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

A3 - Sistemas y Señales I

Trabajo Práctico Nº 3

"Análisis frecuencial de señales"

Autores: DIALE, Guillermo Daniel (D-3812/1)

FERRERO, Ulises (F-3284/1)

Realización: 7 de Junio de 2018.
Entrega: 24 de Julio de 2018.

INTRODUCCIÓN:

El propósito de este trabajo práctico es desarrollar e implementar algoritmos para el procesamiento de señales de distinto tipo, como ser ADSL o señales de audio, manipulando la información que estas portan en el dominio frecuencial. Para ello se aplican los conceptos de Transformada de Fourier en tiempo discreto (DTFT) vistos en clases teóricas.

Por otra parte se pretende seguir familiarizándose con la herramienta MatLab, aplicando conceptos y funciones ya conocidos, a la vez que se incorporan nuevos, ampliándose asi el dominio de esta valiosa herramienta.

RESOLUCIÓN DE LOS PROBLEMAS A INFORMAR

Problema 1: procesamiento de una señal ADSL.

El objetivo de este problema consiste en aplicar filtros en el dominio frecuencial para separar las distintas bandas de frecuencia de una señal de ADSL. Se pretende además recuperar la señal de voz transmitida.

Introducción

ADSL son las siglas en inglés de *Asymmetric Digital Subscriber Line*, o en castellano "Línea Digital Asimétrica de Abonado". Esta es una tecnología en particular de las diferentes tecnologías existentes de DSL (Línea Digital de Abonado), la cual permite una mayor tasa de transmisión de datos que en el caso de los antiguos módems telefónicos mediante el uso de frecuencias superiores a las del canal telefónico. Adicionalmente los sistemas de ADSL permiten simultáneamente transmitir información de datos y voz en distintas bandas de frecuencia.

El rango de frecuencias utilizado (0 a 1.104 MHz) se divide en 256 canales de 4.3125 KHz. Generalmente los canales del 1 al 6 (hasta 25.875 KHz) se utilizan sólo para telefonía analógica (señal de voz). Los canales 7 al 31 (hasta 138 KHz) se utilizan para enviar datos (**UPSTREAM**) mientras que los canales del 32 al 256 (hasta 1.104 MHz) se reservan para la recepción de datos (**DOWNSTREAM**).

A) Nos proponemos encontrar la mínima frecuencia necesaria para muestrear correctamente la señal de datos ADSL. La componente de mayor frecuencia en este tipo de señales es de 1104Khz, teniendo en cuenta esto y el teorema de muestreo que plantea que, como mínimo, la frecuencia de muestreo debe ser el doble de la mayor componente en frecuencia presente en la señal a muestrear, se adopta como frecuencia de muestreo para este caso Fs=2208 Khz.

Con el objetivo de visualizar la evolución temporal de la señal ADSL contenida en el archivo 'datosADSL.mat' se escribió y ejecutó el siguiente script que carga la información contenida en el archivo antes mencionado y lo muestra en una gráfica. Arrojó como resultado la gráfica que puede verse en la (Figura 1).

```
load('datosADSL.mat');

Fmax=1.104e6;
Fs=2*Fmax;

t=[0:1/Fs:(length(datosADSL)-1)/Fs];

plot(t,datosADSL)
title('Señal ADSL');
xlabel('Tiempo [s]');
ylabel('Señal ADSL')
grid on
```

