EE387 - LAB 04

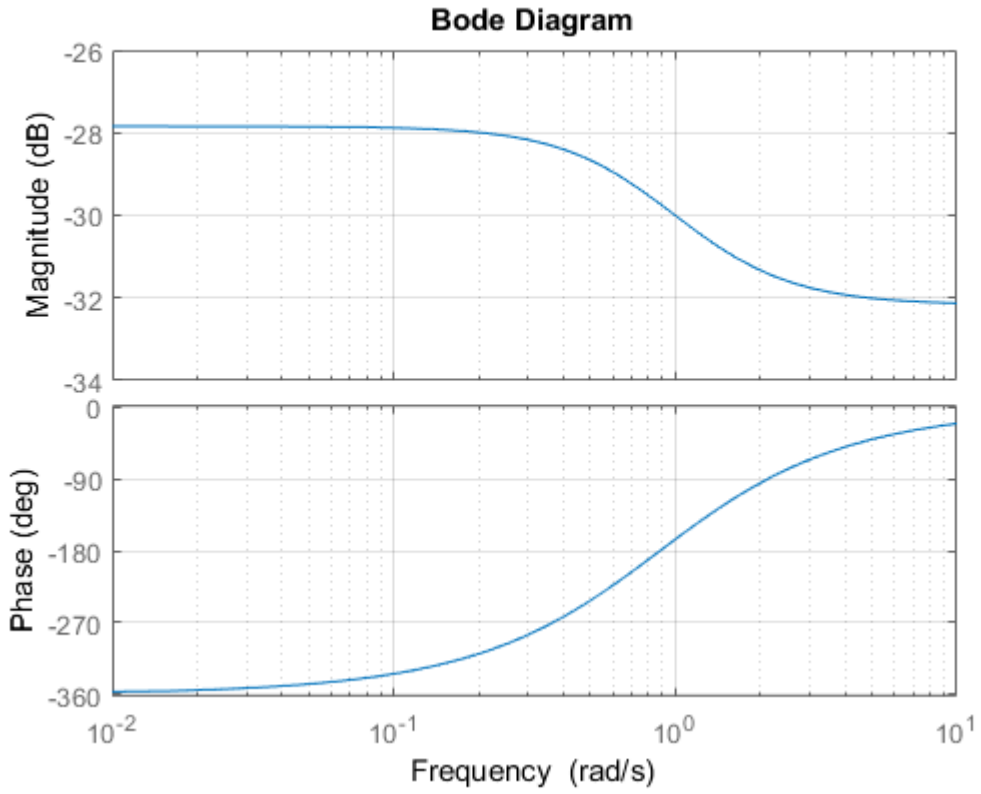
FILTER DESIGN USING MATLAB

DE SILVA K.G.P.M.
E/15/065
SEMESTER 06
06/06/2020

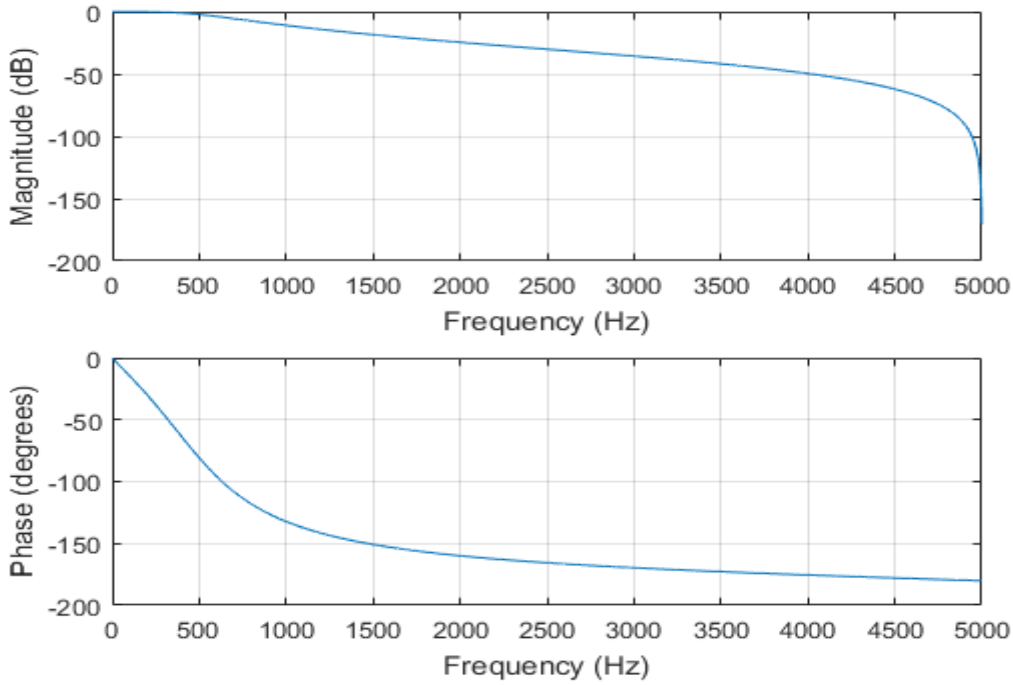
Exercises:

1) Design the Butterworth filter with the following specifications: Fp = 1000 Hz; Fs = 5000 Hz;

```
clear all;close all;

%frequencies
Fp = 1000;
Fs = 5000;

%number of samples
samples = 10000;

%find omega values
Wp = Fp / samples;
Ws = Fs / samples;

%maximum passband attenuation and minimum passband attenuation
Rp=3;Rs=30;

[N,Wn] = buttord(Wp,Ws,Rp,Rs);% find Butterworth filter order and cutoff
frequency
[z,p,k]=butter(N,Wn); % find zeros,ploes and scale
[num,den]=butter(N,Wn);% find numerator and denominator

%plot the bode plot
tf=zpk(z,p,k);
bode(tf);
grid on

%plot frequency response of digital filte
figure
freqz(num,den,5000,samples)
```

