

Desarrollo de Software para el Cálculo de Parámetros Acústicos ISO 3382

Nicolás Cavasso¹, Franco Gazzoti², Ezequiel Bordone³, Gustavo Heisser⁴

22 de junio de 2021

Señales & Sistemas, Ingeniería de sonido

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

¹ cavasso46095@estudiantes.untref.edu.ar, ² gazzotti10@gmail.com, ³ ezequielbordone@gmail.com, ⁴ gustavoheisser@gmail.com

Resumen

El siguiente trabajo utiliza Python como lenguaje de programación, en el cual se busca crear una interfaz desarrollada en módulos, que permitan generar señales como ruido rosa, Sine-Sweep logarítmico con su respectivo filtro inverso, y respuesta al impulso a través del parámetro TR60. Se busca poder realizar procesamiento de estas señales, con la finalidad de poder estimar parámetros acústicos a partir de la norma ISO 3382 (UNE-EN ISO 3382, 2010). Para la estimación del nivel de sonido en estado estable en una habitación a partir de su tiempo de reverberación, es apropiado utilizar la primera parte de la descomposición. La relación señal-ruido es a menudo un problema en las mediciones de campo y, a menudo, es difícil o imposible obtener un rango de evaluación de más de 20 dB. Es por ello que se utilizan parámetros como el Rt10, Rt20 y el Rt30.

Keywords: ISO 3382, IR

1. Introducción

Bajo la normativa ISO 3382 se decide caracterizar acústicamente cierto recinto. Se decide desarrollar mediante Python un código que nos permita implementar distintas funciones. Dicho lenguaje nos permite reproducir y grabar la respuesta impulsiva del recinto anteriormente mencionado. Se hace uso de ruido rosa, un barrido de frecuencias sine-sweep y un filtro inverso. Conseguida la respuesta impulsiva se calculan diversos parámetros acústicos contemplados por dicha normativa, por ejemplo EDT, C80, T60, T30, T20, T10 y D50. [1]

2. Marco Teórico

2.1. Ruido Rosa

El ruido rosa puede definirse como una presión sonora caracterizada por su densidad espectral inversamente proporcional a la frecuencia f , $1/\sqrt{f}$. Por lo general es útil implementarlo para calibrar fuentes sonoras. Otra de sus cualidades es que posee igual energía por octava, por lo cual posee un decaimiento de 3 dB por octava. [3]

2.2. Sine-Sweep

El sine-sweep logarítmico es un barrido sinusoidal, en el cual la frecuencia varía exponencialmente respecto del tiempo. Se define con la siguiente expresión:

$$x(t) = \sin \left[\frac{T\omega_1}{\ln(\frac{\omega_2}{\omega_1})} (e^{\frac{t \ln(\frac{\omega_2}{\omega_1})}{T}} - 1) \right] \quad (1)$$

Donde t es la variable tiempo en segundos, T es la duración total del barrido y ω_1 y ω_2 las frecuencias angulares iniciales y finales respectivamente.

2.3. Filtro Inverso

Un filtro inverso es la inversión respecto del tiempo de una señal, de forma que la frecuencia instantánea disminuye respecto del tiempo. Se compensa en el caso logarítmico la diferencia de energía entre las distintas frecuencias.

$$k(t) = m(t)x(-t) \quad (2)$$

$$m(t) = \frac{\omega 1}{2\pi\omega(t)} \quad (3)$$

$$\omega(t) = \frac{d[\theta(t)]}{dt} = \frac{K}{L} e^{\frac{t}{L}} \quad (4)$$

2.4. Respuesta al Impulso

Es la respuesta de un sistema ante una señal de entrada. Dicha señal contiene información sobre el comportamiento del sistema en si, lo que nos permite calcular la señal de salida del sistema ante cualquier señal de entrada. [4] Podemos obtenerla mediante:

$$y_i = A_i e^{(\pi_i t)} \cos(2\pi_i t) \quad (5)$$

$$\pi_i = \frac{-\ln(10^{-3})}{T_{60}} \quad (6)$$

$$y = \sum_{i=1}^n y_i \quad (7)$$

2.5. Tiempo de Reverberación

El tiempo de reverberación de una señal se define como el tiempo requerido en el que decaen las reflexiones de un sonido en un recinto. Según la norma ISO 3382 se define al T60 como el tiempo requerido para que el nivel de presión sonora decaiga 60 dB posterior a que la fuente deje de emitir sonido. [2]

2.6. Parámetros Acústicos

■ EDT (Early Decay Time)

Corresponde al tiempo de reverberación temprano y se considera que refleja mejor cómo percibimos la reverberación en un recinto. Es el tiempo que la señal tarda en decaer de 0 dBFS a -10dBFS. Un tiempo corto de EDT es un buen indicador de la claridad del habla.

