EXAMEN UNIDAD III

Diseñe un filtro IIR Chebyshev Tipo 1 de paso bajo para las siguientes especificaciones:

Frecuencia de borde de banda de paso: 1 kHz

Frecuencia de borde de banda de parada: 3 kHz

Frecuencia de muestreo: 10 kHz

Ondulación de banda de paso: 1 dB

Ondulación de la banda de parada: 40 dB

Hallar H(s) y H(z)

SOLUCIÓN

Frecuencias normalizadas

$$\omega_p = \frac{2\pi f_p}{F_T} = \frac{2\pi \times 1000}{10000} = 0.2\pi$$
$$\omega_s = \frac{2\pi f_s}{F_T} = \frac{2\pi \times 3000}{10000} = 0.6\pi$$

Acondicionamiento de frecuencias

By prewarping these frequencies, we get

$$\hat{\Omega}_p = \tan(\omega_p/2) = 0.32492; \hat{\Omega}_s = \tan(\omega_s/2) = 1.3764.$$

For the prototype analog lowpass filter

$$\Omega_p = 1, \Omega_s = \hat{\Omega}_s/\hat{\Omega}_p = 1.3764/0.32492 = 4.236, \alpha_p = 1\,\mathrm{dB}, \alpha_s = 40\,\mathrm{dB}$$

Orden del filtro

$$N \geq \frac{\cosh^{-1}\sqrt{\frac{10^4-1}{10^{0.1}-1}}}{\cosh^{-1}(4.236)} = 2.45$$

Por tal motivo el orden es N=3

Table 5.2 List of normalized Type 1 Chebyshev transfer functions for passband ripple = 1 dB

N	Denominator of $H_N(s)$	H_0
1	$s + 1.9652$	1.9652
2	$s^2 + 1.0977s + 1.1025$	0.98261
3	$s^3 + 0.98834s^2 + 1.2384s + 0.49131$	0.49131
4	$s^4 + 0.95281s^3 + 1.4539s^2 + 0.74262s + 0.27563$	0.24565
5	$s^5 + 0.93682s^4 + 1.6888s^3 + 0.9744s^2 + 0.58053s + 0.12283$	0.12283

De la tabla de Chebyshev se obtiene la función de transferencia

$$H_N(s) = \frac{0.49131}{s^3 + 0.988s^2 + 1.238s + 0.49131}$$

The transfer function corresponding to $\hat{\Omega}_p = 0.32492$ is obtained by substituting $s = (s/\hat{\Omega}_p) = (s/0.32492)$ in $H_N(s)$; hence,

$$H_a(s) = \frac{0.016849}{s^3 + 0.32099s^2 + 0.13068s + 0.016849}$$

The digital transfer function $H_{LP}(z)$ of the desired lowpass filter is now obtained as

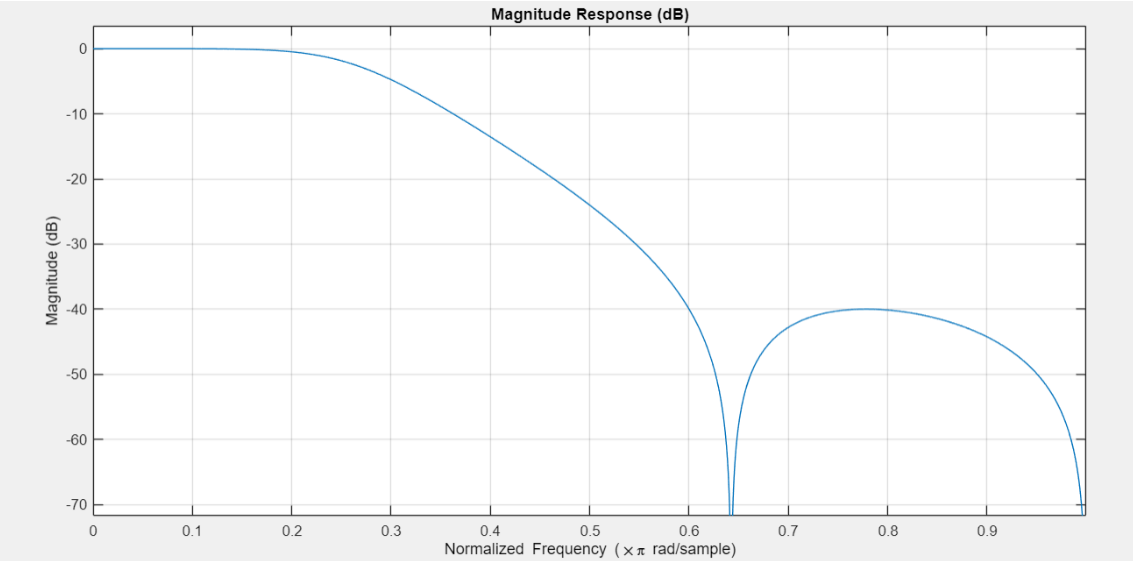
$$H_{LP}(z) = H_a(s)\big|_{s=\frac{(z-1)}{(z+1)}}$$

