IEE239 - Procesamiento de Señales e Imágenes Digitales
Laboratorio 3 - Ejercicios Propuestos
Segundo Semestre 2017

1. Considerar el siguiente filtro pasabajos butterworth analógico:

$$H(s) = \frac{1}{(s+1)(s^2+s+1)}$$

(a) Graficar la respuesta en frecuencia analógica del filtro y estimar los valores de amplitud y frecuencia de corte.

Solución:

(a) Para ingresar el sistema $H(s)$ a Matlab primero debemos expresar tanto el numerador como el denominador de la función de transferencia como polinomios:

$$H(s) = \frac{1}{(s+1)(s^2+s+1)} = \frac{1}{s^3+2s^2+2s+1}$$

ahora debemos usar la función **freqs** para calcular la respuesta en frecuencia, en la función de magnitud de la respuesta en frecuencia buscamos el valor de frecuencia discreta w correspondiente al valor $-3dB$ del valor pico de magnitud.

```

% Ingresamos los coeficientes del filtro
num = 1;
den = [1 2 2 1];

% Graficamos la respuesta en frecuencia del filtro analogico
w = logspace(-1,1, 1000);
h = freqs(num, den, w);

% el vector h contiene la respuesta en frecuencia del filtro
mag = abs(h);

% ahora a partir de la magnitud de la resp. en frecuencia ...
% debemos calcular la frec que corte que corresponde a la magnitud ...
% que tiene el valor 0.707 (-3db) del valor maximo
v_max = max(mag);
aux = find(mag<0.707*v_max);

% Donde el primier elemento del vector auxiliar aux(1) contiene el valor ...
% mas cercano a 0.707*v_max
frec_corte = w(aux(1));
```

(b) Calcular la frec. de muestreo adecuada para el filtro $H(s)$ de tal forma que la frecuencia de corte del filtro digital sea 0.4π y verificar con la el gráfico de respuesta en frecuencia digital (Usar la función **freqz**).

Solución:

Para calcular la frecuencia de muestreo debemos usar la ecuación:

$$\Omega = \frac{2}{T}\tan(\frac{w}{2})$$

despejando el periodo se tiene:

$$T = \frac{2}{\Omega}\tan(\frac{w}{2})$$

y una vez hecho esto se aplica la transformación bilineal al sistema analógico

```
% La frecuencia de corte esta almacenada en la variable frec_corte ...
% calculada en la seccion anterior
w = 0.4*pi; % frecuencia digital deseada

% calculamos el periodo
T = (2/frec_corte)*tan(w/2);

% Ahora usando la transformacion bilineal
[numd, dend] = bilinear(num, den, 1/T);

% Graficamos la respuesta en freciencia digital
freqz(numd, dend);
% Como se puede observar la frecuencia de corte esta en 0.4*pi
```

(c) Graficar la respuesta en frecuencia del filtro digital.

Solución:

Una vez obtenido el periodo de muestreo (o frec. de muestreo) se utiliza la transformación bilineal para pasar del dominio de la frecuencia analógica a digital.

```
% Ahora usando la transformacion bilineal
[numd, dend] = bilinear(num, den, 1/T);

% Graficamos la respuesta en freciencia digital
freqz(numd, dend);
% Como se puede observar la frecuencia de corte esta en 0.4*pi tal como se requeria
```

(d) Graficar la respuesta al impulso digital del filtro diseñado para las primeras 100 muestras.

Solución:

Debido a que los filtros Butterworth son filtros IIR, tienen una respuesta al impulso infinita. Para graficar una cantidad finita como lo que se solicita en el inciso (d) se debe usar la función **impz**.

```
% Respuesta al impulso del filtro para 100 muestras
impz(numd, dend, 100);
```

(e) Comparar los resultados usando la función de MATLAB **butter**.

