DISEÑO DE FILTROS CON MATLAB

Contenido:

I. Procedimiento de diseño de filtros.

El diseño de un filtro consiste en obtener un circuito en el que la respuesta frecuencial de su función de transferencia satisfaga una especificación dada. Para ello, se sigue un proceso que divide el problema en tres etapas diferenciadas:

a) Determinar matemáticamente la función de transferencia cuya respuesta frecuencial mejor aproxima las especificaciones dadas para la respuesta frecuencial del filtro.

$$G(s) = \frac{V_o}{V_i} = \frac{b_m s^m + b_{m-1} s^{m-1} + \ldots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \ldots + a_1 s + a_0}$$

b) Descomponer los polinomios resultantes en factores de segundo orden que permitan su implementación por una secuencia de etapas bicuadráticas conectadas en cascada.

$$G(s) = \prod_{j=1}^{N} G_j(s) = \prod_{j=1}^{N} K_j \frac{s^2 + e_j s + f_j}{s^2 + c_j s + d_j}$$

c) Seleccionar el circuito con el que se implementará cada etapa del filtro y estimar los valores de los componentes que los circuitos elegidos contienen.

II. Especificaciones de un filtro.

Las especificaciones de un filtro suelen darse a partir de su función de pérdidas. Si ***G(s)*** es la función de transferencia de un filtro, su función de pérdidas, ***H(s)***, se define como:

$$H(s) = \frac{1}{G(s)}$$

Las especificaciones de un filtro real son las que se muestran en las figuras siguientes. La función de pérdidas del filtro tiene que quedar fuera de la zona sombreada.

III. Funciones de aproximación de filtros.

- **Butterworth.** The magnitude response of a Butterworth filter is maximally flat in the passband and monotonic overall.

- **Chebyshev.** The magnitude response of a Chebyshev type I filter is equiripple in the passband and monotonic in the stopband.

- **Cauer o elíptica.** The magnitude response of an elliptic filter is equiripple in both the passband and the stopband.

IV. Obtención del orden de un filtro con MATLAB.

- **Butterworth.**

```
[N, Wn] = BUTTORD(Wp, Ws, Rp, Rs, 's')
```

Returns the order N of the lowest order analog Butterworth filter that loses no more than Rp dB in the passband and has at least Rs dB of attenuation in the stopband. Wp and Ws are the passband and stopband edge frequencies in **radians/second**.

`BUTTORD` also returns Wn in **rad/s**, the Butterworth natural frequency (or, the "3 dB frequency") to use with `BUTTER` to achieve the specifications.

- **Chebyshev.**

```
[N, Wnc] = CHEB1ORD(Wp, Ws, Rp, Rs, 's')
```

Returns the order N of the lowest order analog Chebyshev Type I filter that loses no more than Rp dB in the passband and has at least Rs dB of attenuation in the stopband. Wp and Ws are the passband and stopband edge frequencies in **radians/second**.

`CHEB1ORD` also returns Wnc in **rad/s**, the Chebyshev natural frequency at which the magnitude response of the filter is -Rp dB. Wnc is used with `CHEBY1` to achieve the specifications.

- **Cauer o elíptica.**

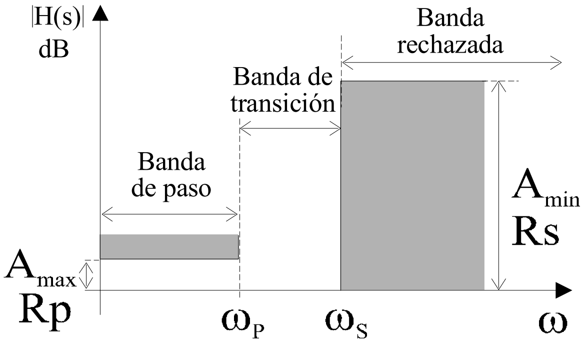
```
[N, Wne] = ELLIPORD(Wp, Ws, Rp, Rs, 's')
```

Returns the order N of the lowest order analog elliptic filter that loses no more than Rp dB in the passband and has at least Rs dB of attenuation in the stopband. Wp and Ws are the passband and stopband edge frequencies in **radians/second**. `ELLIPORD` also returns Wne in **rad/s**, the elliptic natural frequency to use with `ELLIP` to achieve the specifications. Wne (angular cutoff frequency) is the edge of the passband, at which the magnitude response of the filter is -Rp dB.

NOTE: If Rs is much much greater than Rp, or Wp and Ws are very close, the estimated order can be infinite due to limitations of numerical precision.

V. Diseño de filtros con MATLAB.

1) Filtros de paso bajo.



