Solution Set for Homework #5

By Prof. Brian Evans and Mr. Firas Tabbara

PROBLEM 1: AVERAGING FILTER AND THE UNIT STEP SIGNAL

Prologue: The unit step signal models a physical action such as turning on a switch and leaving it on indefinitely or suddenly hitting the brakes and holding the brakes indefinitely. We represent "on" with an amplitude of 1 and "off" with an amplitude of 0. In control systems, such as a braking system, we evaluate the response of the control system to the unit step signal to see how quickly the output reaches and holds a value of "on". That is, the control system response time would be how quickly the braking system fully engages the mechanical brakes.

MATLAB has a function that is like the discrete-time unit step called `heaviside`. The difference is that at the origin, the `heaviside` function has value of 1/2 instead of 1. Please don't use the `heaviside` function for the discrete-time unit step function. MATLAB had a function to implement the discrete-time unit step function u[n] called `stepfun` but it is obsolete. We can mimic $u[n]$ by using comparison operations in MATLAB. In MATLAB, a comparison operation will return 1 if true and 0 if false.
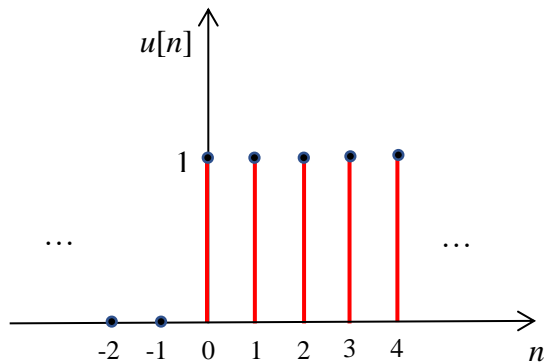
Problem: *Signal Processing First*, Problem P-5.6, page 127.

The unit step signal "turns on" at $n = 0$ and is usually denoted by $u[n]$.
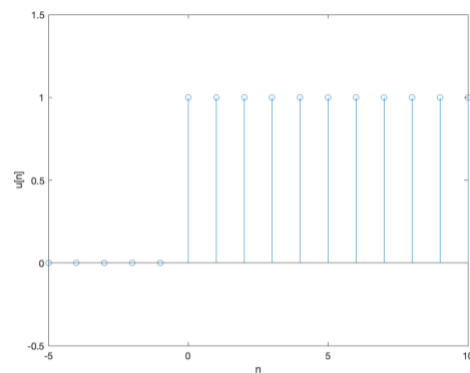
Part (a): Make a plot of $u[n]$.

Solution for (a):

$$u[n] = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases}$$

```
n = -5:10;
unitstep = ( n >= 0 );
stem(n, unitstep)
xlabel('n')
ylabel('u[n]')
ylim([-0.5 1.5])
```
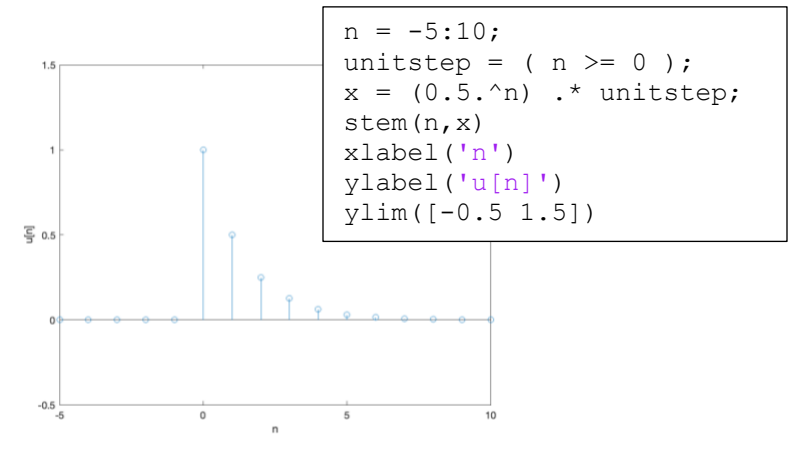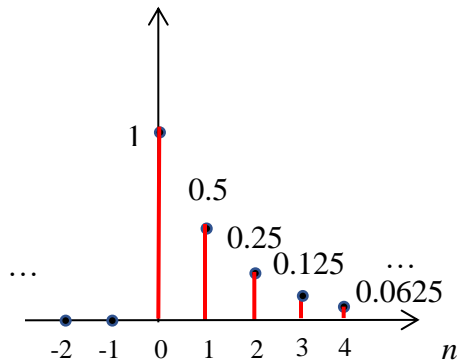

Part (b): We can use the unit step signal to represent other sequences that are zero for $n < 0$. Plot the sequence $x[n] = (0.5)^n u[n]$.

