SS-Computer Assignment - 01 - Spring 2019

Shaik Masihullah,
S20180010159.

Function :

convolution.m :

```
function y = convolution(x,h)
    M = length(x);
    N = length(h);
    L = M+N-1;
    Xe = zeros(1,L);
    He = zeros(1,L);
    Xe(1:M) = x;
    He(1:N) = h;
    X = zeros(L,L);
    for n = 1:L
        Xtemp = Xe(n:-1:1);
        X(n,1:n) = Xtemp;
    end
    y = He * transpose(X);
return;
```

1 Linear Convolution

Write a matlab code for linear convolution of two signals. Then

1. Generate the causal signals

$$\begin{aligned} x_1[n] &= \{\underset{\uparrow}{3},\ 1,\ 4,\ 16,\ 2\} \\ x_2[n] &= \{\underset{\uparrow}{3}, -1, 3, -1\} \\ h[n] &= \{\underset{\uparrow}{2},\ -1,\ -4,\ 1,\ -3\} \end{aligned}$$

Now, determine the output of the given systems

$$y_1[n] = (x_1[n] + x_2[n]) * h[n]$$
$$y_2[n] = x_1[n] * h[n] + x_2[n] * h[n]$$

(a) Perform the calculations using your matlab code and verify the results using the inbuilt function *conv* and on-paper calculations.

(b) Verify if the outputs $y_1[n]$ and $y_2[n]$ are identical or not.

(c) Using the *stem* function, plot the signals $x_1[n]$, $x_2[n]$, $h[n]$, $y_1[n]$ and $y_2[n]$.

Main Code :

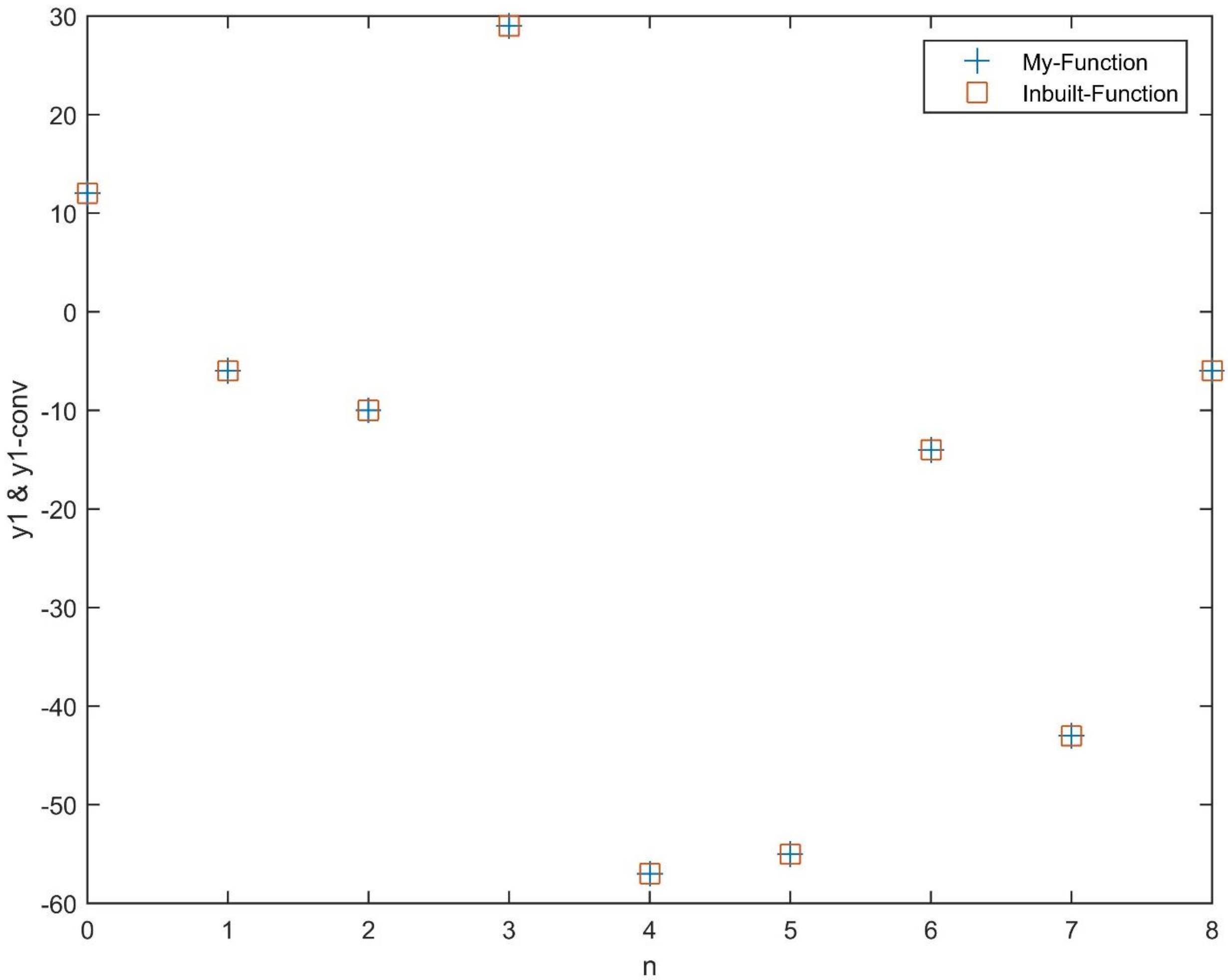
```
%%
%Defining Signals
x1 = [ 3, 1, 4, 16, 2];
x2 = [ 3, -1, 3, -1];
h = [ 2, -1, -4, 1, -3];

%%
%Calculating lengths of signals
M = max(length(x1),length(x2));
N = length(h);
L = M+N-1;
```

```
%%
%Calling convolution fucntion seperately to both x signals
y_1 = convolution(x1,h);

y_2 = convolution(x2,h);

%Adding seperately convoluted signals (adding 0 to make both of same length)
y1 = y_1 + [y_2,0];

%Convoluting using inbuilt function to verify
y_1_conv = conv(x1,h);
y_2_conv = conv(x2,h);

y1_conv = y_1_conv + [y_2_conv,0];

%%
%Adding both x signals to convolute after adding
X = x1 + [x2,0];

%Calling convolution function
y2 = convolution(X,h);

%Convoluting using inbuilt function to verify
y2_conv = conv(X,h);

%%
%Verifying my convolution function with inbuilt convolution function
nvec = 0:L-1;
figure();
plot(nvec,y1,'+','MarkerSize',10);
hold on;
plot(nvec,y1_conv,'s','MarkerSize',10);
xlabel('n');
ylabel('y1 & y1-conv');
legend('My-Function','Inbuilt-Function');

figure();
plot(nvec,y2,'+','MarkerSize',10);
hold on;
plot(nvec,y2_conv,'s','MarkerSize',10);
xlabel('n');
ylabel('y2 & y2-conv');
legend('My-Function','Inbuilt-Function');

%%
%Verifying both y1 and y2 are same
figure();
plot(nvec,y1,'+','MarkerSize',10);
hold on;
plot(nvec,y2,'s','MarkerSize',10);
xlabel('n');
ylabel('y1 & y2');
legend('y1[n] = (x1[n] + x2[n])*h[n]','y2[n] = x1[n]*h[n] + x2[n]*h[n]');

%%
%Plotting x and h signals
nx1 = 0:M-1;
nx2 = 0:length(x2)-1;

figure();
subplot(231);
stem(nx1,x1,'filled','LineWidth',2);
xlabel('n');
ylabel('x1[n]');
```

