

IEE239 - Procesamiento de Señales e Imágenes Digitales

Laboratorio 2 - Ejercicios Propuestos

Segundo Semestre 2017

1. El siguiente ejercicio se enfoca en analizar señales a partir de su espectro de frecuencia. El procedimiento hace uso de la rutina **dtmf_spectrum()**¹, escrita especialmente para este ejercicio. La función recibe como entrada una señal en tiempo discreto y la frecuencia de muestreo F_s a la que fue adquirida. A partir de ello, **dtmf_spectrum()** genera una gráfica de su **espectro de magnitud**. El script incluido muestra el uso de la rutina al ingresar una secuencia sinusoidal:

$$x[n] \triangleq x_c(nT) = \cos(80\pi nT),$$

para $T = \frac{1}{1200}$ s, $n \in \{0, \dots, 89\}$. Según la identidad de Euler, la señal puede re-expresarse:

$$x_c(t) = \frac{1}{2}e^{j80\pi t} + \frac{1}{2}e^{-j80\pi t},$$

por lo que su espectro de magnitud se compone de pesos correspondientes a las frecuencias 40 y -40 Hz. La Figura 1 muestra las gráficas resultantes, en donde los pesos descritos son coherentes con la identidad de Euler.

```
1 n_v= 0: 299; % espacio de muestras
2 Fs= 1200;    % frecuencia de muestreo
3 x01_v= cos(80* pi/ Fs* n_v);    % señal de interes
4
5 fig_ejemplo= figure;
6 subplot( 2, 1, 1); stem( n_v, x01_v, 'r'); title( 'secuencia x[n]');
7 subplot( 2, 1, 2); dtmf_spectrum( x01_v, Fs); title( 'espectro de ...
    magnitud');    % espectro de magnitud
```

Para el ejercicio, se hará uso de señales **Dual-Tone Multi-Frequency (DTMF)**, típicamente asociadas a sistemas de telefonía convencional para enviar información de control a través de la línea telefónica. Bajo este modelo, cada tecla del teléfono genera una señal DTMF única, la cual está compuesta por dos **señales sinusoidales** con frecuencias distinguibles y libres de ambigüedad. Esto permite enviar comandos bien establecidos como señales de audio, asegurando que no haya interferencias y aprovechando el mismo canal por el cual se transmite la voz del usuario.

La Figura 2 muestra las frecuencias de ambas sinusoidales para diferentes teclas según el modelo DTMF. Como un ejemplo simple, la tecla 1 genera dos tonos sinusoidales que son superpuestos y transmitidos por el canal de voz: uno de frecuencia 697 Hz y otro de 1209 Hz. Por otro lado, la tecla 9 genera los tonos sinusoidales 852 Hz y 1477 Hz. Entonces, dado que cada tecla tiene una combinación única de frecuencias, no es posible que se genere interferencia entre ellas.

¹El archivo está almacenado en la carpeta `'/laboratorio/lab02/'`

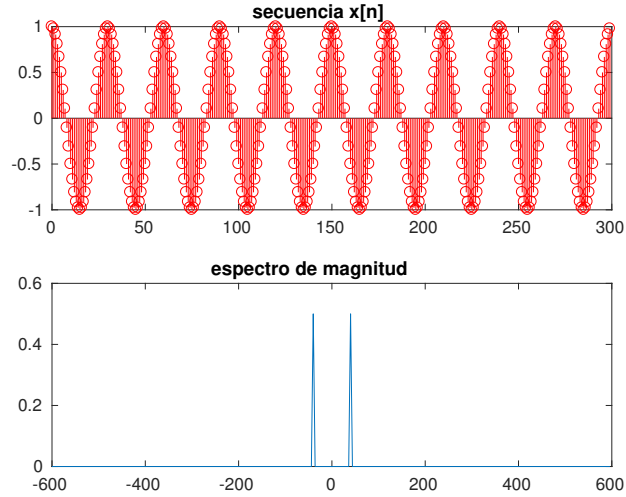


Figura 1: Señal de interés vista en espacio de muestras y en espectro de frecuencia.

