



Cátedra
Fundamentos de Acústica y Electroacústica
(Año 2017)

Docente: Ing. Sebastián P. Ferreyra

Trabajo Práctico Nro: 3 (individual)

Alumno: Álvarez, Mariano Legajo: 54976

Tema: Auralización por convolución

Objetivo: Realizar la auralización mediante la convolución de señales anecoicas con respuestas impulsivas monoaurales y binaurales. Comparar los espectros de la señal anecoica y la señal auralizada mediante PSD (Power Spectral Density) y FFT (Fast Fourier Transform). El alumno aprenderá a aplicar técnicas de procesamiento digital de señales en aplicaciones específicas de audio.

Materiales e Instrumental necesarios.

- Señales anecoicas (registradas en campo libre, con instrumentación de medición)
- Respuestas impulsivas monoaurales y binaurales (medidas ó simuladas)
- Software de cálculo numérico matricial (MATLab, Scilab, Octave)
- Computador personal (PC)

Desarrollo.

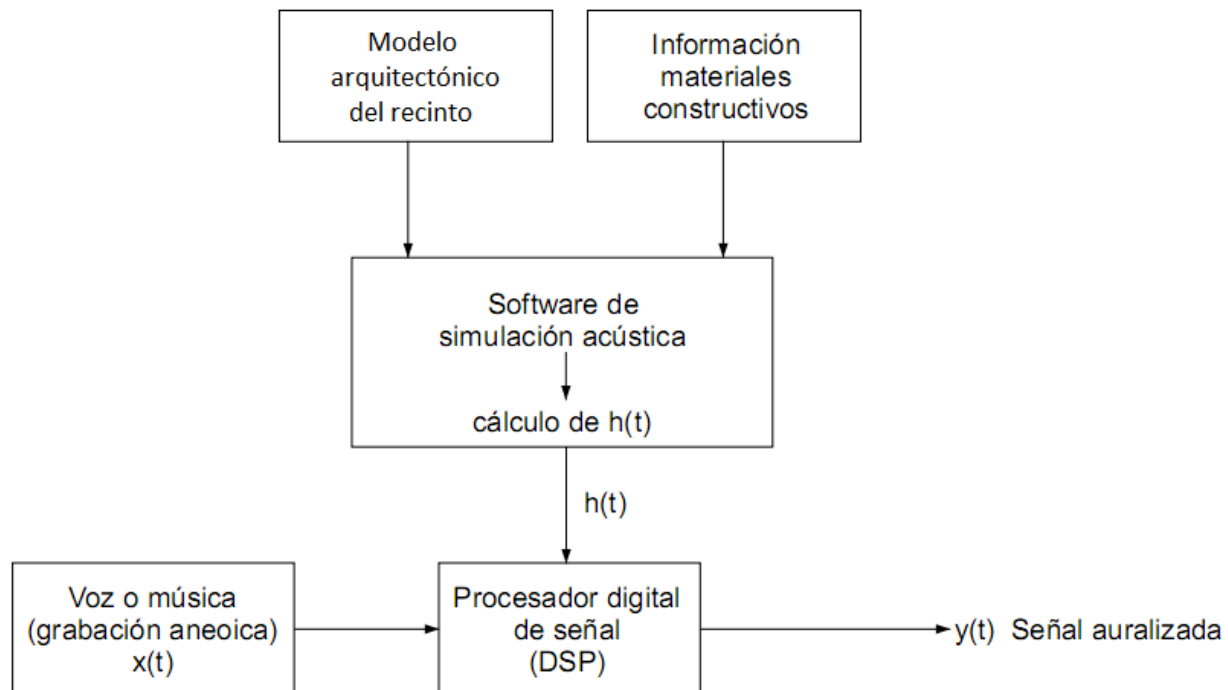
Seleccionamos del sitio web de la cátedra FAYE seis (6) respuestas impulsivas de recintos (3 monoaurales “MRIR” y 3 binaurales “BRIR”) registradas en formato .WAV. Las mismas han sido medidas para diferentes posiciones de la fuente y el receptor, en recintos con diferentes condiciones Acústicas. Las mediciones de MRIR fueron registradas utilizando un micrófono de medición tipo electrostático de ½ pulgada (Brüel & Kjaer, 4134); mientras que para el caso de las BRIR se utilizó un simulador de cabeza y torzo (Brüel & Kjaer, 4128). En ambos casos, se utilizó una fuente sonora omnidireccional tipo dodecaedro.

Las respuestas elegidas fueron: MRIR-A608-FSO1M2-esw.wav, MRIR-CU-F1M2-esw.wav y MRIR-PSCJ-FSO1M3-esw.wav para el caso de las monoaurales y BRIR-SUM NORES-esw-p1.wav, BRIR-SUM NORES-esw-p6.wav y BRIR-SUM NORES-esw-p7-gradas.wav para las binaurales.



A su vez seleccionamos una señal de excitación, del DVD entregado por el profesor de la catedra, la cual es una señal anecoica ya que fue grabada en un campo libre. La señal elegida fue XILOFON.WAV.