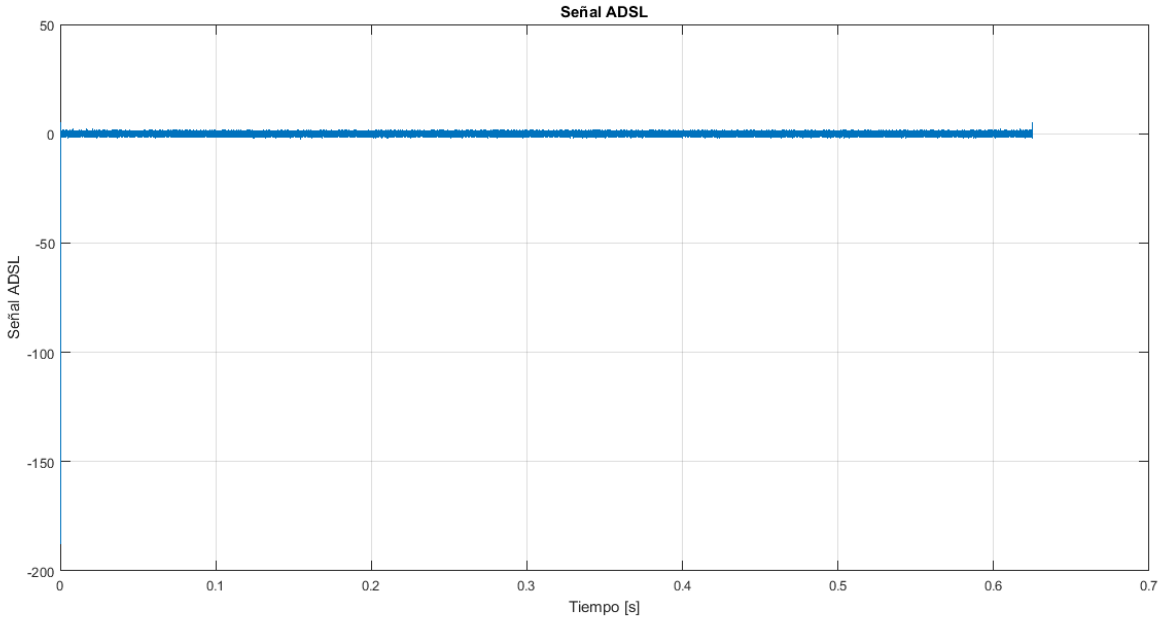

Figura 1: evolución temporal de la señal contenida en datosADSL.mat

Algunos puntos con valores extremos no permiten apreciar el contenido de la señal por una cuestión de rangos en el gráfico, es por ello que en la (Figura 2) se muestra un acercamiento sobre una porción de la señal.

