SOMAIYA VIDYAVIHAR UNIVERSITY
K. J. Somaiya College of Engineering, Mumbai -77
(A Constituent College of Somaiya Vidyavihar University)

Batch: B1 **Roll No.: 16010121045**

Experiment No. 03

Title: Write a program to Compute linear and circular convolution of two discrete time signal sequences using Matlab.

Objective: To familiarize the beginner to MATLAB by introducing the basic features and commands of the program.

Expected Outcome of Experiment:

CO	Outcome
CO3	To understand the concept of convolution and perform different convolution operations on the given input signals.

Books/ Journals/ Websites referred:

1. http://www.mathworks.com/support/
2. www.math.mtu.edu/~msgocken/intro/intro.html
3. www.mccormick.northwestern.edu/docs/efirst/matlab.pdf
4. A.Nagoor Kani "Digital Signal Processing", 2nd Edition, TMH Education.

Pre Lab/ Prior Concepts:

Convolution

Discrete time convolution is a method of finding the response of a linear time invariant system. It is based on the concepts of linearity and time invariance and assumes that the system information is known in terms of its impulse response h[n].

Convolution is defined as

$$Y[n] = \sum_{k=-\infty}^{\infty} h[k]x\,[n-k] = h[n]*x[n]$$

Convolution consists of folding, shifting, Multiplication and summation operations.

Circular Convolution

Circular convolution between two length N sequences can be carried out as shown by the expression below:

$$y_C[n] = \sum_{m=0}^{N-1} g[m]h\left[\langle n-m\rangle_N\right]$$

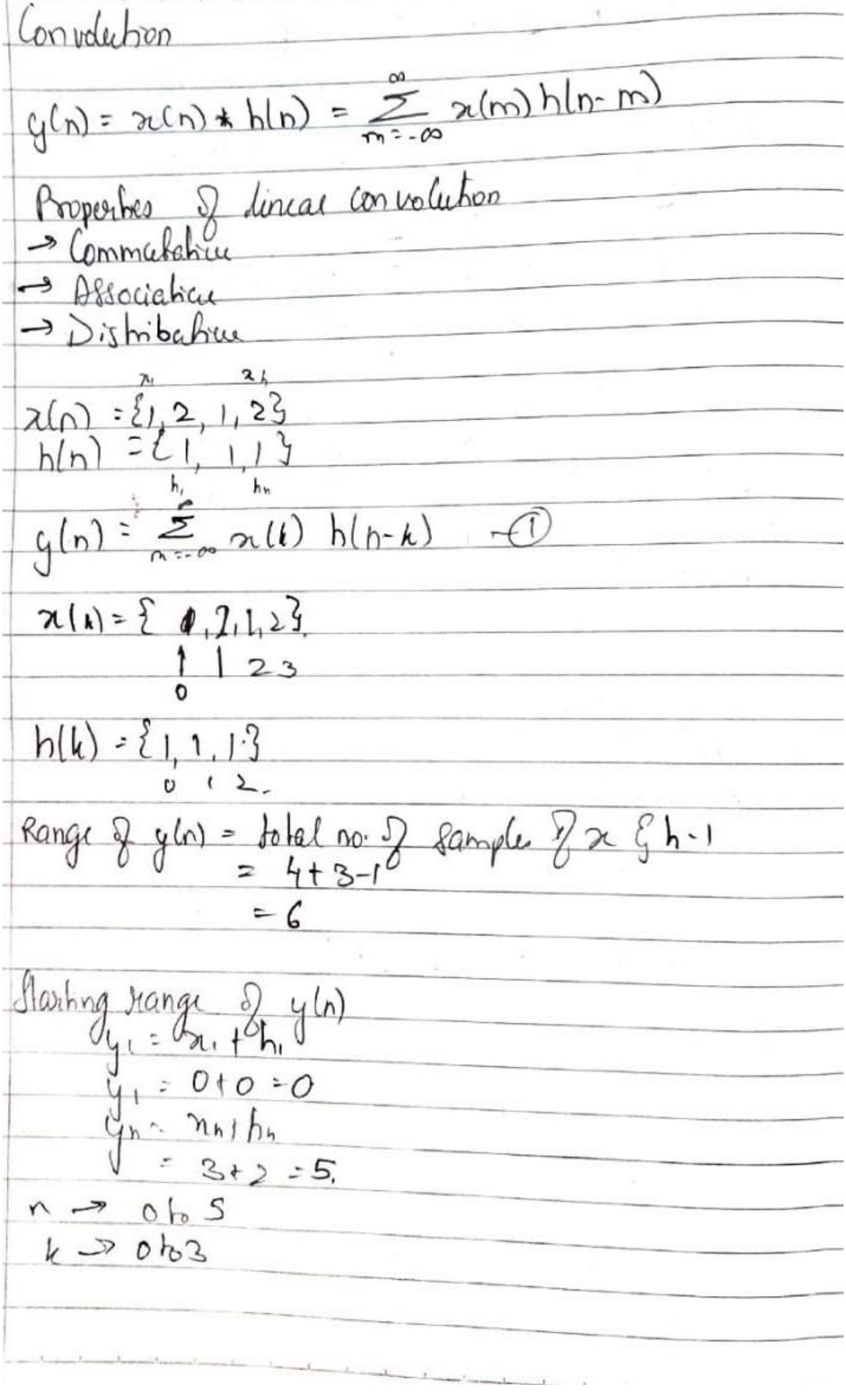
Since the above operation involves two length-N sequences it is referred to as the N-point circular convolution and denoted by:

$$y_C[n] = g[n] \,Ⓝ\, h[n]$$

As in linear convolution circular convolution is commutative.
i.e.

$$g[n] \,Ⓝ\, h[n] \equiv h[n] \,Ⓝ\, g[n]$$

Example Of Linear Convolution:



Convolution

$$y(n) = x(n) * h(n) = \sum_{m=-\infty}^{\infty} x(m)\, h(n-m)$$

Properties of linear convolution
→ Commutative
→ Associative
→ Distributive

$x(n) = \{\overset{x_l}{1}, 2, 1, \overset{x_h}{2}\}$
$h(n) = \{\underset{h_l}{1}, 1, \underset{h_h}{1}\}$

$$y(n) = \sum_{n=-\infty}^{\infty} x(k)\, h(n-k) \quad -①$$

$x(k) = \{\underset{\overset{\uparrow}{0}}{1}, \underset{1}{2}, \underset{2}{1}, \underset{3}{2}\}$

$h(k) = \{\underset{0}{1}, \underset{1}{1}, \underset{2}{1}\}$

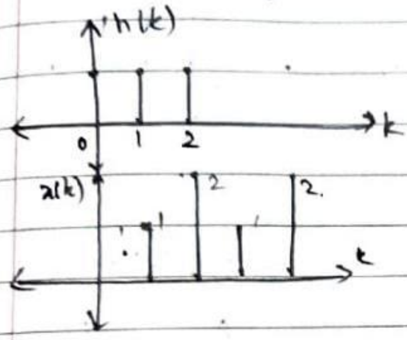
Range of y(n) = total no. of sample of x & h - 1
= 4+3-1
= 6

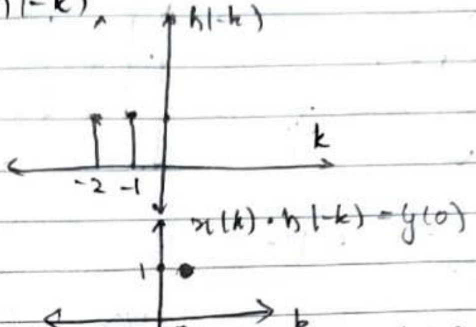
Starting range of y(n)
$y_l = x_l + h_l$
$y_l = 0+0 = 0$
$y_h = x_h + h_h$
$= 3+2 = 5.$
n → 0 to 5
k → 0 to 3

1) Graphical Method

n = 0

$y(0) = \sum_{k=0}^{3} x(k)\, h(-k)$

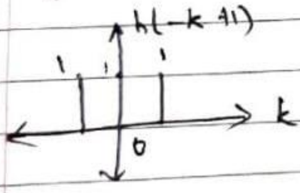


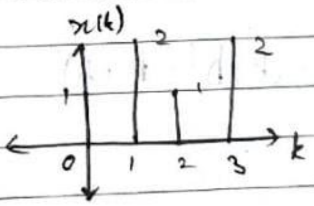


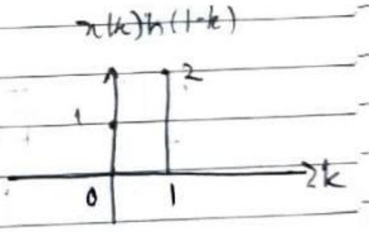
y(0) = 1

n = 1

$y(1) = \sum_{k=0}^{3} x(k)\, h(1-k)$



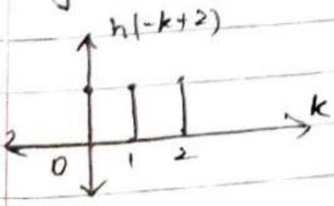


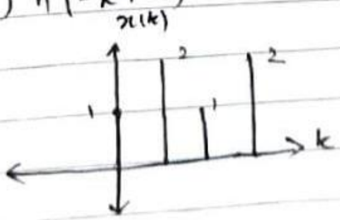


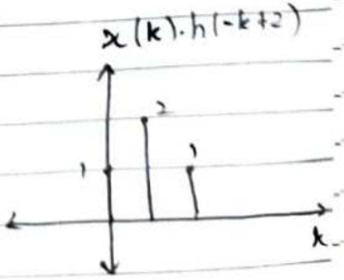
$\therefore y(1) = 1+2 = 3.$

n = 2.

