GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL of ELECTRICAL and COMPUTER ENGINEERING

ECE 2025 Fall 1999
Problem Set #5

Assigned: 10 September 1999
Due Date: 17 September 1999 (FRIDAY)

Problem 5.1

(a) We have four frequencies of $\hat{\omega}_1 = 0$, $\hat{\omega}_2 = 0.5\pi$, $\hat{\omega}_3 = 0.8\pi$, and $\hat{\omega}_4 = 1.2\pi$, and the corresponding complex amplitudes are $X_1 = 1$, $X_2 = -j2$, $X_3 = j$, and $X_4 = 1 - j = \sqrt{2}e^{-j\pi/4}$. The `makedcos()` function implements the following mathematical equation:

$$x[n] = \sum_{k=1}^{N} \Re e\left\{X_k e^{j\hat{\omega}_k n}\right\} = \sum_{k=1}^{N} A_k \cos(\hat{\omega}_k + \phi_k)$$

where $A_k = |X_k|$ and $\phi_k = \angle X_k$.

Now for some *folding*: since $\cos(1.2\pi n - \pi/4) = \cos(-0.8\pi n - \pi/4) = \cos(0.8\pi n + \pi/4)$, we get

$$\begin{aligned} x[n] &= 1 + 2\cos(0.5\pi n - \pi/2) + \cos(0.8\pi n + \pi/2) + \sqrt{2}\cos(1.2\pi n - \pi/4) \\ &= 1 + 2\cos(0.5\pi n - \pi/2) + \cos(0.8\pi n + \pi/2) + \sqrt{2}\cos(0.8\pi n + \pi/4) \\ &= 1 + 2\cos(0.5\pi n - \pi/2) + \sqrt{5}\cos(0.8\pi n + 0.352\pi) \end{aligned}$$

where the last line results from a "phasor addition" of the two cosines that both have a frequency of $\hat{\omega} = 0.8\pi$ (after the folding). We must add X_3 and the conjugate of X_4 because X_4^* was the complex amplitude at $\hat{\omega} = -1.2\pi$ which is then mapped to $\hat{\omega} = 0.8\pi$ because we add 2π to $\hat{\omega} = -1.2\pi$. Thus, we add complex amplitudes and convert to polar form:

$$X_3 + X_4^* = j + (1-j)^* = j + 1 + j = 1 + j2 = \sqrt{5}e^{j0.352\pi}$$

Problem 5.2

Sampling rate is $f_s = 2000$ Hz. We have to convert the $\hat{\omega}$ frequencies back to continuous-time frequencies in hertz, using the conversion:

$$f = \frac{\hat{\omega}}{2\pi} f_s$$

Thus we get: $f_1 = 0$ which is the DC component, $f_2 = (0.5\pi/2\pi)(2000)$ Hz = 500 Hz, and $f_3 = f_4 = (0.8\pi/2\pi)(2000)$ Hz = 800 Hz.

(b) The formula for the continuous-time output signal $x(t)$ has the same magnitudes and phases as $x[n]$, just different frequencies:

$$x(t) = 1 + 2\cos(2\pi(500)t - \pi/2) + \sqrt{5}\cos(2\pi(800)t + 0.352\pi)$$

Notice that the output signal contains no frequencies greater than 1000 Hz which is half the sampling frequency.

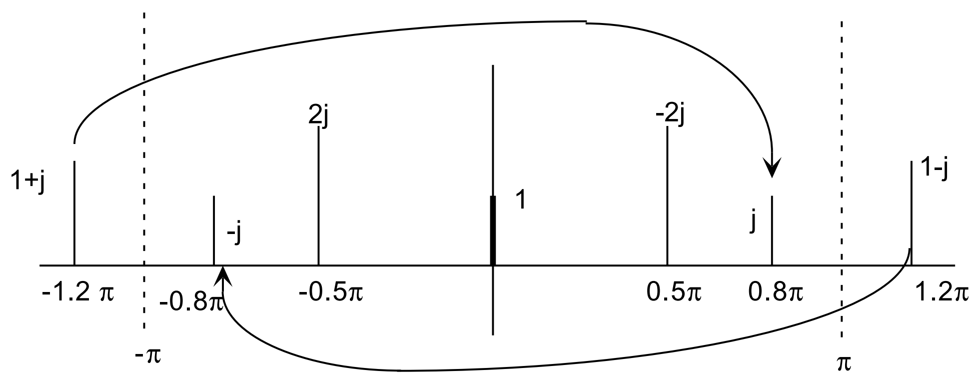
(c) Number of Samples = 100001, but we can say that the range of n is $n = 0$ to $n = 100,000$. Since the sampling rate = 2000 samples/sec., we can convert n to t via the equation:

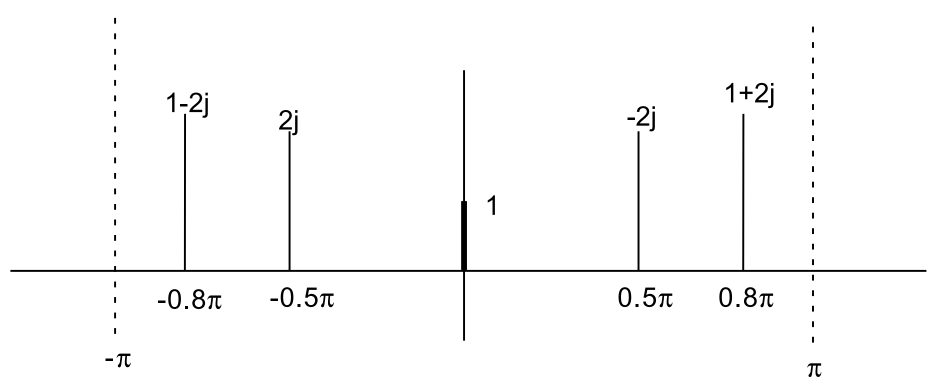
$$t = n/f_s$$

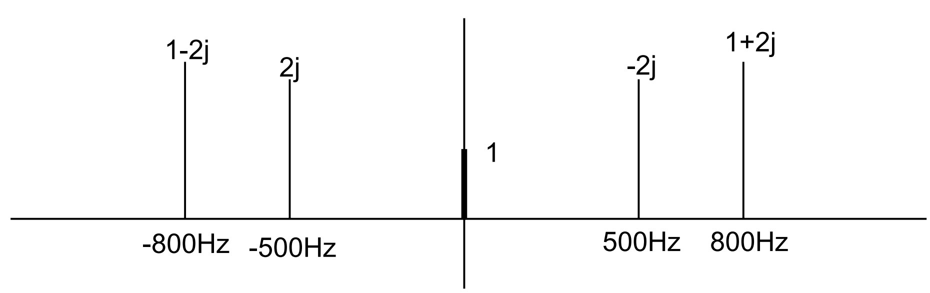
Thus $x(t)$ starts at $t = 0$ and ends at $t = (100,000)/f_s$, and the duration is

$$\frac{100000 \text{ samples}}{2000 \text{ samples/sec}} = 50\,s$$

The following plots solve 5.1(b) and 5.2(a)

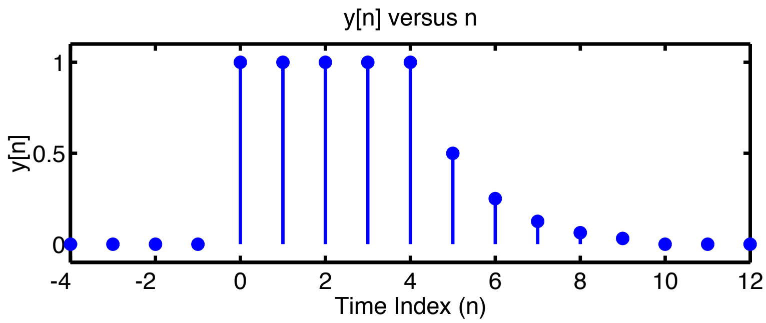






Problem 5.3

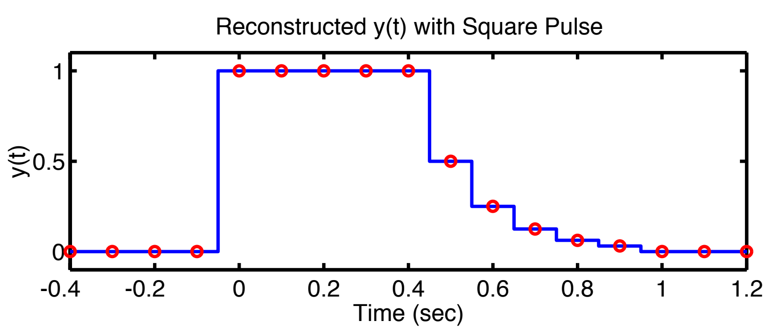
(a)



