**Batch: A1 Roll No.: 1911004**

**Experiment No. 4**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of the Staff In-charge with date**

|  |
| --- |
| **Title:** Write a program to Compute linear and circular convolution of two discrete time signal 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:

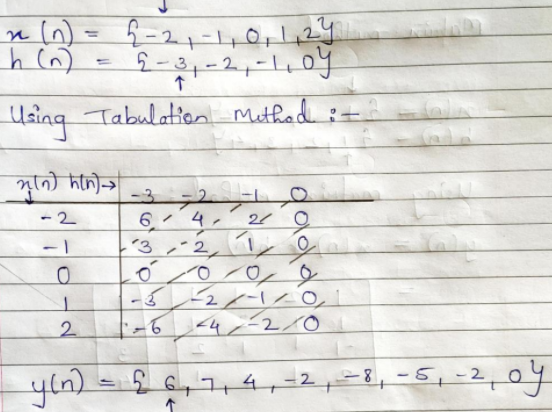


As in linear convolution circular convolution is commutative.

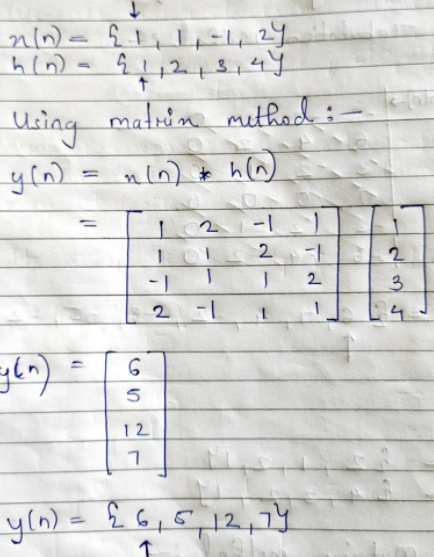
i.e.



**Example Of Linear Convolution:**



**Example Of Circular Convolution:**



**Implementation details along with screenshots:**

**Linear Convolution:**

**Code:**

n1= input('Enter number of discrete points for signal 1: ');

n2= input('Enter number of discrete points for signal 2: ');

p1= input('Enter zero position for signal 1 or press 1 for default index: ');

p2= input('Enter zero position for signal 2 or press 1 for default index: ');

xi= [];

hi= [];

if p1==1

xi=(0:n1-1);

else

xi=(-(p1-1):1:(n1-p1));

end

if p2==1

hi=(0:n2-1);

else

hi=(-(p2-1):1:(n2-p2));

end

xn= zeros(1,n1);

hn= zeros(1,n2);

disp("Enter values of signal 1: ");

for i=1:n1

xn(i) = input('');

end

disp("Index of signal 1: ");

disp(xi);

disp("Values of Signal 1: ");

disp(xn);

disp("Enter values of signal 2: ");

for i=1:n2

hn(i) = input('');

end

disp("Index of signal 2: ");

disp(hi);

disp("Values of Signal 2: ");

disp(hn);

total = n1 + n2 - 1; %Number of samples

fprintf("Total values in convoluted signal: %d",total);

pos = xi(1) + hi(1);

fprintf("\nStarting index of convoluted signal: %d\n",pos);

conv\_x = (pos:1:(pos + total - 1));

conv\_y = zeros(1,total);

xn\_p = [xn,zeros(1,n2-1)];

hn\_p = [hn,zeros(1,n1-1)];

for i=1:total

for j =1:i

conv\_y(i) = conv\_y(i)+xn\_p(j)\*hn\_p (i-j+1);

end

end

disp("Index of convoluted signal: ");

disp(conv\_x);

disp("Values of convoluted signal: ");