■ C₈₀

También conocido como índice de claridad, es la relación entre las energías acústicas precoces y tardías. En este caso en particular nos interesa aquel cuyo tiempo de retardo son 80 ms [milisegundos]. Se define como:

$$C_{80} = \frac{\int_{0ms}^{80ms} p^2(t) dt}{\int_{80ms}^{\infty} p^2(t) dt} dB \quad (8)$$

■ D₅₀

Tomando la definición de C_{te} en 50 ms, el D₅₀ puede definirse como la relación de energía dentro de los primeros 50 ms frente a la energía general del sonido. Se define como:

$$D_{50} = \frac{\int_{0ms}^{50ms} p^2(t) dt}{\int_0^{\infty} p^2(t) dt} dB \quad (9)$$

■ T₆₀

Se define el T60 como el tiempo requerido para un decrecimiento de 60 dB (en dBFS es de -5dB a -65 dB). Al requerirse un piso de ruido muy bajo para que se de el decaimiento, se suele utilizar el T10, T20 o T30 y luego se realiza una extrapolación para obtener el T60. Según la ISO 3822 el T30 es de -5dB a -35 dB, el T20 de -5dB a -25dB y el T10 de -5dB a -15dB.

3. Desarrollo

La implementación del código se puede seguir utilizando el [siguiente diagrama de flujo](#).

3.1. Ruido Rosa

En una primer etapa, se utilizará ruido rosa para excitar el recinto a medir. Como buscaremos medir, por ejemplo, el t30, según la ISO 3382 es necesario que el ruido rosa este al menos 45 dB por encima del nivel de ruido de fondo. La función toma como parámetros de entrada la duración y la frecuencia de muestreo. Además cuenta con la posibilidad de grabación y reproducción de audio mediante la librería soundfile, por lo que al ser llamada crea un array y un archivo .wav bajo el nombre de "pink_noise.wav". Para llevar a cabo el diseño del mismo se utilizó la técnica de filtrado de ruido blanco 1/f

3.2. Sine-Sweep y Filtro Inverso

La implementación en el caso del sine sweep se basó en la fórmula [1] y en el caso del filtro inverso en la [2], [3] y [4], mediante la librería numpy. Se debe llamar ingresando la duración y la frecuencia inferior y superior. La función al ser ejecutada, genera dos arrays y graba dos archivos wav bajo el nombre de "sine_sweep.wav" "filt_inv.wav".

3.3. Respuesta al Impulso mediante Convolución

Luego mediante el método propuesto por Farina, se creó la función “IR” que consiste en realizar la transformada rápida de Fourier del sine-sweep y el filtro inverso, para luego convolucionarlos y obtener la respuesta al impulso. En este caso el código se implementó utilizando la función de la librería `scipy fftconvolve` que recibe los arrays, calcula el FFT y convolucionan, siendo esta una buena técnica que optimiza el uso de recursos computacionales. La función IR recibe como entrada los arrays del sine sweep y el filtro inverso, y devuelve un .wav llamado “IR_ht.wav”, correspondiente a la respuesta impulsiva.

3.4. Sintetización de Respuesta al Impulso

En el caso de requerirse, el código cuenta con una alternativa para sintetizar una respuesta al impulso. La sintetización se lleva a cabo mediante las ecuaciones [5], [6] y [7] y la librería `numpy`, además se especifica en el código las frecuencias centrales para cada tipo de filtro según la norma IEC 61260. Como entrada, la función debe recibir si es banda de octava o de tercio de octava, luego mediante el uso de un array se ingresa el T60 por frecuencia central, y la duración requerida de la respuesta al impulso. La función devuelve mediante el uso de la librería `soundfile`, un .wav bajo el nombre de “audio_RI.wav”.

