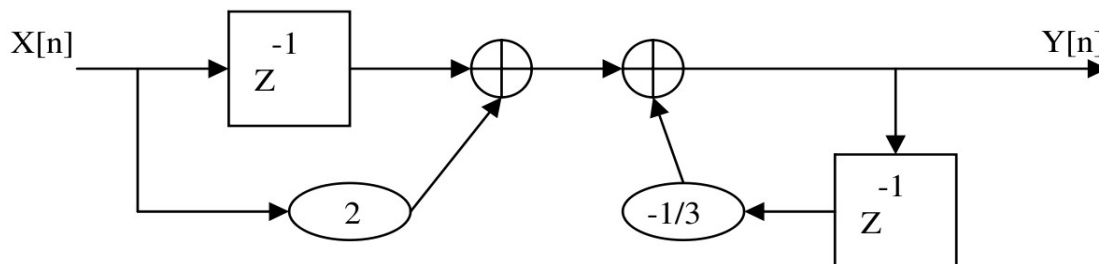




## T.P.Nº 3: Sistemas discretos - Diseño de filtros digitales.

### 3.1 Sistemas discretos - Ecuación en diferencias

**3.1.1** Dado el siguiente diagrama de un sistema discreto, obtenga la ecuación en diferencias.



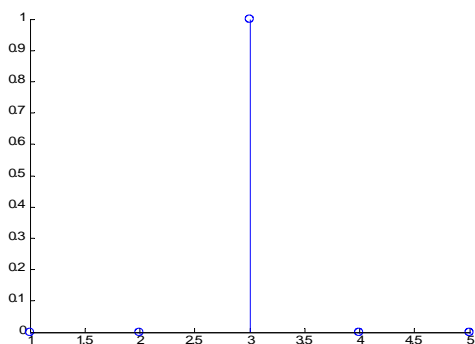
**Resolución:**

- ¿De qué orden es el sistema?  
Ecuación de 1º orden.
- ¿Quién determina el orden?

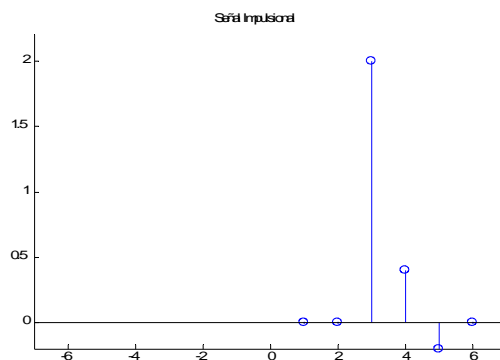
$$Y[n] = 2X[n] + X[n-1] + \frac{1}{3}Y[n-1]$$

Según el retraso que tiene la ecuación en diferencia nos da el orden de la señal  $X[n-1] = X_{(z)}Z^{-1}$  que en este caso corresponde a primer orden.

- Determine gráficamente la respuesta al impulso del sistema anterior. Utilice una señal impulso de 5 muestras de longitud.



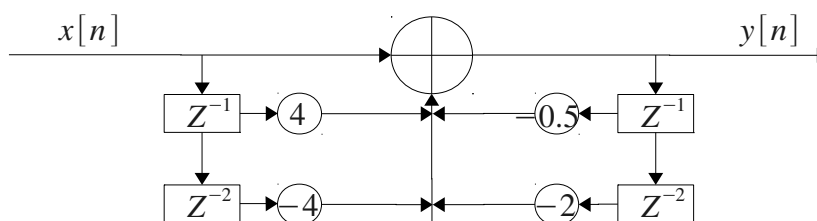
Dibujo 1: Señal impulso



Dibujo 2: Señal impulsional

**3.1.2** Dada la siguiente ecuación en diferencias obtenga el diagrama del sistema discreto.

$$y[n] - 4y[n-1] + 4y[n-2] = x[n] - 0.5x[n-1] - 2x[n-2]$$



- ¿De que orden es el sistema?

El sistema es de 2° orden ya que se observan 2 atrasos de la muestra presente.

### 3.2 Introducción a los filtros digitales: filtros IIR (Infinite Impulse Response) y filtros FIR (Finite Impulse Response).

menina.

- Realice un estudio de los filtros digitales que le permita generar un listado de ventajas y desventajas frente a filtros analógicos.

Ventajas	Desventajas
Económicos	La conversión de analógico a digital impide la reconstrucción de la señal analógica original.
De fácil implementación y modificación de sus característica, o coeficientes.	
Alta inmunidad al ruido.	No se pueden tomar infinitas muestras y esto conlleva a su sacrificio en la precisión del filtro.
Buena precisión.	
Se puede almacenar información en medios de almacenamiento masivos.	La buena precisión depende de la cantidad de bits que se utilizan para la conversiones y los coeficientes del filtro y a mayor cantidad de bits el proceso del filtrado se vuelve lento.
Confiabilidad.	

- Realice un estudio de los filtros digitales IIR y FIR para poder generar una tabla comparativa entre dichos filtros, desde el punto de vista matemático, de la fase entre la señal de entrada y de salida (en el dominio de la frecuencia), de la respuesta al impulso (en el dominio del tiempo), de la implementación mediante ecuaciones en diferencia, etc. Enumere las ventajas y desventajas de cada uno.

Característica	FIR	IIR
Respuesta en frecuencia	Es lineal para la fase	No es lineal
Matemáticamente	Alto procesamiento. Demanda memoria	No requiere muchos coeficientes, esto limita el cálculo.
Estabilidad	Son estables por su falta de recursividad	Podrían llegar a presentar algún tipo de inestabilidad
Traspaso a analógico	No tiene equivalencia en analógico	La matemática proviene de filtros analógicos como Butterworth, Chebyshev, etc.
Respuesta al impulso	Son menos sensibles y más lentos	Respuesta infinita por su realimentación



### 3.3 Análisis de filtros IIR (Infinite Impulse Response)

#### 3.3.1. En MatLab

a) Analice y describa las siguientes funciones:

- *Butter:*

la instrucción *Butter*, este comando diseña filtros Butterworth pasa-bajas, pasa-altas, pasa-bandas y rechaza bandas tanto en forma digital como analógica. Este filtro se caracteriza por una respuesta plana en la banda de transición.

La función  $[B,A] = \text{BUTTER}(N,W_n)$  retorna los coeficientes del filtro digital pasa bajo Butterworth de orden  $N$  en dos vectores de  $N+1$  posiciones,  $B$  es el numerador y  $A$  en el denominador. La frecuencia  $W_n$  debe estar normalizada, o sea que a nuestra frecuencia de corte  $W_c$  la debemos dividir por la frecuencia de muestreo  $W_s$ , de esta forma  $W_n$  siempre debe estar comprendida entre 0 y 1.

$[B,A] = \text{BUTTER}(N,W_n,'pass')$  Si  $W_n = [W_1 W_2]$  la función Butter retorna un filtro pasabanda de orden  $2N$  y la frecuencia de paso esta entre  $W_1 < W < W_2$ .

$[B,A] = \text{BUTTER}(N,W_n,'high')$  Retorna los coef. de un pasa alto.

$[B,A] = \text{BUTTER}(N,W_n,'stop')$  Retorna los coef. De un elimina banda si  $W_n = [W_1 W_2]$ .

$[Z,P,K] = \text{BUTTER}(\dots)$  La función de esta forma retorna los polos ' $P$ ' y los ceros ' $Z$ ' en dos vectores de  $N$  posiciones y también retorna la ganancia en el escalar  $K$ .

- *Cheby1*

La function  $[B,A] = \text{CHEBY1}(N,R,W_n)$  retorna los coeficientes del filtro digital pasa bajo Chebyshev de orden  $N$  en dos vectores de  $N+1$  posiciones,  $B$  es el numerador y  $A$  en el denominador. El parámetro  $R$  indica el ripple pico a pico en decibels en la banda de paso. La frecuencia  $W_n$  debe estar normalizada.

Si  $W_n = [W_1 W_2]$  la función Cheby1 retorna un filtro pasabanda de orden  $2N$  y la frecuencia de paso esta entre  $W_1 < W < W_2$ .

$[B,A] = \text{CHEBY1}(N,R,W_n)$  Retorna los coef. de un pasa alto.

$[B,A] = \text{CHEBY1}(N,R,W_n)$  Retorna los coef. de un elimina banda si  $W_n = [W_1 W_2]$ .

- *Cheby2:*

La function  $[B,A] = \text{CHEBY2}(N,R,W_n)$  retorna los coeficientes del filtro digital pasa bajo Chebyshev de orden  $N$  en dos vectores de  $N+1$  posiciones,  $B$  es el numerador y  $A$  es el denominador. El parámetro  $R$  indica la atenuación en decibels en la banda de rechazo que sera  $W_n$ . La frecuencia  $W_n$  debe estar normalizada.

Si  $W_n = [W_1 W_2]$  la función Cheby2 retorna un filtro pasabanda de orden  $2N$  y la frecuencia de paso esta entre  $W_1 < W < W_2$ .

$[B,A] = \text{CHEBY2}(N,R,W_n)$  Retorna los coef. de un pasa alto.

$[B,A] = \text{CHEBY2}(N,R,W_n)$  Retorna los coef. de un elimina banda si  $W_n = [W_1 W_2]$ .

- *Ellip:*

Es una combinación de los dos anteriores:

La function  $[B,A] = \text{ELLIP}(N,R_p,R_s,W_n)$  retorna los coeficientes del filtro digital pasa bajo Elíptico de orden  $N$  en dos vectores de  $N+1$  posiciones,  $B$  es el numerador y  $A$  es el denominador. El parámetro  $R_p$  indica el ripple pico a pico en decibels en la banda de paso y el parámetro  $R_s$  indica la atenuación en decibels que tendrá la banda de rechazo.

La frecuencia  $W_n$  debe estar entre 0 y 1 y sera la frecuencia de corte.

Si  $W_n = [W_1 W_2]$  la función Ellip retorna un filtro pasabanda de orden  $2N$  y la frecuencia de paso esta entre  $W_1 < W < W_2$ .

$[B,A] = \text{ELLIP}(N,R_p,R_s,W_n)$  Retorna los coef. de un pasa alto.

