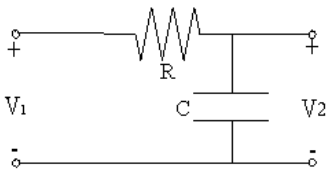
Funciones de transferencia de Filtros

Filtros pasabajas

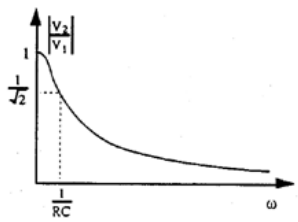
Si tenemos el siguiente circuito:

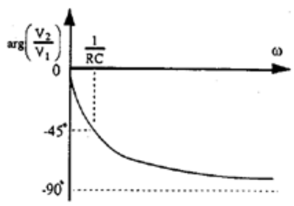


Si calculamos la función de transferencia de voltaje:

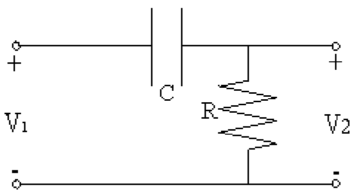
$$\frac{V_2}{V_1} = \frac{\dfrac{-j}{\omega C}}{R - \dfrac{j}{\omega C}} = \frac{\dfrac{-j}{\omega C}}{\dfrac{\omega RC - j}{\omega C}} = \frac{-j}{\omega RC - j} = \frac{1}{1 + j\omega RC}$$

Esto representado queda:



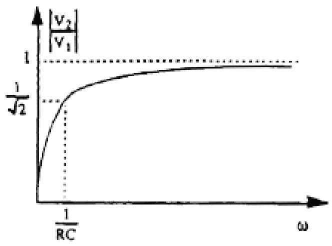


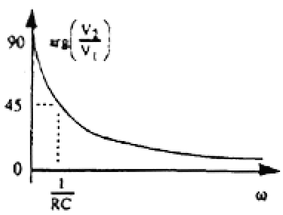
Filtro pasa altas



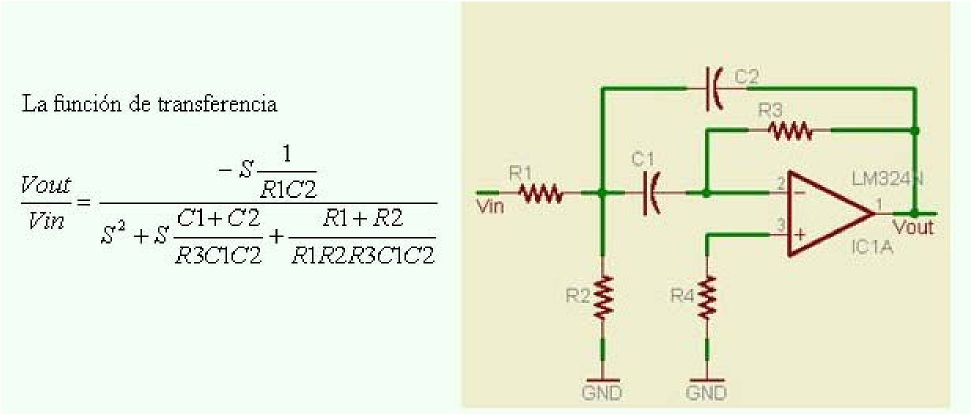
La función de transferencia es: $\frac{V_2}{V_1} = \frac{R}{R - \dfrac{j}{\omega C}} = \frac{\omega RC}{\omega RC - j} = \frac{j\omega RC}{1 + j\omega RC}$

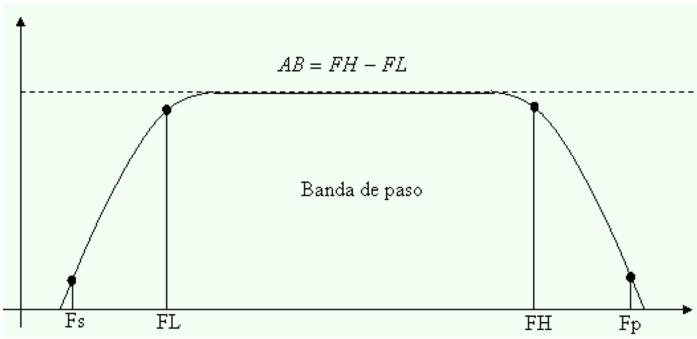
Representando la función obtenemos que queda de la siguiente forma:





Filtro pasabanda





1. Filtros analógicos Butterworth con Matlab

En Matlab podemos encontrar la instrucción *Butter* , este comando diseña filtros Butterwoth 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.

En el dominio analógico tenemos dos opciones:

- La instrucción para generar un filtro del grado que el usuario desee, así como la manipulación de la frecuencia de corte.(esta instrucción genera los polos y ceros necesarios)
- El comando para que a partir del grado, tipo y amortiguamiento, se obtengan los polos, zeros y ganancia que pueda tener el filtro.

