



Universidad Nacional de Rosario
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Escuela de Ingeniería Electrónica

Sistemas y señales I

Año 2021

Trabajo Práctico 3

Autor:

Alumno	Legajo
BELLINI, Valentin	B-6127/1
GASPOZ, Gastón Leando	G-5580/8

Docentes
Dr. Juan Carlos Gómez Ing. Franco Del Colle Dr. Ariel Bayá Dr. Gonzalo Sad

Fecha de entrega: 16/07/2021

Problema 1: procesamiento de una señal ADSL.

Introducción.

En el problema 1 se plantea una señal ADSL ('Asymmetric Digital Subscriber Line') que contiene 3 componentes; señal de voz en el rango [0.3 – 3.4] KHz, señal de datos UPSTREAM en el rango [25.875 - 138] KHz y señal de datos DOWNSTREAM [138 – 1104] KHz. Se puede observar un esquema frecuencial de la señal en la figura 1.

Se usará una frecuencia de muestreo para muestrear la señal dada y a partir del uso de la transformada de Fourier encontrar el espectro de amplitud de la misma. Luego, en el dominio frecuencial se aplicarán filtros para separar la componente de interés de la señal ADSL (Señal de voz) y, a través de la transformada inversa de Fourier volver al dominio temporal para reproducir la señal de audio filtrada.

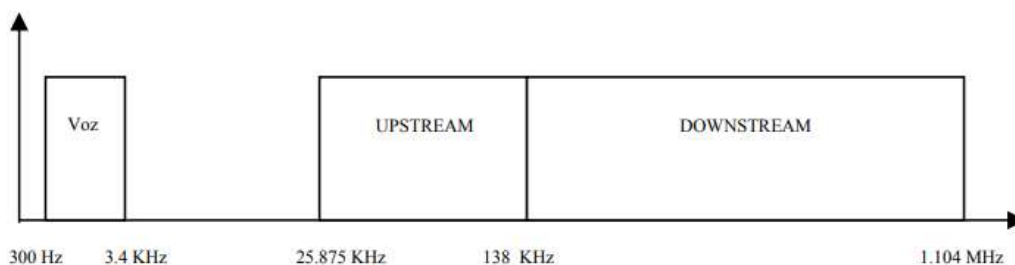


Figura 1. Esquema en frecuencias de la señal.

Para muestrear de manera correcta la señal de datos ADSL se utiliza el teorema del muestreo, que propone una frecuencia de muestreo F_s cuyo valor sea superior al doble de la máxima frecuencia presentada en la señal de datos para que no se produzca aliasing.

En este caso, se observa que: $F_{max} = 1.104 MHz \rightarrow F_s = 2 \cdot F_{max} = 2.208 MHz$

Una vez conocida la frecuencia de muestreo, se importa la señal de datos ADSL a Matlab a través del comando "load" para poder graficar la señal temporal de la figura 2.

A continuación, se observa el script de Matlab utilizado para el ploteo de la señal.

Script "Problema1a.m" Matlab
<pre>%Carga de la señal load('datosADSL.mat'); Fmax = 1.104e6; %Definicion de la frec maxima Fs = 2*Fmax; %Frecuencia de muestreo L = length(datosADSL); %Longitud de la señal t = [0:1/Fs:(L-1)/Fs]; %Def tiempo %Ploteo subplot(211), plot(t,datosADSL); %Ploteo señal ADSL title('Señal ADSL'); xlabel('Tiempo [s]'); ylabel('Señal ADSL'); grid on axis([0 0.625 -200 100]) subplot(212), plot(t,datosADSL); %Ploteo señal con zoom title('Señal ADSL [Zoom]'); xlabel('Tiempo [s]'); ylabel('Señal ADSL'); grid on axis([0 0.625 -3 3])</pre>

