KATHMANDU UNIVERSITY

DHULIKHEL, KAVRE

Subject: COMP-407: Digital Signal Processing

Lab no: 5

Submitted By:

Name: Ayush Kumar Shah
Roll no: 44
Group: CE 4th year 1st sem
Level: UNG

Submitted To:

Satyendra Nath Lohani Sir

Convolution of two signals (Linear and Circular Convolution).

Theory:

Linear convolution:

It is computed by multiplying one sequence by a time reversed and linearly shifted version of other and summing the values of the product over all m.

i.e. $y[n] = x1[n] * x2[n] = \sum_{m=-\infty}^{\infty} x1[m]x2[n-m]$

If x1[n] and x2[n] have L and M number of discrete signals then y[n] will have L+M-1 discrete signals.

Linear convolution is defined for infinite length signals and is not periodic.

Circular Convolution:

It is computed by multiplying one sequence by a circularly time reversed and circularly shifted version of other and summing the values of the product from m=0 to m=N-1.

i.e. $y[n] = x1[n] \otimes x2[n] = \sum_{m=0}^{N-1} x1[m]x2[(n-m)_N]$

If x1[n] and x2[n] have L and M number of discrete signals then y[n] will have Max(L,M) discrete signals.

For circular convolution, both signals must be of equal length. If they are of unequal length, zeros are padded to make them equal.

Circular convolution is defined for finite length signals and is periodic.

To perform circular convolution, we use property of DFT i.e. circular convolution is obtained by taking inverse DFT of product of DFTs of two signals.

1. **Perform Linear Convolution:**
    a. **x[n]= $\{1,\underset{\uparrow}{1},1\}$ & h[n]=$\{1,\underset{\uparrow}{1},1\}$.**