3.5. Función de Carga de Archivos de Audio

El código también cuenta con función “carga_data.wav”, que mediante el módulo de `soundfile` transforma de .wav a array, en el caso de requerirse.

3.6. Función Filtros

Esta función será la encargada de discretizar la respuesta al impulso, en las frecuencias centrales de banda de octava o tercios de octava, definidas por la norma IEC 61620. Como entrada recibe el array correspondiente al audio .wav, las frecuencias centrales que se desean filtrar indicando si es de octava o tercio de octava y el grado de filtro. A la salida entrega un archivo .wav por cada frecuencia central.

3.7. Conversión a Escala Logarítmica Normalizada

Esta función se encarga de convertir la señal ingresada (en este caso el array de la RI) mediante el uso de la siguiente ecuación:

$$R(t) = 20 \log_{10} \left[\frac{A(t)}{A(t)_{max}} \right] \quad (10)$$

y devuelve un array con la señal en escala logarítmica normalizada.

3.8. Suavizado

Las señales una vez procesadas se les aplica la función “Suavizado” con el objetivo de eliminar ruidos y tener un decaimiento más suave. La misma permite la entrada del array del audio a analizar, y el tipo de suavizado. Un tipo es mediante la transformada de Hilbert. Esta, obtiene la llamada función analítica, transformando la señal del dominio real al dominio complejo. Luego lo que nos interesa es el módulo de esta señal compleja, que va a representar la envolvente de la señal. El otro parámetro de la función para la opción de suavizado, es `medfilt`. Aplica el filtro de promedio móvil, que realiza un promedio en L muestras, dando una lista de números las cuales cada uno es el promedio de un subconjunto de los datos originales. En ambos casos se usaron funciones ya de la librería `scipy`.

3.9. Integral de Schroeder

La integral de Schroeder, representa la curva de decaimiento de la energía obtenida, una vez apagada la fuente. Los parámetros derivados de la respuesta al impulso se pueden calcular directamente sobre la integral de Schroeder. Para el cálculo de la integral se utilizó la técnica por aproximación de fórmula de punto medio compuesta. El criterio utilizado para hallar el límite superior de la integral fue el de estimar la duración de la señal, multiplicar la misma por mil para así lograr que tienda a un número considerablemente mayor, en comparación a la duración de la señal original. La misma luego deberá ser convertida a escala logarítmica para continuar con las operaciones.

3.10. Regresión Lineal por Método de Cuadrados Mínimos

Esta función llamada “min_squares” va a recibir como entrada el array de la respuesta al impulso,

previamente pasada por la función de suavizado, y va a entregar una recta donde se podrán calcular los parámetros acústicos correspondientes. El código fue implementado bajo el concepto de minimizar el error de la suma de una función cuadrática. Se buscan obtener los parámetros a_0 y a_1 que son los que hacen mínimo el error. Luego desde un punto de vista algebraico se buscará proyectar ortogonalmente el vector b sobre el subespacio de columnas A . Y mediante el uso de la ecuación normal se podrán obtener los parámetros a_0 y a_1 . Es importante que el array ingresado esté previamente normalizado para que a la salida se obtenga la recta en dBFS, ya que será un requerimiento para calcular los parámetros acústicos.

3.11. Cálculo Parámetros Acústicos

Para el cálculo de los parámetros acústicos se utilizará la regresión lineal. En el caso del t_{60} , se debe ingresar a la función el array y se debe indicar si se desea calcular mediante el uso de T_{10} , T_{20} o T_{30} . La misma busca, por ejemplo en el caso de T_{10} , el tiempo en el que la señal pasa de -5dBFS a -15 dBFS y luego extrapola el valor obtenido para obtener el T_{60} . Para el caso

4. Resultados

Al analizar espectralmente el ruido rosa, se puede observar en la figura 1 una tendencia de decaimiento de 3 dB/oct.