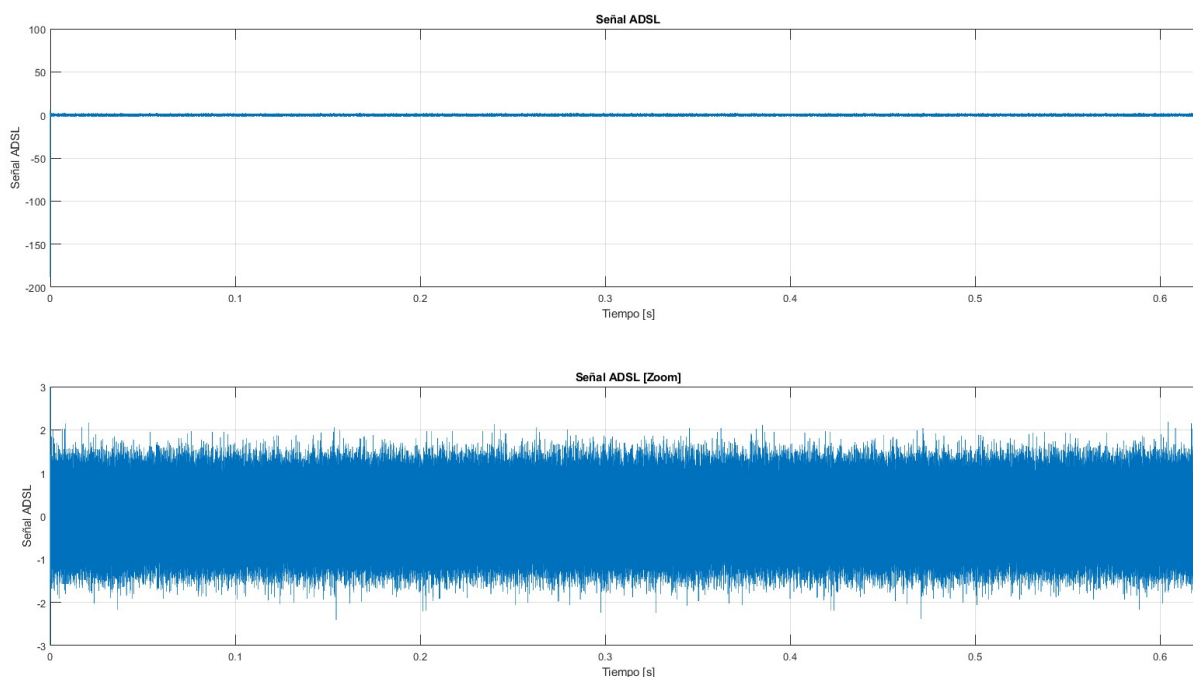


Figura 2. Ploteo de la señal ADSL vs tiempo.

A fin de realizar una gráfica del espectro de amplitud, utilizamos la transformada de fourier rápida (FFT) con un orden de $\frac{N}{2} \log_2(N)$ multiplicaciones complejas cuando N es potencia entera de 2. Para esto se define un N mayor a la longitud de los datos de la señal y que a su vez sea potencia entera de 2 para optimizar el cálculo computacional.

Script "Problema1b.m" Matlab
<pre> %Carga de datos de la señal load('datosADSL.mat'); L = length(datosADSL); %Este valor es cercano a 138.000 Fmax = 1.104e6; %Definicion de la frec maxima Fs = 2*Fmax; %Frecuencia de muestreo N = 2^21; %Notar que N > L F=[-Fs/2:Fs/N:Fs/2-Fs/N]'; %Vector de frecuencias %Transformada de Fourier X = fft(datosADSL,N); %Ploteo figure plot(F,abs(fftshift(X))); title('Espectro de Amplitud'); xlabel('Frecuencia [Hz]'); grid on </pre>

Observación: Para poder graficar las frecuencias de ω entre $-\pi$ y π se utilizó el comando *fftshift* en Matlab.

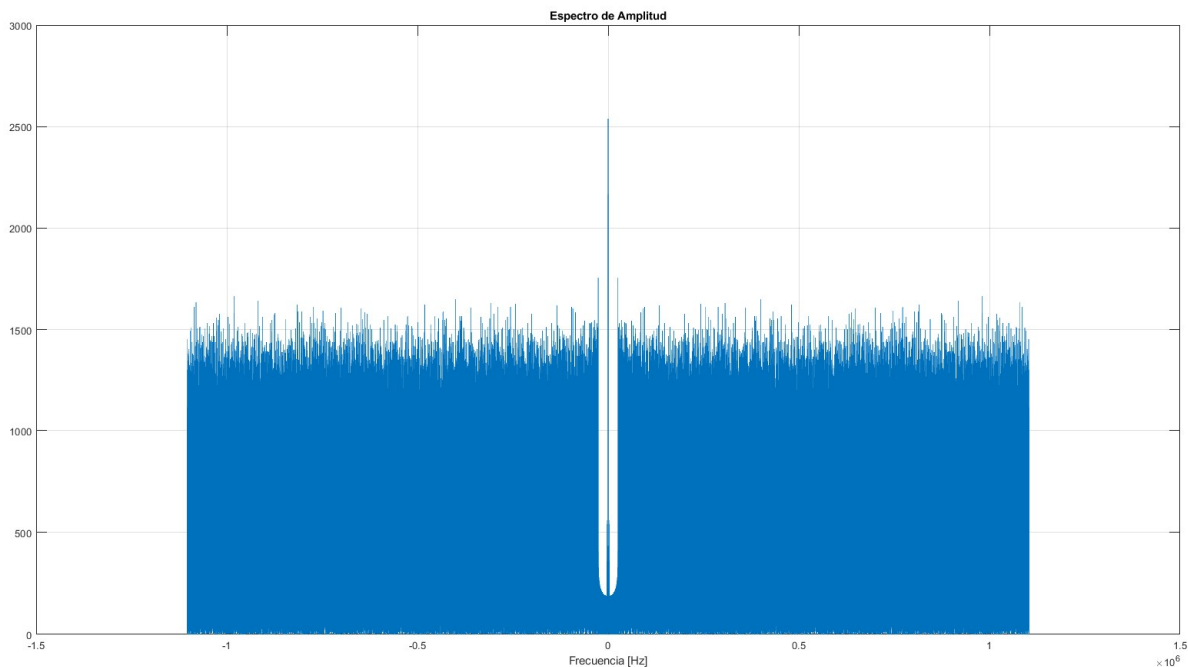


Figura 3a. Espectro de amplitud de la señal ADSL.

Observar que el espectro de amplitud queda definido con todos esos picos debido a que se realizó el cálculo de la FFT con los N puntos. Esto sería más útil en el caso de que la señal de datos tenga una longitud mucho mayor. Sin embargo, se puede observar que, en el rango de frecuencias deseado, la amplitud es uniforme.

