

Filtros software en el proyecto SHFD

En este documento se muestra como se ha realizado el cálculo de coeficientes de la librería de filtros que se utilizan en el proyecto SHFD. Hay tres familias de filtros: filtros de primer orden, filtros de segundo orden, y filtros de orden superior.

Este documento no tiene más ámbito que el de documentar la obtención de los coeficientes digitales de filtros de primer y segundo orden, a partir de la especificación de los parámetros que los definen. En los filtros de primer orden los parámetros serán : la frecuencia de muestreo , y la frecuencia de corte. En los filtros de segundo orden los parámetros serán: frecuencia de muestreo, frecuencia de corte/paso y factor de calidad.

En las siguientes secciones veremos como obtener los coeficientes en filtros de primer orden, y en filtros de segundo orden.

1. Filtros de primer orden:

1.1. Filtro paso bajo

1.1.1. Transformada de Laplace del filtro

$$Hs(s) = \frac{1}{\frac{s}{\omega_o} + 1} = \frac{\omega_o}{s + \omega_o} \quad (1)$$

siendo $\omega_o = 2\pi fc$ y fc la frecuencia de corte del filtro.

1.1.2. Mapeo de polos y ceros en el plazo z

Como se puede extraer de (1) tenemos un polo en $s = -\omega_o$, y por lo tanto el filtro digital tendrá un polo en $z = e^{-\omega_o T}$

$$H(z) = K \frac{1}{z - e^{-\omega_o T}} \quad (2)$$

1.1.3. Valor de la ganancia

Buscamos determinar el valor de la ganancia K para obtener el mismo resultado en el filtro digital y analógico en el rango de frecuencias de los filtros. Comparamos la ganancia de ambos filtros a la misma frecuencia

1.1.4. Cálculo de ganancia a frecuencias bajas

$$\lim_{\omega \rightarrow 0} Hs(j\omega) = 1 \quad (3)$$

$$\lim_{\omega \rightarrow 0} H(e^{j\omega_o T}) = \frac{K}{1 - e^{-\omega_o T}} \quad (4)$$

igualando 3 y 4 obtenemos $K = 1 - e^{-\omega_o T}$

1.1.5. Ecuación final en diferencias

Sustituyendo K en la en la función de transferencia en z obtenemos

$$H(z) = \frac{1 - e^{-\omega_o T}}{z - e^{-\omega_o T}} = \frac{(1 - e^{-\omega_o T})z^{-1}}{1 - e^{-\omega_o T}z^{-1}} \quad (5)$$

La ecuación en diferencias es:

$$y_n = e^{-\omega_o T} y_{n-1} + (1 - e^{-\omega_o T}) x_{n-1} \quad (6)$$

1.1.6. Eliminación de retardo en el filtro

Como se aprecia en (6) no hacemos uso de x_n sino de x_{n-1} , para eliminar este retardo basta con añadir un zero en el origen, veamos como queda la función de transferencia final:

$$H(z) = \frac{(1 - e^{-\omega_o T})z}{z - e^{-\omega_o T}} = \frac{1 - e^{-\omega_o T}}{1 - e^{-\omega_o T}z^{-1}} \quad (7)$$

La ecuación en diferencias final es:

$$y_n = e^{-\omega_o T} y_{n-1} + (1 - e^{-\omega_o T}) x_n \quad (8)$$

1.1.7. Coeficientes del filtro

$$y_n = e^{-\omega_o T} y_{n-1} + (1 - e^{-\omega_o T}) x_n \Rightarrow a_0 y_n = a_1 y_{n-1} + b_0 x_n \quad (9)$$

siguiendo la estructura estándar de filtros digitales:

$$y[n] = (b_0 x[n] + b_1 x[n-1] + \dots) - (a_1 y[n-1] + a_2 y[n-2] + \dots) \quad (10)$$

los coeficientes del numerador son:

$$b_0 = 1; \quad (11)$$

$$b_1 = e^{-\omega_o T}; \quad (12)$$

los coeficientes del denominador son:

$$a_0 = 1; \quad (13)$$

$$a_1 = e^{-\omega_o T}; \quad (14)$$

1.1.8. Código de Matlab

```
close all;
clear all;
s=tf('s');
z=tf('z');
f0=0.1e3;
w0=2*pi*f0;
Hs=1/(1+(s/w0));
Fs=10e3;
Ts=1/Fs;
Hz=c2d(Hs,Ts)
a=exp(-w0*Ts)
HzMin=Hz*z %adding un zero en el origen busando el mnimo retardo del
           sistema.
figure;
opts = bodeoptions;
opts.FreqUnits = 'Hz';
bode(Hz,HzMin,opts);
figure;
step(Hz);
hold on;
step(HzMin);
a1=-exp(-2*pi*f0*Ts)
b0=1-exp(-2*pi*f0*Ts)
% input=ones(1,100);
% output=filter(cell2mat(HzMin.Numerator),cell2mat(HzMin.Denominator),
               input);
% output(1:20)
```

1.1.9. Respuesta en frecuencia del filtro

En la siguiente gráfica 1 se muestra la respuesta del filtro de paso bajo digital, y la misma respuesta pero con fase mínima (eliminando el retardo del filtro). Como se puede observar la utilización del zero para eliminar el retardo, implica no tener la misma respuesta en fase.