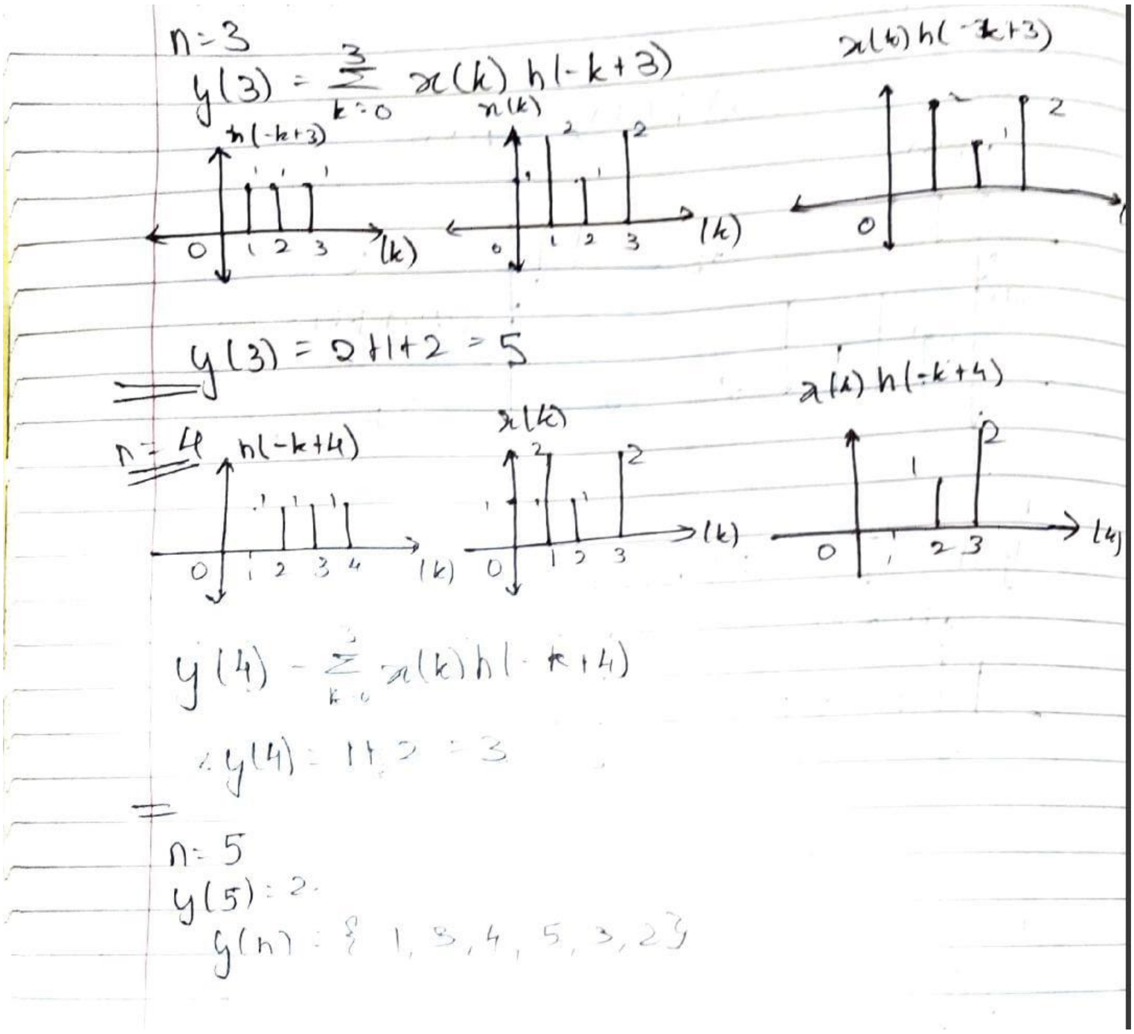
$y(2) = \sum_{k=0}^{3} x(k)\, h(-k+2)$







$\therefore y(2) = 1+2+1 = 4$



n=3

$y(3) = \sum_{k=0}^{3} x(k)\, h(-k+3)$

$y(3) = 0+1+2 = 5$

n=4

$y(4) = \sum_{k=0}^{3} x(k)\, h(-k+4)$

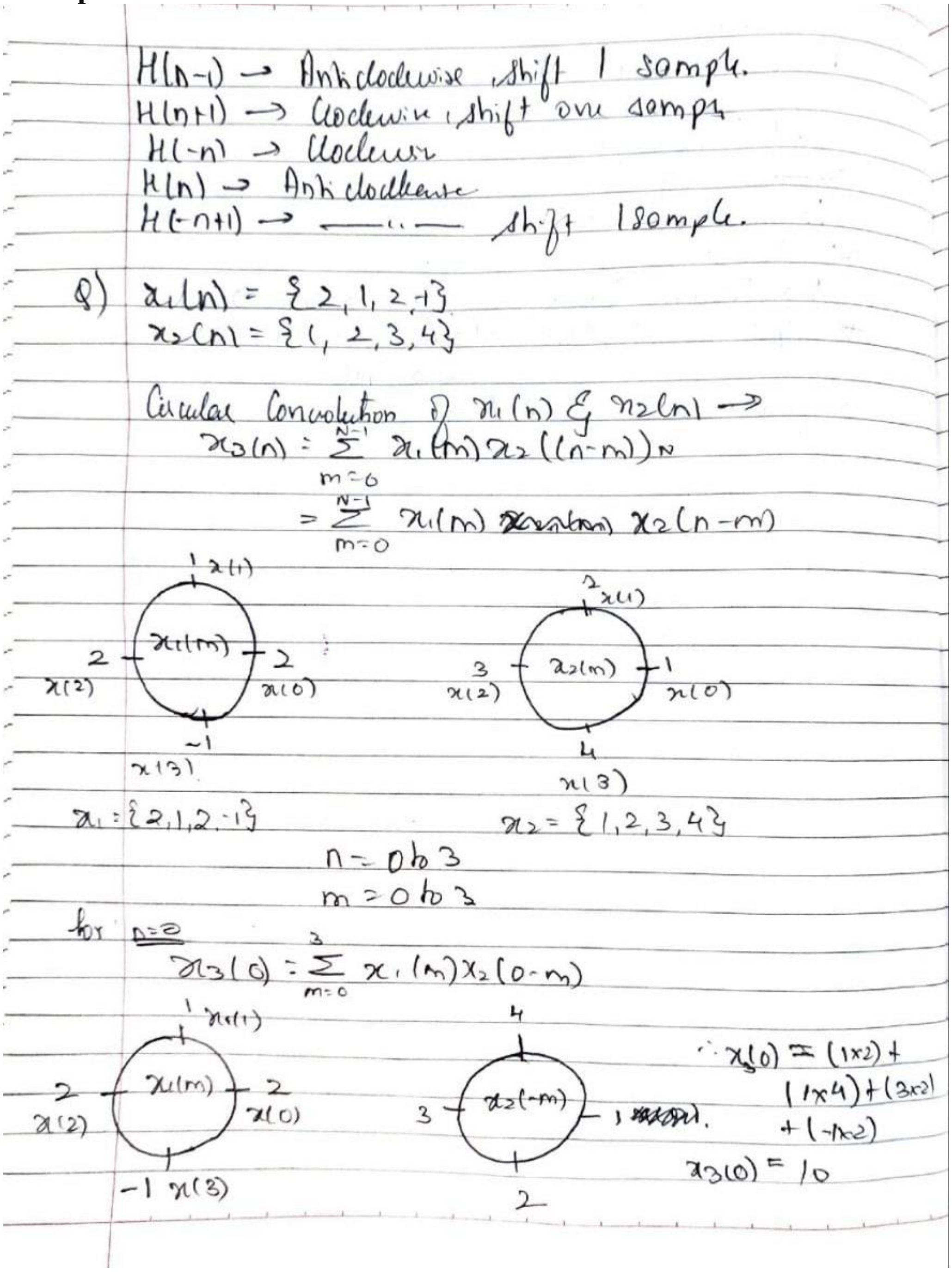
$y(4) = 1+2 = 3$

n=5

$y(5) = 2$

$y(n) = \{1, 3, 4, 5, 3, 2\}$

Example Of Circular Convolution:



$H(n-1)$ → Anticlockwise shift 1 sample.
$H(n+1)$ → Clockwise shift one sample
$H(-n)$ → Clockwise
$H(n)$ → Anticlockwise
$H(-n+1)$ → —— " —— shift 1 sample.

