Facultad de Ingeniería

Sistemas Comunicacionales

Profesor Fernando Castillo Rivera

Tarea 1 Sistemas Comunicacionales

Autores:

Tomás Aguayo

Martín Gallegos

Concepción, 2025

Nombre	Coevaluacion %
Martín Gallegos:	100 %
Tomás Aguayo	100 %

Índice

1. Problema 1

Descargue desde el siguiente enlace https://drive.google.com/drive/folders/1rs8YtTGCDWFDMelUcIVndZbqxgaktwr9?usp=sharing (no borrar) el set de imágenes de llama en tres regímenes: laminar, transición y turbulenta. Las imágenes fueron adquiridas en el tiempo; considere que 10 imágenes corresponden a 1 segundo. Puede trabajar con una sola imagen o con una secuencia.

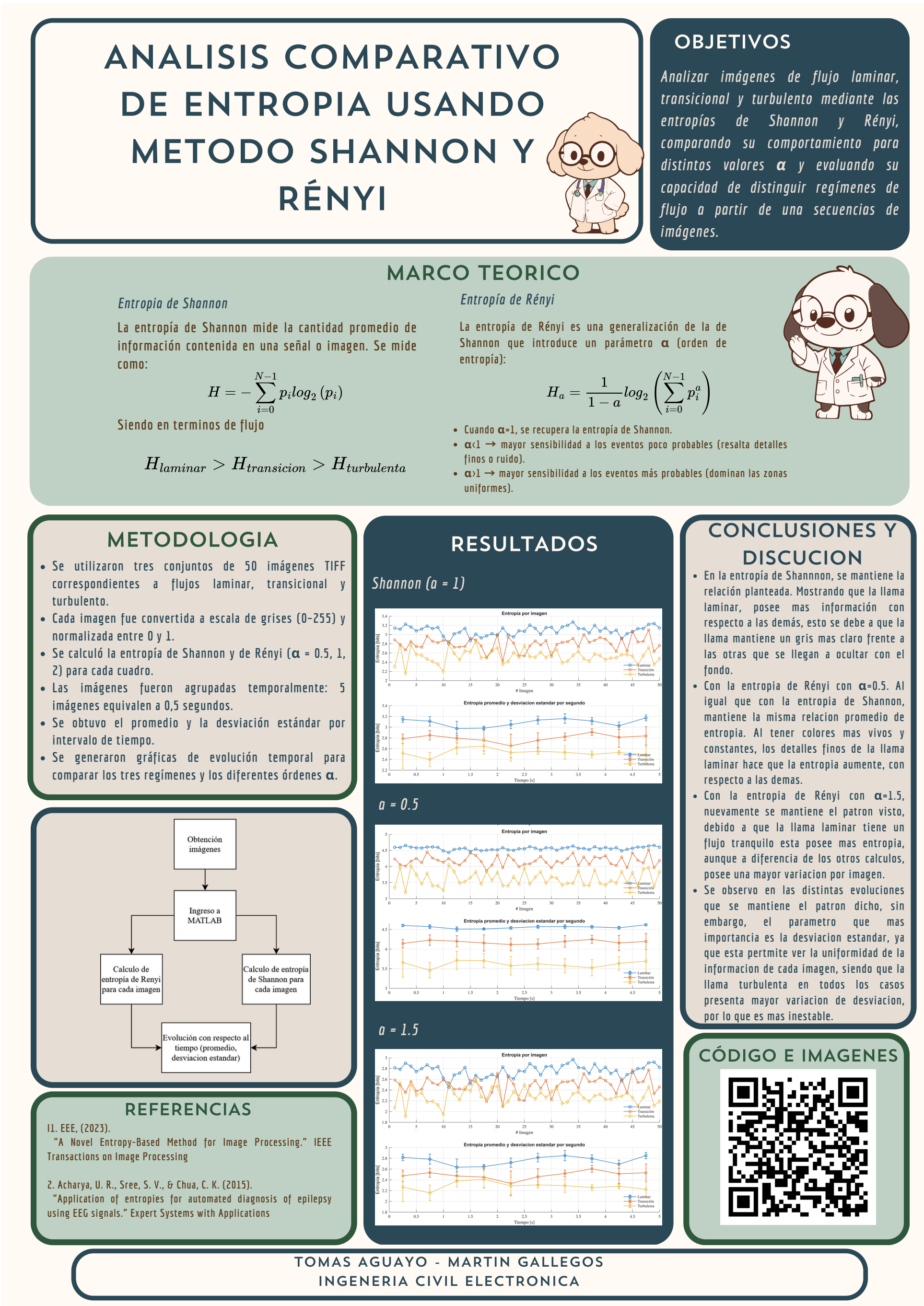

Figura 1: Póster científico

https://github.com/Aguayo3/Analisis-Imagenes

2. Problema 2

Para el desarrollo del siguiente problema, debe descargar la señal de audio, EN FORMATO *.WAV, en el siguiente link:

https://science.nasa.gov/resource/perseverance-rovers-supercam-records-wind-on-mars/

Como antecedentes, señal de audio fue adquirida por la cámara SuperCam en el rover Perseverance (localizado en el planeta rojo), con una frecuencia de muestreo de Fs=100KHz.

2.1. 1)

