University of Illinois at Urbana-Champaign
Department of Electrical and Computer Engineering

ECE 311: Digital Signal Processing Lab

LAB 4: **System Properties**
Fall 2015

1 Objective

The objective of this lab is to help you understand the properties of a discrete-time system, especially linearity, shift-invariance, and BIBO (Bounded-Input Bounded-Output) stability.
Key concepts:

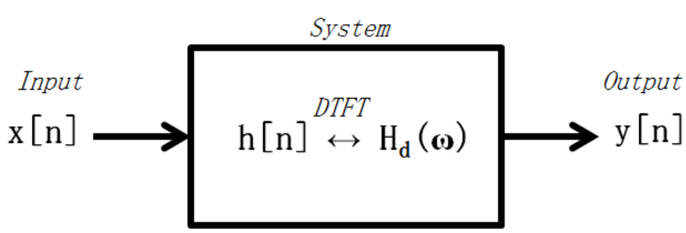
- System properties: linearity, shift-invariance, and BIBO stability
- LSI system: impulse response, transfer function and stability

2 Background

A system has a DTFT $H_d(\omega)$. In the following sections we represent its input-output relationship as $x[n] \to y[n]$, which refers to

$$x[n] \bigotimes h[n] = y[n] \text{ in the discrete time domain}$$
$$X_d(\omega)H_d(\omega) = Y_d(\omega) \text{ in the DTFT domain}$$

where $\bigotimes$ represents a convolution, and $h[n]$ is the impulse response of the system to $\delta[n]$.



2.1 System properties

Linearity
A discrete-time system is linear if it satisfies the decomposition property and it is both "zero-state" linear and "zero-input" linear. These terms are defined below.

- Decomposition property
  A system satisfies the decomposition property if its output $y[n]$ can be written as:

$$y[n] = y_x[n] + y_s[n],$$

where $y_x[n]$ is the response of the system due only to the input $x[n]$ while the initial conditions (states) of the system are set to zero ("zero state"), and $y_s[n]$ is the response of the system due only to the initial conditions (states) of the system, while the input is set to zero ("zero-input").

- Zero-state linear
  A system is zero-state linear if, when the initial conditions of the system are set to zero, it satisfies both (1) *homogeneity* and (2) *additivity*, defined as follows:

  1. A system with input $x[n]$ and output $y[n]$ satisfies *homogeneity* if for every input $x[n]$ and for every real-valued constant a, the following holds,
  $$\text{if } x[n] \to y[n], \text{ then } ax[n] \to ay[n].$$
  2. A system with input $x[n]$ and output $y[n]$ satisfies *additivity*, if for every pair of inputs $x_1[n]$ and $x_2[n]$, the following holds,
  $$\begin{array}{l}\text{if } x_1[n] \to y_1[n] \text{ and } x_2[n] \to y_2[n], \\ \text{then } x_1[n] + x_2[n] \to y_1[n] + y_2[n].\end{array}$$

- Zero-input linear
  A system with a set of N initial conditions $\{y_k[n_k] = c_k\}_{k=0}^N$ and the corresponding zero-input response to these initial conditions $y_{s,1}[n]$ and a second set of N initial conditions $\{y_m[n_m] = d_m\}_{\{m=0\}}^N$ and the corresponding zero input response to these initial conditions $y_{s,2}[n]$ is zero-input linear if the following holds,

$$\begin{array}{c}\text{if } \{y_k[n_k] = c_k\}_{k=0}^N \to y_{s,1}[n] \text{ and } \{y_m[n_m] = d_m\}_{m=0}^N \to y_{s,2}[n] \\ \text{then } \{y_k[n_k] = ac_k + bd_k\}_{k=0}^N \to ay_{s,1}[n] + by_{s,2}[n],\end{array}$$

  for all real-valued constants a and b.

Shift-invariance
A discrete-time system is shift-invariant if a shift in the input always leads to a corresponding shift in the output, meaning the system satisfies the following:

$$\text{if } x[n] \to y[n], \text{ then } x[n - n_0] \to y[n - n_0], \forall n_0 \text{ and } \forall x[n].$$

Causality
A system is causal if for every n, $y[n]$ depends only on $x[m]$, $m \leq n$. Thus, for causal systems, current outputs depend only on current and past inputs (do not depend on future inputs).

Stability
A signal $x[n]$ is bounded if there exists a positive constant a, such that

$$|x[n]| < a < \infty, \forall n.$$

A system is bounded-input, bounded-output (BIBO) stable if for every bounded input $x[n]$, the resulting output $y[n]$ is bounded.

2.2 Linear shift-invariant (LSI) system

Impulse response
The relationship between the input and output of an LSI system is given by the convolution sum,

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = x[n] \bigotimes h[n],$$

where the sequence $h[n]$ is the response of the system to the discrete time impulse $\delta[n]$.

Transfer function
The transfer function of an LSI system with input $x[n]$ and output $y[n]$ is defined as

$$H(z) = \frac{Y(z)}{X(z)}|_{\text{zero initial conditions}},$$

Where $Y(z)$ and $X(z)$ are the z-transforms of $x[n]$ and $y[n]$, respectively.