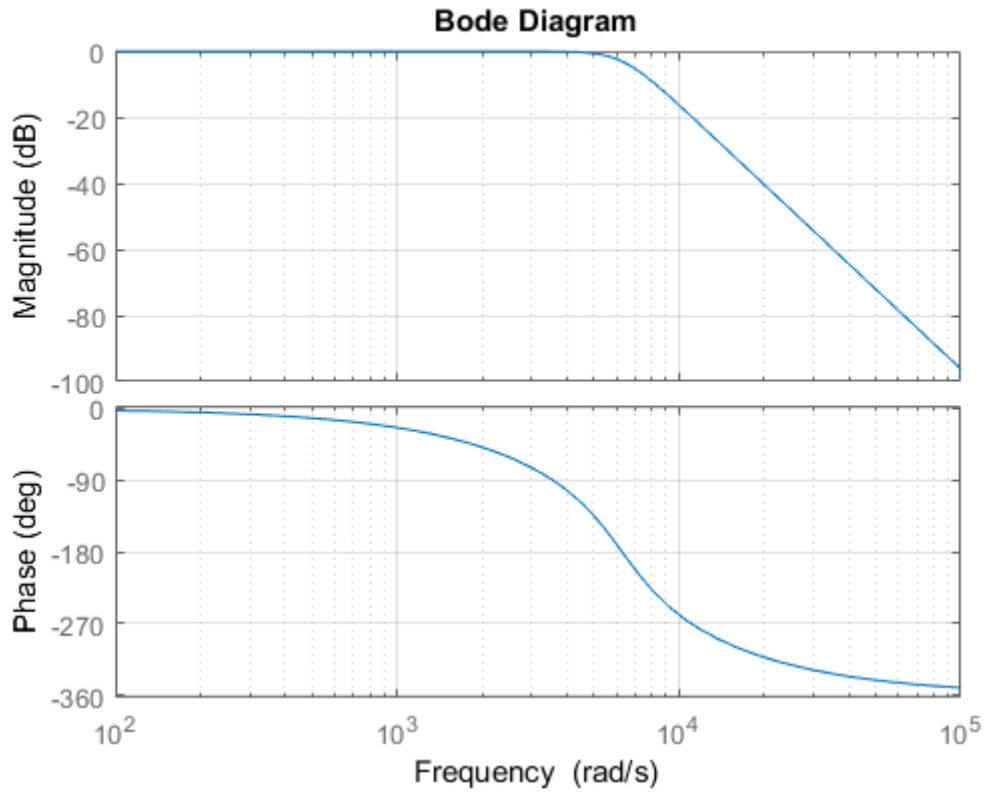


Frequency response of the filter



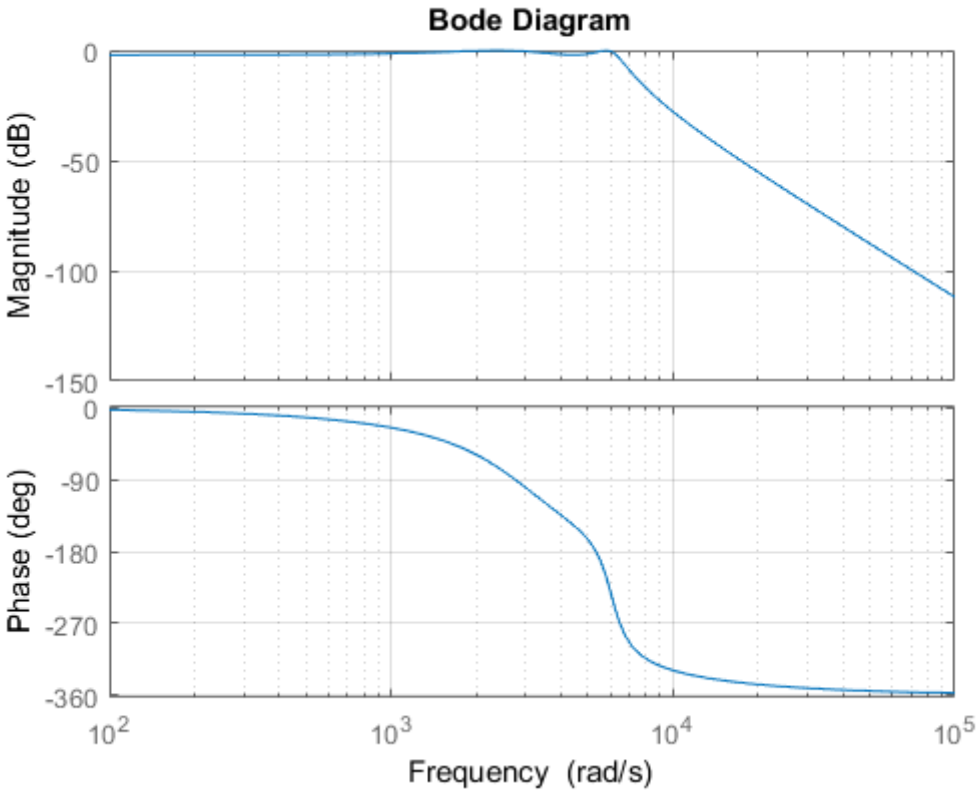
2) Design the Butterworth filter with Fp = 1000 Hz, N = 4;

```
clear all;close all;

N=4;
Fp = 1000;

Wp = 2*pi*Fp;% find omega

[num,den]=butter(N,Wp,'s');% find numerator and denominator
H=tf(num,den); %find transfer function

%plot the bode plot
figure;
bode(H)
grid on
```



3) Design Chebyshev Type 1 filter with N = 4, Rp = 2 ; Fp = 1000.

```
clear all;close all;

N = 4;
Rp = 2;
Fp = 1000;

Wp = 2*pi*Fp; % find omega

[num,den] = cheby1(N,Rp,Wp,'s'); % find numerator and denominator

H= tf(num,den);%find transfer function

%plot the bode plot
figure;
bode(H);
grid on
```