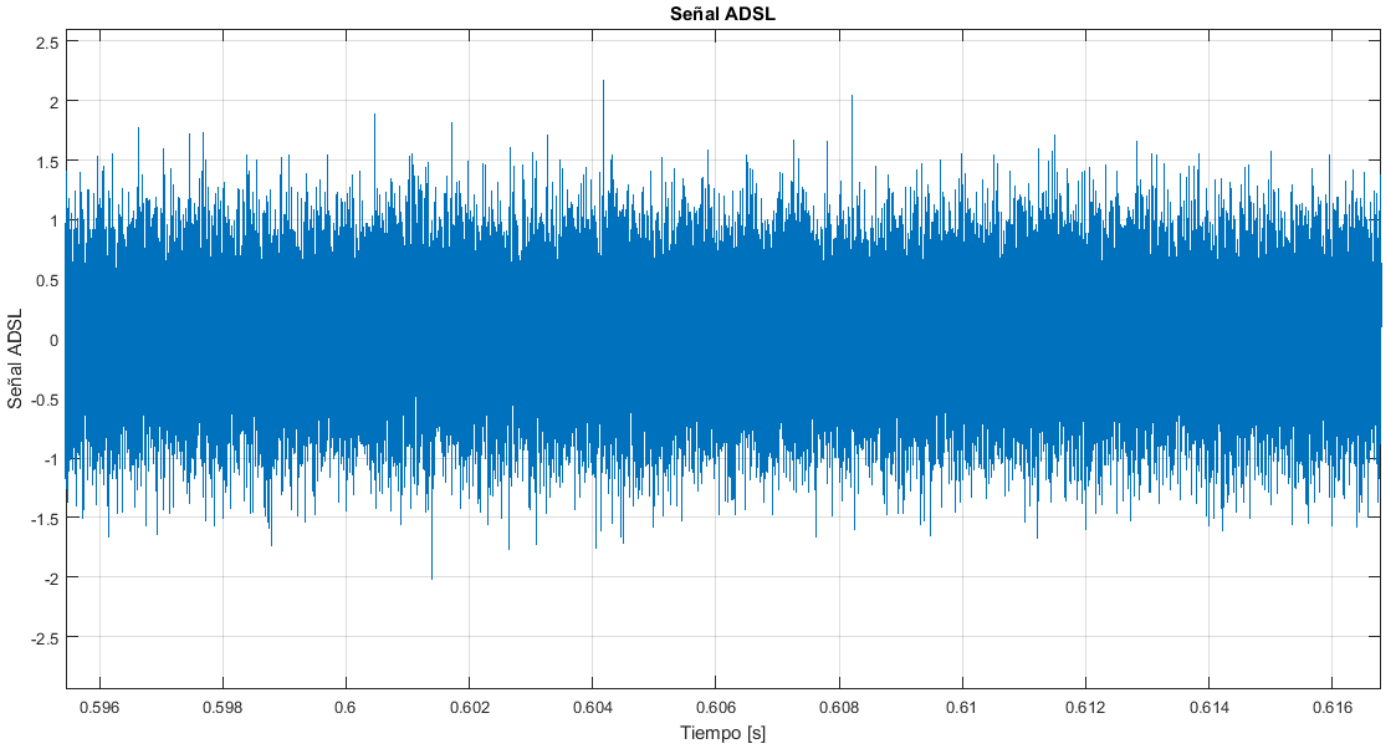

Figura 2: acercamiento al gráfico de la señal datosADSL.mat

B) Para obtener el espectro en frecuencia de la señal se generó el script que se muestra a continuación, donde se elige un número de muestras "N" potencia de 2, suficientemente grande (mayor a la cantidad de muestras de la señal L=1380275) y se le aplica la transformada rápida de Fourier a la señal, obteniéndose el espectro de amplitud que se muestra en la (Figura 3).

```
N=2^21;
X = fft(datosADSL,N);
F=[-Fs/2:Fs/N:Fs/2-Fs/N]';

figure
plot(F,abs(X))
title('espectro de
amplitud');
xlabel('frecuencia [Hz]');
grid on
```

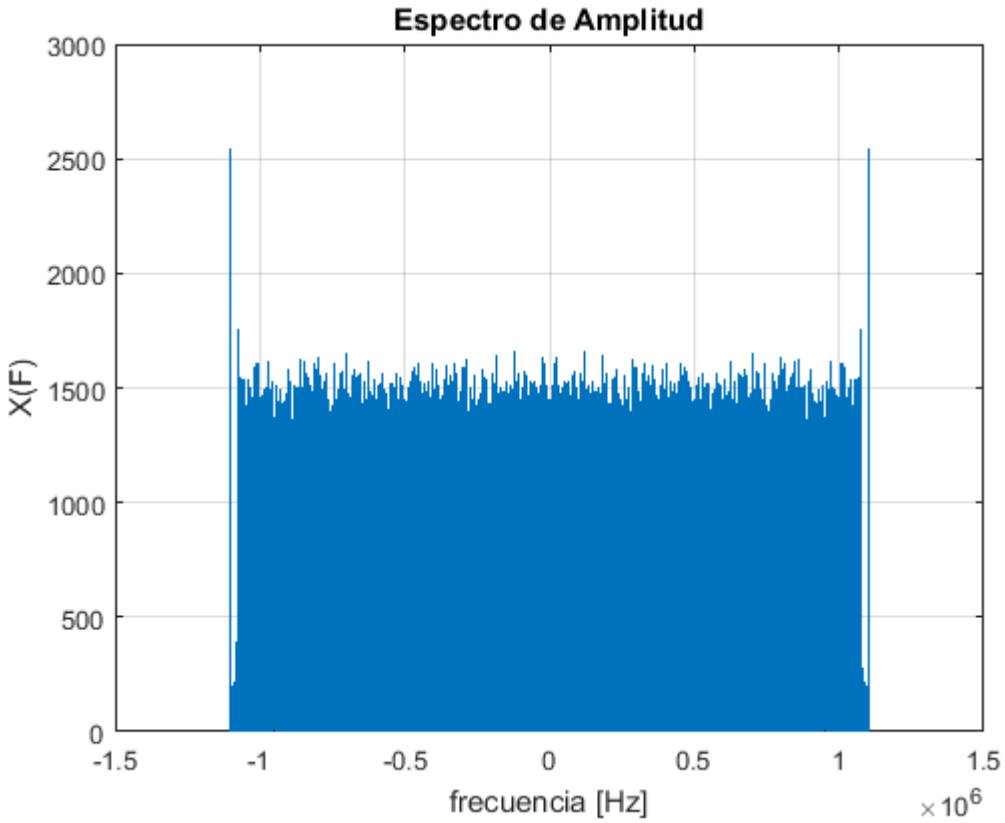

Figura 3: Espectro de amplitud de la señal datosADSL en función de la frecuencia en Hz

Se observa que el contenido frecuencial de la señal es bastante uniforme a lo largo del eje de frecuencias.