Como primer paso para este problema se leyó el audio descargado del link, utilizando el comando `audioread()`, que entrega la frecuencia de muestreo de la señal y un arreglo con los valores de esta.

```
%Extraer audio
[m,Fs]=audioread("AUDIO.wav");

%% 1
%PARAMETROS DE LA SEÑAL
Ts=1/Fs;
L=length(m);
t=(0:Ts:(L-1)/Fs)';
```

Listing 1: Obtención de parámetros de la señal de audio

Así la señal de audio se observa en Figura 2

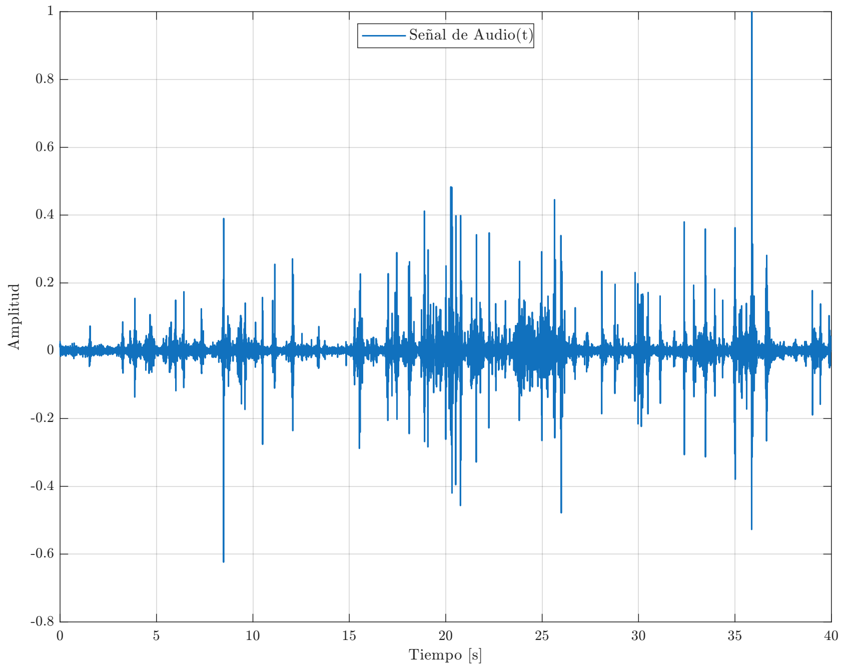

Figura 2: Señal de audio graficada en el tiempo

2.2. 2)

Para obtener el espectro de la señal analizada, se debe aplicar la transformada de Fourier. En este caso, debido a que la señal fue muestreada, esta es discreta, por lo que se aplica la transformada discreta de Fourier. Utilizando `fft()` se calcula la transformada rápida de Fourier en `Matlab`. Con el siguiente código se obtiene el espectro de la señal.

```
%% 2
%transformada rapida de fourier
M=fft(m);
%Modulo de la transformada
M2 = abs(M);
%plegado de fft()
M1=fftshift(M2);
%vector frecuencia
f=linspace(-1,1,L)*Fs/2;
```

Listing 2: Obtención de espectro de m(t)

Ahora bien, para el gráfico se escaló el vector obtenido, dividiendo por el número de muestras, de manera que este refleje la verdadera amplitud de la señal analizada, obteniéndose la Figura 3.