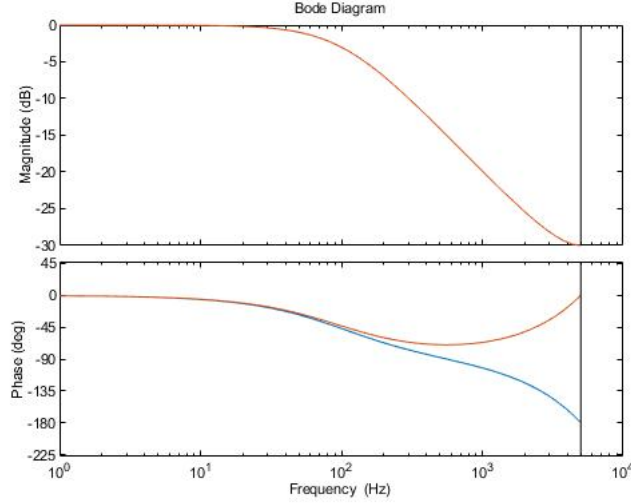


Figura 1: Respuesta en frecuencia del filtro paso bajo con retardo mínimo

1.1.10. Implementación

Al método software que se encarga de ejecutar este filtrado , se le deberán pasar los siguientes parámetros:

1. Frecuencia de muestro
2. Frecuencia de corte

Con estos parámetros el método software se encargará de extraer los coeficientes como se ha visto en los puntos anteriores y, realizar el filtrado de la muestra de entrada.

1.2. Filtro paso alto

1.2.1. Transformada de Laplace del filtro

$$Hs(s) = \frac{\frac{s}{\omega_o}}{\frac{s}{\omega_o} + 1} = \frac{s}{s + \omega_o} \quad (15)$$

siendo $\omega_o = 2\pi fc$;fc es la frecuencia de corte del filtro

1.2.2. Mapeo de polos y ceros en el plazo z

Como se puede extraer de (12) tenemos:

1. Un polo en $s = -\omega_o$, y por lo tanto el filtro digital tendrá un polo en $z = e^{-\omega_o T}$
2. Un cero en el origen $s = 0$, y por lo tanto el filtro digital tendrá un cero en $z = 1$ ” Por lo tanto la función de transferencia en la transformada Z tendrá este aspecto:

$$H(z) = K \frac{z - 1}{z - e^{-\omega_o T}} \quad (16)$$

1.2.3. Valor de la ganancia

Buscamos determinar el valor de la ganancia del filtro K, para obtener el mismo resultado en el filtro digital y analógico en el mismo rango de frecuencia. Por lo tanto, comparamos la ganancia de ambos filtros a la misma frecuencia ω_0 .

1.2.4. Cálculo de ganancia a frecuencias bajas

$$\lim_{\omega \rightarrow \omega_o} Hs(j\omega) = 10^{\frac{-3}{20}} (3db) \quad (17)$$

$$\lim_{\omega \rightarrow \omega_o} H(e^{j\omega_o T}) = K \frac{e^{j\omega_o T} - 1}{e^{j\omega_o T} - e^{-\omega_o T}} \quad (18)$$

igualando (14) y (15) obtenemos el valor de la ganancia K.

1.2.5. Ecuación final en diferencias

$$H(z) = K \frac{z - 1}{z - e^{-\omega_o T}} = K \frac{1 - z^{-1}}{1 - e^{-\omega_o T} z^{-1}} \quad (19)$$

