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

EE345S Real-Time Digital Signal Processing Systems
Problem Set #1: Signal, Systems, Sinusoids, Transforms, and FIR Filters

Date assigned:	February 2, 2006
Date due:	February 16, 2006

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

Reading: Johnson & Sethares, chapters 1–4, and Tretter, chapters 1–3

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."

TAs may be reached during the lab hours for the course to answer questions about homework problems if they are not busy with lab duties. Feel free to go to any TA with questions.

Matlab's strength is in performing matrix-vector calculations which are convenient for computing signals and test signal processing algorithms. Please see the Matlab handout in Appendix D of the course reader for more information.

Alternately, you might find Mathematica useful. The complex number $1 + 2j$ is represented as 1 + 2 I. The Signals and Systems Pack (a commercial package) can be accessed on the LRC Unix machines

```
In[1]:= AppendTo[$Path, "/home/ecelrc/faculty/bevans" ];
In[2]:= Needs[ "SignalProcessing`Master`" ];
```

The single quotes in the Needs command are back quotes and not forward quotes. The Signals and Systems Pack can compute z, Fourier, and Laplace transforms in symbolic form. For example, the z-transform of $a^n u(n)$ can be computed as

```
In[3]:= ZTransform[ a^n DiscreteStep[n], n, z ]
```

Mathematica is installed on the Unix machines sunfire1 and sunfire2. The Unix commands to run Mathematica are `math` for the command-line interface and `mathematica` for the GUI.

Problem 1.1 Transfer Functions. 20 points.

Find the transfer function in the z-domain and the associated region of convergence for the z-transform function of the following linear time-invariant discrete-time systems:

(a) Causal three-tap averaging filter

(b) Causal discrete-time approximation to first-order derivative

(c) Causal discrete-time approximation to an integrator

(d) Oscillator whose impulse response is a causal (one-sided) sine signal with fixed-frequency ω_0

For this problem, the following z-transform references might be helpful:

- Appendix F in Johnson & Sethares book *Telecommunication Breakdown*
- Sections 5.1 and 5.2 in Lathi's book *Linear Systems and Signals*
- Slides from lecture 5 may also be helpful, esp. slides 5-21 and 5-22.

Recall that transfer functions of the form $H(z) = Y(z)/X(z)$ only apply for linear time-invariant systems. A linear time-invariant system is uniquely defined by its impulse response. The generalized transform of the impulse response is also the transfer function.

Problem 1.2 Sinusoidal Demodulation. 20 points.
Johnson & Sethares, 2.3, on page 21.
For this problem, sections 7.3 and 7.4 in Lathi's book Linear Systems and Signals *may be helpful.*
Additional questions:

(a) Assume that $f_1 = f_0$. How would you propose to recover $w(t)$?

(b) Assume that $f_1 \approx f_0$ but $f_1 \neq f_0$. How would you propose to recover $w(t)$?

Problem 1.3 Spectral Analysis of a Random Signal. 20 points.
Johnson & Sethares, problem 3.3, on page 43. Please submit the plots with your homework solution.

Problem 1.4 Spectral Analysis for a Deterministic Signal. 20 points.
Johnson & Sethares, problem 3.10, on page 48. Please comment on the plots. Please submit the plots with your homework solution.

Problem 1.5 Finite Impulse Response Filter Design. 20 points.
This problem asks you to design a digital filter to apply to a digital discrete-time audio signal (one signal) before the signal is converted to an analog continuous-time message signal

for frequency modulated (FM) radio transmission. In FM transmission, the message bandwidth is limited to 15 kHz. The sampling rate for audio CD and hence the digital-to-analog converter 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 finite impulse response (FIR) filters *with the minimum filter order* to meet the specification by using the Remez (a.k.a. Parks-McClellan), FIR Least Squares (FIRLS), and Kaiser Window design methods. Turn in a plot of the magnitude response for the each filter.

(b) Plot the impulse response of the FIR filter designed by the Remez algorithm. If you type `help filtdemo` in Matlab, then you will see how to retrieve the transfer function for the current filter designed by filtdemo. Is there any structure in the impulse response?

(c) Give the filter lengths required for filters designed for each method. Which method gives the shortest filter length?

(d) Assuming that the input data samples were in single precision floating point and the FIR 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 FIR filter routine were handcoded in assembly for optimal performance?
  - How much storage in bytes would it take to store the FIR coefficients and the circular buffer for the current and past inputs?

For FIR filters, the filter order is one less than the filter length. That is, the filter order is the number of zeros, and the filter length is the number of coefficients.

Sometimes, the automatic order estimation for Remez and Kaiser FIR filters may be too high or too low. So, if the magnitude response of the designed filter does not meet the specifications, then increase the filter order until it does. Once the magnitude response of the designed filter does in fact meet the specifications, then decrease the filter order to see if you can find a filter of lower order that also meets the filter specifications.

For the FIR Least Squares (FIRLS) filter design, you will have to experiment with the right setting of the filter order. Increasing the order does not always produce a filter that is closer to meeting the magnitude response specification. This happens because the FIRLS tries to minimize the average error between the designed filter and the ideal filter across the entire range of frequencies, instead of minimizing the maximum error across the entire range of frequencies.

In all three FIR filter design algorithms, you will have to search for the design with minimum filter order.

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.