- **Butterworth.**

```
[N, Wn] = BUTTORD(Wp, Ws, Rp, Rs, 's')
```

```
[NUM, DEN]=BUTTER (N,Wn,'s')
```

Designs an Nth order lowpass analog Butterworth filter and returns the filter coefficients in length N+1 vectors NUM (numerator) and DEN (denominator). The coefficients are listed in descending powers of s. The cutoff frequency Wn is in rad/s. When Rp is chosen as 3 dB, the Wn in BUTTER is equal to Wp in BUTTORD.

- **Chebyshev.**

```
[N, Wnc] = CHEB1ORD(Wp, Ws, Rp, Rs, 's')
```

```
[NUM, DEN]=CHEBY1(N,Rp,Wnc,'s')
```

Designs an Nth order lowpass analog Chebyshev Type I filter with Rp decibels of peak-to-peak ripple in the passband and returns the filter coefficients in length N+1 vectors NUM (numerator) and DEN (denominator). The coefficients are listed in descending powers of s. The cutoff frequency Wnc is in rad/s.

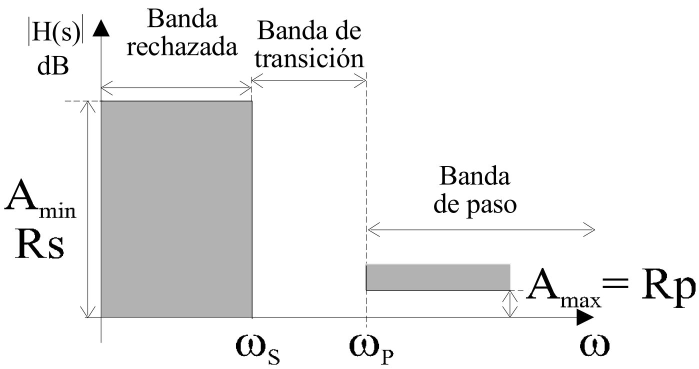
- **Cauer o elíptica.**

```
[N, Wne] = ELLIPORD(Wp, Ws, Rp, Rs, 's')
```

```
[NUM, DEN]=ELLIP(N,Rp,Rs,Wne,'s')
```

Designs an Nth order lowpass analog elliptic filter with Rp decibels of peak-to-peak ripple and a minimum stopband attenuation of Rs decibels. `ELLIP` returns the filter coefficients in length N+1 vectors NUM (numerator) and DEN (denominator). The coefficients are listed in descending powers of s. The cutoff frequency Wne is in rad/s.

2) Filtros de paso alto.



- **Butterworth.**

```
[N, Wn] = BUTTORD(Wp, Ws, Rp, Rs, 's')
```

```
[NUM, DEN]=BUTTER(N,Wn,'high','s')
```

Designs an Nth order highpass analog Butterworth filter and returns the filter coefficients in length N+1 vectors NUM (numerator) and DEN (denominator). The coefficients are listed in descending powers of s. The cutoff frequency Wn is in rad/s. When Rp is chosen as 3 dB, the Wn in BUTTER is equal to Wp in BUTTORD.

- **Chebyshev.**

```
[N, Wnc] = CHEB1ORD(Wp, Ws, Rp, Rs, 's')
```

```
[NUM, DEN]=CHEBY1(N,Rp,Wnc,'high','s')
```

Designs an Nth order highpass analog Chebyshev Type I filter with Rp decibels of peak-to-peak ripple in the passband and returns the filter coefficients in length N+1 vectors NUM (numerator) and DEN (denominator). The coefficients are listed in descending powers of s. The cutoff frequency Wnc is in rad/s.

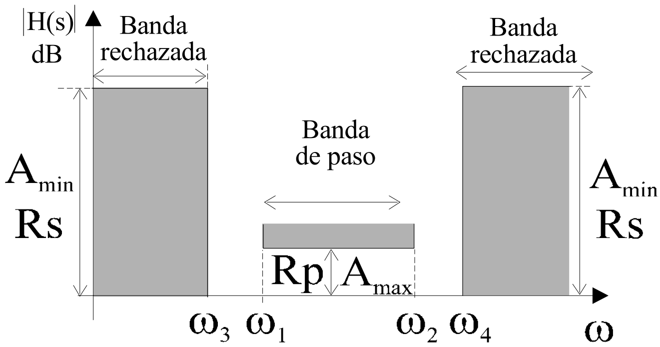
- **Cauer o elíptica.**

```
[N, Wne] = ELLIPORD(Wp, Ws, Rp, Rs, 's')
```

```
[NUM, DEN]=ELLIP(N,Rp,Rs,Wne,'high','s')
```

Designs an Nth order lowpass analog elliptic filter with Rp decibels of peak-to-peak ripple and a minimum stopband attenuation of Rs decibels. `ELLIP` returns the filter coefficients in length N+1 vectors NUM (numerator) and DEN (denominator). The coefficients are listed in descending powers of s. The cutoff frequency Wne is in rad/s.