$[B,A] = \text{ELLIP}(N,R_p,R_s,W_n)$  Retorna los coef. de un elimina banda si  $W_n = [W_1 W_2]$ .



b) Obtenga las funciones de transferencias de los siguientes filtros pasa bajos:

Datos:

- La frecuencia de corte normalizada será de 0.4 y el orden del pasa bajos será de 3.
- En el caso del filtro Chebyshev (tipo 1) tome 1dB de ripple en la banda pasante.
- En el caso del filtro Chebyshev II tome 20 dB de atenuación para el ripple en la banda de rechazo.

$$\text{Butterworth} \rightarrow H_{(z)} = \frac{0.0985 \cdot Z^3 + 0.2956 \cdot Z^2 + 0.2956 \cdot Z^1 + 0.0985 \cdot Z^0}{1.0000 \cdot Z^3 - 0.5772 \cdot Z^2 + 0.4218 \cdot Z^1 - 0.0563 \cdot Z^0}$$

$$\text{Chebyshev} \rightarrow H_{(z)} = \frac{0.0736 \cdot Z^3 + 0.2208 \cdot Z^2 + 0.2208 \cdot Z^1 + 0.0736 \cdot Z^0}{1.0000 \cdot Z^3 - 0.9761 \cdot Z^2 + 0.8568 \cdot Z^1 - 0.2919 \cdot Z^0}$$

$$\text{ChebyshevII} \rightarrow H_{(z)} = \frac{0.1397 \cdot Z^3 + 0.0911 \cdot Z^2 + 0.0911 \cdot Z^1 + 0.1397 \cdot Z^0}{1.0000 \cdot Z^3 - 1.1454 \cdot Z^2 + 0.7275 \cdot Z^1 - 0.1205 \cdot Z^0}$$

$$\text{Elliptic} \rightarrow H_{(z)} = \frac{0.1868 \cdot Z^3 + 0.2036 \cdot Z^2 + 0.2036 \cdot Z^1 + 0.1868 \cdot Z^0}{1.0000 \cdot Z^3 - 0.8739 \cdot Z^2 + 0.9217 \cdot Z^1 - 0.2671 \cdot Z^0}$$

c) Utilice la función freqz para graficar la respuesta en frecuencia de cada uno de los filtros obtenidos.

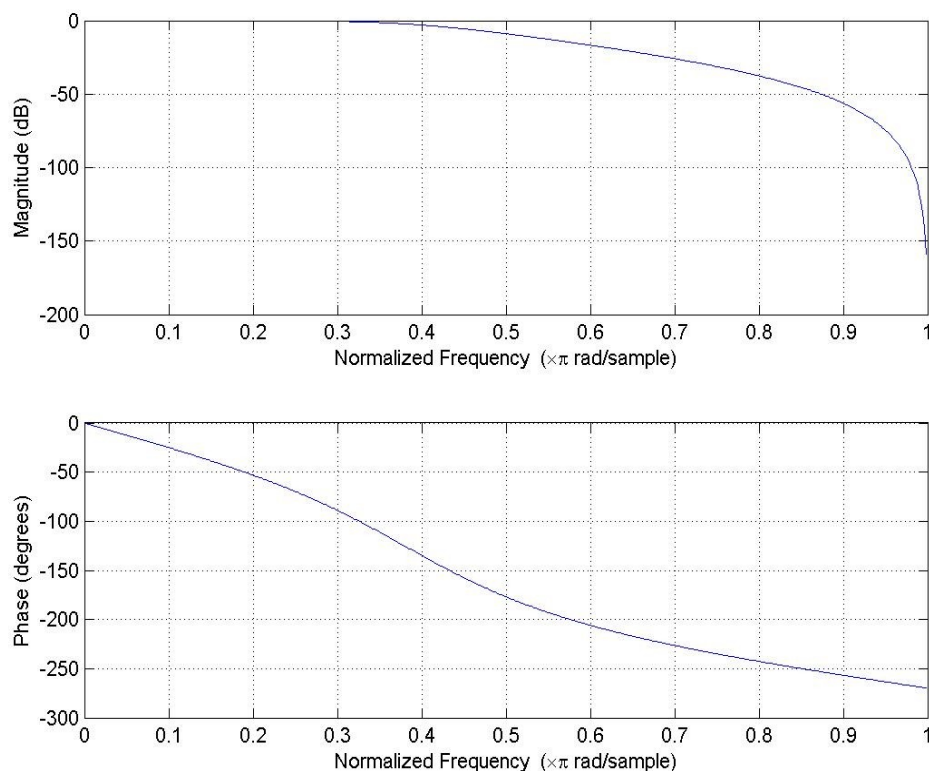


Ilustración 1: Butterworth pasa bajos

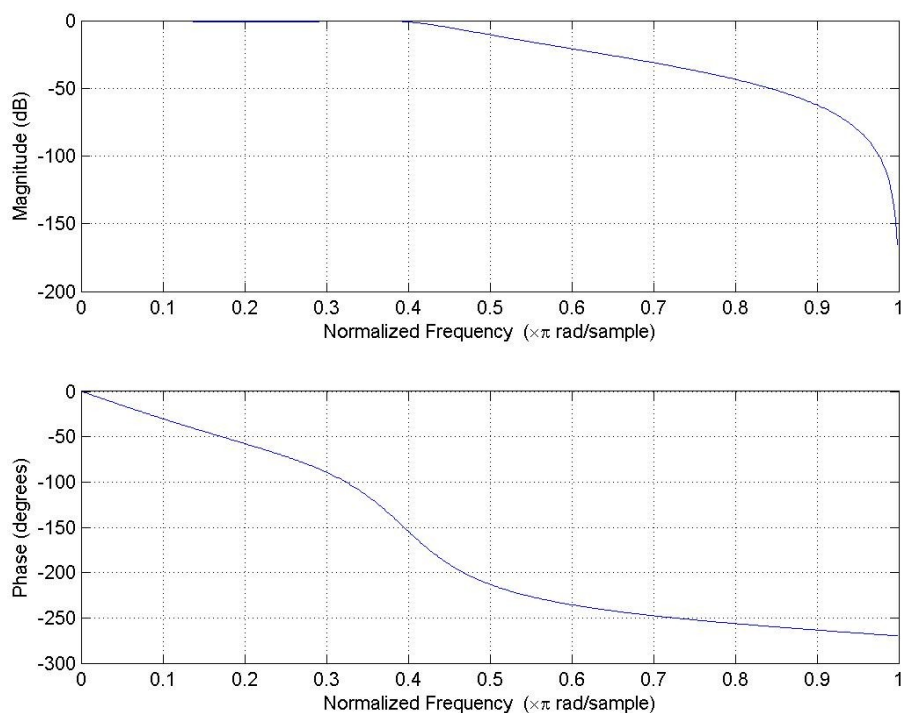


Ilustración 2: Chebyshev filtro pasa bajos

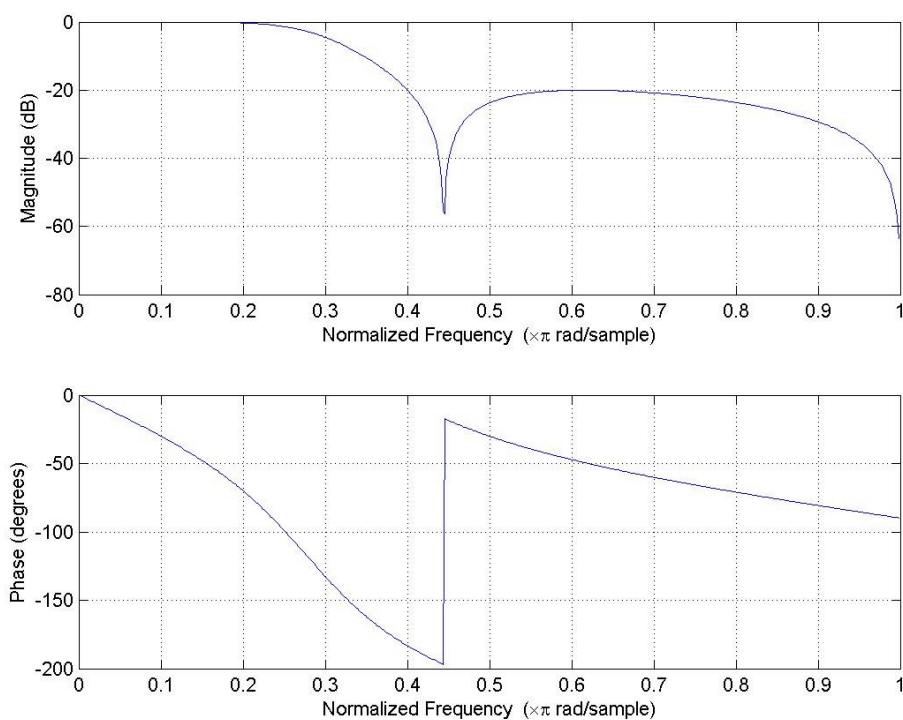


Ilustración 3: Chebyshev II filtro pasa bajos

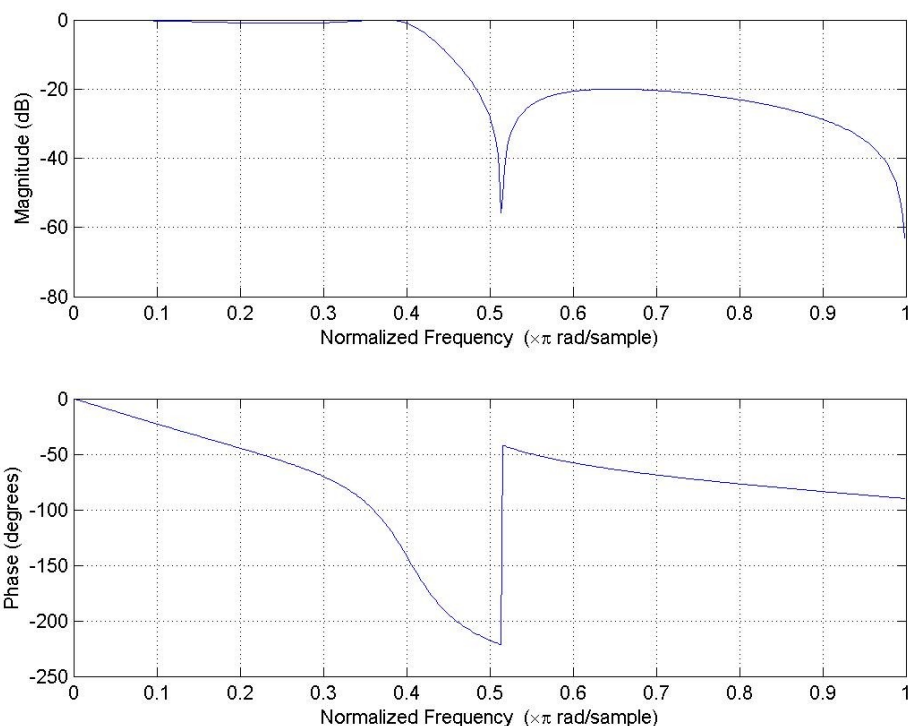


Ilustración 4: *Elíptico filtro elíptico, pasa bajos.*

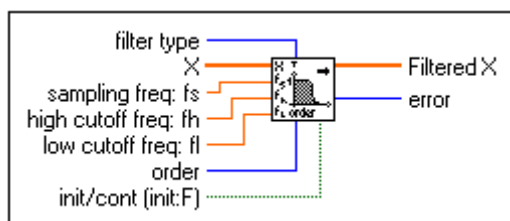
d) Realice una tabla comparativa con los resultados obtenidos.