En el caso de realizar la transformada de la señal de la forma $X = \text{fft}(\text{DatosADSL})$; se obtendría un espectro con la siguiente forma:

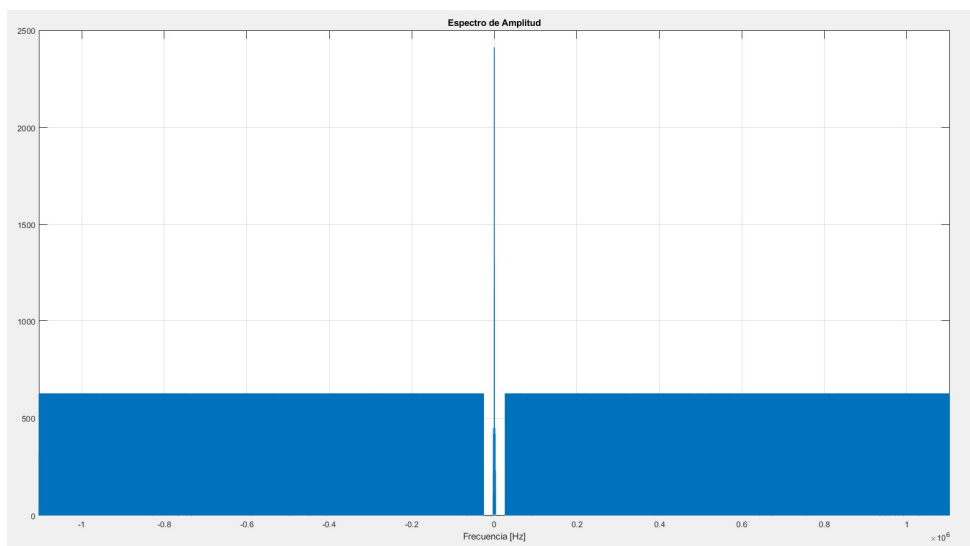


Figura 3b. Espectro de amplitud de la señal ADSL.

Luego se procede a extraer las diferentes partes que componen la señal de ADSL utilizando el espectro de amplitud en el dominio de la transformada de Fourier hallado anteriormente. Para ello se generan 3 filtros pasabanda (uno para cada componente a extraer: Voz, Upstream y Downstream), los cuales poseen valor igual a 1 en aquellas frecuencias donde se encuentren las componentes de la señal correspondiente y 0 en las demás.

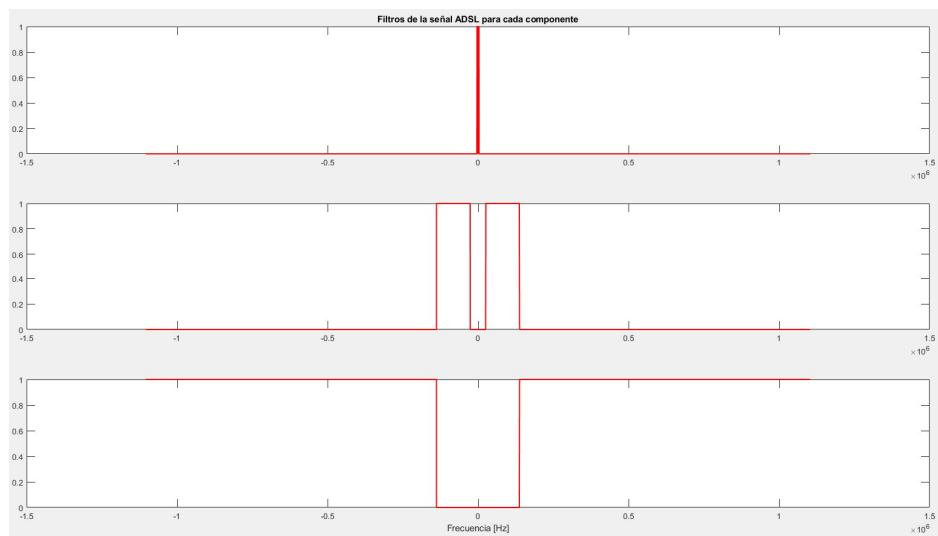


Figura 4. Filtros correspondientes a cada señal.

Al aplicar dichos filtros a las señales correspondientes se obtienen las componentes filtradas de la señal de voz, de Upstream y de Downstream. En la figura 5 se puede observar la representación gráfica de las componentes filtradas.

Observación: La señal de voz fue plotada en distinta escala para apreciar mejor su espectro.

