

# Desarrollo de software para el cálculo de parámetros acústicos ISO 3382

Agustin Anton<sup>1</sup>, Franco Ojeda<sup>2</sup>

9 de noviembre de 2021

Señales & Sistemas, Ingeniería de sonido

Universidad Nacional de Tres de Febrero, Buenos Aires, Argentina.

<sup>1</sup> agustin.anton.d@hotmail.com, <sup>2</sup> franco.almendra@hotmail.com,

## Resumen

El siguiente trabajo tiene como objetivo diseñar un software en Python que calcule distintos parámetros acústicos de un recinto, descritos por la normativa ISO-3382, a partir de su respuesta a un sine sweep logarítmico, grabado con el software o cargado de alguna grabación previa. El software se divide en tres grandes rasgos, la sintetización de señales, el procesamiento y filtrado de señales, y por último el cálculo de parámetros acústicos. No se pudo llegar a un resultado aceptable debido al fallo en el cálculo de los parámetros acústicos, no se pudo llegar a la raíz del error, se deben reiterar las funciones de suavizado y de cálculo de parámetros acústicos para encontrar las fallas.

**Keywords:** ISO 3382, IR

## 1. INTRODUCCIÓN

El presente trabajo tiene como objetivo calcular los distintos parámetros acústicos de un recinto (EDT, T60, D50, C80) propuestos por la normativa ISO 3382[1], realizando los distintos métodos numéricos y matemáticos a través de un software diseñado con Python 3. Dicho software se encarga de la sintetización, adquisición y procesamiento de señales para luego calcular los parámetros buscados. Como antecedente para este proyecto se toman como base los trabajos "Simultaneous Measurement of Impulse Response and Distortion With a Swept-Sine Technique"[3] de Angelo Farina y "Impulse Response Measurement With Sine Sweeps and Amplitude Modulation Schemes"[4] de Q. Meng, D. Sen, S. Wang y L. Hayes.

## 2. MARCO TEÓRICO

### 2.1. Ruido Rosa



El ruido rosa es un ruido que tiene la característica de que su densidad espectral de potencia es proporcional a la inversa de la frecuencia  $1/f$ , con  $f$  siendo la frecuencia. Esto implica que el nivel del ruido rosa decae a razón de 3 dB/octava, como

se ve en la figura 1, pero que su nivel en una distribución de tercios de octava es constante.

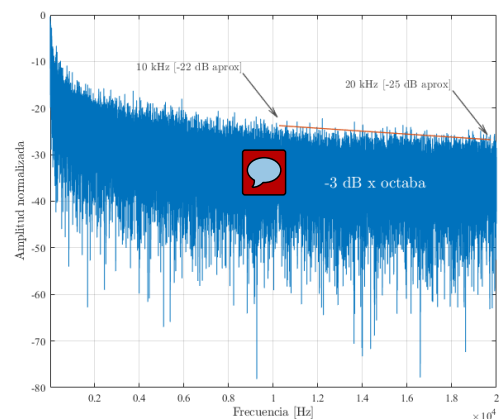


Figura 1: Distribución espectral de ruido rosa [dBFS].

### 2.2. Sine Sweep Logarítmico + Filtro Inverso

Un Sine Sweep es una señal del tipo seno que barre todo el espectro entre una frecuencia angular inicial  $w_1$  y una final  $w_2$ , excitando una frecuencia

a la vez, durante un período de tiempo  $T$ . El Sine Sweep logarítmico varía la frecuencia instantánea exponencialmente en el tiempo y como tiene la característica de excitar sólo una frecuencia a la vez permite concentrar toda la energía de la señal en un ancho de banda estrecho. Su amplitud depende de la frecuencia y su distribución espectral se caracteriza por decaer a razón de 3 dB/octava. Está dado por la siguiente ecuación:

$$x(t) = \sin\left(\frac{Tw_1}{\ln\left(\frac{w_2}{w_1}\right)} \cdot e^{\frac{t \ln\left(\frac{w_2}{w_1}\right)}{T}} - 1\right) \quad (1)$$

En la figura 2 se puede apreciar un sine sweep desarrollándose en el tiempo.

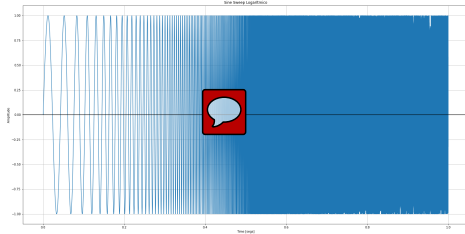


Figura 2: Sine Sweep Logarítmico.

El filtro inverso es el sine sweep invertido en el tiempo, pero debido al decaimiento en el espectro del sine sweep logarítmico se requiere una modulación en amplitud para compensar la pérdida de energía, dicha modulación está dada por la frecuencia instantánea y su ecuación es la siguiente:

$$m(t) = \frac{w_1}{2\pi \cdot w_1 \cdot e^{\frac{t \ln\left(\frac{w_2}{w_1}\right)}{T}}} \quad (2)$$

De esta forma, el filtro inverso tiene la característica de que su espectro crece a razón de 3 dB/octava, siendo  $x(-t)$  el sine sweep invertido en el tiempo, el filtro inverso queda definido como:

$$k(t) = m(t) \cdot x(-t) \quad (3)$$

En la figura 3 se puede ver como se desarrolla el filtro inverso en el tiempo.