Características	Butterworth	Chebyshev	Chebyshev II	Elliptic
Banda de paso	0 - 0,4	0 - 0,43	0 - 0,28	0 - 0,4
Banda de stop	0,87 - 1	0,84 - 1	0,43 - 1	0,51 - 1
Banda de transición	0,4 - 0,87	0,43 - 0,84	0,28 - 0,43	0,4 - 0,51
Fase en Fc [°]	-130	-153	-180	-150

### 3.3.2. En LabVIEW

a) Analice las funciones Butterworth Filter.vi, Chebyshev Filter.vi, Inverse Chebyshev Filter.vi y Elliptic Filter.vi. Las funciones antes mencionadas las encontrará en la paleta Functions/Analyze/Signal Processing/Filtres.

*Butterworth filter.vi*



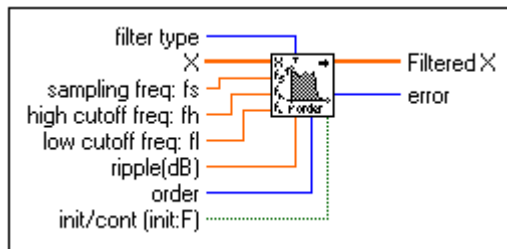
- 0: Pasa bajo
- 1: Pasa alto
- 2: Pasa banda
- 3: Elimina banda

Filter type: Se especifica el tipo de filtro por un número entero.



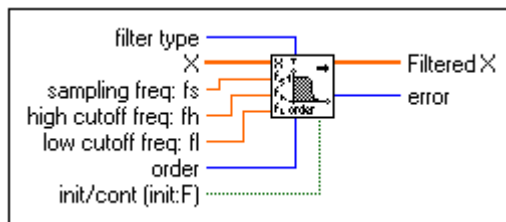
X: Señal de entrada que se va a filtrar, arreglo dbl.  
Sampling freq (frecuencia de muestreo) fs  
High cutoff freq: frecuencia de corte alta.  
Low cutoff freq: frecuencia de corte baja.  
Order: este es el orden del filtro debe ser mayor que 0.  
Init/cont: sirve para resetear el filtro o seguir con el ultimo estado del filtro.  
Filtered X: Señal de salida del filtro

#### Chebyshev Filter.vi:



Filter type: Se especifica el tipo de filtro por un número entero.

#### Butterworth filter.vi:

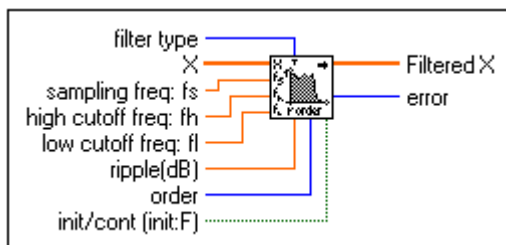


0: Pasa bajo  
1: Pasa alto  
2: Pasa banda  
3: Elimina banda

Filter type: Se especifica el tipo de filtro por un número entero.

X: Señal de entrada que se va a filtrar, arreglo dbl.  
Sampling freq (frecuencia de muestreo) fs  
High cutoff freq: frecuencia de corte alta.  
Low cutoff freq: frecuencia de corte baja.  
Order: este es el orden del filtro debe ser mayor que 0.  
Init/cont: sirve para resetear el filtro o seguir con el ultimo estado del filtro.  
Filtered X: Señal de salida del filtro

#### Chebyshev Filter.vi:



0: Pasa bajo  
1: Pasa alto  
2: Pasa banda  
3: Elimina banda

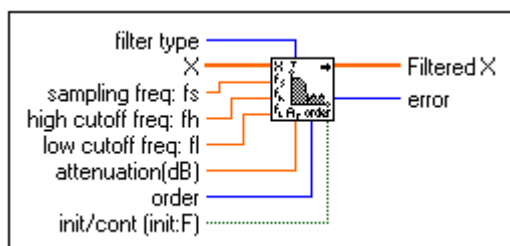
Filter type: Se especifica el tipo de filtro por un número entero.

X: Señal de entrada que se va a filtrar, arreglo dbl.  
Sampling freq (frecuencia de muestreo) fs  
High cutoff freq: frecuencia de corte alta.  
Low cutoff freq: frecuencia de corte baja.  
Order: este es el orden del filtro debe ser mayor que 0.  
Ripple: es la atenuación en dB que va a tener el filtro en la banda de paso.



Init/cont: sirve para resetear el filtro o seguir con el ultimo estado del filtro.  
Filtered X: Señal de salida del filtro

*Inverse chebyshev filter.vi:*



- 0: Pasa bajo
- 1: Pasa alto
- 2: Pasa banda
- 3: Elimina banda

Filter type: Se especifica el tipo de filtro por un número entero.

X: Señal de entrada que se va a filtrar, arreglo dbl.

Sampling freq (frecuencia de muestreo) fs

High cutoff freq: frecuencia de corte alta.

Low cutoff freq: frecuencia de corte baja.

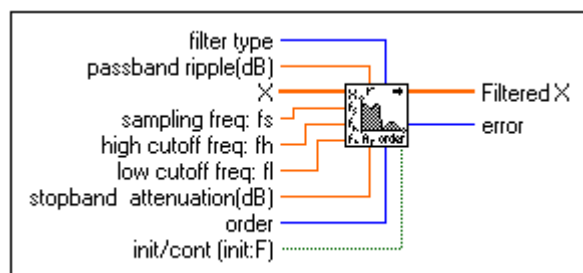
Attenuation (dB): atenuación que tendra la banda de supresión.

Order: este es el orden del filtro debe ser mayor que 0.

Init/cont: sirve para resetear el filtro o seguir con el ultimo estado del filtro.

Filtered X: Señal de salida del filtro.

*Elliptic filter.vi*



- 0: Pasa bajo
- 1: Pasa alto
- 2: Pasa banda
- 3: Elimina banda

Filter type: Se especifica el tipo de filtro por un número entero.

X: Señal de entrada que se va a filtrar, arreglo dbl.

Passaband ripple(db): atenuación de la banda pasante.

Sampling freq (frecuencia de muestreo) fs

High cutoff freq: frecuencia de corte alta.

Low cutoff freq: frecuencia de corte baja.

Stopband attenuation(dB): atenuación de la banda eliminada.

Order: este es el orden del filtro debe ser mayor que 0.

Init/cont: sirve para resetear el filtro o seguir con el ultimo estado del filtro.

Filtered X: Señal de salida del filtro.





### 3.4 Cálculo del orden del filtro IIR, selección óptima

Las especificaciones a cumplir para un filtro pasa bajos son:

Frecuencia de muestreo: 48 kHz.

Ripple en la banda de paso: 0.5 dB

Límite de la banda de paso: 8 kHz

Límite de la banda eliminada: 10 kHz

Ripple en la banda eliminada: por debajo de los 20 dB.

#### 3.4.1. Utilice las funciones `buttord`, `cheby1ord`, `cheby2ord` y `ellipord` de Matlab para determinar el orden necesario en cada uno de los filtros (Butterworth, Chebyshev, Chebyshev II y Elliptic) para cumplir con las especificaciones dadas.

*Butterworth*: el orden necesario para las especificaciones anteriores es 12.

*Chebyshev*: el orden necesario para las especificaciones anteriores es 6.

*Chebyshev II*: el orden necesario para las especificaciones anteriores es 6.

*Elliptic*: el orden necesario para las especificaciones anteriores es 4.

#### 3.4.2. ¿Cuál es el filtro de menor orden? ¿Por qué?

El filtro de menor orden es el Elliptic, esto se debe a que se permite un cierto ripple en la banda de paso y la banda de rechazo, por supuesto si se desea disminuir el ripple en la banda de paso el orden aumentaría

### 3.5 Diseño de un filtro digital IIR a partir de un prototipo analógico.

#### 3.5.1. Dada la siguiente función de transferencia en tiempo continuo, la cuál permanece a un filtro pasa bajo Butterworth de segundo orden:

Para una frecuencia de corte  $f_c = 10$  kHz y una frecuencia de muestreo de 48 kHz.

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

a) La función de transferencia en el plano Z mediante una transformación bilineal, utilice la función bilinear de Matlab para tal fin.

$$H(z) = \frac{0.1820 \cdot Z^2 + 0.3639 \cdot Z^1 + 0.1820}{1.0000 \cdot Z^2 - 0.4857 \cdot Z^1 + 0.21}$$

---

```
%filtro pasa bajos butterworth IIR
fc=10000;
fs=48000;
wc=2*pi*fc;
Ns=[wc^2];%coef del numerador en el plano S
Ds=[1 wc*sqrt(2) wc^2];%coef del denominador en el S
[Nz,Dz]=bilinear(Ns,Ds,fs)%aplico transf bilineal para pasarlo al plano Z-
discretizo
```

---

b) Gráfico de la ubicación de los polos y ceros en el plano Z, utilice la función `zplane` de Matlab. Este filtro es totalmente estable ya que los polos se encuentran dentro del radio unitario, del plano Z.

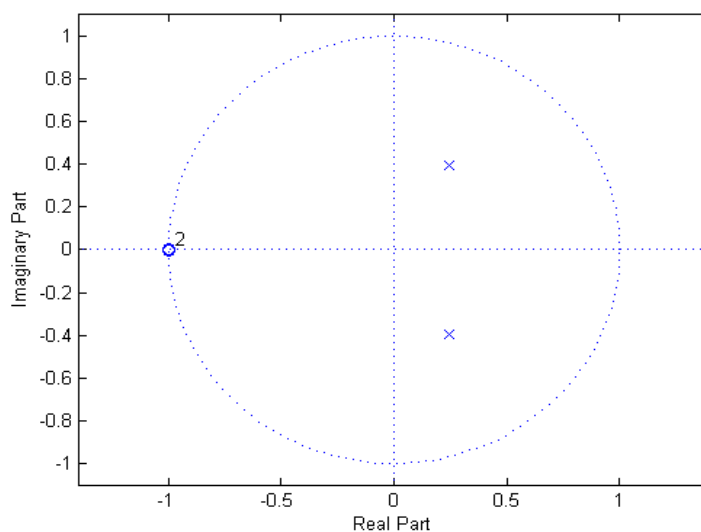


Ilustración 5: Ceros y polos en el plano Z

c) Respuesta en frecuencia del filtro, utilice la función freqz de Matlab.