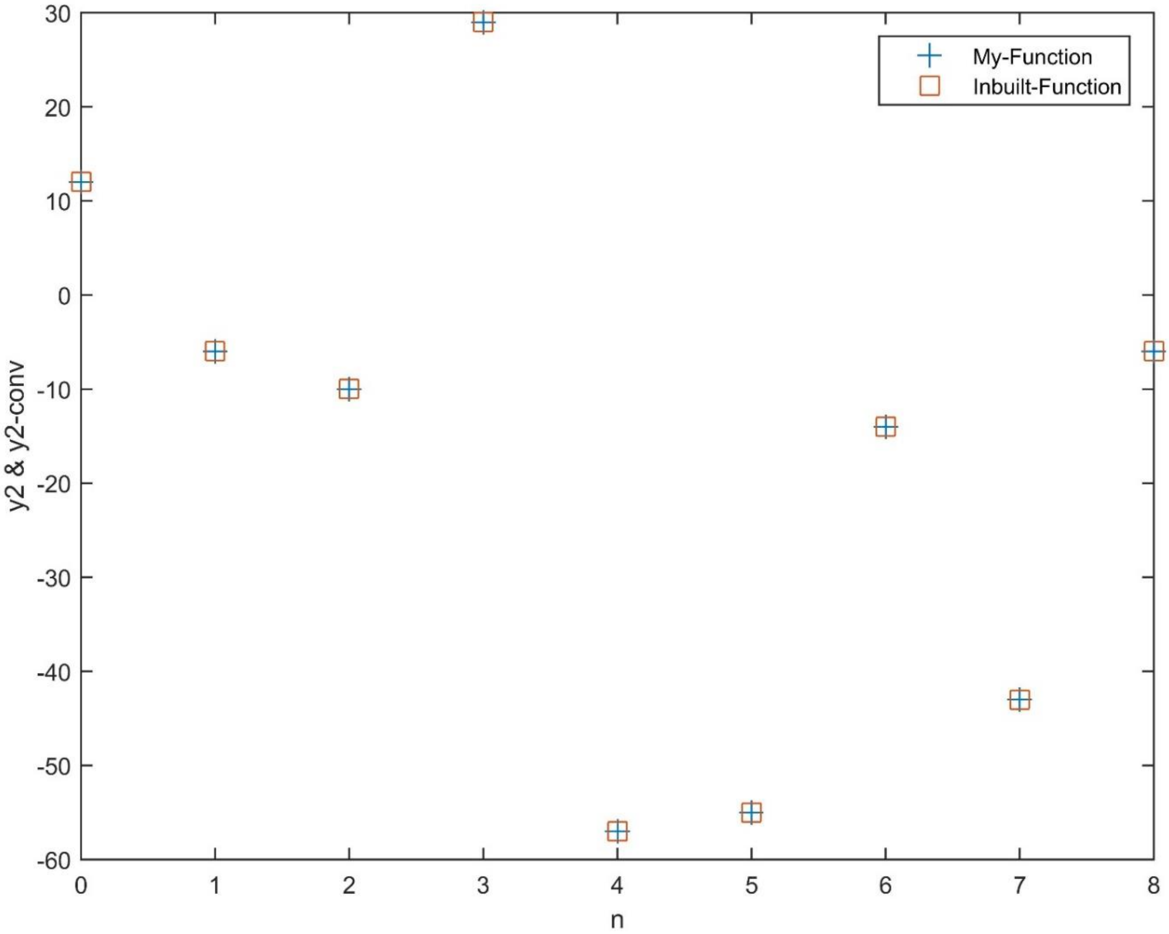
```
subplot(232);
stem(nx2,x2,'filled','LineWidth',2);
xlabel('n');
ylabel('x2[n]');

n2 = 0:N-1;

subplot(233);
stem(n2,h,'filled','LineWidth',2);
xlabel('n');
ylabel('h[n]');

%Plotting y signals
figure();
subplot(221);
stem(nvec,y1,'filled','LineWidth',2);
xlabel('n');
ylabel('y1[n]');

subplot(222);
stem(nvec,y2,'filled','LineWidth',2);
xlabel('n');
ylabel('y2[n]');
```

PLOTS :