Para el procesamiento digital de las señales se utilizó el programa de cálculo matricial MATLAB. Siguiendo como ejemplo el script facilitado por el docente de la catedra. Para obtener la señal auralizada el procedimiento fue el siguiente.



Para simular las señales monoaurales se parte de una MRIR y de la señal de excitación las cuales se convolucionan, mediante un procesador digital de señal, para crear la señal auralizada.

El código para realizar lo anterior es:

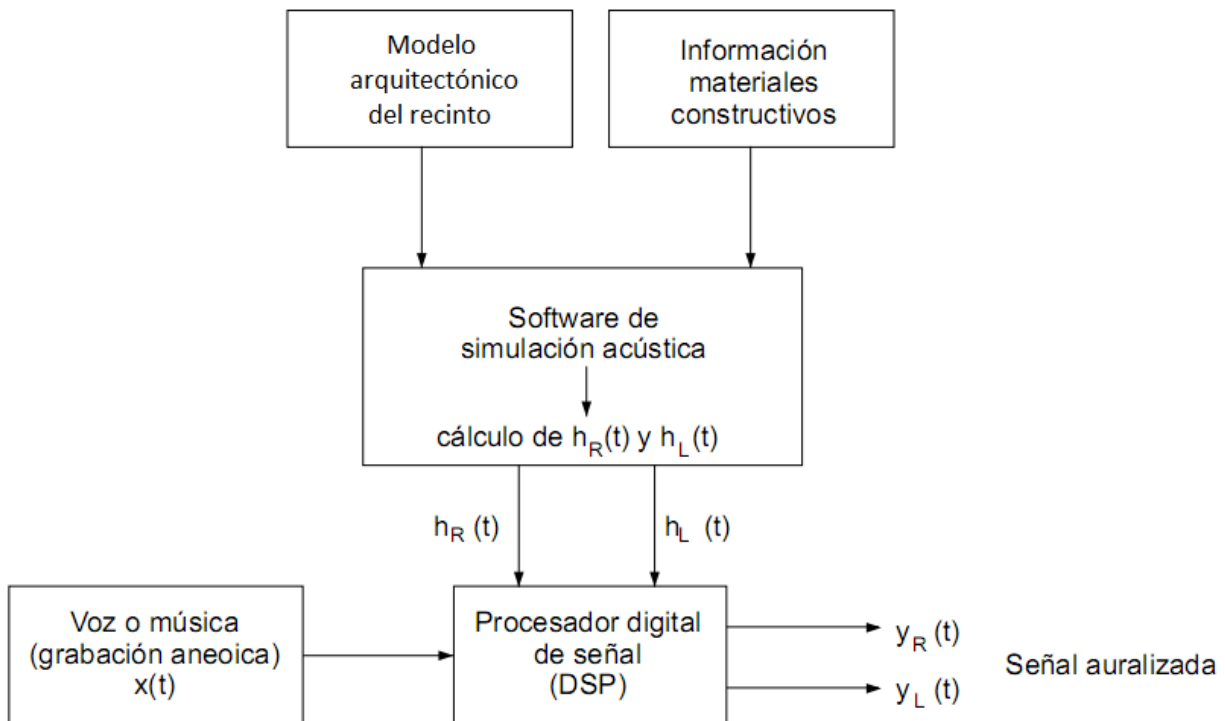
```
clc;
clear;
%%
% Inicialización y convolución

[y1 fs] = audioread('C:\Users\Bill\Documents\Facultad\6to\Acustica\Señales (signals)\Nativas (anecoicas)\XILOFON.WAV'); % Señal anecoica
y1=y1(1:1472940).*0.05;
[y2 fs] = audioread('C:\Users\Bill\Documents\Facultad\6to\Acustica\Tp2\Señales\MRIR-PSCJ-FSO1M3-esw.wav'); % Respuesta impulsiva medida

y3=conv(y1,y2);
audiowrite('XILOFON-MRIR-PSCJ-FSO1M3-esw.wav',y3, fs);
```



Para simular las señales binaurales fue necesario seguir el siguiente procedimiento.



Es similar al de las señales monoaurales solo que en este caso es necesario separar los dos canales de la BRIR para convolucionar la señal de excitación con cada canal por separado y una vez realizado esto se vuelven a unir las dos señales para crear una sola señal binaural.

El código es el siguiente para señales binaurales.

```
clc;
clear;
%%

% Inicialización y convolución
tic

[x fs] = audioread('C:\Users\Bill\Documents\Facultad\6to\Acustica\Señales (signals)\Nativas (anecoicas)\XILOFON.WAV'); % Señal anecoica
x=x(1:1472940).*0.05;
[h fs] = audioread('C:\Users\Bill\Documents\Facultad\6to\Acustica\Tp2\Señales\BRIR-SUM NORES-esw-p6.wav'); % Respuesta impulsiva medida

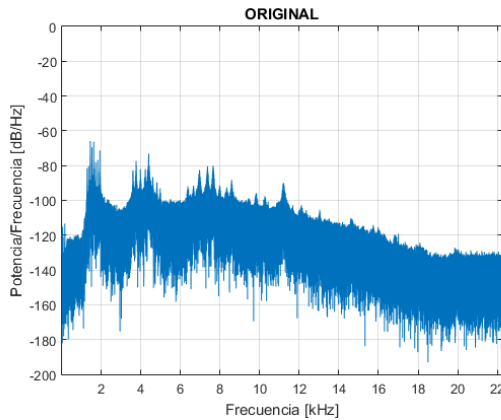
size=length(h);
hr=h(1:size,1);
hl=h(1:size,2);
yr=conv(x,hr);
yl=conv(x,hl);
y=[y1',yr'];

audiowrite('XILOFON-BRIR-SUM NORES-esw-p6.wav',y, fs);
toc
```

