2.161 Signal Processing: Continuous and Discrete
Fall 2008

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
DEPARTMENT OF MECHANICAL ENGINEERING

2.161 Signal Processing - Continuous and Discrete
Fall Term 2008

Problem Set 7: FIR Linear Filters

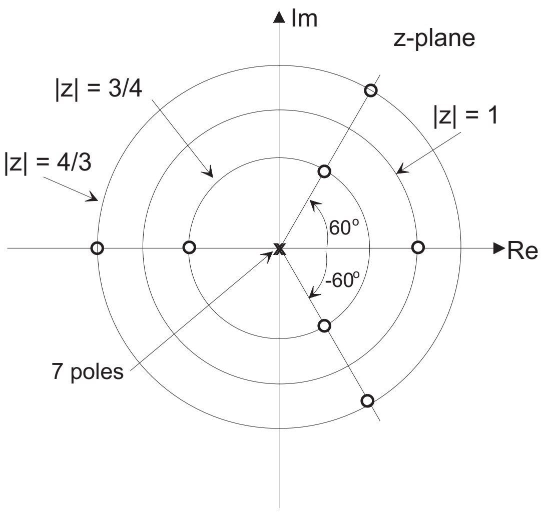
Assigned: October 30, 2008 **Due:** November 6, 2008

Problem 1: Prove the linear phase property of FIR filters in which the impulse response $\{h_n\}$, $n = 0 \ldots N-1$ (for N odd) is odd and symmetrical about its mid-point, that is when

$$h_n = -h_{N-1-n} \qquad 0 \le n \le (N-1)/2$$

Determine the phase response, and the equivalent delay.
Note: If N is odd then $h_{(N-1)/2} = 0$ for the impulse response to be odd and symmetrical about the mid-point.

Problem 2: Consider the pole-zero plot shown below



(a) Does the plot represent an FIR filter?

(b) Is it a linear phase system? Find the delay imposed by the filter.

(c) Use MATLAB to plot the magnitude and phase of the frequency response of the filter.

Problem 3: MATLAB exercise. Use this example to become familiar with the MATLAB functions `kaiserord`, `fir1` and `filter`. The task is to create a data set that is contaminated with additive random noise and to then examine the effects of a low-pass filter on the noisy signal. You will use `fir1` to design the filter and `filter` to execute the filtering operations. I have deliberately left the problem loosely defined so that you should experiment to find (what is to you) a satisfactory result.

(a) Create a data array of a 1 Hz. sinusoidal signal sampled at 10 Hz. in MATLAB, for example:

```
t = 0:.1:100;
sig = sin(2*pi*t);
```

(b) Contaminate your signal with a zero-mean noise signal:

```
noise = 2*(rand(size(signal)) - 0.5);
noisysig = sig + noise;
```

(c) Use the `kaiserord()` and `fir1` functions to design a Kaiser window based low-pass FIR filter (you choose the parameters) to "clean up" the signal. Use the `filter()` function to filter the noisy signal. Make plots of 1) the contaminated data and 2) the filtered output.

(d) Experiment with filter design until you are satisfied with the results. Hand in plots from your final choice.

(e) Plot out the first 100 or so points of the signal (sinusoid) and the filtered output and estimate the filter delay.

(f) Use MATLAB's `freqz()` to plot the frequency response of you filter.

(g) Use MATLAB's `zplane()` to make a pole-zero plot of your final filter, and describe how the low-pass action results.

Make sure you hand in your plots and design parameters for your filter.

Problem 4: Repeat Problem 3, but this time use `fir1` to design a bandpass filter that will attenuate noise components above and below the signal frequency. You can choose any window function here.

Problem 5: Another MATLAB problem using `fir1` and `filter`. You have taken a set of data, 2000 samples, sampled at 300Hz. The signal you are interested in consists of the sum of two sinusoids

$$f(t) = 1.5\sin(60\pi t) + 2\cos(180\pi t)$$

Unfortunately your instrumentation suffers from electromagnetic line frequency (60 Hz) interference so that the data you actually recorded was

$$s(t) = 1.5\sin(60\pi t) + 2\cos(180\pi t) + 1.0\sin(120\pi t)$$

Your task is to

(a) Create a data record in MATLAB representing the samples of $s(t)$.

(b) Design a "notch" (band-stop) filter, using `fir1`, that will attenuate the 60 Hz component significantly while leaving the other two components relatively "unscathed".

(c) Use `filter()` to filter you data record, and hand in magnitude spectra plots of the before and after operations. Label your output plots with the cut-off frequencies and filter length that you used.

(d) Use `freqz()` and `zplane()` to plot the frequency response and pole-zero plots.