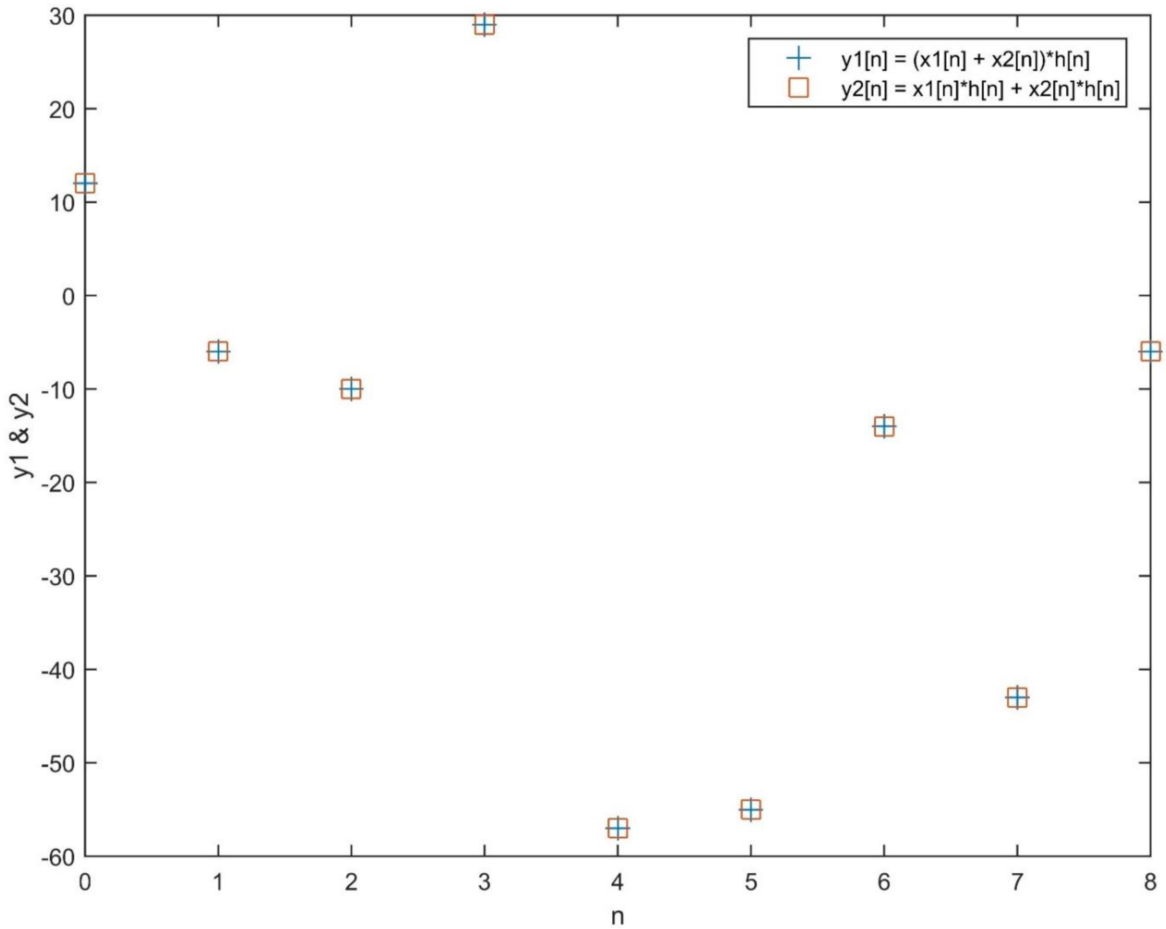
Verifying y1[n] with inbuilt function:

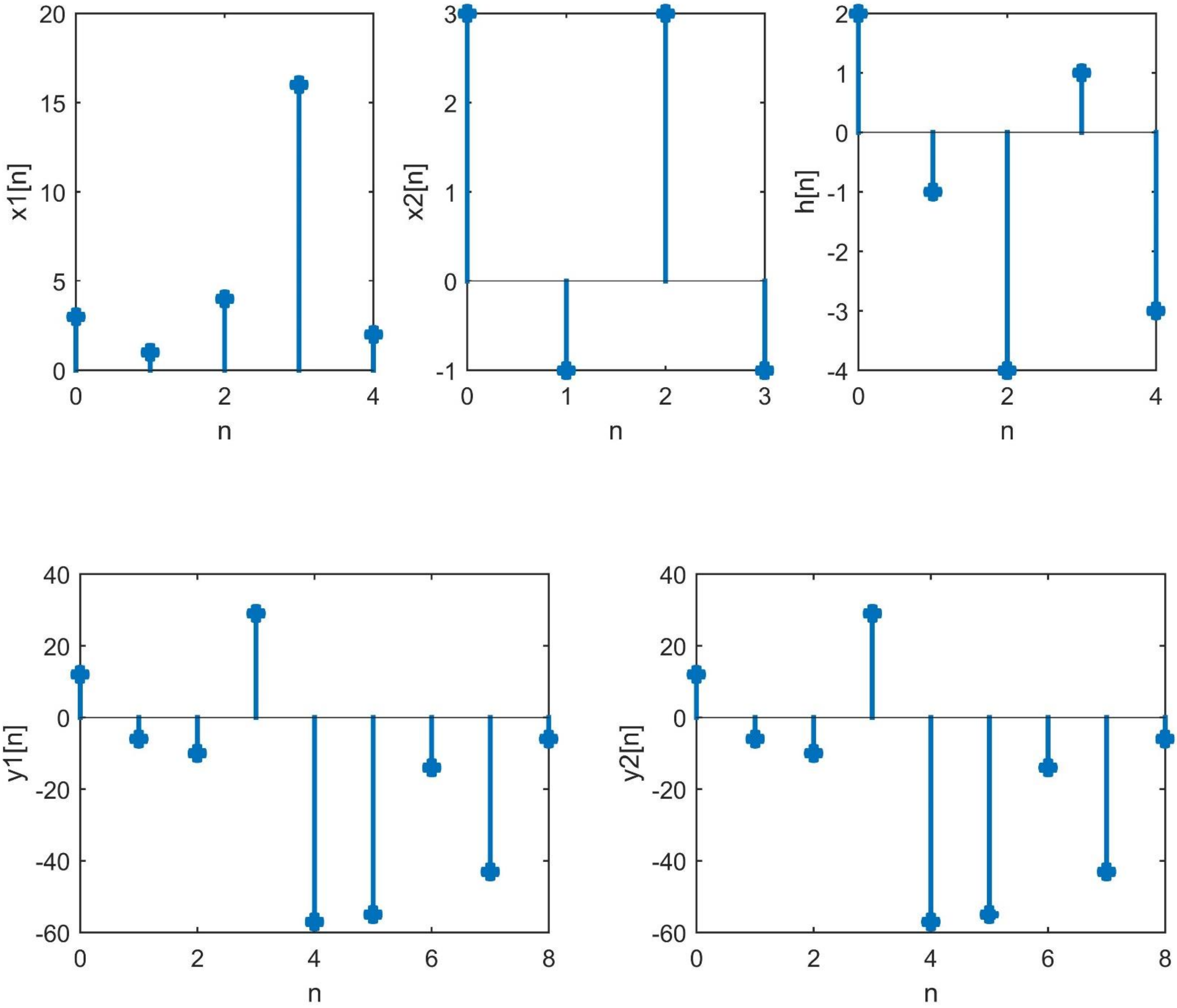


Verifying y2[n] with inbuilt function:



Verifying y1[n] & y2[n] are identical:




x1[n]
x2[n]
h[n]
y1[n]
y2[n]
n

On paper Verification:

1A)

$x_1[n] = \{\underset{\uparrow}{3}, 1, 4, 16, 2\}$

$x_2[n] = \{\underset{\uparrow}{3}, -1, 3, -1\}$

$h[n] = \{\underset{\uparrow}{2}, -1, -4, 1, -3\}$

$x_1[n] + x_2[n] = \{6, 0, \underset{\uparrow}{7}, 15, 2\}$

$y_1[n] = (x_1[n] + x_2[n]) * h[n]$

$y_1[n]$

	2	-1	-4	1	-3
6	12	-6	-24	6	-18
0	0	0	0	0	0
7	14	-7	-28	7	-21
15	30	-15	-60	15	-45
2	4	-2	-8	2	-6

$y_1[n] = \{\underset{\uparrow}{12}, -6, -10, 29, -57, -55, -14, -43, -6\}$

$y_2[n] = x_1[n] * h[n] + x_2[n] * h[n]$

$x_1[n] * h[n]$

	2	-1	-4	1	-3
3	6	-3	-12	3	-9
1	2	-1	-4	1	-3
4	8	-4	-16	4	-12
16	32	-16	-64	16	-48
2	4	-2	-8	2	-6

$\{\underset{\uparrow}{6}, -1, -59, 27, -36, -65, -4, -46, -6\}$

$x_2[n] * h[n]$

	2	-1	-4	1	-3
3	6	-3	-12	3	-9
-1	-2	1	4	-1	3
3	6	-3	-12	3	-9
-1	-2	1	4	-1	3

$\{\underset{\uparrow}{6}, -5, -5, 2, -21, 10, -10, 3\}$

$y_2[n] = \{\underset{\uparrow}{12}, -6, -10, 29, -57, -55, -14, -43, -6\}$

$y_1[n] = y_2[n]$

2. Next, generate the signals

$$x[n] = \{-3, \ -2, 0, \underset{\uparrow}{1}, \ 2, \ 3,\}$$
$$h[n] = \{\underset{\uparrow}{3}, \ 1, \ 1, \ 3, \ 1, \ 1\}$$

Now, determine the output of the given system

$$y[n] = x[n-3] * h[n]$$

(a) Perform the calculations using your matlab code and verify the results using the inbuilt function *conv* and on-paper calculations.

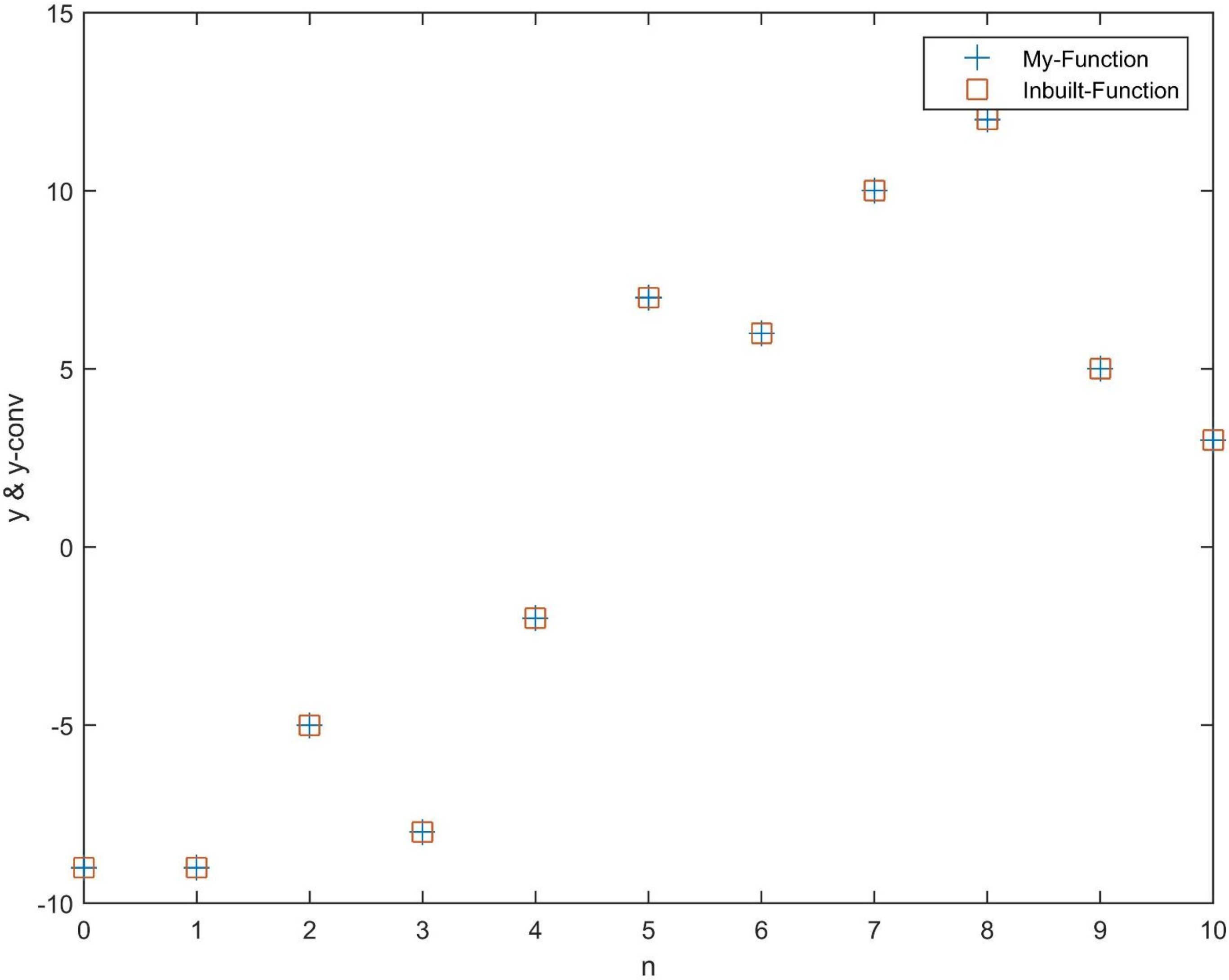
(b) Using the *stem* function, plot the signals $x[n]$, $h[n]$, and $y[n]$.