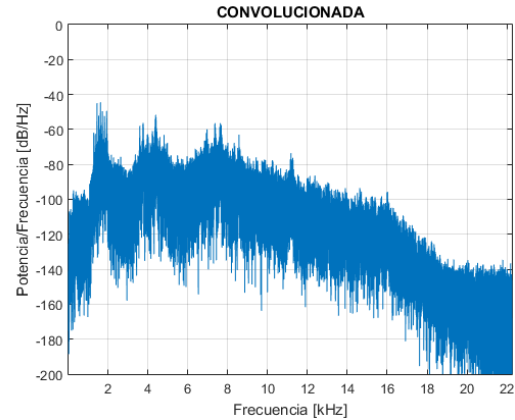


Para cada caso se realizó una comparación entre la señal anecoica y la señal auralizada aplicando la FFT y PSD en donde se pueden ver los cambios en el espectro de frecuencia.

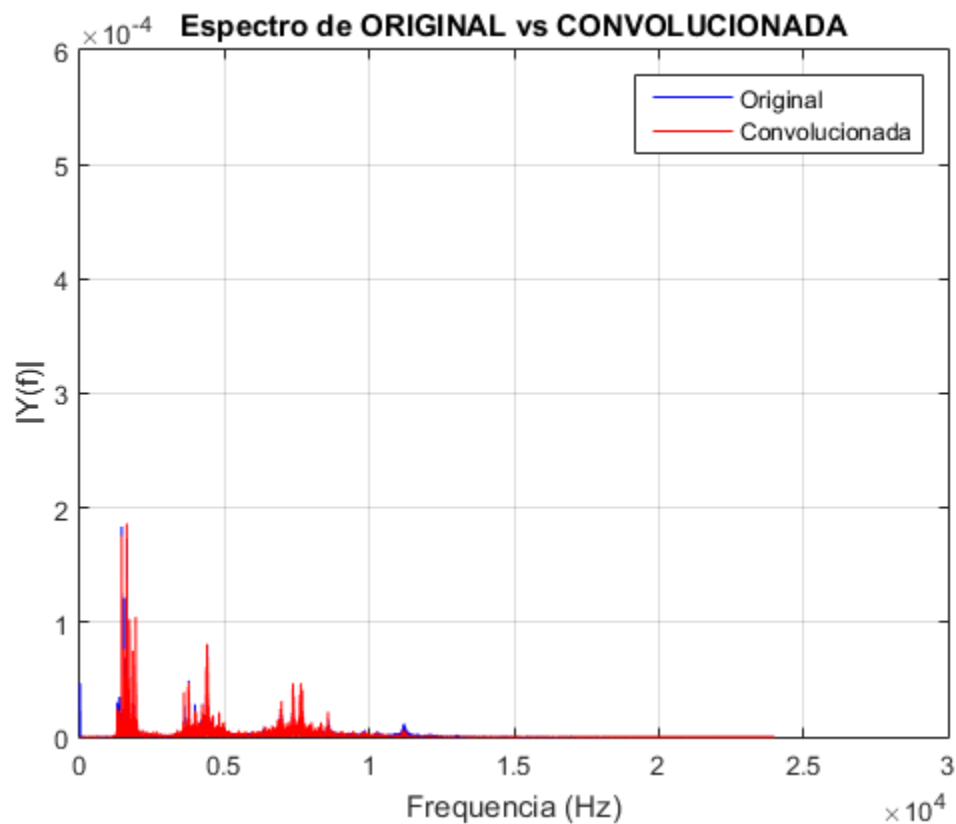
XILOFON-MRIR-A608-FSO1M2-esw