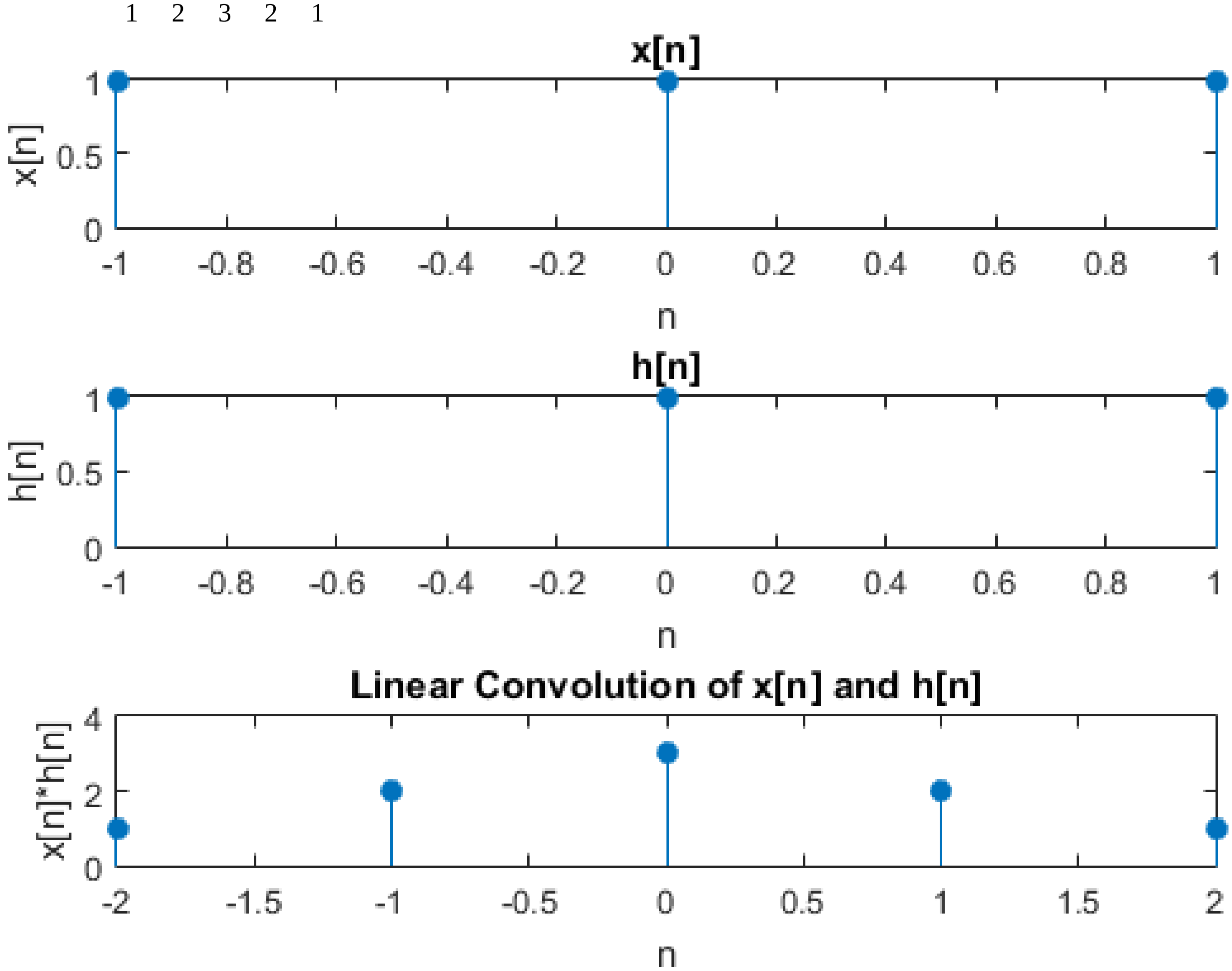
    Code:

```
%linear convolution
%ques i
x = [1 1 1]
h = [1 1 1]
figure;
% x[n]
subplot(3,1,1);
n=[-1:1];
stem(n,x,'filled');
title('x[n]');
xlabel('n');
ylabel('x[n]');

%h[n]
subplot(3,1,2);
n=[-1:1];
```

```
stem(n,h,'filled');
title('h[n]');
xlabel('n');
ylabel('h[n]');

%convolution
subplot(3,1,3)
lin_conv = conv(x,h)
xlim=[-2:2];
stem(xlim,lin_conv,'filled');
ylim([0 4]);
xlabel('n');
ylabel('x[n]*h[n]');
title('Linear Convolution of x[n] and h[n]');
```

Output:

```
x =
    1     1     1

h =
    1     1     1

lin_conv =
    1     2     3     2     1
```