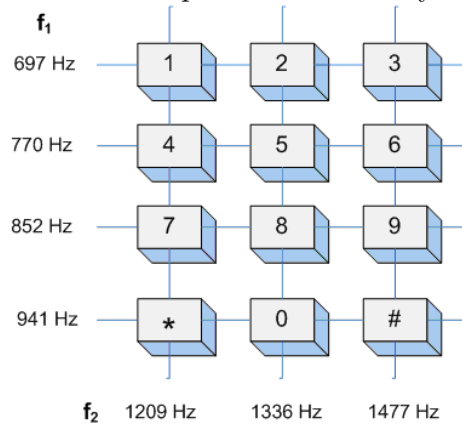


Figura 2: Sistema DTMF: distribución de los pares de sinusoidales para cada tecla y sus correspondientes frecuencias en Hz.

A partir de estos conceptos, realizar lo siguiente:

- a. Crear un sistema que permita generar tonos DTMF a partir de una secuencia de números. Para ello, realizar el siguiente procedimiento:
 - I. Generar los tonos DTMF para los dígitos del 0 al 9. Para ello, usar la información en la Figura 2 y generar la superposición de los dos tonos que le corresponden a cada dígito de interés. Dado que los dígitos serán almacenados en memoria, es necesario muestrearlos. Usar una frecuencia de muestreo $F_s = 44100$ Hz. Además, asumir que cada señal tiene una duración de 0.25 segundos.
 - II. Desarrollar un script que reciba un string de 7 dígitos enteros entre los valores de 0 a 9. El sistema debe generar un mensaje de error en caso no se cumpla con estas restricciones.
 - III. Crear un vector que almacene la secuencia de dígitos recibidos por el sistema en su forma de tono DTMF. El vector resultante debe incluir de forma ordenada las 7 señales de 0.25 segundos cada una. No deben haber vacíos entre tonos.
 - IV. Luego de creado el vector de tonos, reproducir la secuencia generada.

- b. Crear un sistema que permita decodificar los dígitos que han sido presionados. Para ello, desarrollar el siguiente procedimiento:
- I. Se cuenta con la señal en tiempo discreto '**dtmf01.wav**'², la cual contiene 7 tonos DTMF desconocidos. Leer la secuencia y la frecuencia de muestreo a la cual fue capturada y generar una matriz con los 7 intervalos de 0,25 segundos que la componen. Asumiendo que no hay silencios entre tonos, cada fila de la matriz deberá almacenar un tono DTMF. Luego, describir gráficamente el espectro de magnitud de cada tono DTMF de la matriz a partir de **dtmf_spectrum()**. Por inspección visual del espectro de magnitud, se puede identificar de qué tecla se trata en cada caso? Habría sido más fácil identificarla analizando directamente en espacio de muestras? Justificar claramente su respuesta.
 - II. Para realizar la clasificación de forma automática, crear un conjunto de sistemas LTI que preserven una de las 7 posibles sinusoidales que pueden aparecer en cada segmento almacenado. con la información de la Figura 2, se sabe que los tonos que pueden aparecer en la señal pueden ser de frecuencias $\{697, 770, 852, 941, 1209, 1336, 1477\}$ Hz. Por lo tanto, se requiere construir sistemas (filtros) que preserven de forma exclusiva cada una de estas sinusoidales. Para ello, se propone los siguientes filtros LTI de duración finita (FIR):
 - filtro h_1 : Preserva componentes en 697 Hz.
 - filtro h_2 : Preserva componentes en 770 Hz.
 - filtro h_3 : Preserva componentes en 852 Hz.
 - filtro h_4 : Preserva componentes en 941 Hz.
 - filtro h_5 : Preserva componentes en 1209 Hz.
 - filtro h_6 : Preserva componentes en 1336 Hz.
 - filtro h_7 : Preserva componentes en 1477 Hz.

Crear cada filtro h a partir del comportamiento deseado de su espectro de frecuencia. Para ello, usar la rutina **fir1()**, cuyos argumentos de entrada son la duración de la respuesta al impulso (duración finita) y el intervalo de frecuencias a preservar (banda de paso). Es decir, **fir1()** generará una respuesta al impulso $h[n]$ cuyo espectro tendrá las siguientes características:

$$\mathcal{F}\{h[n]\} = \begin{cases} G & , F \in [F_{\min}, F_{\max}] \\ 0 & , \text{otros casos} \end{cases},$$

donde $[F_{\min}, F_{\max}]$ es un intervalo muy cercano a la frecuencia de interés y $G > 0$.

