su comportamiento es similar. Este procedimiento se realizó únicamente para esta etapa de comparación entre la simulación y la estimación de la respuesta al impulso del recinto. No se implementó en las demás secciones del proyecto.



**Figura 26:** Respuestas al impulso estimadas (arriba) y simuladas (abajo) del estudio para las dos posiciones del receptor. Las respuestas al impulso simuladas fueron procesadas por el algoritmo *FDN*.

### 4.3). Validación subjetiva

Como se menciona en los objetivos, parte del proyecto consiste en validar el algoritmo acorde con la percepción auditiva de las personas. La manera en que se realizó esta actividad está basada en una adaptación de la prueba (test) *MUSHRA* (*Multi Stimulus test with Hidden Reference and Anchor*), la cual es adecuada para evaluar la calidad sonora de archivos de audio, cuyos resultados tienden a ser

precisos y confiables<sup>48</sup>. Esta prueba consiste en comparar un audio de referencia de alta calidad, con audios que posean pequeñas diferencias respecto a esta. Los audios a evaluar son calificados (entre 0 y 100) con respecto al de referencia; es decir, qué tan parecido suenan con base en algún parámetro determinado.

#### 4.3.1). Procedimiento previo

Se estableció la metodología de la prueba. Como se mencionó en el párrafo anterior, se realizó una adaptación de la prueba *MUSHRA*. Esta adaptación básicamente consiste en comparar una señal grabada dentro de un recinto (audio de referencia) con la misma señal, procesada por dos algoritmos diferentes<sup>49</sup> (audios a evaluar). La prueba se divide en 4 etapas (tabla 2), las cuales son partes en las que se dividió la prueba con el fin de evaluar los siguientes tres parámetros, cada uno para dos situaciones distintas:

- Recinto
  - Estudio (Imagen 1).
    - Dimensiones:  $6.58m \times 15.54m \times 4.88m$
  - Cuarto
    - Dimensiones:  $3.03m \times 4.35m \times 2.44m$
- Posición de la fuente.
- *Pre-Delay* o *ITDG (Initial Time Delay Gap)*<sup>50</sup>
  - Profundidad. Intimidad del sonido.
    - Está relacionada con la distancia fuente a receptor dentro del recinto.

Cabe aclarar que los parámetros en cada algoritmo se definieron con base en las características de cada recinto. Una de las limitaciones era que no se sabía cual era el coeficiente de absorción de los materiales en las superficies en los recintos. Lo que si se sabía era que el tiempo de reverberación del estudio era más largo que el del cuarto. Otro factor que sí se sabía era que el estudio era más grande que el cuarto, lo cual influye en el tiempo que tardan las reflexiones en llegar al receptor. Con base en lo mencionado, se definió lo siguiente en cada algoritmo:

Fuente Imagen:

- $\alpha = 0.5$  para el estudio,  $\alpha = 0.75$  para el cuarto
  - Los demás parámetros (posición fuente y receptor, dimensiones del recinto) se definieron tal cual como aparecen en la tabla 2.

---

<sup>48</sup> ITU Radiocommunication Assembly. "Method for the subjective assessment of intermediate quality level of coding systems". 2001-2003.

<sup>49</sup> Algoritmos Fuente imagen y *FDN*.

<sup>50</sup> Tiempo transcurrido entre la llegada del sonido directo y la llegada de la primera reflexión al receptor.



*FDN:*

- $b = [0.1 \ 0.2 \ 0.3 \ 0.4]$  para ambos recintos
- $c = [0.1 \ 0.2 \ 0.3 \ 0.4]$  para el estudio,  $c = [0.2 \ 0.3 \ 0.4 \ 0.5]$  para el cuarto
- $D = [137 \ 249 \ 353 \ 593]$  para el estudio,  $D = [93 \ 149 \ 73 \ 47]$  para el cuarto.
- $A =$  (para el estudio)

0	-0.45	0	0.45
-0.45	0	0.45	0
0	0	-0.45	0.45
0	-0.45	-0.45	0

- $A =$  (para el cuarto)

0	-0.15	0	0.15
-0.15	0	0.15	0
0	0	-0.15	0.15
0	-0.15	-0.15	0

	<b>Etapas 1</b>	<b>Etapas 2</b>	<b>Etapas 3</b>	<b>Etapas 4</b>
Tipo de recinto	Estudio	Cuarto	Estudio	Cuarto
Posición fuente <sup>51</sup> [ $x, y, z$ ]	[0, 2, -0.97]	[0, 2, -0.05]	[1, 2, -0.97]	[-0.60, 2, -0.05]
Posición receptor [ $x, y, z$ ]	$l$ [-0.10, 0, -0.94] $r$ [0.10, 0, -0.94]	$l$ [-0.10, 0, -0.05] $r$ [0.10, 0, -0.05]	$l$ [-0.10, 0, -0.94] $r$ [0.10, 0, -0.94]	$l$ [-0.10, 0, -0.05] $r$ [0.10, 0, -0.05]

**Tabla 2:** Etapas de la prueba subjetiva de acuerdo a los parámetros a evaluar.

<sup>51</sup> Teniendo en cuenta que las coordenadas [0,0,0] corresponden al centro del recinto.



**Imagen 1:** Foto de uno de los recintos escogidos (estudio) con los equipos utilizados para la grabación de los audios de referencia de la validación subjetiva.

Es importante mencionar que al momento de grabar los audios de referencia se tuvieron muy en cuenta las etapas de salida y entrada de la señal. En cuanto a los niveles de salida (hacia el parlante) y de grabación (micrófonos) se calibraron en el dispositivo (interfaz<sup>52</sup>) utilizado. El nivel de salida del parlante se ajustó en -12 dBFS; la salida de la interfaz hacia el parlante (salida 1 (*Main l*), nivel de línea) fue de +5 dBu RMS aproximadamente. La ganancia de los micrófonos se ajustó en 30 dB (entradas 1 y 2). Como el archivo de la señal de referencia es *stereo*, se utilizó solo el canal *l* de este (solo se utilizó un parlante). Esto también se tuvo en cuenta en el algoritmo.