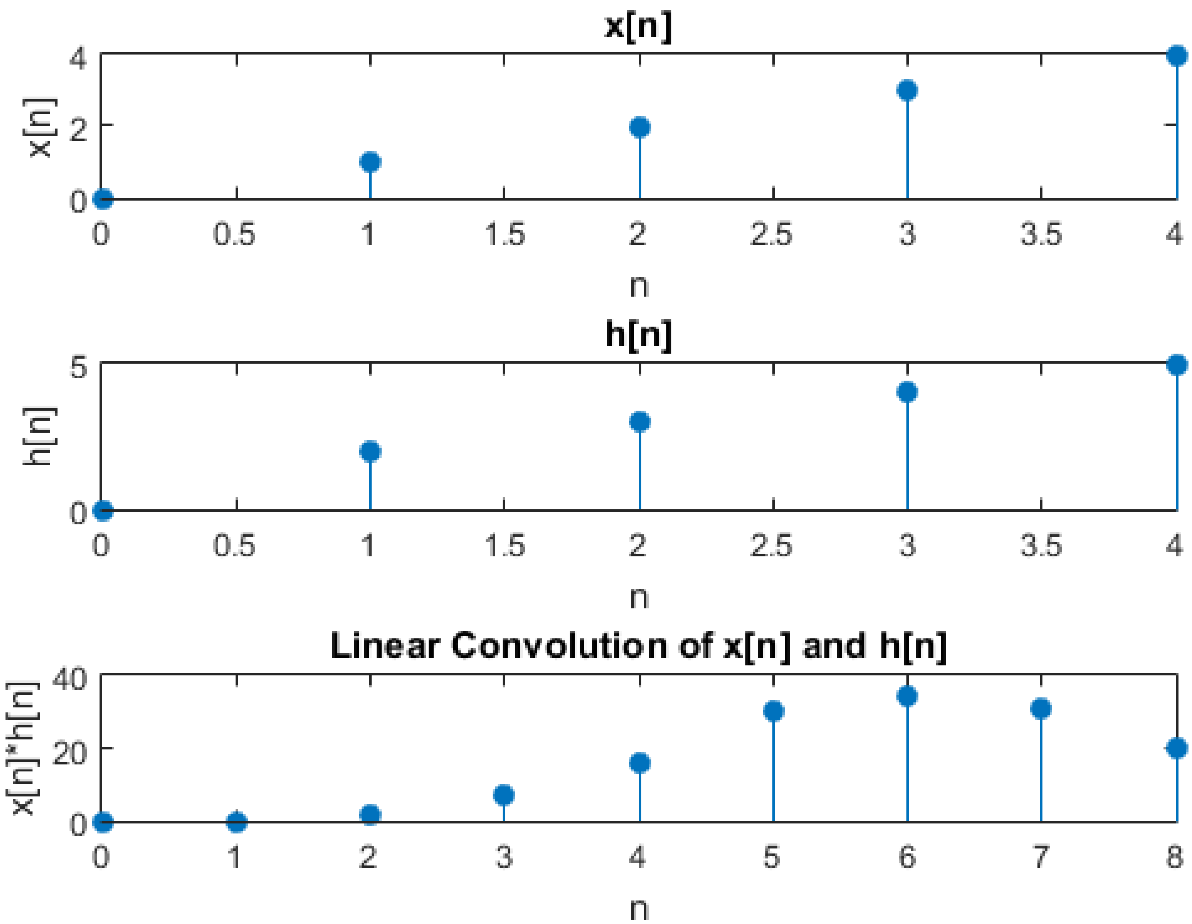
b. x[n]={0,1,2,3,4} & h[n]={0,2,3,4,5}

Code:

```
%ques ii
x = [0 1 2 3 4]
h = [0 2 3 4 5]
figure;
% x[n]
subplot(3,1,1);
n=[0:4];
stem(n,x,'filled');
title('x[n]');
xlabel('n');
ylabel('x[n]');

%h[n]
subplot(3,1,2);
n=[0:4];
stem(n,h,'filled');
title('h[n]');
xlabel('n');
ylabel('h[n]');

%convolution
subplot(3,1,3)
lin_conv = conv(x,h)
xlim=[0:8];
stem(xlim,lin_conv,'filled');
xlabel('n');
ylabel('x[n]*h[n]');
title('Linear Convolution of x[n] and h[n]');
```

Output:

```
x =
    0    1    2    3    4

h =
    0    2    3    4    5

lin_conv =
    0    0    2    7   16   30   34   31   20
```