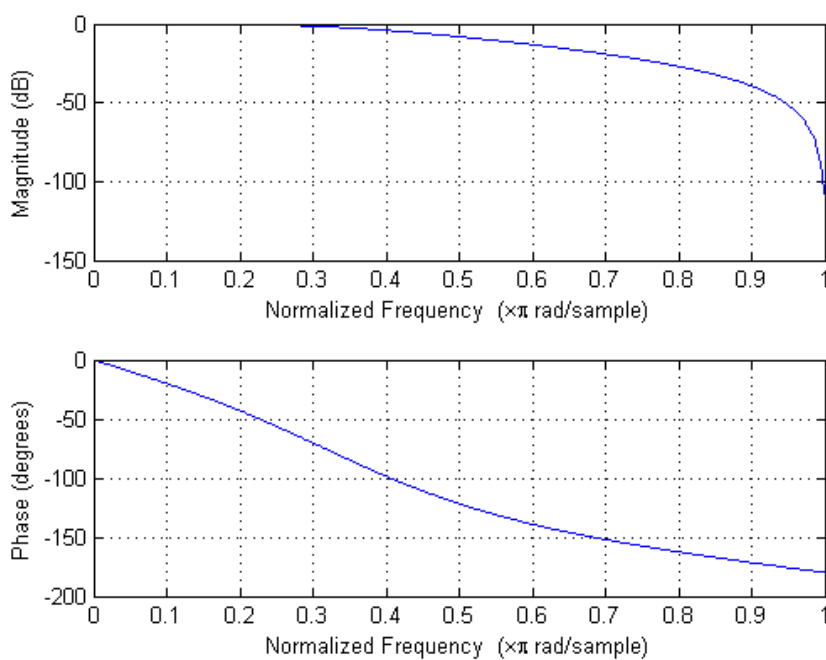
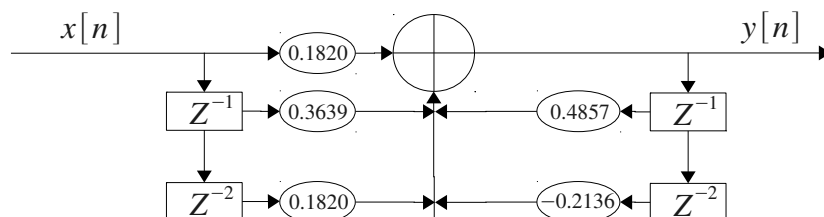


Ilustración 6: Respuesta en frecuencia

d) La ecuación en diferencias, grafique el esquema de dicha ecuación.

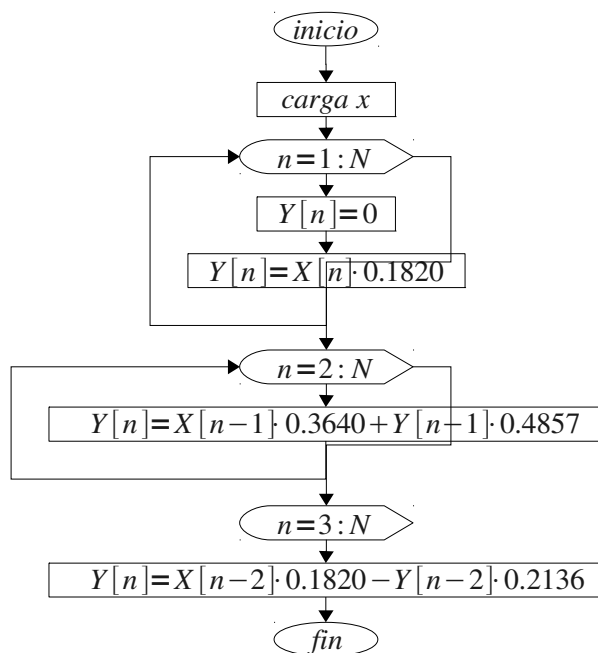
$$Y[n] = 0.1820 X[n] + 0.3639 X[n-1] + 0.1820 X[n-2] + 0.4857 Y[n-1] - 0.2136 Y[n-2]$$





e) El algoritmo que realice el filtrado, esto es, diagrama de flujo y código en lenguaje C o script de Matlab que permita filtrar una señal de entrada

```
A=10;
f=10000;
fs=48000;
N=200;
n=1:N;
x=A*sin((2*pi*(f/fs)*n));
r=rand(1,200);
X=r+x;
subplot(3,1,1)
plot(X);
axis([0 200 -12 12]);
title('Señal de entrada');
for n=(1:N);
    Y(n)=0;
    Y(n)=X(n)*0.1820;
end
for n=(2:N);
    Y(n)=(0.3640)*X(n-1)+(0.4857)*Y(n-1);
end
for n=(3:N);
    Y(n)=(0.1820)*X(n-2)-(0.2136)*Y(n-2);
end
subplot(3,1,2)
plot(Y);
axis([0 200 -5 5]);
title('Señal filtrada con programa');
Yf=filter([0.1820 0.3640 0.1820],[1.0000 -0.4857 0.2136],X);
subplot(3,1,3)
plot(Yf)
axis([0 200 -7 7]);
title('Señal filtrada con Filter');
```





f) Genere una señal de 10 kHz y mediante el algoritmo realizado en el punto anterior verifique el funcionamiento del filtro. Luego utilice la función filter de Matlab para volver a verificar el funcionamiento, compare los resultados obtenidos.

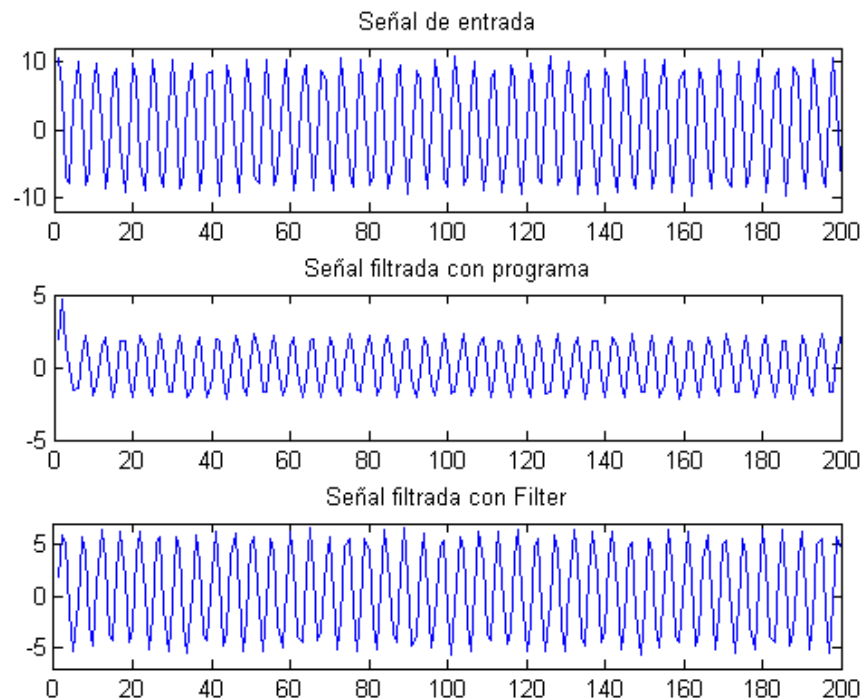


Ilustración 7: Respuesta en el MatLAB

### 3.5.2. PREWARPING (predistorsión)

a) Obtenga la función de transferencia  $H(z)$  teniendo en cuenta la frecuencia de prewarping.

```
NUM = [0 0 3947841760] % [0 0 (2*pi*10000)^2]
DEN = [1 88857.6587 3947841760] % [1 sqrt(2)* (2*pi*10000) (2*pi*10000)^2]
Fs=48000;
Fp=11723.89;
Ns= [0 0 3947841760];
Ds=[1 88857.6587 3947841760];
[N,D] = bilinear(Ns,Ds,Fs,Fp)
```

$$H(Z) = \frac{0.2383 + 0.4766Z^{-1} + 0.2383Z^{-2}}{1 - 0.2276Z^{-1} + 0.1807Z^{-2}}$$

b) Compare los resultados obtenidos con los del punto 4.5.1

Los resultados que encontramos al aplicar la frecuencia de prewarping al filtro tradicional Butterworth, se pudo ver en el diagrama de respuesta en frecuencia que en la frecuencia de corte existe una atenuación muy cercana a los 3dB que sería una señal filtrada con el valor en amplitud del 70.7% de la señal de entrada, en cambio en el filtro que no utilizamos esta corrección la atenuación en la frecuencia de corte es un poco mayor que 3dB (alrededor de 4.2dB) lo cual es un error no deseado en el diseño de filtros.



### 3.6 Diseño de un filtro Notch (IIR)

Diseñe un filtro notch mediante el método de ubicación de polos-zeros. Las especificaciones del filtro son:

Frecuencia del notch,  $f_c = 50$  Hz

Ancho del notch,  $BW = 5$  Hz

Frecuencia de muestreo,  $f_s = 1000$  Hz

Pasos a seguir en el diseño:

1°- La componente (frecuencia del notch) en una señal puede ser rechazada mediante la ubicación de un par de ceros sobre el círculo unidad del plano Z correspondiente a los 50 Hz, por lo tanto tendremos los siguientes ceros:

$$2\pi \frac{f_c}{f_s} = C_{12} \text{ rad/seg} = C_{12}^\circ \quad \text{en} \quad (z - e^{-jC_1}) \quad \text{y} \quad \text{en} \quad (z - e^{-jC_2})$$

calculo del arco de los ceros:

$$2\pi \frac{f_c}{f_s} = \pm 0.31415 [\text{rad/seg}] = \pm 18^\circ$$

2° - La posición de los polos en relación a los ceros determinan el ancho del notch (BW) y la amplitud de ambos lados del notch. Estos polos son ubicados sobre el mismo radio de los ceros en un valor  $r < 1$ , donde:

$$r = 1 - \frac{BW}{f_s} \cdot \pi = 1 - \frac{5}{1000} \cdot \pi = 0.98429$$

por lo tanto tendremos los siguientes polos:

$$(z - r e^{-jC_1}) \quad \text{y} \quad \text{en} \quad (z - r e^{-jC_2})$$

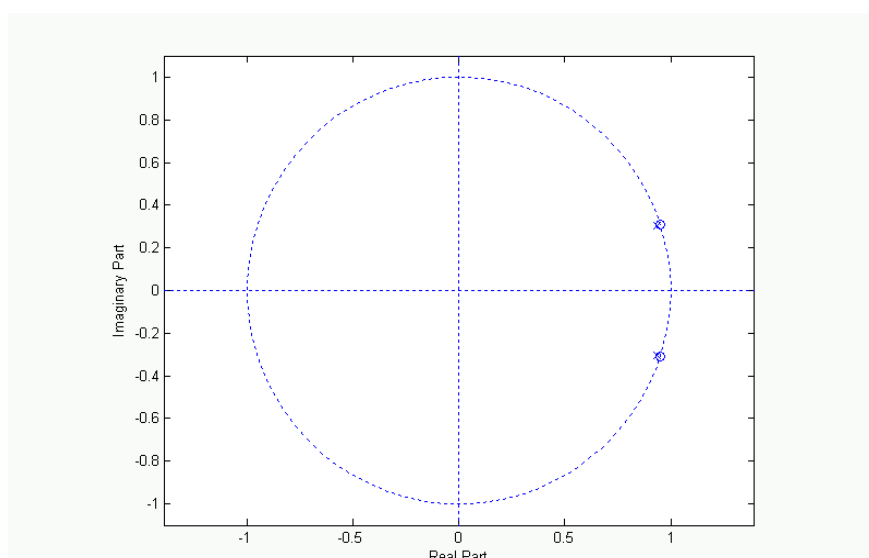


Ilustración 8: ubicación de las raíces



3° - Obtenga la función de transferencia  $H(z)$ .

$$H(z) = \frac{(z - e^{j0.31415}) \cdot (z - e^{-j0.31415})}{(z - 0.98429 e^{j0.31415}) \cdot (z - 0.98429 e^{-j0.31415})}$$

Según Euler:

$$e^{jC_1} = \cos(C_1) + j\sin(C_1)$$

$$e^{-jC_2} = \cos(C_2) - j\sin(C_2)$$

$$H(z) = \frac{z^2 - 1.9021z + 1}{z^2 - 1.8722z + 0.9688}$$

4° - Grafique la respuesta en frecuencia del filtro Notch.