La ecuación en diferencias es:

$$y_n = e^{-\omega_o T} y_{n-1} + K x_n - K e^{-\omega_o T} x_{n-1} \quad (20)$$

1.2.6. Coeficientes del filtro

$$y_n = e^{-\omega_o T} y_{n-1} + (x_n - e^{-\omega_o T} x_{n-1}) \Rightarrow a_0 y_n = a_1 y_{n-1} + b_0 x_n + b_1 x_{n-1} \quad (21)$$

siguiendo la estructura estándar de filtros digitales:

$$y[n] = (b_0 x[n] + b_1 x[n-1] + \dots) - (a_1 y[n-1] + a_2 y[n-2] + \dots) \quad (22)$$

los coeficientes numerador son :

$$b_0 = 1; \quad (23)$$

$$b_1 = -1 \quad (24)$$

los coeficientes denominador son :

$$a_0 = 1; \quad (25)$$

$$a_1 = e^{-\omega_o T}; \quad (26)$$

1.2.7. Código de Matlab

```
close all;
clear all;
s=tf('s');
z=tf('z');
opts = bodeoptions;
opts.FreqUnits = 'Hz';
f0=0.1;
w0=2*pi*f0;
Hs=(s/w0)/(1+(s/w0));
bode(Hs,opts);
Fs=10;
Ts=1/Fs;
num_in_w0=exp(i*w0*Ts)-1;
den_in_w0=exp(i*w0*Ts)-exp(-w0*Ts);
gain_in_w0=10^(-3/20); % -3db
K1=gain_in_w0/abs(num_in_w0/den_in_w0);
Hz=K1*c2d(Hs,Ts);
opts = bodeoptions;
opts.FreqUnits = 'Hz';
bode(Hs,Hs,opts);
% Depurar a nivel de muestra
input=ones(1,100);
output=filter(cell2mat(Hz.Numerator),cell2mat(Hz.Denominator),input);
output(1:20)
```

1.2.8. Respuesta en frecuencia del filtro

En la siguiente gráfica (2) se muestra la respuesta del filtro de paso alto digital.

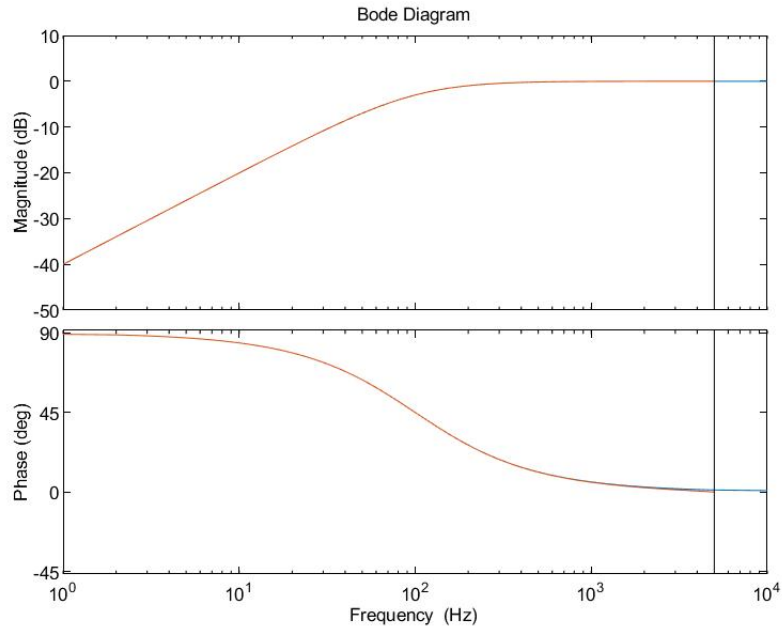


Figura 2: Respuesta en frecuencia del filtro paso alto

1.2.9. Implementación

Al método software que se encarga de ejecutar este filtrado, se le deberán pasar los siguientes parámetros:

1. Frecuencia de muestro
2. Frecuencia de corte

Con estos parámetros el método software se encargará de extraer los coeficientes como se ha visto en los puntos anteriores y, realizar el filtrado de la muestra de entrada.

2. Filtros de segundo orden:

En las siguientes secciones se muestran los filtros de segundo orden implementados:

2.1. Filtro paso bajo

2.1.1. Transformada de Laplace del filtro

$$Hs(s) = \frac{1}{\left(\frac{s}{\omega_0}\right)^2 + 2\zeta \frac{s}{\omega_0} + 1} \quad (27)$$

sabiendo que el factor de calidad del filtro paso bajo es $Q = \frac{1}{2\zeta}$ y, $\omega_0 = 2\pi fc$ siendo fc la frecuencia de corte.

