2.3 Tutorial lsim with Differential Equations

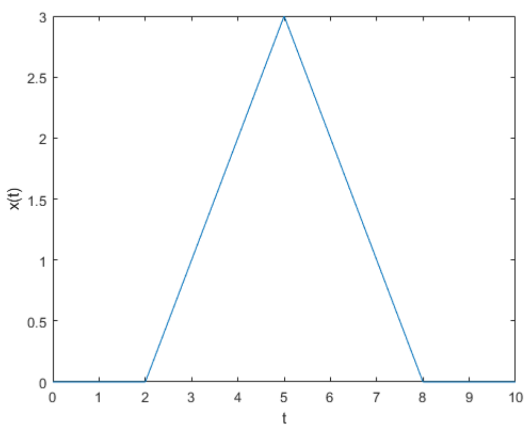
The function l***sim*** can be used to simulate the output of continous-time, causal LTI systems described by linear constant-coefficient differential equations of the form

$$\sum_{k=0}^{N} a_k \frac{d^k y(t)}{dt^k} = \sum_{m=0}^{M} b_m \frac{d^m x(t)}{dt^m} \qquad (2.11)$$

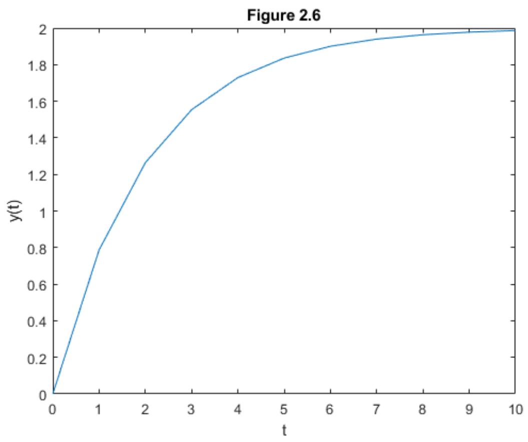
To use ***lsim***, the cofficients a_k and b_m must be stored in MATLAB vectors ***a*** and ***b***, respectively, in descending order of the indices ***k*** and ***m***. Rewriting Eq. (2.11) in terms of the vectors ***a*** and ***b*** gives

$$\sum_{k=0a}^{N} a(N+1-k) \frac{d^k y(t)}{dt^k} = \sum_{m=0}^{M} b(M+1-m) \frac{d^m x(t)}{dt^m} \qquad (2.12)$$

```
clf;

% plot x(t)
t = [0 1 2 5 8 9 10];
x = [0 0 0 3 0 0 0];

figure
plot(t,x);
xlabel('t');
```




```
ylabel('x(t)');
% plot y(t)
t = [0:10];
x = ones(1, length(t));
b = 1;
a = [1 0.5];
s = lsim(b, a, x, t);

figure;
plot(t, s);
title('Figure 2.6')
xlabel('t');
```




```
ylabel('y(t)');
```

The plot above shows the solid line reprsents the actual step response

$$s(t) = 2(1 - e^{-t/2})u(t) \qquad (2.14)$$

2.3 (a)

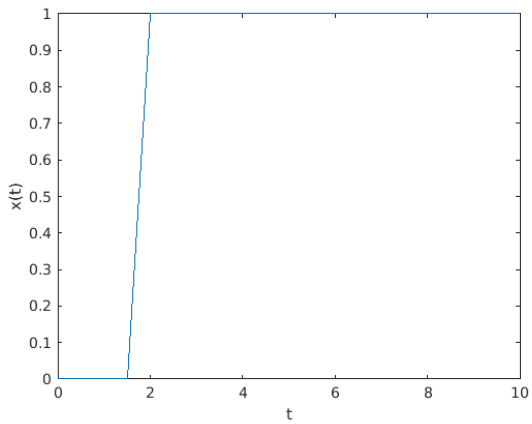
On your own, use lsim to compute the response of the causal LTI system descripted by

$$\frac{dy(t)}{dt} = -2y + x(t) \qquad (2.15)$$

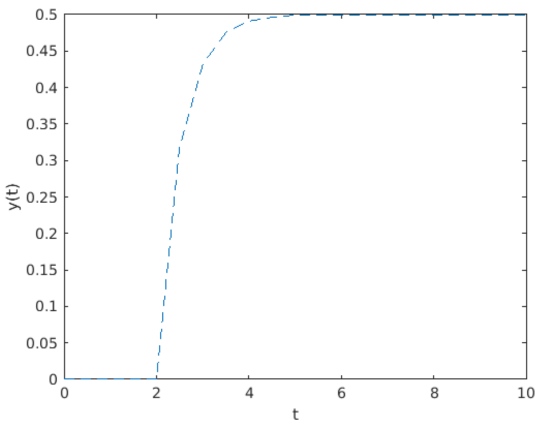
to the imput $x(t) = u(t-2)$. Your response should look like the plot in Figure 2.7, which is computed using t=[0:0.5:10].

