

|  |
| --- |
| **Title:** To compute linear and circular convolution of two discrete time sequences using Matlab. |

**Objective:** To familiarize the beginnerto 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 response of 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

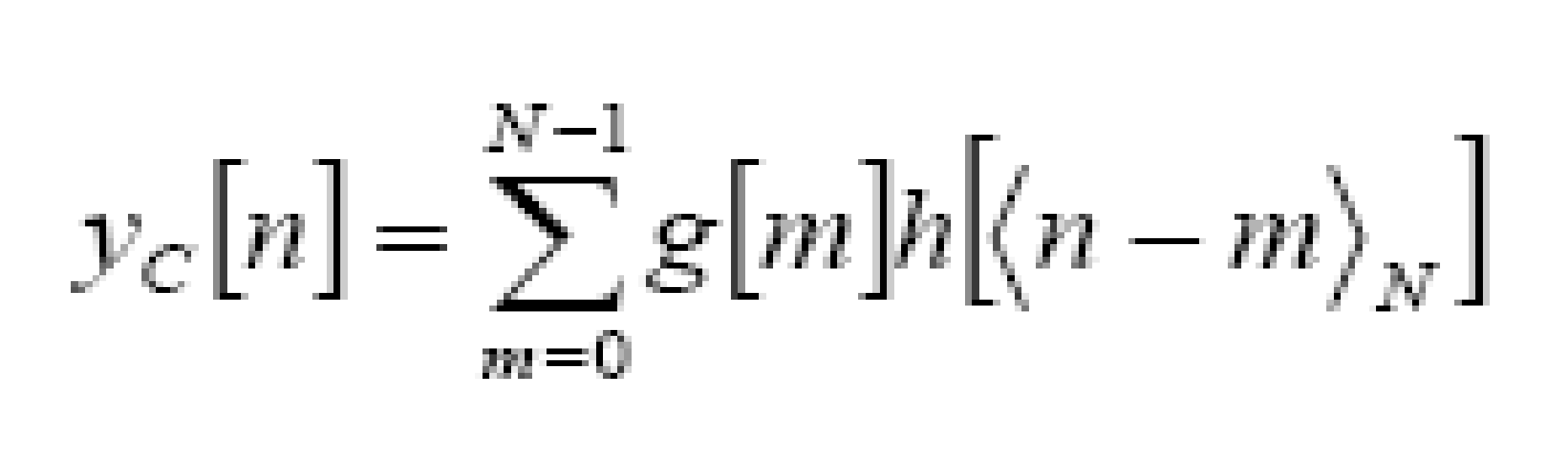
∞

Y[n] = Σ h[k]x [n-k] =h[n]\*x[n] k=-∞

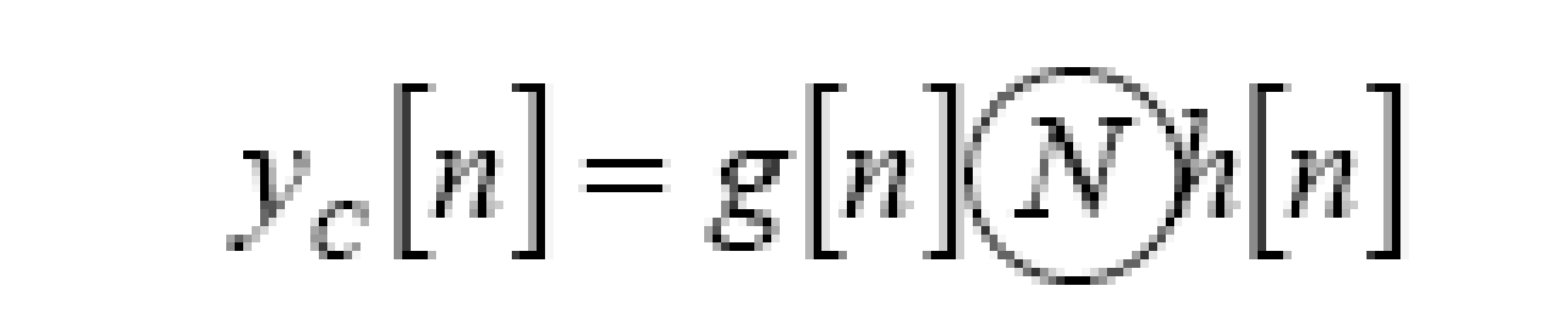
Convolution consists of folding, shifting, Multiplication and summation operations.

