



PROCESADO DIGITAL DE SEÑAL

PROYECTO ESPECÍFICO - LABORATORIO 4 (FS0)

ANÁLISIS FRECUENCIAL – SONIDO

Componentes del grupo: - Alex Beltrán
- Daniel Cañadillas
- Ainhoa Serrano

Nota: Enviar este documento “Lab4_FSo_resultados.doc” completado con las tareas solicitadas, el código generado y los comentarios y aclaraciones que consideréis oportunos, junto con los correspondientes ficheros .m, en un fichero .zip vía eGela.

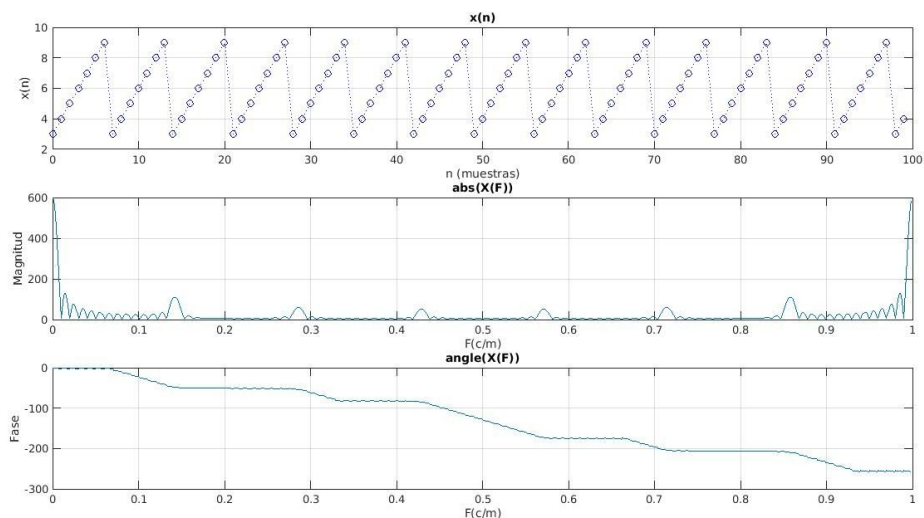
DTFT Y SEÑALES (P4_1_dft_x4_x5.m)

Genera las señales x4, y x5 y el resultado de los espectros obtenidos con la función `dtft`. Genera las figuras etiquetadas de manera adecuada y comenta los resultados obtenidos.

```
% x4
nn = 0:99;
x4 = 3 - rem(-nn,7);
dtft(x4,0,1000);

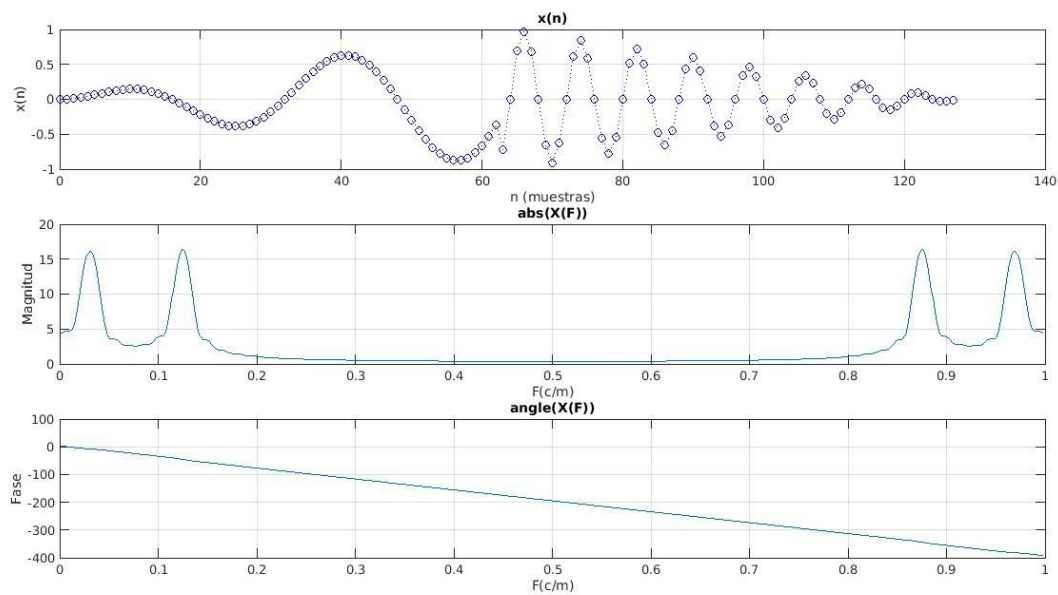
% x5
nn = 0:128;
x5_1 = (nn(1:63)/64).*sin(nn(1:63).*pi./16);
x5_2 = (2-(nn(64:128)/64)).*sin(pi./4.*nn(64:128));
x5 = [x5_1 x5_2];
dtft(x5,0,1000);
```

- X4



Tal y como se puede observar en la fórmula, existe una periodicidad debido al resto de la división, ya que se trata de valores periódicos que se irán repitiendo hasta llegar a algún múltiplo de 7. Dicha periodicidad se puede observar en espectro de magnitud estudiando los picos que resaltan de los demás valores, además, podemos apreciar en la señal cambios abruptos en los periodos, por lo que viene reflejado en el espectro de amplitud por picos pronunciados.

