

LABORATORIO

Luis Ariza Ramos

Erick Barros Escorcia

Universidad del Magdalena, Ingeniería Electrónica,

Colombia

Luisarizaar@unimagdalena.edu.co

Erickbarrosde@unimagdalena.edu.co

Abstract In the following project, he presents the study on the mathematical and algorithmic foundations and the practical application of audio recordings, Fourier series performing their spectrum and the use of filters studied and created by the MATLAB software. The Fourier transform is a mathematical tool that is widely used in relation to digital signal processing, it is implemented in the form of electronic voice recognition devices, so much so that it is very useful when finding the frequencies of each signal. .

Resumen En el siguiente Laboratorio, presenta el estudio sobre los fundamentos matemáticos, algorítmicos y la aplicación práctica de grabaciones de audio, series de Fourier realizando su espectro y la utilización de filtros estudiadas y creadas por el software MATLAB. La transformada de Fourier es una herramienta matemática que tiene como uso muy amplio en lo referente al tratamiento digital de señales, se encuentra implementada bajo la forma de dispositivos electrónicos de reconocimiento de voz, es muy útil al momento de hallar las frecuencias de cada señal.

Key words- Señales, Frecuencias, Series, Pasa banda.

I. DESARROLLO

1. Utilizando Matlab® realizar un sistema de adquisición o cargue de señales de audio.
2. Utilizando la Transformada Rápida de Fourier u otro método, identificar los rangos de frecuencia en los que se encuentran las señales adquiridas.
3. Mezclar las señales adquiridas en caso de ser múltiples y evidenciar su espectro.
4. Utilizando Filter Design de Matlab® u otra herramienta, diseñar (3) o más filtros que permitan pasar las diferentes señales en las bandas de

frecuencia identificadas de manera independiente, por ejemplo, si se trabaja con señales de audio, primero los bajos, luego los medios y finalmente los brillos. (Se recomienda el uso de la función `filter(b,a,x)`), en donde `x` contendrá la mezcla de sonidos

5. Evidenciar gráficamente los resultados deseados

% PROCESAMIENTO DE CONTENIDO MULTIMEDIA

**% Luis Ariza - Erick Barros
% Estudiantes de Ingeniería Electrónica**

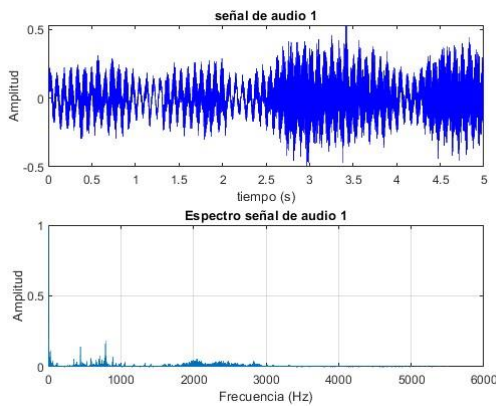
%% Importar Señales de Audio
`clear all; close all; clc;`

`fs = 11025;`

`sn = importdata('grave.mat');
gn = importdata('flauta.mat');
sn = getaudiodata(sn);
gn = getaudiodata(gn);`

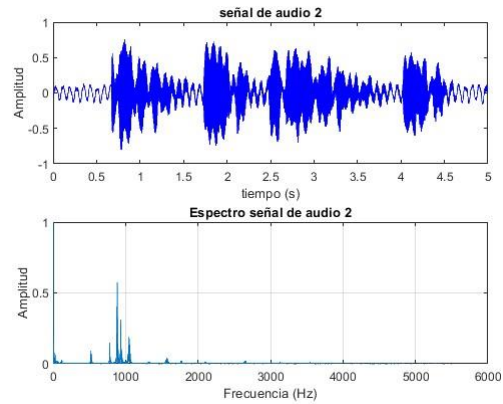
%% Audio 1
% FFT
`transf1=abs(fft(sn));
L=length(transf1);
espectro=transf1(1:L/2);
espectro=espectro/max(espectro);
frec= fs*(1:(L/2))/L;
%
n=length(sn);
t=n/fs;
Ts=1/fs;
tiempo= (0:Ts:(t-Ts))`

```
% grafica
figure(1)
subplot(2,1,1); plot(tiempo,sn,'b');
title('señal de audio 1')
xlabel('tiempo (s)'); ylabel
('Amplitud')
subplot(2,1,2); plot(frec,espectro)
title('Espectro señal de audio 1')
xlabel ('Frecuencia (Hz)');ylabel
('Amplitud'); grid('on')
sound(sn,fs)
```



