In this assignment, you will reinforce what we did in lecture today regarding MATLAB's filter toolbox.

1. Generate a signal that consists of a sum of sine waves of frequencies 1 to 50 kHz. Set t to be from 0 to 2 seconds, using an interval of 0.001s.

$$signal = \sum_{f=1}^{50000} \sin(2\pi f t)$$

For each of the following questions, you will create a filter, create magnitude-phase plots for the filter and apply the filter to a signal. Follow these steps:

- Design the filter using ***filterDesigner***. The issue now is that when I run your submissions I don't want to have to design the filters myself obviously! Therefore, *generate MATLAB code from filterDesigner for each filter and include the four MATLAB function files in your submission.* Obviously give the functions different names. Remember that these functions return filter objects, not numerator and denominator coefficients but you should be fine if you follow the below steps correctly. Also, very important, make sure to design the filters in a single section, not in second-order sections as is the default!
- Create a filter object by calling the generated function.
- Use the DSP toolbox's version of ***freqz*** on the filter object. Make sure to include the sampling frequency in the function call as this is hardly mentioned in the documentation. For example, if $filter$ is a filter object, n is the number of points (you can use 1024) and fs is the sampling frequency, run $[H, f] = freqz(filter, n, fs)$. Note I use f instead of w since by including the sampling frequency, MATLAB scales the frequencies from $[0, \pi]$ to $[0, fs/2]$. Hence these frequencies have units of Hertz. Keep that in mind when including units in your plots and setting the axis limits.
- Create magnitude-phase plots akin to homework 6 except for the difference mentioned above regarding f.
- Apply the filter to the signal using ***filter*** (check the documentation for the DSP toolbox version).
- Lastly, plot the Fourier Transform of the final result using ***fft*** and ***plot***. Refer to the notes for the proper way to use ***fft*** and obtain the proper scaling.

This may seem daunting, but with properly defined functions, you may only have to do most of the work once. However, I still want unique titles for plots (maybe pass in a string?).

2. Create a Butterworth bandpass filter with a sampling frequency of Fs = 100 kHz, a stopband frequency of below the frequency Fstop1 = 25 kHz and above Fstop2 = 45 kHz, a passband frequency of between Fpass1 = 30 kHz Fpass2 = 40 kHz, a passband attenuation of Apass = 5dB, and a stopband attenuation of Astop = 50dB.

3. Create a Chebychev I bandstop filter with a sampling frequency of Fs = 100 kHz, a passband frequency of below the frequency Fpass1 = 5 kHz and above Fpass2 = 40 kHz, a stopband frequency of between Fstop1 = 15 kHz Fstop2 = 35kHz, a passband attenuation of Apass = 5dB, and a stopband attenuation of Astop = 50dB.

4. Create a Chebychev II highpass filter with a sampling frequency of Fs = 100 kHz, a passband frequency of Fpass = 45 kHz, a stopband frequency of Fstop = 25 kHz, a passband attenuation of Apass = 2dB, and a stopband attenuation of Astop = 40dB.

5. Create a Elliptic lowpass filter with a sampling frequency of Fs = 100 kHz, a passband frequency of Fpass = 20 kHz, a stopband frequency of Fstop = 30 kHz, a passband attenuation of Apass = 5dB, and a stopband attenuation of Astop = 50dB.