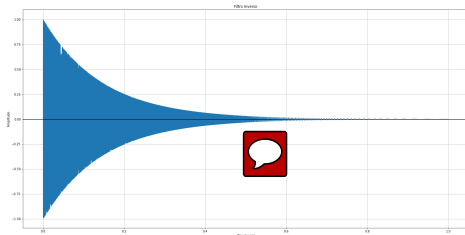


Figura 3: Filtro Inverso.

## 2.3. Respuesta al Impulso

La respuesta al impulso de un recinto se define como la evolución en el tiempo de la presión sonora en un punto dado del recinto ante una señal de entrada del tipo impulsiva. En un sistema LTI si se conoce la respuesta al impulso se puede predecir la salida del sistema ante cualquier entrada a través de la convolución de la señal de entrada con la respuesta al impulso. Se puede sintetizar la respuesta al impulso de una frecuencia central  $f_i$  con la siguiente fórmula:

$$y_i = A_i e^{\frac{\ln(10^{-3})}{T_{60}}} \cos(2\pi f_i t) \quad (4)$$

Siendo  $A_i$  el coeficiente de amplitud para esa frecuencia, de la misma forma que  $T_{60}$  es el tiempo de reverberación para esa frecuencia. La respuesta al impulso se obtiene sumando la respuesta de las  $n$  frecuencias centrales  $f_i$ :

$$Y_i = \sum_{n=1}^n y_i \quad (5)$$

Para obtener la respuesta al impulso de un recinto con un barrido de sine sweep logarítmico se utilizó el método propuesto por Farina[3] que consiste en realizar la convolución de la respuesta del recinto al sine sweep con el filtro inverso generado para ese sine sweep específico, de forma que, siendo  $y(t)$  la respuesta del recinto captada por el micrófono y  $k(t)$  el filtro inverso, la respuesta al impulso  $h(t)$  está dada por:

$$h(t) = y(t) * k(t) \quad (6)$$

## 2.4. Conversión a escala logarítmica normalizada

Debido a la naturaleza de las señales con las que se trabajan para calcular los parámetros acústicos de interés es conveniente realizar un cambio de escala para visualizar mejor el fenómeno de la respuesta al impulso. Siendo  $A(t)$  la señal de interés, la conversión está dada por:

$$R(t) = 20 \log_{10} \frac{A(t)}{A(t)_{max}} \quad (7)$$

## 2.5. Suavizado de Señal

### 2.5.1. Transformada de Hilbert

La transformada de Hilbert es un filtro que consiste en la convolución de la señal que se quiere transformar  $A(t)$  con la función  $\frac{1}{\pi t}$ :

$$H(A(t)) = A(t) * \frac{1}{\pi t} \quad (8)$$

Una vez aplicada la transformada se construye la señal analítica  $A_a(t)$  y se obtiene la envolvente de la señal aplicando el módulo a  $A_a(t)$ :

$$A_a(t) = A(t) + j \cdot H(A(t)) \quad (9)$$

### 2.5.2. Integral de Schroeder

La Integral de Schroeder se utiliza para obtener una curva de decaimiento de la respuesta al impulso más suavizada, facilitando el cálculo de los parámetros acústicos[5]. Está dada por:

$$E(t) = \int_t^\infty p^2(\tau) d\tau = \int_0^\infty p^2(\tau) d\tau - \int_0^t p^2(\tau) d\tau \quad (10)$$

Siendo  $E(t)$  la energía de la señal en función del tiempo  $p(\tau)$  la respuesta al impulso. Aunque el límite superior de la integral sea infinito, en la práctica ese límite sería el final del impulso, mientras más tiempo de "silencio" (siempre hay ruido de fondo) se incluya en la integral menos precisa va a ser esta transformada.

### 2.5.3. Regresión Lineal por Mínimos Cuadrados

Se tiene un conjunto de datos, que puede ser forzado a que se parezca a una función continua, el polinomio de grado  $n$  que mejor aproxima ese conjunto de puntos. Este polinomio no tiene que pasar por cada uno de los puntos (interpola, no ajusta). Entonces, con  $k$  puntos  $(x_0, y_0)$ ,  $(x_1, y_1)$ , ...,  $(x_k, y_k)$  se arma un polinomio de grado  $n$ :

$$p(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n \quad (11)$$

El objetivo es hallar los coeficientes del polinomio  $(a_0, a_1, \dots, a_n)$ . Para eso, se evalúa cada punto del polinomio y se arman las siguientes matrices:

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_k \end{pmatrix} = \begin{pmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_k & x_k^2 & \dots & x_k^n \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix}$$

Más compacto:

$$Y = P \cdot A \quad (12)$$

Se quiere despejar  $A$ , pero como  $P$  no es cuadrada, para invertirla se multiplica por su transpuesta, y despejando para obtener  $A$ :

$$A = (P^T \cdot P)^{-1} \cdot P^T \cdot Y \quad (13)$$

## 2.6. Parámetros Acústicos

### 2.6.1. Tiempo de Reverberación

Es el tiempo, medido en segundos, que tarda la presión sonora en un recinto en caer 60 dB a partir del momento en que se apaga la fuente sonora. Debido a la presencia de ruido de fondo generalmente se calcula a partir de una extrapolación calculando la caída, a partir de un decaimiento de 5 dB del nivel promedio, de 30 (T30), 20 (T20) o 10 dB (T10) y multiplicando ese valor por el escalar correspondiente. Este parámetro se calcula en bandas de octava o de tercios de octava según las normativas IEC-61260[2] e ISO-3382[1].