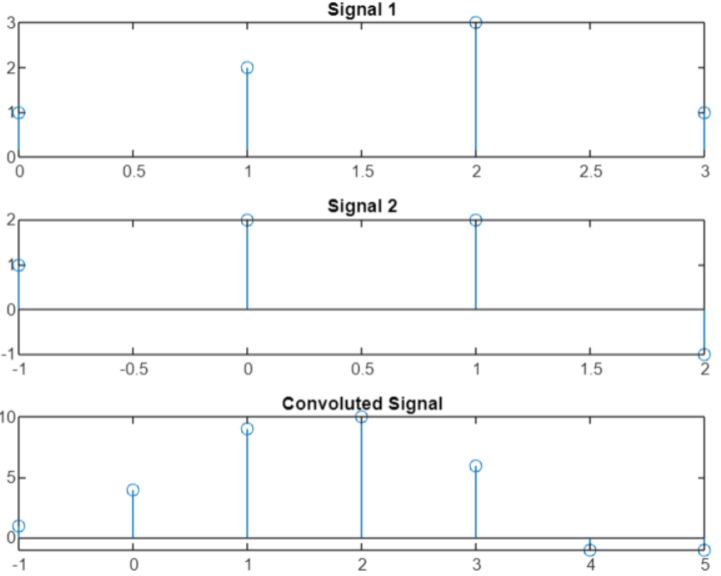
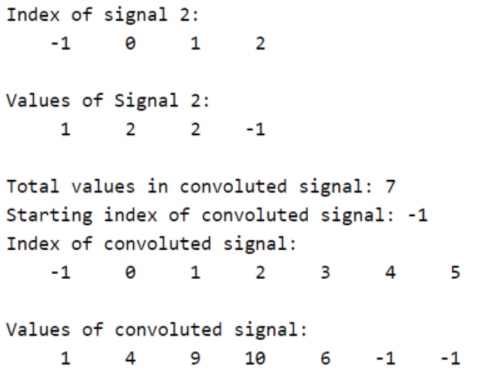
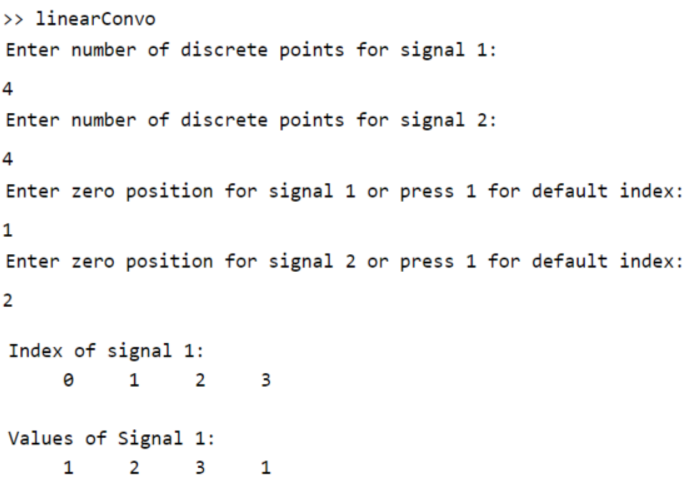
disp(conv\_y);

subplot(3,1,1), stem(xi,xn), title('Signal 1');

subplot(3,1,2), stem(hi,hn), title('Signal 2');

subplot(3,1,3), stem(conv\_x,conv\_y), title('Convoluted Signal');

**Output:**



**Circular Convolution:**

**Code:**

n1= input('Enter number of discrete points for signal 1: ');

n2= input('Enter number of discrete points for signal 2: ');

p1= input('Enter zero position for signal 1 or press 1 for default index: ');

p2= input('Enter zero position for signal 2 or press 1 for default index: ');

xi= [];

hi= [];

if p1==1

xi=(0:n1-1);

else

xi=(-(p1-1):1:(n1-p1));

end

if p2==1

hi=(0:n2-1);

else

hi=(-(p2-1):1:(n2-p2));

end

xn= zeros(1,n1);

hn= zeros(1,n2);

disp("Enter values of signal 1: ");

for i=1:n1

xn(i) = input('');

end

disp("Index of signal 1: ");

disp(xi);

disp("Values of Signal 1: ");

disp(xn);

disp("Enter values of signal 2: ");

for i=1:n2

hn(i) = input('');

end

disp("Index of signal 2: ");

disp(hi);

disp("Values of Signal 2: ");

disp(hn);

if n1>n2

N=n1

conv\_x = xi

else

N=n2

conv\_x = hi

end

xn\_p = [xn,zeros(1,N-n1)];

hn\_p = [hn,zeros(1,N-n2)];

conv\_y = zeros(1,N);

for n=1:N

for m =1:N

j=mod(n-m,N)

j=j+1

conv\_y(n) = conv\_y(n)+xn\_p(m)\*hn\_p (j);

end

end

disp("Index of convoluted signal: ");

disp(conv\_x);

disp("Values of convoluted signal: ");