2.1.2. Mapeo de polos y ceros en el plano z

La función de transferencia anterior se puede expresar como el producto de dos polos complejos conjugados y una ganancia.

$$Hs(s) = \frac{K}{(s+p)(s+\hat{p})} \quad (28)$$

donde $p = -\sigma + j\omega$ y $\hat{p} = -\sigma - j\omega$. Mapeando (13) en el plano z obtenemos

$$H(z) = \frac{K}{(z - e^{(-\sigma+j\omega)T})(z - e^{(-\sigma-j\omega)T})} \quad (29)$$

desarrollando la multiplicación del dividendo de (14) obtenemos

$$z^2 + z[e^{(-\sigma+j\omega)T} + e^{(-\sigma-j\omega)T}] + e^{-(2\sigma)T} \quad (30)$$

De la ecuación (12) podemos obtener los valores de σ y ω

$$\sigma = -(\omega_0/Q)/2 \quad (31)$$

$$\omega = \frac{\sqrt{\left(\frac{\omega_0}{Q}\right)^2 - 4\omega_o^2}}{2} \quad (32)$$

2.1.3. Valor de la ganancia

Se busca determinar el valor de la ganancia K para obtener el mismo resultado en el filtro digital y en el filtro analógico. Para ello se iguala la ganancia de ambos filtros a la misma frecuencia.

2.1.4. Cálculo de ganancia a frecuencias bajas

Para obtener la ganancia K, se realiza el límite de las funciones H(s) y la anterior H(z) en la banda de paso, para un filtro paso bajo se ha decidido que la frecuencia de comparación de ω_0 sea 0 rad/s.

$$\lim_{\omega \rightarrow 0} Hs(j\omega) = 1 \quad (33)$$

$$\lim_{\omega \rightarrow 0} H(e^{j\omega}) = \frac{K}{a_2 + a_1 + a_0} \quad (34)$$

igualando 27 y 28 obtenemos $K = a_2 + a_1 + a_0$

2.1.5. Ecuación final en diferencias

$$H(z) = \frac{K}{z^2 - z(e^{(-\sigma+j\omega)T} + e^{(-\sigma-j\omega)T}) + e^{-2\sigma T}} \quad (35)$$

La ecuación en diferencias es:

$$a_0 y_n = K x_{n-2} - [a_1 y_{n-1} + a_2 y_{n-2}] \quad (36)$$

2.1.6. Coeficientes del filtro

Siguiendo la estructura estándar de filtros digitales:

$$a_0 y[n] = (b_0 x[n] + b_1 x[n-1] + \dots) - (a_1 y[n-1] + a_2 y[n-2] + \dots) \quad (37)$$

los coeficientes finales que obtenemos son:

$$a_0 = 1 \quad (38)$$

$$a_1 = -(e^{(-\sigma+j\omega)T} + e^{(-\sigma-j\omega)T}) \quad (39)$$

$$a_2 = e^{-2\sigma T} \quad (40)$$

2.1.7. Código de Matlab

```
close all;
s=tf('s');
z=tf('z');
opts = bodeoptions;
opts.FreqUnits = 'Hz';
fc=100;
wn = 2*pi*fc;
fs=10e3;
Ts=1/fs;
%% Filtro segundo orden
Q=4;
zeta = 1/(2*Q);
F2th = 1/[(s/wn)^2 + 2*zeta*(s/wn) + 1]

bode(F2th);
F2thD=c2d(F2th,Ts)
figure;
step(F2th,F2thD);

polos=pole(F2th)
r1=-((wn/Q)+sqrt((wn/Q)^2 - 4*(wn^2)))/2
r2=-((wn/Q)-sqrt((wn/Q)^2 - 4*(wn^2)))/2

%% Cálculos para parametrizar el filtro de segundo orden en C
close all;
```

```

a0=1
a1=-(exp(r1*Ts)+exp(r2*Ts));
a2=exp(r1*Ts)*exp(r2*Ts);

K=a0+a1+a2;

Hz_estimada= (K)/(a0*z^2+a1*z^1+a2)
Hz_estimada.Ts=Ts;
bode(F2thD, Hz_estimada, opts)

input=ones(1,100);
output=filter(cell2mat(Hz_estimada.Numerator), cell2mat(Hz_estimada.
    Denominator), input);
output(1:20)

F2thD
Hz_estimada

```

2.1.8. Respuesta en frecuencia del filtro

En la siguiente gráfica se muestra la respuesta del filtro de segundo orden paso bajo, y su respuesta digital con la aproximación de mapeo de polos que hemos realizado.