Xil-A608psdOriginal



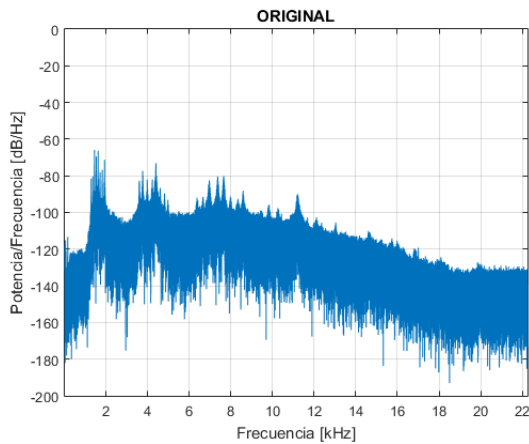
Xil-A608psdConv



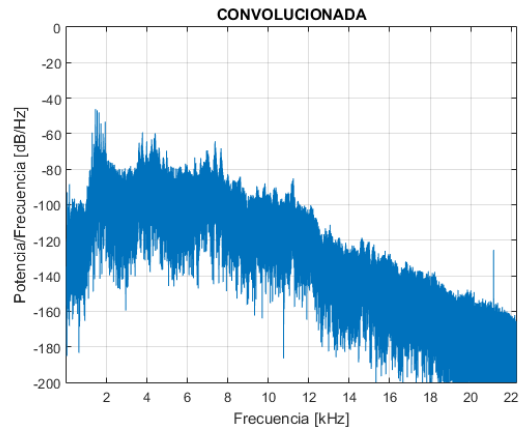
Xil-A608Espectro



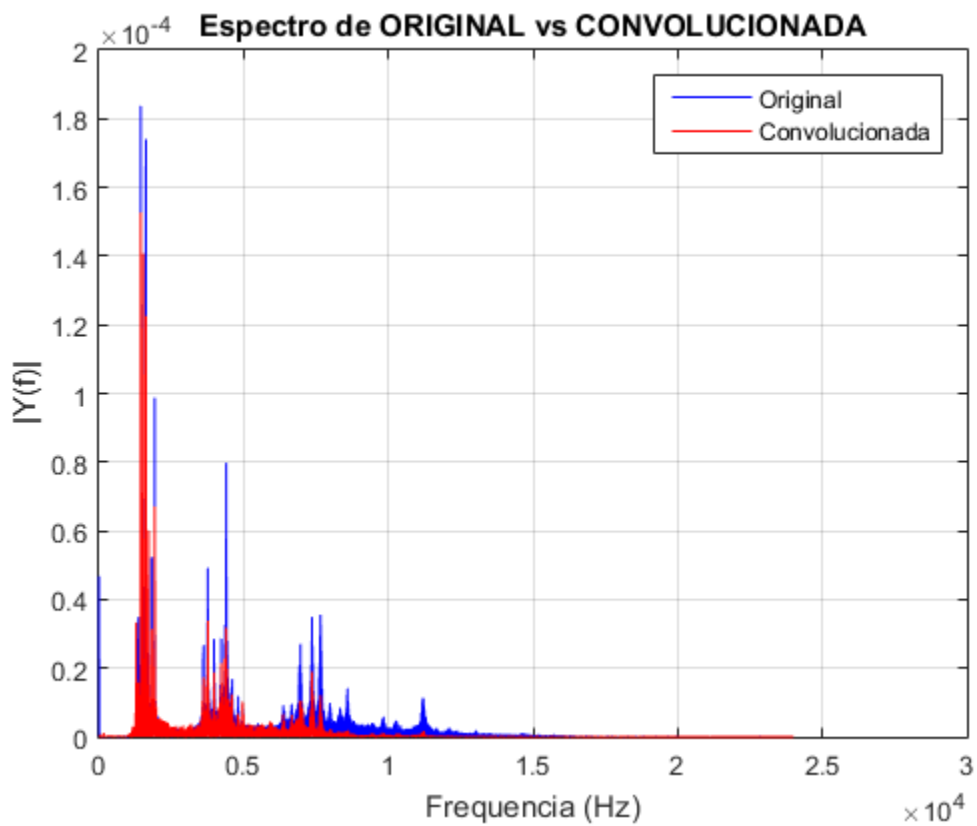
XILOFON-MRIR-CU-F1M2-esw



Xil-CUpsdOriginal



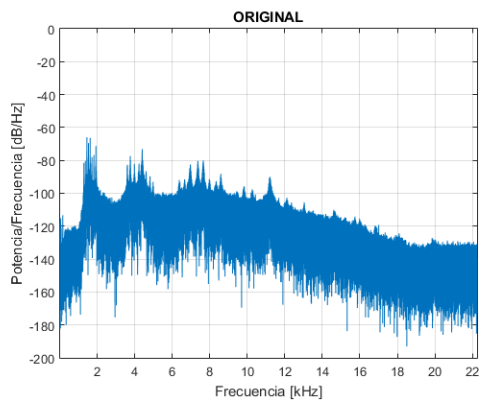