Code used:

```
nn = -4:12;
yn = [zeros(1,4), ones(1,5), (0.5).^(1:5), zeros(1,3)];
subplot('position',[0.2,0.6,0.7,0.3])
stem( nn, yn, 'filled' )
xlabel('Time Index (n)');ylabel('y[n]');
title('y[n] versus n')
axis([-4 12 -0.1 1.1]);
```

(b) Since $T_s = 0.1s$, the pulse shape, $p(t)$ goes from halfway before each value to halfway after each value. Using this midpoint approach typically gives a cleaner solution than chosing either endpoint.

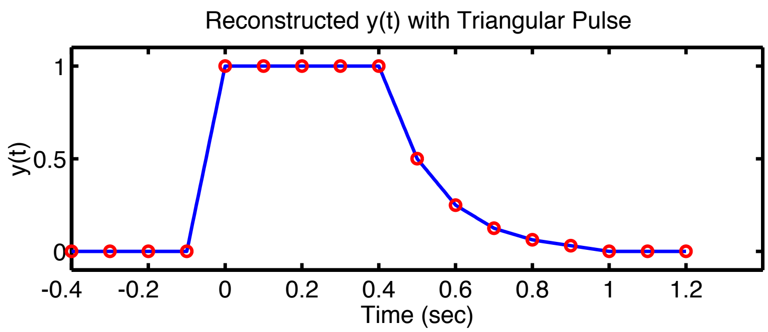


Code to generate this plot:

```
nn = -4:12;
yn = [zeros(1,4), ones(1,5), (0.5).^(1:5), zeros(1,3)];
for i=1:17;
  xx1(2*i-1) = nn(i)-0.5;
  xx1(2*i) = nn(i)+0.5;
```

```
  yy1(2*i-1) = yn(i);
  yy1(2*i) = yn(i);
end;
subplot('position',[0.2,0.6,0.7,0.3])
plot(xx1*0.1,yy1,'b-',nn*0.1,yn,'ro');
axis([-0.4 1.2 -0.1 1.1]);
xlabel('Time (sec)');ylabel('y(t)');
title('Reconstructed y(t) with Square Pulse')
```

(c) For this next pulse shape, we interpolate with a triangle waveform that stretches between the the previous and the next samples, with its peak directly at the sample value.



Code to generate this plot:

```
dt = 0.05;
ttt = (-1:dt:1)*0.1;
Lt = length(ttt);
jkl = 0:Lt-1;
pulse = interp1( [-0.1,0,0.1], [0,1,0], ttt );
nn = -4:12;
yn = [zeros(1,4), ones(1,5), (0.5).^(1:5), zeros(1,3)];
tt = -4:dt:12;
yt = zeros(size(tt));
for i=2:length(nn)-1;
  nstart = (i-2)*(1/dt) + 1;
  yt(nstart+jkl) = yt(nstart+jkl) + yn(i)*pulse;  %<<-- add in one triangular pulse
end;
plot(tt*0.1,yt,'b-',nn*0.1,yn,'ro');
axis([-.4 1.4 -0.1 1.1]);
xlabel('Time (sec)');ylabel('y(t)');
title('Reconstructed y(t) with Triangular Pulse')
```

Problem 5.4

We need to compute $h_1[n] * h_2[n]$, and we start with $h_2[n] = \delta[n] - \beta\delta[n-1]$. Therefore,

$$\begin{aligned} h_1[n] * h_2[n] &= h_1[n] * (\delta[n] - (\beta)\delta[n-1]) \\ &= h_1[n] * \delta[n] - (\beta)h_1[n] * \delta[n-1] \;=\; h_1[n] - \beta h_1[n-1] \end{aligned}$$