Solution for (b): Since the unit step signal is 0 for $n < 0$, $x[n]$ for $n < 0$ is equal to 0. For $n \geq 0$,

$n = 0$: $x[0] = (0.5)^0\, u[0] = 1$ $n = 3$: $x[3] = (0.5)^3\, u[3] = 0.125$
$n = 1$: $x[1] = (0.5)^1\, u[1] = 0.5$ $n = 4$: $x[4] = (0.5)^4\, u[4] = 0.0625$
$n = 2$: $x[2] = (0.5)^2\, u[2] = 0.25$ etc.

```
n = -5:10;
unitstep = ( n >= 0 );
x = (0.5.^n) .* unitstep;
stem(n,x)
xlabel('n')
ylabel('u[n]')
ylim([-0.5 1.5])
```

Part (c): The *L*-point running average is defined as

$$y[n] = \frac{1}{L} \sum_{k=0}^{L-1} x[n-k]$$

For the input sequence $x[n] = (0.5)^n\, u[n]$, compute the numerical value of $y[n]$ over the index range $-5 \le n \le 10$, assuming $L = 4$.

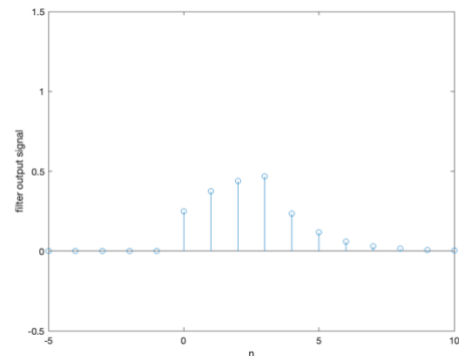**Solution for (c):** We can expand the calculation for the four-point averaging filter:

$$y[n] = \frac{1}{L} \sum_{k=0}^{L-1} x[n-k] = \frac{1}{4} \sum_{k=0}^{3} x[n-k] = \frac{1}{4}(x[n] + x[n-1] + x[n-2] + x[n-3])$$

We can use Matlab or calculate the answers by hand.

```
averagingFilterCoeffs = [ 1/4, 1/4, 1/4, 1/4 ];
y = filter(averagingFilterCoeffs, 1, x);
figure;
stem(n, y);
xlabel('n');
ylabel('filter output signal');
ylim([-0.5 1.5]);
```


In the plot, the output signal is zero when n < 0. The output signal for 0 <= n <= 2 corresponds to a partial response by the filter to the change in the input signal at the origin. Once we reach n = 3, the sliding window of input samples would be filled.

For $n < 0$, terms $x[n], x[n-1], x[n-2],$ and $x[n-3]$ will always be zero, and hence $y[n] = 0$.

For $n = 0$: $y[0] = \frac{1}{4}\{x[0] + x[-1] + x[-2] + x[-3]\} = \frac{1}{4}\{(0.5)^0 + 0 + 0 + 0\} = \frac{1}{4}$

For $n = 1$: $y[1] = \frac{1}{4}\{x[1] + x[0] + x[-1] + x[-2]\} = \frac{1}{4}\{(0.5)^1 + (0.5)^0 + 0 + 0\} = \frac{3}{8}$

For $n = 2$: $y[2] = \frac{1}{4}\{x[2] + x[1] + x[0] + x[-1]\} = \frac{1}{4}\{(0.5)^2 + (0.5)^1 + (0.5)^0 + 0\} = \frac{7}{16}$

For $n = 3$: $y[3] = \frac{1}{4}\{x[3] + x[2] + x[1] + x[0]\} = \frac{1}{4}\{(0.5)^3 + (0.5)^2 + (0.5)^1 + (0.5)^0\} = \frac{15}{32}$

At this point, when $n = L$-1, the sliding window of L samples is now filled with values of $x[n]$ for $n \geq 0$.