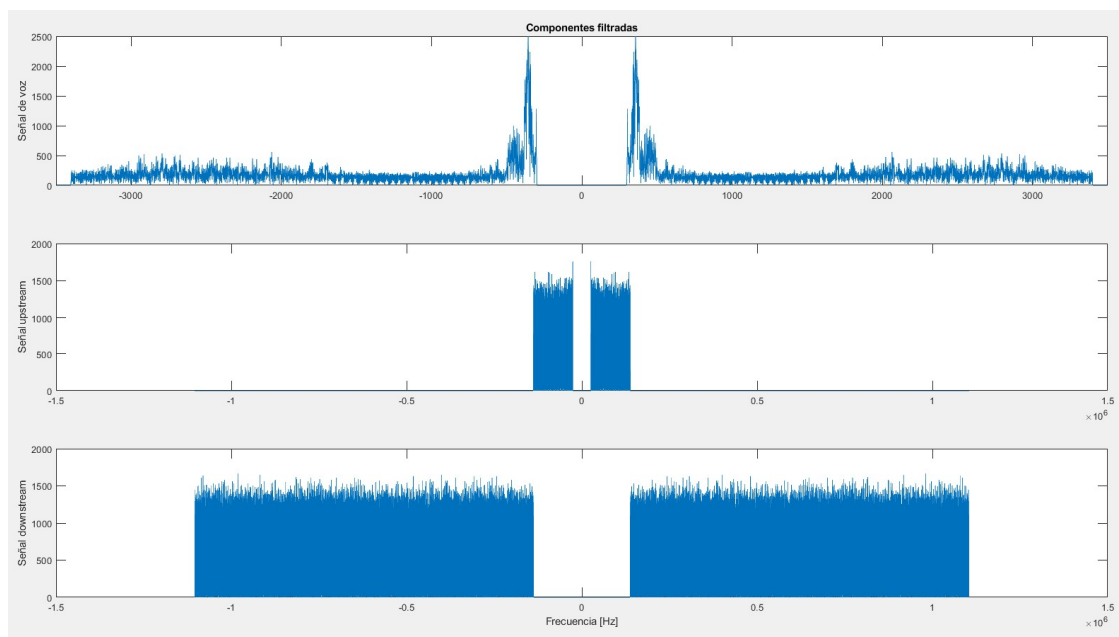


Figura 5. Componentes de la señal filtradas.

La implementación de los filtros en Matlab se realizó según el siguiente script.

Script “Problema1C.m” Matlab
<pre>load('datosADSL.mat'); L = length(datosADSL); %Longitud de la señal ADSL Fmax = 1.104e6; %Definicion de la frec maxima Fs = 2*Fmax; %Frecuencia de muestreo N = 2^21; %Notar que N > L F=[-Fs/2:Fs/N:Fs/2-Fs/N]'; X = fft(datosADSL,N); %Transformada de Fourier Y = fftshift(X); % % ceil() rounds to the next higher integer, in the +infinity direction. % % floor() rounds to the next lower integer, in the -infinity direction. %Filtro de voz k1 = ceil(300*N/Fs); k2 = floor(3400*N/Fs); Hvoz = zeros(N,1); Hvoz(k1:k2) = 1; Hvoz(N-k2:N-k1) = 1; % Filtro de Upstream k3 = ceil(25875*N/Fs); k4 = floor(138e3*N/Fs); Hup = zeros(N,1); Hup(k3:k4) = 1; Hup(N-k4:N-k3) = 1; % Filtro de Downstream k5 = ceil(138e3*N/Fs); k6 = floor(1.104e6*N/Fs); Hdown = zeros(N,1); Hdown(k5:k6) = 1; Hdown(N-k6:N-k5) = 1;</pre>
<pre>% %Ploteo del filtro H subplot(311), plot(F,fftshift(Hvoz),'r','LineWidth',1.5); title('Filtros de la señal ADSL para cada componente'); subplot(312), plot(F,fftshift(Hup),'r','LineWidth',1.5); subplot(313), plot(F,fftshift(Hdown),'r','LineWidth',1.5); xlabel('Frecuencia [Hz]');</pre>
<pre>%Señales filtradas Svoz = X.*Hvoz; %Señal de voz filtrada Sup = X.*Hup; %Señal de Upstream filtrada Sdown = X.*Hdown; %Señal de Downstream filtrada</pre>
<pre>%Ploteo señales filtradas figure (2) subplot(311), plot(F,fftshift(abs(Svoz))); title('Componentes filtradas') axis([-3500 3500 0 2500]) ylabel('Señal de voz'); subplot(312), plot(F,fftshift(abs(Sup))); ylabel('Señal upstream'); subplot(313), plot(F,fftshift(abs(Sdown))); ylabel('Señal downstream'); xlabel('Frecuencia [Hz]');</pre>

Por último, mediante la transformada inversa de la señal de voz filtrada (ifft), se reproduce la misma mediante el comando “*soundsc*” con una frecuencia de 6800 Hz. Puede identificarse el sonido de la señal de voz: “**Bienvenidos**”.

Se utilizó una frecuencia de muestreo de 6800Hz para reproducir la señal de audio debido a que la misma es el doble de la máxima frecuencia contenida en la señal de voz filtrada, aunque puede cambiarse para escucharla en otra frecuencia. Se puede apreciar que, al aumentar la frecuencia de muestreo, la duración

de la señal será menor y el tono del sonido será más agudo, mientras que, si se reduce la frecuencia de muestreo, la duración de la señal de audio será mayor y con un tono más grave.

Script “Problema1C.m” Matlab

```
%Reproducir señal de audio.  
VozFiltrada = real(ifft(Svoz)); %Parte real de la transformada inversa  
Vozresample = resample(VozFiltrada, 6800,Fs); %Resample para poder reproducir  
soundsc(Vozresample,6800);
```

Se plotó la señal de voz en función del tiempo para una frecuencia de 8192 Hz (valor default del comando soundsc) y se observa que la duración total de la señal es próxima a los 0.52 segundos mientras que para la frecuencia escogida de 6800 Hz la señal tiene una duración próxima a los 0.625 segundos.