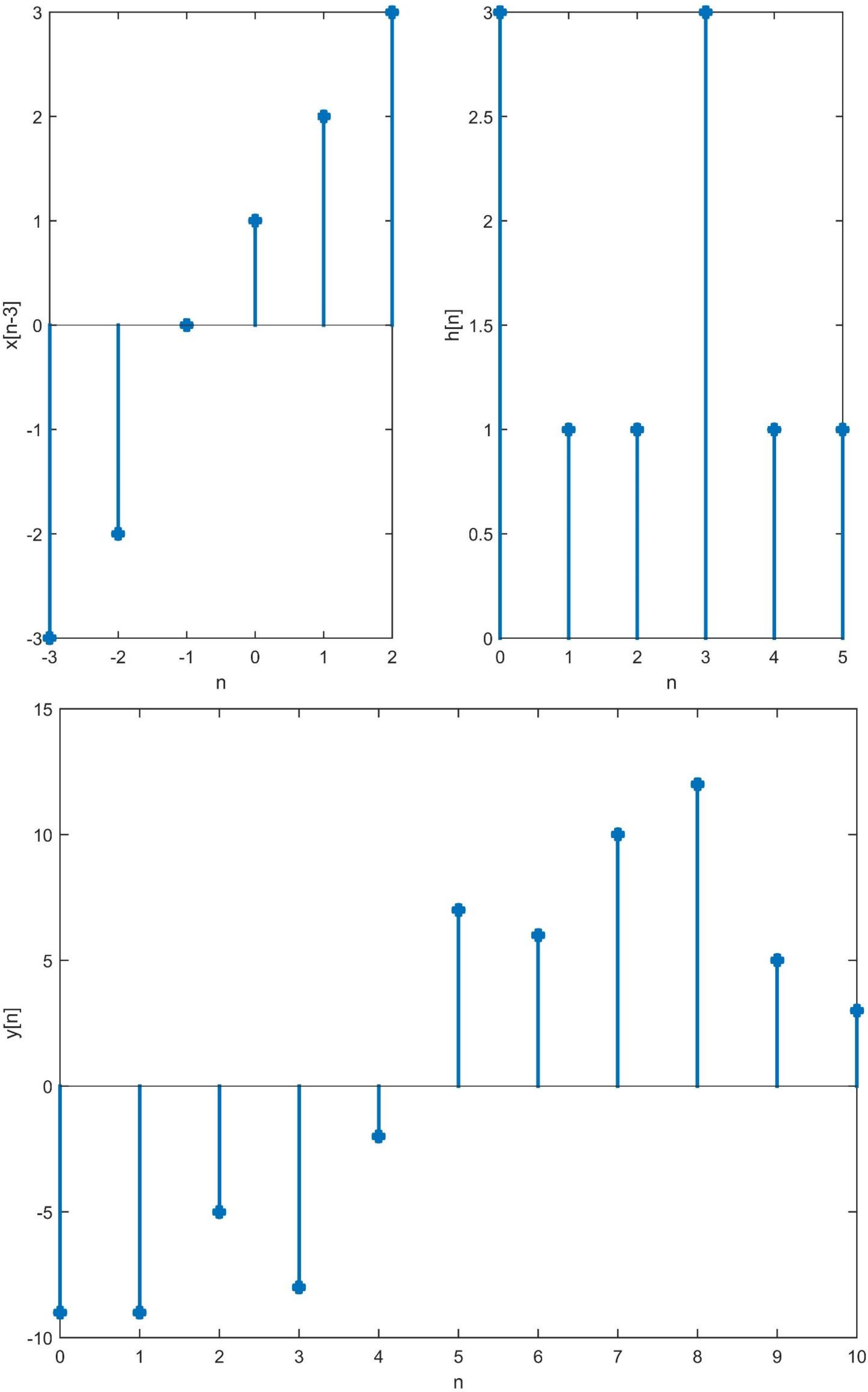
Code :

```
%%
%Defining Signals
x = [ -3, -2, 0, 1, 2, 3];
h = [ 3, 1, 1, 3, 1, 1];

%Calculating lengths of signals
M = length(x);
N = length(h);
L = M+N-1;
%%
%Convoluting the signals
y = convolution(x,h)

%Using Inbuilt function
y_conv = conv(x,h)

%%
%Defining ranges
n = -3:M-4;
n1 = 0:N-1;
nvec = 0:L-1;

%%
%Verifying that both gives the same result
figure();
plot(nvec,y,'+','MarkerSize',10);
hold on;
plot(nvec,y_conv,'s','MarkerSize',10);
xlabel('n');
ylabel('y & y-conv');
legend('My-Function','Inbuilt-Function');

%%
%Plotting x,h signals
figure();
subplot(121);
stem(n,x,'filled','LineWidth',2);
xlabel('n');
ylabel('x[n-3]');

subplot(122);
stem(n1,h,'filled','LineWidth',2);
xlabel('n');
ylabel('h[n]');

%Plotting y signal
figure();
stem(nvec,y,'filled','LineWidth',2);
xlabel('n');
ylabel('y[n]');
```

PLOTS:

Verifying y[n] with inbuilt function:




x[n-3]
n
h[n]
n
y[n]
n

On paper Verification:

2a) $x[n] = \{-3, -2, 0, \underset{\uparrow}{1}, 2, 3\}$

$h[n] = \{\underset{\uparrow}{3}, 1, 1, 3, 1, 1\}$

$x[n-3] = \{\underset{\uparrow}{-3}, -2, 0, 1, 2, 3\}$

$y[n] = x[n-3] * h[n]$

	3	1	1	3	1	1
-3	-9	-3	-3	-9	-3	-3
-2	-6	-2	-2	-6	-2	-2
0	0	0	0	0	0	0
1	3	1	1	3	1	1
2	6	2	2	6	2	2
3	9	3	3	9	3	3

$y[n] = \{\underset{\uparrow}{-9}, -9, -5, -8, -2, 7, 6, 10, 12, 5, 3\}$

3. Next generate the causal signals

$$x[n] = \{\underset{\uparrow}{1},\ 2,\ -3,\ 8,\ -9\}$$
$$h[n] = \{\underset{\uparrow}{3},\ 2,\ 1,\ 2,\ 3\}$$

Now, compute the ouput of the given systems

$$y_1[n] = x[n] * h[1-n]$$
$$y_2[n] = x[1-n] * h[n]$$

(a) Perform the calculations using your matlab code and verify the results using the inbuilt function *conv* and on-paper calculations.