$$H_{\mathrm{LP}}(z) = \frac{0.011474z^3 + 0.034421z^2 + 0.034421z + 0.011474}{z^3 - 2.178z^2 + 1.7698z + 0.53976}$$

CODIGO MATLAB

```
% Especificaciones del filtro
fp = 1e3;     % Frecuencia de corte en la banda de paso (1 kHz)
fs = 3e3;     % Frecuencia de corte en la banda de parada (3 kHz)
Fs = 10e3;    % Frecuencia de muestreo (10 kHz)
Rp = 1;       % Ripple máximo en la banda de paso (1 dB)
Rs = 40;      % Atenuación mínima en la banda de parada (40 dB)

% Normalizar las frecuencias
W_p = fp / (Fs / 2);  % Frecuencia de banda de paso normalizada
W_s = fs / (Fs / 2);  % Frecuencia de banda de parada normalizada

% Calcular el orden del filtro y la frecuencia de corte
[N, Wn] = cheb2ord(W_p, W_s, Rp, Rs); % Orden y frecuencia de corte

% Diseñar el filtro Chebyshev Tipo 2
[b, a] = cheby2(N, Rs, Wn, 'low')

% Visualizar la respuesta en frecuencia
fvtool(b, a); % Mostrar la respuesta de frecuencia del filtro
```