Xil-CUpsdConv



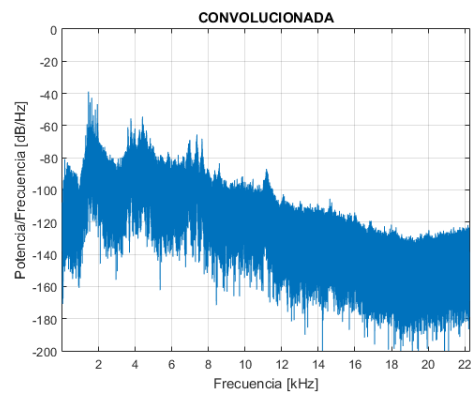
Xil-CUEspectro



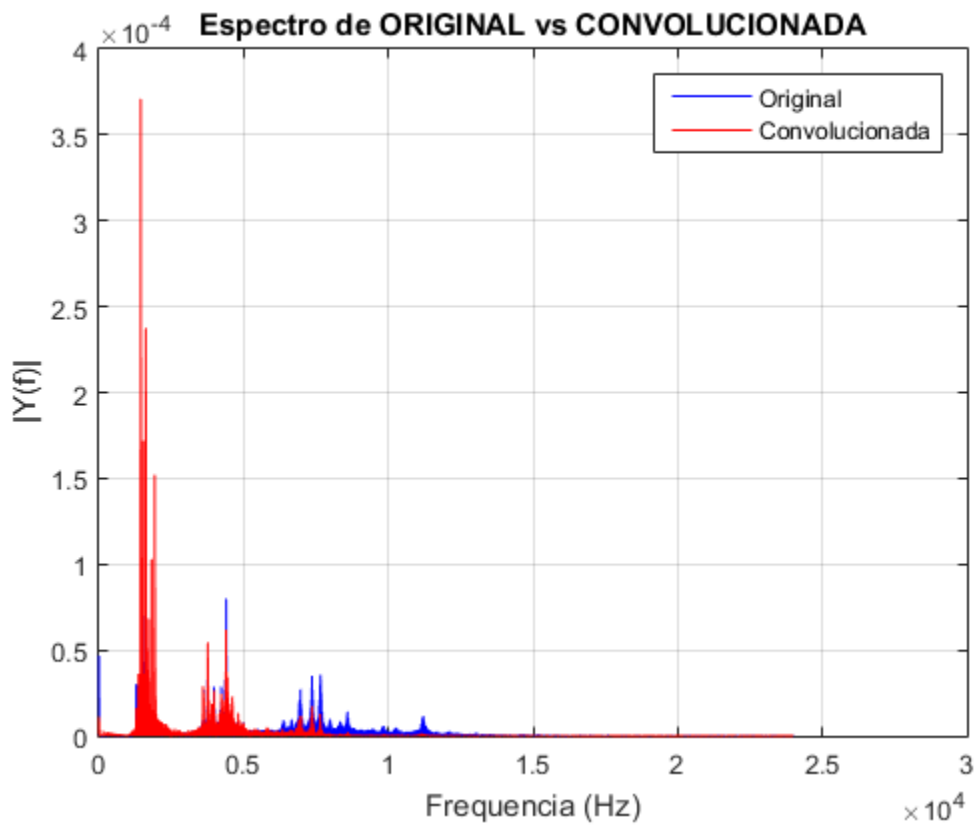
XILOFON-MRIR-PSCJ-FSO1M3-esw



Xil-PSCJpsdOriginal



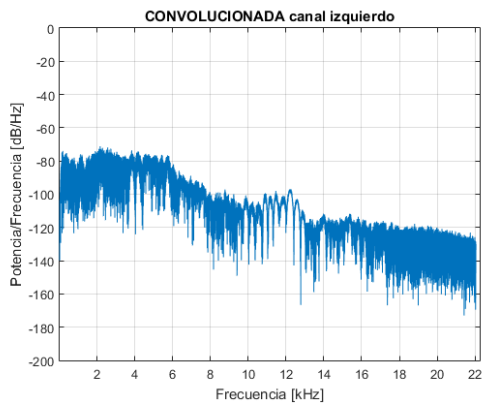
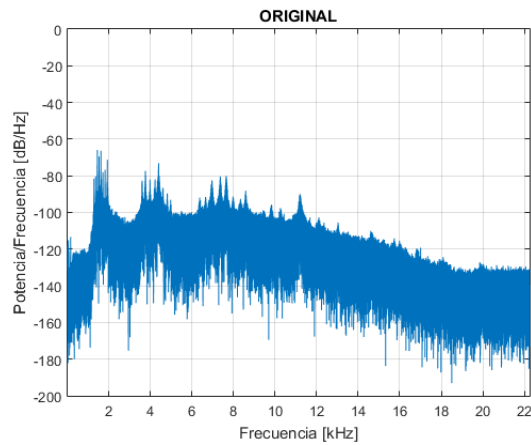
Xil-PSCJpsdConv