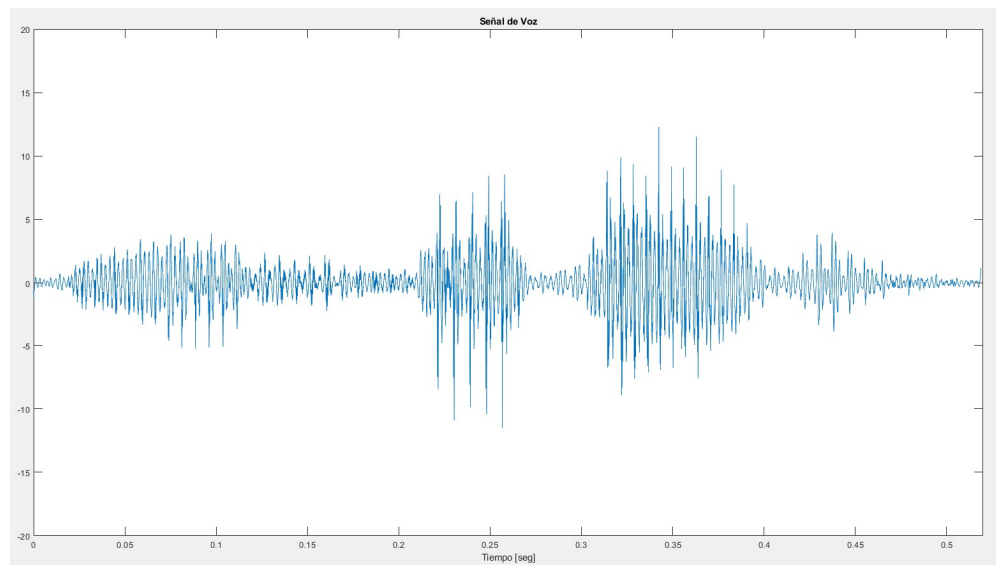


Figura 6a. Señal de voz filtrada con $F_s = 8192$ Hz

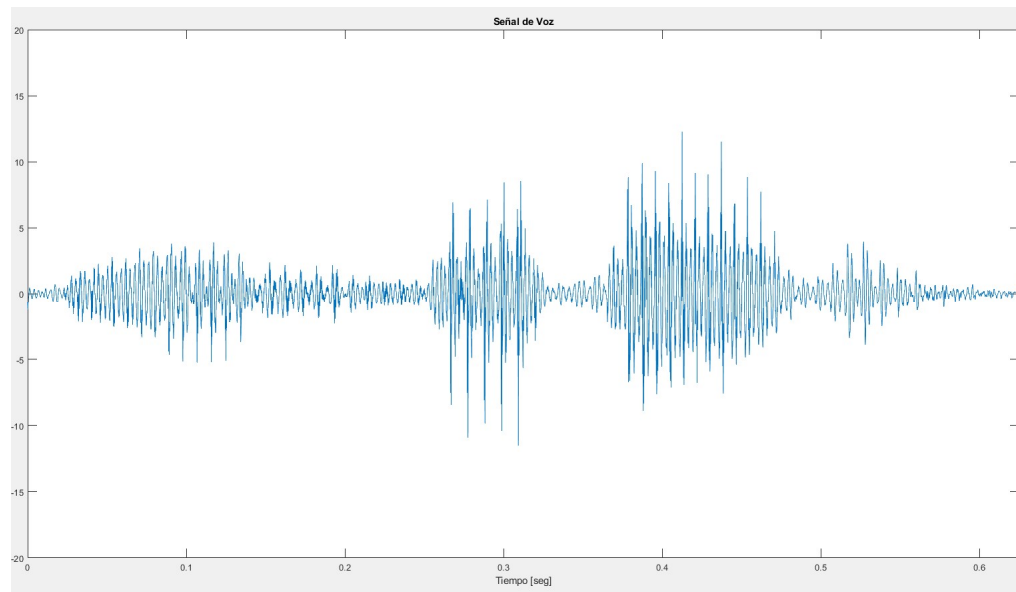


Figura 6b. Señal de voz filtrada con $F_s = 6800$ Hz

Problema 2: Análisis frecuencial utilizando ventanas

Introducción.

En el sistema de marcación por tonos utilizado en telefonía, también llamado sistema multifrecuencial o DTMF (*Dual-Tone Multi-Frequency*), cuando un usuario pulsa en el teclado de su teléfono la tecla correspondiente a un dígito que quiere marcar, se envían dos tonos de distinta frecuencia (uno por columna y otro por fila de acuerdo a la Tabla 1).

Así, la señal generada al presionar un dígito, tendrá dos componentes y una forma genérica:

$$x(t) = A \sin(2\pi F_L t) + A \sin(2\pi F_H t) \xrightarrow{F_s} x(n) = A \sin(2\pi f_L n) + A \sin(2\pi f_H n)$$

Luego, la central telefónica detecta las frecuencias contenidas en la señal y determina el dígito que se marcó. En los casos en que las frecuencias FL y FH difieren de sus valores nominales, indicados en la Tabla 1, en $\pm 1.8\%$, la central telefónica descarta el dígito enviado.

		F _H [Hz]			
		1209	1336	1477	1633
F _L [Hz]	697	1	2	3	A
	770	4	5	6	B
	852	7	8	9	C
	941	*	0	#	D

Tabla 1. Frecuencias asociadas a cada dígito del sistema DTMF.

El problema nos presenta una señal de audio de duración finita ("**tonos.wav**") que contiene los tonos de las diferentes señales correspondientes a las teclas presionadas por el usuario. Primero intentamos calcular la transformada de Fourier de la señal completa y verificamos si se puede o no determinar el número marcado a partir del espectro de ella.