C) Resulta de interés discriminar los diferentes tipos de datos que componen la señal ADSL: voz, upstream y downstream. Para ello se aplican filtros en el dominio frecuencial, en función de la banda de frecuencias que corresponde a cada tipo de dato.
Para diseñar los filtros se construye una función que vale "1" entre las muestras correspondientes a las frecuencias que se quiere conservar, los valores "k" correspondientes a los límites superior e inferior de cada rango de frecuencias están dados por: $K_F = \frac{F}{Fs}.N$ y sus componentes simétricas. A continuación se detalla el script con el cual se filtran los diferentes tipos de datos que componen la señal y se los grafica por separado, obteniendo la gráfica que se ve en (Figura 4).

```
%filtro de voz
k300=ceil(300*N/Fs);
k3400=floor(3400*N/Fs);

Filtrovoz=zeros(N,1);
Filtrovoz(k300:k3400)= 1;
Filtrovoz(N-k3400:N-k300)= 1;
```

```
%filtro de upstream
k25875=ceil(25875*N/Fs);
k138k=floor(138000*N/Fs);

Filtroup=zeros(N,1);
Filtroup(k25875:k138k)= 1;
Filtroup(N-k138k:N-k25875)= 1;

%filtro de downstream
k138kd=ceil(138000*N/Fs);
k1104k=floor(1104000*N/Fs);

Filtrodn=zeros(N,1);
Filtrodn(k138kd:k1104k)= 1;
Filtrodn(N-k1104k:N-k138kd)= 1;

%grafica espectros filtrados
figure
subplot(311),plot(F,fftshift(abs(X.*Filtrovoz)));grid on
xlabel('Frecuancia [Hz]');
ylabel('Filtro de voz');
subplot(312),plot(F,fftshift(abs(X.*Filtroup)));grid on
xlabel('Frecuancia [Hz]');
ylabel('Filtro de Upstream');
subplot(313),plot(F,fftshift(abs(X.*Filtrodn)));grid on
xlabel('Frecuancia [Hz]');
ylabel('Filtro de Downstream')
```