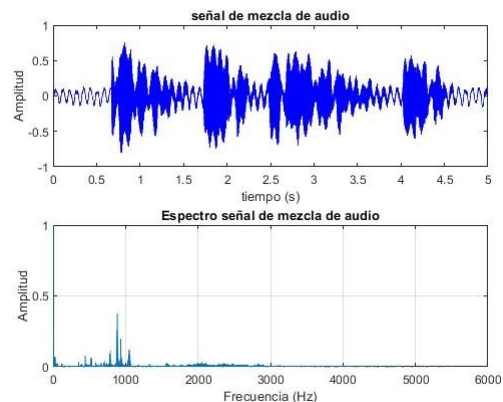
Grafica1. Señal de audio 1

```
%% Audio 2
%FFT
transf1=abs(fft(gn));
L=length(transf1);
espectro=transf1(1:L/2);
espectro=espectro/max(espectro);
frec= fs*(1:(L/2))/L;
%
n=length(gn);
t=n/fs;
Ts=1/fs;
tiempo= (0:Ts:(t-Ts));
% grafica
figure(2)
subplot(2,1,1); plot(tiempo,gn,'b');
title('señal de audio 2')
xlabel('tiempo (s)'); ylabel
('Amplitud')
subplot(2,1,2); plot(frec,espectro)
title('Espectro señal de audio 2')
xlabel ('Frecuencia (Hz)');ylabel
('Amplitud'); grid('on')
sound(gn,fs)
```



Grafica2. Señal de audio 2

```
%% Sumatoria
yn = sn + gn;
%FFT
transf1=abs(fft(yn));
L=length(transf1);
espectro=transf1(1:L/2);
espectro=espectro/max(espectro);
frec= fs*(1:(L/2))/L;
%
n=length(yn);
t=n/fs;
Ts=1/fs;
tiempo= (0:Ts:(t-Ts));
% grafica
figure(3)
subplot(2,1,1); plot(tiempo,gn,'b');
title('señal de mezcla de audio')
xlabel('tiempo (s)');
ylabel('Amplitud')
subplot(2,1,2); plot(frec,espectro)
title('Espectro señal de mezcla de audio')
xlabel ('Frecuencia (Hz)');ylabel
('Amplitud'); grid('on')
sound(yn,fs)
```

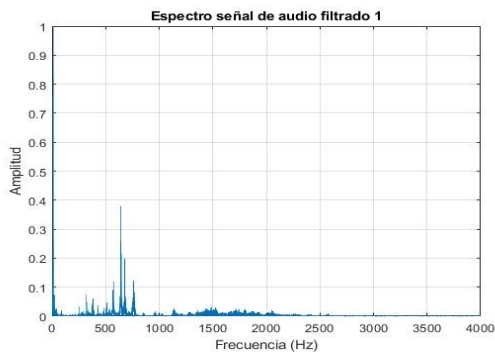


Grafica3. Señal Mezcla de audio

```

%% filtro pasa banda
fs1=8000;
fc1 = [800,1200];
Orden1 = 10;
filtro1 =
fdesign.bandpass('n,f3db1,f3db2',Ord
en1,fc1(1),fc1(2),fs1);
hfiltro = design(filtro1,'butter');
% Filtrado
grave = filter(hfiltro,yn);
% FFT
transf1=abs(fft(yn));
L=length(transf1);
espectro=transf1(1:L/2);
espectro=espectro/max(espectro);
frec= fs1*(1:(L/2))/L;
%grafica
figure(4)
plot(frec,espectro); title('Espectro
señal de audio filtrado 1')
xlabel ('Frecuencia (Hz)'); ylabel
('Amplitud'); grid on;
% Reproducción
grave = 2*grave;
player1 = audioplayer(grave,fs1);
play(player1)

```



Grafica4. Señal filtro 1 de audio

%% Filtro pasa baja

% All frequency values are in Hz.
Fs = 8000; % Sampling Frequency

```

N = 10; % Order
Fstop = 650; % Stopband Frequency
Astop = 80; % Stopband Attenuation (dB)