Script "Problema2a.m" Matlab

```
clear;clc;
% muestra-Frecuencia de muestreo
[Y, Fs, Nbits] = wavread('tonos.wav');
L=length(Y); % Longitud total de 'tonos.wav' una vez muestreada
N=2^16; % N > L
F=[-Fs/2:Fs/N:Fs/2-Fs/N]';
X=fft(Y,N); % Transformada de Fourier rápida con N muestras

% Ploteo
plot(F,abs(fftshift(X)));
title('Espectro de Amplitud');
xlabel('Frecuencia [Hz]');
```

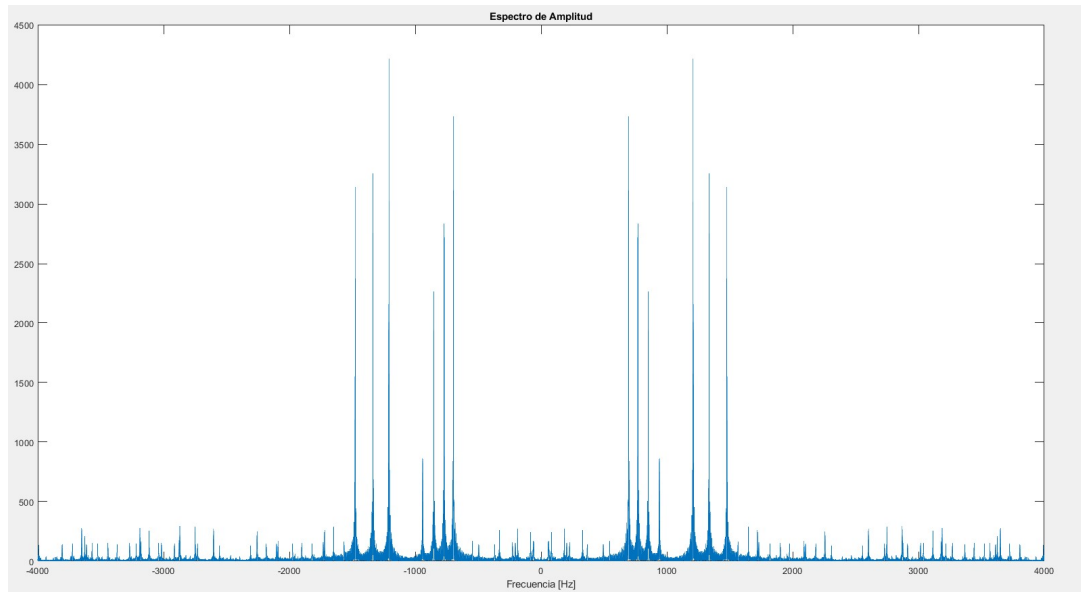



Figura 7. Espectro de amplitud de la señal de entrada "Tonos.wav"

Como se puede observar, aunque sabemos que frecuencias son utilizadas en el marcado de los números (por la amplitud de las componentes en cada frecuencia), es muy complicado verificar cuantas veces aparece cada una de ellas, siendo imposible determinar con qué combinación lo hacen.

Para solucionar este problema, se procederá a subdividir la señal en intervalos de tiempo (frames) lo suficientemente chicos como para que solo aparezca un número a la vez (450 ms), y así poder identificarlo.

Como en cada frame se van a poder identificar 2 frecuencias predominantes ($F_l < 1000$ Hz y $F_h > 1000$ Hz), se va a tener que buscar un pico máximo en la FFT del frame entre 0 y 1000 Hz, que determinará la fila del número que se discó, y otro entre 1000 Hz y $F_s/2$, que determinará la columna del número discado.

Luego, con dichas frecuencias, se llamará a la función "marcado" para que esta encuentre el número que se marcó en esta subdivisión de la señal. Esta función devolverá el carácter '/' si las frecuencias que se les pasan difieren de los valores nominales más de un 1.8%.

Script "Problema2d.m" Matlab
<pre>[Y, Fs] = audioread('tonos.wav'); % Subida del archivo L = (450e-3)*Fs; % Longitud de cada frame Nframes = length(Y)/L; % Número de frames en total F1000 = 1000*L/Fs; % Posición correspondiente a los 1KHz F=[0:Fs/L:Fs-Fs/L]'; Liminf=0; %Limite inferior for i = 1:Nframes Limsup=L*i; % Limite superior de la subdivisión Yx(:,i)=Y(Liminf+1:Limsup , 1); % Subdivisión de Y en los frames Liminf=L*i; % Limite inferior de la subdivisión Ytf(:,i) = fft(Yx(:,i)); % Transformada de las subdivisiones [valor1,pos1]=max((Ytf(1:F1000,i))); % Máximo antes de los 1KHz [valor2,pos2]=max(Ytf(F1000:(L/2),i)); % Máximo después de los 1KHz F1=pos1*Fs/L; %Posición Frecuencia 1 F2=(pos2+F1000)*Fs/L; %Posición frecuencia 2 numero(i)=marcado(F1,F2); % Número correspondiente a esas frecuencias end</pre>

Función “marcado” Matlab

```
function digito = marcado(F1,F2)
F1 = [697 770 852 941]; %Frecuencias fila
Fh = [1209 1336 1477 1633]; %Frecuencias columna
%Matriz de 4X4 con los dígitos del sistema DTMF
numero = [ '1','2','3','A';
            '4','5','6','B';
            '7','8','9','C';
            '*','0','#','D'];

%Seteamos los valores de columna y fila en cero
columna = 0;
fila = 0;
%Algoritmo de ubicación de digito
for i=1:4
    if (F1(i)*0.988 < F1) && (F1 < F1(i)*1.018)
        fila = i;
        break
    end
end
for j=1:4
    if (Fh(j)*0.988 < F2) && (F2 < Fh(j)*1.018)
        columna = j;
        break
    end
end
if columna == 0 || fila == 0
    digito = ['/'];
end
digito = numero(fila,columna); %Valor devuelto por la función
```

A continuación, se puede observar la gráfica de la transformada de fourier de una de las subdivisiones de la señal, en particular la 3er subdivisión.