Analysis: The plot from the MATLAB program blow does agree with the plot in Figure 2.7. It shows a shift to right by two units.

```
clf;

t = [0:0.5:10];
x = t*0;

x(5:end)=1;

figure;
plot(t, x);
xlabel('t');
ylabel('x(t)');
```




```
b = 1;
a = [1 2];
s = lsim(b, a, x, t);

figure;
plot(t, s, '--');
xlabel('t');
ylabel('y(t)');
```



2.3 (b)

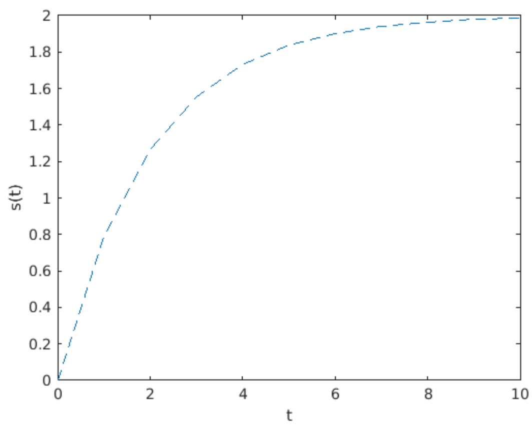
Use the step and impulse to compute the ***step*** and ***impulse*** reponses of the causal LTI system characterized by Eq. (2.13). Compare the ***step*** resonse comuted by step with that shown in Figure 2.6. Compare the signal returned by ***impulse*** with the exact impulse response, given by the derivative of s(t) in EQ. (2.14).

Analysis:

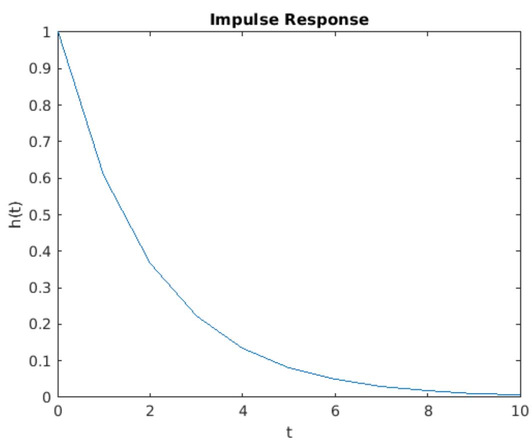
1. The plot of step resonse blow does agree with Figure 2.6.
2. The plot of impulse resonse is shown below also.

```
clf;
```

```
t = [0:1:10];
b = 1;
a = [1 0.5];

figure;
s = step(b, a, t);
plot(t, s, '--');
xlabel('t');
ylabel('s(t)');
```




```
h = impulse(b, a, t);

figure;
plot(t, h);
title('Impulse Response');
xlabel('t');
ylabel('h(t)');
```

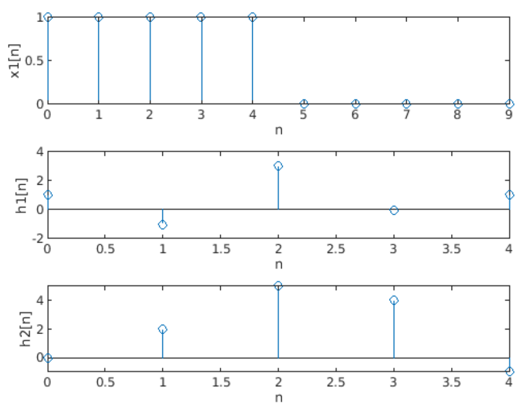


2.4 Properties of Discrete-Time Systems

I this exercise, you will verify the commutative, associative and distriutive properties of convolution for a specific set signals. In addition, you will examine the implications of these properties for serier and parallel connections of LTI systems. This problems in this exercise will assume that you are confortable and familliar the conv function described in Tutorial 2.1.