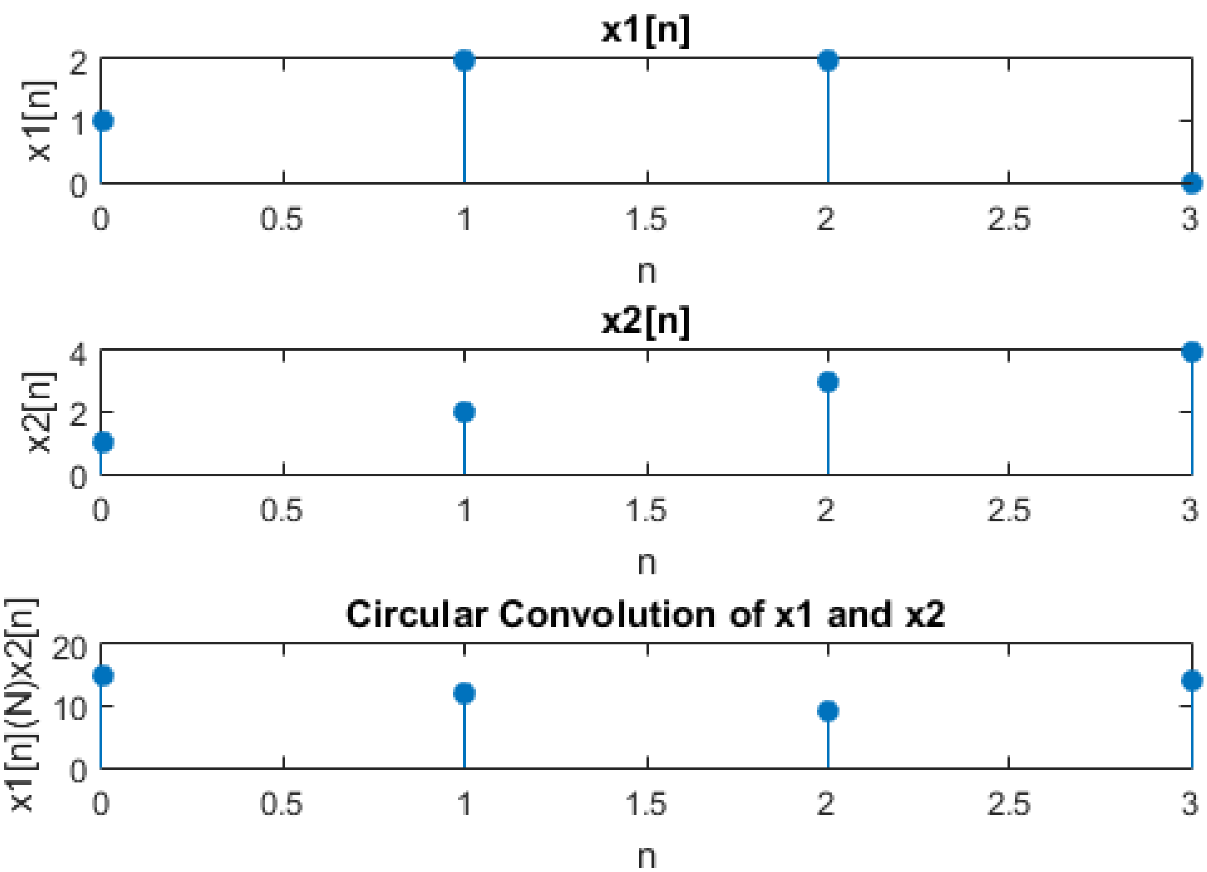
2. **Perform Circular Convolution:**
    a. **x1=[1 2 2 0] & x2=[1 2 3 4]**

```
%circular convolution ques 2
x1=[1 2 2 0]
x2=[1 2 3 4]

figure;
% x1[n]
subplot(3,1,1);
n=[0:3];
stem(n,x1,'filled');
title('x1[n]');
xlabel('n');
ylabel('x1[n]');

%x2[n]
subplot(3,1,2);
n=[0:3];
stem(n,x2,'filled');
title('x2[n]');
xlabel('n');
ylabel('x2[n]');
```

```
%convolution
subplot(3,1,3)
circ_conv = ifft(fft(x1).*fft(x2))
stem(n,circ_conv,'filled');
title('Circular Convolution of x1 and x2');
xlabel('n');
ylabel('x1[n](N)x2[n]');
```

Output:

```
x1 =
    1     2     2     0

x2 =
    1     2     3     4

circ_conv =
   15    12     9    14
```



Conclusion:
Thus, we performed linear and circular convolution of few signals and viewed the graph in Matlab.