- X5



En la primera gráfica, se pueden apreciar dos funciones definidas para los dos distintos rangos, siendo el límite entre ambas 63. Al ser una señal formada por dos senos se puede apreciar en el espectro de amplitud como suceden dos picos que representan ambas sinusoides y sus correspondientes frecuencias.

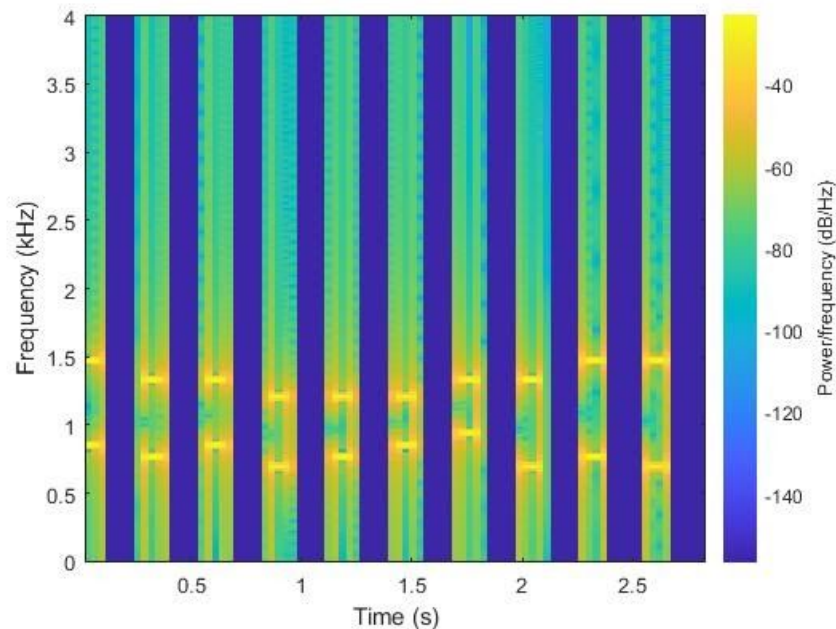
APLICACIÓN DE LA DFT A SONIDOS, ESPECTROGRAMA Y DTMF

Ejemplo 3: (P4_2_numero.m)

Genera la imagen del espectrograma asociado al fichero `numero.wav` (comenta los parámetros del comando `spectrogram` seleccionados).

`->spectrogram(numero,512,256,512,fs,'yaxis');`

Numero corresponde al double donde se ha guardado el sonido, `fs` corresponde a la frecuencia a la que se ha muestreado el sonido, 512 es la ventana usada, 256 es el overlap entre ventanas, y el segundo 512 indica el numero de puntos que utilizamos (nfft), 'yaxis' define sobre que eje lo hacemos.



¿Cuál es el número de teléfono codificado en la señal del fichero `numero.wav`? Describe el procedimiento seguido para calcularlo.

El número obtenido es: 9581470263.

La forma en la que lo hemos calculado ha sido observando el espectrograma asociado al fichero `numero.wav`, observando las frecuencias de cada par de sinusoides. Usando esas frecuencias en la tabla podemos decodificar el número.

Sinusoide 1 : 859,4 Hz / Sinusoide 2 : 1469 Hz	9
Sinusoide 1 : 765,6 Hz / Sinusoide 2 : 1344 Hz	5
Sinusoide 1 : 859,4 Hz / Sinusoide 2 : 1344 Hz	8
Sinusoide 1 : 703,1 Hz / Sinusoide 2 : 1203 Hz	1
Sinusoide 1 : 781,3 Hz / Sinusoide 2 : 1219 Hz	4
Sinusoide 1 : 843,8 Hz / Sinusoide 2 : 1219 Hz	7
Sinusoide 1 : 921,9 Hz / Sinusoide 2 : 1344 Hz	0
Sinusoide 1 : 687,5 Hz / Sinusoide 2 : 1344 Hz	2
Sinusoide 1 : 781,3 Hz / Sinusoide 2 : 1484 Hz	6
Sinusoide 1 : 703,1 Hz / Sinusoide 2 : 1484 Hz	3

```

[numero, fs] = audioread('numero.wav');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure; plot(numero);

spectrogram(numero,512,256,512,fs,'yaxis');

% 0.8594/1.469          ----> 9
% 0.7656/1.344          ----> 5
% 0.8594/1.344          ----> 8
% 0.7031/1.203          ----> 1
% 0.7813/1.219          ----> 4
% 0.8438/1.219          ----> 7
% 0.9219/1.344          ----> 0
% 0.6875/1.344          ----> 2
% 0.7813/1.484          ----> 6
% 0.7031/1.484          ----> 3

% 9581470263