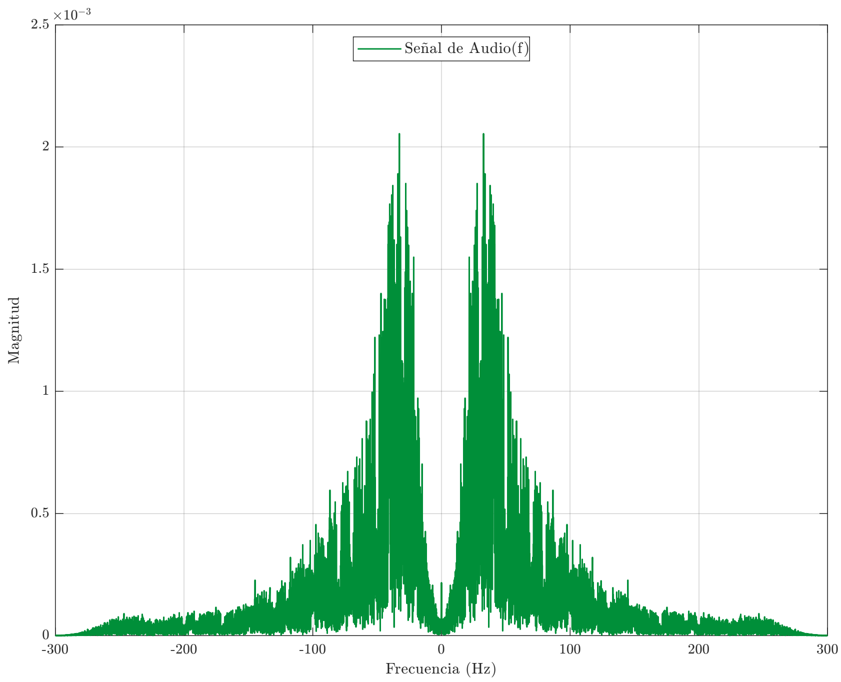

Figura 3: Espectro de Señal de audio

2.3. 3)

Para obtener la frecuencia fundamental de la señal se debe obtener la frecuencia donde este la amplitud mas alta. En Figura 3 se observa fácilmente la amplitud máxima. Asi en Figura 4 se indica el valor máximo.

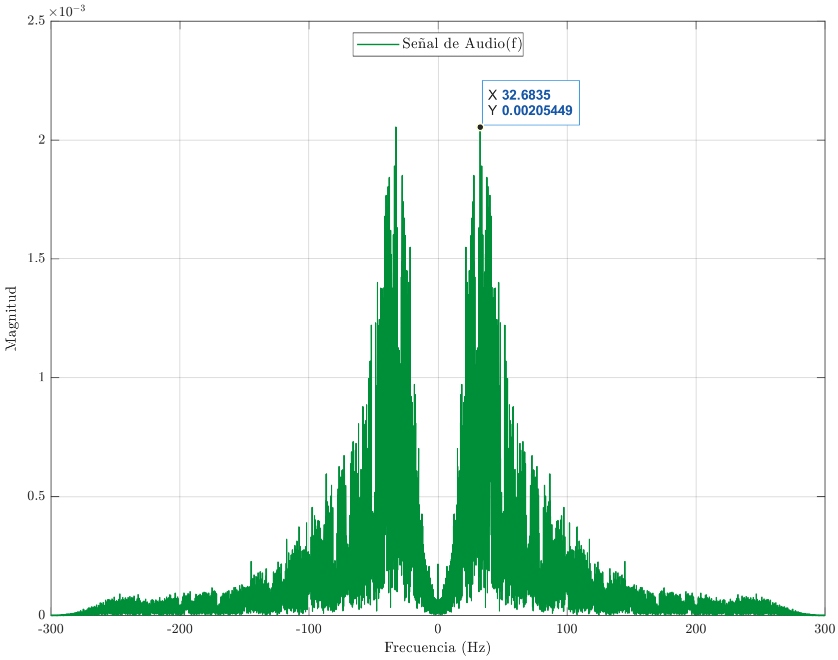

Figura 4: Frecuencia fundamental de m(t)

Para certificar este valor, se utilizo el comando `max()` de `Matlab`, lo que entrega el valor máximo en el arreglo y el indice de este.

```
%% 3
[MAX,f0]=max(M2/L);
f0=(f0-1)*Fs/L;
fprintf('La frecuencia fundamental es %.4f [Hz] \n',f0);
```

Listing 3: Obtención frecuencia fundamental

Obteniendo la siguiente salida:

```
La frecuencia fundamental es 32.6835 [Hz]
```

Corroborando el resultado de Figura 4.