(b) Using the *stem* function, plot the signals $x[n]$, $h[n]$, $y_1[n]$ and $y_2[n]$.

(c) Verify if the outputs $y_1[n]$ and $y_2[n]$ are identical or not.

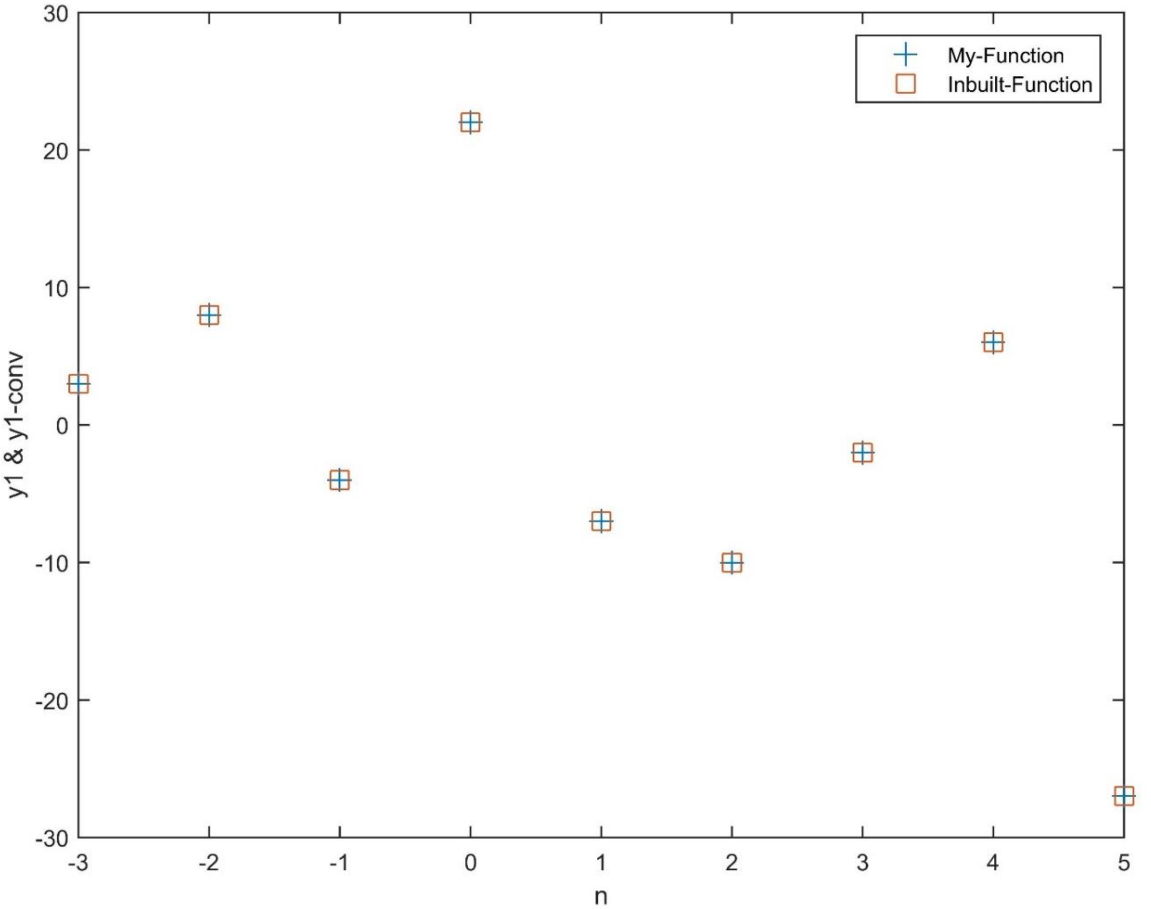
Code :

```
%%
%Defining Signals
x = [ 1, 2, -3, 8, -9];
h = [ 3, 2, 1, 2, 3];

%%
%Finding the lengths of signals
M = length(x);
N = length(h);
L = M+N-1;

%Flipping the signals using the fliplr() function
%for x[1-n] and h[1-n]
x_new = fliplr(x);
h_new = fliplr(h);

%%
%Convoluting the signals
y1 = convolution4(x,h_new);

y2 = convolution4(x_new,h);

%Using inbuilt functino
y1_conv = conv(x,h_new);
y2_conv = conv(x_new,h);

%%
%Defining ranges
nvec = -3:L-4;
n_new = -3:M-4;
n = 0:M-1;

%%
%Verifying my convolution function with inbuilt convolution function
figure();
plot(nvec,y1,'+','MarkerSize',10);
hold on;
plot(nvec,y1_conv,'s','MarkerSize',10);
xlabel('n');
ylabel('y1 & y1-conv');
legend('My-Function','Inbuilt-Function');
```