Ambas opciones generan los coeficientes de s de la función de transferencia:

$$H(s)=\frac{B(s)}{A(s)}=\frac{b(1)s^{n}+b(2)s^{n-1}+\ldots\ldots+b(n+1)}{s^{n}+a(2)s^{n-1}+\ldots\ldots+a(n+1)}$$

A continuación diseñaremos un filtro pasa altas de noveno orden que corte en 300Hz.

[b,a]=butter(9,300/500,'high'); [b,a]=butter(n,Wn,'Tipo'), se establecen las

características del filtro.

Freqz(b,a,128,1000) Una vez establecido el filtro lo graficaremos

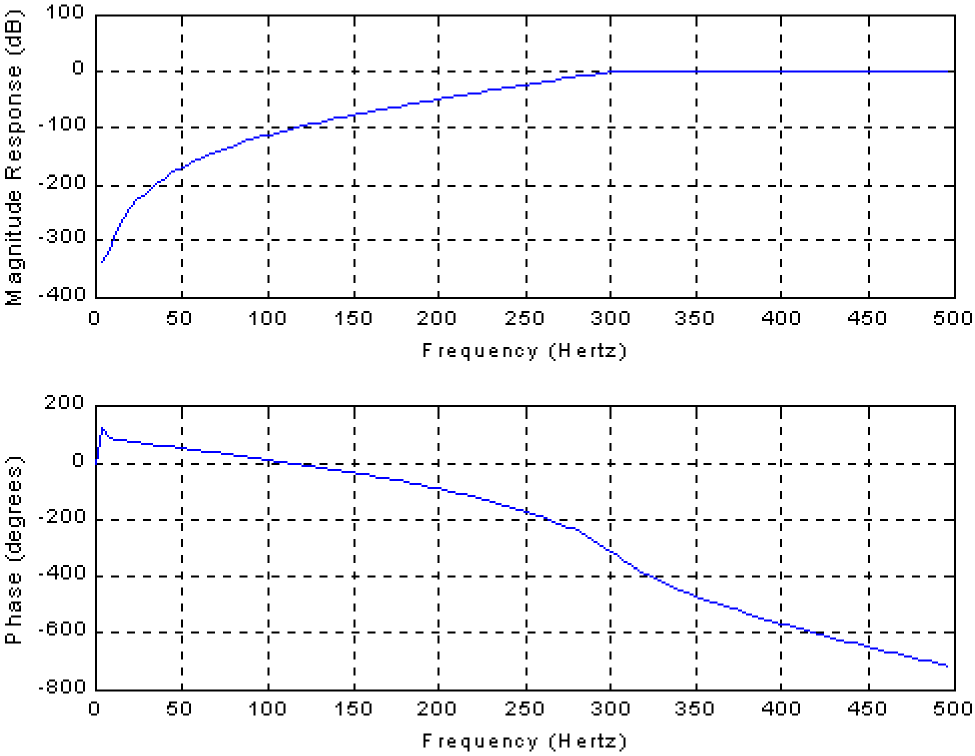


Figura 1.1.1 Gráfica en decibeles y en fase de un filtro Butterworth de 9no grado.

En la figura 1.1.1 podemos observar que al llegar a 300Hz se encuentra el primer polo y se empieza a atenuar las frecuencias arriba de éste punto. De la misma forma en la gráfica de fase se puede ver que tiene una respuesta lineal.

A continuación se graficará la respuesta a impulso de un filtro butterworth pasa bandas de 100 a 200 Hz de quinto orden.

n=5;Wn=[100 200]/500; parámetros del filtro

[b,a]=butter(n,Wn); comando para un filtro butterworth

[y,t]=impz(b,a,101); cambio de dominio

stem(t,y)

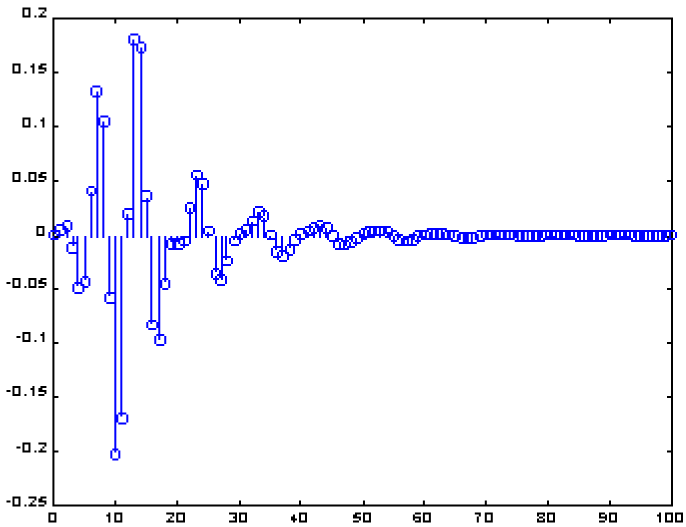


Figura 1.1.2 Respuesta a impulso de un BPF de quinto grado.