2.4. 4)

Para obtener la PSD(f), se utiliza la definicion de esta, siendo:

$$\text{PSD}(f) = \lim_{T\to\infty} \frac{1}{T}|x(f)|^2 \left[\frac{\text{W}}{\text{Hz}}\right]$$
$$\text{PSD}[k] = \frac{1}{N \cdot Fs}|X[k]|^2$$

Aplicando la definición para el caso discreto, se tiene el siguiente codigo:

```
%% 4
PSD = (1/(Fs*L)) * abs(M).^2;    % PSD [W/Hz]
PSD_shifted = fftshift(PSD);
f = linspace(-Fs/2, Fs/2, L);
```

Listing 4: Obtencion PSD(f)

Asi el grafico de PSD(f) se ve en Figura 5

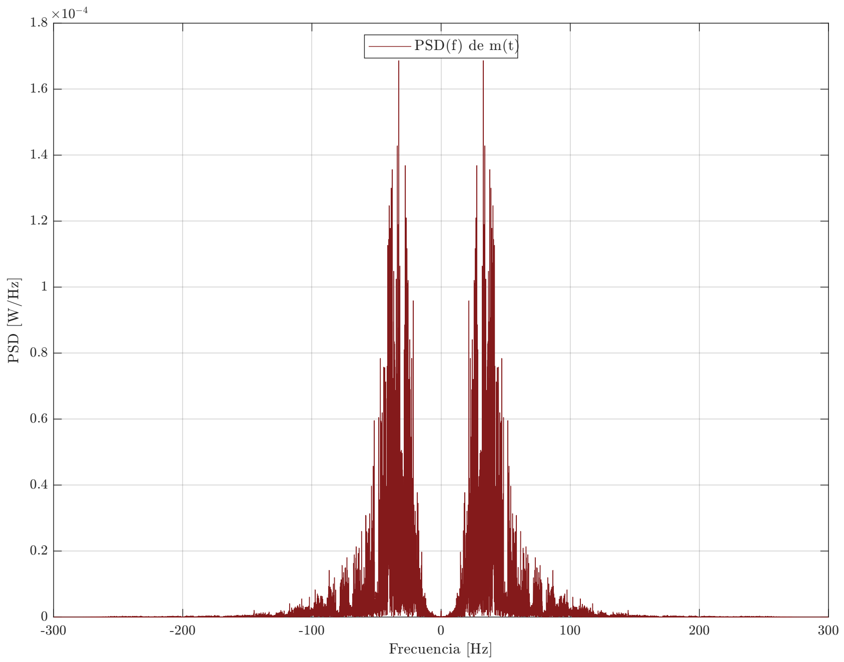

Figura 5: PSD(f) de m(t)

2.5. 5)

El criterio utilizado para definir el ancho de banda, es utilizando el PSD(f), visto en Figura 5, observando como va evolucionando la densidad de potencia a medida que se agregan mas frecuencias. Utilizando `cumtrapz()` se obtiene la evolución de la potencia, ver Figura 6.