**Circular Convolution**

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

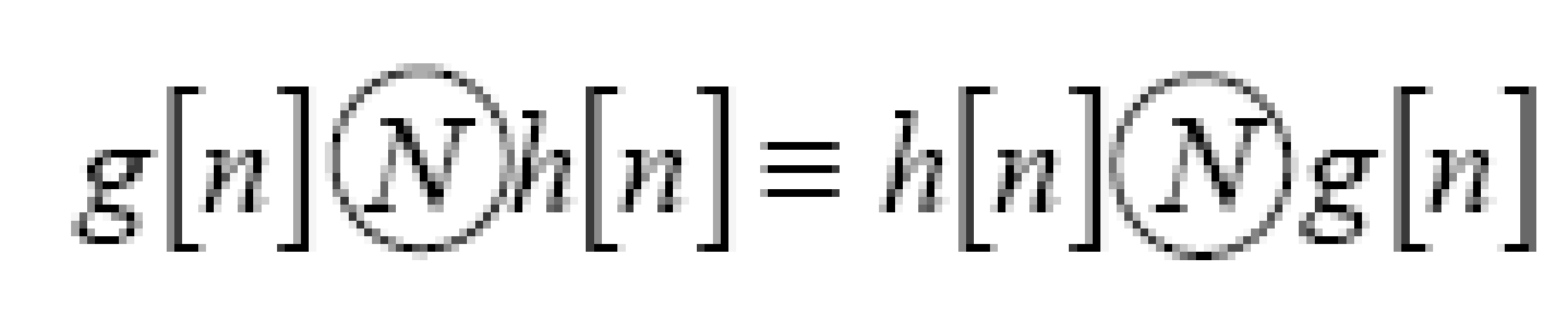


Since the above operation involves two length-N sequences it is referred to as the N-point circular convolution and denoted by:

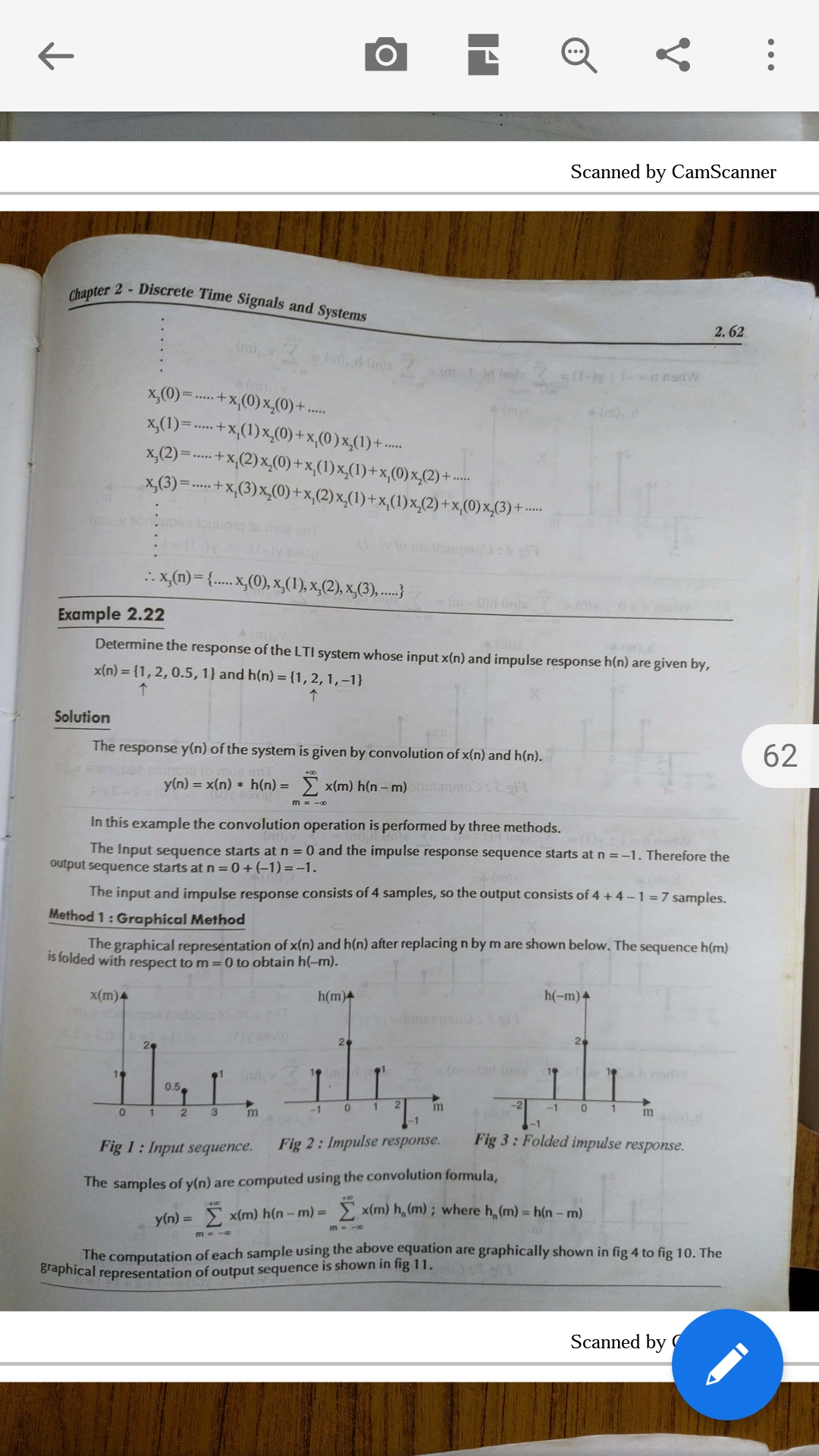


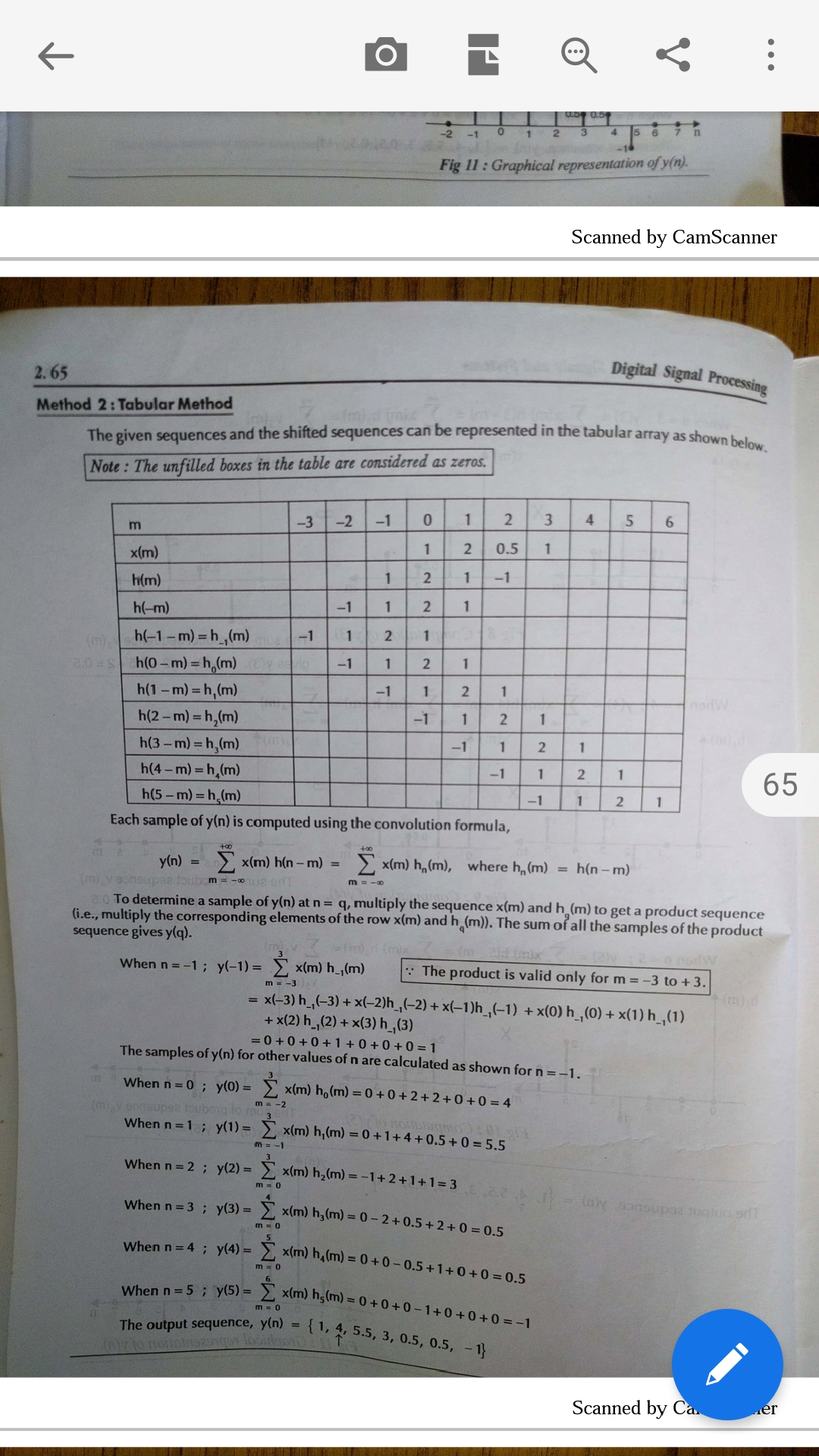
As in linear convolution circular convolution is commutative.