An LSI system can be written in the form of a *linear constant-coefficient difference equation* (LCCDE):

$$y[n] + a_1 y[n-1] + \ldots + a_N y[n-N] = b_0 x[n] + b1x[n-1] + \ldots + b_M x[n-M],$$

or more compactly:

$$y[n] + \sum_{k=1}^{N} a_k y[n-k] = \sum_{l=0}^{M} b_l x[n-l].$$

The transfer function of an LCCDE in the above form is

$$H[z] = \frac{\sum_{l=0}^{M} b_l z^{-l}}{1+\sum_{k=1}^{N} a_k z^{-k}} \ .$$

It follows that the zeros of the system are all solutions for

$$\sum_{l=0}^{M} b_l z^{-l} = 0,$$

and the poles are all solutions for

$$1 + \sum_{k=1}^{N} a_k z^{-k} = 0.$$

Stability
An LSI system is BIBO stable if and only if $\sum_{n=-\infty}^{\infty} |h[n]| < \infty$.

A causal pole-zero LSI system is BIBO stable if and only if all of its poles are inside the unit circle (of the z plane).

3 Lab Demo

Calculate the system response by solving system difference equation
If the system difference equation is given, one can calculate the system response by solving the difference equation directly. This approach applies to both LSI and non-LSI systems.

For instance, suppose a system is given by:

$$y[n] = (\tfrac{3}{7})^n y[n-1] + x[n]$$

The following MATLAB code calculates the system response with a given input $x[n]$, assuming it is a zero-stat signal ($y[n] = 0$ for $n < 0$):

```
function y = system(x);
% change x to a column vector no matter its original orientation
x = x(:);
N =length(x(:));
% set up the system initial state
y = [];
y(0+1) = 0 +x(0+1);
% solve the difference equation recursively
for n = 1:N-1
    y(n+1)= ((3/7)^n)*y(n) + x(n+1);
end
```

An example of linear vs non-linear systems

Consider the following two systems:

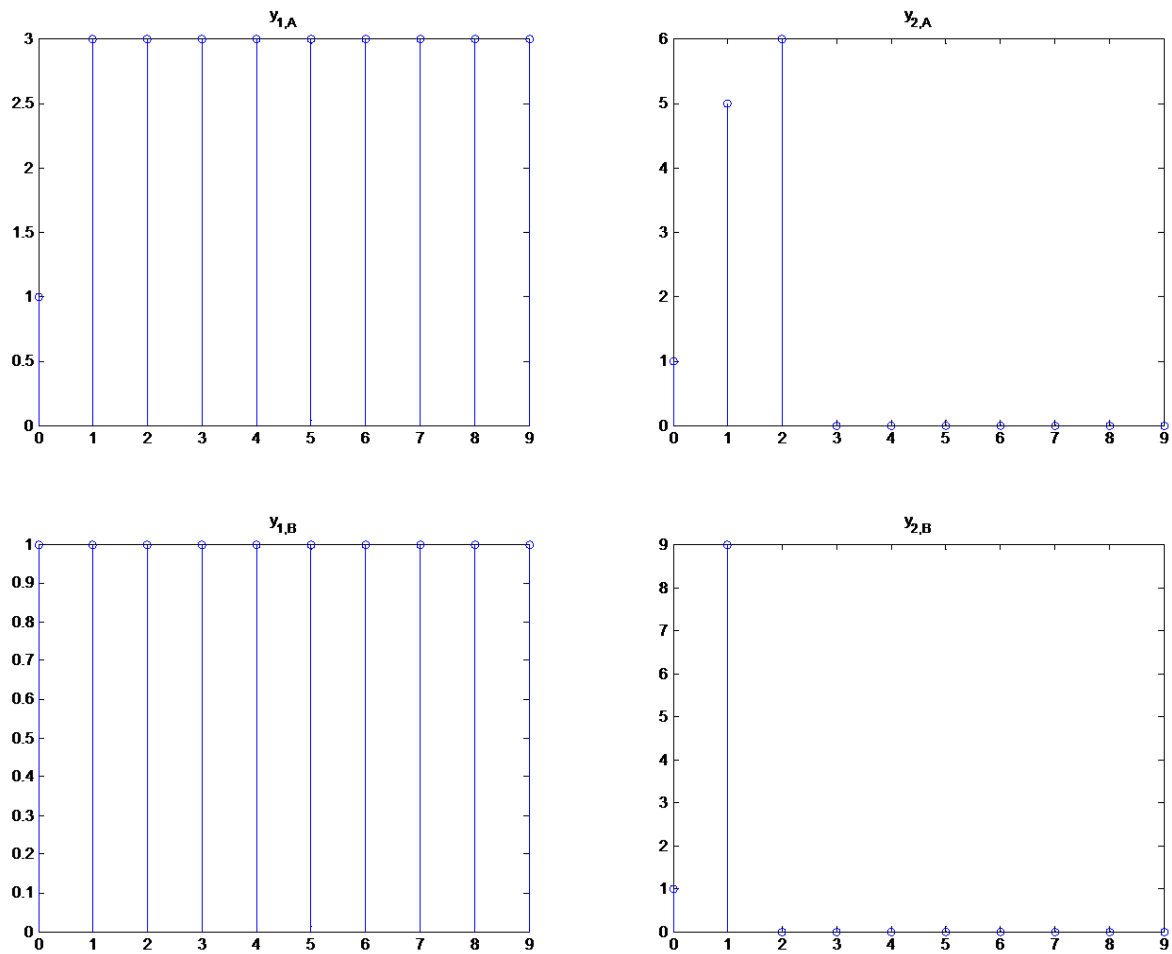
$$y_A[n] = x[n] + 2x[n-1],$$
$$y_B[n] = (x[n])^2.$$