A manera de resumen, se tienen 4 etapas que contienen diferentes valores de los parámetros a evaluar. Se tiene un archivo de audio de referencia y dos archivos de audio a evaluar, con un total de 3 archivos de audio por etapa. La duración de todos los archivos es de 20 segundos. Teniendo listos y organizados los archivos, se creó una sesión en *ProTools* (figura 27), la cual se utilizó para llevar a cabo la prueba. Los dos clips en cada *playlist* de cada canal corresponden a la señal procesada por los algoritmos. Los clips de audio principales de cada canal corresponden a las señales de referencia (grabadas).

---

<sup>52</sup> *Native Instruments. Komplete Audio 6. Especificaciones. © 2018 Native Instruments GmbH.*  
<https://www.native-instruments.com/es/products/komplete/audio-interfaces/komplete-audio-6/specifications/>.  
[Consultado el 19 de Octubre de 2018].

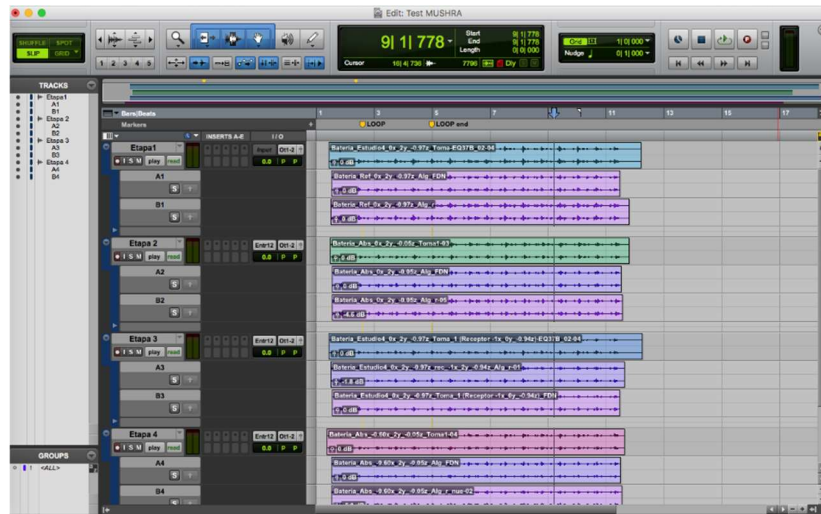


Figura 27: Sesión de ProTools en la que se realizó la prueba de validación subjetiva.

#### 4.3.2). Proceso de evaluación

Siguiendo las recomendaciones de la *ITU* para llevar a cabo esta prueba adecuadamente, es importante que las personas que vayan a realizarla tengan audición crítica. Otra de las recomendaciones es tener muy claro el diseño de la prueba, en cuanto a evitar malentendidos en los puntos de comparación. También en cuanto a que, por cada par de audios que se evalúen, haya una adecuada aleatorización del orden en que el evaluador vaya a escuchar. Al seguir estas (entre otras) recomendaciones se evita la incorrecta calificación.

Las calificaciones se consignaron en un archivo de Excel (tabla 3) que cada evaluador llenaba a medida que escuchaba los audios de la prueba. En la prueba, cada uno calificaba ambos algoritmos<sup>53</sup> en los tres parámetros definidos para las cuatro etapas mencionadas en la sección 4.3.1.

Etapas 1	¿Está en el mismo recinto?	
	<b>A</b>	<b>B</b>
Calificación		
	Posición de la fuente	
	<b>A</b>	<b>B</b>
Calificación		
	Cercanía de la fuente	
	<b>A</b>	<b>B</b>
Calificación		

Etapas 2	¿Está en el mismo recinto?	
	<b>A</b>	<b>B</b>
Calificación		
	Posición de la fuente	
	<b>A</b>	<b>B</b>
Calificación		
	Cercanía de la fuente	
	<b>A</b>	<b>B</b>
Calificación		

<sup>53</sup> En las etapas 1, 2 y 4, el algoritmo fuente imagen correspondía a la columna B, mientras que el algoritmo *FDN*, correspondía a la columna A.

Etapa 3	¿Está en el mismo recinto?	
	<b>A</b>	<b>B</b>
Calificación		
	Posición de la fuente	
	<b>A</b>	<b>B</b>
Calificación		
	Cercanía de la fuente	
	<b>A</b>	<b>B</b>
Calificación		

Etapa 4	¿Está en el mismo recinto?	
	<b>A</b>	<b>B</b>
Calificación		
	Posición de la fuente	
	<b>A</b>	<b>B</b>
Calificación		
	Cercanía de la fuente	
	<b>A</b>	<b>B</b>
Calificación		

**Tabla 3:** Archivo de calificaciones de la evaluación subjetiva.

En total fueron 20 evaluaciones tal y como lo recomienda la *ITU* para esta prueba. Todas fueron realizadas por estudiantes de pregrado de la carrera de Estudios Musicales con Énfasis en ingeniería de sonido de la Pontificia Universidad Javeriana.

### 4.3.3). Resultados

Habiendo realizado las 20 pruebas, se realizó un promedio de las calificaciones obtenidas en cada uno de los 3 parámetros, por cada etapa, para cada audio evaluado. También se obtuvo la desviación estándar de estos. El link de acceso al archivo completo con las 20 evaluaciones (hojas 12 a 31 del archivo de Excel) y los resultados está en la sección A.4 (anexo 4).

Analizando los resultados:

- En la etapa 1:
  - Para el parámetro de reverberación, 16 personas dieron más puntaje al algoritmo fuente imagen que al algoritmo *FDN*.
  - Para el parámetro de espacialización, 13 personas dieron más puntaje al algoritmo fuente imagen.
  - Para el parámetro de percepción de distancia, 13 personas dieron más puntaje al algoritmo fuente imagen.
- En la etapa 2:
  - Para el parámetro de reverberación, 12 personas dieron más puntaje al algoritmo fuente imagen.
  - Para el parámetro de espacialización, 6 personas dieron más puntaje al algoritmo fuente imagen.

