ENEE 425

Section 0101 Fall 2004

Homework #10

You must attach a printed version of Matlab code (a Matlab file is preferable to copying and pasting a session, if possible). You may put several problems into a single printout, if the start of each problem is made obvious. Don't forget to answer the verbal questions too.

1) Compare the rectangular ('boxcar') window function with the Hanning window function. Use $M = 15$, so the windows will be 16 points long, and the windows will be Type II. Create a figure with 3 subplots.
   a) Create a window sequence of `N = 128` zeros, and replace the center 16 points with a rectangular window. Repeat for a Hanning window. Plot both sequences together in the first subplot (in different colors, if you can). Where does the Hanning window lie with respect to the rectangular window?
   b) For each window sequence `w` above, calculate the magnitude of the Fourier transform. In Matlab, the easiest way to do this is `magW = abs(fftshift(fft(w)));`. Scale the magnitudes so both have unit maximum, so we can compare them more easily, e.g. `magWNorm = magW/max(magW);`. Plot them together in the second subplot (in different colors if you can). On the horizontal axis, make sure that the plot runs from `-pi` to `pi` (instead of the default `1:N`). One way to do this is `omega=(([1:N]/N)*(2*pi))-pi;plot(omega,magWNorm)`. Compare the widths at half-max.
   c) In the third subplot, plot the same quantities as in (b), except in dB. Do not plot any values below -80 dB (you can control this using `axis([-inf inf -80 0])`. At what values of omega do the highest side peaks occur? What are the values in dB of those side peaks?

2) Repeat (2) for a Hanning window versus a Hamming window.

3) Repeat (2) for a Hanning window versus a Kaiser window with $\beta = 5$. If you have time, take a look at other values of β, to get a feel for how changing β, changes the window's properties.

4) Create a lowpass filter using the Kaiser window with the minimum order that meets these specifications:
   `rp = 3.0` dB [maximum gain distortion allowed in passband]
   `wp = 0.5` [passband upper limit, in Matlab normalized frequency (multiply by π to get radians)]
   `rs = 51.0` dB [minimum attenuation in stopband]
   `ws = 0.6` [stopband lower limit, in Matlab normalized frequency (multiply by π to get radians)]
   You will need to convert the distortion specifications from dB into fractional tolerances.

a) What is the minimum order needed? (Remember that the order of an FIR filter is one less than its length.) Is this the same as the value suggested by Kaiser (and Matlab) to be the minimum order?

b) Create the Kaiser window and the FIR filter. Remember that the `kaiser` function takes window size as an argument, not order (e.g. you should use something like `w = kaiser(n+1,beta)`;). Plot the filter magnitude (in dB) and phase (in degrees). This is easiest using `freqz()`. Does this filter have generalized linear phase? If it does but your phase plot is not exactly linear, how do you explain?

c) Verify that the filter meets the specifications (you can check with something like `H = freqz(b,1,[0.5 0.6]*pi);`). If it doesn't, you'll need to raise the window order.

5) Create a highpass filter using the Kaiser window with the minimum order that meets these specs:

`rs = 51.0` dB [minimum attenuation in stopband]

`ws = 0.5` [stopband upper limit, in Matlab normalized frequency (multiply by π to get radians)]

`rp = 3.0` dB [maximum gain distortion allowed in passband]

`wp = 0.6` [passband lower limit, in Matlab normalized frequency (multiply by π to get radians)]

You will need to convert the distortion specifications from dB into fractional tolerances.

a) What is the minimum order needed? (Remember that the order of an FIR filter is one less than its length.) Is this the same as the value suggested by Kaiser (and Matlab) to be the minimum order?

b) Create the Kaiser window and the FIR filter. Remember that the `kaiser` function takes window size as an argument, not order (e.g. you should use something like `w = kaiser(n+1,beta)`;). Plot the filter magnitude (in dB) and phase (in degrees). This is easiest using `freqz()`.Does this filter have generalized linear phase? If it does but your phase plot is not exactly linear, how do you explain?

c) Verify that the filter meets the specifications (you can check with something like `H = freqz(b,1,[0.5 0.6]*pi);`). If it doesn't, you'll need to raise the window order.

6)

a) Does anything prevent a Type I FIR filter from being used to window a lowpass filter? If so, what?

b) Does anything prevent a Type II FIR filter from being used to window a lowpass filter? If so, what?

c) Does anything prevent a Type I FIR filter from being used to window a highpass filter? If so, what?

d) Does anything prevent a Type II FIR filter from being used to window a highpass filter? If so, what?

e) Does anything prevent a Type I FIR filter from being used to window a bandpass filter? If so, what?

f) Does anything prevent a Type II FIR filter from being used to window a bandpass filter? If so, what?

g) Does anything prevent a Type I FIR filter from being used to window a bandstop filter? If so, what?

h) Does anything prevent a Type II FIR filter from being used to window a bandstop filter? If so, what?

After you're done with all the problems, take a look at and play around with the Matlab filter demonstration tool, `filtdemo`, *the filter design and analysis tool,* `fdatool`, *and the filter visualization tool,* `fvtool(b,a)`.