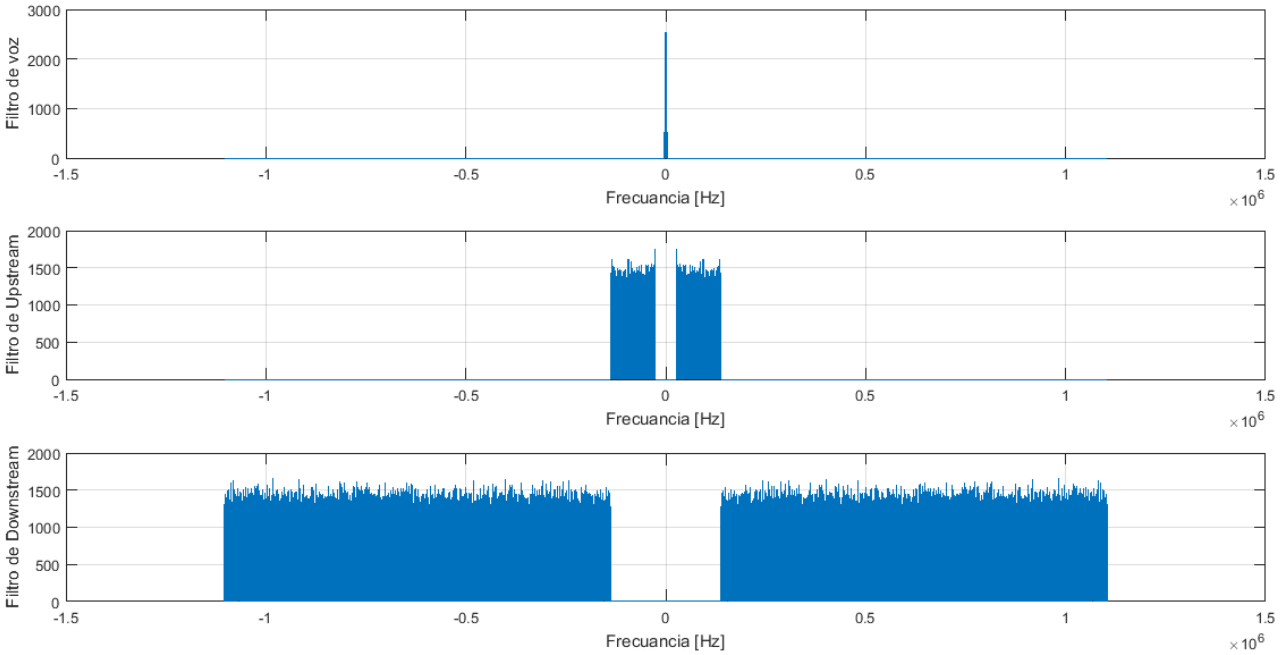

Figura 4: Distintos espectros filtrados a partir de la señal original

Las muestras correspondientes a la señal de voz están en los intervalos [2093923-2096867] y [285-3229]

D) Para obtener la señal de voz en dominio temporal se aplica la función ifft a la señal resultante de aplicar el filtro correspondiente en dominio frecuencial.
La señal esperada como resultado a partir de la ifft es una señal a valores reales, no obstante, pueden aparecer por errores de cómputo partes imaginarias en la señal temporal, para evitar este problema se toma como válida solamente la parte real de dichos datos.
A continuación se muestra el script que implemente los puntos antes mencionados.

```
VozFiltrada = X.*Filtrovoz;

voz = real(ifft(VozFiltrada));
```

E)En este punto se pretende escuchar la señal de voz extraída de la señal ADSL mediante el comando soundsc. Este comando no admite la reproducción de la señal con la frecuencia de muestreo que se pretende (Fs=2.208e6), entonces se recurre a hacer un remuestreo de la señal con el comando "resample", para adoptar ahora la frecuencia estándar que sugiere la ayuda del comando soundsc: 8192Hz. A continuación se muestra la implementación de esta solución:

```
vozresample = resample(voz,8192,Fs);
soundsc(vozresample,8192);
```

Tras ejecutar estas instrucciones se puede escuchar la palabra "Bienvenidos". Si se reproduce la señal "vozresample" con una frecuencia de la mitad de 8192 Hz, se oye que la voz pasa lento y en un tono grave. Caso contrario, si se reproduce con el doble de la frecuencia de muestreo el mensaje pasa rápido y en un tono agudo.

Problema 2: Análisis frecuencial utilizando ventanas.

El objetivo de este problema es analizar el contenido frecuencial de una señal asociada al marcado de un número telefónico, por medio de la Transformada de Fourier en Tiempo Corto, para identificar el número telefónico marcado, determinando cada uno de los dígitos que lo componen.
La Transformada de Fourier en Tiempo Corto es una técnica que se basa en dividir la información a analizar en intervalos cortos de tiempo para identificar su contenido y así, a diferencia de la transformada de Fourier convencional, no perder la información temporal de cuándo estuvo presente el espectro que se está procesando.

A) La señal a analizar está dada como un archivo de audio "tonos.wav", valiéndonos de la función wavread se cargan los datos contenidos en dicho archivo y se lo grafica, como muestra el script que sigue. El gráfico de la señal de audio obtenida se muestra en la (Figura 5), se puede ver claramente que consta de una sucesión de tonos.

```
[Y, Fs, Nbits] = wavread('tonos.wav');
t=[0:1/Fs:(length(Y)-1)/Fs];
plot(t,Y);grid on
xlabel('Señal a analizar')
ylabel('Tiempo [s]')
```