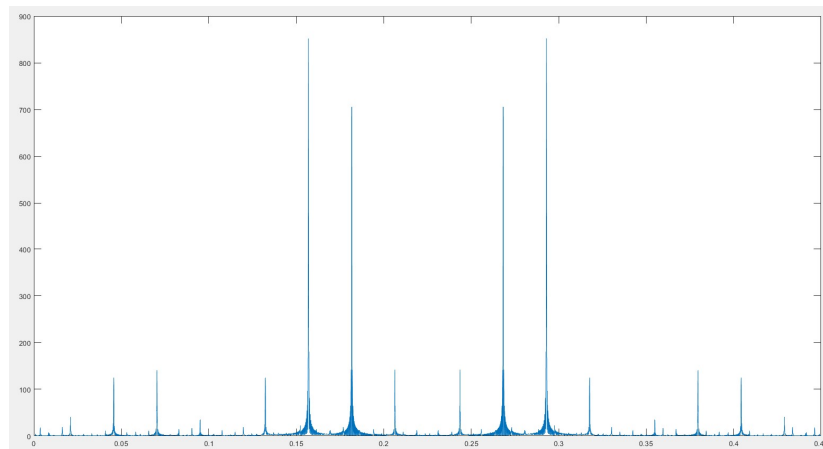


Figura 8. Ploteo del 3er frame de la señal transformada.

Una vez ejecutado el script, se puede apreciar que a la señal presente en “tonos.wav” le corresponde el número '0341197645328/'. Se puede pensar que el carácter '/' al finalizar el número corresponde al fin del marcado o una señal ajena a la telefónica ya que su frecuencia no se encuentra en el rango de alguno de los dígitos.

Ahora se intentará verificar cómo se modifica dicho número al pasar la señal por un filtro con respuesta al impulso $h(n)$ obtenida a partir del comando `h = fir1(80,0.325)`, cuya respuesta en frecuencia se puede ver a continuación.

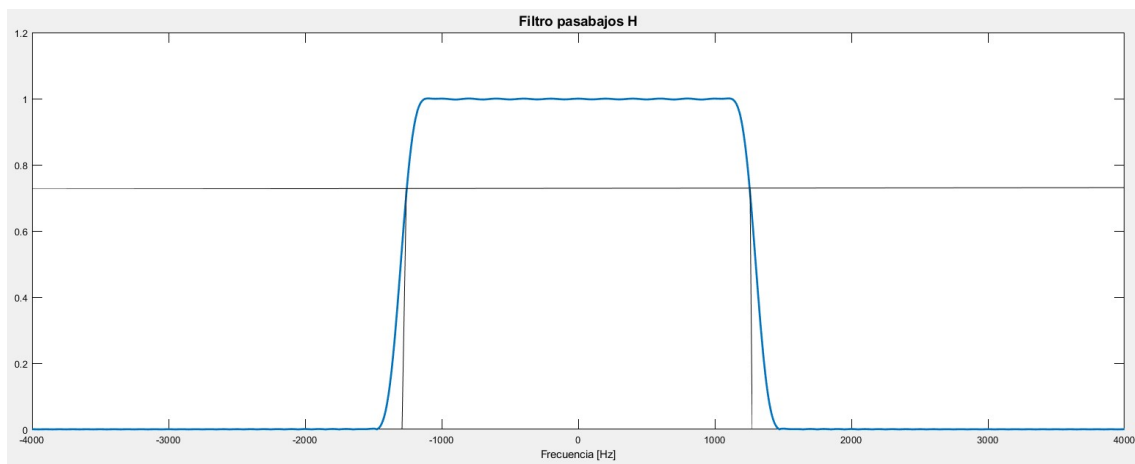


Figura 9. Filtro H del canal de transmisión

A partir del trazado de líneas en $\frac{1}{\sqrt{2}}$, este filtro se puede considerar un filtro pasa bajos con una frecuencia de corte próxima a 1250Hz. Sin embargo, su amplitud cae a cero recién para frecuencias cercanas a los 1400Hz. La acción de este filtro nos podría llevar a pensar que va a haber frecuencias que se van a perder, produciendo que no podamos identificar algunos números.

En la siguiente figura se puede observar una superposición del filtro aplicado (escalado en 1:1500) junto a la señal de entrada y como resultado del producto de ambos, la señal Y filtrada.