3) Filtros de paso banda.



Se definen:
Wp=[ω_1,ω_2]
Ws=[ω_3,ω_4]

- **Butterworth.**

```
[N, Wn] = BUTTORD(Wp, Ws, Rp, Rs, 's')
```

`BUTTORD` also returns Wn=[Wn1,Wn2], the Butterworth low (Wn1) and high (Wn2) natural frequencies (or, the "3 dB frequencies") to use with `BUTTER` to achieve the specifications.

```
[NUM, DEN]=BUTTER(N,Wn,'s')
```

If Wn is a two-element vector, Wn = [Wn1 Wn2], `BUTTER` returns an order 2N bandpass analog filter with passband Wn1 < W < Wn2 and returns the filter coefficients in length 2N+1 vectors NUM (numerator) and DEN (denominator). The coefficients are listed in descending powers of s. When Rp is chosen as 3 dB, the Wn in `BUTTER` is equal to Wp in `BUTTORD`.

- **Chebyshev.**

```
[N, Wnc] = CHEB1ORD(Wp, Ws, Rp, Rs, 's')
```

`CHEB1ORD` also returns Wnc=[Wn1,Wn2], the Chebyshev low (Wn1) and high (Wn2) natural frequencies to use with `CHEBY1` to achieve the specifications.

```
[NUM, DEN]=CHEBY1(N,Rp,Wnc,'s')
```

If Wnc is a two-element vector Wnc=[Wn1,Wn2] with Wn1<Wn2, then `cheby1(n,Rp,Wnc,'s')` returns an order 2N bandpass analog Chebyshev Type I filter with passband Wn1 < w< Wn2 and with Rp decibels of peak-to-peak ripple in the passband and returns the filter coefficients in length 2N+1 vectors NUM (numerator) and DEN (denominator). The coefficients are listed in descending powers of s. The cutoff frequency Wnc is in rad/s.

- **Cauer o elíptica.**

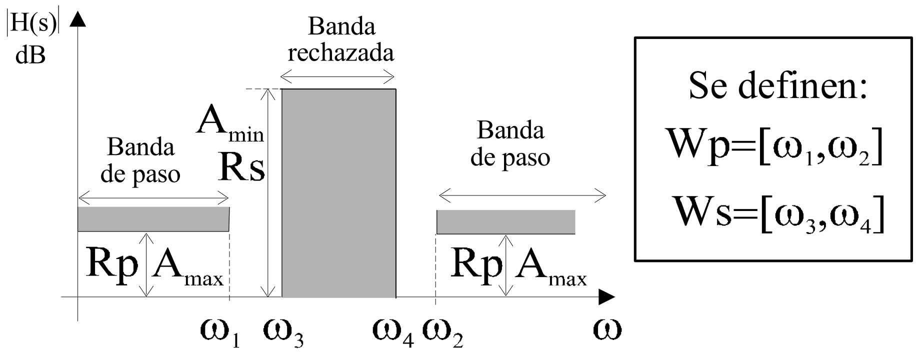
```
[N, Wne] = ELLIPORD(Wp, Ws, Rp, Rs, 's')
```

`ELLIPORD` also returns Wne=[Wn1,Wn2] in rad/s, the elliptic low (Wn1) and high (Wn2) natural frequencies to use with `ELLIP` to achieve the specifications.

```
[NUM, DEN]=ELLIP(N,Rp,Rs,Wn,'s')
```

If Wne is a two-element vector Wne=[Wn1,Wn2] with Wn1<Wn2, then `ELLIP(n,Rp,Rs,Wne,'s')` returns an order 2N bandpass analog elliptic filter with passband Wn1<w<Wn2 and with Rp decibels of peak-to-peak ripple and a minimum stopband attenuation of Rs decibels. `ELLIP` returns the filter coefficients in length 2N+1 vectors NUM (numerator) and DEN (denominator). The coefficients are listed in descending powers of s.

4) Filtros de rechazo de banda.