We get the following partial expressions as

$$\begin{aligned} h_1[n] &= \sum_{k=0}^{6} \beta^k \delta[n-k] \\ \beta h_1[n-1] &= \sum_{k=0}^{6} \beta^{k+1} \delta[n-k-1] \end{aligned}$$

which results in

$$\begin{aligned} h_1[n] * h_2[n] &= \sum_{k=0}^{6} \beta^k \delta[n-k] - \sum_{k=0}^{6} \beta^{k+1} \delta[n-k-1] \\ &= \sum_{k=0}^{6} \beta^k \delta[n-k] - \sum_{k=1}^{7} \beta^k \delta[n-k] \\ &= \delta[n] + \left(\sum_{k=1}^{6} \beta^k \delta[n-k] - \sum_{k=1}^{6} \beta^k \delta[n-k] \right) - \beta^7 \delta[n-7] \\ &= \delta[n] - \beta^7 \delta[n-7] \end{aligned}$$

(b) Now we can use convolution to write a general expression that relates $y[n]$ to $x[n]$:

$$\begin{aligned} y[n] &= (h_1[n] * h_2[n]) * x[n] \\ &= (\delta[n] - \beta^7 \delta[n-7]) * x[n] \\ &= \delta[n] * x[n] - \beta^7 \delta[n-7] * x[n] \\ &= x[n] - \beta^7 x[n-7] \end{aligned}$$

If $\beta = 0.5$, then $\boxed{y[n] = x[n] - 0.0078125x[n-7].}$

Problem 5.5

(a) $y[n] = x[n]\cos(0.3\pi n)$

Linear: because scaling $x[n]$ by α will give the same scaling of the output:

$$y[n] = (\alpha x[n])\cos(0.3\pi n) = \alpha(x[n]\cos(0.3\pi n))$$

Also, the *superposition* property holds:

$$y[n] = (x_1[n] + x_2[n])\cos(0.3\pi n) = x_1[n]\cos(0.3\pi n) + x_2[n]\cos(0.3\pi n)$$

Causal: because $y[n]$ only depends on the current value of $x[n]$.

Not Time-Invariant: because we can make the following counter-example. Let $x[n] = \delta[n]$ so that the output is

$$y[n] = x[n]\cos(0.3\pi n) = \delta[n]\cos(0.3\pi n) = \delta[n]\cos(0.3\pi(0)) = \delta[n]$$

Now change the input to $x[n] = \delta[n-1]$, so that we expect the output to shift by 1 time index. However, the output is actually

$$y[n] = x[n]\cos(0.3\pi n) = \delta[n-1]\cos(0.3\pi n) = \delta[n-1]\cos(0.3\pi(1)) = 0.588\delta[n-1]$$

(b) $y[n] = |x[-n]|$

Not causal: $y[-1] = x[1]$, so $y[n]$ at $n = -1$ depends on a future value of $x[n]$ at $n = +1$.

Not linear: because when we multiply the input by -3, the output does not get multiplied by -3. Here is the equation for the output $y[n] = |-3(x[-n])|$ which equals $y[n] = 3|x[-n]|$.

Not Time-Invariant: because we can show the following counter-example: when $x[n] = \delta[n]$, the output is $y[n] = |x[-n]| = |\delta[-n]| = \delta[n]$. However, when we shift the input to $n = 1$ by using the input $\delta[n-1]$, the output does *not* shift by the same amount: $y[n] = |x[-n]| = |\delta[-n-1]| = \delta[n+1]$.