- Para el parámetro de percepción de distancia, 10 personas dieron más puntaje al algoritmo fuente imagen.
- En la etapa 3:
  - Para el parámetro de reverberación, 16 personas dieron más puntaje al algoritmo fuente imagen.
  - Para el parámetro de espacialización, 18 personas dieron más puntaje al algoritmo fuente imagen.
  - Para el parámetro de percepción de distancia, 18 personas dieron más puntaje al algoritmo fuente imagen.
- En la etapa 4:
  - Para el parámetro de reverberación, 14 personas dieron más puntaje al algoritmo fuente imagen.
  - Para el parámetro de espacialización, 9 personas dieron más puntaje al algoritmo fuente imagen.
  - Para el parámetro de percepción de distancia, 13 personas dieron más puntaje al algoritmo fuente imagen.

En resumen:

- Para el parámetro de reverberación, el algoritmo fuente imagen obtuvo más puntaje que el algoritmo *FDN* en todas las etapas.
- Para el parámetro de espacialización, el algoritmo Fuente Imagen obtuvo más puntaje que el algoritmo *FDN* en todas las etapas, a excepción de las etapas 2 y 4.
- Para el parámetro de percepción de distancia, el algoritmo fuente imagen obtuvo más puntaje en todas las etapas, a excepción de la etapa 2.

#### **4.4). Aplicación artística del algoritmo**

Como último objetivo específico planteado en el proyecto se decidió realizar un ejercicio en el que se evidencia el uso práctico y artístico del desarrollo del algoritmo fuente imagen con espacialización. El ejercicio consistió en la grabación de un video dentro del estudio (imagen 1, sección anterior). En el video hay tres personas (estudiantes de la carrera) realizando ejercicios de percusión corporal, que se trabajaron durante el semestre en la clase de *Stomp*. Los archivos de audio se grabaron en un estudio del Centro Ático, y posteriormente se procesaron en el algoritmo.

#### 4.4.1). Grabación del video y de los audios

Antes de grabar el video, se determinó que tres personas harían ejercicios de percusión corporal frente a la cámara. Se planeó la trayectoria<sup>54</sup> que harían (tabla 4), teniendo cada uno una porción de espacio para moverse.



**Imagen 2:** Foto de personajes en el recinto durante la grabación del video.

Trayectorias	Receptor <sup>55</sup>		Fuente 1	Fuente 2	Fuente 3	Duración Trayectorias
	Oído L	Oído R				
1	Desde [-0.070, -0.070, -0.64] Hasta [-0.070, 0.070, -0.64]	Desde [0.070, 0.070, -0.64] Hasta [0.070, -0.070, -0.64]	Desde [-2,3, -0.10] Hasta [-2,2,7, -0.10]	Desde [0,2, -0.18] Hasta [0,3, -0.18]	Desde [2,3, -0.13] Hasta [2,2,7, -0.13]	14 segundos
2	Desde [-0.070, 0.070, -0.64] Hasta [-0.070, -0.070, -0.64]	Desde [0.070, -0.070, -0.64] Hasta [0.070, 0.070, -0.64]	Desde [-2,2,7, -0.10] Hasta [-2,2, -0.10]	Desde [0,3, -0.18] Hasta [0,2, -0.18]	Desde [2,2,7, -0.13] Hasta [2,2, -0.13]	11 segundos
3	Desde [-0.070, -0.070, -0.64] Hasta [-0.070, 0.070, -0.64]	Desde [0.070, 0.070, -0.64] Hasta [0.070, -0.070, -0.64]	Desde [-2,2, -0.10] Hasta [-2,2,3, -0.10]	Desde [0,2, -0.18] Hasta [0,3, -0.18]	Desde [2,2, -0.13] Hasta [2,2,4, -0.13]	9 segundos

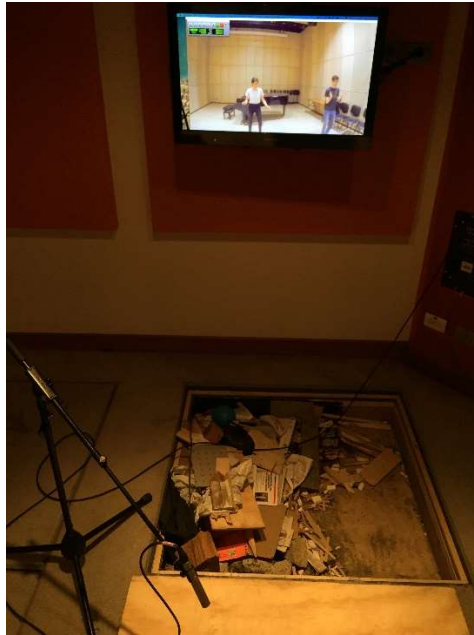
**Tabla 4:** Lista de trayectorias de fuentes sonoras y receptor correspondientes al video grabado para la aplicación artística

Luego de grabar el video, se grabaron los audios (de cada personaje por separado) en el estudio *Foley* (Imagen 3), con un micrófono de condensador *Neumann KM 184*, preamplificador *Avid* e interfaz *Avid HDX*. Se grabó cada uno de estos sonidos por separado para cada persona. Es decir, en total se grabaron 3 canales de

<sup>54</sup> Se define una trayectoria como el cambio de una posición a otra, por pasos.

<sup>55</sup> Se manejó la ubicación de los oídos con coordenadas polares, como se explicó en la sección 4.1.2.2.

aplausos, 3 canales de golpes en los muslos y 3 canales de pasos. Después se juntaron los sonidos correspondientes a cada persona y se exportaron 3 archivos de audio ('D.wav', 'P.wav', 'F.wav'), los cuales posteriormente serían procesados. En el orden en que están escritos, corresponden a fuente sonora 1, fuente sonora 2 y fuente sonora 3.

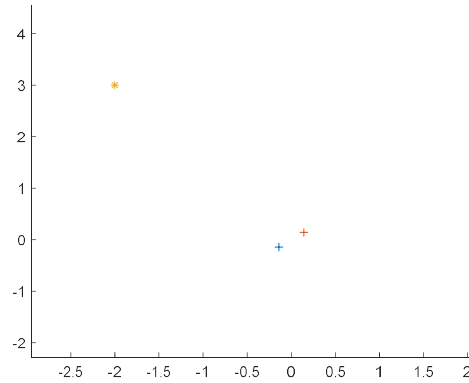


**Imagen 3:** Foto del montaje de la grabación de los audios para la aplicación artística del algoritmo en el video.

#### **4.4.2). Procesamiento de los audios**

En esta sección se muestra el desarrollo del objetivo general de este proyecto, el cual consiste en combinar reverberación con espacialización, agregando el movimiento tanto de las fuentes sonoras como del receptor. Teniendo en cuenta las trayectorias (tabla 4), se modificó el algoritmo fuente imagen (con espacialización, sección 4.2.3) para calcular las respuestas al impulso en cada paso de las trayectorias.