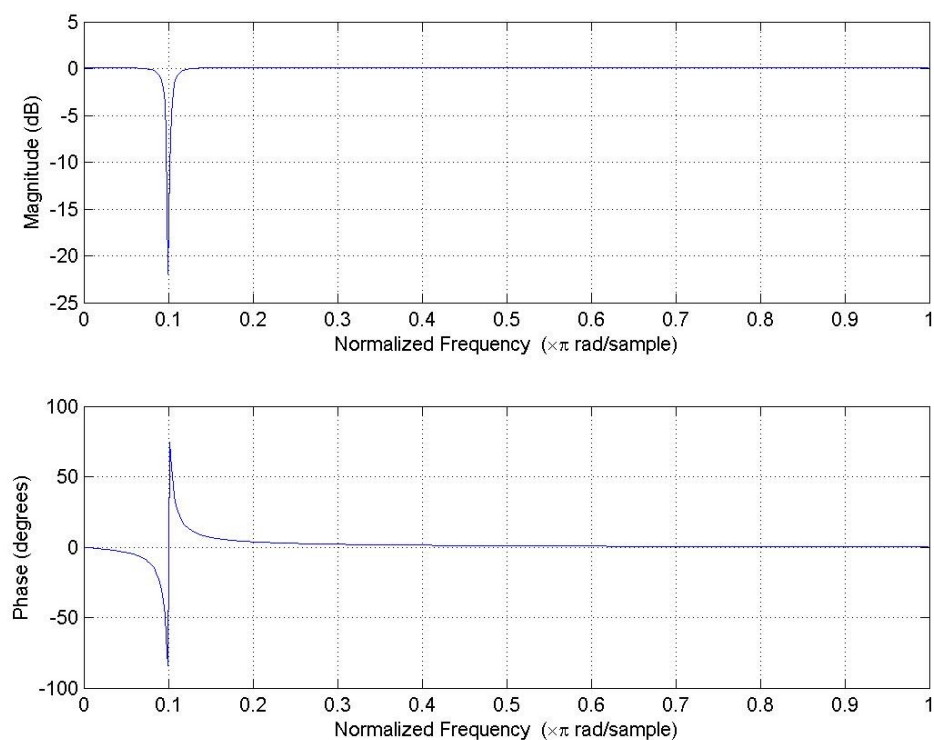


Ilustración 9: Respuesta en frecuencia del filtro Notch

5° - Obtenga la ecuación en diferencias del filtro Notch.

$$Y[n] = X[n] - 1.9021X[n-1] + X[n-2] + 1.8722Y[n-1] - 0.9688Y[n-2]$$



### 3.7 Filtro digital de respuesta finita al impulso, FIR (Finite Impulse Response)

#### 3.7.1. Analice la función *fir1* de Matlab para el diseño de filtros FIR mediante el uso de ventanas.

$B = \text{fir1}(N, W_n, \text{WIN})$  diseña un filtro FIR pasa bajos de orden  $N$  que responde al vector  $\text{WIN}$  de  $N+1$  posiciones el cual determina la respuesta al impulso de la señal de entrada a una determinada ventana.  $W_n$  es la frecuencia de interés normalizada.

La ventana por defecto es la de Hamming de longitud  $N+1$ .

Existe una excepción para las siguientes ventanas:

$B = \text{fir1}(N, W_n, \text{kaiser}(N+1, 4))$ , usa una ventana Kaiser con un beta de 4.

$B = \text{fir1}(N, W_n, \text{'high'}, \text{chebwin}(N+1, R))$ , usa una ventana Chebyshev con una atenuación de 4 decibels en la banda de paso.

Para obtener el vector de las demás ventanas se utiliza la función  $\text{WIN} = \text{WINDOW}(@\text{nombre}, N)$ .

Los nombres disponibles son:

@bartlett	- Bartlett window.
@barthannwin	- Modified Bartlett-Hanning window.
@blackman	- Blackman window.
@blackmanharris	- Minimum 4-term Blackman-Harris window.
@bohmanwin	- Bohman window.
@chebwin	- Chebyshev window.
@flattopwin	- Flat Top window.
@gausswin	- Gaussian window.
@hamming	- Hamming window.
@hann	- Hann window.
@kaiser	- Kaiser window.
@nuttallwin	- Nuttall defined minimum 4-term Blackman-Harris window.
@parzenwin	- Parzen (de la Valle-Poussin) window.
@rectwin	- Rectangular window.
@tukeywin	- Tukey window.
@triang	- Triangular window.

Para filtros con ganancia distinta de cero en  $F_s/2$  (por ej: un pasa altos o un pasa banda), el número de elementos del vector de la ventana debe ser par.

#### 3.7.2. Diseñe mediante la función *fir1* un filtro pasa bajos de longitud (TAPS) 32 con una frecuencia de corte de 100 Hz, para una frecuencia de muestreo de 1KHz, empleando las siguientes ventanas:

##### *Rectangular*

```
N=32;  
Wc=2*pi*100;  
Ws=2*pi*1000;  
Win=Window(@rectwin,N+1);  
b=fir1(N,Wc/Ws,Win);
```

##### *Triangular o Barlett*

```
N=32;  
n=1:N+5;  
Wc=2*pi*100;  
Ws=2*pi*1000;  
Win=Window(@triang,N+1);
```



```
b=fir1(N,Wc/Ws,Win);
```

### Hanning

```
N=32;  
n=1:N+5;  
Wc=2*pi*100;  
Ws=2*pi*1000;  
Win=Window(@hann,N+1);  
b=fir1(N,Wc/Ws,Win);  
Hamming  
N=32;  
n=1:N+5;  
Wc=2*pi*100;  
Ws=2*pi*1000;  
Win=Window(@hamming,N+1);  
b=fir1(N,Wc/Ws,Win);  
Blackman  
N=32;  
n=1:N+5;  
Wc=2*pi*100;  
Ws=2*pi*1000;  
Win=Window(@blackman,N+1);  
b=fir1(N,Wc/Ws,Win);
```