Discussion

In signal processing, a filter is a device or process that removes some unwanted components or features from a signal. Filtering is a class of signal processing, the defining feature of filters being the complete or partial suppression of some aspect of the signal. Most often, this means removing some frequencies or frequency bands. However, filters do not exclusively act in the frequency domain, especially in the field of image processing many other targets for filtering exist. Correlations can be removed for certain frequency components and not for others without having to act in the frequency domain. Filters are widely used in electronics and telecommunication, in radio, television, audio recording, radar, control systems, music synthesis, image processing, and computer graphics.

Filter is mainly classified into two types. They are active and passive filters. Filter Circuit which consists of active components like Transistors and Op-amps in addition to Resistors and Capacitors is called as Active Filter. Filter circuit which consists of passive components such as Resistors, Capacitors and Inductors is called as Passive Filter. The operating frequency range of the filter banks on the components used to build the circuit. Hence the filter can be further categorized based on the operating frequency of a particular circuit. They are low pass filters, high pass filters, band pass filters, band stop filter, all pass filters.

Filter design is the process of designing a signal processing filter that satisfies a set of requirements, some of which are contradictory. The purpose is to find a realization of the filter that meets each of the requirements to a sufficient degree to make it useful.

In this lab we designed Butterworth filters and a Chebyshev filter. The Butterworth filter is a popular form of filter providing a maximally flat in-band response. Whilst the most common method of calculating the values these days is to use an app or other computer software, it is still possible calculate them using more traditional methods. There are formulas or equations that can be sued for these calculations and in this way it is possible to understand the trade-offs and workings more easily. In matlab there are in built function to design a Butterwoth filter. Below is how those functions can be used to design the filter.

[b,a] = butter(n,Wn) returns the transfer function coefficients of an nth-order lowpass digital Butterworth filter with normalized cutoff frequency Wn.

[b,a] = butter(n,Wn,ftype) designs a lowpass, highpass, bandpass, or bandstop Butterworth filter, depending on the value of ftype and the number of elements of Wn. The resulting bandpass and bandstop designs are of order 2n.

[z,p,k] = butter(___) designs a lowpass, highpass, bandpass, or bandstop digital Butterworth filter and returns its zeros, poles, and gain. This syntax can include any of the input arguments in previous syntaxes.

[A,B,C,D] = butter(___) designs a lowpass, highpass, bandpass, or bandstop digital Butterworth filter and returns the matrices that specify its state-space representation

Chebyshev filters are used to separate one band of frequencies from another. Although they cannot match the performance of the windowed-sinc filter, they are more than adequate for many applications. The primary attribute of Chebyshev filters is their speed, typically more than an order of magnitude faster than the windowed-sinc. This is because they are carried out by recursion rather than convolution. As below we can use matlab to design the Chebyshev type1 filters.

[b,a] = cheby1(n,Rp,Wp) returns the transfer function coefficients of an nth-order lowpass digital Chebyshev Type I filter with normalized passband edge frequency Wp and Rp decibels of peak-to-peak passband ripple.

[b,a] = cheby1(n,Rp,Wp,ftype) designs a lowpass, highpass, bandpass, or bandstop Chebyshev Type I filter, depending on the value of ftype and the number of elements of Wp. The resulting bandpass and bandstop designs are of order 2n.

[z,p,k] = cheby1(___) designs a lowpass, highpass, bandpass, or bandstop digital Chebyshev Type I filter and returns its zeros, poles, and gain. This syntax can include any of the input arguments in previous syntaxes.

[A,B,C,D] = cheby1(___) designs a lowpass, highpass, bandpass, or bandstop digital Chebyshev Type I filter and returns the matrices that specify its state-space representation.

[___] = cheby1(___,'s') designs a lowpass, highpass, bandpass, or bandstop analog Chebyshev Type I filter with passband edge angular frequency Wp and Rp decibels of passband ripple