i.e.

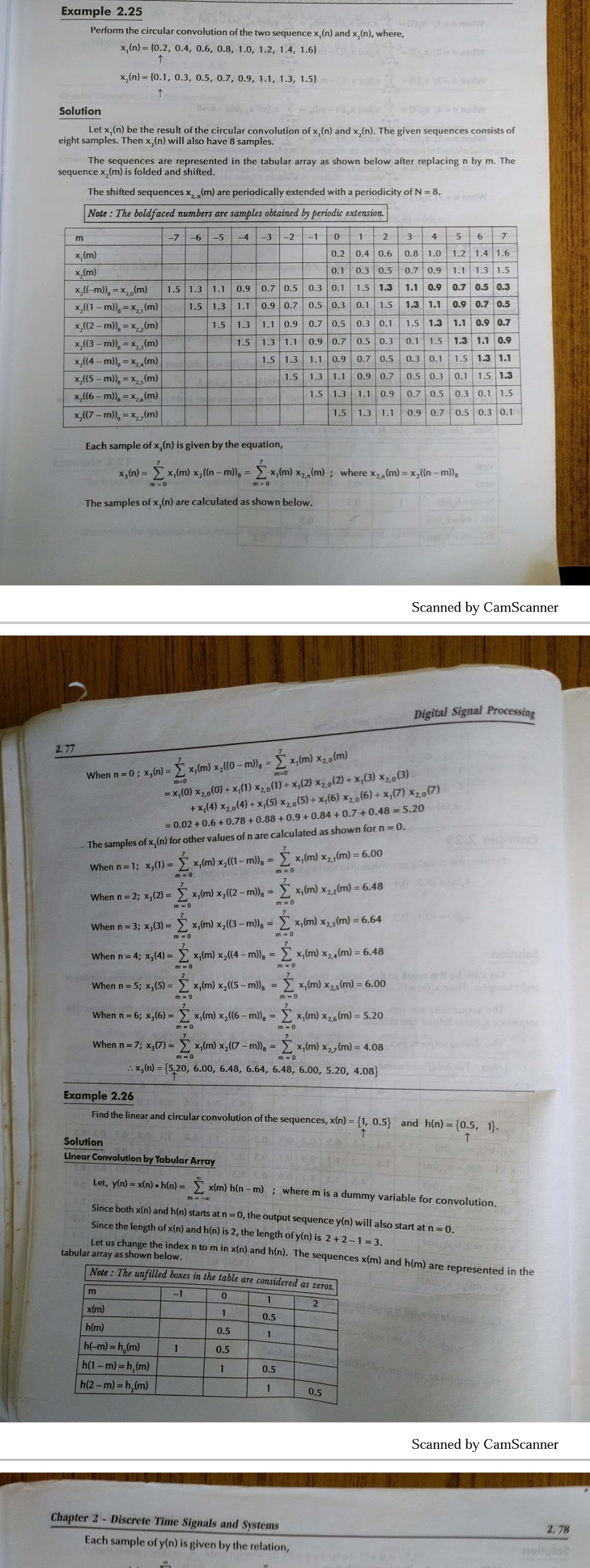


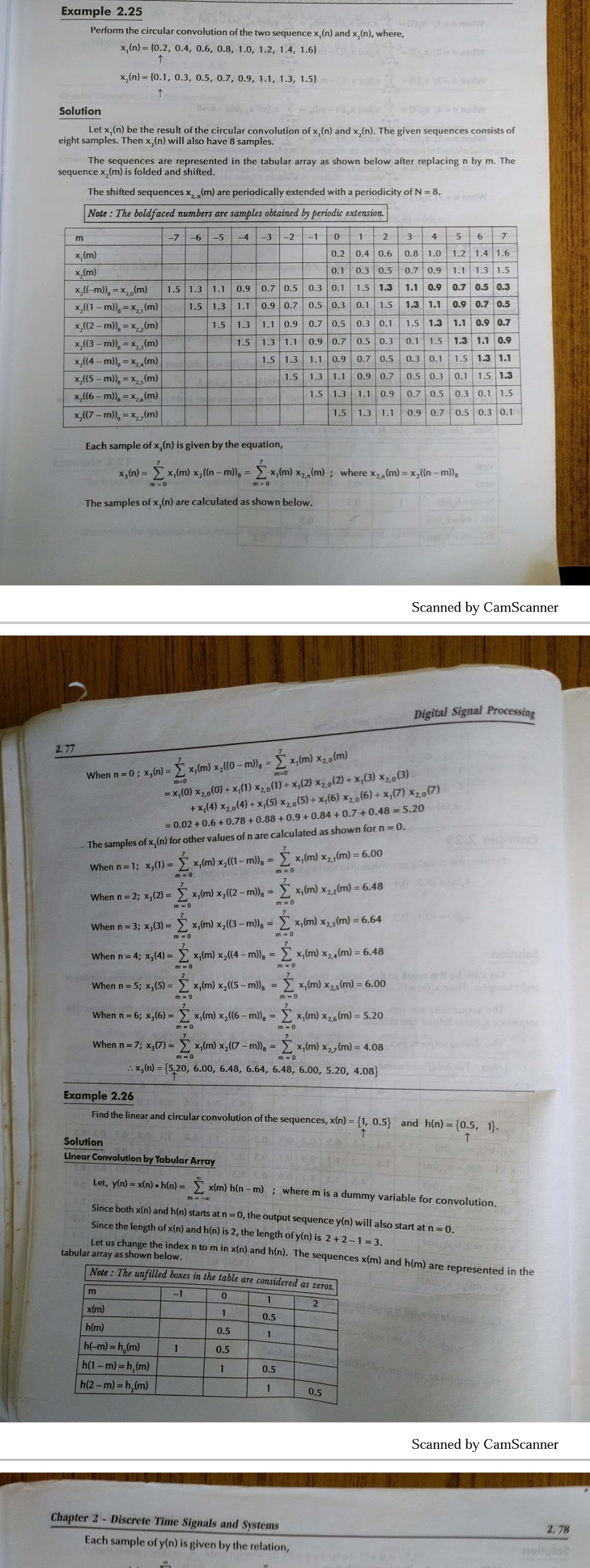
**Example Of Linear Convolution:**

****

****

**Example Of Circular Convolution:**

****

****

**Implementation details along with screenshots:**

**Linear Convolution**

**Code**

x = [1,2,0.5,1]

n1 = 0

h = [1,2,1,-1]

n2 = -1

N1 = size(x,2);

N2 = size(h,2);

N = N1+N2-1;

h2 = flip(h);

n3 = -(N2+n2-1);

ni = n1 + n2;

nf = ni + N - 1;

mstart = n3+ni;

mend = mstart+N-1;

rows = 4+N;

columns = abs(mstart)+mend+N2;

t = zeros(rows, columns);

for i = 1:columns

t(1,i) = mstart+i-1;

end

for i = 1:N1

t(2,abs(mstart)+n1+i) = x(i);

end

for i = 1:N2

t(3,abs(mstart)+n2+i) = h(i);

end

for i = 1:N2

t(4,abs(mstart)+n3+i) = h2(i);

end

for j = 1:N

k=1;