Recordando, a través del algoritmo se obtiene una respuesta al impulso por cada oído respecto a una fuente sonora en una posición fija (figura 28).

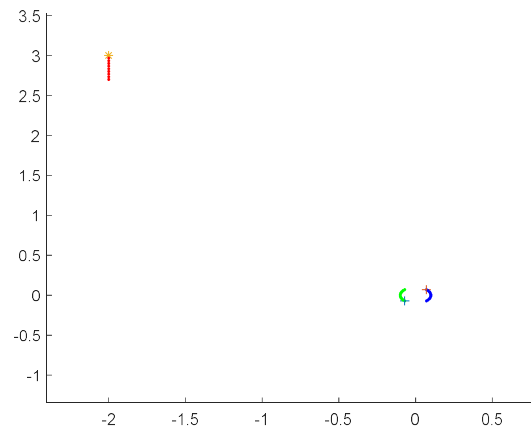


**Figura 28:** Fuente sonora (\*) y receptor (L→+ y R→+) en una posición fija.

Ahora, es importante tener en cuenta que las fuentes sonoras y el receptor se mueven con determinadas trayectorias (figura 29). Se definió que cada trayectoria tenía 10 pasos. Por cada uno, se calcula una respuesta al impulso, para las tres fuentes sonoras, respecto a las dos posiciones de receptor. Esto es, en total se calculan 6 respuestas al impulso por cada paso:

$$h_{D_l}, h_{D_r}, h_{P_l}, h_{P_r}, h_{F_l}, h_{F_r}$$

Estas respuestas al impulso fueron guardadas en matrices. En total se requirieron 6 matrices, cada una de 30 filas (ya que eran 30 pasos en total, al ser 3 trayectorias) y el número de columnas corresponde a la longitud (duración) de las respuestas al impulso.



**Figura 29:** Fuente sonora (\*), pasos de su primera trayectoria (.), receptor (L→+ y R→+) y pasos de su primera trayectoria (L→'.', R→'.').

Habiendo calculado todas las 6 respuestas al impulso para cada paso, se procedió a utilizarlas en otro algoritmo, en el cual se realiza la convolución entre cada una de las señales grabadas y sus correspondientes respuestas al impulso. Para ello se utilizó un ciclo *for* que controla la variable  $n$ , la cual corresponde al número de



muestra para el que se esté realizando la convolución<sup>56</sup>. Se realizó un ciclo por cada *fuentes* sonora, para cada posición del receptor *posrecep* (*l* y *r*); es decir, en total 6 ciclos. En cada uno, se utiliza un vector auxiliar  $h_{fuente_{posrecep}}$ <sup>57</sup> que extrae las respuestas al impulso de las matrices, valor por valor. Con la ecuación (20) se realiza el siguiente procedimiento a través del cual se obtiene la salida  $y_{fuente_{posrecep}}$  muestra a muestra:

$$y_{fuente_{posrecep}}[n] = h_{fuente_{posrecep}} \cdot x_{fuente\_frag}$$

De donde:

$$x_{fuente\_frag} = x_{fuente}(n: -1: n - M) \text{ }^{59}$$

- $x_{fuente}$  es el archivo de la señal de entrada.
- $x_{fuente\_frag}$  es un fragmento del archivo de audio de la señal de entrada.
- $M$  es la longitud del vector  $h_{fuente_{posrecep}}$  (en otras palabras, la duración de la respuesta al impulso).

A manera de ejemplo, para calcular la salida de la fuente 1 respecto a la posición *l* del receptor en la muestra *n*, se realizó lo siguiente:

$$x_{D\_frag} = x_D[n: -1: n - M]$$

$$y_{D_l}[n] = h_{D_l} \cdot x_{D\_frag}$$

Luego, se agruparon las salidas de cada ciclo a señales estéreo:

$$y_D = [y_{D_l} \ y_{D_r}]$$

$$y_P = [y_{P_l} \ y_{P_r}]$$

$$y_F = [y_{F_l} \ y_{F_r}]$$

Posteriormente, se sumaron para obtener la salida general  $y_{out}$ :

$$y_{out} = y_D + y_P + y_F$$

Finalmente, se exportó la salida general.

$$audiowrite('Señal\_resultante.wav', y_{out}, F_s)$$

Al haber realizado todo este proceso se encontró que se puede simular el movimiento continuo de una o varias fuentes sonoras dentro de un recinto, dividiendo el movimiento en pasos y calculando una respuesta al impulso en cada uno. Técnicamente el procedimiento que hizo esto posible, como se menciona en el primer párrafo de la página anterior, es el cambio paulatino (no inmediato) de las respuestas al impulso durante la convolución. En la sección A.3 (Anexo 3) se puede ver el código correspondiente a tal procedimiento.

<sup>56</sup> Como la convolución ya no se realiza con una respuesta al impulso para una sola posición, sino con varias para distintas posiciones que van cambiando, ya no se utiliza la función *conv()* que se venía utilizando normalmente.

<sup>57</sup> La palabra *fuentes* se reemplaza por *D, P* o *F* en el algoritmo, dependiendo del archivo que este siendo procesado (*D.wav, P.wav, F.wav*).

<sup>58</sup> La palabra *posrecep* se reemplaza por *l* o *r* en el algoritmo, dependiendo del *lado* que este siendo procesado.

<sup>59</sup> El fragmento de la señal de entrada va desde *n* (decreciendo de -1 en -1) hasta llegar a  $n - M_{fuente}$ .