Q) $x_1(n) = \{2, 1, 2, -1\}$
$x_2(n) = \{1, 2, 3, 4\}$

Circular Convolution of $x_1(n)$ & $x_2(n)$ →

$$x_3(n) = \sum_{m=0}^{N-1} x_1(m)\, x_2((n-m))_N$$

$$= \sum_{m=0}^{N-1} x_1(m)\, x_2(n-m)$$

1 $x(1)$
2 $x(2)$
$x_1(m)$
2 $x(0)$
-1 $x(3)$

2 $x(1)$
3 $x(2)$
$x_2(m)$
1 $x(0)$
4 $x(3)$

$x_1 = \{2, 1, 2, -1\}$ $x_2 = \{1, 2, 3, 4\}$

$n = 0$ to 3
$m = 0$ to 3

for $\underline{n=0}$

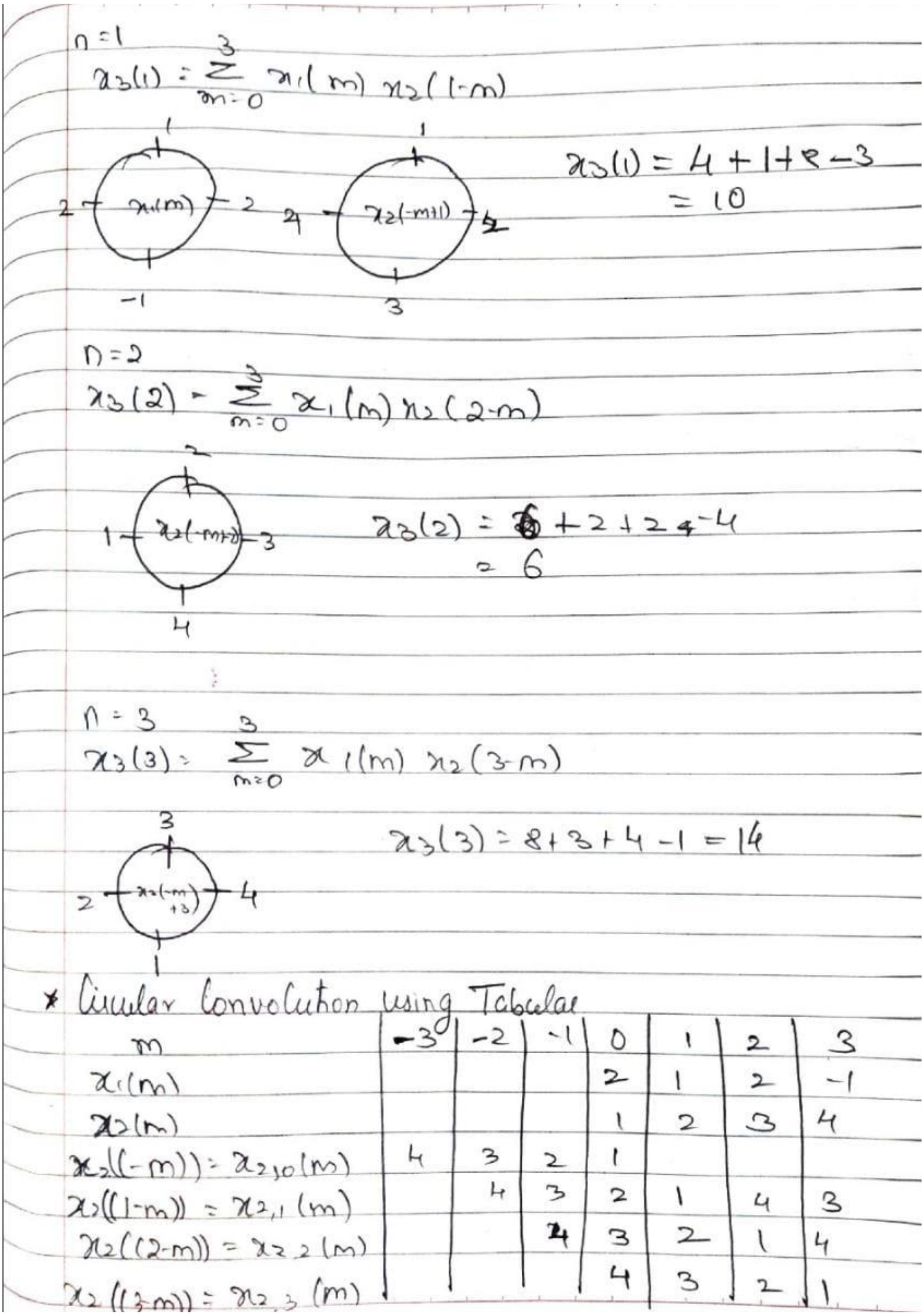
$$x_3(0) = \sum_{m=0}^{3} x_1(m)\, x_2(0-m)$$