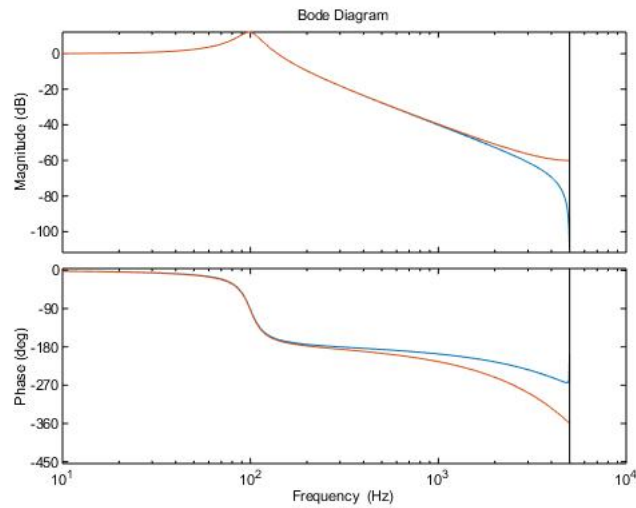


Figura 3: Respuesta en frecuencia del filtro paso bajo de segundo orden

2.1.9. Implementación

Al método software que se encarga de ejecutar este filtrado, se le deberán pasar los siguientes parámetros:

1. Frecuencia de muestro
2. Frecuencia de corte

3. Factor de calidad

Con estos parámetros el método software se encargará de extraer los coeficientes como se ha visto en los puntos anteriores y, realizar el filtrado de la muestra de entrada.

2.2. Filtro paso banda

2.2.1. Transformada de Laplace del filtro

$$Hs(s) = \frac{\frac{2\zeta}{\omega_0}s}{\left(\frac{s}{\omega_0}\right)^2 + 2\zeta\frac{s}{\omega_0} + 1} \quad (41)$$

2.2.2. Mapeo de polos y ceros en el plazo z

La función de transferencia anterior se puede expresar como el producto de dos polos complejos conjugados, un cero en el origen y una ganancia.

$$Hs(s) = \frac{Ks}{(s+p)(s+\hat{p})} \quad (42)$$

donde $p = -\sigma + j\omega$ y $\hat{p} = -\sigma - j\omega$. Mapeando (27) en el plano z obtenemos

$$H(z) = \frac{K(z-1)}{(z-e^{(-\sigma+j\omega)T})(z-e^{(-\sigma-j\omega)T})} \quad (43)$$

esta función de transferencia también la podemos expresar como :

$$H(z) = \frac{K(z-1)}{z^2 - z(e^{(-\sigma+j\omega)T} + e^{(-\sigma-j\omega)T}) + e^{-2\sigma T}} \quad (44)$$

2.2.3. Valor de la ganancia

Buscamos determinar el valor de la ganancia K para obtener el mismo resultado en el filtro digital y analógico en el rango de frecuencias de los filtros. Comparamos la ganancia de ambos filtros a la misma frecuencia

2.2.4. Cálculo de ganancia a frecuencias de paso

Ahora realizamos el límite de la funciones H(s) y la anterior H(z) en la frecuencia de paso ω_0 , al igualar el resultado de ambas f.d.t podemos obtener la ganancia K necesaria para obtener la misma respuesta en el dominio continuo y en el dominio discreto.

$$\lim_{\omega \rightarrow \omega_0} Hs(j\omega) = 1 \quad (45)$$

$$\lim_{\omega \rightarrow \omega_0} H(e^{j\omega}) = K \frac{e^{j\omega_0 T} - 1}{a_0 e^{j2\omega_0 T} + a_1 e^{j\omega_0 T} + a_2} \quad (46)$$

Igualando 39 y 40 obtendremos el valor final de K.

2.2.5. Ecuación final en diferencias

$$H(z) = \frac{K(z-1)}{z^2 - z(e^{(-\sigma+j\omega)T} + e^{(-\sigma-j\omega)T}) + e^{-2\sigma T}} \quad (47)$$

La ecuación en diferencias es:

$$a_0 y_n = K(x_{n-2} + x_{n-1}) - [a_1 y_{n-1} + a_2 y_{n-2}] \quad (48)$$

2.2.6. Coeficientes del filtro

Siguiendo la estructura estándar de filtros digitales:

$$a_0 y[n] = (b_0 x[n] + b_1 x[n-1] + \dots) - (a_1 y[n-1] + a_2 y[n-2] + \dots) \quad (49)$$

los coeficientes del numerador son:

$$b_0 = 1 \quad (50)$$

$$b_1 = -1 \quad (51)$$

los coeficientes del denominador son:

$$a_0 = 1 \quad (52)$$

$$a_1 = -(e^{(-\sigma+j\omega)T} + e^{(-\sigma-j\omega)T}) \quad (53)$$

$$a_2 = e^{-2\sigma T} \quad (54)$$

2.2.7. Código de Matlab

```
close all;
s=tf('s');
z=tf('z');
opts = bodeoptions;
opts.FreqUnits = 'Hz';
fc=1e3;
wn = 2*pi*fc;
fs=10e3;
Ts=1/fs;
%% Filtro segundo orden
Q=12;
zeta = 1/(2*Q);
F2th = (2*zeta/wn)*s / [(s/wn)^2 + 2*zeta*(s/wn) + 1]

bode(F2th);
F2thD=c2d(F2th,Ts)
figure;
step(F2th,F2thD);

polos=pole(F2th)
r1=-((wn/Q)+sqrt((wn/Q)^2 - 4*(wn^2)))/2
r2=-((wn/Q)-sqrt((wn/Q)^2 - 4*(wn^2)))/2

%% Clculos para parametrizar el filtro de segundo orden en C
close all;
a0=1
a1=-(exp(r1*Ts)+exp(r2*Ts));
```

```

a2=exp(r1*Ts)*exp(r2*Ts);

%%Calculo de ganancia en wn
num_in_wn=exp(i*wn*Ts)-1
den_in_wn=a0*exp(i*2*wn*Ts)+a1*exp(i*wn*Ts)+a2
gain_in_wn=1;
K1=gain_in_wn/abs(num_in_wn/den_in_wn);

Ffinal=(K1*(z-1))/(a0*z^2+a1*z+a2)
Ffinal.Ts=Ts;
figure;
bode(F2thD, Ffinal, opts)

input=ones(1,100);
output=filter(cell2mat(Ffinal.Numerator), cell2mat(Ffinal.Denominator),
    input);
output(1:20)

```

2.2.8. Respuesta en frecuencia del filtro

En la siguiente gráfica se muestra 4 la respuesta del filtro de segundo orden paso banda, y su respuesta digital con la aproximación de mapeo de polos/ceros que hemos realizado.

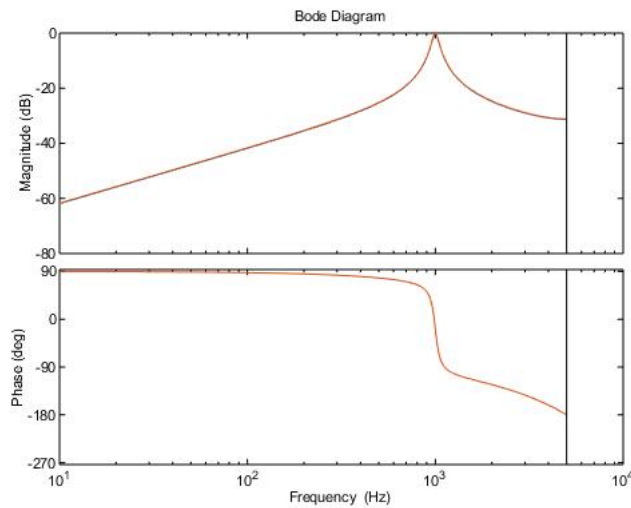


Figura 4: Respuesta en frecuencia del filtro paso banda de segundo orden

2.2.9. Implementación

Al método software que se encarga de ejecutar este filtrado, se le deberán pasar los siguientes parámetros:

1. Frecuencia de muestro
2. Frecuencia de corte

3. Factor de calidad

Con estos parámetros el método software se encargará de extraer los coeficientes como se ha visto en los puntos anteriores y, realizar el filtrado de la muestra de entrada.

2.3. Filtro paso alto

2.3.1. Transformada de Laplace del filtro

$$Hs(s) = \frac{\left(\frac{s}{\omega_0}\right)^2}{\left(\frac{s}{\omega_0}\right)^2 + 2\zeta\frac{s}{\omega_0} + 1} \quad (55)$$

2.3.2. Mapeo de polos y ceros en el plazo z

La función de transferencia anterior se puede expresar como el producto de dos polos complejos conjugados, un cero doble en el origen, y una ganancia.

$$Hs(s) = \frac{Ks^2}{(s+p)(s+\hat{p})} \quad (56)$$

Mapeando 48 en el plano z obtenemos

$$H(z) = \frac{K(z-1)^2}{(z-e^{(-\sigma+j\omega)T})(z-e^{(-\sigma-j\omega)T})} \quad (57)$$

esta función de transferencia también la podemos expresar como :

$$H(z) = \frac{K(z-1)^2}{z^2 - z(e^{(-\sigma+j\omega)T} + e^{(-\sigma-j\omega)T}) + e^{-2\sigma T}} \quad (58)$$

2.3.3. Valor de la ganancia

Buscamos determinar el valor de la ganancia K para obtener el mismo resultado en el filtro digital y analógico en el rango de frecuencias de los filtros. Comparamos la ganancia de ambos filtros a la misma frecuencia