### 2.6.2. EDT (Early Decay Time)

Es el tiempo, en segundos, en que la señal cae de 0 dBFS a -10 dBFS, está vinculado al decaimiento de las reflexiones tempranas.

### 2.6.3. Definition (D50)

Relación entre la energía acumulada durante los primeros 50 ms con el resto de la señal. Este parámetro se utiliza para evaluar la inteligibilidad de un recinto. Está definida por:

$$D_{50} = \frac{\int_0^{0,05} p^2(t) dt}{\int_0^\infty p^2(t) dt} \quad (14)$$

### 2.6.4. Clarity (C80)

Es la relación entre la energía de las reflexiones tempranas y las tardías. Se utiliza para evaluar la claridad del sonido en un recinto, a mayor energía energía temprana mayor claridad. Viene dado por:

$$C_{80} = 10 \log_{10} \frac{\int_0^{0,08} p^2(t) dt}{\int_{0,08}^\infty p^2(t) dt} [dB] \quad (15)$$

## 3. DESARROLLO EXPERIMENTAL

El software sigue el siguiente diagrama de flujo:

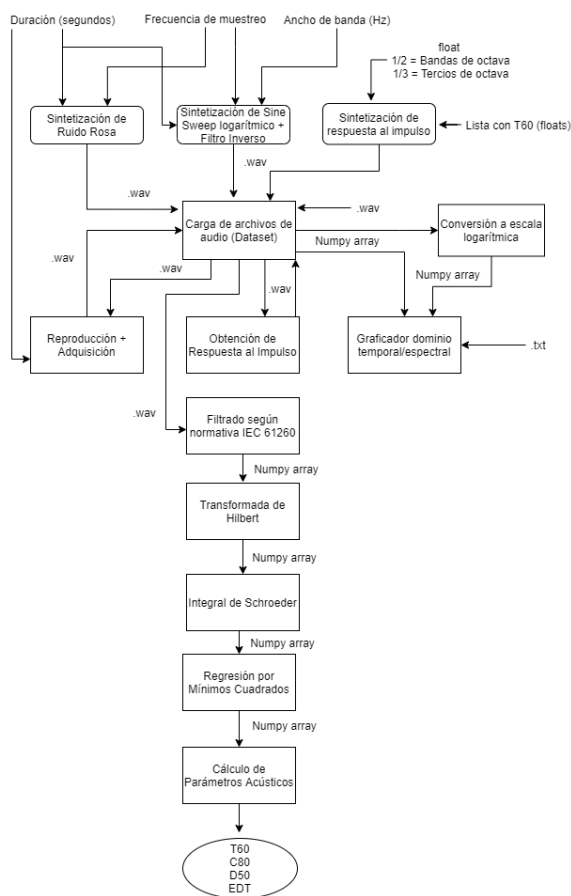


Figura 4: Diagrama de flujo del software.

Para poder acceder a las distintas funciones del software es necesario sintetizar alguna señal o cargar algún archivo de audio, de otra forma el funcionamiento del programa se verá interrumpido. Como se puede ver en el diagrama de flujo la mayoría de las funciones que procesan audio cargan sus resultados en el dataset, esto se realiza de forma automática para que el usuario no tenga que preocuparse por hacerlo manualmente, dando la opción de cargar sus propios audios para su posterior procesamiento. De forma que, viendo el menú en la figura 5 se tiene que acceder a la opción 1,2,4 o 6 antes de poder acceder a las demás, cabe destacar también que las funciones graficadoras y de procesamiento de señal van a pedir el nombre del archivo .wav (por ejemplo "Sweep.wav") por lo que el usuario tiene que estar atento a los nombres de los archivos y escribirlos correctamente, o el software no va a funcionar.

```
Menu Principal
1. Generar ruido rosa
2. Generar sinewave logarítmico + filtro inverso
3. Reproducción de audio y grabación de la respuesta en simultáneo
4. Cargar archivos de audio
5. Graficar dominio temporal o espectral
6. Sintetizar respuesta al impulso
7. Obtener respuesta al impulso
8. Calcular parámetros acústicos
9. Finalizar programa
```

Figura 5: Menú del software.

Una vez cargados los audios sintetizados o propios se puede acceder a las distintas funciones del software libremente, y una vez realizada la acción requerida se vuelve al menú principal, y para finalizar el programa basta con introducir la opción de finalizar programa, una vez finalizado se pierden todos los datos cargados y todos los archivos de audio se almacenan en la carpeta donde se encuentra "Main.py".

### 3.1. Sintetización de archivos de audio

#### 3.1.1. Ruido Rosa

Esta función genera ruido rosa utilizando el algoritmo de Voss-McCartney[x], como valores de entrada tiene la duración que desea el usuario y la frecuencia de muestreo que va a tener el audio. La función devuelve un Numpy array con la información del audio y genera el archivo "ruidorosa.wav" en la carpeta donde se encuentre "Main.py", si ya existe "ruidorosa.wav" este se verá sobrescrito.