Now consider two input signals:

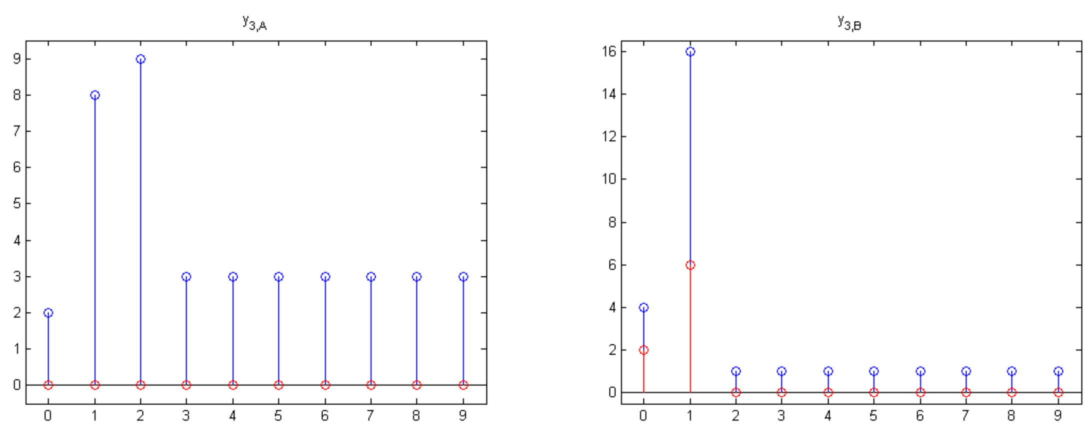
$$x_1[n] = u[n],$$
$$x_2[n] = \delta[n] + 3\delta[n-1].$$

The corresponding output signals are:

$$y_{1,A} = \delta[n] + 3u[n-1]$$
$$y_{2,A} = \delta[n] + 5\delta[n-1] + 6\delta[n-2]$$
$$y_{1,B} = u[n]$$
$$y_{2,B} = \delta[n] + 9\delta[n-1]$$



Now we create a new input signal $x_3[n] = x_1[n] + x_2[n]$. The outputs are shown below, where the black lines show $y_{3,A}$ and $y_{3,B}$, and the red lines show the differences $y_{3,A}[n] - (y_{1,A}[n] + y_{2,A}[n])$ and $y_{3,B} - (y_{1,B}[n] + y_{2,B}[n]))$.



We can see that system A is linear because it obeys superposition (the red points are all zero), but system B is not.

An example of shift-invariant vs shift-varying systems

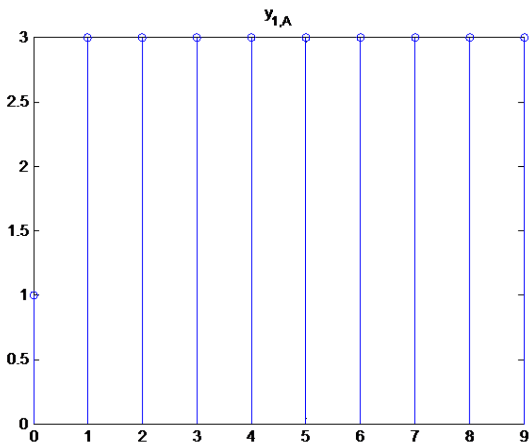
Consider the following two systems:

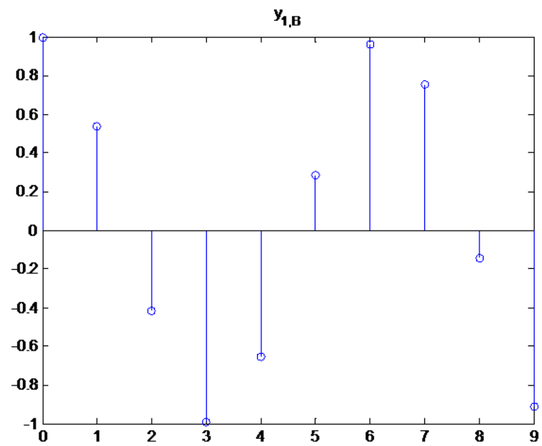
$$y_A[n] = x[n] + 2x[n-1]$$
$$y_B[n] = x[n]\cos(n)$$

Let the input to the systems be

$$x_1[n] = u[n]$$

The outputs of the systems are shown in the graphs below.