Solución:

Para el diseño de filtros MATLAB provee funciones como **butter** y luego para obtener una medida cuantitativa del error usaremos la distancia euclidiana definida por la siguiente ecuación e implementada por la función **norm**:

$$d_E(x,y) = \sqrt{\sum_{n=1}^{n}(x_n - y_n)^2}$$

donde x e y son vectores de n componentes: $x = (x_1, x_2, ..., x_n)$ y $y = (y_1, y_2, ..., y_n)$.

```
% Calculando los coeficientes con matlab
[fnumd, fdend] = butter(3, 0.4,'low');

% Para hacer una comparacion cuantitativa podemos usar la norma 2 o ...
% distancia euclidiana entre ambos vectores

err_num = norm(numd-fnumd)
err_den = norm(dend-fdend)
```

2. Se tiene la señal $d[n]$ contaminada ruido blanco gaussiano $v_1 \sim \mathcal{N}(0, \sigma^2)$:

$$x[n] = d[n] + v_1[n] \qquad \text{ó} \qquad \mathbf{x} = \mathbf{d} + \mathbf{v_1}$$

donde $\mathbf{d} = [d[n], ..., d[n-p+1]]^T$ y $\mathbf{v_1} = [v_1[n]], ..., v_1[n-p+1]$.

Para recuperar la señal original $d[n]$ a partir de la señal observada $x[n]$ se ha planteado el esquema descrito en la Figura 1, en la cual se registra la señal de ruido $v_2[n]$ y a partir de ésta estimar la señal de ruido original $v_1[n]$ con el fin de recuperar $d[n]$.

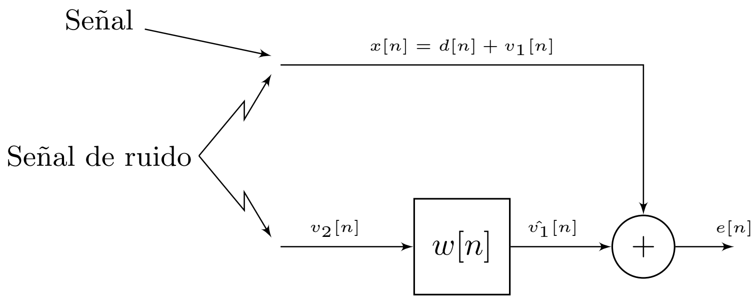

Figura 1: Sistema de cancelación de ruido.

Con lo que se pide:

(a) Escribir las ecuaciones de Wiener-Hopf para el sistema descrito en la Figura 1.
(b) Se adjunta el archivo `data.mat` el cual contiene la digitalización de las señales x y v_2 (ver Figura 1), cárgarlo y graficar las señales.
(c) Generar el sistema de ecuaciones y calcular los coeficientes del filtro Wiener para los valores de $p = 5$, $p = 8$ y $p = 12$.
(d) Estimar la señal $d[n]$ y graficar la señal observada con la señal estimada para cada uno de los valores de p mencionados en (c).

- Funciones útiles: filter, toeplitz, xcorr.

3. Considerar $H_d(e^{j\omega})$ como la respuesta ideal de un filtro pasabanda que se encuentra normalizado en el rango fundamental de $-\pi$ a π en el cual se le pide lo siguiente:

$$H_d(e^{j\omega}) = \begin{cases} 1, & 0.3\pi \leq \omega \leq 0.6\pi \\ 0, & \text{otros casos} \end{cases}$$

(a) Diseñar el filtro real pasabanda $H_1(e^{j\omega})$ utilizando el método del enventanado (*fir1*). Para ello, utilizar una ventana hanning con orden N=100. Graficar su espectro de frecuencia en la escala logaritmica (dB). y el espectro de fase en el rango de $-\pi$ a π.

Solución:

(a) Para diseñar el filtro pasa banda por el método del enventanado, se utilizará la función fir1 de MATLAB. Esta función tiene como parámetros el orden del filtro y el corte.

```
    % Orden del filtro
    N=100;
```

```
% Establecemos frecuencias de corte
fc=[0.3,0.6];
% Realizamos el filtrado por enventanado
b1=fir1(N,fc);
A1=freqz(b1);
% Normalizacion de frecuencias
n_size=2*size(A1,1);
w= 2*pi* ( 0: ( n_size- 1))/ n_size;
w= fftshift( w);
w= unwrap( w - 2*pi);
P1=[A1' A1'];
% Espectro de magnitud y fase
subplot(121),plot(w,20*log10(abs(P1)))
subplot(122),plot(w,unwrap(angle(P1)))
```

(b) Diseñar el filtro real pasabanda $H_2(e^{j\omega})$ utilizando el método de muestreo en frecuencia. Para ello, utilizar un orden de $N = 100$ y 4 posiciones de frecuencia. Graficar espectro de frecuencia en el rango de $-\pi$ a π en la escala logaritmica (dB). Comentar cuál es la diferencia entre el lóbulo principal y los lóbulos laterales con respecto al filtro del inciso (a).