for i = j:j+N2-1

t(j+4,i) = h2(k);

k=k+1;

end

end

t

y=zeros(1,N);

for j = 1:N

for i = 1:N1

y(j) = y(j)+t(2,abs(mstart)+n1+i)\*t(4+j,abs(mstart)+n1+i);

end

end

y

ni

subplot(2,2,1);

x1=n1:n1+N1-1;

stem(x1,x);

xlabel('n');

ylabel('x(n)');

title('x(n)');

subplot(2,2,2);

x2=n2:n2+N2-1;

stem(x2,h);

xlabel('n');

ylabel('h(n)');

title('h(n)');

subplot(2,2,3);

x3=n3:n3+N2-1;

stem(x3,h2);

xlabel('n');

ylabel('h(-n)');

title('h(-n)');

subplot(2,2,4);

x4=ni:nf;

stem(x4,y);

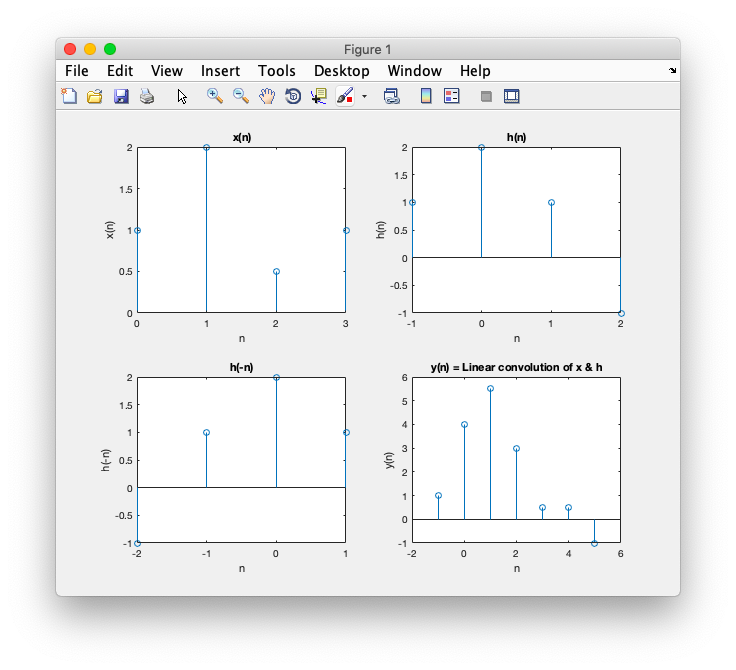
xlabel('n');

ylabel('y(n)');

title('y(n) = Linear convolution of x & h');

**Output**





**Circular Convolution**

**Code**

x = [2,1,2,-1]

n1 = 0

h = [1,2,3,4]

n2 = 0

N1 = size(x,2);

N2 = N1;

N = N1;

h2 = flip(circshift(h,N-1));

n3 = 0;

ni = 0;

nf = N-1;

mstart = -(N-1);

mend = N-1;

rows = 3+N;

columns = abs(mstart)+mend+1;

t = zeros(rows, columns);

for i = 1:columns

t(1,i) = mstart+i-1;

end

for i = 1:N

t(2,abs(mstart)+n1+i) = x(i);

end

for i = 1:N

t(3,abs(mstart)+n2+i) = h(i);

end

for j = 1:N

k=1;

for i = j:columns

in = mod(k,N)+1;

t(j+3,i) = h2(in);

k=k+1;

end

end

t

y=zeros(1,N);

for j = 1:N

for i = 1:N

y(j) = y(j)+t(2,i+N-1)\*t(3+j,i+N-1);

end

end

y

ni

subplot(2,2,1);

x1=ni:nf;

stem(x1,x);

xlabel('n');

ylabel('x(n)');

title('x(n)');

subplot(2,2,2);

x2=ni:nf;

stem(x2,h);

xlabel('n');

ylabel('h(n)');

title('h(n)');

subplot(2,2,3);

x3=ni:nf;

stem(x3,h2);

xlabel('n');

ylabel('h(-n)');

title('h(-n)');

subplot(2,2,4);

x4=ni:nf;