For $n = 4$: $y[4] = \frac{1}{4}\{x[4] + x[3] + x[2] + x[1]\} = \frac{1}{4}\{(0.5)^4 + (0.5)^3 + (0.5)^2 + (0.5)^1\} = \frac{15}{64}$

For $n = 5$: $y[5] = \frac{1}{4}\{x[5] + x[4] + x[3] + x[2]\} = \frac{1}{4}\{(0.5)^5 + (0.5)^4 + (0.5)^3 + (0.5)^2\} = \frac{15}{128}$

For $n = 6$: $y[6] = \frac{1}{4}\{x[6] + x[5] + x[4] + x[3]\} = \frac{1}{4}\{(0.5)^6 + (0.5)^5 + (0.5)^4 + (0.5)^3\} = \frac{15}{256}$

For $n = 7$: $y[7] = \frac{1}{4}\{x[7] + x[6] + x[5] + x[4]\} = \frac{1}{4}\{(0.5)^7 + (0.5)^6 + (0.5)^5 + (0.5)^4\} = \frac{15}{512}$

For $n = 8$: $y[8] = \frac{1}{4}\{x[8] + x[7] + x[6] + x[5]\} = \frac{1}{4}\{(0.5)^8 + (0.5)^7 + (0.5)^6 + (0.5)^5\} = \frac{15}{1024}$

For $n = 9$: $y[9] = \frac{1}{4}\{x[9] + x[8] + x[7] + x[6]\} = \frac{1}{4}\{(0.5)^9 + (0.5)^8 + (0.5)^7 + (0.5)^6\} = \frac{15}{2048}$

For $n = 10$: $y[10] = \frac{1}{4}\{x[10] + x[9] + x[8] + x[7]\} = \frac{1}{4}\{(0.5)^{10} + (0.5)^9 + (0.5)^8 + (0.5)^7\} = \frac{15}{4096}$

Part (d): For the input sequence $x[n] = a^n u[n]$, derive a general formula for $y[n]$ that will apply for any value a, for any length L, and for the index range $n \geq 0$. In doing so, you may have use for the formula:

$$\sum_{k=M}^{N} \alpha^k = \frac{\alpha^M - \alpha^{N+1}}{1 - \alpha}$$

Solution for (d):

$x[n] = a^n u[n]$

$$x[n-k] = \begin{cases} 0, & n < k \\ a^{n-k} & n \geq k \end{cases}$$

$$y[n] = \begin{cases} 0, & n < 0 \\ \dfrac{1}{L}\displaystyle\sum_{k=0}^{n} a^{n-k} = \dfrac{a^n}{L}\displaystyle\sum_{k=0}^{n} a^{-k} = \dfrac{a^n}{L}\left(\dfrac{1 - a^{-(n+1)}}{1 - a^{-1}}\right) = \dfrac{1}{L}\left(\dfrac{a^{n+1} - 1}{a - 1}\right), & 0 \leq n \leq L-1 \\ \dfrac{1}{L}\displaystyle\sum_{k=0}^{L-1} a^{n-k} = \dfrac{a^n}{L}\displaystyle\sum_{k=0}^{L-1} a^{-k} = \dfrac{a^n}{L}\left(\dfrac{1 - a^{-L}}{1 - a^{-1}}\right) = \dfrac{a^{n-L+1}}{L}\left(\dfrac{a^L - 1}{a - 1}\right), & n \geq L \end{cases}$$

$$y[n] = \begin{cases} 0, & n < 0 \\ \dfrac{1}{L}\left(\dfrac{a^{n+1} - 1}{a - 1}\right), & 0 \leq n \leq L-1 \\ \dfrac{a^{n-L+1}}{L}\left(\dfrac{a^L - 1}{a - 1}\right), & n \geq L \end{cases}$$

PROBLEM 2: SYSTEM PROPERTIES

Prologue: System properties indicate what analysis tools are available for their analysis. Our course is entitled *Linear Systems and Signals*. Even though we are emphasizing linear systems in the class, we are covering nonlinear systems, such as the squaring system. In this course, the widest set of analysis tools will be available for linear time-invariant (LTI) systems.

