Filters

1. Simple FIR Digital Filter Design
   (10 points) In a single m-file script, do the following:
   
   (a) Use the Matlab function `fir1` to design a 10th order FIR lowpass filter that cuts off at one-fourth
       the sampling rate.
   
   (b) Use `freqz` to display the amplitude and phase response of this filter (do a >> `help freqz` for
       more information).
   
   (c) Repeat, but this time increasing the order to 50. What is the difference in the amplitude and
       phase responses? Include your (short) answer as a comment in the script.
   
   (d) Apply the FIR filter to 1000 samples of a white noise signal (generated using `randn`). With
       your sound volume TURNED WAY DOWN (at first), listen to the input and output signals.

2. Linear and Zero Phase Filters

   When one wishes to modify only the amplitude of a signal and not the phase, a linear-phase filter
   is desirable. Linear phase filters have a symmetric impulse response, i.e.
   
   \[ h(n) = h(N - 1 - n), \]
   
   for an impulse response of length N. A symmetric impulse response corresponds to a real frequency
   response times a linear-phase term \( e^{-j\alpha \omega T} \), where \( \alpha \) is the slope of the linear phase. Linear phase
   filters are useful because they delay all frequency components by the same amount, thus preserving
   the waveshape as much as possible for a given amplitude response.

   (a) (5 points) It was shown in class that the phase response \( \Theta(\omega) = -\omega T/2 \) of our simple first-order
       lowpass filter \( (y(n) = x(n) + x(n - 1)) \) corresponds to a waveform delay of one-half sample \( (T/2 \)
       seconds), at all frequencies. What is the phase response of the other 3 simple filters discussed
       in class? Are they linear?

   (b) (5 points) Zero-phase filters are a special case of linear-phase filters in which the phase slope
       is \( \alpha = 0 \). The real impulse response \( h(n) \) of a zero-phase filter is even. That is, it is symmetric
       about time 0 and therefore satisfies
   
   \[ h(n) = h(-n). \]

   Note, this implies that zero phase filters are not causal!

   Due to Fourier symmetry, real, even signals have real, even Fourier transforms.

   i. If the spectrum is real (and not complex), what are the possible values of the phase (Hint: 
      there are two—think of the complex plane)?
   
   ii. Implement the following lowpass filter using Matlab’s `firpm` function (you will need the
       signal processing toolbox for this). Call a `help firpm` to get more details on the functions.

   \[
   N = 11; \quad \% \text{filter order}
   
   b = [0 0.1 0.2 0.5]*2; \quad \% \text{specify band edges}
   
   M = [1 1 0 0]; \quad \% \text{specify amplitudes}
   
   h = firpm(N-1,b,M); \quad \% \text{impulse response (or ‘B’ coefficients)}
   \]
Notice the returned impulse response is in linear phase form (plot using \texttt{stem} rather than the \texttt{plot} function for a better view).

iii. Convert to zero phase by left-shifting \((N - 1)/2 = 5\), that is, by making it \textit{even}, or symmetric about zero. By doing this, you are now making the filter non-causal. Plot the impulse response.

iv. Take the FFT of both linear and zero phase impulse responses. What do you notice? How do their amplitude and phase responses compare?

v. Again, apply white noise and listen to the fruits of your labour. Even better, dig out some of your sound files and try out your filter designs.

3. (10 points) Submit your simple Karplus-Strong plucked string, implemented as an m-file function, which uses a circular delay-line with linear interpolation for implementation of fractional delay. Your function should have a single input parameter: frequency in Hz. Be prepared to show and discuss your synthesis in class.