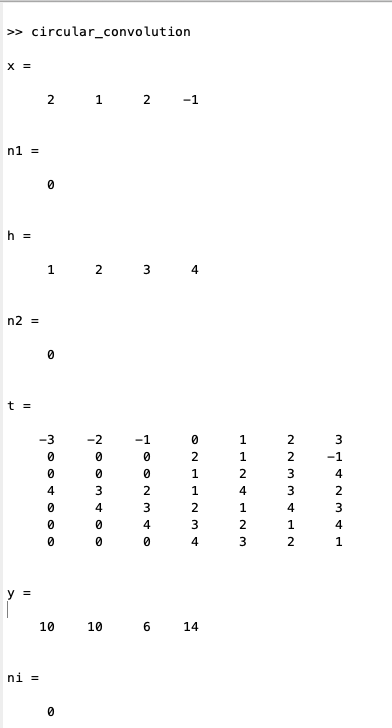
stem(x4,y);

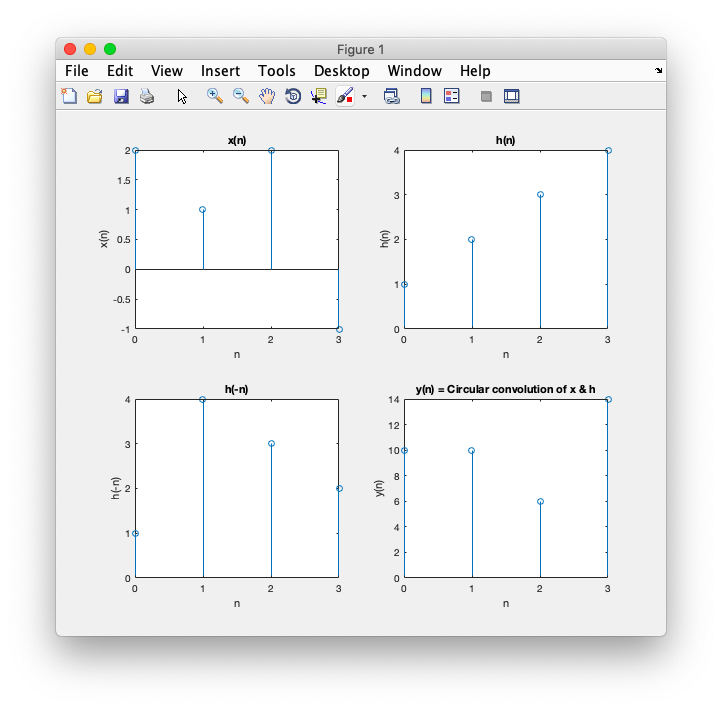
xlabel('n');

ylabel('y(n)');

title('y(n) = Circular convolution of x & h');

**Output**





**Conclusion:-**

Thus, we have successfully computed linear and circular convolution of two discrete time sequences using Matlab.

**Date: \_\_08/03/2019\_\_ Signature of faculty in-charge**

**Post Lab Descriptive Questions**

* 1. **Explain the role of convolution in signal processing.**

Ans.

Convolution of signals occurs in several contexts in signal processing. The input-

output characteristic of linear time-invariant (LTI) systems, the most widely used type

of system in signal processing, is described entirely in terms of the impulse response of

the system. The impulse response is the output of the system due to an impulse input

signal. Given the input signal, the output of an LTI system is the convolution of the

input signal with the impulse response of the system. Hence, convolution plays a key

role in relating the input and output signals of an LTI system.

Convolution also arises when we analyze the effect of multiplying two signals in the

frequency domain. If we multiply two signals in time, the Fourier representation for

the product is the convolution of the Fourier representations of the individual signals.

This type of analysis occurs when the discrete Fourier transform is applied to truncated

(finite length) signals, and in the design of finite impulse response (FIR) filters.

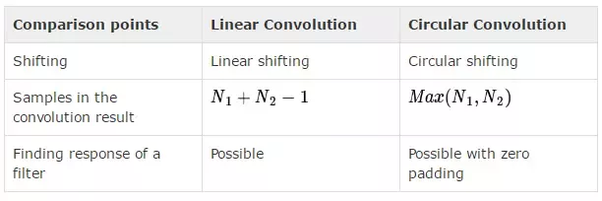
* 1. **Explain the difference between linear and circular convolution?**

Ans.

Linear convolution is the basic operation to calculate the output for any linear time invariant system given its input and its impulse response. Circular convolution is the same thing but considering that the support of the signal is periodic (as in a circle, hence the name).

