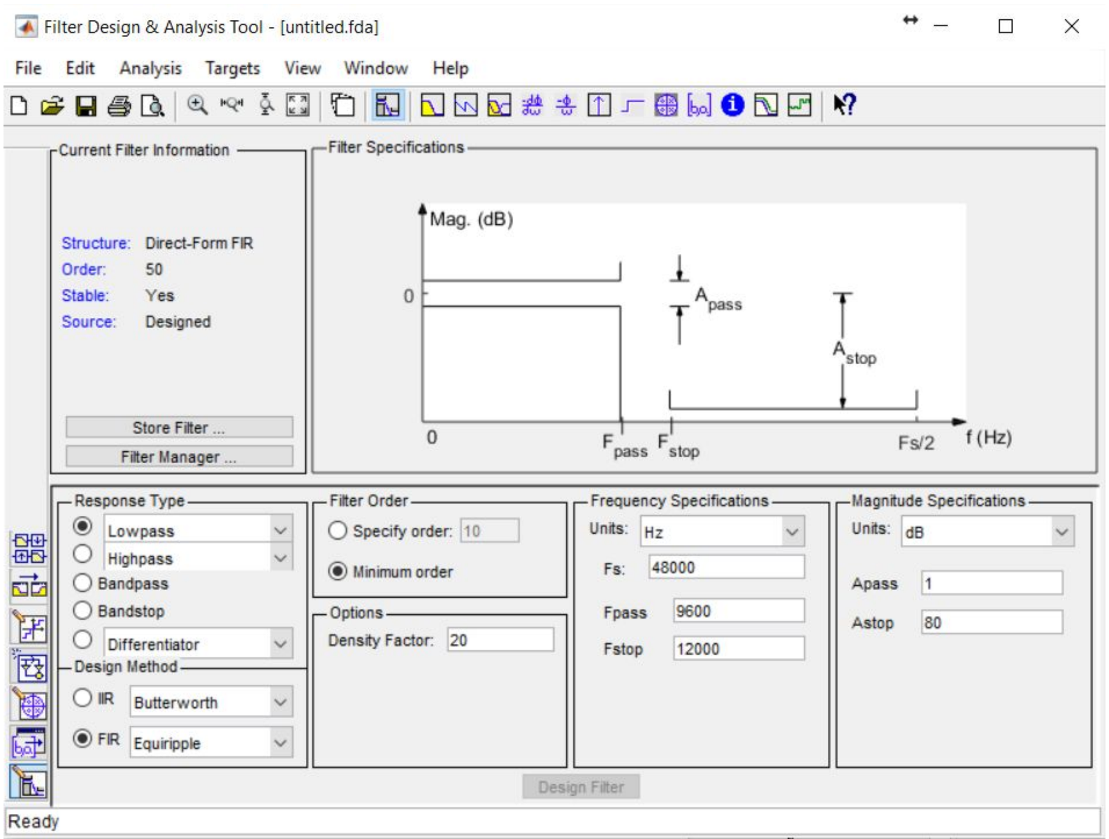
Problem Set May 23: Filtering and amplitude
Fundamentals of Statistics and Computation for Neuroscientists
Instructor: Scott Cole

1. <u>Filter Design & Analysis</u>: This problem investigates the properties of filters from the perspective of filter design. There is no single correct way to filter your data, but there are definitely bad ways to filter data that can lead to misleading results. It is just important that you are familiar with the properties of the filter. We are mainly concerned in the frequency response of the filter: how each frequency across the spectrum is attenuated. However, familiarity with the filter kernel (time-domain filter that is convolved with the signal) is also important. Namely, the length of this kernel will determine the extent of the edge artifacts in the filter.

   For this problem, you will be designing multiple filters. Use your favorite screenshot tool to capture the plot of the filter and paste it into your problem set document to turn in.
   <u>When capturing plots, zoom in so that you can see the most relevant features of the plot. Turn in screen captures of plots whenever the instructions say to plot something.</u>
   To launch the Filter Design & Analysis Tool in MATLAB, open up MATLAB and type 'fdatool' into the command window. A new window will appear that looks like this:



We will fill out the form in the bottom half of this screen in order to design our filter. Before starting, you should visit this website to review what each section of this form controls: http://www.mathworks.com/help/signal/ug/opening-fdatool.html . If you have any questions about what a parameter is, refer to this guide from MathWorks.

a. First, design a Lowpass filter. Under *Response Type* choose 'Lowpass.' Under *Design Method* choose 'FIR' and 'Window'. Under *Filter Order* select 'Specify order' and set it to 500. Under *Options*, under *Window*, select 'Hamming'. Keep 'Scale passband' checked. Under *Frequency Specifications*, make sure the units is Hz, and change F_s (the sampling rate) to 1000. Change Fc (the cutoff frequency) to 10. Then click the 'Design Filter' button. Go to File>Generate MATLAB code>Filter design function and save it somewhere to be used in Problem 2.
   i. Zoom into the 'Magnitude Response' plot that appears between 0 and 50 Hz. Screen-capture this plot to turn in. Use the zooming functionality in fdatool.
   ii. The transition width is the frequency range between the passband (low frequencies) and stopband (high frequencies). Let us define the transition width here as the frequency range with magnitude between -3dB and -60dB. What is the approximate transition width (in Hz) for this filter?
   iii. Change the filter length (order) to 200 samples. How long is this in seconds?
   iv. Again, plot the magnitude response of this filter. What is the approximate transition width for this filter?
   v. Try changing the filter order to a few more values (e.g. 100, 1000) How does changing the filter order affect the filter's transition width?
   vi. For the filter of order=200 designed above, visualize the filter kernel (a.k.a. Impulse response) by clicking the button with the upward arrow in the top toolbar. Plot the kernel. Examine the kernel and think intuitively how convolving this kernel with a signal would have the effect of a low-pass filter (no need to write anything for this part).