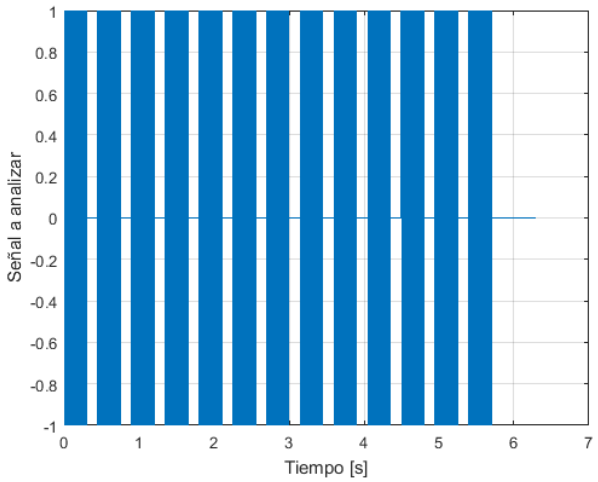

Figura 5: señal de marcado a analizar

B) Para computar la DTFT se debe elegir un número de muestras N, múltiplo de 2 y mayor a la cantidad de muestras de la señal "Y". En este caso se elige N=2^16 y se aplica la transformada rápida de Fourier mediante el script que se muestra a continuación que arroja los resultados que pueden apreciarse en (Figura 6).

```
N = 2^16;
X = fft(Y,N);
F = [-Fs/2:Fs/N:Fs/2-Fs/N];

plot(F,abs(fftshift(X))./Fs)
grid on
xlabel('frecuencia [Hz]')
ylabel('amplitud')
```

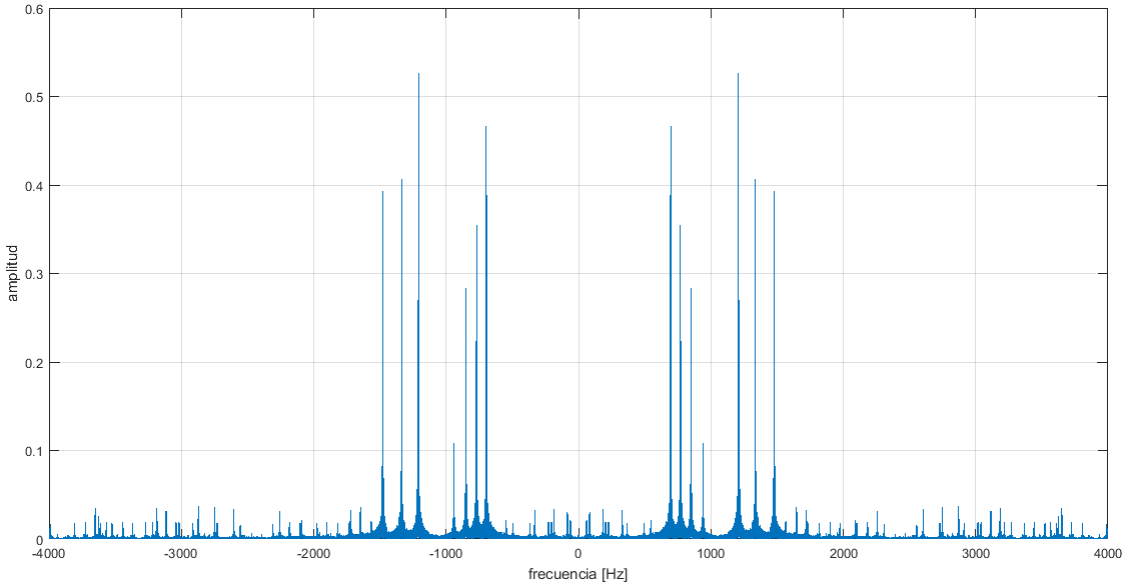

Figura 6: Transformada de Fourier de la señal contenida en "tonos.wav"

Como puede verse, en la transformada de Fourier están presentes las frecuencias de los dígitos marcados, no obstante, al trabajar con la señal completa se ha perdido la información temporal y por lo tanto no se puede identificar la secuencia en la cual dichos dígitos fueron marcados.