- **Butterworth.**

```
[N, Wn] = BUTTORD(Wp, Ws, Rp, Rs, 's')
```

BUTTORD also returns Wn=[Wn1,Wn2], the Butterworth low (Wn1) and high (Wn2) natural frequencies (or, the "3 dB frequencies") to use with BUTTER to achieve the specifications.

```
[NUM, DEN]=BUTTER(N,Wn,'stop','s')
```

Designs an 2Nth order bandstop analog Butterworth filter and returns the filter coefficients in length 2N+1 vectors NUM (numerator) and DEN (denominator). The coefficients are listed in descending powers of s. The cutoff frequency Wn is in rad/s. When Rp is chosen as 3 dB, the Wn in BUTTER is equal to Wp in BUTTORD.

- **Chebyshev.**

```
[N, Wnc] = CHEB1ORD(Wp, Ws, Rp, Rs, 's')
```

CHEB1ORD also returns Wnc=[Wn1,Wn2] in rad/s, the Chebyshev natural frequency to use with CHEBY1 to achieve the specifications.

```
[NUM, DEN]=CHEBY1(N,Rp,Wnc,'stop','s')
```

Designs an 2Nth order bandstop analog Chebyshev Type I filter with Rp decibels of peak-to-peak ripple in the passband and returns the filter coefficients in length 2N+1 vectors NUM (numerator) and DEN (denominator). The coefficients are listed in descending powers of s. The cutoff frequency Wnc is in rad/s.

- **Cauer o elíptica.**

```
[N, Wne] = ELLIPORD(Wp, Ws, Rp, Rs, 's')
```

ELLIPORD also returns Wne=[Wn1,Wn2] in rad/s, the elliptic natural frequency to use with ELLIP to achieve the specifications.

```
[NUM, DEN]=ELLIP(N,Rp,Rs,Wne, 'stop','s')
```

Designs an 2Nth order bandstop analog elliptic filter with Rp decibels of peak-to-peak ripple and a minimum stopband attenuation of Rs decibels. ELLIP returns the filter coefficients in length 2N+1 vectors NUM (numerator) and DEN (denominator). The coefficients are listed in descending powers of s. The cutoff frequency Wne is in rad/s.

5) Operaciones sobre funciones de transferencia con MATLAB.

s=tf('s')

`s = tf('s')` specifies the transfer function H(s) = s (Laplace variable).

g1=tf(NUM, DEN)

Creates a continuous-time transfer function g1 with numerator NUM and denominator DEN.

bode(g1)

Draws the Bode plot of the continuous-time transfer function g1.

[sos,g] = tf2sos(B,A)

tf2sos: Transfer Function to Second Order Section conversion.
`[sos,g] = tf2sos(B,A)` finds a matrix `sos` in second-order sections form and a gain `g` which represent the same system as the one with numerator B and denominator A.

`sos` is an L by 6 matrix with the following structure:

$$\mathrm{sos}=\begin{bmatrix} b01 & b11 & b21 & 1 & a11 & a21 \\ b02 & b12 & b22 & 1 & a12 & a22 \\ . & . & . & . & . & . \\ b0L & b1L & b2L & 1 & a1L & a2L \end{bmatrix}$$

Each row of the `sos` matrix describes a 2nd order transfer function:

$$Hk(s)=\frac{bok+b1k\times s^{-1}+b2k\times s^{-2}}{1+a1k\times s^{-1}+a2k\times s^{-2}}$$

where k is the row index.

`g` is a scalar which accounts for the overall gain of the system. If `g` is not specified, the gain is embedded in the first section. The second order structure thus describes the system H(s) as:

$$H(s)=g\times H1(s)\times H2(s)\times\cdots\times HL(s)$$

[B,A] = sos2tf(sos,g)

sos2tf 2nd-order sections to transfer function model conversion.
`[B,A] = sos2tf(sos,g)` returns the numerator and denominator coefficients `B` and `A` of the linear system given by the gain `g` and the matrix `sos` in second-order sections form.

`sos` is an L by 6 matrix which contains the coefficients of each second-order section in its rows (see **`[sos,g] = tf2sos(B,A)`**).

The system transfer function is the product of the second-order transfer functions of the sections times the gain `g`. If `g` is not specified, it defaults to 1. Each row of the sos matrix describes a 2nd order transfer function as in **`[sos,g] = tf2sos(B,A)`**.

`[Z,P,K] = sos2zp(sos,g)`

`sos2zp` Second-order sections to zero-pole-gain model conversion.
`[Z,P,K] = sos2zp(sos,g)` returns the zeros `Z`, poles `P` and gain `K` of the system given by the gain `g` and the matrix `sos` in second-order sections form.

`sos` is an L by 6 matrix which contains the coefficients of each second-order section in its rows (see **`[sos,g] = tf2sos(B,A)`**).