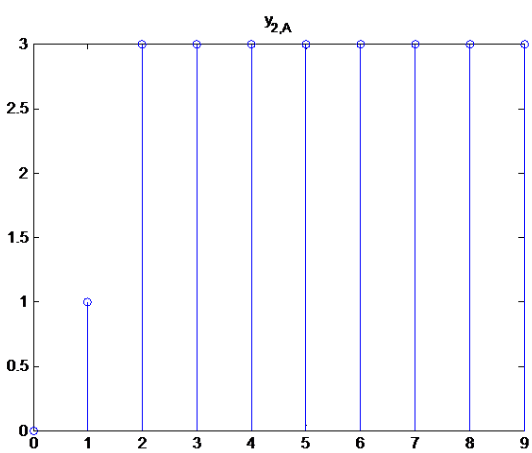


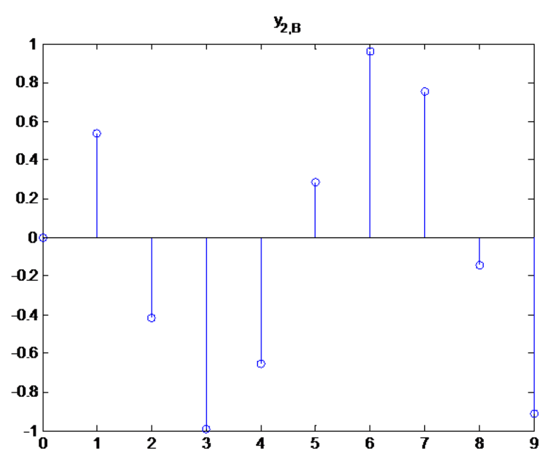


Now if the input is shifted such that

$$x_2[n] = x_1[n-1] = u[n-1].$$

then the outputs of the system are as shown below.





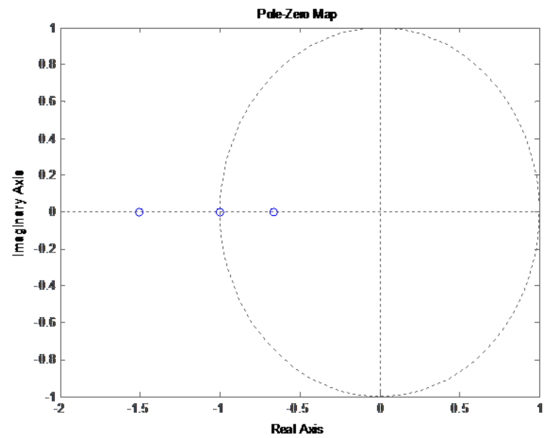
From the above graphs, it can be seen that for a shifted input $x_1[n-1]$, the output is indeed $y_{1,A}[n-1]$ for the system A, which does not hold for the system B. Therefore, system A is shift-invariant, and system B is shift-varying.

An example of stable vs unstable systems

A low-pass filter may be implemented via the transfer function

$$H(z) = 0.1199 + 0.3801z^{-1} + 0.3801z^{-2} + 0.1199z^{-3} \ .$$

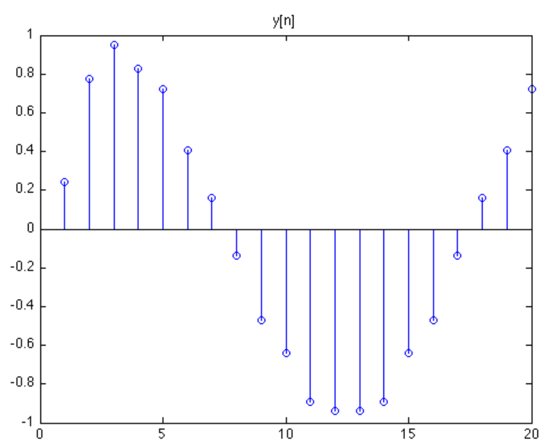
The pole-zero diagram of the system is shown below. The system is stable since its poles ($z = 0$) are in the unit circle.



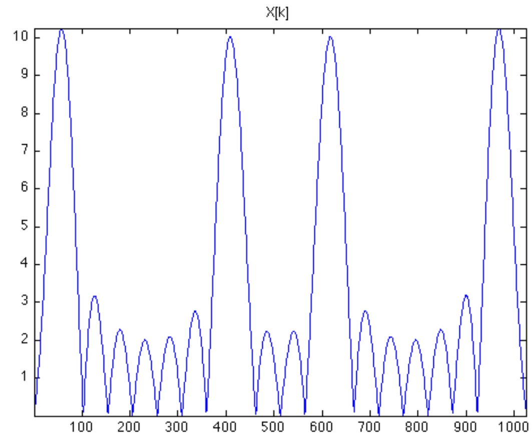
This pole-zero diagram is displayed using