```

Ejemplo 4: (P4_3_filtrado.m)

Completa la función para filtrar bandas de frecuencia **filtrb.m**. Además, genera las imágenes de la señal original y su espectro de amplitud antes y después de filtrar para cada una de las señales de los apartados a) y b) del guión. También puedes dibujar los espectrogramas para comprobar el efecto de los filtros. Comenta los resultados.

```
function y = filtrb(x,finf,fsup,fs)
% filtro rechazo banda en el dominio de la frecuencia
% y=filtrb(x,finf,fsup,fs) elimina de la señal x las componentes asociadas
% al rango de frecuencias comprendido entre finf Y fsup en Hz.
% Para ello calcula la DFT de la señal (X(F))
% Anula las componentes frecuenciales correspondientes y
% Obtiene la IDFT para volver al dominio n.

N = length(x);
% Calculamos la DFT
X = fft(x);
f = (0:N-1)'/N*fs;      % vector con frecuencias en Hz de cada valor de la DFT
% Anular los valores dentro de [finf,fsup]
% sin olvidar los elementos simetricamente colocados respecto de F=0.5

mask = (f <= finf) | (f >= fsup);
mask_simetrica = (f < (fs-fsup)) | (f >= (fs - finf));

X_n = X .* mask;
X_n_sim = X_n .* mask_simetrica;

figure;
subplot(3,1,1);
plot(f,x);
title("Señal original");

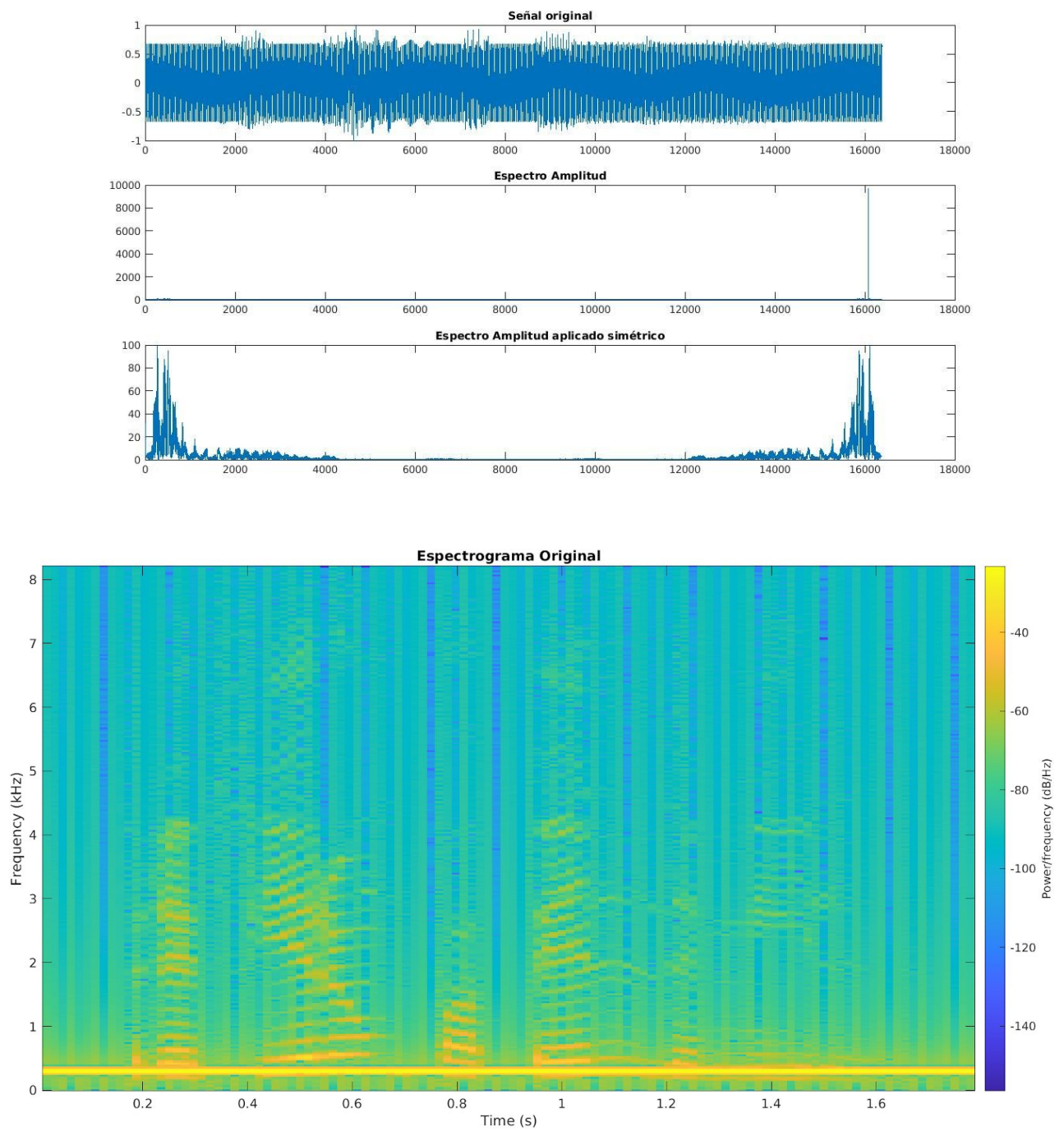
subplot(3,1,2);
plot(f,abs(X_n));
title("Espectro Amplitud");

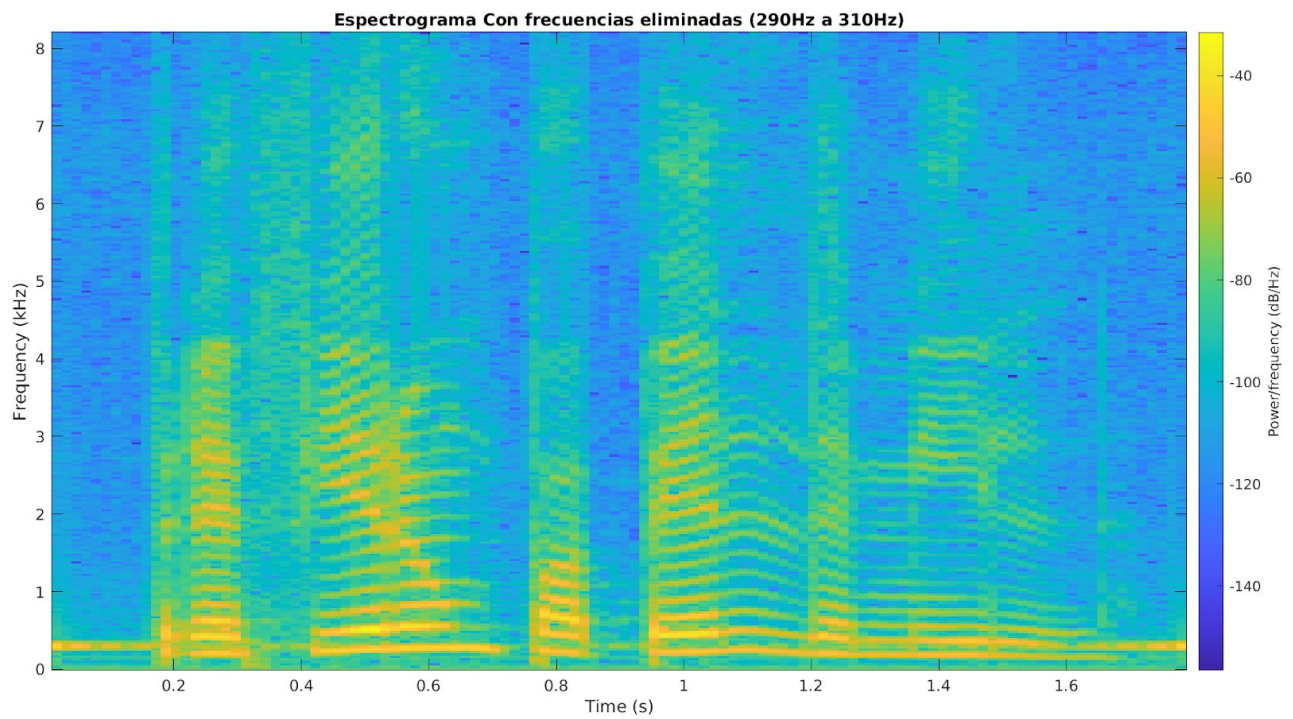
subplot(3,1,3);
plot(f,abs(X_n_sim));
title("Espectro Amplitud aplicado simétrico");

% Calculamos la nueva senal filtrada y mediante IDFT

y = real(ifft(X_n_sim));
```