(a) Make appropriately labeled plots of all the signals using stem (see x1, h1 and h2 signals on page 30).

```
clf;

nx1 = [0:9];
x1 = nx1*0;
x1(1:5) = 1;

subplot(3, 1, 1);
stem(nx1, x1);
xlabel('n');
ylabel('x1[n]');

nh = [0:4];
h1 = [1 -1 3 0 1];
subplot(3, 1, 2);
stem(nh, h1);
xlabel('n');
ylabel('h1[n]');

h2 = [0 2 5 4 -1];
subplot(3, 1, 3);
stem(nh, h2);
xlabel('n');
ylabel('h2[n]');
```



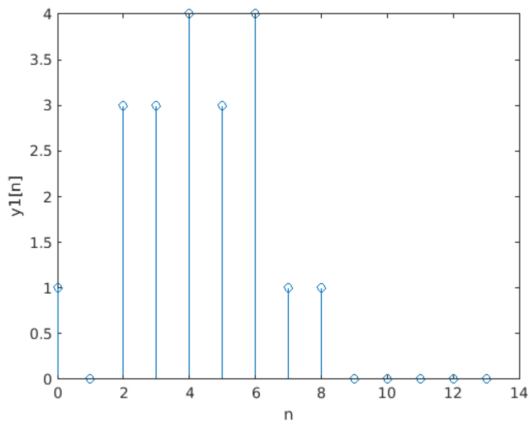
2.4 (b) Verify the commutative property of convoluaton operator.

If $y_1[n] = x[n] * h[n]$ and $y_2[n] = h[n] * x[n]$, then $y_1[n] = y_2[n]$

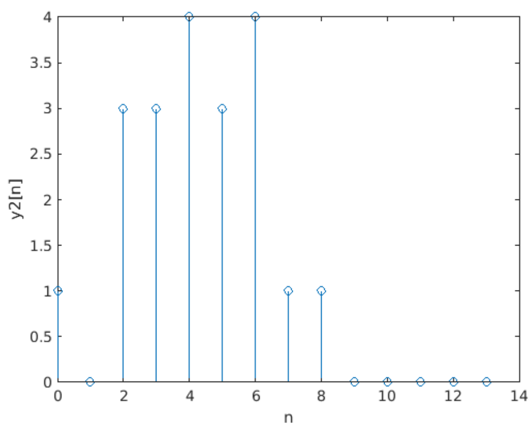
Analysis: The MATLAB code and plots does verify that $y_1[n] = y_2[n]$

```
clf;

nx1 = [0:9];
x1 = nx1*0;
```

```
x1(1:5) = 1;

nh = [0:4];
h1 = [1 -1 3 0 1];

hy = [0:13];
y1 = conv(x1, h1);

figure;
stem(hy, y1);
xlabel('n');
ylabel('y1[n]');
```




```
y2 = conv(h1, x1);

figure;
stem(hy, y2);
xlabel('n');
ylabel('y2[n]');
```



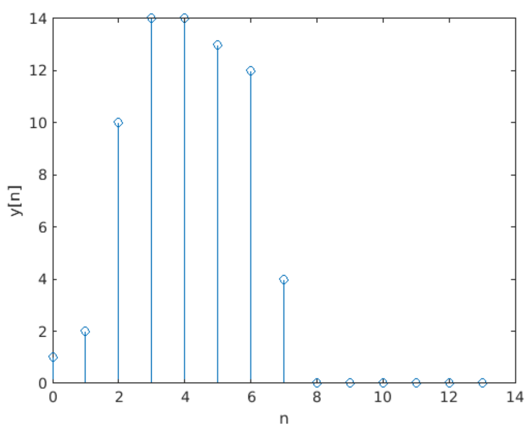
2.4 (c) Convolution is also distributive. This means that

$$x[n] * (h_1[n] + h_2[n]) = x[n] * h_1[n] + x[n] * h_2[n]$$

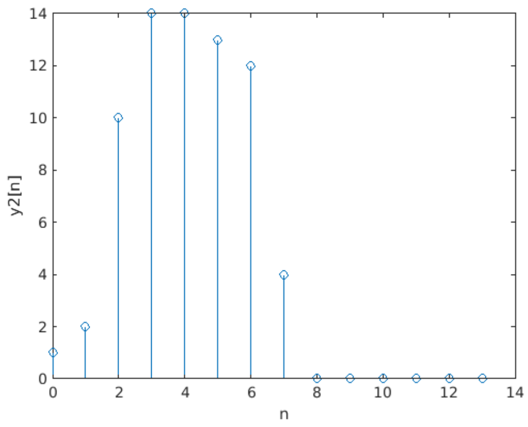
Do these two methods of computing the output give the same result?