#### 3.1.2. Sine Sweep + Filtro Inverso

Ambas funciones tienen como valores de entrada la duración deseada para la señal, la frecuencia de muestreo y el ancho de banda que van a barrer. Una vez ingresados los parámetros necesarios se aplican en las ecuaciones (1), (2) y (3) y al tener todo listo las funciones devuelven los Numpy arrays correspondientes y graban los archivos "Sweep.wav" y "Filtro\_inverso.wav"

#### 3.1.3. Respuesta al impulso

Esta función toma como argumentos de entrada un Numpy array con los valores de T60 que se desean recrear y el argumento "G", que solo admite 1/2 o 1/3 para poder sintetizar la respuesta al impulso por bandas de octava o de tercios de octava. Dentro de "Main.py" el usuario va llenando una lista con los valores requeridos, a la hora de ingresar los valores el software señala la frecuencia central para no perder de vista el orden y evitar errores, una vez completada la lista se utiliza Numpy para convertirla en array y correr la función, que aplica las ecuaciones (4) y (5). La función devuelve un

Numpy array con la información del audio y graba en la carpeta en la que se encuentra "Main.py" el archivo "impulso\_sintetizado.wav", si ya existe un archivo con el mismo nombre va a ser sobrescrito.

### 3.2. Adquisición + Reproducción

Esta función toma como argumentos de entrada un string con el nombre del archivo de audio a reproducir y un float que respresenta la cantidad de tiempo en segundos que el usuario desea grabar la respuesta. Por ejemplo, se puede tener un audio que dure 5 segundos pero al tener un recinto que no disipe rápidamente la energía se puede querer seguir grabando después de finalizada la reproducción, por ejemplo, unos 8 segundos para terminar de captar bien la respuesta del recinto. Al ejecutar la función se va a imprimir en pantalla los dispositivos por default de entrada y de salida, al menos los que utiliza el módulo Sounddevice, que es el encargado de reproducir y grabar los archivos de audio, posteriormente se da la opción de cambiar estos dispositivos por default, el usuario puede cambiar el dispositivo de entrada, de salida o ambos para luego proceder a la reproducción y grabación en simultáneo. La función graba el archivo "audio.wav", si ya existe un archivo con ese nombre se sobrescribe, y devuelve un array con la información del audio grabado.

### 3.3. Función de carga de archivos de audios y funciones graficadoras

#### 3.3.1. Carga de archivos de audio (Dataset)

Esta función toma como argumento de entrada una lista con strings que contienen los nombres de los archivos de audio a cargar, y devuelve un diccionario cuyas keys son los nombres de los archivos (ej: "Sweep.wav") y el contenido son tuplas que en la primer posición contienen un Numpy array con la información del audio y en la segunda posición tiene la frecuencia de muestreo del archivo (para más información leer el docstring de la función cargar\_archivos.py). Para ejecutar esta función hay que crear una variable para asignarle el diccionario, en "Main.py" cuando se ejecuta esta función primero se consulta al usuario cuantos archivos quiere cargar, se debe ingresar un entero o el programa va a dejar de funcionar, luego se llena una lista con los nombres de los archivos, como se muestra en el ejemplo anterior, se cargan dichos archivos a un diccionario que se asigna a la variable "archivos" para luego seguir siendo utilizada, por esta razón es que la mayoría de funciones del software piden los nombres de los archivos, ya que se buscan en el dataset según su nombre.

#### 3.3.2. Funciones graficadoras

La función que grafica el dominio temporal de una señal toma como argumentos de entrada el Numpy array con la información del audio, su frecuencia de muestreo, la duración (que se puede ingresar como la división entre la cantidad de elementos del array sobre la frecuencia de muestreo) y como argumento opcional el título del gráfico, que por defecto es "Signal". Para graficar el dominio espectral de una señal es necesario primero realizar el análisis espectral en el software Audacity, o algún programa similar que exporte los datos en un archivo .txt dividido en dos columnas, siendo la primera la frecuencia en Hz y la segunda la amplitud en dBFS, excluyendo o borrando manualmente los nombres de las columnas, dejando solamente los datos numéricos. La función toma como argumento de entrada un string del nombre del archivo .txt (ej: "Espectro.txt"), los valores máximos y mínimos de los ejes (es recomendable no ingresar nada en esos argumentos) y el título del gráfico, que por defecto es "Espectro".

### 3.4. Procesamiento de señales

#### 3.4.1. Obtención de la respuesta al impulso

Esta función tiene como argumentos de entrada dos strings, siendo el primero el nombre del archivo de audio de la respuesta del recinto al sine sweep y el segundo el nombre del archivo de audio del filtro inverso para ese sine sweep. La función lee los archivos y aplica la ecuación (6), utilizando el módulo Signal de Scipy, para obtener la respuesta al impulso, se obtiene realizando la multiplicación de las transformadas de Fourier de ambas señales, que es equivalente a la convolución de ambas señales en el dominio del tiempo, y luego realiza la antitransformada. La función genera el archivo de audio "respuesta\_al\_impulso.wav", que si ya existe va a ser sobrescrito, y devuelve un Numpy array con la información del audio obtenido.

#### 3.4.2. Filtrado según norma IEC-61260

Esta función toma como argumentos de entrada un string con el nombre del archivo de audio a filtrar, un float que debe ser 1/2 o 1/6 para filtrar por bandas de octava o de tercios de octava, respectivamente, y un int que indica el orden del filtrado que se va a aplicar. Una vez filtrada la señal, la función devuelve tres elementos; un diccionario principal que tiene de keys las frecuencias centrales de las bandas que se filtraron con los arrays de la señal filtrada como valores; un diccionario que contiene los centros de bandas de frecuencia como keys y los arrays de la respuesta en frecuencia del

filtro como valor; y un diccionario que contiene los centros de bandas de frecuencia como keys y los arrays de las frecuencias angulares de los valores de H como valor, estos dos últimos diccionarios tienen información que sirve para graficar la respuesta del filtro. Dependiendo de la división por bandas el diccionario principal contiene 10 o 30 Numpy arrays.