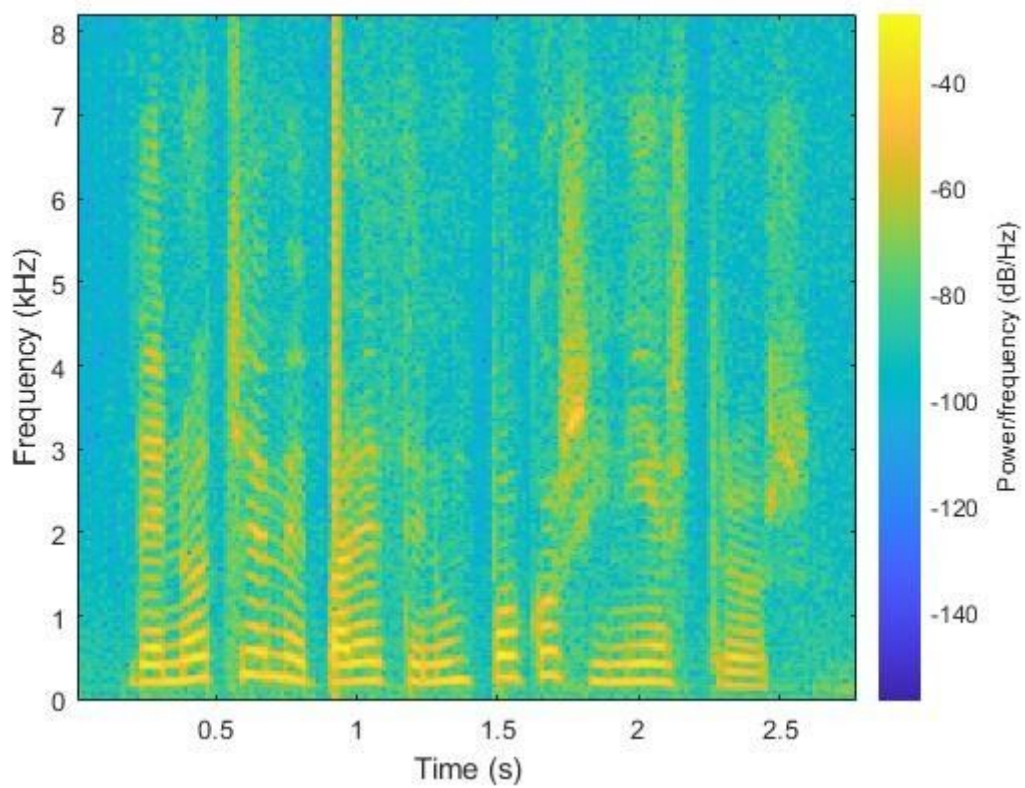
● a)