1 $x_1(1)$
2 $x(2)$
$x_1(m)$
2 $x(0)$
-1 $x(3)$

4
3
$x_2(-m)$
1
2

$\therefore x_3(0) = (1\times2) + (1\times4) + (3\times2) + (-1\times2)$

$x_3(0) = 10$



$n = 1$

$$x_3(1) = \sum_{m=0}^{3} x_1(m)\, x_2(1-m)$$

$x_3(1) = 4 + 1 + 8 - 3$
$= 10$

$n = 2$

$$x_3(2) = \sum_{m=0}^{3} x_1(m)\, x_2(2-m)$$

$x_3(2) = 6 + 2 + 2_4 - 4$
$= 6$

$n = 3$

$$x_3(3) = \sum_{m=0}^{3} x_1(m)\, x_2(3-m)$$

$x_3(3) = 8 + 3 + 4 - 1 = 14$

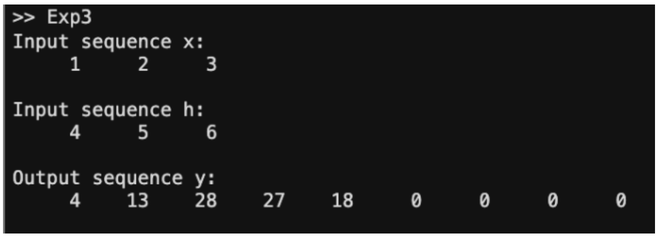
* Circular Convolution using Tabular

m	-3	-2	-1	0	1	2	3
$x_1(m)$				2	1	2	-1
$x_2(m)$				1	2	3	4
$x_2((-m)) = x_{2,0}(m)$	4	3	2	1			
$x_2((1-m)) = x_{2,1}(m)$		4	3	2	1	4	3
$x_2((2-m)) = x_{2,2}(m)$			4	3	2	1	4
$x_2((3-m)) = x_{2,3}(m)$				4	3	2	1

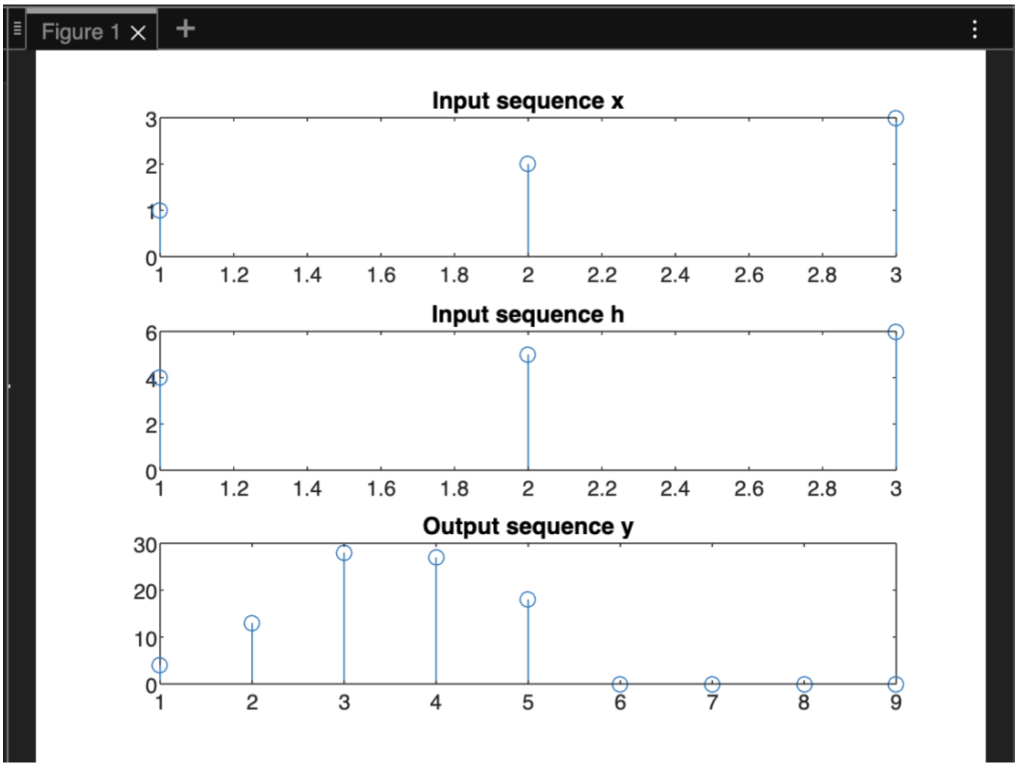
Implementation details along with screenshots:

Linear convolution:

```
x = [1, 2, 3];
h = [4, 5, 6];
linearConvolution(x, h);
% circularConvolution(x,h);
function result = linearConvolution(x, h)
   len_x = length(x);
   len_h = length(h);
   len_result = len_x + len_h - 1;

   x_padded = [x, zeros(1, len_result - len_x)];
   h_padded = [h, zeros(1, len_result - len_h)];

   result = conv(x_padded, h_padded);

   disp('Input sequence x:');
   disp(x);
   disp('Input sequence h:');
   disp(h);
   disp('Output sequence y:');
   disp(result);

   figure;
   subplot(3, 1, 1); stem(x); title('Input sequence x');
   subplot(3, 1, 2); stem(h); title('Input sequence h');
   subplot(3, 1, 3); stem(result); title('Output sequence y');
end
```