### 3.4.3. Suavizado de señal

La primer etapa de suavizado es la transformada de Hilbert, esta función toma como argumento de entrada un Numpy array con la información del audio, como argumentos opcionales toma dos enteros, el primero indica la cantidad de componentes de Fourier y el segundo el eje del array sobre el cual se va a ejecutar la transformada, se recomienda no alterar los valores por defecto de estos argumentos. La función ejecuta la ecuación (8) para armar la función de la ecuación (9) y finalmente devolver un Numpy array con la envolvente de la señal, que es el módulo de la ecuación (9). La segunda etapa de suavizado es la integral de Schroeder, cuya función toma como argumento de entrada el Numpy array devuelto por la función anterior para aplicar la ecuación (10) y devolver un Numpy array con la señal transformada, en dBFS. Por último, la señal pasa por la función de mínimos cuadrados, que toma como argumentos de entrada primero un Numpy array con la cantidad de elementos necesarios para representar la duración del audio, como segundo argumento toma el Numpy array devuelto por la función de suavizado y como último argumento toma un entero que representa el grado del polinomio que se quiere construir, en este caso está por default en 1 ya que se quiere construir una recta. Esta función aplica las ecuaciones (11) y (12) para luego devolver un Numpy array con los coeficientes del polinomio para luego construir la recta.

## 3.5. Cálculo de parámetros acústicos

La función de cálculo del T60 toma como argumento de entrada el Numpy array de la señal interpolada que devuelve la función de mínimos cuadrados, un entero que es la frecuencia de muestreo del archivo de audio original y un string donde se debe indicar T10, T20 o T30 para calcular el T60 a partir de eso, la función devuelve un float. Para calcular el T60 la función busca el valor de -5 dBFS en el array y realiza una resta entre la preimagen de ese punto con el punto de caiga -15, -25 o -35 dBFS, dependiendo de la opción que se haya elegido, luego multiplica ese valor por 6, 3 o 2 y se obtiene el T60. La función del EDT funciona exactamente igual que la del T60, solo que esta busca los valores de 0 y -10 dBFS. Toma como argumento

de entrada un numpy array y la frecuencia de muestreo, y devuelve un float. Las funciones de C80 y D50 tienen los mismos argumentos de entrada que la función de EDT y simplemente aplican las ecuaciones (14) y (15), respectivamente, a la señal de entrada, devolviendo un float.

## 4. RESULTADOS

### 4.1. Sintetización de archivos de audio

#### 4.1.1. Ruido Rosa

La función de Ruido Rosa cumple con los objetivos planteados de tener un espectro que decae a razón de 3 dB/octava, en la figura 6 se puede apreciar el espectro de dicha señal, graficado por la función de dominio espectral:

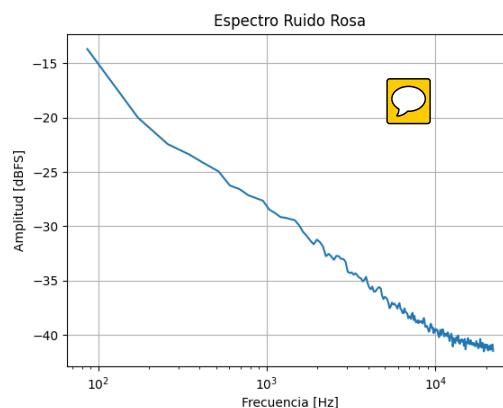


Figura 6: Espectro del Ruido Rosa.

#### 4.1.2. Sine Sweep + Filtro Inverso

Las funciones de Sine Sweep logarítmico y Filtro Inverso también cumplen con los comportamientos requeridos, con los espectros decayendo 3 dB/octava y aumentando 3 dB/octava, respectivamente, se pueden ver los espectros de las señales generadas en las figuras 7 y 8:

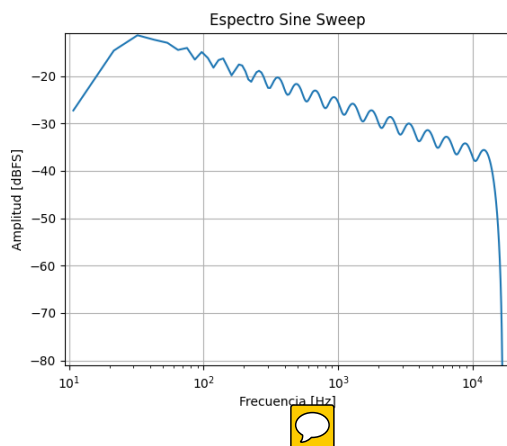


Figura 7: Espectro del Sine Sweep Logarítmico.

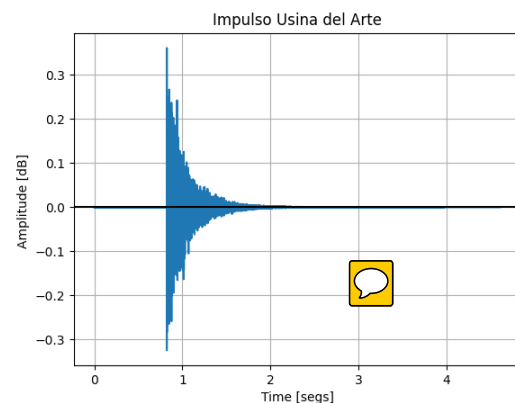


Figura 9: Impulso grabado en la Usina del Arte.