GRAFICA



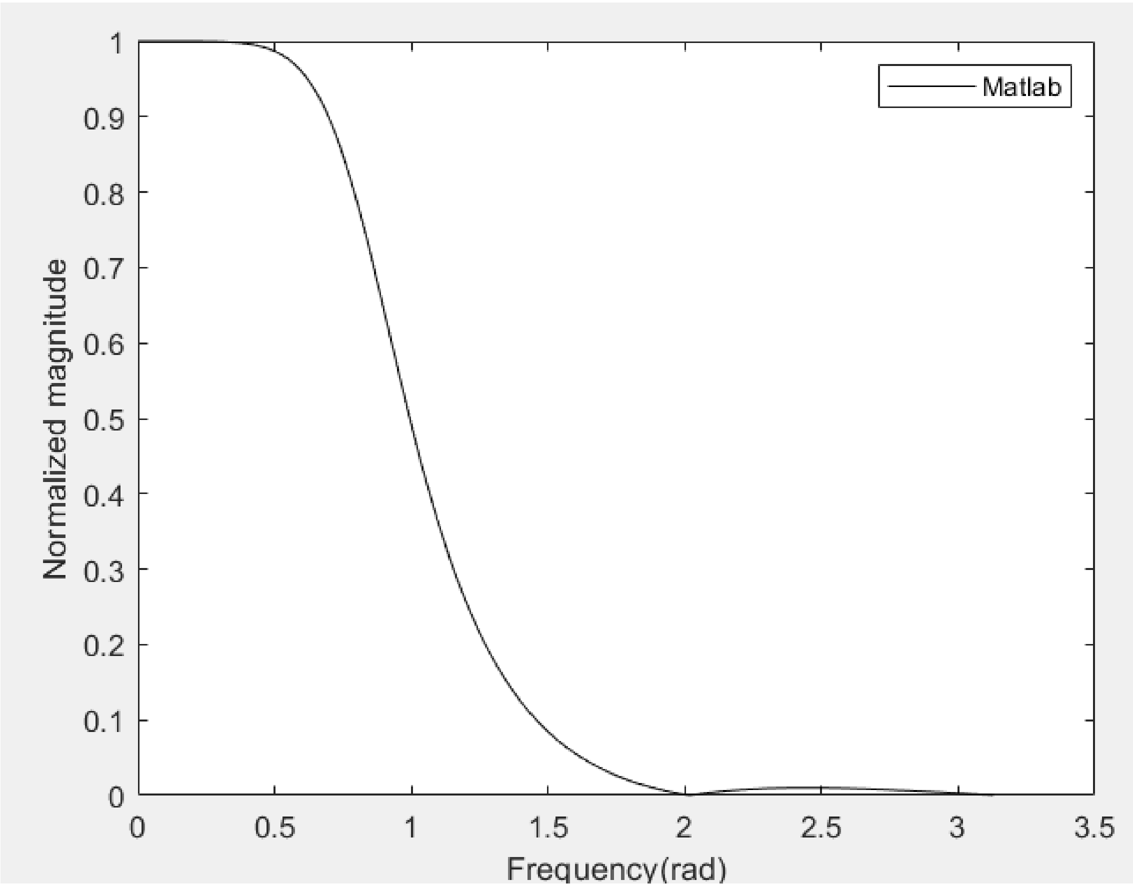
OTRA SOLUCIÓN

```
% Solution quiz  unit III

% Design of an IIR lowpass digital filter


% Especificaciones del filtro

fp = 1e3;     % Frecuencia de corte en la banda de paso (1 kHz)

fs = 3e3;     % Frecuencia de corte en la banda de parada (3 kHz)

Fs = 10e3;    % Frecuencia de muestreo (10 kHz)

Rp = 1;       % Ripple máximo en la banda de paso (1 dB)

Rs = 40;      % Atenuación mínima en la banda de parada (40 dB)


% Normalizar las frecuencias

W_p = fp / (Fs / 2);  % Frecuencia de banda de paso normalizada

W_s = fs / (Fs / 2);  % Frecuencia de banda de parada normalizada
```

```
% Calcular el orden del filtro y la frecuencia de corte

[N, Wn] = cheb2ord(W_p, W_s, Rp, Rs); % Orden y frecuencia de corte


% of the type Chebyshev 2 using

% bilinear transformation

%clear

% the arguments for cheby2 are: filter order, minimum

% stopband attenuation in dB, stopband edge, and 's' for analog

% filter

% find the zeros & poles of a lowpass 3rd-order Chebyshev 2

% analog filter

%N = 3;% filter order

A = 40; % minimum stopband attenuation in dB

omega_s = W_s*pi % stopband edge of digital filter

ws = tan(omega_s/2);% warped staopband edge of the analog filter

%ws = W_s;

[z,p,k] = cheby2(N,A,ws,'s');

% apply bilinear transform to the analog filter

% use sampling frequency of 1/2 to normalize the scale factor 2/T

[zd,pd,kd] = bilinear(z,p,k,0.5);

% convert zero-pole to transfer function H(z) = Nd(z)/Dd(z)

[Nd,Dd] = zp2tf(zd,pd,kd);

% compute the frequency response

[Hz,Wz] = freqz(Nd,Dd,256);

% results from analysis

N2 = 2*N;

A1 = 10^(A/20);

% compute the zeros
```

```
Za = [0.0 + 1.1547i 0.0 - 1.1547i];
gamma = (A1+sqrt(A1*A1 - 1))^(1.0/N);
e1 = (gamma*gamma-1)/(2*gamma);
e2 = (gamma*gamma+1)/(2*gamma);
% compute the poles
for n = 1:N
  alfa(n) = -e1*sin((2*n-1)*pi/N2);
  beta(n) = -e2*cos((2*n-1)*pi/N2);
  Pa(n) = alfa(n)/(alfa(n)^2 + beta(n)^2)...
    + (beta(n)/(alfa(n)^2 + beta(n)^2))*1i;
end
[Zd,Pd,Kd] = bilinear(Za',Pa',1.0,0.5);
[Nz,Dz] = zp2tf(Zd,Pd,Kd);
H = freqz(Nz,Dz,Wz);
% Plot the magnitude of the frequency response
figure,plot(Wz,abs(Hz)/max(abs(Hz)),'k')
%hold on
%plot(Wz,abs(H)/max(abs(H)),'r--')
%hold off
ylabel('Normalized magnitude')
xlabel('Frequency(rad)')
%legend('Matlab','Analysis')
legend('Matlab')
% calculate the actual minimum stopband attenuation
indx = find(Wz == omega_s);
sprintf('Min. stopband atten. dB is %5.3f\n',...
  20*log10(abs(H(indx))/max(abs(H))))
```

GRAFICA



Matlab
Normalized magnitude
Frequency(rad)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.5
1.5
2
2.5
3
3.5