## 5). Conclusiones

La programación es una herramienta bastante útil gracias a la cual se pueden realizar tareas de gran dificultad en poco tiempo, siguiendo un procedimiento determinado. Realizar la estimación de una respuesta al impulso para obtener el efecto de reverberación dentro de un recinto, es un proceso que requiere el uso de equipos específicos y de una determinada cantidad de tiempo, entre otros factores. Para poder escuchar una fuente sonora dentro de un recinto moviéndose, se requiere que tanto la fuente como quien la escucha estén físicamente dentro del recinto. Mientras que en este proyecto se evidencia que estas dificultades se pueden evadir, aprendiendo algunos elementos de un lenguaje de programación y dividiendo un problema general en tareas específicas. Con ayuda de la simplificación es posible disminuir el grado de dificultad de cada una de las tareas, para así poder desarrollar con eficiencia el problema planteado.

Específicamente, el problema que se desarrolló fue, como se menciona en el último párrafo de la introducción (sección 1), poder combinar la espacialización sonora con la reverberación dentro de un recinto rectangular simulado. Teniendo en cuenta lo mencionado en el párrafo anterior, se dividió el problema en tareas específicas. La primera fue obtener la reverberación, la cual se desarrolló modelando el comportamiento de las reflexiones dentro de un recinto. Para ello, se estudiaron (secciones 3.4 a 3.6) e implementaron (sección 4.1) tres algoritmos distintos, de los cuales, en uno de estos (*FDN*) no se consideran parámetros de un recinto. En los otros dos (Fuente imagen y *SDN*) se consideran parámetros de un recinto, y, además, la posición de una fuente sonora y un receptor dentro del mismo. La segunda tarea fue incorporar la espacialización para cada caso. No hubo necesidad de modificar los algoritmos, sino agregar algunos pasos adicionales que se explican en la sección 4.2.

Habiendo realizado las dos tareas generales del párrafo anterior, se analizaron ventajas y desventajas de cada algoritmo. En el caso del algoritmo *FDN*, al no considerarse parámetros de un recinto, no se desarrolla con suficiente eficacia el problema planteado. Por otro lado, en el caso del algoritmo *SDN* se tiene la ventaja de la utilización de *HRTF* para obtener la espacialización. Pero, se presentó un problema relacionado con la inestabilidad de la reverberación (coloración en la señal de salida), por lo cual tampoco responde adecuadamente al problema planteado. Por estas razones, el algoritmo con el que se solucionó el problema planteado fue el Fuente Imagen (sección 4.2.3), en el que se obtuvo espacialización con base a diferencias de nivel y de tiempo interaurales (*IID* e *ITD*).

Luego de esta escogencia, se realizó una evaluación subjetiva del procesamiento de dos de los algoritmos mencionados en el párrafo anterior: *FDN* y Fuente Imagen. La prueba con base a la cual se llevó a cabo (*MUSHRA*) tiene la rigurosidad tal que

la evaluación se dividió en partes (etapas). En cada una se calificaron dichos algoritmos de acuerdo con tres aspectos específicos: reverberación, espacialización y percepción de distancia del sonido dentro de un recinto rectangular. Según los resultados, el algoritmo fuente imagen obtuvo mayor calificación en todas las etapas para todos los parámetros, a excepción del parámetro de espacialización en las etapas 2 y 4, y el parámetro de percepción de distancia en la etapa 2. Con base a estos resultados y a la práctica personal, se recomienda utilizar el algoritmo fuente imagen si lo que se busca es la posibilidad de ubicar sonora y espacialmente algún sonido deseado dentro de un recinto rectangular simulado.

A manera de verificación y comprobación de la recomendación mencionada en el párrafo anterior, la etapa final del proyecto consistió en utilizar el algoritmo para una aplicación artística audiovisual (sección 4.4). En esta, tres personas fueron grabadas en el estudio (imagen 1) realizando ejercicios de percusión corporal y moviéndose, simultáneamente. Con base a lo realizado hasta la sección 4.3, en el algoritmo no se había desarrollado la posibilidad de obtener un resultado sonoro con movimiento. Pero, a partir de la posibilidad de obtener respuestas al impulso para cualquier posición de fuente y receptor dentro del recinto simulado, se incorporó (sección 4.4.2). Gracias a este desarrollo del algoritmo se pueden combinar varios archivos de audio y, definiendo una o varias trayectorias para cada fuente sonora que los representa, obtener movimiento automatizado a lo largo de esta(s) trayectoria(s). Como estudiante de artes, esta herramienta la considero de gran aporte a la música y a la postproducción audiovisual, ya que se pueden realizar proyectos que aprovechen creativamente este recurso. Como, por ejemplo, grabar un ensamble acústico en algún estudio insonorizado, y luego, con ayuda del algoritmo, espacializar cada instrumento dentro del recinto simulado.

Para futuros trabajos se busca mejorar el algoritmo en varios aspectos. Por un lado, introduciendo funciones de absorción de los materiales en las superficies del recinto simulado con el fin de modificar la señal en espectro. También se busca dar más involucencia al sonido proveniente de las reflexiones con el uso de *HRTF*, así como se hizo con el algoritmo *SDN* (sección 4.2.2). Incorporar la función de directividad de la fuente (como se hizo con el algoritmo *SDN*) es otro de los aspectos que se busca añadir.

## 6). BIBLIOGRAFÍA

- [1] Oppenheim, A; Schafer, R; Buck, J. *Discrete-Time Signal Processing*. 1989. Prentice-Hall, Inc.
- [2] Hansen, C. *Fundamentals of Acoustics*.
- [3] Beranek, L. *Acústica*. 1961.
- [4] Díaz, R. *Fuentes sonoras y propagación del sonido*. 2007.
- [5] Viveiros, E; Gibbs, B. *An image model for predicting the field performance of acoustic louvres from impulse measurements*. *Applied Acoustics* 64. 2003.
- [6] Smith, J. *CCRMA. Physical Audio Signal Processing for Virtual Musical Instruments and Audio Effects*.
- [7] De Sena, E; Hacıhabiboğlu; Cvetković, Z. *Scattering Delay Network: an interactive Reverberator for Computer Games*. *AES 41<sup>st</sup> International Conference. London, UK. 2011*.
- [8] Hacıhabiboğlu, H; Günel, B; Cvetković. *Simulation of Directional Microphones in Digital Waveguide Mesh-Based Models of Room Acoustics*. 2009-2010. © 2009 IEEE.
- [9] *Mathworks® Documentation. Audio System Toolbox™*. 2016.
- [10] Chemistruck, M; Marcolini, K; Pirkle, W. *Generating Matrix Coefficients for Feedback Delay Network Using Genetic Algorithm*. 2012. *AES Convention Paper No. 8795, 133<sup>rd</sup> Convention, San Francisco, CA, USA*.
- [11] Sierra, J. *Métodos alternativos de espacialización compuesta*. Tesis de Grado Estudios Musicales con énfasis en Ingeniería de sonido. 2014.