Consider a system that gives output $y_1[n]$ for input $x_1[n]$ and output $y_2[n]$ for input $x_2[n]$.

- *Linearity* means the system satisfies the following two properties
  - *Homogeneity*. For input $\alpha\, x[n]$, where α is any constant, the output $y_{scaled}[n] = \alpha\, y[n]$.
  - *Additivity*: For input $x_1[n] + x_2[n]$, the output is the sum of their individual responses $y_1[n] + y_2[n]$.
- *Time Invariance* means shifting the input by n_0 samples always produces the same shift of n_0 samples in the output, for an integer constant n_0. Mathematically, if the input were $x[n - n_0]$, then the output $y_{shifted}[n] = y[n - n_0]$.
- *Causality* means the system output only depends on the current and previous values of the input and previous values of the output.

This problem assumes all signals are observed for all time, i.e. from $-\infty < n < \infty$.

Problem: For each of the following systems, determine whether or not the system is (1) linear, (2) time-invariant, and (3) causal.

- $y[n] = x[n]\cos(0.2\pi n)$
- $y[n] = x[n] - x[n - 1]$
- $y[n] = |x[n]|$
- $y[n] = A\, x[n] + B$

Solution for (a): $y[n] = x[n]\cos(0.2\pi n)$.

- *Homogeneity*. For input $\alpha\, x[n]$,

$$y_{scaled}[n] = (\alpha\, x[n])\cos(0.2\pi n) = \alpha\, x[n]\cos(0.2\pi n) \overset{??}{=} \alpha\, y[n] \quad \textbf{YES}$$

- *Additivity*: For input $x_1[n] + x_2[n]$,

$$y_{additive}[n] = (x_1[n] + x_2[n])\cos(0.2\pi n) = x_1[n]\cos(0.2\pi n) + x_2[n]\cos(0.2\pi n)$$

$$y_{additive}[n] \overset{??}{=} y_1[n] + y_2[n] \quad \textbf{YES}$$

- *Time Invariance*. For input $x[n - n_0]$,   **NO**

$$y_{shifted}[n] = x[n - n_0]\cos(0.2\pi n) \overset{??}{=} y[n - n_0] \text{ where } y[n - n_0] = x[n - n_0]\cos\big(0.2\pi(n - n_0)\big)$$

- *Causality* means the system output only depends on current and previous values of the input and previous values of the output. For $y[n] = x[n]\cos(0.2\pi n)$, the output only depends only on the current input value. **YES**. It's a *pointwise system*; i.e., the current output point (sample) only depends on the current input point (sample). All pointwise systems are causal. *NOTE:* All pointwise systems are also anti-causal. An anti-causal system processes samples in the opposite direction towards $n = -\infty$ instead of towards $n = \infty$. An anti-causal system depends only on the current and future values of the input and future values of the output.

Solution for (b): $y[n] = x[n] - x[n-1]$. This system takes the current input value and subtracts from it the previous input value. Because all signals are being observed for all time, i.e. from $-\infty < n < \infty$, we don't have to check initial conditions.

- *Homogeneity.* For input $\alpha\, x[n]$,   **YES**

$$y_{scaled}[n] = (\alpha\, x[n]) - (\alpha\, x[n])_{n \to n-1} = \alpha\, x[n] - \alpha\, x[n-1] = \alpha\,(x[n] - x[n-1]) \stackrel{??}{=} \alpha\, y[n]$$

- *Additivity*: For input $x_1[n] + x_2[n]$,

$$y_{additive}[n] = (x_1[n] + x_2[n]) - (x_1[n] + x_2[n])_{n \to n-1} = x_1[n] - x_1[n-1] + x_2[n] - x_2[n-1]$$

$$y_{additive}[n] \stackrel{??}{=} y_1[n] + y_2[n] \quad \textbf{YES}$$

- *Time Invariance.* For input $x[n-n_0]$,   **YES**

$$y_{shifted}[n] = (x[n-n_0]) - (x[n-n_0])_{n \to n-1} = x[n-n_0] - x[n-n_0-1] = y[n-n_0] \text{ where}$$
$$y[n-n_0] = x[n-n_0] - x[n-n_0-1]$$

- *Causality.* The output only depends only on the current and previous input value. **YES**.