```

```

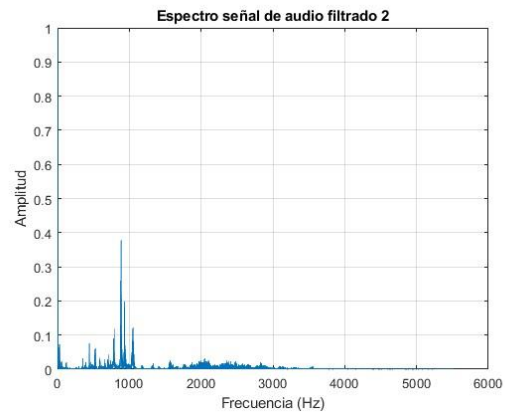
% Construct an FDESIGN object and
call its CHEBY2 method.
h = fdesign.lowpass('N,Fst,Ast', N,
Fstop, Astop, Fs);
Hd = design(h, 'cheby2');

```

```

signalfilt = filter(Hd,yn);
pause(5)
sound(signalfilt,Fs);
% FFT
transf1=abs(fft(yn));
L=length(transf1);
espectro=transf1(1:L/2);
espectro=espectro/max(espectro);
frec= fs*(1:(L/2))/L;
%grafica
figure(5)
plot(frec,espectro); title('Espectro
señal de audio filtrado 2')
xlabel ('Frecuencia (Hz)'); ylabel
('Amplitud'); grid on;

```



Grafica5. Señal filtro 2 de audio

%% filtro pasa alta
Fs = 8000; % Sampling Frequency

```

Fstop = 1700; % Stopband
Frequency
Fpass = 1800; % Passband
Frequency
Astop = 100; % Stopband
Attenuation (dB)
Apass = 1; % Passband
Ripple (dB)
match = 'passband'; % Band to match
exactly

```

```

% Construct an FDESIGN object and
call its BUTTER method.
h = fdesign.highpass(Fstop, Fpass,
Astop, Apass, Fs);
Hd2 = design(h, 'butter',
'MatchExactly', match);

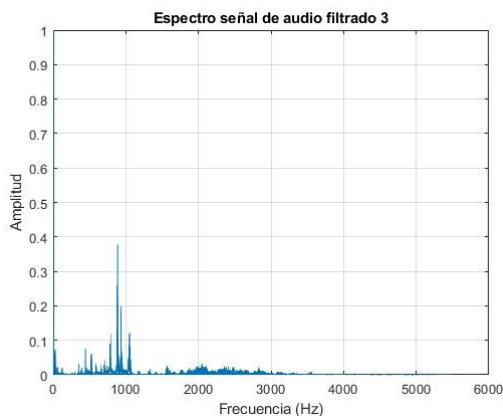
```

% [EOF]

```

signalfilter = filter(Hd2,yn);
pause(5)
sound(signalfilter,Fs);
% FFT
transf1=abs(fft(yn));
L=length(transf1);
espectro=transf1(1:L/2);
espectro=espectro/max(espectro);
frec= fs*(1:(L/2))/L;
%grafica
figure(6)
plot(frec,espectro); title('Espectro
señal de audio filtrado 3')
xlabel ('Frecuencia (Hz)'); ylabel
('Amplitud'); grid on;

```



Grafica6. Señal Filtro 3 de audio

IV CONCLUSIONES

Matlab es un programa eficiente y muy completo que nos permite trabajar con distintas señales y a su vez tiene a su vez múltiples herramientas para hallar la serie de Fourier, transformadas y frecuencias y mucho más, en este caso grabamos tres señales de audio permitiendo grabar reproducir y mezclar más de una señal generada mediante un algoritmo brindado por nuestro tutor, permitiendo desarrollar el objetivo de generar muestras y estas son utilizadas para generar una transformada de Fourier y crear su espectro a su vez luego de estar mezcladas las tres señales debimos distinguir cada señal en tres distintas frecuencias alto, medio y bajo. Es muy satisfactorio conocer investigar y apropiarse del tema de señales mas por lo que son aplicadas en muchos campos, gracias a los algoritmos y clases de nuestro tutor.

V REFERENCIAS

Kamen, E. W., & Heck, B. S. (2008). Fundamentos de Señales y Sistemas Usando la Web y MATLAB (3ra ed.). México.

Proakis J. G. y Manolakis D. G., Tratamiento Digital de Señales, Madrid: Prentice Hall, 2007.

<https://es.slideshare.net/loiju/la-transformada-de-fourier-informe>

Oppenheim, A. V., & Willsky, A. S. (1998). Señales y Sistemas (2da ed.).

The MathWorks Inc. (2017). MathWorks - Makers of MATLAB and Simulink. Retrieved October 9, 2018, from

<https://www.mathworks.com/https://www.dspace.espol.edu.ec/bitstream/123456789/7267/1/D-39012.pdf>

The MathWorks Inc. (2017). MathWorks - Makers of MATLAB and Simulink. Retrieved April