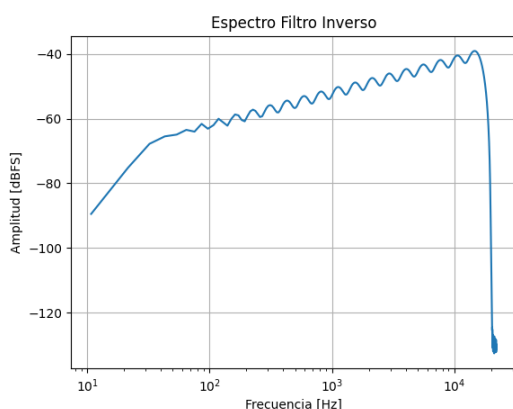


Figura 8: Espectro del Filtro Inverso.

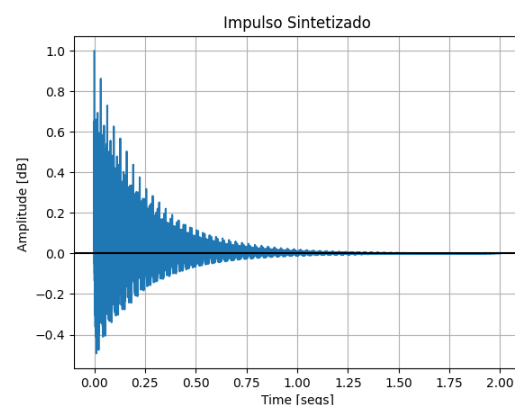


Figura 10: Impulso Sintetizado.

#### 4.1.3. Respuesta al impulso

Para sintetizar la respuesta al impulso se utilizó como base el T60, medido en bandas de octava, de la Usina del Arte, provistos por la siguiente [página web](#). Graficando el desarrollo de ambas señales en el tiempo se puede apreciar que en primera instancia el comportamiento de la señal sintetizada es prácticamente idéntico a su contraparte real:

La mayor diferencia entre ambos gráficos son las amplitudes, una interpretación para esta diferencia es que con la señal grabada el micrófono está a cierta distancia de la fuente sonora, por lo que la energía llega en cierta medida disipada, por otro lado, con la señal sintetizada no ocurre esto, se podría interpretar como si la fuente sonora y "el micrófono" estuviesen juntos, además de que no hay disipación ni absorción ya que no hay recinto que las aplique a la señal. Siguiendo la comparativa, en la figura 11 y 12 se pueden apreciar los espectros de ambas señales:



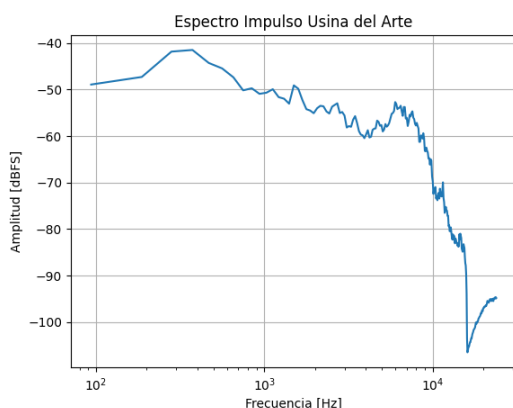


Figura 11: Espectro del impulso grabado en la Usina del Arte.

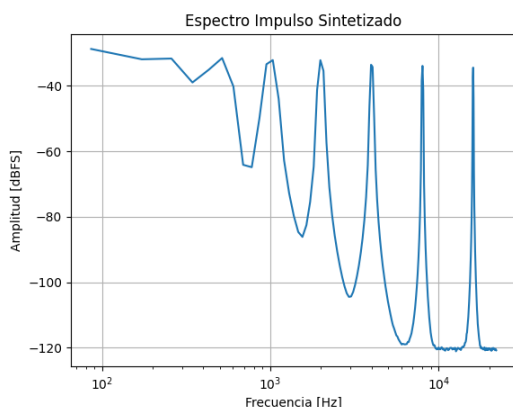


Figura 12: Espectro del Impulso Sintetizado.

De manera similar al desarrollo en el tiempo, se puede ver la diferencia de niveles, sobretodo en las altas frecuencias, se puede seguir la misma interpretación que en el caso anterior, al no haber recinto no hay absorción de altas frecuencias, por lo que estas no decaen. Además, están claramente marcadas la mayoría de las frecuencias centrales con las que se sintetizó el impulso y también se nota la caída de las frecuencias que no están incluidas debido a la división por bandas de octava, es probable que sintetizando el impulso en tercios de octava el espectro del impulso resulte más plano.

## 4.2. Procesamiento de señales

### 4.2.1. Obtención de la respuesta al impulso

Para probar la función de obtención de la respuesta al impulso se generó un sine sweep + filtro inverso de 5 segundos de duración, con ancho de banda de 20-20000 Hz y se grabó la respuesta

con la función de adquisición + reproducción durante 7 segundos. Cabe destacar que la reproducción y la grabación se realizaron con los parlantes y micrófonos integrados en una notebook Lenovo Legion Y520, por lo que el equipo utilizado no está pensado para medición. A continuación en las figuras 13 y 14 se pueden ver el desarrollo en el tiempo y el espectro del impulso, respectivamente:

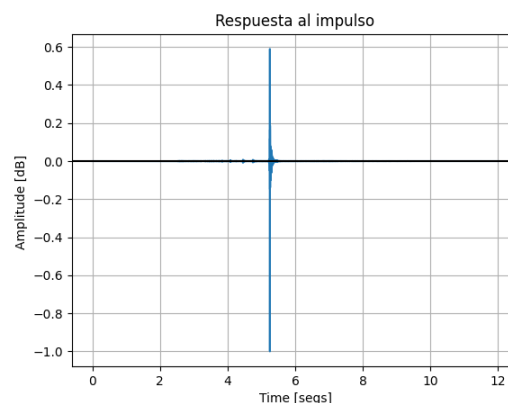


Figura 13: Respuesta al impulso obtenida con el software.