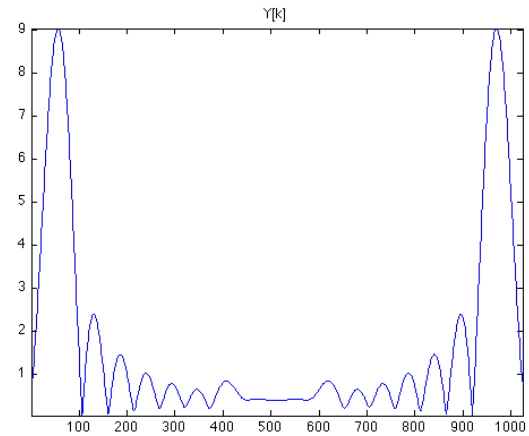
```
% Create the system by specifying the cofficients of an LCCDE equation
b = [0.1199,0.3801,0.3801,0.1199];
a = 1;
sysA = tf(b,a,-1);
% Plot the zero-pole digram
figure; axis auto; pzplot(sysA);
```

If the input is $x[n] = \cos(\frac{\pi n}{10}) + \cos(\frac{8\pi n}{10})$, the output will also be a bounded signal.



The input and output DFT spectra are shown below. Note the frequency component $\cos(\frac{8\pi n}{10})$ (near $k = 400$, $k = 600$) of the input signal is suppressed by the low-pass filter.

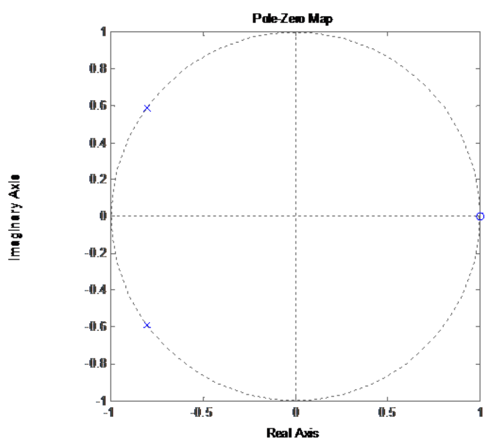




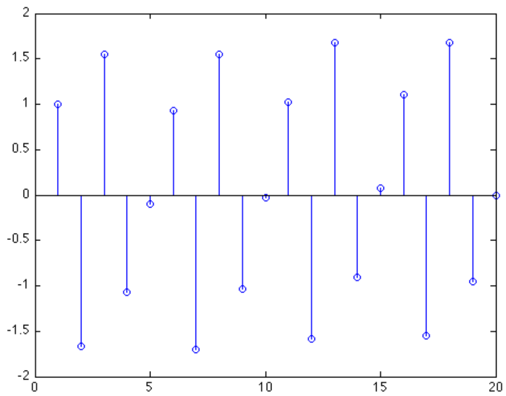
Now consider another system

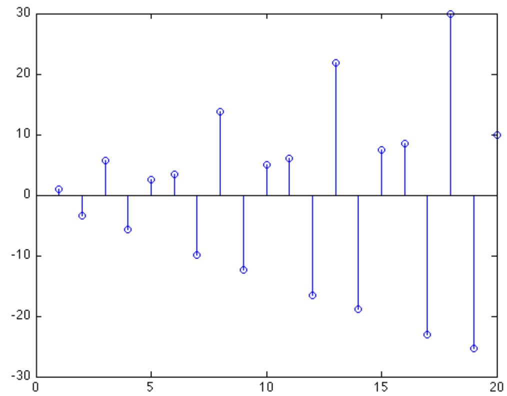
$$H(z) = \frac{z}{(z+e^{-\frac{j8\pi}{10}})(z+e^{\frac{j8\pi}{10}})}.$$

The pole-zero diagram of the system is shown below. This system is not stable because it has poles on the unit circle.



With an input of $x[n] = \cos(\frac{\pi n}{10})$, the output is a bounded signal as shown below (left). However, with an input of $x[n] = cos(\frac{8\pi n}{10})$, the output is not bounded! This is also shown below (right). In the case of $x[n] = cos(\frac{8\pi n}{10})$, we have excited the pole frequency of the system (where $H(z) \to \infty$).