Problem 5.6

$$x[n] = \sum_{k=0}^{6} \beta^k \delta[n-k]$$

$$y[n] = x[n] - \beta x[n-1]$$

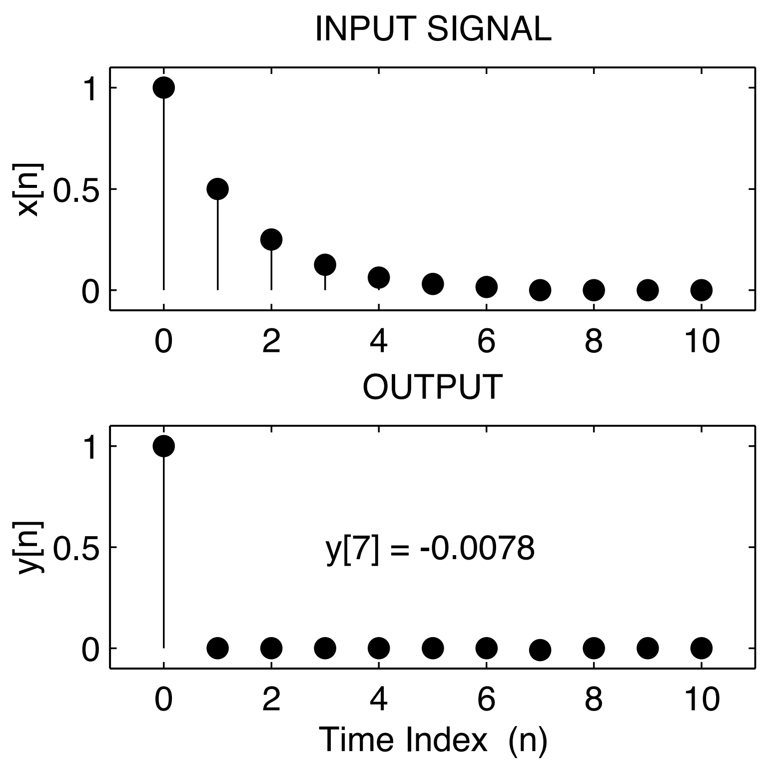
(a) This problem is similar in computation to Problem 5.3.

$$\begin{aligned}
y[n] &= x[n] - \beta x[n-1] \\
&= \sum_{k=0}^{6} \beta^k \delta[n-k] - \beta \sum_{k=0}^{6} \beta^k \delta[n-1-k] \\
&= \sum_{k=0}^{6} \beta^k \delta[n-k] - \beta \sum_{k=1}^{7} \beta^{k-1} \delta[n-k] \\
&= \delta[n] + \left(\sum_{k=1}^{6} \beta^k \delta[n-k] - \sum_{k=1}^{6} \beta\beta^{k-1} \delta[n-k] \right) - \beta^7 \delta[n-7] \\
&= \delta[n] - \beta^7 \delta[n-7]
\end{aligned}$$

Using another approach, we can make a table that contains the values for $x[n]$ and $\beta x[n-1]$ and then subtract the results to get $y[n] = x[n] - \beta x[n-1]$:

	$n=0$	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	$n=7$	$n=8$	$n=9$	$n=10$
$x[n]$	1	β	β^2	β^3	β^4	β^5	β^6	0	0	0	0
$\beta x[n]$	0	β	β^2	β^3	β^4	β^5	β^6	β^7	0	0	0
$y[n]$	1	0	0	0	0	0	0	$-\beta^7$	0	0	0

(b) We plot $x[n]$ and $y[n]$ for the case of $\beta = 0.5$, but notice that β^7 is a very small number so it hardly shows up on the plot of the output signal.

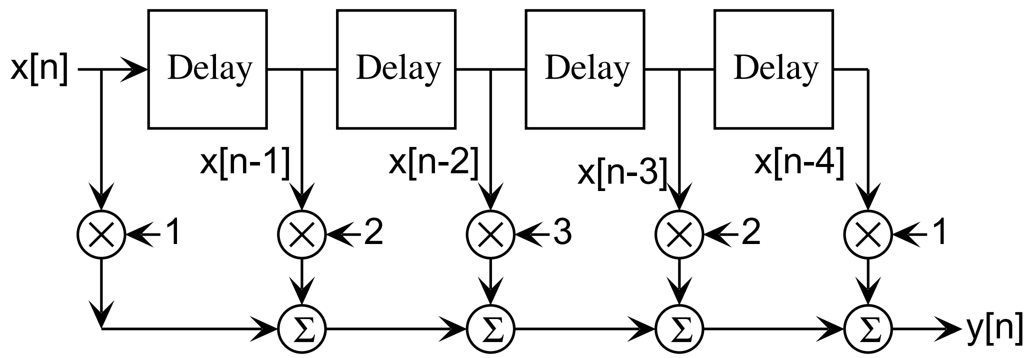


Code to generate this plot:

```
n = [0:10];
beta = 0.5;
x = [ beta .^(0:6), zeros(1,4) ];
y = conv(x,[1,-beta]);
y = y(1:length(n));
subplot(4,2,1),stem(n,x,'filled')
axis([-1 11 -0.1 1.1]);
ylabel('x[n]');
title('INPUT SIGNAL')
h2 = subplot(4,2,3),stem(n,y,'filled')
axis([-1 11 -0.1 1.1]);
xlabel('Time Index  (n)'); ylabel('y[n]');
title('OUTPUT')
text(3,0.5,'y[7] = -0.0078')
h2p = get(h2,'position')
h2p(2) = h2p(2) - 0.025;
set(h2,'position',h2p)
```