2.3.4. Cálculo de ganancia a frecuencias de paso

La función transferencia H(z) también la podemos expresar de la siguiente manera:

$$H(z) = \frac{K(b_0z^2 + b_1z + b_2)}{a_0z^2 + a_1z + a_0} \quad (59)$$

Ahora realizamos el límite de las funciones H(s) y la anterior H(z) en la frecuencia de paso ω_0 , al igualar el resultado de ambas f.d.t podemos obtener la

ganancia K necesaria para obtener la misma respuesta en el dominio continuo y en el dominio discreto.

$$\lim_{\omega \rightarrow \omega_0} Hs(j\omega) = \frac{1}{2\zeta\sqrt{1-\zeta^2}} \quad (60)$$

$$\lim_{\omega \rightarrow \omega_0} H(e^{j\omega}) = K \frac{b_0 e^{j2\omega_0 T} + b_1 e^{j\omega_0 T} + b_2}{a_0 e^{j2\omega_0 T} + a_1 e^{j\omega_0 T} + a_2} \quad (61)$$

Igualando 52 y 53 obtendremos el valor final de K.

2.3.5. Ecuación final en diferencias

$$H(z) = \frac{K(z-1)^2}{z^2 - z(e^{(-\sigma+j\omega)T} + e^{(-\sigma-j\omega)T}) + e^{-2\sigma T}} \quad (62)$$

La ecuación en diferencias es:

$$a_0 y_n = K(b_0 x_n + b_1 x_{n-1} + b_2 x_{n-2}) - [a_1 y_{n-1} + a_2 y_{n-2}] \quad (63)$$

2.3.6. Coeficientes del filtro

Siguiendo la estructura estándar de filtros digitales:

$$a_0 y[n] = (b_0 x[n] + b_1 x[n-1] + \dots) - (a_1 y[n-1] + a_2 y[n-2] + \dots) \quad (64)$$

los coeficientes finales que obtenemos para el numerador son:

$$b_0 = 1 \quad (65)$$

$$b_1 = -2 \quad (66)$$

$$b_2 = 1 \quad (67)$$

los coeficientes finales que obtenemos para el denominador son:

$$a_0 = 1 \quad (68)$$

$$a_1 = -(e^{(-\sigma+j\omega)T} + e^{(-\sigma-j\omega)T}) \quad (69)$$

$$a_2 = e^{-2\sigma T} \quad (70)$$

2.3.7. Código de Matlab

```
close all;
s=tf('s');
opts = bodeoptions;
opts.FreqUnits = 'Hz';
fc=2e3;
wn = 2*pi*fc;
fs=10e3;
Ts=1/fs;
%% Filtro segundo orden
```



```

Q=20;
zeta = 1/(2*Q);
F2th = (s/wn)^2/[(s/wn)^2 + 2*zeta*(s/wn) + 1]

bode(F2th,opts);
F2thD=c2d(F2th,Ts)
figure;
bode(F2thD);

polos=pole(F2thD)
ceros=zero(F2thD)
r1=-((wn/Q)+sqrt((wn/Q)^2 - 4*(wn^2)))/2
r2=-((wn/Q)-sqrt((wn/Q)^2 - 4*(wn^2)))/2

%% Clculos para parametrizar el filtro de segundo orden en C
close all;
a0=1
a1=-(exp(r1*Ts)+exp(r2*Ts));
a2=exp(r1*Ts)*exp(r2*Ts);

NumZ=((z-1)/wn)^2
zeros=zero(NumZ)
b0=1;
b1=-2;
b2=1;

%Calculo de ganancia
num_in_wn=(b0*exp(i*2*wn*Ts)+b1*exp(i*wn*Ts)+b2)
den_in_wn=a0*exp(i*2*wn*Ts)+a1*exp(i*wn*Ts)+a2
gain_in_wn=1/((2*zeta)*sqrt(1-zeta^2));

K1=gain_in_wn/abs(num_in_wn/den_in_wn);

Hz_estimada= (K1)*(b0*z^2+b1*z^1+b2)/(a0*z^2+a1*z^1+a2);
Hz_estimada.Ts=Ts;
bode(F2th,Hz_estimada,opts)

input=ones(1,100);
output=filter(cell2mat(Hz_estimada.Numerator),cell2mat(Hz_estimada.
    Denominator),input);
output(1:20)

```

2.3.8. Respuesta en frecuencia del filtro

En la siguiente gráfica se muestra 5 la respuesta del filtro de segundo orden paso alto, y su respuesta digital con la aproximación de mapeo de polos/ceros que hemos realizado.