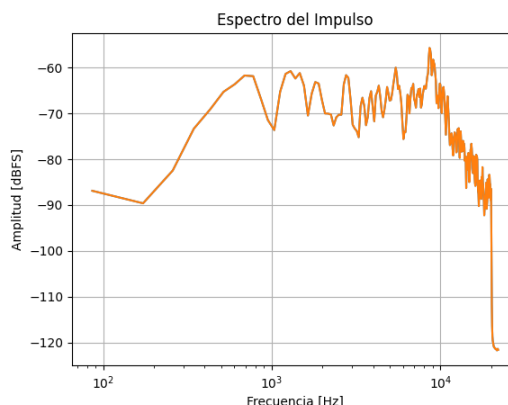
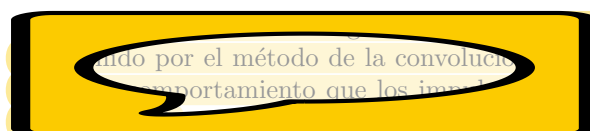


Figura 14: Espectro de la respuesta al impulso obtenida con el software.



volución el archivo de audio resultante termina con una duración de casi el doble de la original, esto se debe al método en sí que alarga la señal por el corrimiento que se realiza para solapar ambas señales, por lo que termina alargando la señal final antes y después del solapamiento entre ambas señales originales. Escuchando el archivo de audio resultante se puede claramente escuchar el impulso, el sonido



se podría describir como un leve aplauso o un click. Analizando el espectro se nota que es muy parecido al espectro del impulso de la Usina del Arte, exitando de manera bastante pareja todas las frecuencias, aunque las menores a 100 Hz tienen menos carga energética, es probable que la respuesta en frecuencia del micrófono con el que se grabó haya tenido influencia en el resultado final, aún así cumple con los requerimientos espectrales para un impulso, por lo que los resultados son satisfactorios.

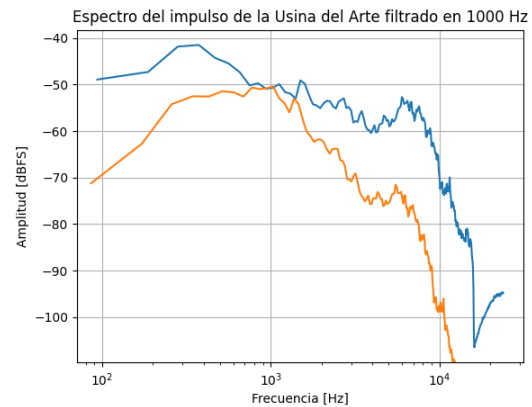


Figura 16: Comparación del espectro del impulso grabado con el filtrado en 1 KHz.

#### 4.2.2. Filtrado según norma IEC-61260

Para probar esta función se utilizó el archivo de audio de la respuesta al impulso de la Usina del Arte, se utilizó un filtro de primer orden en 1000 Hz, la señal desarrollada en el tiempo queda como se ve en la figura 14, con la señal original en azul y la filtrada en naranja:

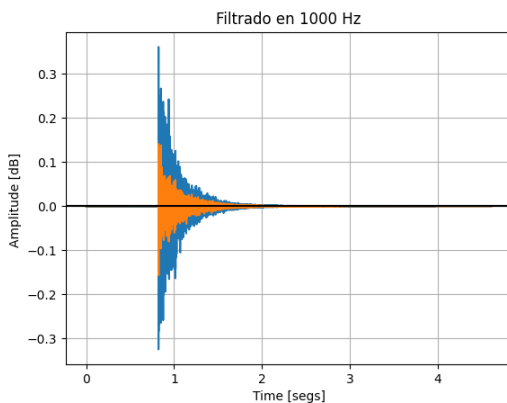


Figura 15: Comparación del impulso grabado con el filtrado en 1 KHz.

Esuchando el audio se percibe claramente la diferencia de nivel y de timbre. En la figura 15 se puede ver el espectro de ambas señales, siendo la curva azul el espectro original y el naranja el filtrado:

Se puede apreciar que en los 1000 Hz los niveles del espectro se superponen, por lo que el funcionamiento del filtro pasabanda es satisfactorio. Por otro lado, en la figura 17 se puede ver la respuesta del filtro, al ser de primer orden la curva es bastante suave y los niveles de atenuación son relativamente bajos.

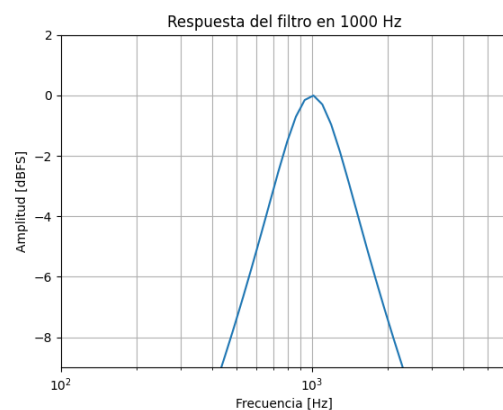


Figura 17: Respuesta del filtro.

