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

EE313 Linear Systems and Signals
Problem Set #5: Discrete-Time Systems

Prof. Brian L. Evans

Date assigned:	September 23, 2010
Date due:	September 30, 2010

Homework is due at 11:00 am sharp in class. Late homework will not be accepted.

Reading: *Signals and Systems*, Section 3.5

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 absolutely sure to submit your own independent homework solut ion."

The office hours in ENS 433B for Prof. Evans follow:

- Tuesdays 12:15pm–1:00pm (right after lecture)
- Wednesdays 12:30pm–2:00pm
- Thursdays 12:15pm–1:00pm (right after lecture)
- Fridays 9:30am–11:00am

Prof. Evans will not be available for his coffee hour on Friday afternoon, September 24th. Prof. Evans can be reached at `bevans@ece.utexas.edu`.

The teaching assistant is Mr. Jackson Massey. His office hours will be on Wednesdays 4:00pm–7:00pm in ENS 138. Mr. Massey can be reached at `jackson.massey@gmail.com`.

The ECE Department is offering tutoring sessions for all basic sequence ECE courses, including EE 313, on Sundays through Thursdays, 7:00–10:00 pm, in ENS 314. Mr. Massey will be a tutor during the Monday and Wednesday evening sessions.

Problem 5.1 Discrete-Time System Properties

Roberts, Chapter 3, Problem 41.

Problem 5.2 Discrete-Time Convolution

Let $x_1[n]$ be a causal rectangular pulse of L_1 samples and let $x_2[n]$ be a causal rectangular pulse of L_2 samples.

(a) Compute $y[n] = x_1[n] * x_2[n]$. Please give your answer in terms of L_1 and L_2.

Hint: It may help you to define $L_{\min} = \min(L_1, L_2)$ *and* $L_{\max} = \max(L_1, L_2)$.

(b) Plot $y[n]$ for $L_1 = 5$ and $L_2 = 10$.

Problem 5.3 Discrete-Time Integrator

In class, I mentioned that the analogy to integration in discrete-time is running summation. With input $x[n]$ and output $y[n]$, running summation is defined as

$$y[n] = x[n] + y[n-1]$$

for $n \geq 0$.

(a) What is the initial condition?

(b) To what value should the initial condition be set to guarantee that the system is linear and time-invariant?

(c) What is the impulse response? *Hint: Let* $x[n] = \delta[n]$ *and find the output. You might be able to find this by computing the first several values of the output to see the pattern. Your formula for the impulse response should be valid for all values of* $n \in (-\infty, \infty)$.

(d) What is the output $y[n]$ when $x[n] = u[n]$?

(e) Is the system bounded-input bound-output (BIBO) stable? *Hint: The answer to part (d) will help here.*

Problem 5.4 Filtering Sound

This problem will be worked in Matlab. Please keep the volume on the speaker(s) the same throughout each part of the problem. Set the volume so that the audio signal that you will play in part (a) is audible. Note that the audio signal may not play if you are using a remote connection, so run Matlab locally on your PC or Unix workstation. You might consider using headphones to keep others in the vicinity from pulling your power plug.

Matlab has several sound files built in. Here is how to load and play the gong sound file:

```
load gong
sound(y,Fs);
```

The `load gong` command sets two variables: `y` contains samples loaded in from the sound file, and `Fs` is the sampling rate of the sound file (defaults to 8192 Hz). You can type `Fs` (without a semicolon) to see the value that Matlab uses. Many sound cards only allow a finite number of choices of `Fs`. You may need to load a sound file to initialize the value of `Fs`.

(a) **Sinusoidal tone**. The gong sound has a dominant frequency (tone) at 996 Hz. Generate a sinusoid at 996 Hz continuous-time frequency by filling in the appropriate expression for the discrete-time frequency `w0`, which is in units of radians/sample:

```
load gong             %%% Sets sampling rate Fs of sound card
n = 1 : length(y);    %%% Generate samples as long as gong sound
F0 = 996;             %%% 996 Hz
w0 = ?;               %%% Discrete-time frequency in rad/sample
ycos = cos(w0*n);
sound(ycos,Fs);
```

Question: What did you put for `w0`?

(b) **LTI Filtering the Gong Sound File.** In this problem, you will be filtering the gong sound file by convolving the gong sound file with the impulse response of the LTI filter.

```
load gong                                   %%% Load gong file
sound(y, Fs);                               %%% Play gong file
h1 = [0.538809 -0.778328 0.538809];         %%% Impulse response
y1 = conv(h1, y);                           %%% Filter gong file
sound(y1,Fs)                                %%% Play result
```

Question: How does the filtered gong sound different from the original gong sound?

Question: How many samples are in the filtered gong sound, y1?

(c) **Resampling of the Gong Sound File.**

```
load gong             %%% Load gong file
sound(y, Fs);         %%% Play gong file
y2 = y(1:2:end);      %%% Keep every other sample
sound(y2, Fs);        %%% Play processed gong file
```

Question: How does the processed gong sound different from the original gong sound?

Question: How many samples are in the processed gong sound, y2?