C) Con el fin de identificar un dígito a partir de dos frecuencias F1 y F2, con F1<F2, se desarrolló la función "IdentificarDigito" que toma dichos valores de frecuencias, los compara con los valores de la tabla de dígitos, teniendo en cuenta una tolerancia de ±1.8%, y devuelve el dígito correspondiente. En caso de que los valores ingresados no correspondan a ningún dígito devuelve "vacio". A continuación se muestra el script de esta función.

```
function [ digito ] = IdentificarDigito( F1,F2 )
%IDENTIFICARDIGITO
% toma F1 y F2 con F2>F1
% devuelve el dígito identificado

Fl=[697 770 852 941];
Fh=[1209 1336 1477 1633];

opciones={1   2   3  'A';...
          4   5   6  'B';...
          7   8   9  'C';...
         '*' '0' '#' 'D'};

columna=0;
fila=0;

for l=1:1:4
    if F1>Fl(l)/1.018 && F1<Fl(l)*1.018
        fila=l;
        break
    end
end

for h=1:1:4
    if F2>Fh(h)/1.018 && F2<Fh(h)*1.018
        columna=h;
        break;
    end
end

if fila==0 || columna==0
    digito=[];
else
    digito = opciones(fila,columna);
end
end
```

D) En este apartado se pretende identificar el número discado, para ello se sigue el siguiente procedimiento iterativo:

- Se secciona la señal "Y" en cuadros (o frames) de 0,45 segundos (0.45*Fs muestras).
- Se aplica la fft al frame
- Se identifican dos valores máximos de frecuencia: uno entre 0 y 1000 Hz, que determinará la fila del número que se discó y otro entre 1000Hz (más una muestra) y Fs/2, que determinará la columna del número discado. Se almacena el número de muestra correspondiente a cada valor máximo de frecuencia de esos dos rangos.

- Se le pasan estos dos valores de frecuencias asociados a los números de muestra del ítem anterior a la función “IdentificarDigito” del apartado C) determinando, si corresponde, el número pulsado en el frame que se está procesando.
- Finalmente, el resultado devuelto por dicha función se muestra en pantalla.

A continuación se puede ver el script que implementa este procedimiento. Se obtuvo como resultado el número discado: **0 3 4 1 1 9 7 6 4 5 3 2 8 [ ]**.

```
L =(450e-3)*Fs;
cantiter = length(Y)/L;
muestra1000hz = 1000*L/Fs;

for i=1:1:cantiter
   Xframe = fft(Y(((i-1)*L)+1:1:(i*L)));

   [valor,pos1] = max(Xframe(1:muestra1000hz));
   [valor,pos2] = max(Xframe(muestra1000hz+1:L/2));
   pos2 = pos2+muestra1000hz;

   IdentificarDigito(pos1*Fs/L, pos2*Fs/L)
end
```

E) Se supone que la señal es enviada a través de un canal de transmisión con respuesta al impulso h(n) obtenida a partir del comando h=fir1(80,0.325). Para un análisis previo se grafica la respuesta en frecuencia del canal mediante el script abajo mostrado, obteniéndose los resultados que se ven en (Figura 7).

```
[Y, Fs, Nbits] = wavread('tonos.wav');
N = 2^16;
F = [-Fs/2:Fs/N:Fs/2-Fs/N];

h = fir1(80,0.325);
H = fft(h,N);

plot (F,abs(fftshift(H))),grid on
title('Respuesta en frecuencia del canal h(n)')
xlabel('Frecuencia [Hz]')
ylabel('H(n)')
```