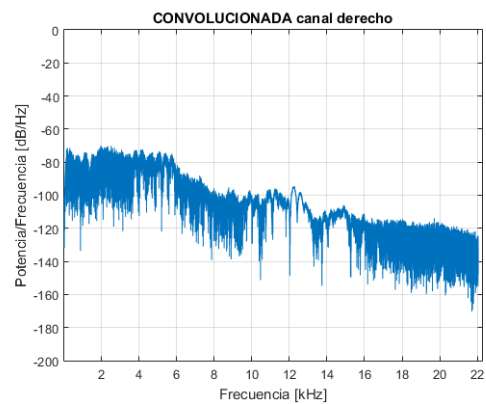
Xil-PSCJEspectro



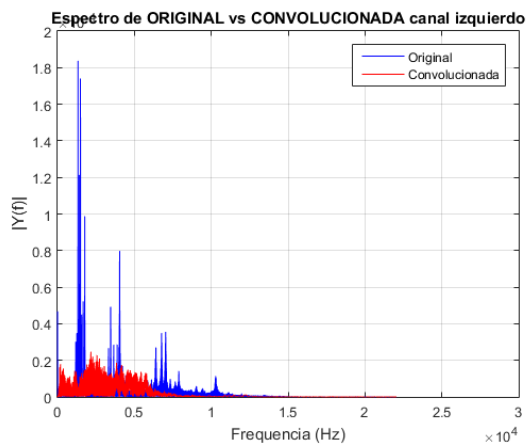
XILOFON-BRIR-SUM NORES-esw-p1



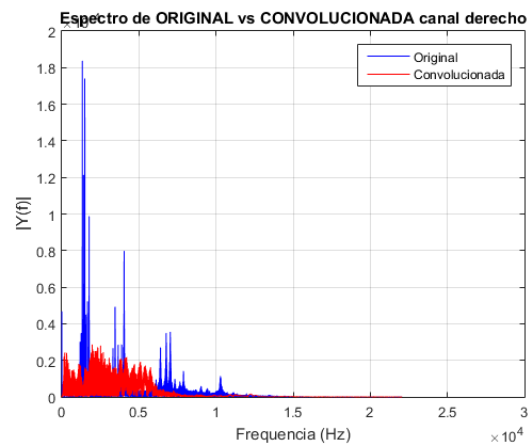
Xil-SUM-p1psdConvCi



Xil-SUM-p1psdConvCd



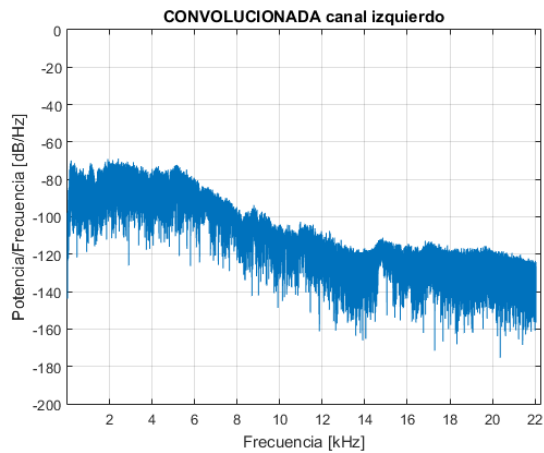
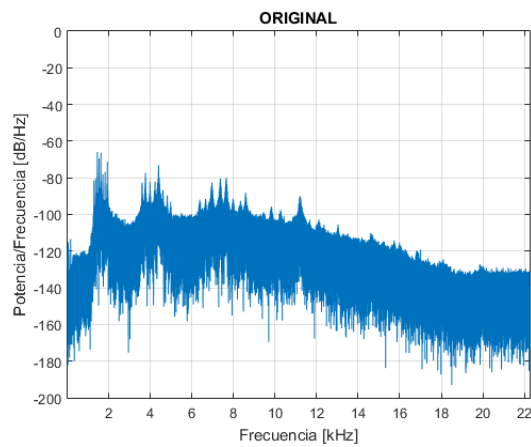
Xil-SUM-p1EspectroCi