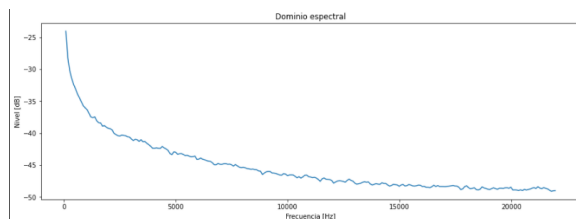


Figura 1: Espectro del ruido rosa

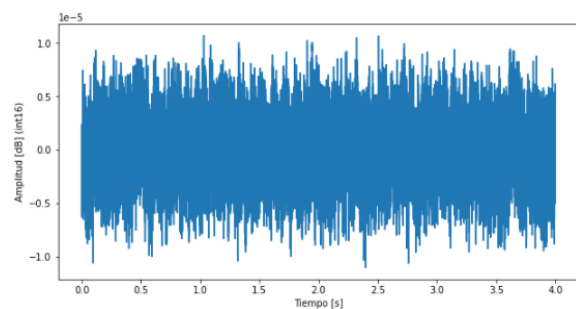


Figura 2: Forma de onda ruido rosa

Con respecto a la respuesta al impulso, en la figura 3 se puede observar la forma de onda de una medición de RI Heslington Church. Y en la figura 4 la RI sintetizada bajo la función "Respuesta_Impulso". Sonoramente se puede percibir una diferencia notable, con un timbre diferente.

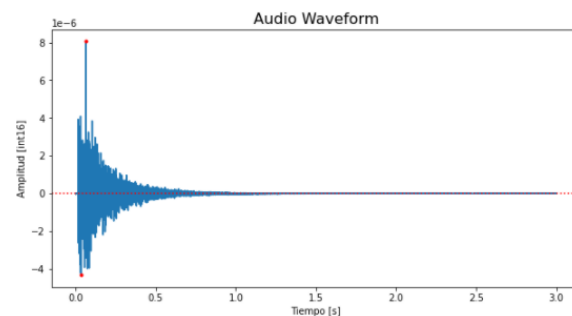


Figura 3: RI Original (Heslington Church)

Valor máximo de amplitud: $8,06e^{-06}$ dB

Valor mínimo de amplitud: $-4,3e^{-06}$ dB

Valor promedio de amplitud : $-1,3e^{-12}$ dB

Para la sintetización del RI se ingresó un array con los siguientes parámetros de T_{60}

Octave Band	31.25 Hz	62.5 Hz	125 Hz	250 Hz	500 Hz	1 kHz	2 kHz	4 kHz	8 kHz	16 kHz
Reverberation Time RT60 T30 (seconds)	3.73	1.28	1.4	1.59	1.75	1.65	1.49	1.14	0.86	0.59

Figura 4: T_{60} por octava de Heslington Church

y se obtuvo lo siguiente:

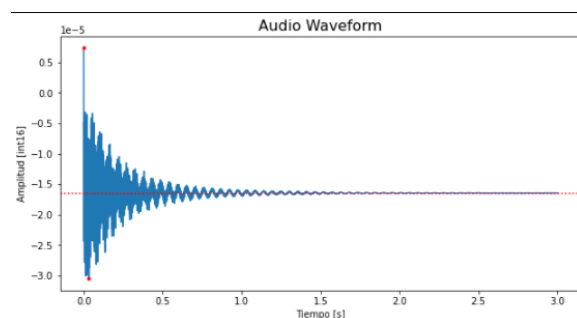


Figura 5: RI Sintetizado a partir del T_{60} de Heslington Church

Valor máximo de amplitud: $7,36e^{-06}$ dB

Valor mínimo de amplitud: $-3,05e^{-05}$ dB

Valor promedio de amplitud: $-1,64e^{-05}$ dB

Luego la respuesta al impulso obtenida mediante la convolución del sine-sweep y el filtro inverso se puede observar en la siguiente figura:

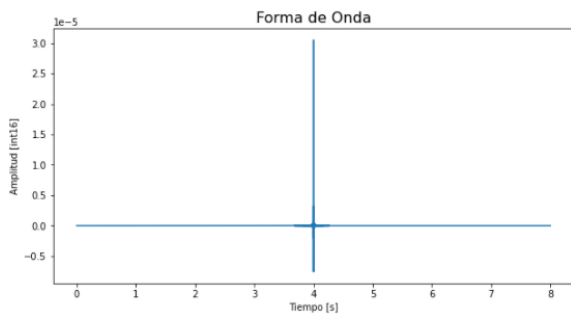


Figura 6: RI mediante convolución

Se puede observar en la forma de onda que se obtiene un impulso muy similar a un delta de Dirac.