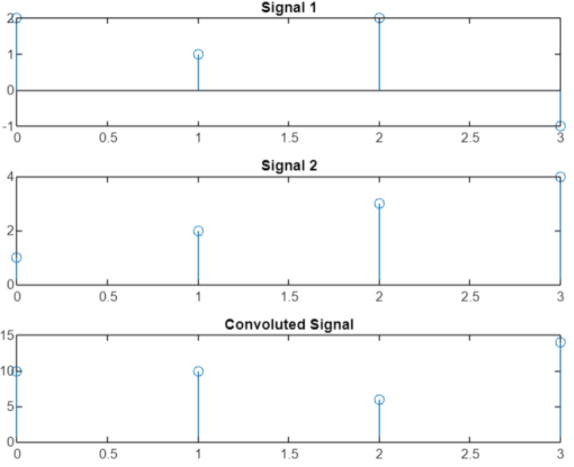
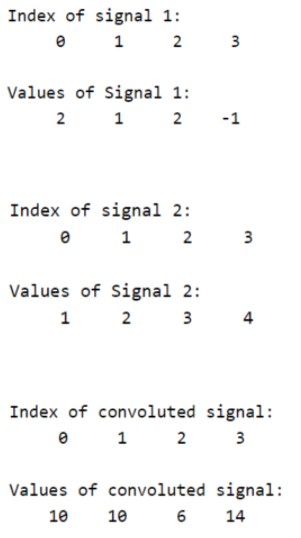
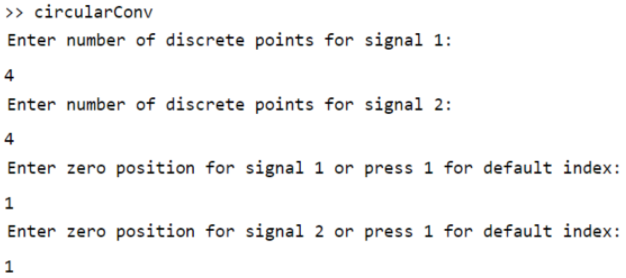
disp(conv\_y);

subplot(3,1,1), stem(xi,xn), title('Signal 1');

subplot(3,1,2), stem(hi,hn), title('Signal 2');

subplot(3,1,3), stem(conv\_x,conv\_y), title('Convoluted Signal');

**Output:**

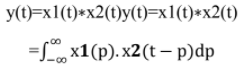


**Conclusion: -**

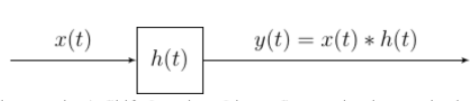
We successfully understood the concept of Linear Convolution, Circular Convolution of 2 discreate time signal and implemented it in MATLAB.

**Post Lab Descriptive Questions**

* 1. Explain the role of convolution in signal processing.
* Convolution of two signals in the time domain is equivalent to multiplication of those two signals in the frequency domain. This is significant because if one can think of some way of modifying signals in the frequency domain then the same modification can be performed in the time domain with the help of convolution.
* Using the strategy of impulse decomposition, systems are described by a signal called the impulse response. Convolution is important because it relates the three signals of interest: the input signal, the output signal, and the impulse response.
* The convolution of two signals in the time domain is equivalent to the multiplication of their representation in frequency domain. Mathematically, we can write the convolution of two signals as



* A linear shift invariant system can be described as convolution of the input signal. The kernel used in the convolution is the impulse response of the system.



* A (continuous time) Shift Invariant Linear System is characterized with its impulse response.
* Convolution can be thought of as performing the job of filtering. That is, convolution is a mathematical operation that combines two filters to produce a third filter that is equivalent to the first two filters cascaded.
* Convolution can be used to identify the magnitude of a single frequency, or magnitude of a band of frequencies in a waveform.
* In the time domain, convolution can be used for reverberation.
  1. Explain the difference between linear and circular convolution?