Solution for (c): $y[n] = |x[n]|$. The system takes the absolute value of the current input value.

- *Homogeneity.* For input $\alpha\, x[n]$,

$$y_{scaled}[n] = |\alpha\, x[n]| = |\alpha||x[n]| \stackrel{??}{=} \alpha\, y[n] = \alpha\,|x[n]| \quad \textbf{NO}. \text{ Only works for non-negative } \alpha.$$

- *Additivity*: For input $x_1[n] + x_2[n]$,

$$y_{additive}[n] = |x_1[n] + x_2[n]| \stackrel{??}{=} y_1[n] + y_2[n] = |x_1[n]| + |x_2[n]| \quad \textbf{NO}.$$

- *Time Invariance.* For input $x[n-n_0]$,

$$y_{shifted}[n] = |x[n-n_0]| \stackrel{??}{=} y[n-n_0] = |x[n-n_0]| \quad \textbf{YES}.$$

- *Causality*. The output only depends only on the current input value. Pointwise system. **YES**.

Solution for (d): $y[n] = A\, x[n] + B$. The system multiplies the input by the constant A and adds the result to another constant B.

- *Homogeneity.* For input $\alpha\, x[n]$,

$$y_{scaled}[n] = A\,(\alpha\, x[n]) + B = \alpha\, A\, x[n] + B \stackrel{??}{=} \alpha\, y[n] = \alpha\, A\, x[n] + \alpha\, B. \quad \textbf{NO}.$$

- *Additivity*: For input $x_1[n] + x_2[n]$,

$$y_{additive}[n] = A\,(x_1[n] + x_2[n]) + B = A\, x_1[n] + A\, x_2[n] + B \stackrel{??}{=} y_1[n] + y_2[n] \text{ where}$$

$$y_1[n] + y_2[n] = A\, x_1[n] + A\, x_2[n] + 2B. \quad \textbf{NO}.$$

- *Time Invariance.* For input $x[n-n_0]$,

$$y_{shifted}[n] = A\, x[n-n_0] + B \stackrel{??}{=} y[n-n_0] = A\, x[n-n_0] + B \quad \textbf{YES}.$$

- *Causality.* The output only depends only on the current input value. Pointwise system. **YES**

PROBLEM 3: SYSTEM PROPERTIES AND DECONVOLUTION

Prologue: A common problem that arises in audio and other systems is characterizing an unknown system. Consider an audio system with a speaker transmitting sound in a concert hall that is recorded by a microphone, and we'd like to characterize the acoustic response in the concert hall. For this example, if we know the FIR filter coefficients for the linear time-invariant model of the concert hall acoustics, then we can mimic the effect of the concert hall by using its FIR coefficients to filter a music track and the output will sound as if it was played in the concert hall. Several audio systems with a "concert hall" effect will list several different concert halls.

Problem: (a) *Signal Processing First*, problem P-5.12, page 128. When working the problem, assume that you do not know the filter coefficients. Using system properties to find the answer.

For part (b), use the deconvolution formulas in homework problem 4.4(a) to manually determine the FIR filter coefficients given that the input signal is $x_1[n]$ and the output signal is $y_1[n]$ for $n \geq 0$. Check your calculations using `utdeconvolve.m` provided on the [tuneup page](#), which is a modified version of the MATLAB code from homework problem 4.4(c). When running the code, use an equal number of input and output values for vectors x and y, which is what would happen in an experimental setup that would input a test signal x and observe the output y. Please see the comments in the code for more info.

This is an interesting case where the input signal is infinite in duration and the output signal is finite in duration.

Solution for (a): For a particular LTI system, when the input is the unit step, $x_1[n] = u[n]$, the corresponding output is $y_1[n] = \delta[n] + 2\,\delta[n-1] - \delta[n-2]$. This problem aims at determining the output when the input to an LTI system is $x_2[n] = 3u[n] - 2u[n-4]$.