### 3.7.3. Grafique la respuesta impulsional del filtro FIR y su respuesta en frecuencia para cada ventana.

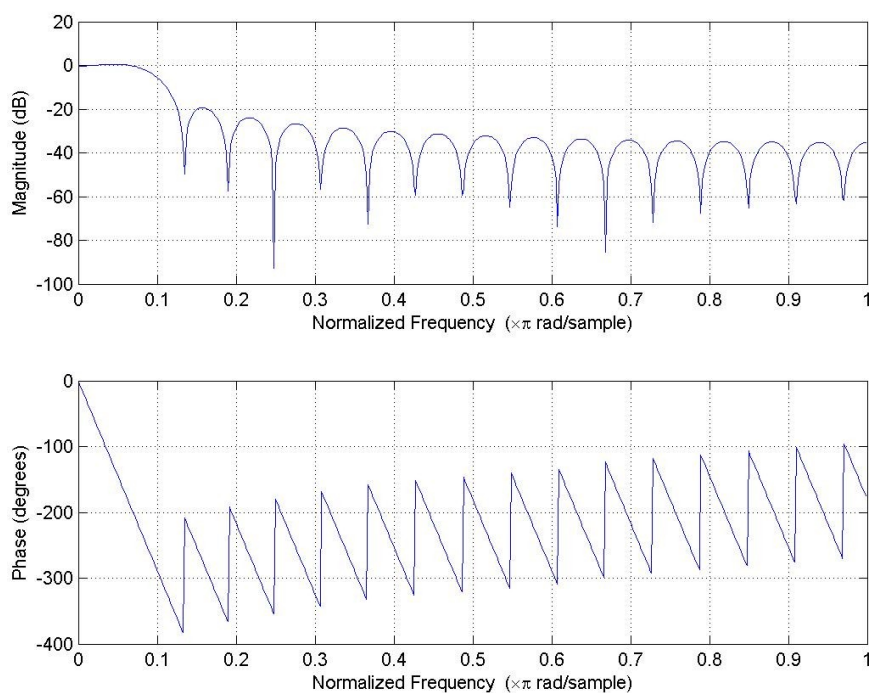


Ilustración 10: Rectangular



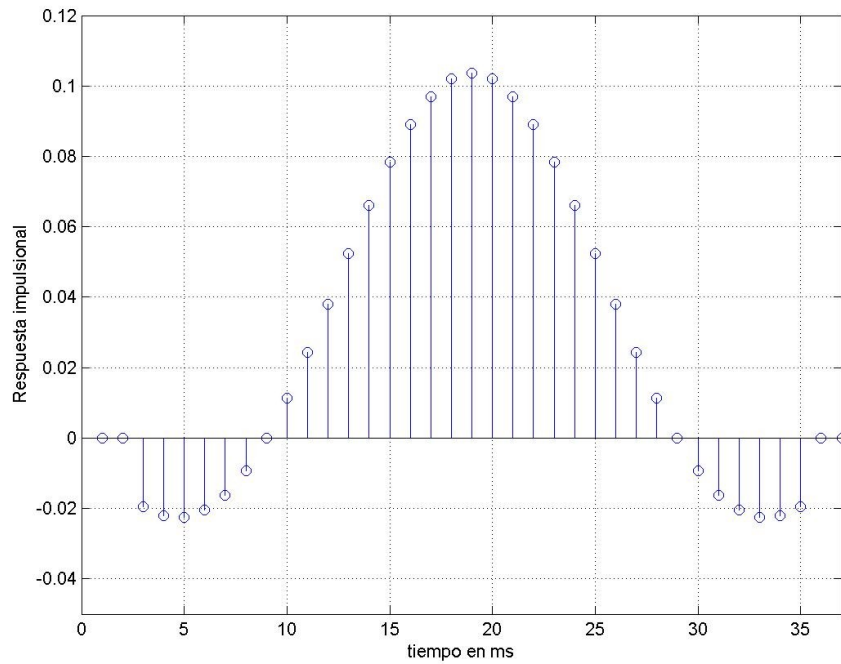


Ilustración 11: Rectangular - Respuesta al impulso

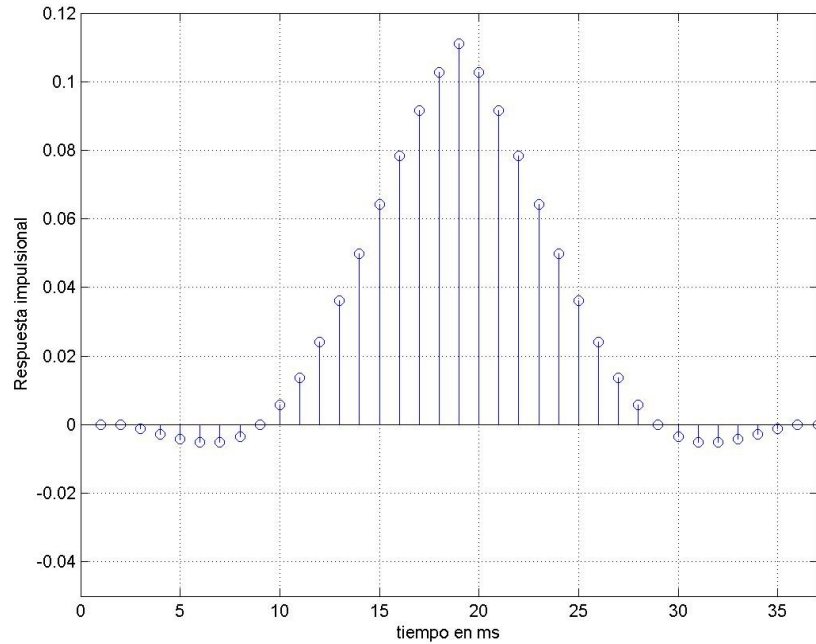


Ilustración 12: Triangular o Bartlett

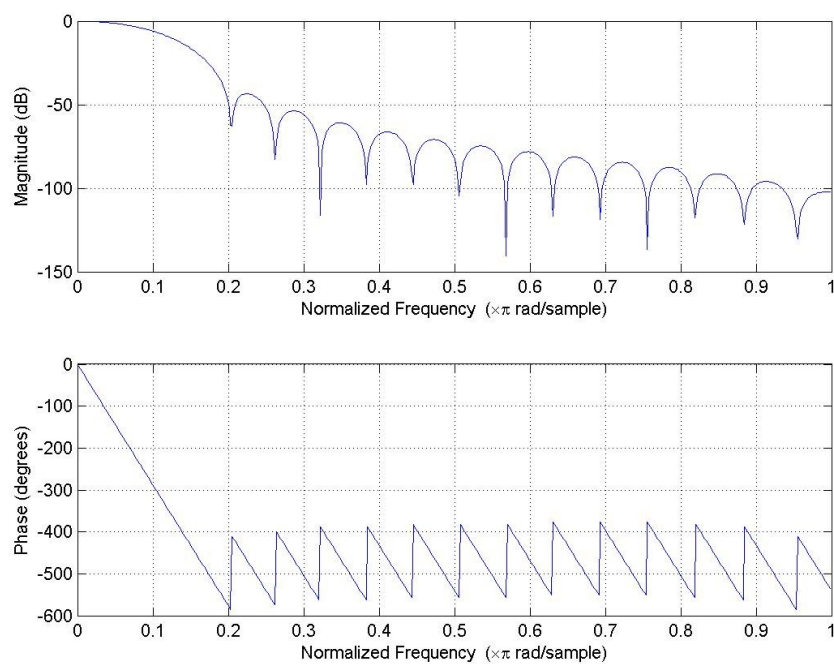


Ilustración 13: Hanning

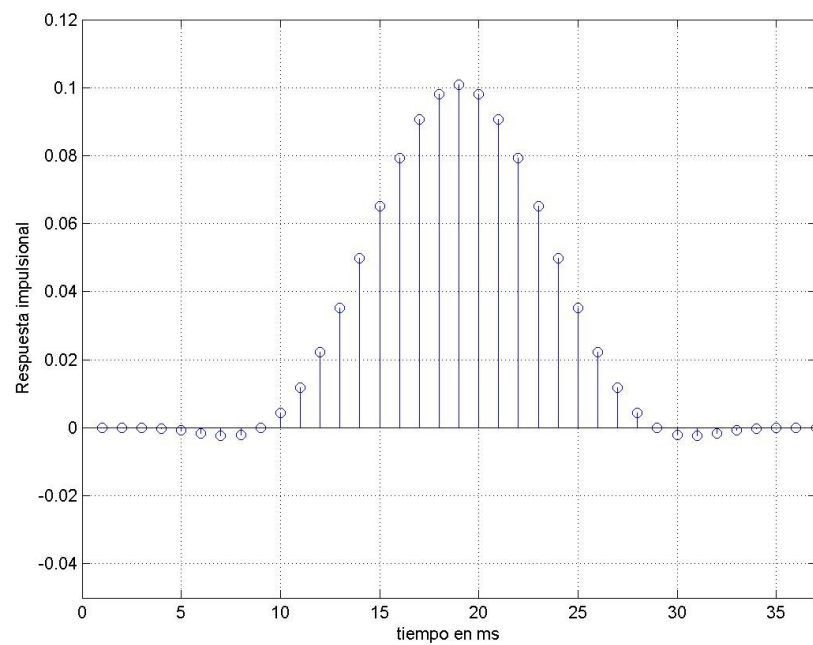


Ilustración 14: Hanning - Respuesta al impulso

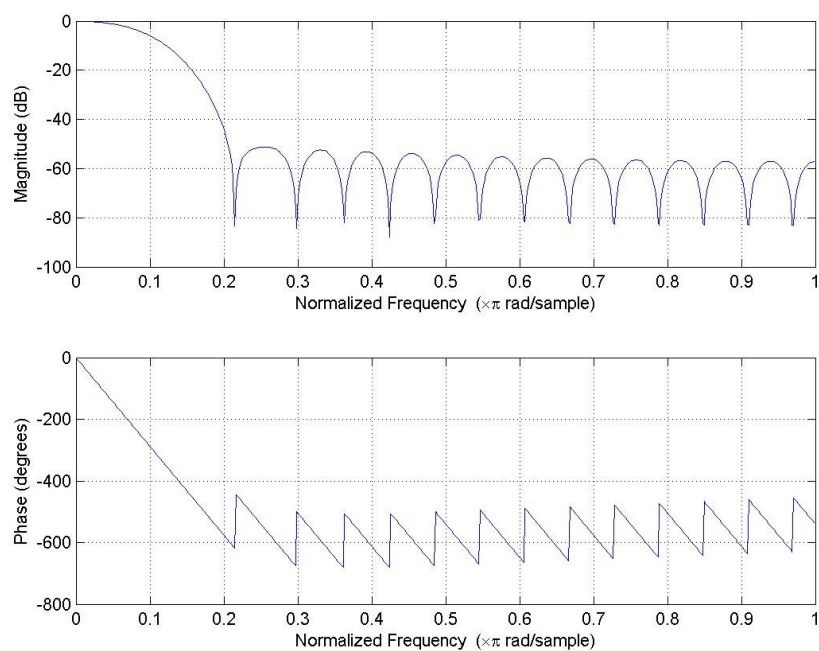


Ilustración 15: Hamming

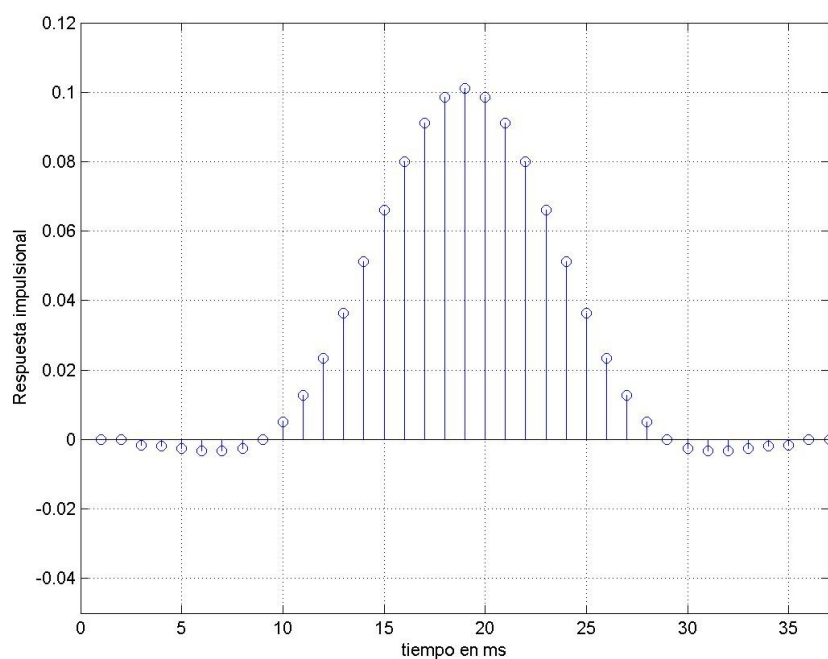


Ilustración 16: Hamming - Respuesta al impulso

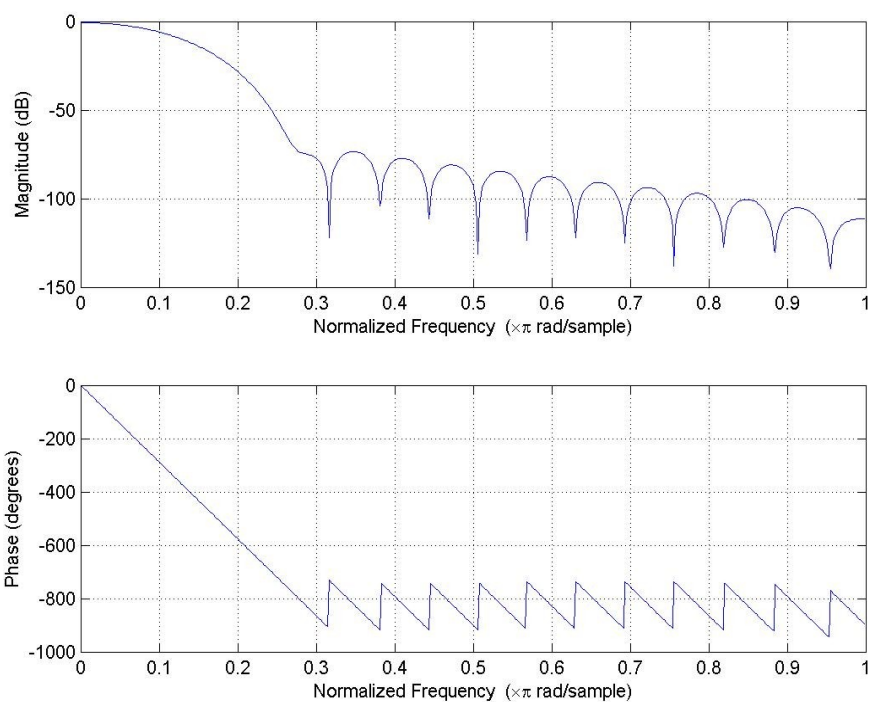


Ilustración 17: Blackman

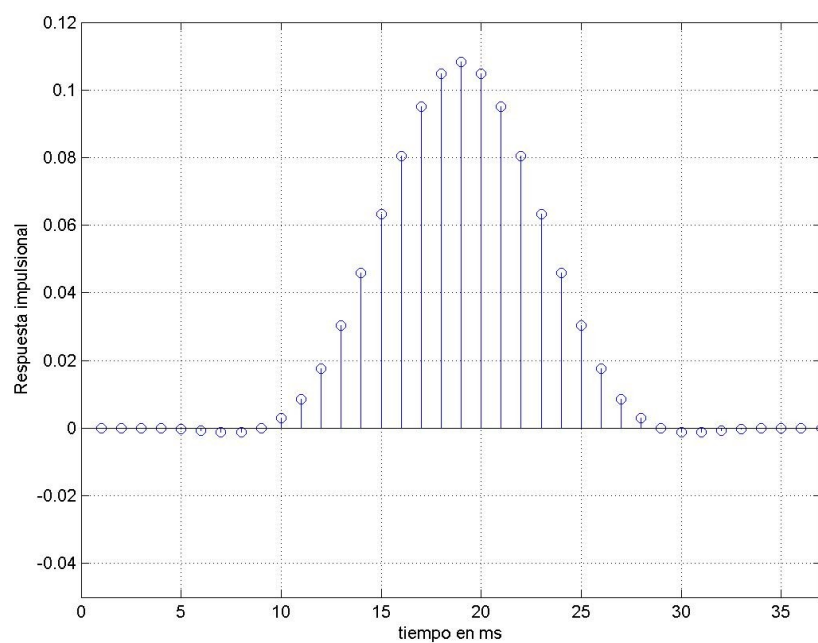
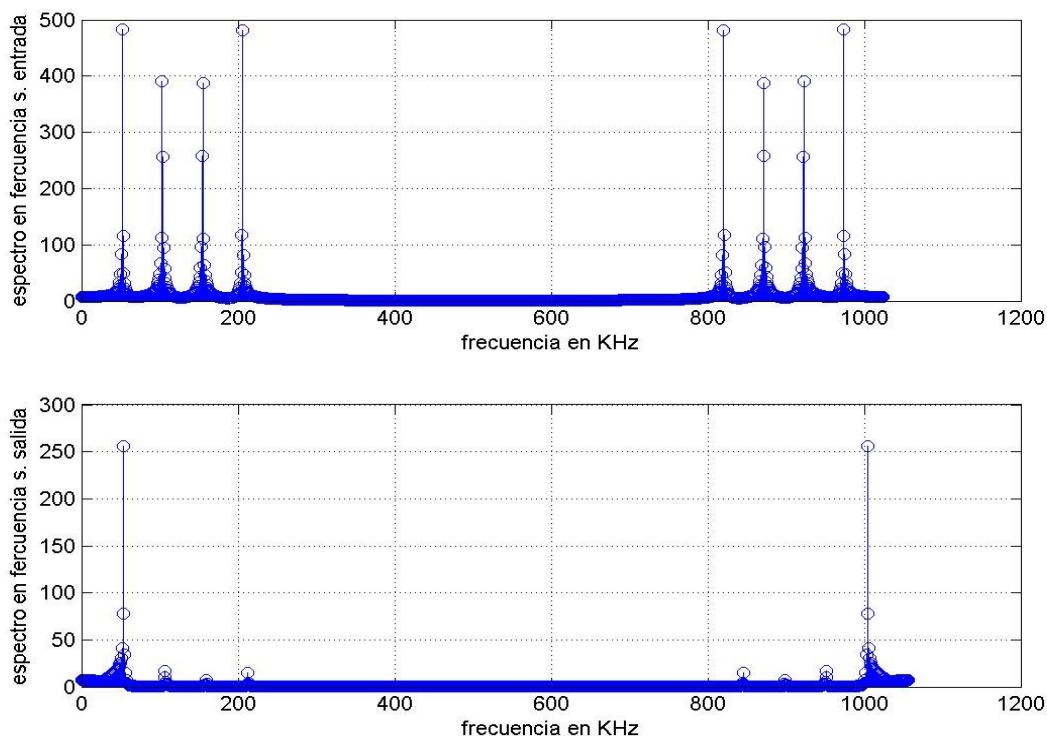


Ilustración 18: Blackman - Respuesta al impulso

**3.7.4. Genere 1024 muestras de una señal formada por la suma de sinusoides de 50 Hz, 100Hz, 150 Hz y 200Hz. Obtenga el espectro de frecuencias de dicha. Pase la señal generada por el filtro para bajos diseñado con una de las ventanas, por ejemplo Hanning, y grafique el espectro de frecuencias de la señal obtenida a la salida del filtro.**



*Ilustración 19: Ventana Rectangular*

### 3.8 Filtro Promediador Móvil

Este es el filtro más común en el procesamiento digital de señales, generalmente utilizado para la reducción de ruido. Es el principal filtro para señales en el dominio del tiempo, mientras que es el peor filtro desde el punto de vista del dominio de la frecuencia ya que no es muy útil a la hora de tener que separar bandas de frecuencias muy cercanas unas de otras.

#### *Implementación convolutiva del filtro promediador móvil*

El filtro promediador móvil opera mediante la promediación de un número  $M$  de muestras de la señal de entrada para producir una muestra de la señal de salida.