En el caso de la función filtros norma IEC 61620, se aplicó sobre el ruido rosa a una frecuencia de 2kHz y se obtuvo lo siguiente:

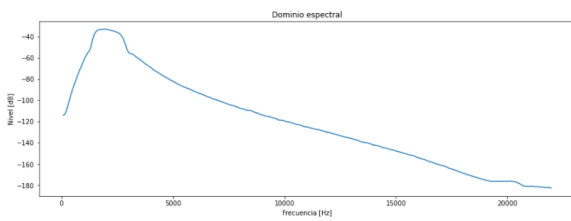


Figura 7: Filtros en ruido rosa a 2kHz grado 4

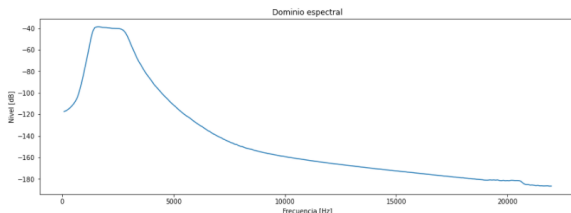


Figura 8: Filtros en ruido a 2kHz grado 7

El resultado de aplicar la función suavizado con hilbert se puede observar en las siguientes figuras:

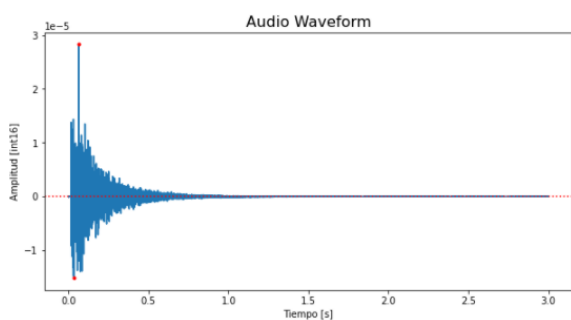


Figura 9: Resultado de suavizado Hilbert en RI de Heslington Church

Valor máximo de amplitud: $2,83e^{-05}$ dB
 Valor mínimo de amplitud: $-1,52e^{-05}$ dB
 Valor promedio de amplitud: $-4,24e^{-12}$ dB

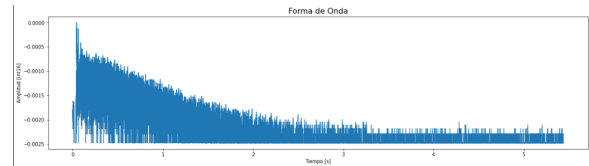


Figura 10: Resultado de suavizado Hilbert en RI de Heslington Church en escala log normalizada

En el caso de suavizado con el filtro promedio móvil obtenemos lo siguiente:

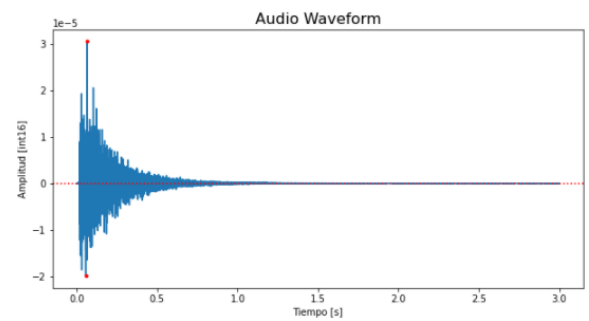


Figura 11: Resultado de suavizado promedio móvil en RI de Heslington Church

Valor máximo de amplitud: $3,05e^{-05}$ dB
 Valor mínimo de amplitud: $-1,99e^{-05}$ dB
 Valor promedio de amplitud: $-1,27e^{-10}$ dB

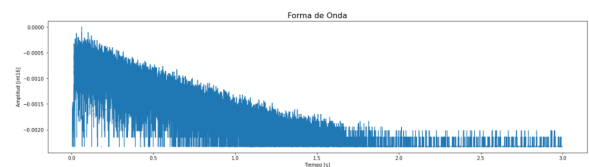


Figura 12: Resultado de suavizado promedio móvil en RI de Heslington Church en escala log normalizada