#### 4.2.3. Suavizado de señal

Se utilizó el mismo archivo de audio filtrado en 1 KHz que en la sección anterior para probar las funciones de suavizado. El primer proceso es la transformada de Hilbert, que solo deja la envolvente de la señal, en la figura 18 se puede apreciar como solo queda la parte superior de la señal de la figura 9:

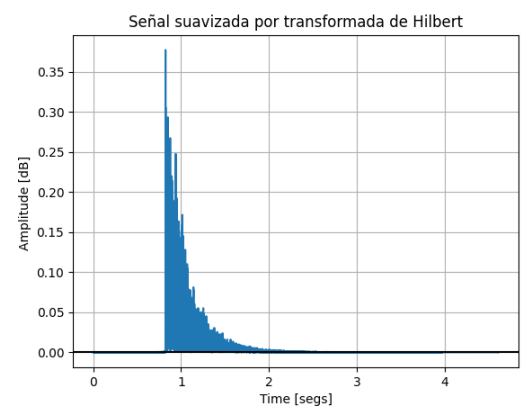


Figura 18: Señal resultante de la transformada de Hilbert.

Luego de pasar por la transformada de Hilbert la señal pasa por la función de la integral de Schroeder y además de suavizar la señal la pasa a dBFS, como se puede ver en la figura 19.

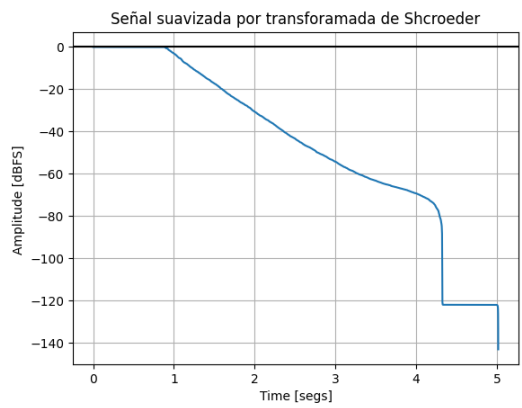


Figura 19: Señal resultante de la transformada de Schroeder.

Una vez suavizada la señal esta pasa por la función de regresión lineal resultando en la recta que se puede ver en la figura 20, con el vector de la imagen de la recta se procede a pasar por las funciones de cálculo de parámetros acústicos.

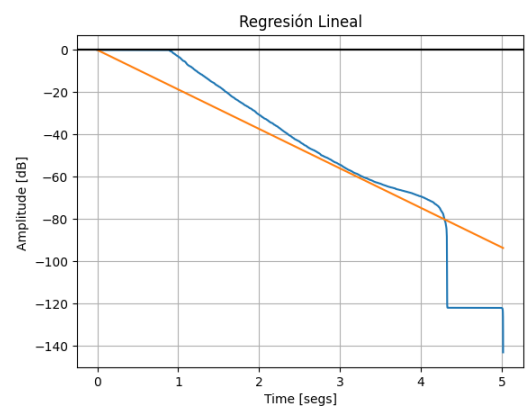


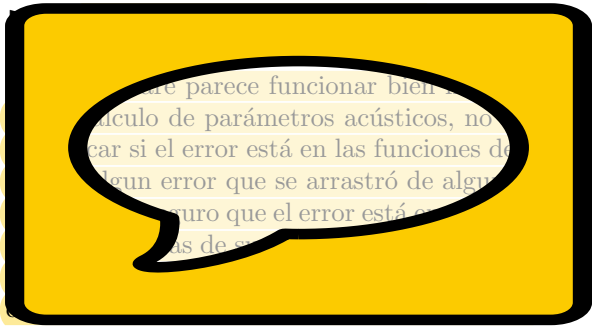
Figura 20: Recta resultante de la regresión lineal vs señal original.

4.3. Cálculo de parámetros acústicos

Las funciones de parámetros acústicos no cumplen con los resultados esperados, el D50 y C80 dan 0 y -52,82 dB para todas las frecuencias. Por otro lado los valores de T60, calculado a partir del T30, y el EDT son los siguientes:

Banda de octava (Hz)	31.5	63	125	250	500	1000	2000	4000	8000	16000
T60 (secs)	7.61	6.54	5.04	3.9	3.4	3.21	3.25	3.57	3.94	5.1
EDT (secs)	1.27	1.09	0.84	0.65	0.57	0.54	0.54	0.6	0.66	0.85

Figura 21: Tabla de resultados.



Referencias

[1] *Iso 3382:1997. acoustics - measurement of room, acoustic parameters.* 1997.

[2] IEC-International Electrotechnical Commission et al. *IEC 61260-3.* 2016.

[3] Angelo. Farina. *Simultaneous Measurement of Impulse Response and Distortion With a Swept-Sine Technique.* Dipartimento di Ingegneria Industriale, Universit'a di Parma., 2000.

- 
- [4] S. Wang L. Hayes. Q. Meng, D. Sen. *Impulse Response Measurement With Sine Sweeps and Amplitude Modulation Schemes*. School of Electrical Engineering and Telecommunications, The University of New South Wales, Sydney, Australia.
- [5] M. R. Schroeder. *New Method of Measuring Reverberation Time*. Bell Telephone Laboratories, Inc. Murray Hill, New Jersey, 1964.