EE 324 LAB 7
Analog filter design

In this lab, you will learn how to design analog filters. You will first design low pass prototypes, and then use transformations to derive low pass, high pass and bandpass filters for given specifications.

Prelab:
1. Given that a Butterworth filter of order $N$ is defined by: $|H(j\omega)|^2 = \frac{1}{1+(\frac{\omega}{\omega_c})^{2N}}$ derive the transfer function $H_0(s)$ (in Laplace representation) of a low pass prototype (i.e. $\omega_c = 1$) of order $N = 4$ analytically. Write down and explain all the steps.

2. Use the transformations $s \rightarrow \frac{s}{\omega_l}$, $s \rightarrow \frac{\omega_h}{s}$, and $s \rightarrow \frac{s^2+\omega_0}{Bs}$ to derive a low pass filter $H_1(s)$ with cutoff frequency $f_l = 100$ Hz, a high pass filter $H_2(s)$ with cutoff frequency $f_h = 100$ Hz, and a bandpass filter $H_3(s)$ with central frequency $f_0 = 100$ Hz and bandwidth $B = 40$ Hz, respectively.

Laboratory Assignment:
Problem 1:
1. Verify your results in MATLAB using the function butter. Note that in order to design an analog filter you have to use a syntax of the form $[z, p, k]=\text{butter}(N, \omega_n, 'filtertype', 's')$. 

Figure 1 Filter Specifications
2. Represent the Bode plots for all three filters.

3. Using MATLAB’s function `buttord` (syntax of the form 
   \[ [N, w_n] = \text{buttord}(w_p, w_s, R_p, R_s, 's') \]), to determine the minimum order and the 
cutoff frequencies, followed by `butter` command, design a Butterworth bandpass filter 
with the specifications of Figure 1, namely: \( f_p = [80 \text{ Hz}, 120 \text{ Hz}] \), \( f_s = [60 \text{ Hz}, 140 \text{ Hz}] \), \( R_p = 2 \text{ dB} \), \( R_s = 60 \text{ dB} \).

4. Compute the Bode plot for your filter, and verify the specifications are met.

5. Using the MATLAB functions `cheb1ord` and `cheb1`, design a bandpass Chebyshev 
filter with the specifications in Figure 1.

6. Compute the Bode plot for your filter, and verify the specifications are met.

7. Comment on the differences between the two filters.

Problem 2:
1. Write a MATLAB program which records five seconds of your voice (mono), and stores 
it into a one-dimensional vector \( x \). You may use instructions like \( y = \text{audiorecorder} \), 
\( \text{record}(y, 5) \), and \( x = \text{getaudiodata}(y) \).

2. Look at the frequency characteristics of \( x \) (you may use the MATLAB function `fft` to 
compute a fast Fourier transform).

3. In a Simulink model, use the vector \( x \) as a source input to one of the bandpass filters 
designed in this lab.

4. Store the output in a vector \( y \) and play it using the MATLAB function `sound`. 