$$y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i+j]$$

La ecuación del filtro promediador móvil es:

donde  $x[]$  es la señal de entrada,  $y[]$  es la señal de salida y  $M$  es el número de muestras. Una alternativa es tomar  $M/2$  muestras en torno a la muestra de salida a calcular, para lo cual la sumatoria será entre  $-(M-1)/2$  y  $(M-1)/2$ .



Esta ecuación puede implementarse mediante la convolución haciendo  $h[] = \{..., 1/M, 1/M, 1/M, 0, ...\}$ , por ejemplo si  $M=5$  el kernel (respuesta al impulso) de filtro será:

$$h[] = [\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}]$$

Implementación recursiva del filtro promediador móvil

La ecuación recursiva del filtro es:

$$y[i] = y[i-1] + x \frac{[i+p]}{M} + x \frac{[i-q]}{M}$$

$$\text{donde } p = \frac{M-1}{2}; q = p+1$$

para demostrar esto supongamos que  $M=5$  y estamos calculando la salida  $y[10]$ ,

$$y[10] = \frac{(x[8] + x[9] + x[10] + x[11] + x[12])}{5}$$

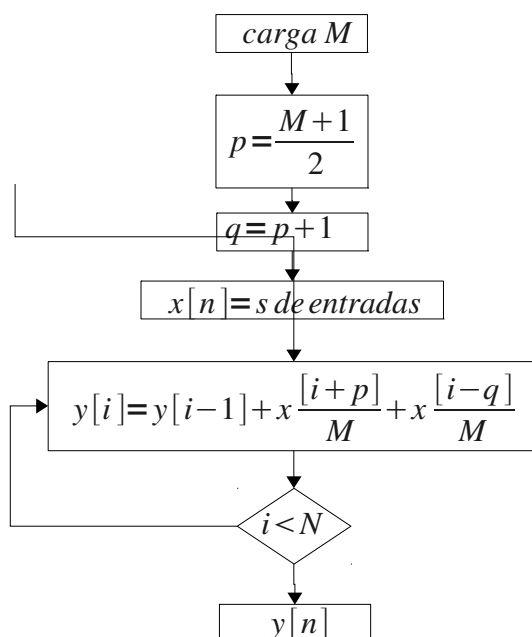
luego:

$$y[11] = \frac{(x[9] + x[10] + x[11] + x[12] + x[13])}{5}$$

como podemos ver el cálculo de  $y[11]$  lo podemos hacer mediante

$$y[11] = \frac{y[10] + x[13] + x[8]}{5}$$

### 3.8.1. Realice el diagrama de flujo del algoritmo del filtro promediador móvil.

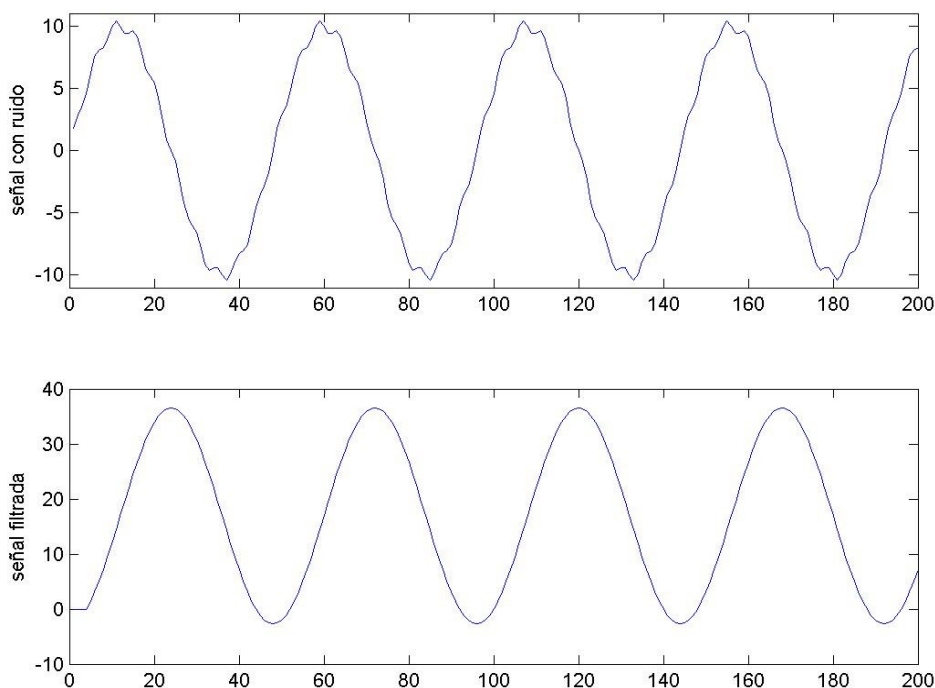




**3.8.2. Desarrolle el código, en lenguaje SCRIPT de Matlab o en lenguaje C, para el filtro promediador móvil.**

**3.8.3. Verifique el funcionamiento de dicho filtro en Matlab, generando una señal senoidal con ruido y pasándola por el filtro.**

```
fc=1000;  
fs=48000;  
fruido=10000;  
N=1000;  
n=1:N;  
M=7;  
p=(M-1)/2;  
q= p+1;  
seno=10*sin(2*pi*(fc/fs)*n);  
ruido=0.5*sin(2*pi*(fruido/fs)*n);  
x=seno+ruido;  
subplot(2,1,1),plot(x);  
axis([0 1200 -11 11]);  
y=zeros(1,N);  
for k=q+1:N+p  
    y(k)=y(k-1)+x(k+p)/M+x(k-q)/M;  
end  
subplot(2,1,2),plot(y);
```