Con la salida de la señal ya suavizada por el promedio móvil, se procesó con la función integral de Schroeder y se obtuvo lo siguiente:

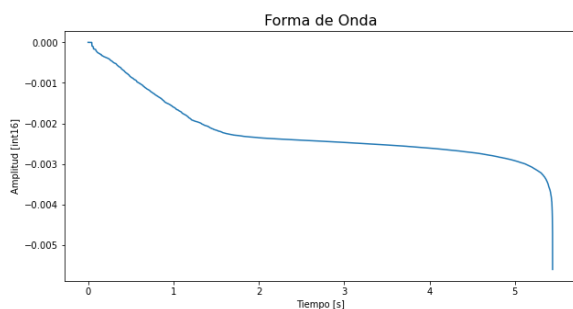


Figura 13: Integral de Schroeder

Finalmente se procesó la salida de la función integral de Schroeder con la función de regresión lineal de cuadrados mínimos y se obtuvo lo siguiente:

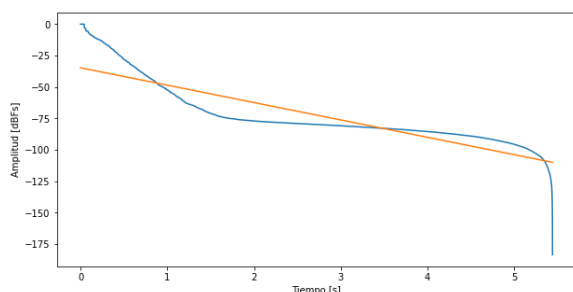


Figura 14: Regresión Lineal de la Integral de Schroeder

La desviación estándar obtenida es $R = 0,87$

5. Conclusiones

En primer lugar, comparando el RI real con el RI sintetizado, se puede concluir que existe un margen de error del 9 % para el máximo, 30 % para el valor mínimo y 21 % para el valor promedio. Además timbricamente, suenan distinto.

En segundo lugar, con el RI a partir de la convolución, se obtiene lo buscado, pero se puede percibir sonoramente que la señal resultante presenta pre-ringing, que son pequeñas oscilaciones que aparecen en la señal al aplicar el filtro inverso, cuando idealmente el impulso resultante debería ser una delta de Dirac ideal.

Después, con respecto a la implementación de los filtros se concluye que a mayor número de grado, mayor será la selectividad del filtro, aunque acarrea una gran distorsión de fase.

Luego partir de los datos obtenidos mediante la implementación de la función `visualizar_datos`, se concluye que los resultados al utilizar `hilbert` o `medfilter`, son distintos aunque no difieren en sonoridad.

Con respecto a la función integral de Schroeder, funciona correctamente, pero la integración numérica de la caída experimental se interrumpe por el ruido de fondo que interviene en la parte final de la caída. Un dato no menor sería en un futuro aplicar la función de Lundebly, la cual permite hallar con mejor precisión la duración de la respuesta al impulso.

el ajuste no es ideal, posiblemente porque debería indicarse que estudie la pendiente inicial y no la totalidad. Usando la función de Lundebly posiblemente la recta se aproxime mejor, ya que permite discriminar el ruido de fondo de la señal a procesar. Finalmente

6. Referencias

1. ISO 3382:2008. Acoustics - Measurement of rooms acoustic parameters
2. Comments on 'New Method of Measuring Reverberation Time' [P. W. Smith, Jr., J. Acoust. Soc. Am. 38, 359(L) (1965)]
3. 1/f noise, L. M. Ward P. E. Greenwood. Scholarpedia, 2007.
4. Miranda A. Millan, D. Galvan. Diseño e implementación de prototipo funcional de reverberador por convolución en tiempo real. Programa de Ingeniería de Sonido, Facultad de Ingeniería, Universidad de San Buenaventura., 2008.
5. I. Simultaneous Measurement of Impulse Response and Distortion With a Swept-Sine Technique, Angelo Farina (2000)
6. https://es.wikipedia.org/wiki/Minimos_cuadrados