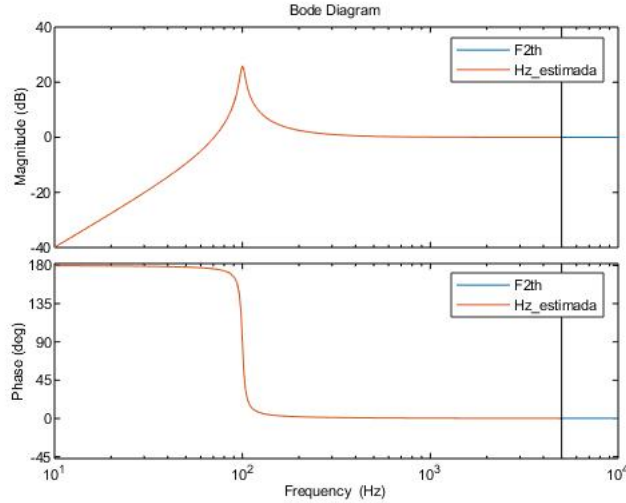


Figura 5: Respuesta en frecuencia del filtro paso alto de segundo orden

2.3.9. Implementación

Al método software que se encarga de ejecutar este filtrado, se le deberán pasar los siguientes parámetros:

1. Frecuencia de muestro
2. Frecuencia de corte
3. Factor de calidad

Con estos parámetros el método software se encargará de extraer los coeficientes como se ha visto en los puntos anteriores y, realizar el filtrado de la muestra de entrada.

3. Filtros de orden superior

3.1. Descripción

Las formas directas no resultan útiles cuando el orden de la función a realizar es elevado, ya que su sensibilidad con respecto a la cuantificación de los coeficientes aumenta enormemente con el grado de la función. Para evitar este problema se suelen descomponer las funciones del sistema como suma o producto de funciones de primer y segundo orden, las cuales podrán realizarse según algunas de las formas ya vistas en el documento.

La forma de realizar el filtro es en cascada, y sigue los siguiente pasos:

1. Factorizamos los polinomios del numerador y denominador, agrupando las raíces conjugadas para que no aparezcan coeficientes complejos.

$$H(z) = A \frac{\prod_{k=1}^{M_1} (1 - g_k z^{-1}) \prod_{k=1}^{M_2} (1 - h_k z^{-1})(1 - h_k^* z^{-1})}{\prod_{k=1}^{N_1} (1 - c_k z^{-1}) \prod_{k=1}^{N_2} (1 - d_k z^{-1})(1 - d_k^* z^{-1})} \quad (71)$$

2. Expresamos $H(z)$ como producto de funciones de segundo orden, donde N es el menor entero mayor o igual que $N/2$.

$$H(z) = \prod_{k=1}^{N_s} \frac{b_{0k} + b_{1k} z^{-1} + b_{2k} z^{-2}}{1 - a_{1k} z^{-1} - a_{2k} z^{-2}} \quad (72)$$

3. Se realizan las funciones de segundo orden en la forma directa II y se conectan en cascada como indica la figura.

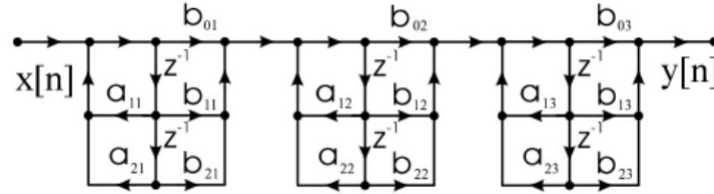


Figura 6: Ejemplo de tres secciones de segundo orden en forma directa II

3.2. Implementación

Al método software que se encarga de ejecutar este filtrado, se le deberán pasar los coeficientes del filtro en dos arrays, uno que identifica los coeficientes a_k de las secciones de segundo orden, y otro con los coeficientes b_k . El método por sí mismo se encarga de ejecutar a partir de estos coeficientes una estructura en cascada como la vista en la figura 6.

3.3. Ejemplo en Matlab de la obtención de secciones de segundo orden

En el siguiente script se puede ver un ejemplo de un filtro Butterworth con dos secciones de segundo orden.

```
close all;
Fs=8000; % Sampling
fc=800; % Cut off
fstop= 1600; % Fstop
Rp=1; % Passband ripple in db
Rs=20; % stop band attenuation in db
Wp=2*fc / Fs;
Ws=2*fstop / Fs;
```

```

[N,Wn]=buttord(Wp,Ws,Rp,Rs);
[b,a]=butter(N,Wn);
[Z,P,K]=tf2zp(b,a);
[sos,G]=zp2sos(Z,P,K);

input=ones(1,100);
output=filter(b,a,input);
output(1:20)
plot(output)

```