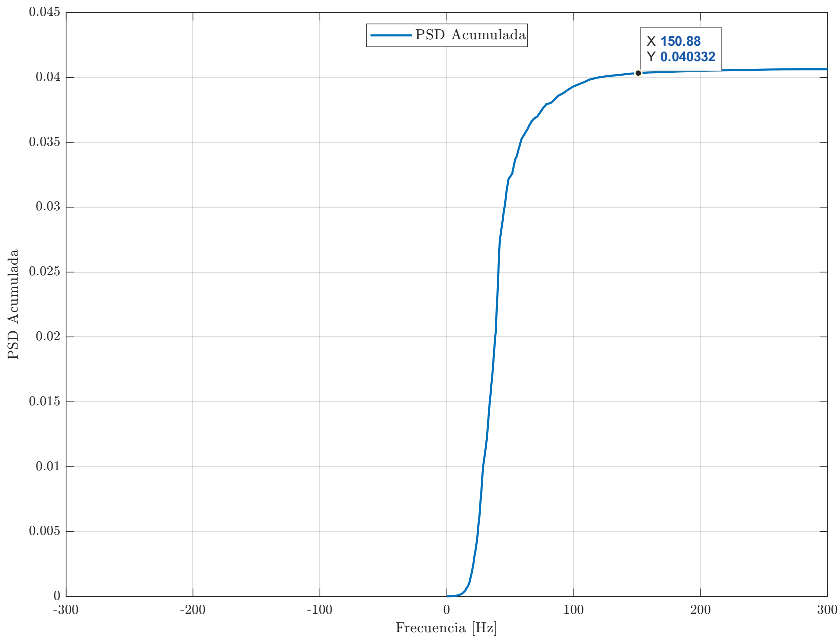

Figura 6: Acumulación de densidad de potencia

Como se observa en la evolución de la densidad de potencia obtenida. Esta llega a un punto estacionario superando los 100 [Hz], por lo que el ancho de banda de esta señal esta cercano a este valor. Bajo nuestro criterio el corte ocurre en la frecuencia indicada en Figura 6. Siendo esta de $BW = 150\,[\text{Hz}]$.

2.6. 6)

Asumiendo que $BW = 280\,[\text{Hz}]$, la frecuencia de muestreo nueva tiene que cumplir:

$$f_{s_{\text{new}}} \geq 2 \cdot BW$$

Por lo que:

$$f_{s_{\text{new}}} \geq 560\,[\text{Hz}]$$

Para este caso, se tiene que elegir $f_{s_{\text{new}}}$ ligeramente mayor. Se tomara para este caso la frecuencia $f_{s_{\text{new}}} = 600\,[\text{Hz}]$.

Así, para simular el muestreo de tren de impulsos se calculo la relación entre la nueva frecuencia y la frecuencia original, esto con el fin de obtener cada cuanto tomar del vector original. Creando un nuevo arreglo de ceros, del tamaño de la señal original, se tomaran los datos utilizando la relación de frecuencias. Esta metodología se observa en el siguiente código:

```
BW=280;
fs=600; %mayor a la frecuencia de muestreo de Nyquist

ts=1/fs;
tiempo_final=L*(1/Fs);
stride=round(Fs/fs);
N=tiempo_final/ts; %nro de muestras total que toma

%impulsos discretos(lugares donde no se muestreo es 0)
mss=zeros(size(m));
mss(1:stride:end)=m(1:stride:end);
%transformada de fourier de la señal muestreada
Ys=fft(mss);
Ys=fftshift(Ys)/N;
```

Listing 5: Muestreo por tren de impulsos

Con este método se obtiene la señal vista en Figura 7.