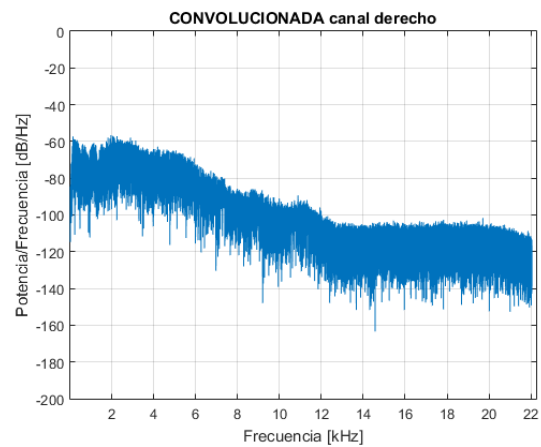
Xil-SUM-p1EspectroCd



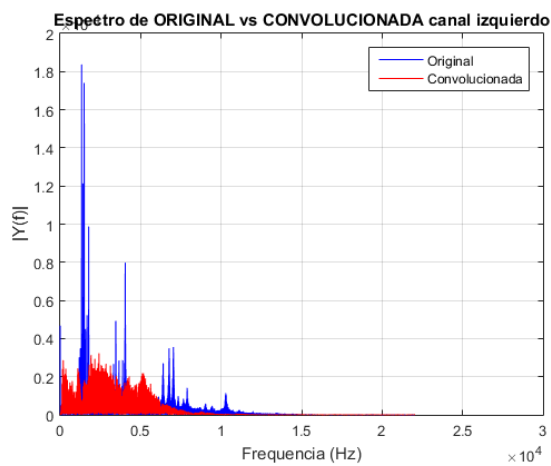
XILOFON-BRIR-SUM NORES-esw-p6



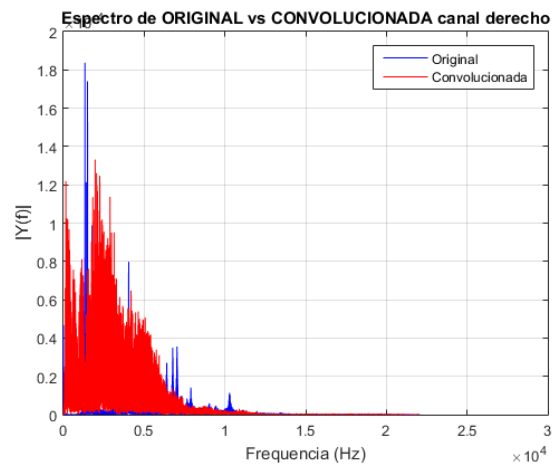
Xil-SUM-p6psdConvCi



Xil-SUM-p6psdConvCd



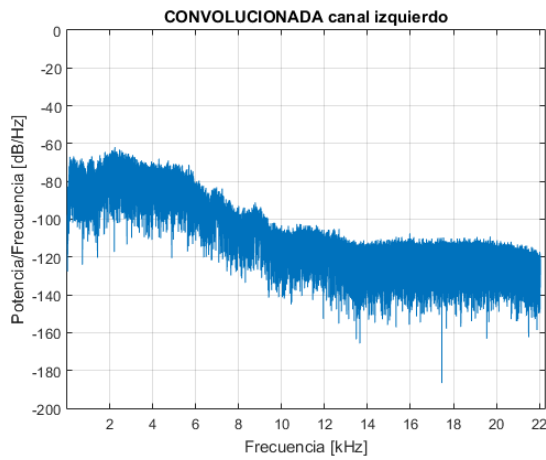
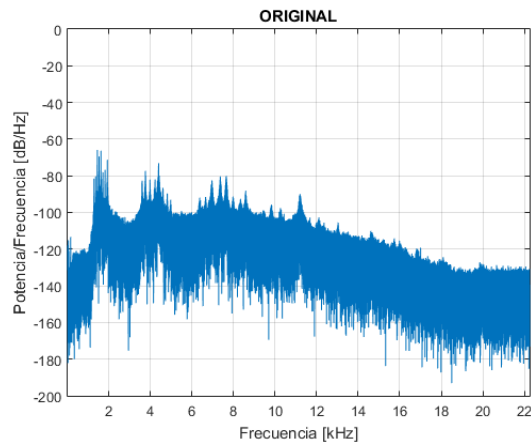
Xil-SUM-p6EspectroCi



