Report

YouTube and Classroom Instruction:

The most prominent difference is that classroom discussion gives the ability to ask questions and have quick, adaptive guidance. With respect to a video this would be searching for the answers to any prompted questions on your own. Whether in another video, textbooks, or informational websites. Further, video in its nature has a set structure. The creative process of producing an informational video is inclined to having a set and thought-out lesson. This lets the learner take it in at their own pace, but it is a double-edged sword because one might not take it in at all. In comparison, in person instruction can frequently deviate from its lesson plan, especially when given to a group. Either enhancing learning or impairing it if the necessary material is not absorbed.

Talking through my Process:

First, I made note of the given parameters and objectives detailed in the email. Next, I needed to define the variables for producing the order and cut-off frequency of the Butterworth filter, using the ‘buttord( )’ function. I set the pass band frequency as 3 Hz, **fp**, and the range of frequencies to be sampled as 100 Hz, **f**. I assumed the sampling rate, **fs**, to be 1 Hz, with the thought process that this would produce an adequate spread of data. I then used ‘**f**’ to convert the pass band frequency and sampling rate to rad/s, **Wp** and **Ws**. Finally, the function, ‘buttord( )’, needed pass band ripple, **Rp**, and stop band attenuation, **Rs**, to be given. Since these were not mentioned, I found that the typical attenuation is around 20 dB and that ripple can be set to nearly 0 dB. This part I am least confident about, since 0 itself produced an error. I chose 1e-6 because the email mentioned 1 micro-Hz as the start of the range. With all of this prepared I was able to produce the filter order, **N**, and cut-off frequency, **wn**. Which I plugged in the ‘butter( )’ function that was set to low pass. This then produced the transfer function coefficients that I plugged into the ‘freqz( )’ function on the time scale 0 to π. The ‘freqz( )’ function produced the gain, **h**, and angular frequency, **ohm**, vectors. I normalized the angular frequency by dividing by pi, and the gain by taking the absolute value. Finally, I plotted the results with normalized gain and converted it to dB. As well as plotted the phase by using the ‘angle( )’ function on the gain.

Results:

Engineering drawing

Description automatically generated with low confidence

Code:

%digital low pass butterworth filter

%parameters: ->3 Hz Bandwidth

% ->range 1 micro-Hz to 100 Hz

%Objectives: ->plot amplitude and phase as function of range

% ->locate frequencies where gain goes to zero

% ->Calculate peak gain at all side lobes past the 3 Hz

% Bandwidth

clc

clear all

close all

%find order of filter and then TF

%Limit:

Rp=1e-6; % pass band position

Rs=20;% stop band attenuation

fp=3; %pass band frequency in Hz

fs=1; % sampling rate (Hz)

f=100; %sampling frequency

Wp=(2\*pi\*fp)/f;%rad/s fp

Ws= (2\*pi\*fs)/f; %rad/s fs

[N,wn] = buttord(Wp,Ws,Rp,Rs); %find order of filter

[B,A]=butter(N,wn,'low');

t=0:0.01:pi;

[h ohm]=freqz(B,A,t);

subplot(3,1,1)

plot(ohm/pi,abs(h))

grid on;

xlabel('normalized frequency')

ylabel('gain')

title('frequency response')

subplot(3,1,2)

plot(ohm/pi,20\*log(abs(h)))

xlabel('normalized frequency')

ylabel('gain in dB')

title('frequency response in dB')

subplot(3,1,3)

plot(ohm/pi,angle(h))

xlabel('normalized frequency')

ylabel('phase')

title('phase response')