b. The magnitude plot you saw above was plotted in units of decibels (dB) as a function of frequency. To calculate dB:

$$dB = 20log_{10}(\frac{Magnitude_{filtered}}{Magnitude_{original}})$$

   i. What is the magnitude (in dB) if the filter does not change the magnitude of a given frequency component?
   ii. What is the magnitude (in dB) if the filter attenuates a given frequency component to 1/100 of its original magnitude?
   iii. In the first filter we designed with an order=500, by zooming in on the magnitude response plot, what is the magnitude in dB at 10Hz, 12Hz, and

20Hz? What is the relative magnitude of the filtered signal relative to the original signal ($\frac{Magnitude_{filtered}}{Magnitude_{original}}$) at these frequencies?

c. Design a band-pass filter to extract a theta rhythm (4-8Hz) from a neural signal. Again use the FIR window design method and keep F_s = 1000 Hz. Set your filter order to 1000.
    i. Plot the frequency response (zoom such that you can see the important dynamics of the filter. For example, x axis limits of 0-20Hz would work).
    ii. What is $\frac{Magnitude_{filtered}}{Magnitude_{original}}$ at 7Hz, 8Hz, and 9Hz?
    iii. Plot the filter kernel. Explain intuitively why convolving this kernel with a neural signal would extract the theta rhythm.
    iv. Change the filter order to 500. Plot the magnitude response. Which filter (order 500 or 1000) would you prefer if you wanted to only extract the theta rhythm from your neural signal?
    v. What is the length of the edge artifact for the order 500 and order 1000 filters? (i.e. how many samples would need to be removed from the start and end of the output of the filter?)
    vi. Design a 4th-order infinite-impulse response (IIR) butterworth filter that is also a bandpass filter with cutoff frequencies 4Hz and 8Hz. Plot the magnitude response. How does the attenuation at high frequencies differ compared to the 1000-order FIR filter. Which would you prefer if you wanted to make sure to isolate a theta rhythm and remove a strong alpha rhythm (10-13 Hz).

d. Design a bandstop filter again using the FIR window design method and F_s = 1000Hz with cutoff frequencies 58Hz and 62Hz. Use a filter order of 100.
    i. Plot the frequency response. How much is this notch filter suppressing 60Hz noise compared to the suppression seen in the other FIT filters?
    ii. Increase the filter order to 1000. Plot the filter kernel. Zoom in to see the fluctuations in the kernel around the center sample. Think about how convolution with this kernel would suppress 60Hz line noise (no need to write anything down).

e. Design a high-pass filter again using the FIR window design method and F_s = 1000Hz with cutoff frequency 4Hz. Use a filter order of 1000.
    i. Plot the frequency response of the filter.
    ii. Plot the filter kernel. Zoom in to see the fluctuations in the kernel around the center sample.

2. Low-pass filter
    a. Open up a blank MATLAB script in the Editor. Generate 2 seconds of white noise sampled at 1kHz. (HINT: use `randn`)
    b. Copy the low-pass filter code (from `Fs = 1000…` to `b = fir1( …`) to your script. The end result, `b`, is the filter kernel.

c. Filter the white noise (HINT: use `filtfilt` and your input for variable `a` should be set to `1`). Plot the filtered signal. Label the x-axis with units. Notice the edge artifacts.

3. Calculate the amplitude envelope of a signal. Load data from `lect9hw1N3.mat`. These are neural recordings from the primary motor cortex. There is a data array in which each row is the time course of voltage over one trial (the columns correspond to the time array in the same file). Before doing anything, plot the time series of each trial to get an idea of what the data look like (do not turn this in).
   a. Calculate and plot the continuous wavelet transform of the signal for the first trial. Use the script included, `WT_wu.m`. Use frequency limits 5Hz and 50Hz.
   b. At time 0, the subject starts moving. For each trial, quantify the mean beta oscillation (13-30Hz) amplitude in the time range -500ms to -250ms by averaging over these ranges. (HINT: Take the absolute value of the wavelet transform in order to get amplitude.) Print the mean beta oscillation amplitude in this period for each trial (1 number per trial).
   c. Repeat the process in (b) for a time period after movement onset, time 250ms to 500ms. Print the mean beta oscillation for this period for each trial.
   d. Assume the amplitude values are normally distributed both before and after movement onset. What test should you use to evaluate if there is a significant difference between these two sets of beta amplitude? Do this test and interpret the result.