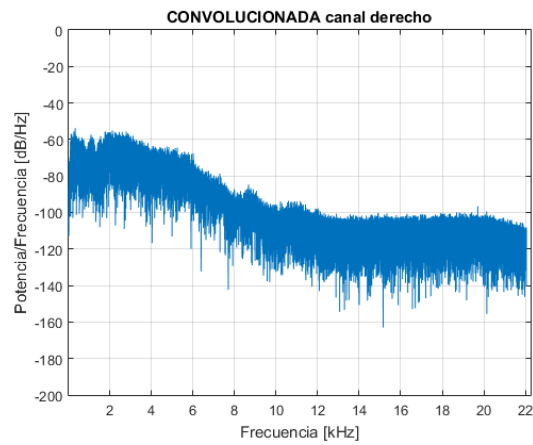
Xil-SUM-p6EspectroCd



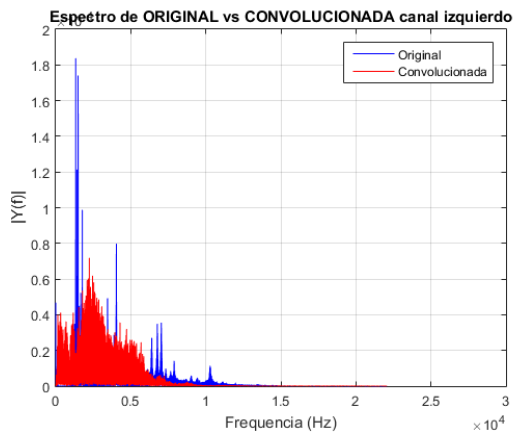
XILOFON-BRIR-SUM NORES-esw-p7-gradas



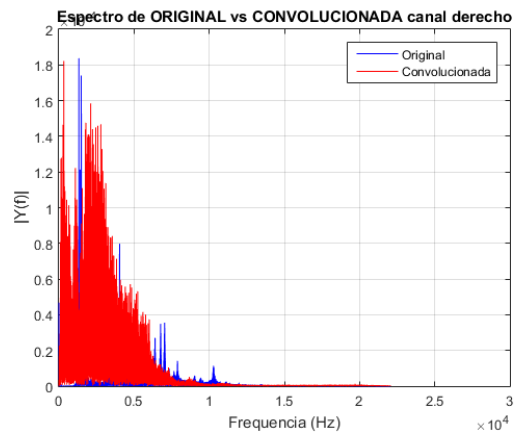
Xil-SUM-p7psdConvCi



Xil-SUM-p7psdConvCd



Xil-SUM-p7EspectroCi



Xil-SUM-p7EspectroCd



Para obtener los gráficos anteriores se utilizó el siguiente script:

```
%%  
% PSD de la señal CONVOLUCIONADA  
  
figure(1)  
nfft1r = 2^nextpow2(length(hr));  
Pxx1r = abs(fft(hr,nfft1r)).^2/length(hr)/fs;  
Hpsd1r = dspdata.psd(Pxx1r(1:length(Pxx1r)/2),'fs',fs);  
plot(Hpsd1r)  
xlabel('Frecuencia [kHz]');  
ylabel('Potencia/Frecuencia [dB/Hz]');  
title('CONVOLUCIONADA canal derecho')  
ylim([-200 0]);  
xlim([0.02 22.25])  
grid on  
  
figure(2)  
nfft1l = 2^nextpow2(length(hl));  
Pxx1l = abs(fft(hl,nfft1l)).^2/length(hl)/fs;  
Hpsd1l = dspdata.psd(Pxx1l(1:length(Pxx1l)/2),'fs',fs);  
plot(Hpsd1l)  
xlabel('Frecuencia [kHz]');  
ylabel('Potencia/Frecuencia [dB/Hz]');  
title('CONVOLUCIONADA canal izquierdo')  
ylim([-200 0]);  
xlim([0.02 22.25])  
grid on  
  
%%  
% PSD de la señal ORIGINAL  
  
figure(3)  
nfft2 = 2^nextpow2(length(x));  
Pxx2 = abs(fft(x,nfft2)).^2/length(x)/fs;  
Hpsd2 = dspdata.psd(Pxx2(1:length(Pxx2)/2),'fs',fs);  
plot(Hpsd2)  
xlabel('Frecuencia [kHz]');  
ylabel('Potencia/Frecuencia [dB/Hz]');  
title('ORIGINAL')  
ylim([-200 0]);  
xlim([0.02 22.25])  
grid on  
  
%%  
% FFT de ambas señales  
  
figure(4)  
L1=length(x); %  
NFFT1 = 2^nextpow2(L1); % fft señal original  
Y1 = fft(x,NFFT1)/L1; % (anecoica)  
f1 = fs/2*linspace(0,1,NFFT1/2+1); %  
  
Lr=length(hr); %  
NFFT3r = 2^nextpow2(Lr); % fft señal convolucionada  
YR = fft(0.1.*hr,NFFT3r)/Lr; % (auralizada)  
f3r = fs/2*linspace(0,1,NFFT3r/2+1); %  
  
Ll=length(hl); %  
NFFT3l = 2^nextpow2(Ll); % fft señal convolucionada  
YL = fft(0.1.*hl,NFFT3l)/Ll; % (auralizada)  
f3l = fs/2*linspace(0,1,NFFT3l/2+1); %  
  
plot(f1,2*abs(Y1(1:NFFT1/2+1)), 'b'); %  
hold on %  
plot(f3r,2*abs(YR(1:NFFT3r/2+1)), 'r'); % gráficos  
hold off %  
axis([0 30000 0 0.2e-3]) %  
legend('Original','Convolucionada') %  
title('Espectro de ORIGINAL vs CONVOLUCIONADA canal derecho')  
xlabel('Frecuencia (Hz)') %  
ylabel('|Y(f)|') %  
grid on %  
  
figure(5)  
plot(f1,2*abs(Y1(1:NFFT1/2+1)), 'b'); %  
hold on %  
plot(f3l,2*abs(YL(1:NFFT3l/2+1)), 'r'); % gráficos
```



```
hold off %  
axis([0 30000 0 0.2e-3]) %  
legend('Original','Convolucionada') %  
title('Espectro de ORIGINAL vs CONVOLUCIONADA canal izquierdo')  
xlabel('Frecuencia (Hz)') %  
ylabel('Y(f)') %  
grid on %
```

Conclusión:

Mediante este trabajo práctico podemos ver uno de los métodos para simular una audición ó escucha, en cualquier punto de un recinto, de un mensaje oral o un pasaje musical, con la particularidad de que ello se lleva a cabo de forma virtual. Este método consiste en obtener la respuesta al impulso de un recinto de una posición, de la fuente sonora y del oyente, dados. Y a la misma convolucionarla con la señal anecoica para generar la señal auralizada. Para lo cual se utilizó la función de MATLAB Conv. A su vez se realizó una comparación de la señal de excitación original y la auralizada mediante el uso de las funciones dspdata.psd y fft de MATLAB. Con la primera podemos obtener un espectrograma con la densidad espectral de potencia la cual representa el aporte de potencia debido a cada componente espectral de la señal. La unidad es el Watt/Hz. Y la segunda sirve para obtener la transformada discreta de Fourier con la cual podemos hacer una análisis del espectro en frecuencia de cada señal y comparando una con la otra.

Referencias

MATLAB r2015a help, The MathWorks, Inc.

Sebastián P. Ferreyra (2013); Fundamentos de Acústica y Electroacústica Cap. IV. Acústica de Recintos .