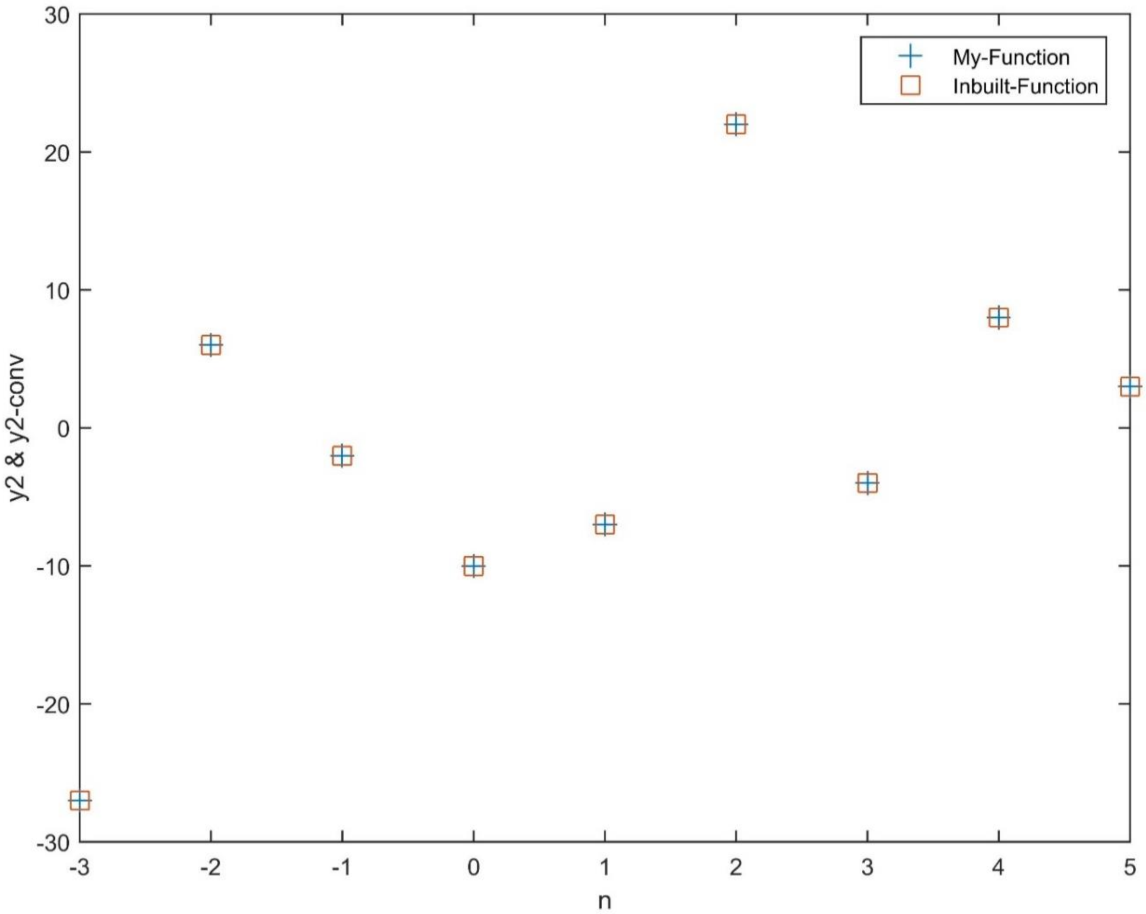
```
figure();
plot(nvec,y2,'+','MarkerSize',10);
hold on;
plot(nvec,y2_conv,'s','MarkerSize',10);
xlabel('n');
ylabel('y2 & y2-conv');
legend('My-Function','Inbuilt-Function');

%%
%Plotting y1 signal
figure();
subplot(231);
stem(n,x,'filled','LineWidth',2);
xlabel('n');
ylabel('x[n]');

subplot(232);
stem(n_new,h,'filled','LineWidth',2);
xlabel('n');
ylabel('h[1-n]');

subplot(233);
stem(nvec,y1,'filled','LineWidth',2);
xlabel('n');
ylabel('y1[n]');

%Plotting y2 signal
subplot(234);
stem(n_new,x,'filled','LineWidth',2);
xlabel('n');
ylabel('x[1-n]');

subplot(235);
stem(n,h,'filled','LineWidth',2);
xlabel('n');
ylabel('h[n]');

subplot(236);
stem(nvec,y2,'filled','LineWidth',2);
xlabel('n');
ylabel('y2[n]');

%%
%Verifying both y1 and y2 are same
figure();
plot(nvec,y1,'+','MarkerSize',10);
hold on;
plot(nvec,y2,'s','MarkerSize',10);
xlabel('n');
ylabel('y1 & y2');
legend('y1[n] = (x1[n] + x2[n])*h[n]','y2[n] = x1[n]*h[n] + x2[n]*h[n]');
```

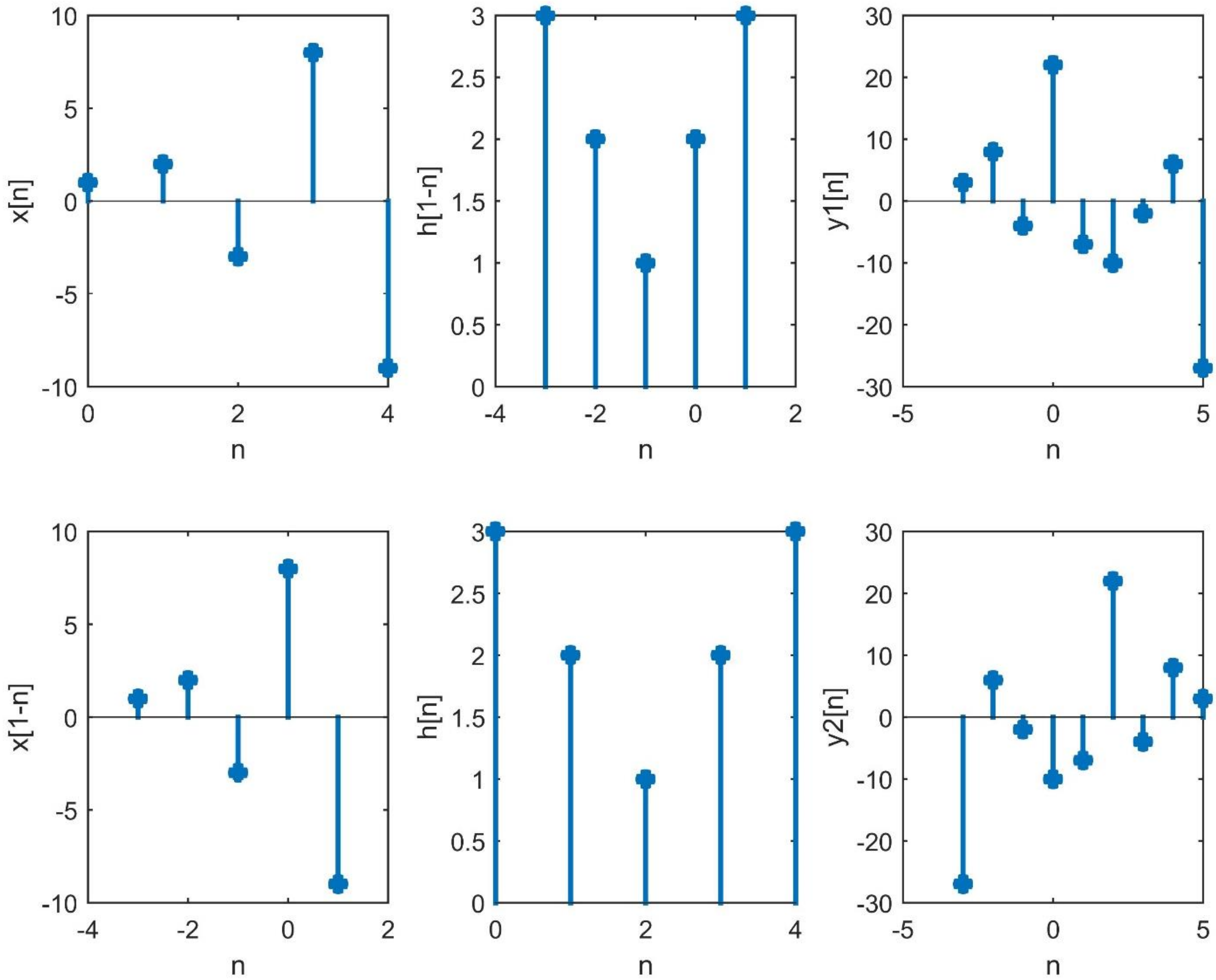
PLOTS:

Verifying y1[n] with inbuilt function:

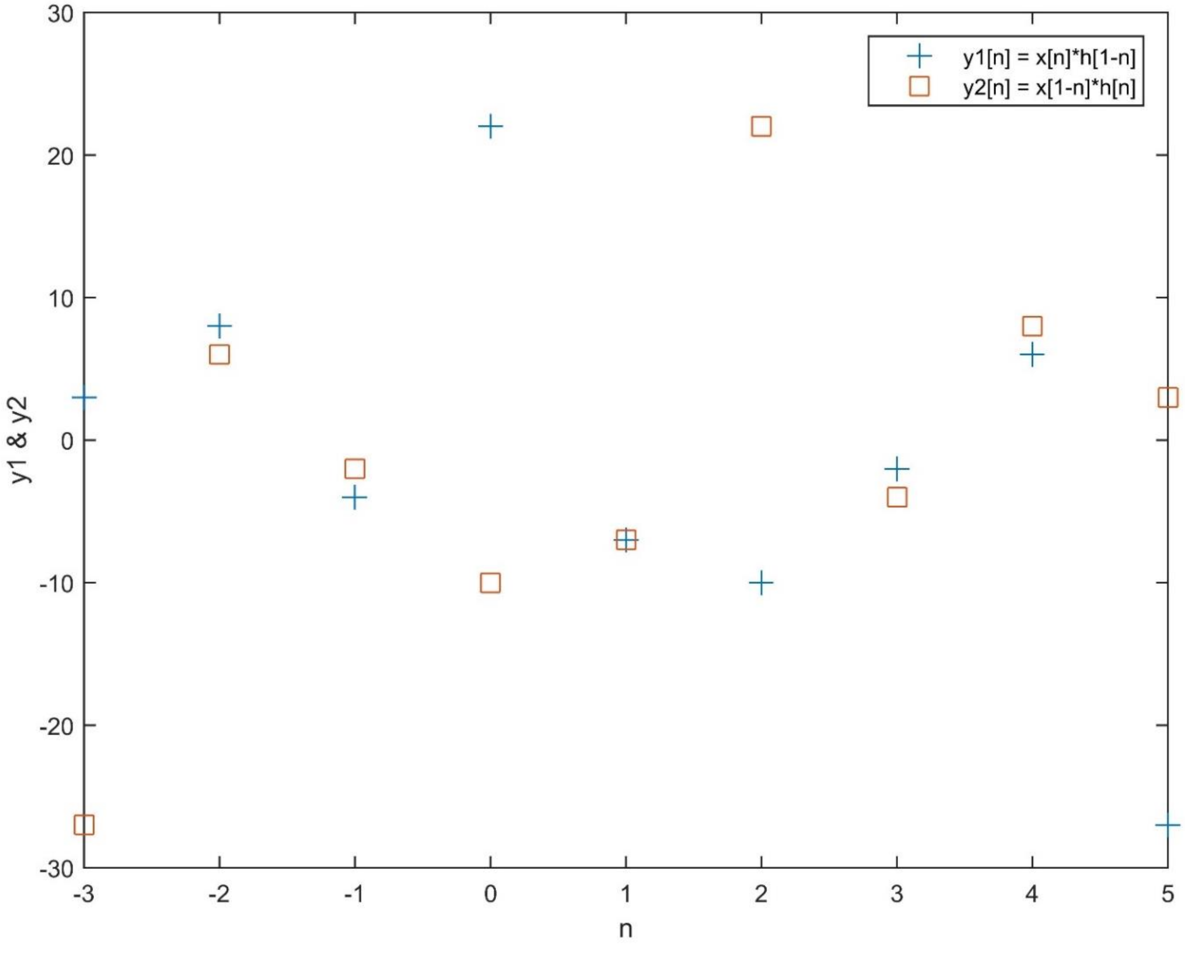


Verifying y2[n] with inbuilt function:





Verifying y1[n] & y2[n] are identical or not:



On paper Verification:

3A) $x[n] = \{\underset{\uparrow}{1}, 2, -3, 8, -9\}$

$h[n] = \{\underset{\uparrow}{3}, 2, 1, 2, 3\}$

$y_1[n] = x[n] * h[1-n]$

$h(1-n) = \{3, 2, 1, \underset{\uparrow}{2}, 3\}$

	3	2	1	2	3
1	3	2	1	2	3
2	6	4	2	4	6
-3	-9	-6	-3	-6	-9
8	24	16	8	16	24
-9	-27	-18	-9	-18	-27

$y_1[n] = \{3, 8, -4, \underset{\uparrow}{22}, -7, -10, -2, 6, -27\}$

$y_2[n] = x(1-n] * h[n]$

$x[1-n] = \{-9, 8, -3, \underset{\uparrow}{2}, 1\}$

	3	2	1	2	3
-9	-27	-18	-9	-18	-27
8	24	16	8	16	24
-3	-9	-6	-3	-6	-9
2	6	4	2	4	6
1	3	2	1	2	3

$y_2[n] = \{-27, 6, -2, \underset{\uparrow}{-10}, -7, 22, -4, 8, 3\}$

$y_1(n) \neq y_2(n)$