En la gráfica podemos observar el efecto del filtro, aunque aquí no a parecen las frecuencias, se puede ver que hay un rango en donde la señal puede pasar sin ser atenuada, este ejemplo nos ayuda a visualizar por donde se encuentra las frecuencias de corte y qué tipo de filtro es.

Ahora bien diseñemos un filtro butterworth pasa bajas de quinto orden que corte en 300Hz.

[b,a]=butter(5,300/500,'high'); Se establece los parámetros del filtro

Freqz(b,a,128,1000); Gráfica de la frecuencia en decibeles y defazamiento de ángulo.

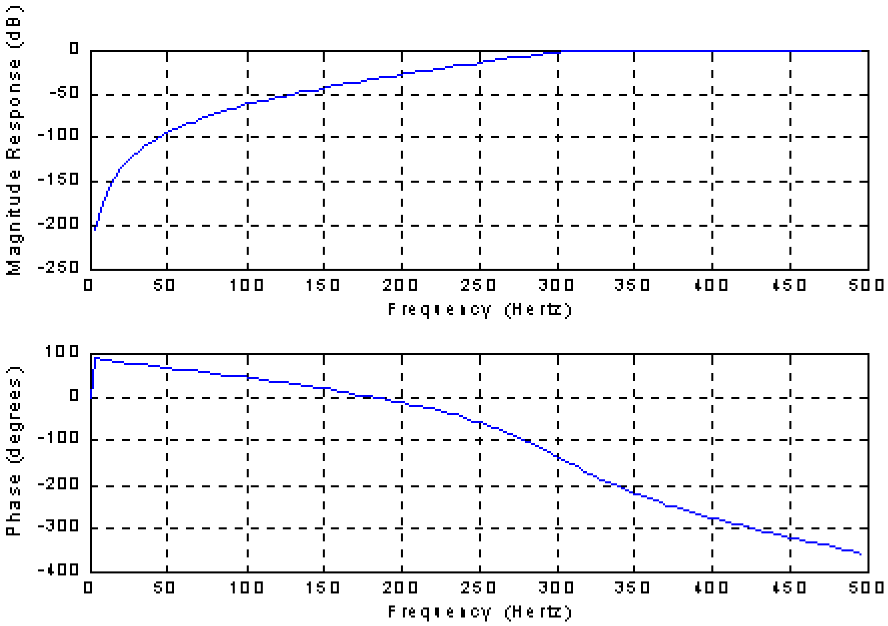


Figura 1.1.3 LPF, butterworth de quinto grado con Fc=300Hz.

En la figura se puede observar que a partir de 300 Hz la señal se empieza a atenuar por lo cual las frecuencias mayores a ésta se filtrarán. Es importante notar que la banda de paso es lineal y plana característica de un filtro butterworth.

Tema :respuesta en frecuencia de filtros

Comando **freqz** respuesta en frecuencia de filtros digitales

Comando **freqs** respuesta en frecuencia de filtros analógicos

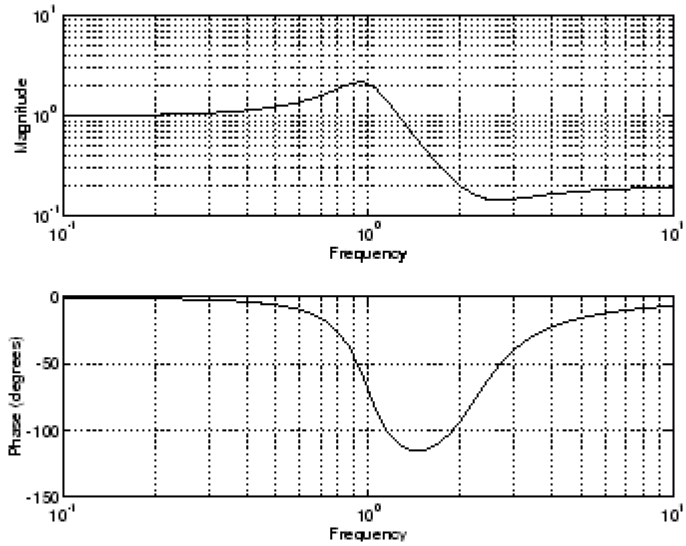
`freqs` works only for real input systems and positive frequencies.

Examples

Find and graph the frequency response of the transfer function given by:

$$H(s) = \frac{0.2s^2 + 0.3s + 1}{s^2 + 0.4s + 1}$$

```
a = [1 0.4 1];
b = [0.2 0.3 1];
w = logspace(-1,1);
freqs(b,a,w)
```



You can also create the plot with

```
h = freqs(b,a,w);
mag = abs(h);
phase = angle(h);
subplot(2,1,1), loglog(w,mag)
subplot(2,1,2), semilogx(w,phase)
```

To convert to hertz, degrees, and decibels, use

```
f = w/(2*pi);
mag = 20*log10(mag);
phase = phase*180/pi;
```

Algorithm

`freqs` evaluates the polynomials at each frequency point, then divides the numerator response by the denominator response:

```
s = i*w;
h = polyval(b,s)./polyval(a,s);
```