Analysis: The MATLAB code and plots below shows that these two methods of computing the output do give the same result. Therefore it verifies that convolution is distributive.

```
clf;

nx1 = [0:9];
x1 = nx1*0;
x1(1:5) = 1;

nh = [0:4];
h1 = [1 -1 3 0 1];
h2 = [0 2 5 4 -1];

ny = [0:14];
y1 = conv(x1, (h1+h2));

figure;
stem(hy, y1);
xlabel('n');
ylabel('y[n]');
```




```
y2 = conv(x1, h1) + conv(x1, h2);

figure;
stem(hy, y2);
xlabel('n');
ylabel('y2[n]');
```



2.4 (d) Convolution also possesses that associative property, i.e.,

$$(x[n] * h_1[n]) * h_2[n] = x[n] * (h_1[n] * h_2[n])$$

Use the following steps to verify the associative property using x1, h1, and h2:

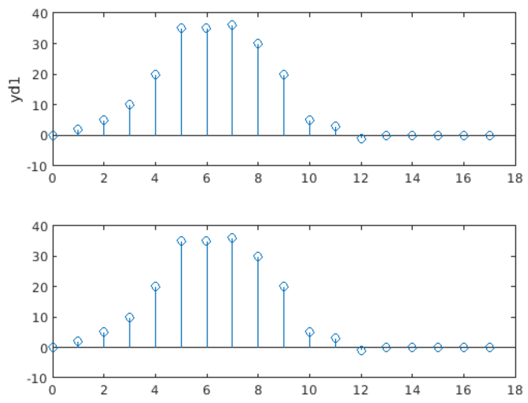
- Let ***w[n]*** be the output of the LTI system with impulse response $h_1[n]$ shown in the Figure 2.9. Computer $w[n]$ by convolving $x_1[n]$ and $h_1[n]$.
- Compute the output $y_{d1}[n]$ of the whole system by convolving $w[n]$ with $h_2[n]$.
- Find the impulse response $h_{series}[n] = h_1[n] * h_2[n]$
- Convolve $x_1[n]$ with $h_{series}[n]$ to get the output $y_{d2}[n]$

Compare $y_{d1}[n]$ and $y_{d2}[n]$. Did you get the same results when you process $x_1[n]$ with the indiviudal impulse responses as when you process it with $h_{series}[n]$. ?

Analysis: The plots bow do show $y_{d1}[n]$ and $y_{d2}[n]$ are same. Therefore, it verifies the associative property is true for convolution operation in LTI systems.

```
clf;

nx1 = [0:9];
x1 = nx1*0;
x1(1:5) = 1;

nh = [0:4];
h1 = [1 -1 3 0 1];
h2 = [0 2 5 4 -1];

w = conv(x1, h1);
yd1 = conv(w, h2);

hs = conv(h1, h2);
yd2 = conv(x1, hs);

ny = [0:17];
```

```
subplot(2, 1, 1);
stem(ny, yd1);
ylabel('yd1');

subplot(2, 1, 2);
stem(ny, yd2);
```



2.5 Linearity and Time-invariance

The problems in this exercise assume that you are familiar with the functions conv and filter. These functions are explained in Tutorial 2.1 and Tutorial 2.2.

Consider the systems:

System 1: $w[n] = x[n] - x[n-1] - x[n-2]$

System 2: $y[n] = cos(x[n])$

System 3: $z[n] = nx[n]$

where x[n] is the input to each system, and w'n], y[n] and z[n] ar the corresponding outputs.

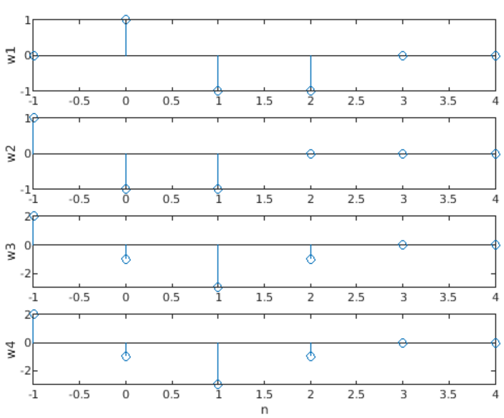
(a**)** Consider the three imputs signals $x_1[n] = \delta[n]$, $x_2[n] = \delta[n-1]$, and $x_3[n] = (\delta[n] + 2\delta[n-1])$. For sysem1, store in w1, w2, and w3 the response to the three inputs. The vectors w1, w2, and w3 need to contain the values of w[n] only on the interval $0 \leq n \leq 5$. To plot the four functions represented by w1, w2, w3, and w1+2*w2 within a single fiture. Make analogous plots for systems 2 and 3.

(b) State whether or not each system is linear. If it is linear, justfy your answer. If it is not linear, use he signals plotted in Part (a) to supply a counter-example.

(c) State whether of not each system is time-invariant. If it is time-invariant, justy your answer. If it is not time-invariant, use the signals plotted in Part (a) to supply a counter-example.