- III. Una vez creado el conjunto de sistemas, obtener su espectro de magnitud usando **dtmf_spectrum()** y verificar su comportamiento en frecuencia. Luego, obtener su respuesta ante cada segmento almacenado en la matriz de señales DTMF. Dado que se trata de sistemas LTI, obtener la respuesta a partir de la sumatoria de convolución. Tener en consideración que por **propiedad de convolución**, esta operación puede ser modelada como un producto entre los espectros de entrada y del sistema. En consecuencia, cada sistema preservará únicamente los tonos dentro del intervalo en el cual se tiene una ganancia G , mientras que el resto de componentes se multiplicará por 0 y serán totalmente eliminadas.
- IV. Para determinar en cuál de los filtros se preservó tonos, se obtendrá la energía de las secuencias resultantes. Para ello, para cada porción de señal analizada, obtener las

²La señal está almacenada en la carpeta '/laboratorio/lab02/'

siete métricas de energía de cada filtro a partir de la siguiente expresión:

$$E\{y[n]\} \triangleq \sum_{n=-\infty}^{+\infty} |y[n]|^2;$$

Donde $y[n]$ corresponde a la respuesta de un determinado sistema a la señal de interés. Una vez obtenida la energía de cada respuesta, determinar qué filtros obtienen la mayor energía y asociar el dígito de interés con dicha combinación. Describir gráficamente el espectro de frecuencia de cada respuesta para el último fragmento de la señal analizada y verificar que las componentes en los sistemas donde se obtuvo más energía son de órdenes de amplitud mayores que en el resto de sistemas.

- v. Finalmente, crear un vector con los dígitos identificados en orden e imprimirlos en línea de comandos.
- c. Evaluar la robustez del sistema DTMF ante ruido. Para ello, agregarle ruido aditivo a la señal leída de la siguiente forma:

$$b[n] = x[n] + \eta,$$

Donde η es vector del mismo tamaño de $x[n]$ con elementos aleatorios con distribución Gaussiana de media $\mu = 0$ y desviación estándar $\sigma = 3$. Usar **randn()**. Una vez modificada la secuencia de entrada, reproducir el vector resultante y decodificar la información. Aumentar el valor de desviación estándar hasta que el sistema deje de decodificar correctamente. Considera que se trata de un sistema de transmisión de datos robusto? Justificar claramente su respuesta.

- d. Reemplazar la señal DTMF leída por aquella creada con el subsistema inicial. Verificar que la decodificación es satisfactoria y la secuencia de dígitos siempre es obtenida correctamente.

2. Dadas las siguientes secuencias, se solicita:

$$\begin{aligned} x_1[n] &= (0,2)^n [u(n) - u(n-11)]; \\ x_2[n] &= (0,5)^n [u(n) - u(n-21)]. \end{aligned}$$

- a. Graficar las señales, en espacio de muestras y espectro de magnitud. Las señales tienen una duración de 2,2 segundos muestreadas con una frecuencia de muestreo de 120 Hz. Usar la función **stem()** para mostrar ambas gráficas. A continuación, se incluye una breve explicación de las funciones para obtener el espectro de magnitud de una secuencia: **fft()**, **fftshift()** y **unwrap()**.
 - i. La rutina **fft()** obtiene el espectro de frecuencia de una secuencia y lo evalúa en N puntos en el dominio ω , de acuerdo a la siguiente expresión:

$$X(k) \triangleq X(e^{j\omega}) \Big|_{\omega = \frac{2\pi}{N}k},$$

donde $k \in \{0, \dots, N-1\}$. Esto quiere decir que el espectro solo se evalúa en el intervalo $\omega \in [0, 2\pi]$. Teóricamente, se sabe que $X(e^{j\omega})$ es **de periodo** 2π , por lo que el intervalo analizado contiene toda la información relevante del espectro. Sin embargo, dado que el **rango fundamental** contempla el intervalo $\omega \in [-\pi, \pi]$ por convención, es necesario realizar modificaciones tanto en el dominio ω como en la función evaluada $X(k)$.