Linear convolution takes two functions of an independent variable, and convolves them using the convolution sum formula. Basically it is a correlation of one function with the time-reversed version of the other function.

Circular convolution is only defined for finite length functions (usually equal in length), continuous or discrete in time. In circular convolution, it is as if the finite length functions repeat in time, periodically. Because the input functions are now periodic, the convolved output is also periodic and so the convolved output is fully specified by one of its periods.

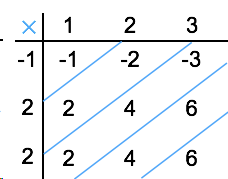


* 1. **Explain with the help of an example the steps required to transform linear convolution with circular convolution and vice-versa.**

Ans.

**Transforming Linear Convolution to Circular Convolution**

Convolute two sequences x[n] = {1,2,3} & h[n] = {-1,2,2} using circular convolution



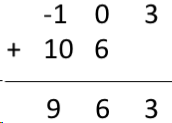
**Linear Convolution**

Normal Convoluted output y[n] = [ -1, -2+2, -3+4+2, 6+4, 6].

= [-1, 0, 3, 10, 6]

Here x[n] contains 3 samples and h[n] also has 3 samples. Hence the resulting sequence obtained by circular convolution must have max[3,3]= 3 samples.

Now to get periodic convolution result, 1st 3 samples [as the period is 3] of normal convolution is same next two samples are added to 1st samples as shown below:



∴ **Circular convolution result *y*[*n*]=[9 6 3]**

**Transforming Circular Convolution to Linear Convolution**

x1(n) = {1, 2, 3, 4}

and x2 (n) = {1, 2, 1, 2}

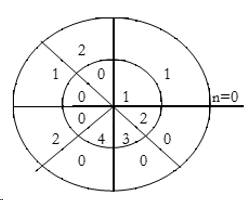
L=4, M=4

Length of y(n) = L+M-1=4+4-1=7

∴,x1(n) = {1, 2, 3, 4, 0, 0, 0}

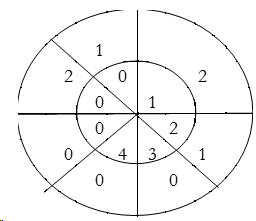
& x2(n) = {1, 2, 1, 2, 0, 0, 0}

For y(0),



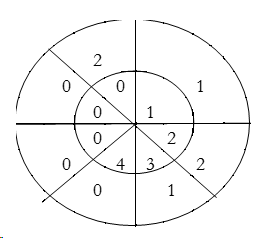
∴ y(0)= 1×1=1

For y(1),



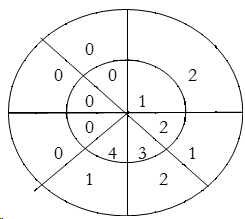
∴ y(1)= 2×1+1×2=4

For y(2),



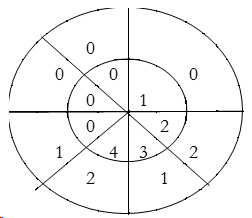
∴ y(2)= 1×1+2×2+3×1=8

For y(3),



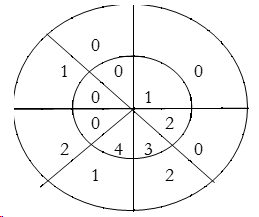
∴ y(3)=1×2+2×1+3×2+4×1=14

For y(4),



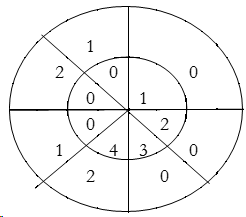
∴ y(4)= 4×2+3×1+2×2=15

For y(5),



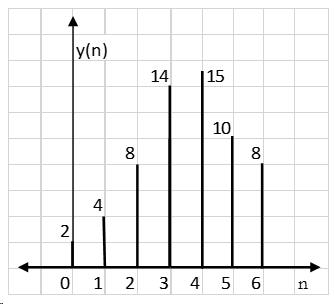
∴ y(5) = 4×1+3×2=10

For y(6),



∴ y(6) = 4×2=8

∴ y(n) = {2, 4, 8, 14, 15, 10, 8}



∴ **Linear convolution result y(n) = {2, 4, 8, 14, 15, 10, 8}**