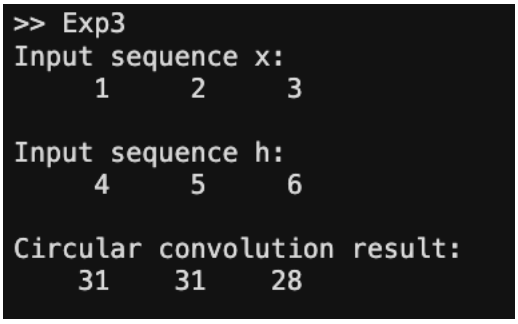
```
>> Exp3
Input sequence x:
     1     2     3

Input sequence h:
     4     5     6

Output sequence y:
     4    13    28    27    18     0     0     0     0
```



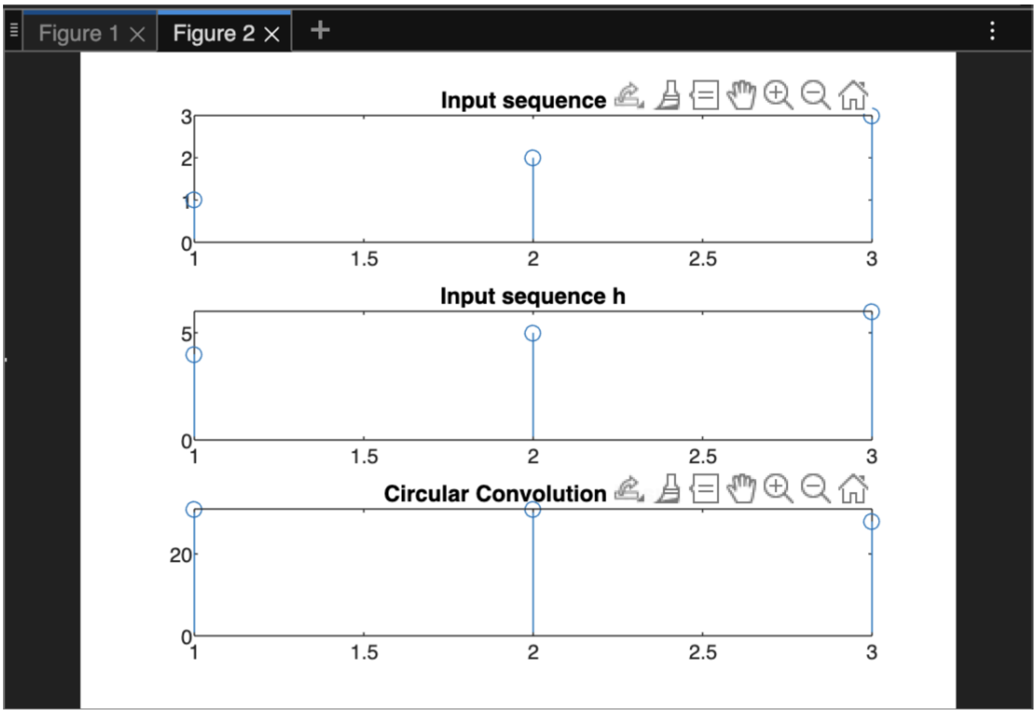
Circular convolution:

```
x = [1, 2, 3];
h = [4, 5, 6];
circularConvolution(x, h);
function result = circularConvolution(x, h)
  len_x = length(x);
  len_h = length(h);

  if len_x ~= len_h
    error('Input sequences must have the same length for circular convolution.');
  end

  result = ifft(fft(x) .* fft(h));

  disp('Input sequence x:');
  disp(x);

  disp('Input sequence h:');
  disp(h);

  disp('Circular convolution result:');
  disp(result);

  figure;
  subplot(3,1,1); stem(x); title('Input sequence x');
```

```
    subplot(3,1,2); stem(h); title('Input sequence h');
    subplot(3,1,3); stem(result); title('Circular Convolution Result');
end
```



```
>> Exp3
Input sequence x:
     1     2     3

Input sequence h:
     4     5     6

Circular convolution result:
    31    31    28
```



x

Conclusion: Thus we have implemented programs to compute convolution and correlation using MATLAB.

Post Lab Descriptive Questions

1. Explain the role of convolution in signal processing.

Convolution is a fundamental operation in signal processing that plays a central role in many applications. It is used to model the relationship between two signals, where one signal (often called the input signal) is transformed by another signal (often called the impulse response or kernel) to produce an output signal.