- II. La rutina **fftshift()** modifica el orden de $X(k)$, de tal manera que la primera mitad de elementos, correspondientes al dominio $\omega \in [0, \pi]$, y la segunda mitad de elementos, correspondientes al dominio $\omega \in [\pi, 2\pi]$ intercambien sus posiciones. Teóricamente, se sabe que este último intervalo, $\omega \in [\pi, 2\pi]$, está **fuera del rango fundamental** (Alias). Restando un valor de 2π , se demuestra que dichas posiciones corresponden al intervalo $\omega' \in [-\pi, 0]$. De esta forma, el efecto de la rutina es re-expresar $X(e^{j\omega})$ en el intervalo $\omega \in [-\pi, \pi]$ (rango fundamental).
- III. Luego, se usa la rutina **unwrap()** para modificar el dominio ω de tal manera que sea coherente con el nuevo orden de $X(k)$. Primero, se genera el dominio $\omega = \frac{2\pi}{N}k$, $k \in \{0, \dots, N-1\}$. Luego, se le aplica el mismo cambio de orden usado en $X(k)$ usando **fftshift()**. El resultado es una secuencia con un gran salto de 2π en el centro. La rutina **unwrap()** permite eliminar este salto, corrigiendo la mitad de la secuencia al introducir un offset de valor 2π . Finalmente se resta un valor de 2π para obtener el un dominio final $\omega' \in [-\pi, \pi]$.

El siguiente script muestra el uso de estas rutinas.

```

1  n_v= 0: 127;      % espacio de muestras
2  x_v= 1+ cos( pi/8* n_v)+ sin( pi/16* n_v); % secuencia de interes ...
    (componentes en \omega= 0, +-pi/8, +-pi/16)
3
4  N= length( n_v); % puntos a evaluar en frecuencia
5  X_v= fft( x_v)/ length( n_v); % transformada de Fourier de f
6  w_v= 2* pi* ( 0: N- 1)/ N; % dominio en frecuencia
7
8  X_v02= fftshift( X_v); % X reordenada para que el elemento central ...
    del vector corresponda a \omega= 0
9  w_v02= fftshift( w_v); % el dominio \omega es reordenado para que ...
    el elemento central del vector corresponda a \omega= 0
10 w_v03= unwrap( w_v02 - 2*pi); % unwrap: corrige saltos de 2pi ...
    debido al reordenamiento del vector \omega
11
12 fig01= figure;
13 stem( x_v);
14 title( 'secuencia de interes');
15 fig02= figure;
16 subplot( 3, 1, 1); plot( w_v); ylabel('\omega'), xlabel( 'k');
17 title( 'vector de dominio \omega');
18 subplot( 3, 1, 2); plot( w_v02, 'r'); ylabel('\omega'), xlabel( 'k');
19 title( 'vector de dominio \omega reordenado por fftshift (salto en ...
    0)');
20 subplot( 3, 1, 3); plot( w_v03, 'g'); ylabel('\omega'), xlabel( 'k');
21 title( 'vector de dominio \omega centrado por fftshift y con fase ...
    corregida por unwrap');
22 fig03= figure;
23 subplot( 2, 1, 1); plot( w_v, abs(X_v));
24 title( 'espectro de magnitud de 0 a 2\pi (sin fftshift)');
25 subplot( 2, 1, 2); plot( w_v03, abs(X_v02), 'r');
26 title( 'espectro de magnitud de -\pi a \pi (con fftshift)'); % ...
    transformada y su dominio expresados en el rango fundamental!

```

- b. Calcular $x_3[n] = x_1[n] + x_2[n]$. Luego graficar $x_3[n]$ en espacio de muestras y espectro de magnitud.
- c. Sumar la DFT de $x_1[n]$ y $x_2[n]$ y comparar el resultado con la DFT de $x_3[n]$. Verificar que se cumple con la propiedad de linealidad. Se sugiere usar la funcion **norm()** para medir la distancia entre vectores.

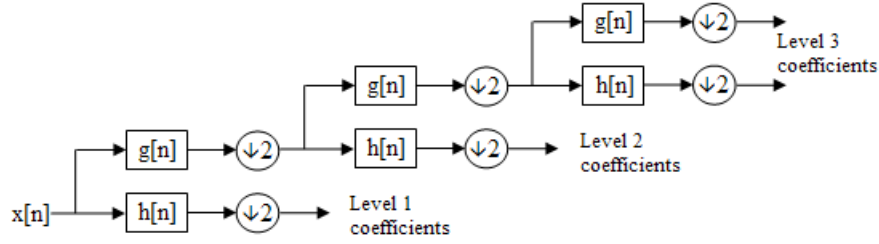


Figura 3: Banco de filtros para análisis frecuencial.