## ANEXOS

### A.1). Anexo 1

```
% Algoritmo de reverberación mediante una FDN
% Autor: Miguel Ángel Amado Devia
% Trabajo de Grado 2018-30
function out_fdn = FDNM(in)
%FDNM Reverberador mediante FDN

d = 0.5;%Factor de amplificación de la señal de entrada original
% Matriz de coeficientes de feedback
A = (1/sqrt(5))*[0,-1,0,1;
               -1,0,1,0;
                0,0,-1,1;
                0,-1,-1,0];
% Inicialización de señales retrasadas
j = 2000;
z1 = zeros(1,j); % Primera señal de feedback
z2 = zeros(1,j); % Segunda señal de feedback
z3 = zeros(1,j); % Tercera señal de feedback
z4 = zeros(1,j); % Cuarta señal de feedback
b = [0.1, 0.2, 0.3, 0.4];%vector de factores de la señal original
m = [1009, 591, 853, 793];%Vector de retrasos de la señal
c = [0.1, 0.2, 0.3, 0.4];%vector de factores de la señal retrasada
out_fdn = zeros(1,j);
xmono=in; % Señal de entrada
for k=j+1:length(in)
    % Obtención de señales retrasadas y atenuadas
    z1(k) = b(1)*xmono(k-j) + A(1,1)*z1(k-m(1)) + A(1,2)*z2(k-m(2)) + A(1,3)*z3(k-m(3)) +
A(1,4)*z4(k-m(4));
    z2(k) = b(2)*xmono(k-j) + A(2,1)*z1(k-m(1)) + A(2,2)*z2(k-m(2)) + A(2,3)*z3(k-m(3)) +
A(2,4)*z4(k-m(4));
    z3(k) = b(3)*xmono(k-j) + A(3,1)*z1(k-m(1)) + A(3,2)*z2(k-m(2)) + A(3,3)*z3(k-m(3)) +
A(3,4)*z4(k-m(4));
    z4(k) = b(4)*xmono(k-j) + A(4,1)*z1(k-m(1)) + A(4,2)*z2(k-m(2)) + A(4,3)*z3(k-m(3)) +
A(4,4)*z4(k-m(4));
end
for k=j+1:length(in)
    % Salida del reverberador
    out_fdn(k-j) = c(1)*z1(k-m(1)) + c(2)*z2(k-m(2)) + c(3)*z3(k-m(3)) + c(4)*z4(k-m(4))+
d*xmono(k);
end
end
```

*Published with MATLAB® R2018a*

## A.2). Anexo 2:

```
% Obtención de la Respuesta al Impulso de un recinto a partir del método
% fuente imagen de Gibbs y Viveiros
% Autor: Miguel Ángel Amado Devia
% Trabajo de Grado 2018-30

%Establecer dimensiones del cuarto
ancho=20;largo=20;alto=20;
%Establecer coordenadas de la fuente
xf=7; yf=8; zf=6;
%Establecer coordenadas del receptor.
xr=0; yr=0; zr=0;
%Establecer coordenadas de la célula fuente-imágen
L=2;
M=2;
N=2;
%Establecer variables de coordenadas y distancia de la fuente imágen
dx=0;dy=0;dz=0;Dlmn=0;
%Establecer coeficiente de absorción
coefabsor = 0.25;
%Establecer constante de absorción del aire
ma = 0.02;
% Inicializar vector amplitud de la IR resultante
h = zeros(1,100);
nmuestra = 0;
%Variable de Tiempo Real de acuerdo a la distancia resultante Dlmn
TReal = 0;
Fs = 44100;%Frecuencia de muestreo
Ts = 1/Fs;%Tiempo de muestreo
%Establecer los ciclos 'for' para el tratamiento espacial de la señal
for l = -L:L
    for m = -M:M
        for n = -N:N
            % Variables que guardan los valores de la posición de la fuente
            % imagen en cada componente
            dx = ((ancho*l)+(((-1)^l)*xf))-xr;
            dy = ((largo*m)+(((-1)^m)*yf))-yr;
            dz = ((alto*n)+(((-1)^n)*zf))-zr;
            % Variable que acumula el valor escalar de la distancia entre
            % Fuente imágen y receptor
            Dlmn = sqrt((dx^2)+(dy^2)+(dz^2));
            Nb = abs(l)+abs(m)+abs(n); % Número de reflexiones
            % Función de Absorción del cuarto
            Faten_abs = ((1-coefabsor)^Nb);
            Faten_air = exp(-ma*Dlmn); % Función de absorción del aire
            % Cálculo de la presión
            Faten = sqrt((1/(4*pi*(Dlmn^2)))*Faten_abs*Faten_air);
            % Tiempo y cálculo del # de muestras
            TReal = abs(Dlmn/340);
            nmuestra = abs(round(TReal*44100,0));
```

```

        % Llenado del vector de la IR con los valores de la presión
        h(nmuestra) = Faten;
    end
end
end
h = h/max(h); % Normalización para evitar clip
% Atenuación del sonido directo debida a la distancia
d_f_rec = sqrt(((xf-xr)^2)+((yf-yr)^2)+((zf-zr)^2));
dis_max = sqrt(power(largo-1,2)+power(ancho-1,2)+power(alto-1,2));
dis_min = sqrt(power(2,2));
atend = (((1/d_f_rec)-(1/dis_max))/((1/dis_min)-(1/dis_max)))*0.6+0.2;
h(h==max(h))= h(h==max(h))*atend;

