EE 381K **DIGITAL SIGNAL PROCESSING** Spring 1998

**COMPUTER HOMEWORK TWO**

Due: Friday, March 13, 1998

Note: This homework is due the Friday before Spring Break. Please drop it off at the lab (ENS 430). If you need to make alternative arrangements for drop-off, please contact me. Don't forget to **turn in** all your MATLAB code, including functions. You may require the function `hann.m` from the ftp site *pepperoni.ece.utexas.edu* in the directory *pub/Matlab*.

**1.** (This is just to get you warmed up.) Using a sampling frequency of 8000 Hz, generate a signal of length 512 samples that is the sum of two sine components of unit amplitude, one at $0.8\pi$ and one at $0.1\pi$. Now **subsample** this signal by a factor of two (i.e., throw away every other sample). This halves the effective sampling rate. We will now compute the frequency content of these two signals. In the first homework, we used `freqz` to compute samples of the Fourier transform directly. In this homework, we will first **window the signals** with a Hanning window before using `freqz`. This prevents problems caused by the signals not joining perfectly at the ends. We multiply the signals point-for-point with a window function which varies smoothly from zero at the ends to unity in the middle. Use the MATLAB function `hanning` to generate this window; if you do not have this function, use the function `hann` on the ftp site. After multiplying each signal with the appropriate length window, use `freqz` to compute the frequency content of the two signals over 256 points. Use the four-input version of `freqz` to specify the sampling frequency. On the same page, **plot the two spectra**, labeled with correct frequency axes (in Hz), and **explain the appearance** of the two plots.

**2.** In Module 7, we talked about reconstruction using a *zero-order hold*, in which the preceding sample is held at the output of the digital-to-analog converter (DAC) until the next sample arrives. Because the output of the DAC is now not a sequence of ideal impulses, its frequency response is modified by the transform of the zero-order-hold impulse response. This places a zero at the sampling frequency and its harmonics, which is quite desirable, but unfortunately causes the frequency response to fall slightly in the passband, being about 4 dB down at the band edge. In this part of the homework, you will design a digital compensation filter to correct for this rolloff.

The frequency response of the ideal digital compensation filter is given on the last page of Module 7. We will first compute the impulse response of the ideal filter by finding the inverse Fourier transform of the ideal frequency response. (Note that this is akin to frequency sampling, but we are using plenty of points and the filter function is quite smooth, so we can proceed.) **Generate** a frequency vector with `w=0:pi/128:255*pi/128`. Now **calculate the response** of the ideal filter from 0 to $\pi$ using the formula in the notes, and **'mirror image'** this about $\pi$ to give the ideal response from $\pi+\frac{\pi}{128}$ to $2\pi-\frac{\pi}{128}$. That is, the response at $\pi+\frac{\pi}{128}$ should equal the response at $\pi-\frac{\pi}{128}$, and the response at $2\pi-\frac{\pi}{128}$ should equal the response at $\frac{\pi}{128}$, with points in between mapping linearly. **Explain why** a filter with a response of this form will be purely real when inverse Fourier transformed.

Now **find** the inverse transform of the filter using `ifft` (inverse fast Fourier transform). **Verify** that the imaginary part is close to zero; if it is not, you have probably made an error in computing the filter symmetry. It is not exactly zero because of mathematical (roundoff) error. **Discard** the imaginary part using `real`. This returns the impulse response of the filter, although it is split into two halves because it is zero phase. Use `fftshift` to fix this and put the peak of the filter in the middle of the vector. Take a look at it (don't turn in this plot). Now **truncate** the filter symmetrically about the middle so that the impulse response is of length 21 samples. This is our *truncation design*. **Create another filter** which is the truncation filter windowed with a Hanning window of the correct length. This is our *windowing design*. We will compare the two.

Over 512 points, and assuming a sampling frequency of 8000 Hz, **compute** the frequency response of the two filters using `freqz`. On the same page, **plot** their magnitudes along with the magnitude response of the ideal filter (that is, turn in a page of two plots, each containing two curves). **Turn in** a second page, also of two plots, showing the *deviation* in dB of the actual filter responses from the ideal response. **Explain** the reason for the difference between the two filters.