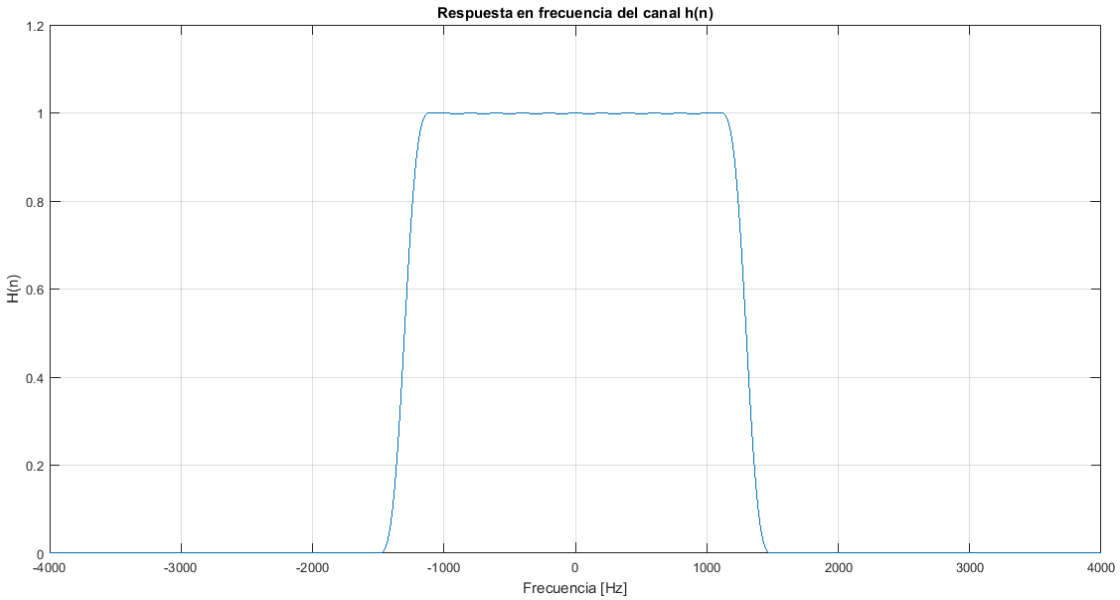

Figura 7: Respuesta en frecuencia del canal h(n)

Como puede apreciarse, el canal funciona como un filtro pasa-bajos con una frecuencia de corte de aproximadamente 1260Hz, la cual puede comprometer a las frecuencias correspondientes a las últimas columnas en la tabla de discado.

Tras pasar por el filtro, la señal filtrada en dominio temporal se obtiene como Y*h(n), que en dominio frecuencial se corresponde con un producto entre el espectro del frame "Xframe" y el espectro del canal calculado para L muestras "H". Es por ello que se modifica el script del punto D) para ahora analizar la señal filtrada con el fin de identificar el número marcado. A continuación se muestra dicho script:

```
 [Y, Fs, Nbits] = wavread('tonos.wav');

N = 2^16;
X = fft(Y,N);
F = [-Fs/2:Fs/N:Fs/2-Fs/N];

L=(450e-3)*Fs;
cantiter=length(Y)/L;
muestra1000hz=1000*L/Fs;

h = fir1(80,0.325);
H = fft(h,L)';

for i=1:1:cantiter
   Xframe=fft(Y(((i-1)*L)+1:1:(i*L))).*H;

   [valor,pos1]= max(Xframe(1:muestra1000hz));
   [valor,pos2]= max(Xframe(muestra1000hz+1:L/2));
   pos2=pos2+muestra1000hz;

   IdentificarDigito(pos1*Fs/L, pos2*Fs/L)
end
```

Tras ejecutar este script se obtuvo el siguiente resultado: **0 2 4 1 1 [ ] 7 5 4 5 2 2 8 [ ].**

Si se lo compara con el resultado original: 0 3 4 1 1 9 7 6 4 5 3 2 8 [ ] se puede ver que los números correspondientes a la tercer columna (3-6-9) han sido distorsionados en el caso de los números 3 y 6 cambiándolos por 2 y 5 respectivamente, o eliminados en el caso del 9, produciendo como resultado a la salida del canal un número erróneo, diferente al que realmente se marcó.

CONCLUSIONES

Respecto al primer problema se destaca la gran utilidad del trabajo con señales en el dominio temporal y cómo aplicando un simple filtro se puede separar a partir de una señal compleja, como la de ADSL, obtener la información útil que estamos buscando, en este caso, la señal de voz. Cabe remarcar la importancia de la elección adecuada de las frecuencias de muestreo, tanto al importar la señal desde un archivo como también al reproducir una señal de audio.

Respecto al problema 2 se pudo notar la gran importancia de la Transformada de Fourier en Tiempo corto cuando se trata de identificar el contenido frecuencial de una señal en un ejercicio parecido a lo que sería el trabajo en tiempo real, parece una herramienta de mucha importancia en muchas aplicaciones del procesamiento de señales. Se concluye además la importancia de tener presente el modelo del canal por donde se transmiten datos, ya que puede distorsionar y perder información.

Como conclusión general valoramos mucho el fortalecimiento de las bases del manejo de MatLab, una herramienta que se muestra útil en todo tipo de situaciones que involucren el procesamiento de datos. Por otra parte la aplicación a problemas ingenieriles de los contenidos teóricos nos fue útil para reforzarlos y valorar la importancia del análisis frecuencial de señales.