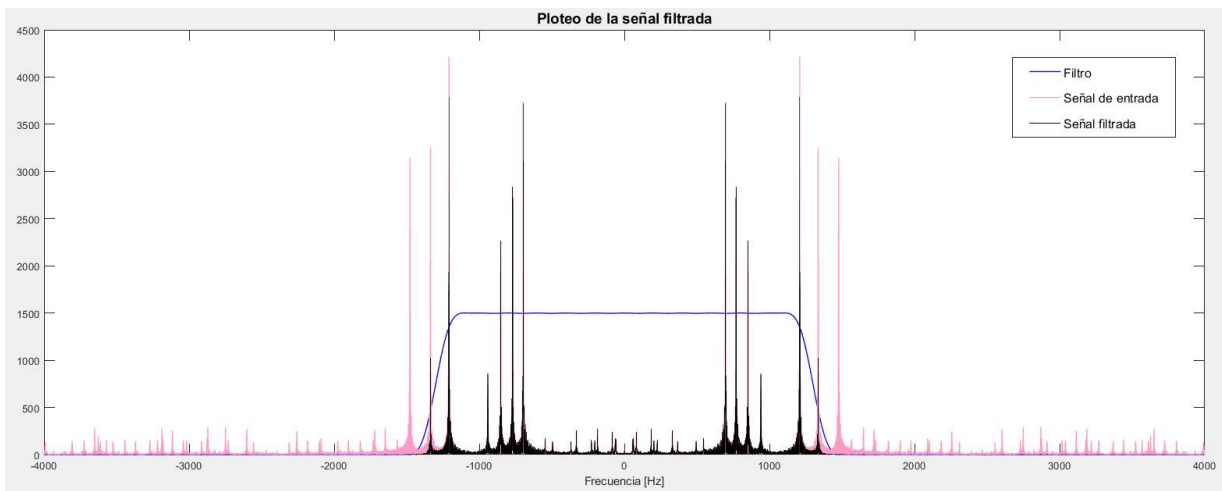


Figura 10. Superposición del filtro, señal original y señal filtrada.

Como se ha dicho, se puede observar como la señal filtrada (en negro), solo tiene picos en las componentes frecuencias donde el filtro lo admite, y pierde importancia en 2 de las otras frecuencias, entre ellas la correspondiente a 1477 Hz.

Para corroborar esto se volverá a repetir los pasos anteriores, con la salvedad de que ahora las frecuencias son calculadas con las subdivisiones de la señal convolución con el filtro (o la transformada de las primeras por la transformada del segundo).

Script "Problema2e2.m" Matlab

```
[Y, Fs] = audioread('tonos.wav'); % Subida del archivo
N = 2^16; % N > L tal que 2^X > L
L = (450e-3)*Fs; % Longitud de cada frame
Nframes = length(Y)/L; % Número de frames en total
F1000 = 1000*L/Fs; % Posición correspondiente a los 1KHz
F = [0:Fs/L:Fs-Fs/L]';

h = fir1(80,0.325); %Respuesta al impulso
H = fft(h,L); %Filtro H de longitud L

Liminf=0; %Limite inferior

for i=1:Nframes
    Limsup=L*i; % Límite superior de la subdivisión
    Yx(:,i)=Y(Liminf+1:Limsup , 1); % Subdivisión de Y en los frames
    Liminf=L*i; %Límite inferior de la subdivisión

    Yfiltrada(:,i)= fft(Yx(:,i)).*H'; % Transformada de las subdivisiones y
    filtrado

    [valor1,pos1]=max((Yfiltrada(1:F1000,i))); % Máximo antes de los 1KHz
    [valor2,pos2]=max(Yfiltrada(F1000:(L/2),i)); % Máximo después de los 1KHz
    F1=pos1*Fs/L;
    F2=(pos2+F1000)*Fs/L;
    numero(i)=marcado(F1,F2); % Número correspondiente a esas frecuencias
end
```

El número calculado es: **número = 0 / 4 1 1 / 7 5 4 5 / 2 8 /**.

Se puede ver que es diferente al calculado antes. (0 3 4 1 1 9 7 6 4 5 3 2 8 /). En particular, se nota la falta de números de la tercera columna (3,6,9) debido a que la frecuencia para identificarlos es 1477Hz, la cual se encuentra en una zona donde la amplitud del filtro $H(F) = 0$, por lo que no es posible identificar a ese decimal por medio de la tabla. Para solucionar este tipo de problemas, debería hacerse un filtro pasabajos, pero con una frecuencia de corte mayor.

Conclusiones.

Tanto en el primer como segundo problema, se aprendió a manipular señales mediante el uso de la transformada de Fourier, filtrado y uso de ventanas en Matlab.

En el primer problema se logró muestrear una señal a partir de una determinada frecuencia de muestreo para luego, mediante un filtro frecuencia pasa banda, separar una señal ADSL en sus 3 componentes respectivas y recuperar la banda de señal de voz para reproducirla.

En el segundo problema, se implementó un análisis de Fourier en tiempo corto mediante el uso de ventanas, necesario para señales de larga duración con el fin de no perder información temporal del contenido frecuencial de la misma. Para ello se utilizó la transformada de Fourier de tiempo corto a partir de la cual se pudo recuperar una señal de marcado como lo haría una central telefónica. Debido a que el rango de frecuencias de los dígitos es conocido, es posible la realización de un filtro pasa banda o pasa bajos en este caso, para filtrar las señales recibidas de frecuencias distintas a las esperadas.

Se destaca, en general, que es interesante respetar las diferencias de las señales tanto en los dominios temporales como frecuenciales y también su relación entre ellos mediante el uso de la transformada de Fourier.