Problem 5.7

(a)



(b) The impulse response can be obtained from the FIR filtering equation, which in turn can be obtained easily from the structure:

$$y[n] = x[n] + 2x[n-1] + 3x[n-2] + 2x[n-3] + x[n-4]$$

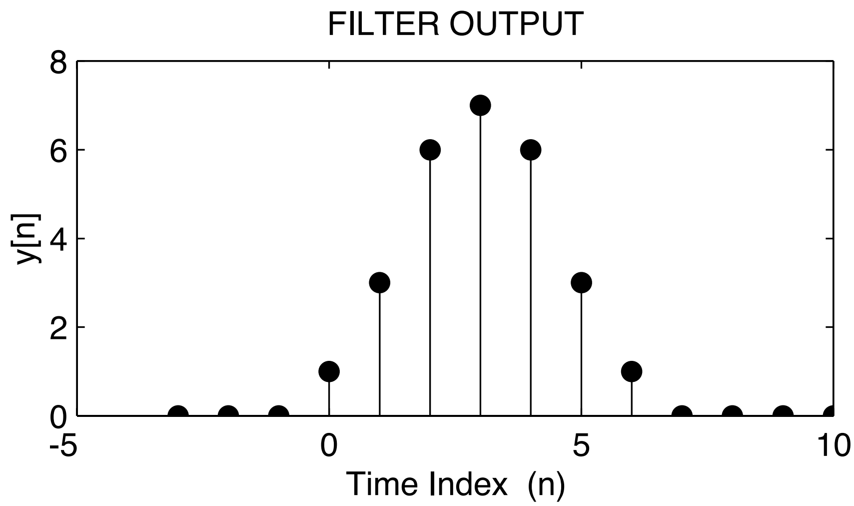
Thus when we let the input signal be the unit-impulse signal, i.e., $x[n] = \delta[n]$ we get the impulse response which is denoted by $h[n]$.

$$h[n] = \delta[n] + 2\delta[n-1] + 3\delta[n-2] + 2\delta[n-3] + \delta[n-4]$$

(c) The easiest method is to construct a table, and add across the rows to get the output values:

n	$x[n]$	$2x[n-1]$	$3x[n-2]$	$2x[n-3]$	$x[n-4]$	$y[n]$
-3	0	0	0	0	0	0
-2	0	0	0	0	0	0
-1	0	0	0	0	0	0
0	1	0	0	0	0	1
1	1	2	0	0	0	3
2	1	2	3	0	0	6
3	0	2	3	2	0	7
4	0	0	3	2	1	6
5	0	0	0	2	1	3
6	0	0	0	0	1	1
7	0	0	0	0	0	0
8	0	0	0	0	0	0
9	0	0	0	0	0	0
10	0	0	0	0	0	0

We plot the output response as:

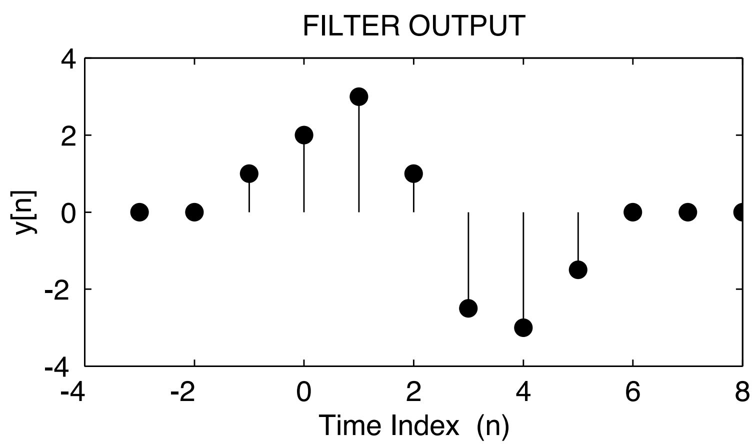


The code to generate this plot:

```
hh = [1, 2, 3, 2, 1];
xx = [1, 1, 1];
yy = conv( xx, hh );
nn = -3:10;
yout = zeros(size(nn));
nz = find(nn==0);
yout(nz+(0:length(yy)-1)) = yy;
subplot(3,1,1)
stem( nn, yout, 'filled' )
axis([-3 6 -1.1 1.1]);
xlabel('Sample');ylabel('y[n]');
```