Since the system is Linear Time-Invariant (LTI),

$$x_2[n] = 3u[n] - 2u[n-4] = 3x_1[n] - 2x_1[n-4]$$

For $y_1[n-4]$, assume that the signal is shifted by 4. And since the system is LTI, the output to the input $x_2[n]$ is as follows:

$$y_2[n] = 3y_1[n] - 2y_1[n-4]$$
$$= 3(\delta[n] + 2\delta[n-1] - \delta[n-2]) - 2(\delta[n-4] + 2\delta[n-5] - \delta[n-6])$$
$$= 3\,\delta[n] + 6\,\delta[n-1] - 3\,\delta[n-2] - 2\,\delta[n-4] - 4\,\delta[n-5] + 2\,\delta[n-6]$$

Solution for (b): We'll model the unknown LTI system as a finite impulse response (FIR) filter with input signal $x[n]$ and output signal

$$y[n] = h[0]\,x[n] + h[1]\,x[n-1] + h[2]\,x[n-2] + h[N-1]\,x[n-(N-1)]$$

From this model and knowledge of $x_1[n]$ and $y_1[n]$, we'll compute the filter coefficients $h[n]$ for $n = 0, 1, \dots, N-1$ and verify that the response of the FIR filter is $y_1[n]$.

Deconvolution. We'll use deconvolution to compute the filter coefficients. We derive the time-domain deconvolution algorithm by evaluating the output at $n = 0$:

$$y[0] = b_0\,x[0] + b_1\,x[-1] + b_2\,x[-2] + \cdots + b_{N-1}\,x[-(N-1)]$$

For LTI systems, it is a necessary (but not sufficient) condition for the system to be "at rest", which means that all initial conditions $x[-1], x[-2], \ldots, x[-(N-1)]$ must be zero. Since we know $x[n]$ and $y[n]$, we have one equation and one unknown at $n = 0$:

$$y[0] = b_0\, x[0]$$

and we can compute

$$b_0 = \frac{y[0]}{x[0]}$$

For this calculation to be valid, the first value of the test signal, $x[0]$, cannot be zero.

The second output value is: $y[1] = b_0\, x[1] + b_1\, x[0]$, and therefore, $b_1 = \frac{y[1] - b_0 x[1]}{x[0]}$.

The third output value is: $y[2] = b_0\, x[2] + b_1\, x[1] + b_2\, x[0]$ and $b_2 = \frac{y[2] - b_0 x[2] - b_1 x[1]}{x[0]}$.

In general, $b_N = \frac{y[N] - \sum_{i=0}^{N-1} b_i x[N-i]}{x[0]}$. MATLAB code utdeconvolve.m realizes the algorithm.

Part (a). Give the vectors for x and y that you used when running utdeconvolve.m and the filter coefficients in vector b that the code computes.

x	y	b
[1 1 1]	[1 2 -1]	[1 1 -3]
[1 1 1 1]	[1 2 -1 0]	[1 1 -3 1]
[1 1 1 1 1]	[1 2 -1 0 0]	[1 1 -3 1 0]
[1 1 1 1 1 1]	[1 2 -1 0 0 0]	[1 1 -3 1 0 0]

Part (b). Verify the coefficients by using them in the LTI FIR filter difference equation:

$$y[n] = h[0]\, x[n] + h[1]\, x[n-1] + h[2]\, x[n-2] + h[N-1]\, x[n-(N-1)]$$

given that the input signal is $u[n]$ and the output signal is $y_1[n]$

$$y_1[n] = u[n] + u[n-1] - 3\, u[n-2] + u[n-3]$$

This is a closed-form solution for $y_1[n]$ which gives the correct output values for $y_1[n]$ for all n. For $n \geq 0$, the values are [1 2 -1 0 0 0 ....].

For $y_2[n] = x_2[n] + x_2[n-1] - 3x_2[n-2] + x_2[n-3]$ and $x_2[n] = 3u[n] - 2u[n-4]$:

$y_2[n] = 3u[n] + 3u[n-1] - 9u[n-2] + 3u[n-3] - 2u[n-4] - 2u[n-5] + 6u[n-6] - 2u[n-7]$

This gives values [3 6 -3 0 -2 -4 2 0 0 ...] for $n \geq 0$.