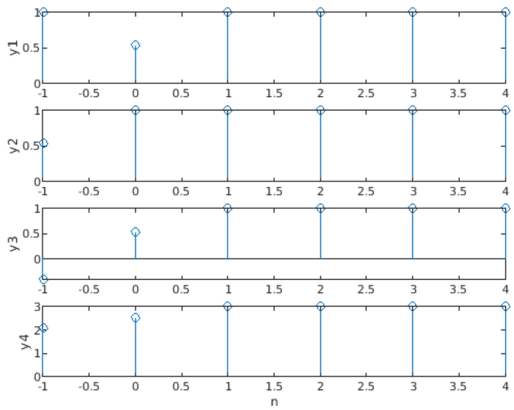
```
clf;
```

```
a = [1];
b = [1 -1 -1];

n1 = [-1:4];
x1 = [0 1 0 0 0 0];
w1 = filter(b, a, x1);
figure;

subplot(4, 1, 1);
stem(n1, w1);
ylabel('w1');

n2 = [-1:4];
x2 = [1 0 0 0 0 0];
w2 = filter(b, a, x2);

subplot(4, 1, 2);
stem(n2, w2);
ylabel('w2');

n3 = [-1:4];
x3 = x1 + 2 * x2;
w3 = filter(b, a, x3);

subplot(4, 1, 3);
stem(n3, w3);
ylabel('w3');

n4 = [-1:4];
w4 = w1 + 2 * w2;

subplot(4, 1, 4);
stem(n4, w4);
xlabel('n');
ylabel('w4');
```



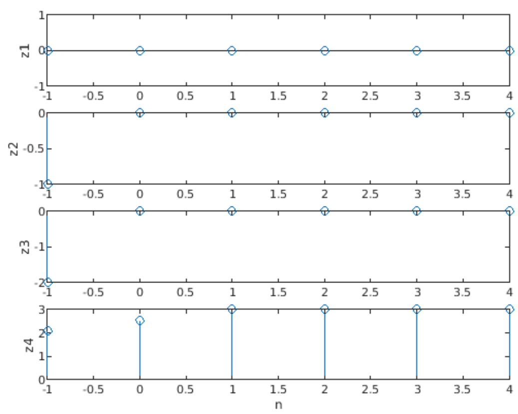

```
figure;

n1 = [-1:4];
```

```
x1 = [0 1 0 0 0 0];
y1 = cos(x1);

subplot(4, 1, 1);
stem(n1, y1);
ylabel('y1');

n2 = [-1:4];
x2 = [1 0 0 0 0 0];
y2 = cos(x2);

subplot(4, 1, 2);
stem(n2, y2);
ylabel('y2');

n3 = [-1:4];
x3 = x1 + 2 * x2;
y3 = cos(x3);

subplot(4, 1, 3);
stem(n3, y3);
ylabel('y3');

n4 = [-1:4];
y4 = y1 + 2 * y2;

subplot(4, 1, 4);
stem(n4, y4);
xlabel('n');
ylabel('y4');
```



```
figure;

n = [-1:4];
x1 = [0 1 0 0 0 0];
z1 = n.*x1;

subplot(4, 1, 1);
stem(n, z1);
```

```
ylabel('z1');

x2 = [1 0 0 0 0 0];
z2 = n.*x2;

subplot(4, 1, 2);
stem(n, z2);
ylabel('z2');

x3 = x1 + 2 * x2;
z3 = n.*x3;

subplot(4, 1, 3);
stem(n, z3);
ylabel('z3');

z4 = y1 + 2 * y2;

subplot(4, 1, 4);
stem(n, z4);
xlabel('n');
ylabel('z4');
```



2.5 (b) Weither or not the above systems is linear? Justify your answer

Results Analysis:

System 1 is a linear system since w3 = w4 which satisfies the linear property of system, which is stated that the system response to a linear combination of input x1 and x2 is the linear combination of system responses to impute x1 and x2.

System 2 is not a linear system since y3 does not equal y4. The plot above shows a counter example.

System 3 is also not a linear system since z3 does not equal z4. The plot aboce shows a counter-example.

2.5 (c) Weither or not the above systems is time-invariant? Justify your answer.

Results Analysis:

System 1 is a time-invariant system since w1[n] = x[n] and w2 = x[n-1], if the system is time-invariant, it must satisfy that w2 = w1[n-1] which is justified by the plot.

System 2 is also a time-invarient system because it can be justified for the same reaspon which is that y2[n] = y1[n].

System 3 is not a time-invariant system since z2[n] does not equal z1[n-1], which is a counter-example for the given inputs.