Problem 5.8

(a)



If we know the response when the input is $x_1[n] = \delta[n-1]$ and we call that output $y_1[n]$, then we can use time-invariance to find the output for $x_0[n] = \delta[n]$ and $x_2[n] = \delta[n-2]$.

$$\begin{aligned} x_1[n] &= \delta[n-1] \quad \rightarrow \quad y_1[n] \\ x_2[n] &= \delta[n-2] \quad \rightarrow \quad y_1[n-1] \quad \text{(because we have delayed the input)} \\ x_0[n] &= \delta[n] \quad \rightarrow \quad y_1[n+1] \end{aligned}$$

Next we can use linearity to find the response due to the combined input $\delta[n] - \delta[n-2]$. The answer is that we must use the same combination of the outputs:

$$y[n] = y_1[n+1] - y_1[n-1]$$

The figure above shows the output signal $y[n] = y_1[n+1] - y_1[n-1]$. Here is a table of values used to construct the output.

	$n=-2$	$n=-1$	$n=0$	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=6$	$n=7$
$y_1[n]$	0	0	1	2	4	3	1.5	0	0	0
$y_1[n+1]$	0	1	2	4	3	1.5	0	0	0	0
$y_1[n-1]$	0	0	0	1	2	4	3	1.5	0	0
$y[n]$	0	1	2	3	1	-2.5	-3	-1.5	0	0

(b) This system is not causal. Because when we use the input $\delta[n-1]$ the input starts at $n=1$, but the corresponding output is $y_1[n]$ which starts at $n=0$. In other words, the output starts before the input—*not causal*.

So one way to look for causality is to note that a system is *causal* when the output response occurs at the start of an impulse or after the impulse. A non-causal system has an output response that occurs before the start of an impulse.

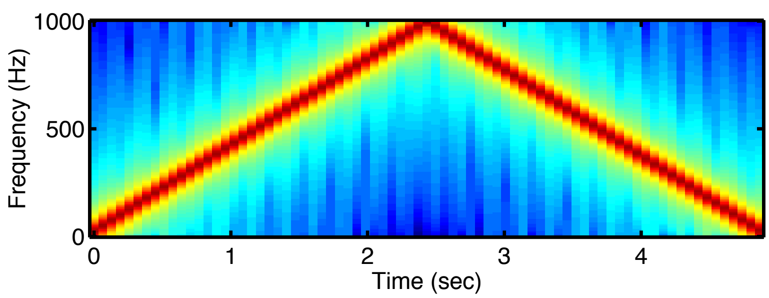
Problem 5.9

(a) When $f_s = 1000$ Hz and the input frequency is 700 Hz, we have a folded alias. The output frequency will be 300 Hz. The $\hat{\omega}$ frequency is $\hat{\omega} = 2\pi f/f_s = 2\pi(700)/1000 = 1.4\pi$, but when we subtract 2π, we find another frequency at $\hat{\omega} = -0.6\pi$. Likewise there will be a positive frequency component at $\hat{\omega} = +0.6\pi$. This frequency maps back to the analog frequency $f = (\hat{\omega}/2\pi) f_s = (0.6\pi/2\pi)1000 = 300$ Hz.

(b) To find the instantaneous frequency, we take the time-derivative of

$$\psi(t) = 400\pi t^2 \qquad \Rightarrow \qquad \frac{d}{dt}\psi(t) = 800\pi t$$

In hertz, the instantaneous frequency is $f_i(t) = 400t$. As time goies from $t = 0$ to $t = 5$, we think that $f_i(t)$ will go from 0 Hz to 2000 Hz, but with a sampling frequency of $f_s = 2000$ Hz, but the output reconstruction by the D-to-A converter cannot produce frequencies about $\frac{1}{2} f_s = 1000$ Hz. Here is the spectrogram produced by MATLAB.




```
fs = 2000;
tt = 0:(1/fs):5;
xx = cos(400*pi*tt.*tt);
specgram(xx,[],fs)
ylabel('Frequency (Hz)')
xlabel('Time (sec)')
```