Filter Design

Learning Objective: In this lab session we will learn about analog filters, and related Matlab tools useful for this purpose. We will consider Butterworth low pass filters in this session.

Experiments: Our final goal is to design an analog Butterworth low pass filter with a given cut-off frequency. In the process we will also learn about ways to use Matlab, not just numerical computation, but for symbolic Math manipulations. We will accomplish this in the following steps.

1. Input signal: In the last lab, we used *fft()* to determine the spectrum of signals. FFT, a way to implement DFT, is an algorithm to analyze the spectrum of periodic discrete signals. In this first part, we will use the symbolic Math toolbox of Matlab to determine the Fourier transform of sinusoidal signals:
   (a) Using syms function of Matlab define two variables, *t* and *w*.
   (b) Define a time domain signal *x*, which is a function of time variable *t*, as a sum of three cosine signals, each having different frequencies and amplitudes.
   (c) Determine and output the Fourier transform of the time domain signal *x* using the symbolic Math toolbox function *fourier(x,t,w)*. Notice that the *fourier()* function provides the Fourier transform of the signal in functional form, which you have to output.
2. Filter parameters: Use the *[b, a] = butter(N, Whp, 's')* function of Matlab to determine the transfer function coefficients of a low pass Butterworth filter with appropriate values of the arguments. Once you have obtained the filter, use *freqs(b,a,Wband*) function of Matlab to obtain the theoretical magnitude response of the filter. Choose the value of the argument *Wband* appropriately so that you can see the pass band, transition band and stop band of your filter properly.
3. Filter transfer function: Use the Matlab statement

   H = poly2sym(b,w)/poly2sym(a,w)

   to obtain the transfer function of your filter.
4. Output signal: You can recall from your 'Signals and Systems' or 'Principles of Communication' courses that the output of a linear time invariant system in time and frequency domain is obtained by

   $$y(t) = h(t) * x(t) \quad \text{OR} \quad Y(w) = H(w)\, X(w)$$

   Use the frequency domain formula to obtain the output signal in frequency domain. Convert the output so obtained in time domain using the Matlab function *y = ifourier(Y,w,t)*. Notice that *y* is in the symbolic math form.
5. Visualizing the input and output in numeric form: Visualize the input and output signals. You may use the following piece of code to achieve this.

   ```
   fs = ??; % choose an appropriate value for the sampling frequency
   t1 = 0:1/fs:??; % choose an appropriate length of time to visualize the signal
   x_numeric = double(subs(x,t,t1));
   y_numeric = double(subs(y,t,t1));
   figure; subplot(2,1,1); plot(t1,x_numeric); subplot(2,1,2); plot(t1,y_numeric);
   ```

   Following this, obtain the spectrum of the signals *x_numeric* and *y_numeric* using the techniques you learnt in the previous software lab session. Note that you should choose the frequencies of the sinusoids in the input signal in (1) and cut-off frequency of the filter in (2) appropriately to see how filter affects the input signal.

Deliverables: You are allowed to discuss your work with others but final code(s) and report should be your own individual work. **You should submit a soft-copy report by 27th January 2021 to Mr. Anurag Bohre**, anuragb@iiserb.ac.in (in the Google form set up by him), which includes:

1. The Matlab codes. You need not repeat the codes if you are only changing the parameters to produce a different set of graphs.
2. The output produced by your code: graphs, numerical output etc.

Sources of Help: “Signals and Systems Using Matlab” by Luis F. Chaparro.