Solución:

(b) Para diseñar el filtro pasa banda por el método de muestreo en frecuencia, se utilizará la función fir2 de MATLAB. Esta función tiene como parámetros el orden del filtro, el vector de frecuencias de corte y el número de muestras.

```
%Orden del filtro
N=100
% Vector de frecuencias de corte del filtro
f21=[0 0.3 0.6 1];
% Vector de amplitud de filtro
m21=[0 1 1 0];
% Diseno del filtro
b2=fir2(N,f21,m21);
A2=freqz(b2);
% Normalizacion de frecuencias
n_size=2*size(A2,1);
w= 2*pi* ( 0: ( n_size- 1))/ n_size;
w= fftshift( w);
w= unwrap( w - 2*pi);
P=[A2' A2'];
subplot(121),plot(w,20*log10(abs(P))),hold on
subplot(122),plot(w,unwrap(angle(P)))
```

(c)Diseñar el filtro real pasabanda $H_d(e^{j\omega})$ utilizando el método de muestreo en frecuencia. Para ello, utilizar un orden de $N = 100$ y 10 posiciones de frecuencia. Graficar espectro de frecuencia en el rango de $-\pi$ a π en la escala logaritmica (dB). Comentar qué filtro diseñado por muestreo en frecuencia a logrado mayor similitud con respecto al filtro por enventanado. Para responder a ello, considerar los valores del lóbulo principal y los lóbulos secundarios.

Solución:

(c) Este diseño se resuelve de forma similar al inciso anterior, se aumentan las posiciones en frecuencia

```
%Orden del filtro
N=100
% Vector de frecuencias de corte del filtro
f22=[0 0.3 0.3 0.3 0.3 0.6 0.6 0.6 0.6 1];
% Vector de amplitud de filtro
m22=[0 0 0 1 1 1 1 0 0 0];
% Diseno del filtro
b3=fir2(N,f22,m22);
[A3,w3]=freqz(b3);
% Normalizacion de frecuencias
n_size=2*size(A3,1);
w_H02= 2*pi* ( 0: ( n_size- 1))/ n_size;w_H02= fftshift( w_H02);
w_H02= unwrap( w_H02 - 2*pi);
P=[abs(A3') abs(A3')];
subplot(121),plot(w,20*log10(abs(P))),hold on
```

```
16      subplot(122),plot(w,unwrap(angle(P)))
```

4. Se le brinda la siguiente señal discreta:

$$x[n] = sin(0.2\pi n + \pi/4) + \cos(0.8\pi n + \pi/2)$$

Se le pide lo siguiente:

(a) Crear la señal $x[n]$ considerando que n $\in$ [0,99]. Graficar en una misma ventana(*subplot*) la señal en tiempo, su espectro de magnitud (dB) y su espectro de fase.

(b) Diseñar un filtro FIR pasabajos por el método de enventanado(*fir1*) utilizando una ventana blackman, frecuencia de corte el valor de 0.4 π y un orden de filtro $N = 50$. Graficar su espectro de magnitud (dB) y de fase en el rango de $-\pi$ a π.

(c)Convolucionar (*conv*) la señal con el filtro con la señal $x[n]$ y obtener su salida. En una misma ventana (*subplot*) graficar la salida de la señal en muestras, su espectro de magnitud (dB) y su espectro de fase.

(c) Partiendo del diseño analógico de un filtro pasabajos butterworth *butter*, diseñar un filtro IIR por el método de invarianza al impulso *impinvar* considerando un orden de filtro $N = 50$, frecuencia de corte 2 Hz y una frecuencia de muestreo de 10 Hz. Graficar su espectro de magnitud (dB) y de fase en el rango de $-\pi$ a π.

(d) Utilizando el comando *filter* convolucionar el filtro con la señal $x[n]$ y obtener su salida. Graficar en una misma ventana la salida de la señal en muestras, su espectro de magnitud y su espectro de fase.

(e) Responder en los comentarios qué filtro ha logrado eliminar la componente de alta frecuencia en su totalidad. Así mismo, responder qué ha sucedido con la fase en ambos casos.