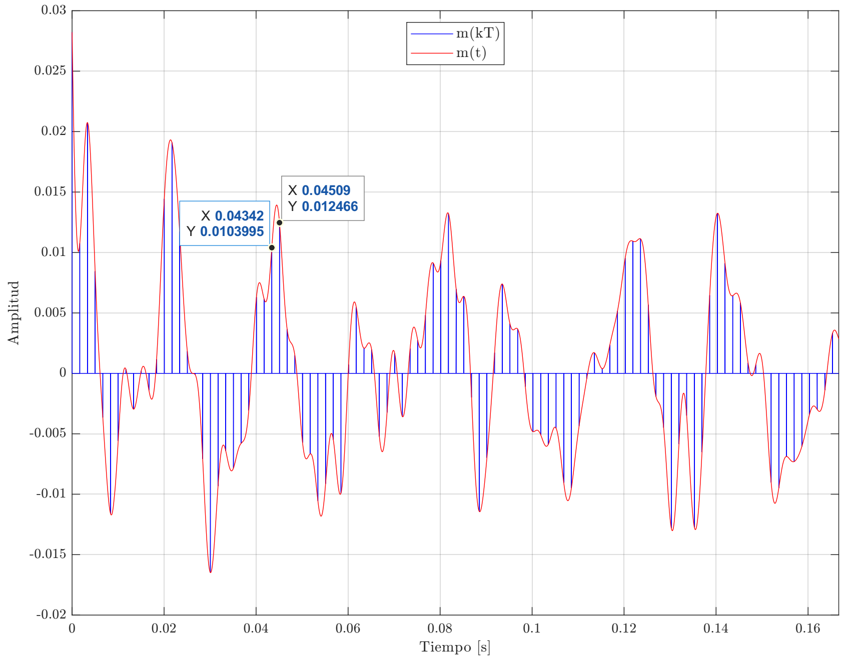

Figura 7: Comparacion señal original vs señal muestreada

Así cumpliendo el requisito de muestrear 600 veces por segundo.

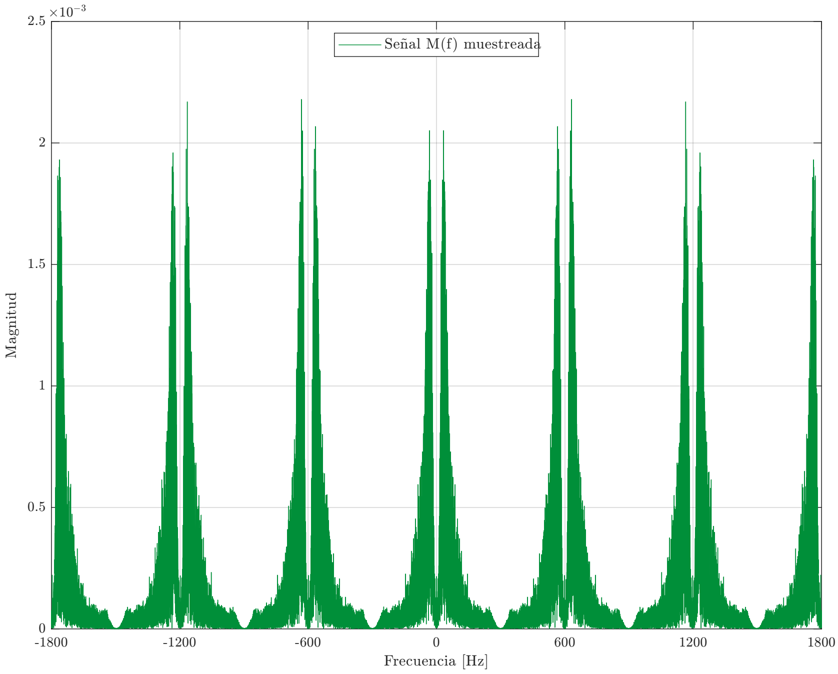

Figura 8: Espectro señal $\mathrm{m}_s(t)$

En la Figura 8, se observa el espectro de la señal muestreada. Donde lo principal que comentar es la periodicidad del espectro de la señal original, repitiéndose cada 600 [Hz], siendo esta la frecuencia de muestreo del tren de impulsos.

2.7. 7)

Utilizando el comando `sound()` en `Matlab`, se escucho el sonido capturado por la Nasa. Siendo apreciado principalmente un leve viento marciano y un silencioso ambiente con cierto misterio.

2.8. 8)

Aplicando lo dicho en el enunciado en el siguiente codigo:

```
%% 8
%filtrado de señal
ml=bandpass(m,[200 300],Fs);

%calculo de espectro
Yl=fft(ml);
Yl=fftshift(Yl);

f8=linspace(-1,1,L)*Fs/2;

%escuchar sonido filtrado
sound(4*ml,Fs)
```