*Ilustración 20: Entrada y salida en el filtro digital*



### 3.8.4. Analice la respuesta en frecuencia del filtro promediador móvil para $M=5$ , $M=15$ y $M=31$

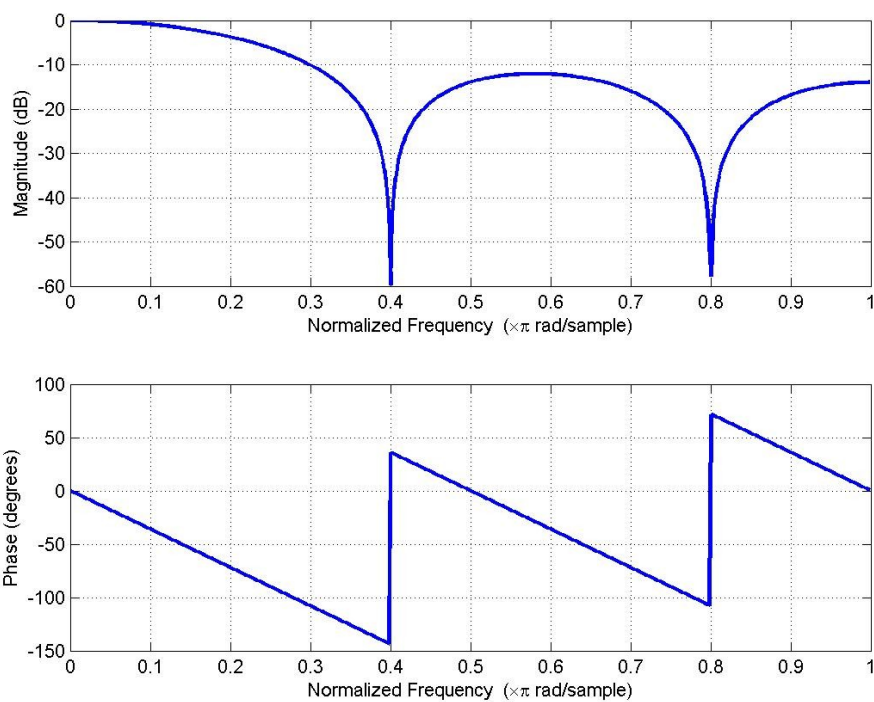


Ilustración 21: Respuesta en frecuencia para  $M=5$

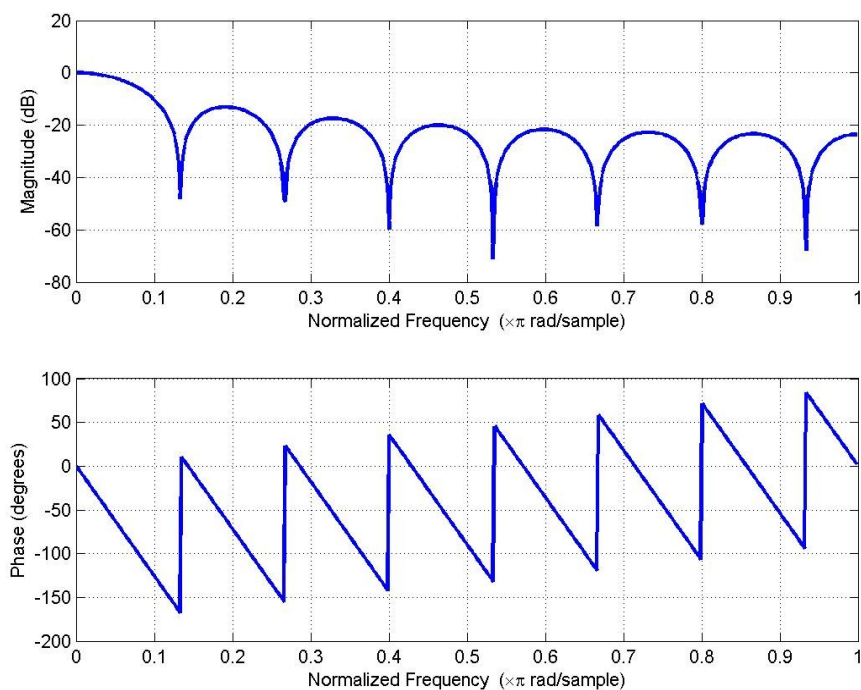


Ilustración 22: Respuesta en frecuencia para  $M=1$



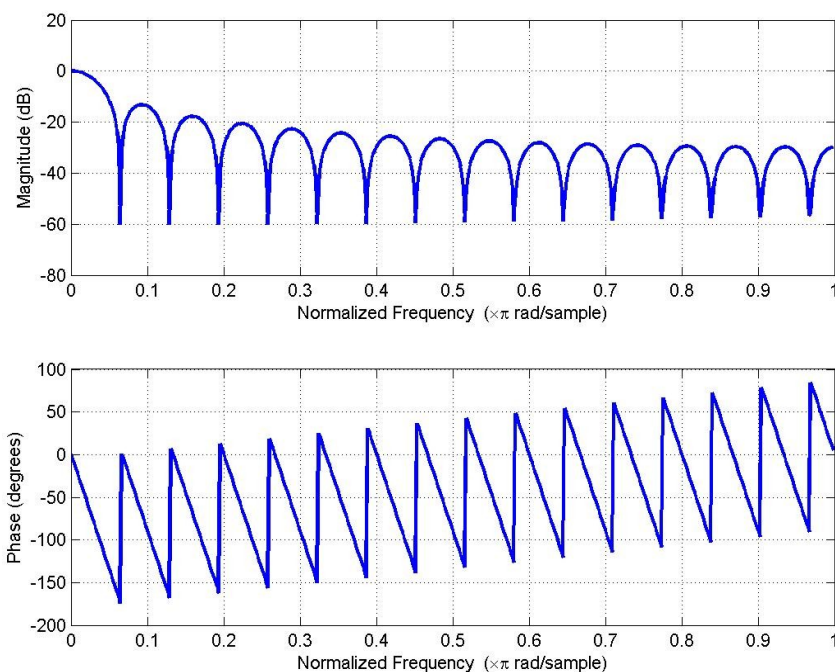


Ilustración 23: Respuesta en frecuencia para  $M=31$

**3.9 Presentación de un alguno de los siguientes temas sobre DSPIC (tema a realizarse en el primer cuatrimestre) Se deberá dar una explicación del uso, un ejemplo de aplicación en MPLABC30 y se deberá entregar un informe en formato Word o PDF con el formato de la cátedra .**

TEMAS	Fechas Propuestas
ALU	15/05/09
CLOCK y TIMERS	22/05/09
ACD 10 y 12bits	05/06/09
MEM RAM	12/06/09
<b>MEM FLASH</b>	<b>12/06/09</b>
DCI	19/06/09
PWM	19/06/09
USART	26/06/09
SPI	26/06/09



### 3.10 Implementación de filtros digitales en un DSPIC

- 3.10.1. Se requiere armar el siguiente hardware para este practico. Implementar un filtro anti-aliasing Butterworth de 4 orden y una frec de corte de 4Khz. Usando el FilterLab de Microchip .**

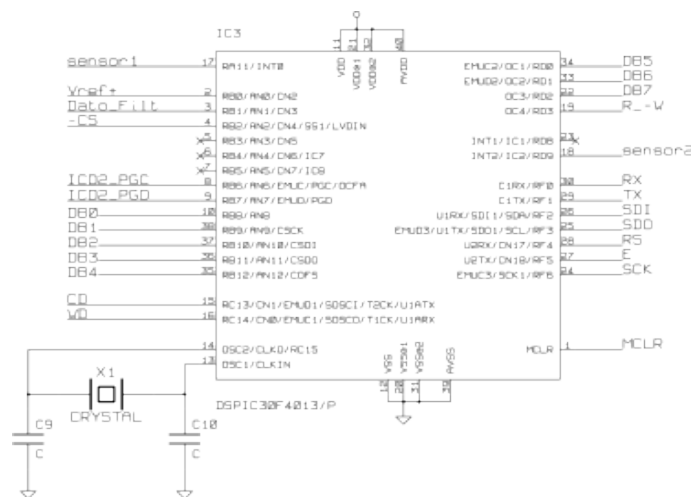


Ilustración 24: Implementación en el dsPIC

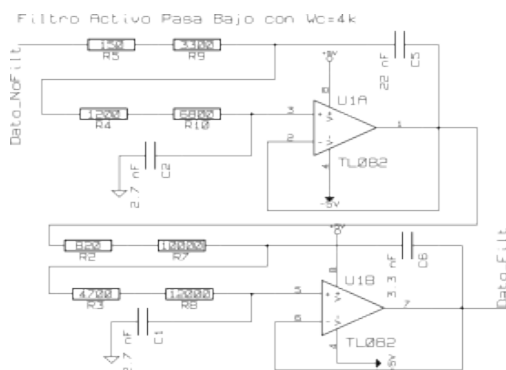


Ilustración 25: Filtro implementado

- 3.10.2. Diseñe un filtro FIR pasa bajo, de longitud 31 y frecuencia de corte  $f_c = 2\text{kHz}$ , para una frecuencia de muestreo de  $8\text{kHz}$ . Utilice la Herramienta FdaTool de MATLAB para el diseño del filtro con una ventana de Hamming.**

Visado en laboratorio.

- 3.10.3. Agregue la función del filtro FIR que posee la librería de Microchip para que procese las muestras provenientes de uno de los canales del A/D antes de ser enviadas por RS232 a la PC.**

Visado en laboratorio.

- 3.10.4. Compile y compruebe el funcionamiento del programa sobre el hardware, haciendo uso de un generador de funciones para verificar a distintas frecuencia.**

Visado en laboratorio.



**3.10.5. Realice mediciones de amplitud y fase para llevarlas a un diagrama de amplitud y fase en función de la frecuencia con los datos recibidos en la PC.**

**Visado en laboratorio.**

**3.10.6. Compare los resultados obtenidos con el diagrama de respuesta en frecuencia obtenido por la función *freqz* de Matlab durante el diseño del filtro. Saque sus propias conclusiones del diagrama de respuesta en frecuencia.**

**Visado en laboratorio.**

**3.11 Utilizando el Circuito anterior Implementar la FFT para obtener la Frecuencia y nivel de Potencia de la Señal medida .**

**Visado en laboratorio.**