```

*[Published with MATLAB® R2018a](#)*

### A.3). Anexo 3:

```

% ConvoMov
% Autor: Miguel Ángel Amado Devia
% Trabajo de Grado 2018-30
clc, clear, close all
% Lectura de archivos de audio
[x_D, Fs] = audioread('Dan.wav'); % Señal 1
x_P = audioread('Pau.wav'); % Señal 2
x_F = audioread('Fel.wav'); % Señal 3

% Inicialización de las señales procesadas
y_D_l = zeros(100,1); y_D_r = zeros(100,1);
y_P_l = zeros(100,1); y_P_r = zeros(100,1);
y_F_l = zeros(100,1); y_F_r = zeros(100,1);
% Inicialización de la señal resultante
y_out = zeros(100,2);
% Cargar las matrices de IRS por cada fuente sonora
load MatrizOidoL_D.mat
load MatrizOidoR_D.mat
load MatrizOidoL_F.mat
load MatrizOidoR_F.mat
load MatrizOidoL_P.mat
load MatrizOidoR_P.mat
% Variable de indexado de cada IR dentro de cada matriz
indiceIRS = 1;
% Variables de IR de acuerdo a cada fuente
hL_D = MatrizOidoL_D(indiceIRS,:); hR_D = MatrizOidoR_D(indiceIRS,:);
hL_P = MatrizOidoL_P(indiceIRS,:); hR_P = MatrizOidoR_P(indiceIRS,:);
hL_F = MatrizOidoL_F(indiceIRS,:); hR_F = MatrizOidoR_F(indiceIRS,:);
ML_D = length(hL_D)-1; MR_D = length(hR_D)-1;
ML_P = length(hL_P)-1; MR_P = length(hR_P)-1;
ML_F = length(hL_F)-1; MR_F = length(hR_F)-1;
% Variable que determina la velocidad de cambio de IR (velocidad de
% movimiento)
controlCambioIR = 0; controlCambiovalor = 0;

```

```

control_IRL_D = length(hL_D)*round(round(((length(x_D)-1)-ML_D+1)/length(hL_D),0)/30,0);
control_IRR_D = length(hR_D)*round(round(((length(x_D)-1)-MR_D+1)/length(hR_D),0)/30,0);
control_IRL_P = length(hL_P)*round(round(((length(x_P)-1)-ML_P+1)/length(hL_P),0)/30,0);
control_IRR_P = length(hR_P)*round(round(((length(x_P)-1)-MR_P+1)/length(hR_P),0)/30,0);
control_IRL_F = length(hL_F)*round(round(((length(x_F)-1)-ML_F+1)/length(hL_F),0)/30,0);
control_IRR_F = length(hR_F)*round(round(((length(x_F)-1)-MR_F+1)/length(hR_F),0)/30,0);

```

## Convolución con automatización de movimiento para cada fuente sonora

Se cambian las IRs muestra a muestra para generar la automatización de movimiento por cada fuente sonora

```

% Fuente 2
for n = ML_D+1:length(x_D)-1 % Lado L
    if (indiceIRs>=1)&&(indiceIRs<10) % En la primera trayectoria
        controlCambioIR = controlCambioIR + 0.92;
    end
    if (indiceIRs>=10)&&(indiceIRs<20) % En la segunda trayectoria
        controlCambioIR = controlCambioIR + 0.921;
    end
    if (indiceIRs>=20)&&(indiceIRs<30) % En la tercera trayectoria
        controlCambioIR = controlCambioIR + 0.9215;
    end
    if indiceIRs == 30
        controlCambioIR = 0;
    end
    % Siguiete posición horizontal de la IR actual
    controlCambioValor = controlCambioValor + 1;
    %Despues de un tiempo perceptible, cambiar valor a valor la IR, sin
    % cortar la cola reverberante de la convolución anterior
    if controlCambioIR >= control_IRL_D
        % Siguiete IR
        indiceIRs = indiceIRs + 1;
        % Reinicia la posición horizontal de la IR
        controlCambioValor = 1;
        hL_D(controlCambioValor) = MatrizOidoL_D(indiceIRs, controlCambioValor);
        % Reiniciar la variable de control de cambio de IR
        controlCambioIR = 0;
    end
    % Controlar indexado horizontal de la IR
    if controlCambiovalor > length(hL_D)
        controlCambiovalor = 1;
    end
    hL_D(controlCambioValor) = MatrizOidoL_D(indiceIRs, controlCambioValor);
    x_D_Vector = x_D(n:-1:n-ML_D);
    % Convolucion (muestra a muestra) con la señal de entrada
    y_D_l(n) = hL_D*x_D_Vector;
end
indiceIRs = 1;
controlCambioIR = 0;
controlCambioValor = 0;
for n = MR_D+1:length(x_D)-1 % Lado R
    if (indiceIRs>=1)&&(indiceIRs<10) % En la primera trayectoria

```



```

        controlCambioIR = controlCambioIR + 0.92;
    end
    if (indiceIRs>=10)&&(indiceIRs<20) % En la segunda trayectoria
        controlCambioIR = controlCambioIR + 0.921;
    end
    if (indiceIRs>=20)&&(indiceIRs<30) % En la tercera trayectoria
        controlCambioIR = controlCambioIR + 0.9215;
    end
    if indiceIRs == 30
        controlCambioIR = 0;
    end
    controlCambioValor = controlCambioValor + 1;
    if controlCambioIR >= control_IRR_D
        % Siguiente IR
        indiceIRs = indiceIRs + 1;
        % Reinicia la posición horizontal de la IR
        controlCambioValor = 1;
        hR_D(controlCambioValor) = MatrizOidoR_D(indiceIRs, controlCambioValor);
        controlCambioIR = 0;
    end
    if controlCambioValor > length(hR_D)
        controlCambioValor = 1;
    end
    hR_D(controlCambioValor) = MatrizOidoR_D(indiceIRs, controlCambioValor);
    x_D_Vector = x_D(n:-1:n-MR_D);
    y_D_r(n) = hR_D*x_D_Vector;%Convolucion con la señal de entrada
end
indiceIRs = 1;
controlCambioIR = 0;
controlCambioValor = 0;
% Fuente 2
for n = ML_P+1:length(x_P)-1 % Lado L
    if (indiceIRs>=1)&&(indiceIRs<10) % En la primera trayectoria
        controlCambioIR = controlCambioIR + 0.92;
    end
    if (indiceIRs>=10)&&(indiceIRs<20) % En la segunda trayectoria
        controlCambioIR = controlCambioIR + 0.921;
    end
    if (indiceIRs>=20)&&(indiceIRs<30) % En la tercera trayectoria
        controlCambioIR = controlCambioIR + 0.9215;
    end
    controlCambioValor = controlCambioValor + 1;
    if controlCambioIR >= control_IRL_P
        indiceIRs = indiceIRs + 1;
        if controlCambioValor == length(MatrizOidoL_P(indiceIRs,:))
            controlCambioValor = 1;
        end
        hL_P(controlCambioValor) = MatrizOidoL_P(indiceIRs, controlCambioValor);
        controlCambioIR = 0;
    end
    if controlCambioValor > length(hL_P)
        controlCambioValor = 1;
    end
    hL_P(controlCambioValor) = MatrizOidoL_P(indiceIRs, controlCambioValor);
end