The system transfer function is the product of the second-order transfer functions of the sections times the gain `g`. If `g` is not specified, it defaults to 1. Each row of the `sos` matrix describes a 2nd order transfer function as in **`[sos,g] = tf2sos(B,A)`**.

`[NUM,DEN] = zp2tf(Z,P,K)`

`zp2tf` Zero-pole to transfer function conversion.
`[NUM,DEN] = zp2tf(Z,P,K)` forms the transfer function:

$$H(s) = \frac{NUM(s)}{DEN(s)}$$

given a set of zero locations in vector `Z`, a set of pole locations in vector `P`, and a gain in scalar `K`. Vectors `NUM` and `DEN` are returned with numerator and denominator coefficients in descending powers of s.

`[Z,P,K] = tf2zp(NUM,DEN)`

`tf2zp` Transfer function to zero-pole conversion.
`[Z,P,K] = tf2zp(NUM,DEN)` finds the zeros, poles, and gains:

$$H(s) = K\frac{(s-z1)(s-z2)\cdots(s-zn)}{(s-p1)(s-p2)\cdots(s-pn)}$$

from a transfer function in polynomial form:

$$H(s) = \frac{NUM(s)}{DEN(s)}$$

Vector `DEN` specifies the coefficients of the denominator in descending powers of s. Matrix `NUM` indicates the numerator coefficients with as many rows as there are outputs. The zero locations are returned in the columns of matrix `Z`, with as many columns as there are rows in `NUM`. The pole locations are returned in column vector `P`, and the gains for each numerator transfer function in vector `K`.

`[MAG, PHASE] = bode(g1,W)`

Return the response magnitude and phase in degrees of the transfer function g1 for the specified frequency vector W.

6) Ejemplo.

Obtener con MATLAB las funciones de transferencia de un filtro paso bajo utilizando las funciones de aproximación de Butterworth, Chebyshev y Cauer. Factorizar la función de transferencia obtenida. Las especificaciones son:

$$ws=2000\times 2\times \pi \text{ rad/seg, } wp=1000\times 2\times \pi \text{ rad/seg, } Amax=0.5dB \text{ y } Amin=40dB$$

```
% Ejemplo de diseño de filtros analogicos con Matlab
% Encontrar las funciones de aproximacion de paso bajo de Butterworth,
% Chebyshev y eliptica para los requerimientos de un filtro:
% Amax=0.5dB, Amin=40dB, fp=1000Hz, fs=2000rad/seg
close all;
clear all;
s=tf('s');
Amax=0.5;
Amin=40;
Rs=Amin;
Rp=Amax;
Wp=2*pi*1000;
Ws=2*pi*2000;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Obtenemos la funcion de aproximacion de Butterworth
[Nb, Wn] = BUTTORD(Wp, Ws, Rp, Rs, 's')
[NUM, DEN]=BUTTER (Nb,Wn,'s');
gbut=tf(NUM,DEN)
hbut=1/gbut;
% Comprobamos que se verifican las especificaciones
w=[Ws,Wp];
[mag]=bode(hbut,w);
magdB=20*log10(mag)
% Factorizamos la funcion de transferencia
[sosb,g]=tf2sos(NUM,DEN); % Ver en Workspace
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Obtenemos la funcion de aproximacion de Chebyshev
[Nc, Wnc] = CHEB1ORD(Wp, Ws, Rp, Rs, 's')
[NUM, DEN]=CHEBY1(Nc,Rp,Wnc,'s');
gcheb=tf(NUM,DEN)
hcheb=1/gcheb;
% Comprobamos que se verifican las especificaciones
[mag]=bode(hcheb,w);
magdB=20*log10(mag)
% Factorizamos la funcion de transferencia
[sosch,g]=tf2sos(NUM,DEN); % Ver en Workspace
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Obtenemos la funcion de aproximacion de Cauer
[Ne, Wne] = ELLIPORD(Wp, Ws, Rp, Rs, 's')
[NUM, DEN]=ELLIP(Ne,Rp,Rs,Wne,'s');
gcauer=tf(NUM,DEN)
hcauer=1/gcauer;
% Comprobamos que se verifican las especificaciones
[mag]=bode(hcauer,w);
magdB=20*log10(mag)
% Factorizamos la funcion de transferencia
[soscauer,g]=tf2sos(NUM,DEN); % Ver en Workspace
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Dibujamos el diagrama de Bode de las tres funciones de perdidas obtenidas
bodemag(hbut,hcheb,hcauer);
title('');
legend(['Butterworth de orden ' num2str(Nb)], ...
    ['Chebyshev de orden ', num2str(Nc)],['Cauer de orden ', num2str(Ne)])
```