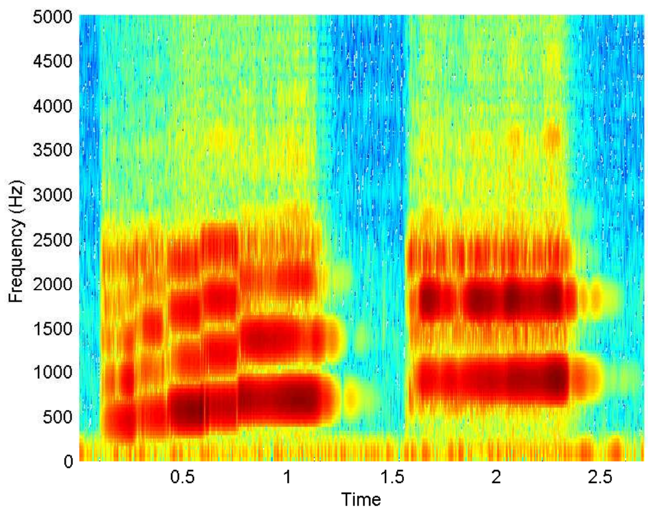




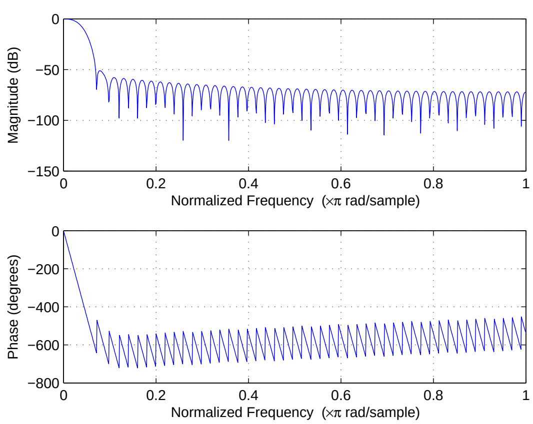
The `Matlab` code is given below.

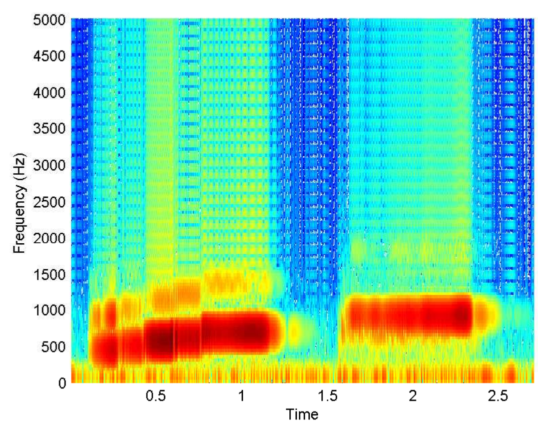
```
%% System A
% Create the system by specifying the cofficients of an LCCDE equation
b = [0.1199,0.3801,0.3801,0.1199];
a = 1;
sysA =tf(b,a,-1);
% Plot the zero-pole diagram
figure; axis auto; pzplot(sysA);
% Create the input signal
n = 0:19;
x = cos(pi*n/10) + cos(8*pi*n/10);
% Calculate the output signal
y = filter(b, a, x);
% Plot the output signal
figure; stem(y); title('y[n]');
% Calculate and plot the DFT spectra
num = 1024; Xk = abs(fft(x,num));
figure; plot(Xk); title('X[k]');axis tight;
Yk = abs(fft(y,num));
figure; plot(Yk); title('Y[k]');axis tight;
%% System B
% Create the system by specifying the coefficients of an LCCDE equation
a = poly([exp(-1i*8*pi/10) exp(1i*8*pi/10)]);
b = poly(1);
sysB = tf(b,a,-1);
% Plot the zero-pole diagram
figure; axis auto;
pzplot(sysB);
% Create the first input signal
n = 0:19; x = cos(pi*n/10);
% Calculate the output signal
y = filter(b, a, x);
% Plot the output signal
figure; stem(y);
% Create the second input signal
n = 0:19; x = cos(8*pi*n/10);
% Calculate the output signal
y = filter(b, a,x);
% Plot the output signal
figure; stem(y);
```

Processing an audio signal by an LSI vs a non-LSI system
The spectrogram of a flute scale is shown below.



The signal is first passed through a low-pass filter with a cutoff frequency of 880 Hz (approximately the high 'A' of the scale, for those with a musical background). The low-pass filter is an LSI system, which is BIBO stable. The filter and the DFT spectrum of the system output are shown below. The high-frequency harmonics are suppressed as expected. You can play the MATLAB code attached below to hear the difference between the signals before and after filtering. After filtering, the signal is much less recognizable as a flute. This is because the higher frequencies in the signal give musical instruments their unique sounds ('timbre').



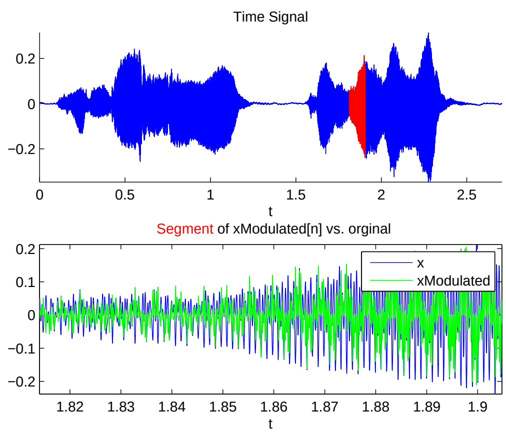


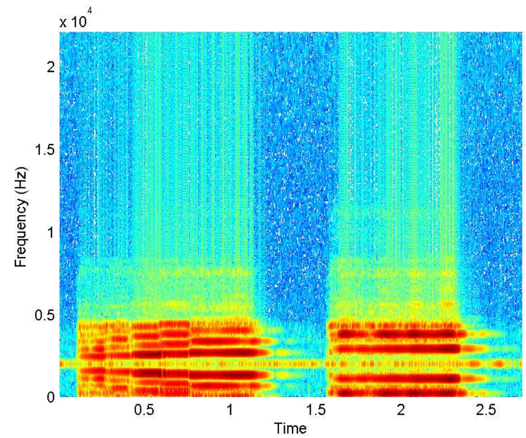
The original signal is then passed through a shift-varying (non-LSI) system defined as follows:

$$y[n] = x[n]cos(\frac{2\pi(2000)n}{Fs})$$

The modulation is a time-varying operation, because we multiply with a cosine that is dependent on n. An example of what this modulation looks like, zooming in on a small segment of the time domain signal, is shown below (left).
The DFT spectrum of the output signal also shown (right). Compared with the DFT spectrum of the original signal, the spectrum has been shifted and centered about 2000 Hz. You can play the MATLAB

code attached below to hear the difference between the input and output signals. Because the negative part of the spectrum is shifted up across f=0, you can hear a descending scale as well!





MATLAB code

```
%% load data
[x,Fs] = wavread('Flute.wav');
soundsc(x,Fs);
figure();
L = 256; nfft = 1024; %Hamming (default) window length is 256, DFT length is 1024 for spectrogram
spectrogram(x,L,128,nfft,Fs,'yaxis');
set(gca,'ylim',[0 5000]); % get a closer look at the frequency content
%% pass the data through an LSI system, i.e. low-pass filter
d = fdesign.lowpass('N,Fc',100,400*2,Fs);
hd = design(d);
b = hd.Numerator; a = 1;
% plot frequency response of filter
figure(); freqz(b,a,1024);
xFiltered = filter(b,a,x); % filter signal
soundsc(xFiltered,Fs);
% show new spectrogram
figure(); spectrogram(xFiltered,L,128,nfft,Fs,'yaxis');
set(gca,'ylim',[0 5000]); % get a closer look at the frequency content
%% pass the data through a non-LSI (Shift-varying) system
w0 = 2*pi*2000;
t = (0:length(x)-1)/Fs;
temp = cos(w0*t);
xModulated = x.*temp.'; % modulation is a shift-varying operation
soundsc(xModulated,Fs);
figure();
spectrogram(xModulated,L,128,nfft,Fs,'yaxis');
% show modulation of the time-domain signal
figure;
subplot(211); hold on
plot(t,x); xlabel('t');
closeup = (80000:84000); plot(t(closeup),x(closeup),'r');
title('Time Signal'); axis tight
subplot(212)
plot(t(closeup),x(closeup),'b'); hold on
plot(t(closeup),xModulated(closeup),'g'); legend('x','xModulated')
title('{\color{red} Segment} of xModulated[n] vs. orginal'); xlabel('t');
axis tight;
```

4 Homework 4 - Due 10/26/2015 at 5:00 PM

1. Consider a causal system defined by the following difference equation

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

   (a) Write a function y=syseqn(x,yn1,a) that computes the response of the system. The input variable x contains $x[n]$ for $0 \le n \le N-1$, where N is the length of input. The input variable $yn1$ supplies the values of y[-1]. The output variable y contains $y[n]$ for $0 \le n \le N-1$.

   (b) Assume $a = 1, y[-1] = 0$. Use the function you write in (a) to compute the system responses to the discrete-time impulse, $x_1[n] = \delta[n]$, and the unit step function, $x_2[n] = u[n]$, respectively, for $0 \le n \le 30$. Plot each response using the `stem` command.

   (c) Assume $a = 1$, $y[-1] = -1$. Use the function you write in a) to compute the outputs $y_1[n]$ and $y_2[n]$ over $0 \le n \le 30$, when the inputs are $x_1[n] = u[n]$ and $x_2[n] = 3u[n]$, respectively. Plot $y_1[n]$, $y_2[n]$, and $3y_1[n] - y_2[n]$ using the `stem` command. Is $3y_1[n] - y_2[n]$ equal to zero? Why?

   (d) When is the system BIBO stable? Assume $a = \frac{1}{5}$ and the input is the unit step function $x[n] = u[n]$. For the two initial states $y[-1] = 0$ and $y[-1] = 3$, compute the outputs $y[n]$, respectively, for $0 \le n \le 30$. Using the `stem` command to display both responses. How do they differ?

2. Consider two causal systems defined by the following linear difference equations:

$$\text{System 1: } y_1[n] = \tfrac{2}{5}y_1[n-1] + x[n]$$
$$\text{System 2: } y_2[n] = (\tfrac{2}{5})^n y_2[n-1] + x[n]$$

   Each system satisfies initial rest conditions, which states that if $x[n] = 0$ for $n \le 0$ then $y[n] = 0$ for $n \le 0$.

   (a) Let $h_1[n]$ and $h_2[n]$ be the system response to the discrete-time impulse $\delta[n]$ of system 1 and 2, respectively. Calculate $h_1[n]$ and $h_2[n]$ for $0 \le n \le 19$. Plot $h_1[n]$ and $h_2[n]$ using the `stem` command.