Listing 6: Filtrado de señal

Dando como resultado el espectro visto en Figura 9

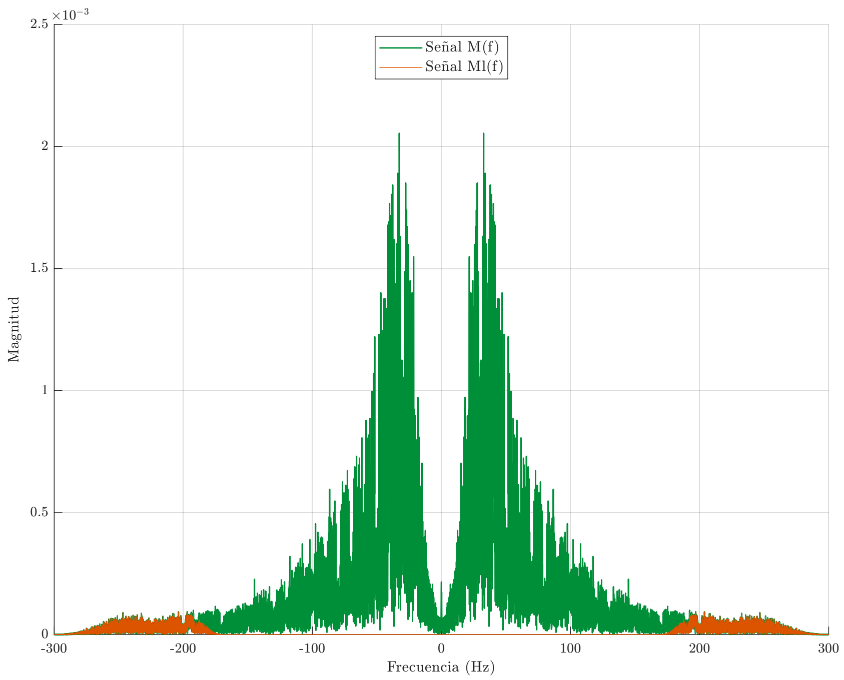

Figura 9: Comparación espectros señal filtrada vs señal original

En la figura se ve como gran parte de las frecuencias importantes fueron eliminadas, manteniendo las frecuencias mas altas de la señal original. A grandes rasgos la señal perdió gran parte de su potencia, por lo que para escucharla es necesario amplificarla.

Por ultimo, el sonido escuchado se escucha mas agudo, comparado con el sonido original, se escucha recortado sintiendo unos pequeños golpes, por ultimo, este se siente mas artificial, ya que el primero era un suave sonido de viento, este se escucha hueco o dentro de un armario.

3. Problema 3

Simule la modulación AM-DSB usando la señal m(t) como moduladora, y una portadora del tipo $c(t) = 1 \cdot cos(2\pi 2000t)$ usando las técnicas DSB-SC y DSB-LC. La señal modulada debe llamarla como $u_{AM-X}(t)$, tal que X=SC o LC, según sea el caso.

3.1. Codigo Matlab

Para realizar la simulación ocupamos dos comandos de Matlab como son el `fft()` que nos permite calcular la Transformada Rápida de Fourier de una señal y `fftshift()` que organiza el resultado de la FFT reorganizando el espectro de frecuencia para que el componente de frecuencia cero esté en el centro en lugar del principio, lo cual es útil para visualizarlo. Además normalizamos las señales para que las amplitudes se mantengan y se visualicen mejor en los gráficos de mas adelante.

```
%Extraer audio
[m,Fs]=audioread("AUDIO.wav");

%PARAMETROS DE LA SEÑAL
Ts=1/Fs;
L=length(m);
t=(0:Ts:(L-1)/Fs)';

%MODULACION
Uam_sc=m.*cos(2*pi*2000*t);
Uam_lc=(1+m).*cos(2*pi*2000*t);

f=Fs/2*linspace(-1,1,L);

%TRANSFORAMDA DE FOURIER
Yam_sc=fft(Uam_sc,L) /L;
Yam_lc=fft(Uam_lc,L) /L;

Yam_sc=fftshift(Yam_sc);
Yam_lc=fftshift(Yam_lc);

figure
plot(f,abs(Yam_sc))
xlim([2e3-1000 2e3+1000])
xlabel('Frequency [Hz]');
ylabel('Magnitude');
title('Transformada de Fourier Modulacion AM-DSB-SC');
grid on

figure
```

```
plot(f,abs(Yam_lc))
xlim([2e3-1000 2e3+1000])
ylim([0 2e-3])
xlabel('Frequency [Hz]');
ylabel('Magnitude');
title('Transformada de Fourier Modulacion AM-DSB-LC');
grid on
```