Alternate solution. One could compute several values of $y[n]$:
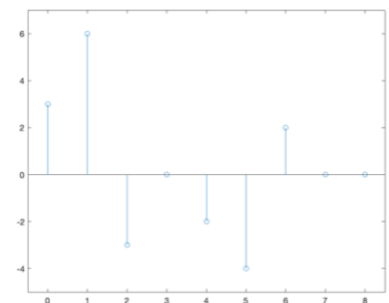
$y[n] = x[n] + x[n-1] - 3\, x[n-2] + x[n-3]$

$y[0] = x[0] + x[-1] - 3\, x[-2] + x[-3] = 1 + 0 + 3 \cdot 0 + 0 = 1$

$y[1] = x[1] + x[0] - 3\, x[-1] + x[-2] = 1 + 1 - 3 \cdot 0 + 0 = 2$

$y[2] = x[2] + x[1] - 3\, x[0] + x[-1] = 1 + 1 - 3 \cdot 1 + 0 = -1$

$y[3] = x[3] + x[2] - 3\, x[1] + x[0] = 1 + 1 - 3 \cdot 1 + 1 = 0$

$y[4] = x[4] + x[3] - 3\, x[2] + x[1] = 1 + 1 - 3 \cdot 1 + 1 = 0$ etc.

```
x2 = [ 3   3   3   3   1   1   1   1   1 ];
h  = [ 1   1  -3   1   0   0   0   0   0 ];
y2 = filter(x2, 1, h);
n = 0 : 8;
stem(n, y2);
xlim( [-0.5, 8.5] ); ylim( [-5 7] );
```

PROBLEM 4. FREQUENCY RESPONSE

Prologue: Linear time-invariant (LTI) systems do not create new frequencies. That is, any frequency in the output signal had to be present in the input signal. Moreover, every frequency in the input signal will experience a scaling in magnitude and shift in phase as it passes through the LTI system. The scaling in amplitude and shift in phase will be a function of the frequency.

Problem: *Signal Processing First*, problem P-6.1, page 157. Suppose the input signal to an FIR system is

$$x[n] = A\, e^{j\,(0.4\pi n - 0.5\pi)}$$

If we defined a new signal $y[n]$ to be the first difference $y[n] = x[n] - x[n-1]$, it is possible to express $y[n]$ in the form

$$y[n] = A\, e^{j\,(\hat{\omega}_0 n + \phi)}$$

Determine the numerical values of A, ϕ and $\hat{\omega}_0$. Please comment on the differences between the input and output signals. What parameter(s) changed? By how much? What parameter(s) didn't change?

Solution: The new signal $y[n]$ is defined as follows:

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

$= e^{j(0.4\pi n - 0.5\pi)} - e^{j(0.4\pi(n-1) - 0.5\pi)}$

$= e^{j(0.4\pi n - 0.5\pi)} - e^{j(0.4\pi n - 0.4\pi - 0.5\pi)}$

$= e^{j(0.4\pi n - 0.5\pi)} - e^{j(0.4\pi n - 0.9\pi)}$

$= e^{j(0.4\pi n)}e^{-j(0.5\pi)} - e^{j(0.4\pi n)}e^{-j(0.9\pi)}$

$= e^{j(0.4\pi n)}\left[e^{-j(0.5\pi)} - e^{-j(0.9\pi)}\right]$

$= e^{j(0.4\pi n)}[\cos(0.5\pi) - j\sin(0.5\pi) - \cos(0.9\pi) + j\sin(0.9\pi)]$

$= e^{j(0.4\pi n)}[0.951 - j0.691]$

Conversion from Cartesian to Polar Form:

$$\sqrt{(0.951)^2 + (0.691)^2} = 1.175$$

$$\arctan\left(\frac{-0.691}{0.951}\right) = -0.6283$$

$y[n] = e^{j(0.4\pi n)}\left(1.175 e^{-j0.6283}\right) = 1.175 e^{j(0.4\pi n - 0.6283)}$

Therefore, in the output signal, $A = 1.175$, the discrete-time frequency $\hat{\omega}_o = 0.4\pi$ rad/sample, and the phase $\phi = $ -0.6283 rad = -0.2π.

The discrete-time frequency in the output remained the same as the input.

However, the amplitude was scaled (multiplied) by 1.175 and the phase was shifted by 0.3π.