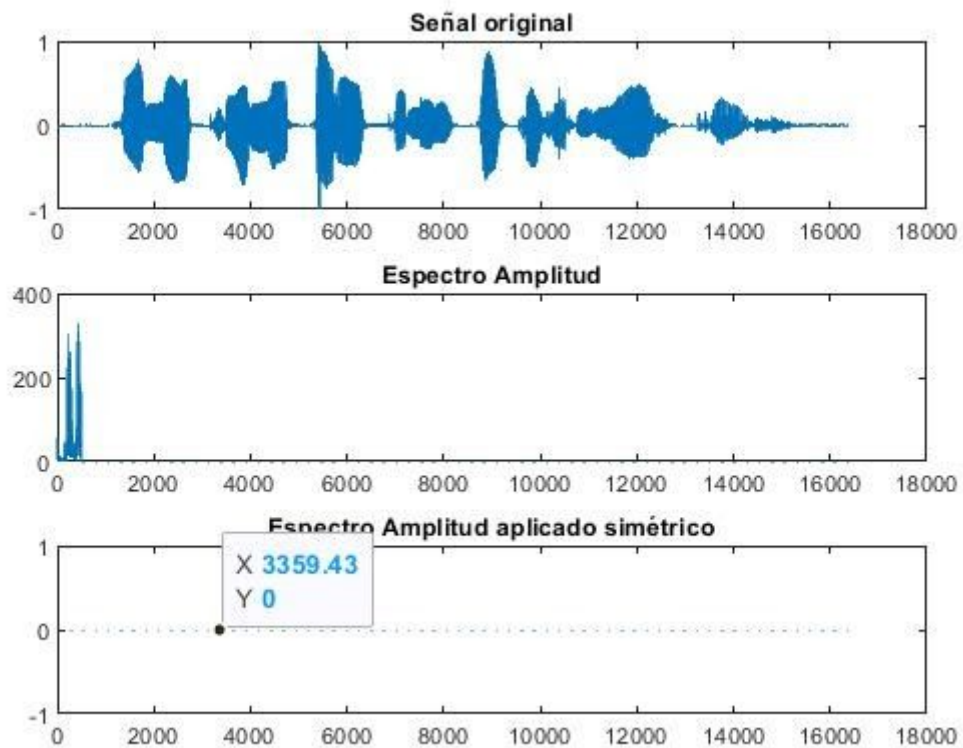


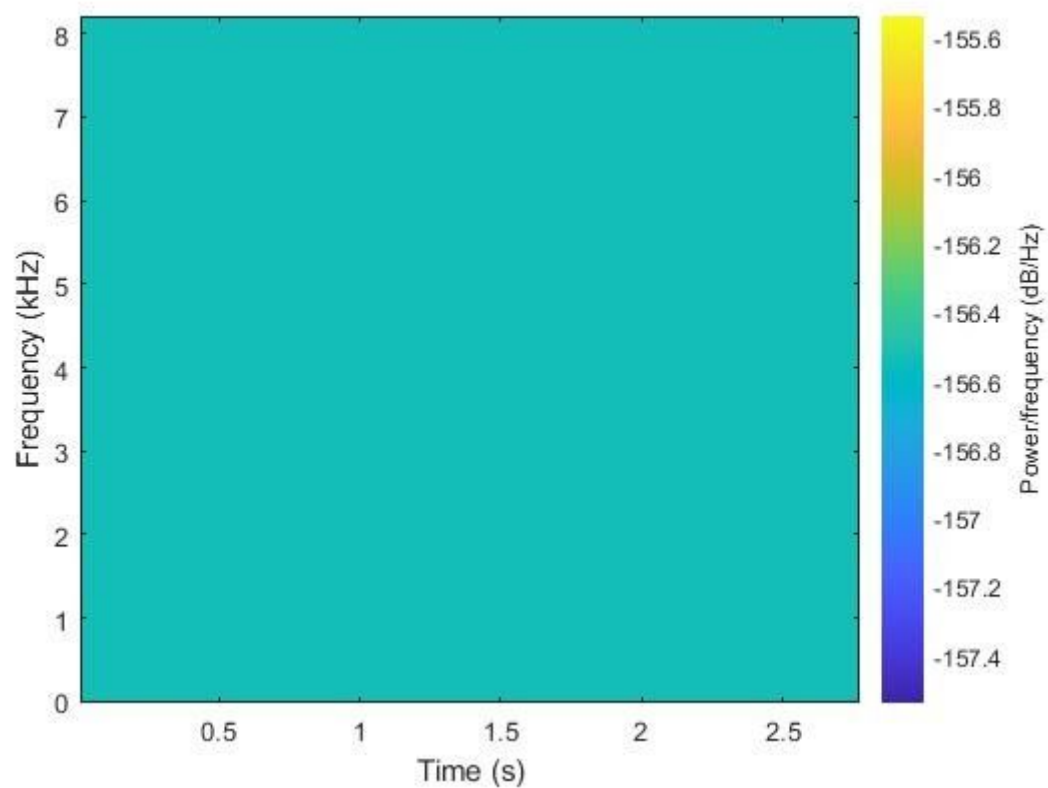
Tal y como se puede apreciar en el espectrograma original, existe una frecuencia constante durante toda la señal, dicha frecuencia se corresponde al pitido de la grabación. Para eliminarlo, se han quitado las frecuencias correspondientes entre 290 Hz y 310 Hz. Como resultado se puede apreciar como la frecuencia ha desaparecido del espectrograma.

Espectrograma original E101.wav:

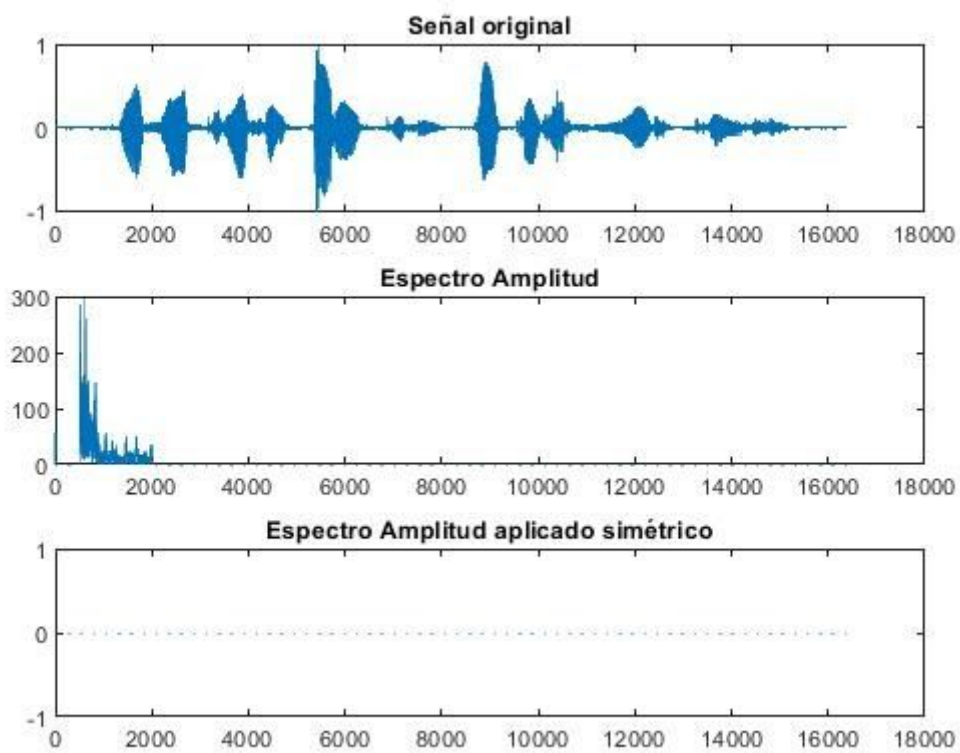


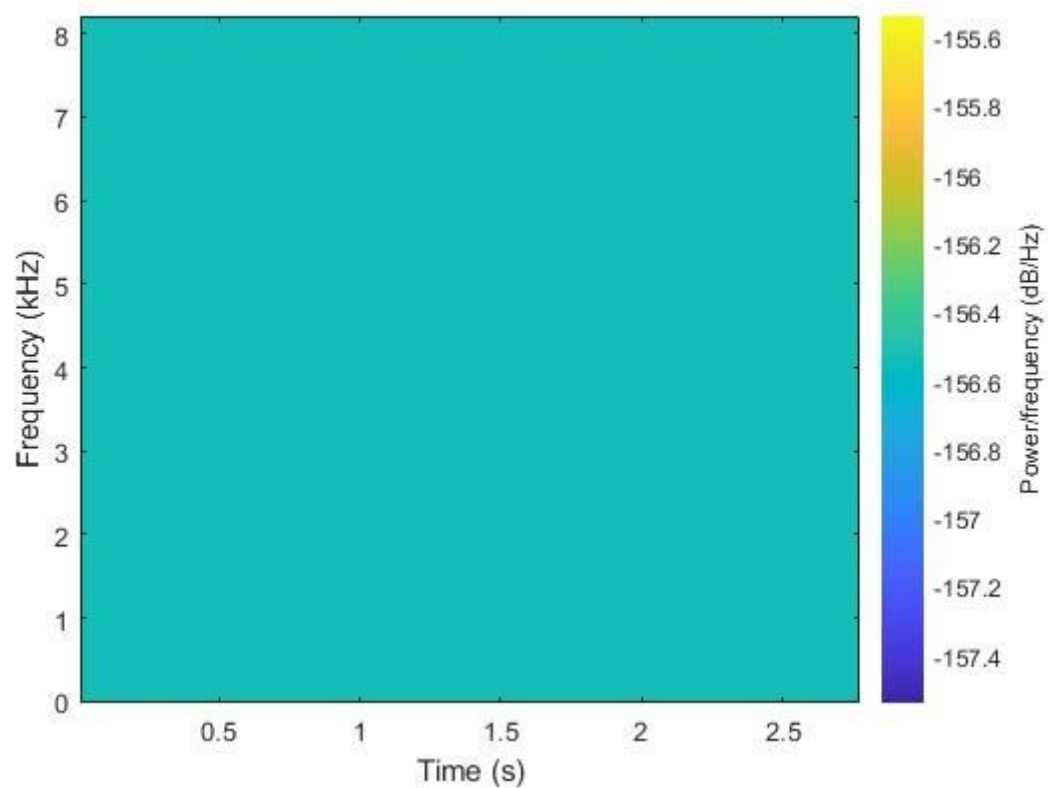
A) Frecuencias menores que 500:



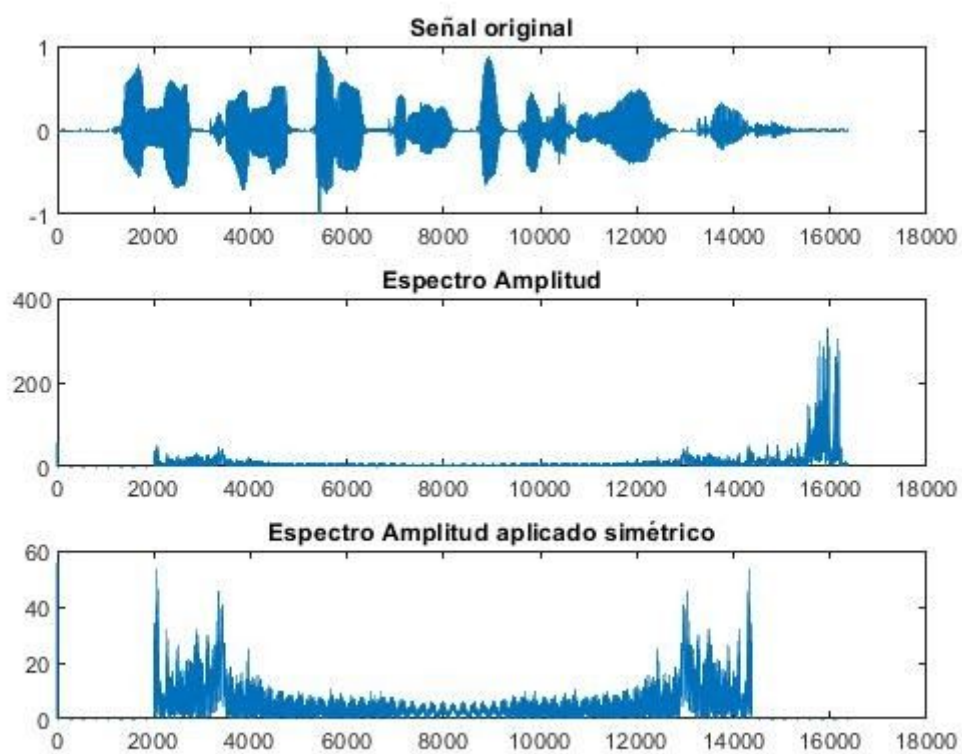


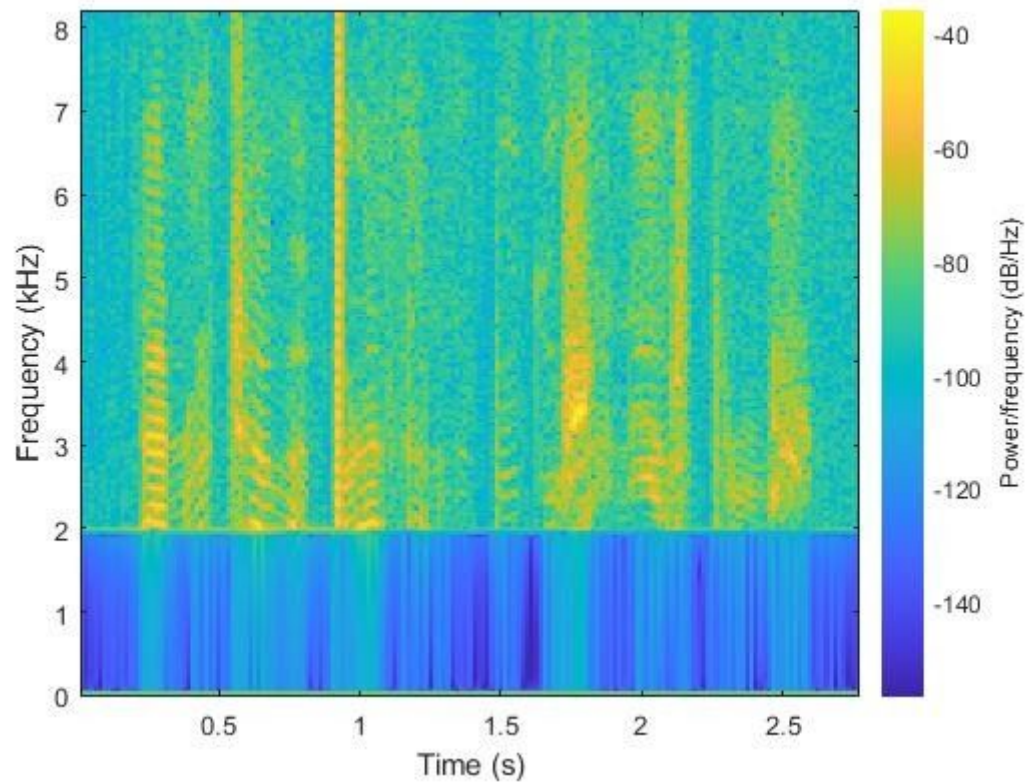
B) Frecuencias entre 500 y 2000:





C) Frecuencias mayores que 2000:





A la hora de sumar las señales, parece que se causa Eco.

```
[a,b] = audioread('e101.wav')
d1 = filtrbf(a, 500, b, b); d2 = filtrbf(filtrbf(a, 0, 500, b), 2000, b, b); d3 = filtrbf(a, 0, 2000, b);
spectrogram(d1,512,256,512,b,'yaxis');
spectrogram(d2,512,256,512,b,'yaxis');
spectrogram(d3,512,256,512,b,'yaxis');
filtrbf(a, 0, 2000, b)
filtrbf(filtrbf(a, 0, 500, b), 2000, b, b);
filtrbf(a, 500, b, b);
```