THE UNIVERSITY OF TEXAS AT AUSTIN
Dept. of Electrical and Computer Engineering

EE345S Real-Time Digital Signal Processing Laboratory
Problem Set #2: Noise, Sampling, Interpolation, and Filter Design

Date assigned:	March 2, 2006
Date due:	March 23, 2006

Homework is due at the beginning of class. Late homework will not be accepted.

Reading: *Telecommunication Breakdown*, chapters 4 and 6

You may use any computer program to help you solve these problems, check answers, etc.

As stated on the course descriptor, "Discussion of homework questions is encouraged. Please be sure to submit your own independent homework solution."

Problem 2.1 Pole-Zero Diagrams and Frequency Responses. 20 points.
For each LTI system in problem 1.1 on the first homework assignment, plot the pole-zero diagram for the transfer function, compute the frequency response in closed-form, plot the magnitude response, indicate the type of filter (lowpass, highpass, bandpass, bandstop, allpass, notch), plot the phase response, and indicate whether or not the LTI system has linear phase.

Problem 2.2 Impulse Train. 20 points.
Johnson & Sethares, 4.5.

Problem 2.3 Upconversion. 20 points.
Johnson & Sethares, 6.13.

Problem 2.4 IIR Filtering. 20 points.
This problem is the complement to problem 1.5 in that it asks you to design discrete-time infinite impulse response filters for processing digital audio prior to FM transmission. After digital filtering, the result would pass through a digital-to-analog converter and then be converted to an FM signal. In the transmission, the bandwidth of the message (CD audio) is limited to 15 kHz. The sampling rate for CD audio is $f_s = 44.1$ kHz.

A rule-of-thumb for filter design is that the distance from the passband frequency to the stopband frequency should be at or greater than 10% of the passband frequency. The filter specification will be that the passband will be from 0 Hz to 13 kHz, and that the stopband will be from 14.5 kHz to $\frac{1}{2} f_s$. Here, $\frac{1}{2} f_s$ is 22.05 kHz. In the passband, the deviation of the magnitude response (sometimes called passband ripple) should be no more than 1 dB. In the stopband, the maximum magnitude frequency response should be 90 dB down from full scale.

(a) Design Butterworth, Chebyshev type I, Chebyshev type II, and Elliptic IIR filters of minimum filter order to meet the specifications. For IIR filters, the filter order is the number of poles. Turn in a plot of the magnitude responses for each IIR filter you have designed to meet the specifications.

(b) List the filter orders required for filters to meet the specification. Which IIR filter family gives the shortest filter order? Plot the pole-zero diagram for the elliptic filter you designed in (a). Submit the feedforward and feedback coefficients for the filter. Type `help filtdemo` to find out how to retrieve the filter coefficients for a design. Type `help filter` to find out what the return values mean.

(c) Use the Elliptic filter you designed in (a) for this part. Assuming that the input data samples were in single precision floating point and the IIR filter coefficients were stored in single precision floating point format,

   - How many instruction cycles on the TMS320C6700 DSP family would it take to compute one output value for each input value if the IIR filter routine were handcoded in assembly for optimal performance?
   - How much storage in bytes would it take to store the IIR coefficients and the circular buffer for the current and past inputs and the circular buffer for past outputs?

(d) How would you advocate using either an FIR filter or IIR filter for this application? In your answer, please include the design method for which you are advocating.

You may use **matlab** or any other software to design the filters and plot the magnitude responses. In Matlab, type `filtdemo` for a graphical user interface for filter design. By looking at the plot of the magnitude responses using `filtdemo`, validate that the filters designed by Matlab meet the specifications. Carefully inspect the response in stopband to make sure that it meets specifications.

Problem 2.5 And Now, For the Rest of the Story. 20 points.

This problem concerns implementing discrete-time IIR filters in fixed-point arithmetic. Working the problem requires LabVIEW 7.1 or above and the Digital Filter Design Toolkit 7.5 for LabVIEW.

If you would like to work on your own computer, please download and install LabVIEW 7.1

https://www.engr.utexas.edu/itg/ni/english71.bat and the Digital Filter Design Toolkit 7.5

https://www.engr.utexas.edu/itg/ni/digfil.bat

To be able to download this software, you must be on campus, or connected through the UT Austin Virtual Private Network (VPN).

Once you run LabVIEW, pull-down Help ... Find Examples ... Toolkits and Modules ... Digital Filter Design ... Fixed Point Filters. Open the demonstration entitled "Analyze Quantized Filter with Different Structure". Run the demonstration by hitting the white right arrow key that is located just under the Edit menu button.

The demonstration represents the filter coefficients in a fixed-point number format. The fixed-point number format consists of a total wordlength (wl) in bits and the number of integer bits (iwl). So, a number represented with 14 bits to the right of the binary point, 1 bit to the left of the binary point, and 1 sign bit would have wl=16 and iwl=1. Is this demonstration, all fixed-point numbers have a sign bit.

The demonstration shows the magnitude response of the filter when using a floating-point number format and arithmetic (in white) and when using a fixed-point number format and arithmetic (in red).

(a) Change the target structure to IIR Cascaded Second Order.

   i. With the wordlength set to 16 and an integer wordlength set to 1, does increasing the integer wordlength from 1 to 10 show any significant improvement or loss in the closeness of the magnitude responses of the fixed-point and floating-point versions of the filter? If so, give the integer wordlength setting.

   ii. What about when increasing the integer wordlength from 11 to 14?

   iii. With the wordlength set to 16 and an integer wordlength set to 1, reduce the wordlength from 16 to 3. What is the smallest wordlength that does not introduce significant degradation in the magnitude response under a fixed-point implementation vs. the magnitude response under a floating-point implementation?

(b) Change the target structure to IIR Direct Form I.

   i. With the wordlength set to 16 and an integer wordlength set to 1, increasing the integer wordlength from 1 to 14. What is the integer wordlength that gives the best match between the magnitude response for the fixed-point implementation vs. the magnitude response under a floating-point implementation?

   ii. With the wordlength set to 16 and an integer wordlength set to 1, what is the smallest wordlength that does not introduce significant degradation in the magnitude response under a fixed-point implementation vs. the magnitude response under a floating-point implementation?

iii. Find the smallest wordlength setting for IIR Direct Form I (and give the integer wordlength setting that you used) that yields essentially the same closeness in magnitude response to the floating-point implementation as you found in part (a)-iii.

(c) Is the any difference in your answers in (b) if you change the target structure to IIR Direct Form II? If yes, then please note the changes.

(d) What conclusions do you draw in choosing what filter structures to use when implementing discrete-time digital IIR filters?