```

```

x_P_Vector = x_P(n:-1:n-ML_P);
y_P_l(n) = hL_P*x_P_Vector; % Convolucion con la señal de entrada
end
indiceIRs = 1;
controlCambioIR = 0;
controlCambioValor = 0;
for n = MR_P+1:length(x_P)-1 % Lado R
    if (indiceIRs>=1)&&(indiceIRs<10) % En la primera trayectoria
        controlCambioIR = controlCambioIR + 0.92;
    end
    if (indiceIRs>=10)&&(indiceIRs<20) % En la segunda trayectoria
        controlCambioIR = controlCambioIR + 0.921;
    end
    if (indiceIRs>=20)&&(indiceIRs<30) % En la tercera trayectoria
        controlCambioIR = controlCambioIR + 0.9215;
    end
    controlCambioValor = controlCambioValor + 1;
    if controlCambioIR >= control_IRR_P
        indiceIRs = indiceIRs + 1;
        if controlCambioValor > length(MatrizOidoR_P(indiceIRs,:))
            controlCambioValor = 1;
        end
        hR_P(controlCambioValor) = MatrizOidoR_P(indiceIRs, controlCambioValor);
        controlCambioIR = 0;
    end
    if controlCambioValor > length(hR_P)
        controlCambioValor = 1;
    end
    hR_P(controlCambioValor) = MatrizOidoR_P(indiceIRs, controlCambioValor);
    x_P_Vector = x_P(n:-1:n-MR_P);
    y_P_r(n) = hR_P*x_P_Vector; % Convolucion con la señal de entrada
end
indiceIRs = 1;
controlCambioIR = 0;
controlCambioValor = 0;
% Fuente 3
for n = ML_F+1:length(x_F)-1 % Lado L
    if (indiceIRs>=1)&&(indiceIRs<10) % En la primera trayectoria
        controlCambioIR = controlCambioIR + 0.92;
    end
    if (indiceIRs>=10)&&(indiceIRs<20) % En la segunda trayectoria
        controlCambioIR = controlCambioIR + 0.921;
    end
    if (indiceIRs>=20)&&(indiceIRs<30) % En la tercera trayectoria
        controlCambioIR = controlCambioIR + 0.9215;
    end
    controlCambioValor = controlCambioValor + 1;
    if controlCambioIR >= control_IRL_F
        indiceIRs = indiceIRs + 1;
        if controlCambioValor > length(MatrizOidoL_F(indiceIRs,:))
            controlCambioValor = 1;
        end
        hL_F(controlCambioValor) = MatrizOidoL_F(indiceIRs, controlCambioValor);
        controlCambioIR = 0;
    end
end

```

```

end
if controlCambiovalor > length(hL_F)
    controlCambiovalor = 1;
end
hL_F(controlCambiovalor) = MatrizOidoL_F(indiceIRs, controlCambiovalor);
x_F_Vector = x_F(n:-1:n-ML_F);
y_F_l(n) = hL_F*x_F_Vector; % Convolucion con la señal de entrada
end
indiceIRs = 1;
controlCambioIR = 0;
controlCambiovalor = 0;
for n = MR_F+1:length(x_F)-1 % Lado R
    if (indiceIRs>=1)&&(indiceIRs<10) % En la primera trayectoria
        controlCambioIR = controlCambioIR + 0.92;
    end
    if (indiceIRs>=10)&&(indiceIRs<20) % En la segunda trayectoria
        controlCambioIR = controlCambioIR + 0.921;
    end
    if (indiceIRs>=20)&&(indiceIRs<30) % En la tercera trayectoria
        controlCambioIR = controlCambioIR + 0.9215;
    end
    controlCambiovalor = controlCambiovalor + 1;
    if controlCambioIR >= control_IRR_F
        indiceIRs = indiceIRs + 1;
        if controlCambiovalor > length(MatrizOidoR_F(indiceIRs,:))
            controlCambiovalor = 1;
        end
        hR_F(controlCambiovalor) = MatrizOidoR_F(indiceIRs, controlCambiovalor);
        controlCambioIR = 0;
    end
    if controlCambiovalor > length(hR_F)
        controlCambiovalor = 1;
    end
    hR_F(controlCambiovalor) = MatrizOidoR_F(indiceIRs, controlCambiovalor);
    x_F_Vector = x_F(n:-1:n-MR_F);
    y_F_r(n) = hR_F*x_F_Vector; % Convolucion con la señal de entrada
end

```

## Señal resultante

```

% Resultado de señales procesadas en estéreo
y_D = [y_D_l y_D_r];
y_P = [y_P_l y_P_r];
y_F = [y_F_l y_F_r];
% Luego, se suman
yout = (y_D + y_P+ y_F);
% Finalmente, se exporta el resultado
audiowrite('Audio_Final_Video.wav', yout, Fs);

```

*Published with MATLAB® R2018a*

#### **A.4). Anexo 4**

Calificaciones y resultados de la validación subjetiva. Pestañas 12 a 31 son las 20 evaluaciones, las hojas anteriores son pruebas previas que no afectan en el análisis realizado. Pestaña de resultados al final del archivo.

<https://drive.google.com/file/d/1OJrFM2sg0sz2emFnXFWSpDoPenjze6Lr/view?usp=sharing>