EE 342 – Homework 12

Due April 20, 2005

1. A continuous-time signal is sampled at a 10 kHz rate. You need a filter which passes frequencies up to 1 kHz, and blocks frequencies above 2 kHz. What discrete-time frequency corresponds to the continuous-time frequency of 1 kHz, and what discrete-time frequency corresponds to the continuous-time frequency of 2 kHz?

2. Design a three-pole highpass Butterworth filter with a 3-dB point of 1 rad/sample.

   (a) Using the IIR design handout, find the low-pass 3-dB point of the continuous-time Butterworth filter.

   (b) Design a three-pole continuous-time Butterworth filter with this 3-dB frequency.

   (c) Convert the CT filter to a DT filter by either using

   $$H_{HP}(z) = H_{LP}(s)\Big|_{s=\frac{z+1}{z-1}}$$

   or by using the mapping $z = \frac{s+1}{s-1}$ to map the s-plane poles and zeros into z-plane poles and zeros.

   (d) Use MATLAB to plot the frequency response of your filter.

   (e) Verify your filter coefficients by using the MATLAB `butter` command. Because you are given the 3-dB point, the MATLAB command to design the filter is one of these:

   ```
   [b,a] = butter(3,1.0/pi,'high')
   [z,p,k] = butter(3,1.0/pi,'high')
   ```

   (f) Use MATLAB to simulate the response to the following signal: $x[n] = 1 + \sin(0.1\,\pi\,n) + \sin(0.6\,pi\,n)$. Plot the input and output signals $x[n]$ and $y[n]$.

3. Problem 12.15

4. Problem 12.16

5. Download the file falling.wav. Use the MATLAB function `wavread` to load the signal, and determine the sampling frequency for the signal.

   (a) Design an elliptic IIR lowpass filter to pass signals below 1 kHz with a ripple of no more than 1 dB, and block signals above 1.5 kHz with an attenuation of at least 50 dB. Use the MATLAB functions `ellipord()` and `ellip()` to do the design.

      i. What order of elliptic filter is needed?

      ii. Plot the poles and zeros of the filter (using the MATLAB `zplane()` function).

      iii. Plot the frequency response of the filter.

      iv. Filter the `falling` signal with the filter to create the signal `fall_ellip_low`. Use `wavwrite` to save this signal (with the same sampling frequency as the `falling` signal).

      v. Listen to the output `fall_ellip_low.wav`. Are the high frequencies blocked?

(b) Design an elliptic IIR highpass filter to block signals below 1 kHz with an attenuation of at least 50 dB, and pass signals above 1.5 kHz with a ripple of no more than 1 dB. Use the MATLAB functions `ellipord()` and `ellip()` to do the design.

  i. What order of elliptic filter is needed?
  ii. Plot the poles and zeros of the filter (using the MATLAB `zplane()` function).
  iii. Plot the frequency response of the filter.
  iv. Filter the `falling` signal with the filter to create the signal `fall_ellip_high`. Use `wavwrite` to save this signal (with the same sampling frequency as the `falling` signal).
  v. Listen to the output `fall_ellip_high.wav`. Are the low frequencies blocked?

(c) Design an FIR lowpass filter using the Hamming window to pass signals below 1 kHz with a ripple of no more than 1 dB, and block signals above 1.5 kHz with an attenuation of at least 50 dB. Use the MATLAB function `fir1()` to do the design. The cutoff frequency should be half way between the passband and stopband frequencies. Use trial and error to find the order of filter needed to meet the specifications.

  i. What order of FIR filter is needed?
  ii. Plot the poles and zeros of the filter (using the MATLAB `zplane()` function).
  iii. Plot the frequency response of the filter.
  iv. Filter the `falling` signal with the filter to create the signal `fall_fir_low`. Use `wavwrite` to save this signal (with the same sampling frequency as the `falling` signal).
  v. Listen to the output `fall_fir_low.wav`. Are the high frequencies blocked?

(d) Design an FIR highpass filter using the Hamming window to block signals below 1 kHz with an attenuation of at least 50 dB, and pass signals above 1.5 kHz with a ripple of no more than 1 dB. Use the MATLAB function `fir1()` to do the design. The cutoff frequency should be half way between the passband and stopband frequencies. Use trial and error to find the order of filter needed to meet the specifications.

  i. What order of FIR filter is needed?
  ii. Plot the poles and zeros of the filter (using the MATLAB `zplane()` function).
  iii. Plot the frequency response of the filter.
  iv. Filter the `falling` signal with the filter to create the signal `fall_fir_high`. Use `wavwrite` to save this signal (with the same sampling frequency as the `falling` signal).
  v. Listen to the output `fall_fir_high.wav`. Are the low frequencies blocked?

(e) Optional: Filter some of your favorite music using the filters designed above. Can you tell the difference between signals filtered with the elliptic filter and the FIR filter? You could also try some different types of filter, such as a bandpass filter.