3.2. Resultados

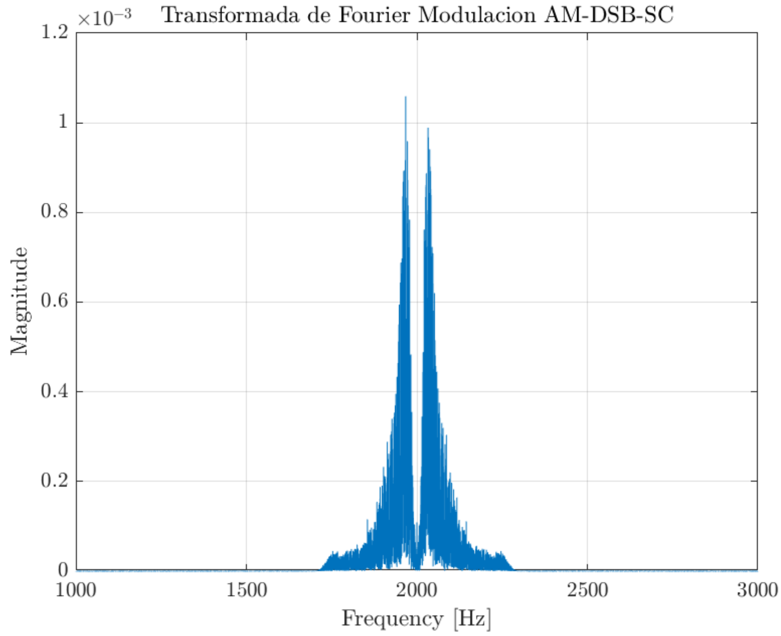

Figura 10: Transformada de Fourier Modulacion AM-DSB-SC

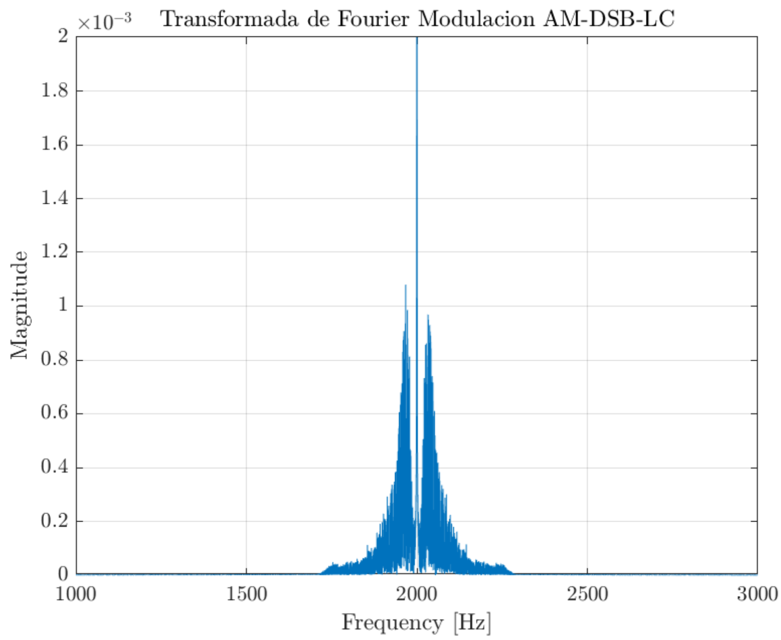

Figura 11: Transformada de Fourier Modulacion AM-DSB-LC

3.3. Comentarios

Lo primero que podemos notar es que en ambas técnicas, el espectro se encuentra correctamente centrado en la frecuencia de la portadora (2000 Hz), validando que la transformada de Fourier refleja el comportamiento esperado de una señal modulada en la amplitud solicitada del enunciado añadido de las dos bandas laterales en ambos casos. Y en base a modulación AM-DSB-SC se ve una clara eficiencia en magnitud, ya que elimina la portadora y transmite únicamente las bandas laterales, pero requiriendo una sincronización exacta entre el transmisor y el receptor. En cambio, la AM-DSB-LC es más fácil de modular ya que conserva la portadora, aunque a costa de una mayor magnitud como se ve reflejado en el gráfico. En general, ambas modulaciones mantienen la información de la señal moduladora en las bandas laterales, pero la diferencia principal radica en la presencia o ausencia de la portadora.