The role of convolution in signal processing can be summarized as follows:
Filtering: Convolution is used for filtering signals by removing or attenuating unwanted frequency components or noise. This is done by convolving the input signal with a filter kernel that has a specific frequency response, such as a low-pass, high-pass, or band-pass filter.
System modelling: Convolution is used to model the behaviour of linear time-invariant (LTI) systems, which are systems that produce an output signal that is a scaled and delayed version of the input signal. The impulse response of an LTI system can be convolved with any input signal to obtain the output signal.
Signal analysis: Convolution is used for signal analysis to extract information about the frequency content or time-domain properties of a signal. This is done by convolving the signal with a window function, such as a Gaussian or a rectangular window.
Image processing: Convolution is used extensively in image processing to perform operations such as blurring, edge detection, and feature extraction. This is done by convolving an image with a kernel that is designed to perform a specific operation.

Overall, convolution is a powerful tool that is used in many different areas of signal processing to analyse, filter, and transform signals. Its versatility and usefulness make it a fundamental concept that is essential to the field of signal processing.

2. Explain the difference between linear and circular convolution?

Linear convolution and circular convolution are two types of convolution operations that are used in signal processing. The main difference between them is the way they handle the boundaries of the signals being convolved.

Linear convolution is used when the signals being convolved are finite and have zero values outside their boundaries. In this case, the convolution operation is performed by aligning the signals and sliding one signal over the other, multiplying the corresponding samples and summing the results. To prevent aliasing or truncation of the output signal, the signals are often padded with zeros before convolution. The output of linear convolution is a sequence that is longer than the input signals.

Circular convolution, on the other hand, is used when the signals being convolved are considered periodic, meaning that they repeat infinitely in both directions. In this case, the convolution operation is performed by treating the signals as circular, so that the last sample of one signal is convolved with the first sample of the other signal. The output of circular convolution has the same length as the input signals, so there is no need for zero-padding. Circular convolution is often used in applications where the signals have a cyclic or periodic nature, such as in digital signal processing or image processing.

In summary, the main difference between linear and circular convolution is in the way they handle the boundaries of the signals being convolved. Linear convolution is used for finite signals with zero-padding to prevent aliasing, while circular convolution is used for periodic signals without zero-padding.

3. Explain with the help of an example the steps required to transform linear convolution with circular convolution and vice-versa.

To transform linear convolution into circular convolution or vice versa, we need to follow certain steps. Let's first consider how to transform linear convolution into circular convolution:

Transforming Linear Convolution into Circular Convolution:

Suppose we have two sequences x[n] and h[n] and we want to transform their linear convolution into circular convolution. The steps involved in this transformation are as follows:

Step 1: Pad the sequences x[n] and h[n] with zeros to make them of length N + M - 1, where N is the length of x[n] and M is the length of h[n]. This is to ensure that the linear convolution is equivalent to circular convolution.

Step 2: Take the DFT (Discrete Fourier Transform) of the two sequences x[n] and h[n] using an FFT (Fast Fourier Transform) algorithm.

Step 3: Multiply the DFT of the two sequences element-wise to obtain the DFT of the circular convolution.

Step 4: Take the inverse DFT of the result from step 3 using an IFFT (Inverse Fast Fourier Transform) algorithm to obtain the circular convolution sequence.

For example, let's consider the following sequences:

x[n] = {1, 2, 3}

h[n] = {2, 1}

To transform their linear convolution into circular convolution, we first pad the sequences with zeros:

x[n] = {1, 2, 3, 0}

h[n] = {2, 1, 0, 0}

Then we take their DFT using an FFT algorithm:

X[k] = {6, -1+1.73j, -1-1.73j, 0}

H[k] = {3, 1-1j, 1+1j, 0}

Next, we multiply the DFT of the two sequences element-wise:

Y[k] = X[k] * H[k] = {18, -4+4.59j, -4-4.59j, 0}

Finally, we take the inverse DFT of the result from step 3 using an IFFT algorithm:

y[n] = IDFT(Y[k]) = {2.5, 0.5, 5, 0}

Therefore, the circular convolution of x[n] and h[n] is given by the sequence y[n] = {2.5, 0.5, 5, 0}.

Transforming Circular Convolution into Linear Convolution:

Similarly, we can transform circular convolution into linear convolution by following the below steps:

Step 1: Take the DFT of the circular convolution sequence using an FFT algorithm.

Step 2: Divide the DFT of the circular convolution sequence by the DFT of one of the

input sequences, either x[n] or h[n].

Step 3: Take the inverse DFT of the result from step 2 using an IFFT algorithm to obtain the linear convolution sequence.

For example, let's consider the following circular convolution sequence:

y[n] = {2.5, 0.5, 5, 0}

To transform this circular convolution sequence into linear convolution, we first take its

DFT:

Y[k] = {8, 2-2j, 2+2j, 0}

Next, we divide the DFT of the circular convolution sequence by the DFT of one of the

input sequences, let's say x[n]:

X[k] = {6, -1+1.73j, -1-1.73j, 0}

H[k] = {2, 1-1j, 1+1j, 0}

Therefore, we divide Y[k] by X[k]

Date: ______________ **Signature of faculty in-charge**