   (b) Let $s_1[n]$ and $s_2[n]$ be the system response to the unit step function $u[n]$ of system 1 and 2, respectively. Calculate $s_1[n]$ and $s_2[n]$ for $0 \le n \le 19$ using the system equations. Plot $s_1[n]$ and $s_2[n]$ using the `stem` command.

   (c) Note that $h_1[n]$ and $h_2[n]$ can be treated as zero for $n \ge 20$ for all practical purposes. Define $z_1[n] = h_1[n] * u[n]$ and $z_2[n] = h_2[n] * u[n]$. Calculate $z_1[n]$ and $z_2[n]$ for $0 \le n \le 19$ using the convolution. Plot $z_1[n]$ and $z_2[n]$ using the `stem` command.

   (d) Plot $s_1[n]$ and $z_1[n]$ on the same set of axes. On a different graph, plot $s_2[n]$ and $z_2[n]$. In one case they are the same, and in the other they are not. What is it about System 2 that causes this discrepancy?

3. (a) Consider the following system

$$y[n] = \textstyle\sum_{m=0}^{M} b_m x[n-m],$$

   where $\{b_m\}_{m=0}^{10}$={-0.0872, 0.0370, 0.0985, 0.1755, 0.2388, 0.2632, 0.2388, 0.1755, 0.0985, 0.0370,-0.0872}. Assume the system satisfies the initial resting conditions. Is the system causal, linear, shift-invariant, or BIBO stable? Calculate and plot the system outputs $y_1[n]$

and $y_2[n]$ over $0 \le n \le 31$ when the inputs are $x_1[n] = \cos(\frac{\pi n}{10})$ and $x_2[n] = \cos(\frac{3\pi n}{10})$ respectively. Compute and plot the DFT spectra of the inputs $x_1[n]$ and $x_2[n]$ and the outputs $y_1[n]$ and $y_2[n]$. Comparing the DFT spectra of the inputs with the DFT spectra of the outputs, how does the DFT magnitude of signal change, and how does the frequency content of the signal change?

(b) Consider the following system

$$y[n] = x[n]\cos(\tfrac{3\pi n}{5}).$$

Assume the system satisfies the initial resting conditions. Is the system causal, linear, shift-invariant, or BIBO stable? Calculate and plot the system outputs $y_1[n]$ and $y_2[n]$ over $0 \le n \le 31$, when the inputs are $x_1[n] = \cos(\frac{\pi n}{10})$ and $x_2[n] = cos(\frac{3\pi n}{10})$ respectively. Compute and plot the DFT spectra of the inputs $x_1[n]$ and $x_2[n]$ and the outputs $y_1[n]$ and $y_2[n]$. Comparing the DFT spectra of the inputs with the DFT spectra of the outputs, how does the DFT magnitude of signal change, and how does the frequency content of the signal change?

4. Given the following LSI systems

   (a) $H(z) = \frac{(z-0.5)(z+5)}{(z^2+\frac{1}{16})(z-0.3)}$, causal

   (b) $H(z) = \frac{z}{z^3+z^2-4.75z+3.75}$, causal

   (c) $H(z) = \frac{z^2+3z+1}{(z+e^{-\frac{j\pi}{3}})(z+e^{\frac{j\pi}{3}})(z+e^{-\frac{j2\pi}{3}})(z+e^{\frac{j2\pi}{3}})}$, causal

   For each case, plot the pole-zero diagram to determine whether the system is BIBO stable, calculate and plot the unit impulse response $h[n]$,for $0 \le n \le 63$, and calculate $\sum_{n=0}^{N} |h[n]|$, for $N = 0, 1, \cdots, 63$. If the system is not BIBO stable, determine a bounded input that can generate an unbounded output; for the specific input signal you choose, plot the system response $y[n], n = 0, 1, \ldots, 63$.

5. More operations on the flute signal

   (a) You hear the flute from across a muffled room, which has filtered the signal based on its acoustics. Let's try to recover the signal, assuming the effect is LSI! The room's LSI system has a rational transfer function H(z) given by

$$H(z) = \frac{1 - 2a\cos(\omega_0) + a^2}{(z - ae^{-j\omega_0})(z - ae^{j\omega_0})}$$

   Plot the spectrogram of the filtered signal for $a = 0.99$ and $\omega_0 = 2\pi(880)/F_s$. Then, calculate the inverse filter, and plot the spectrogram of the recovered signal. *HINT: Once you know the a's and b's, you don't really have to do much calculation to perform the inverse filter operation!*

   (b) Musically, this scale contains the notes [A,B,C#,D,E,A] with fundamental frequencies corresponding to about $f_0 = [440, 494, 554, 587, 659, 880]$ Hz. Try filtering this signal with the unstable system (a special case of the above system where $a = 1$)

$$H(z) = \frac{2 - 2\cos(\omega_0)}{(z - e^{-j\omega_0})(z - e^{j\omega_0})}$$

   for values of $\omega_0 = 2\pi f_0/F_s$. Plot spectrograms of the filtered signal for $f_0 = [440, 659]$. Listen to the signals - can you hear which note excites the pole frequency in each case?