|  |  |
| --- | --- |
| **Linear Convolution** | **Circular Convolution** |
| Linear convolution is a mathematical operation done to calculate the output of any Linear-Time Invariant (LTI) system given its input and impulse response. | Circular convolution is essentially the same process as linear convolution. Just like linear convolution, it involves the operation of folding a sequence, shifting it, multiplying it with another sequence, and summing the resulting products. |
| We can represent it as y(n)=x(n)\*h(n); Here, y(n) is the output (also known as convolution sum). x(n) is the input signal, and h(n) is the impulse response of the LTI system | We can represent it as y(n)=x(n)^ h(n); Here, y(n) is a periodic output, x(n) is a periodic input, and h(n) is the periodic  impulse response of the LTI system. |
| In linear convolution, both the sequences (input and impulse response) may or may not be of equal sizes. That is, they may or may not have the same number of samples. Thus the output, too, may or may not have the same number of samples as any of the inputs. | In circular convolution, both the sequences (input and impulse response) must be of equal sizes. They must have the same number of samples. Thus the output of a circular convolution has the same number of samples as the two inputs. |
| Using it we can find response of a filter. | Using it we can find response of a filter after zero padding. In fact, we will be doing this in overlap-save and overlap-add methods |
| It may or may not result in a periodic output signal. | Its output is always periodic & its period is specified by the periods of one of its inputs. |

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

Ans - **Linear Convolution via Circular Convolution**:

When two numbers of N-point sequences are circularly convolved it produces another N-point sequence. The linear convolution of two sequences of length N1 and N2 produces an output sequence of length N1+N2-1. To perform linear convolution via circular convolution both the sequences should be converted to N1+N2-1 point sequences by padding zeros. The perform circular convolution of N1+N2-1 sequences. The resultant sequence will be same as that of linear convolution of N1 and N2 point sequences.

Example:

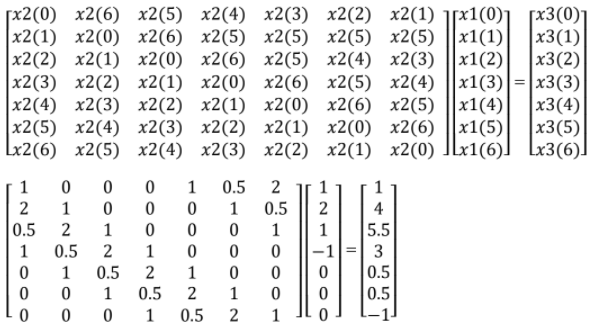
x1(n) = {1,2,0.5,1} x2(n) = {1,2,1,-1}

Here N1=4 and N2=4

The linear convolution output sequence will contain N1+N2-1=4+4-1=7 samples.

Convert both x1 and x2 to sequences with 7 samples. Therefore, x1 = {1,2,0.5,1,0,0,0} and x2 = {1,2,1,-1,0,0,0}

Using Matrix Method for circular convolution:



Therefore, y(n) = {1,4,5.5,3,0.5,0.5,-1}

* **Circular Convolution via Linear Convolution**

Example:

x(n) = {**1**,3,2,1} h(n) = {**1**,1,-1,1}

Circular convolution is given by: y(n) = x(n) \* h(n)

Linear convolution is given by: y1(n) = x(n) \* h(n)

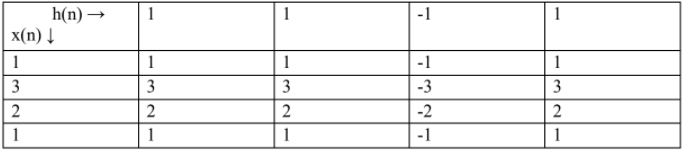
Solving using Matrix Method for linear convolution:

The input sequence and impulse response start at n=0

Therefore, the output response starts at (0+0)) = 0

The input sequence and impulse response contain 4 samples. So, the output response contains 4+4-

1 = 7 samples.



y1(n) = {1,4,4,1,2,1,1}

y1(n) is the output sequence obtained by linear convolution of x(n) and h(n).

The output sequence obtained using circular convolution will have max(n1,n2) samples = max(4,4) = 4 samples.

Dividing output sequence of linear convolution into set of 4 samples and adding corresponding samples to get final output.

{1,4,4,1}, {2,1,1}

y(0) = 1+2 = 3

y(1) = 4+1 = 5

y(2) = 4+1 = 5

y(3) = 1+0 = 1

The output of circular convolution is {3,5,5,1}