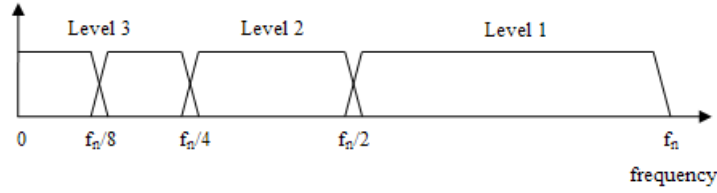


Figura 4: Frecuencias de corte para los filtros $g[n]$ y $h[n]$.

- d. Desfasar la señal $x_3(n)$ medio segundo (determinar el número adecuado de muestras que se tiene que desfasar). Graficar el espectro de magnitud de la nueva señal y generar un plot contrastando la parte real de los espectros de la señal $x_3[n]$ y la señal desfasada. Comentar como varía la parte real de la transformada de Fourier de $x_3[n]$, revisar la propiedad de desplazamiento en tiempo.
3. Analizar la señal de audio *handel.wav*³ utilizando upsampling, downsampling y filtros anti-aliasing.
 - a. Leer el archivo *handel.wav* y extraer su frecuencia de muestreo. Ver la documentación de la función **audioread**.
 - b. Los sistemas $h[n]$ y $g[n]$ vistos en la Figura 3 son filtros cuyas funciones de transferencia vienen descritas de forma analítica de la siguiente manera:

$$H(e^{j\omega}) = \begin{cases} 0, & 0 < \omega \leq \frac{\pi}{2} \\ 1, & \frac{\pi}{2} < \omega \leq \pi \end{cases}$$

$$G(e^{j\omega}) = \begin{cases} 1, & 0 < \omega \leq \frac{\pi}{2} \\ 0, & \frac{\pi}{2} < \omega \leq \pi \end{cases}$$

Siendo así, demostrar analíticamente que $H(e^{j\omega}) = 1 - G(e^{j\omega})$. Adicionalmente, demostrar que $\omega = \pi/2$ corresponde a $Fs/4$ usando la relación $\omega = \Omega T$. Incluir sus respuestas en los comentarios del código.

- c. Demostrar analíticamente que el downsampling de 2 aplicado a la salida de un filtro con frecuencia de corte en $\pi/2$ equivale a preservar las componentes de la señal hasta la frecuencia de $\pi/4$ (es decir, se reduce la frecuencia de corte a un factor de 2).
- d. Implementar el banco de filtros mostrado en la Figura 3. Para ello, considerar que a partir de las frecuencias de corte de cada sistema lineal (filtro) del sistema, se propone crear sus

³El archivo se encuentra en la misma carpeta que el enunciado de los ejercicios propuestos.

respuestas al impulso. Para ello, usar el comando **fir1**, el cual recibe como argumentos de entrada el orden del filtro y la frecuencia de corte; es decir, el rango de frecuencias en el que la transformada de Fourier tiene ganancia igual a 0dB. Revisar la documentación del comando **fir1** y asumir que cada filtro tiene una respuesta al impulso finita de duración 64 muestras ($N=64$). Adicionalmente, considerar que, siguiendo el modelo de la Figura 4, la frecuencia f_n definida en dicha figura, es la frecuencia de Nyquist $= F_s/2$ y que la gráfica se encuentra en Hz con una ganancia igual a 1. Finalmente, debe tener en cuenta que la operación de downsampling seguido del filtrado, equivale a reducir la frecuencia de corte de los filtros por un factor de 2.

NOTA: no es necesario crear filtros por cada etapa, utilizar los conceptos de upsampling y downsampling para crear todas las etapas mostradas. Utilizar los comandos downsample, upsample y conv de MATLAB.

- e. ¿Es posible tener problemas de aliasing al realizar el downsampling luego de utilizar los filtros $g[n]$ y $h[n]$?
- f. Utilizar el comando **sound** para escuchar la señal resultante luego de cada nivel. ¿Qué está ocurriendo con el audio original?