MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

SANTOSH, TANGAIL-1902

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

Lab Report

Lab Report No : 02

Lab Report on : Study of linear convolution of discrete-time signals using MATLAB.

Course Title : Digital Signal Processing Lab

Course Code : ICT-3206

Submitted By	Submitted To
Name: Kuldip Saha Mugdha ID: IT-22018 3rd Year, 2nd Semester Session: 2021-2022 Dept. of ICT, MBSTU	Dr. Mst. Nargis Akter Professor DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

Date of Performance: 28/10/25 Date of Submission: 16/11/25

Introduction:

Convolution is a fundamental operation in Digital Signal Processing used to determine the output of a linear time-invariant (LTI) system for a given input. Linear convolution can be performed directly in time domain, via circular convolution, and using the Discrete Fourier Transform (DFT). This experiment demonstrates different methods of performing linear convolution using MATLAB.

Objective:

- To perform linear convolution of two discrete-time sequences.
- To obtain linear convolution using circular convolution.
- To perform linear convolution using DFT and IDFT.

Theory:

The **linear convolution** of two discrete-time signals $x(n)$and $h(n)$is defined as

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

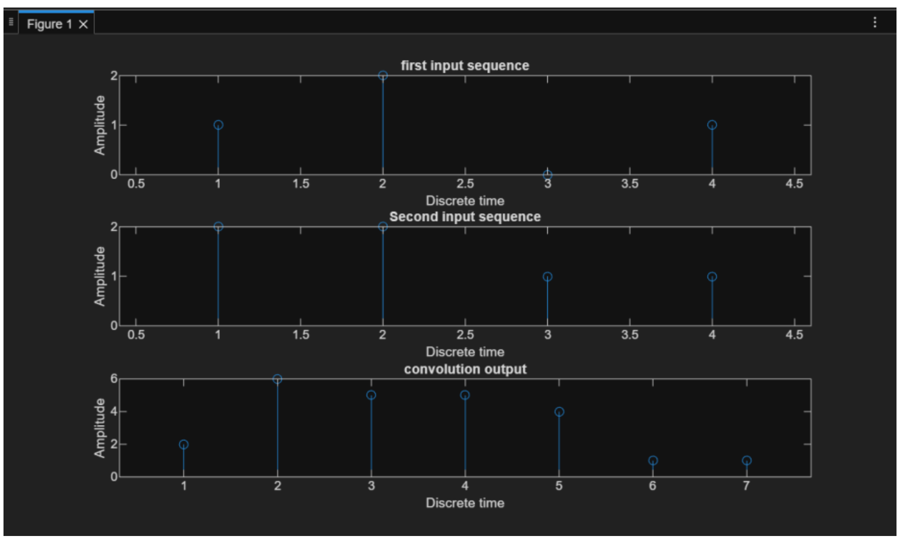
- **Direct convolution** is performed using the conv() function.
- **Circular convolution** can be used to obtain linear convolution by zero-padding the sequences properly.
- **DFT-based convolution** uses FFT and IFFT, based on the convolution theorem.

MATLAB Code:

Program 1: Linear Convolution Using conv()

```
% Convolution of two sequences

clc; clear all; close all;

x1 = [1 2 0 1];
x2 = [2 2 1 1];

y = conv(x1, x2);

disp('the convolution output is')
disp(y)

subplot(3,1,1), stem(x1);
xlabel('Discrete time')
ylabel('Amplitude')
title('first input sequence')

subplot(3,1,2), stem(x2);
xlabel('Discrete time')
ylabel('Amplitude')
title('Second input sequence')

subplot(3,1,3), stem(y);
xlabel('Discrete time')
ylabel('Amplitude')
title('convolution output')
```

Output:



Program 2: Linear Convolution via Circular Convolution

```
% Linear convolution via circular convolution

clc; clear all; close all;

x1 = [1 2 3 4 5];
x2 = [2 2 0 1 1];

x1e = [x1 zeros(1, length(x2)-1)];
x2e = [x2 zeros(1, length(x1)-1)];

ylin = cconv(x1e, x2e, length(x1e));

disp('linear convolution via circular convolution')
disp('ylin')

y = conv(x1, x2);

disp('Direct convolution')
disp(y)
```

Output:

```
linear convolution via circular convolution
ylin
Direct convolution
     2     6    10    15    21    15     7     9     5
```

Program 3: Linear Convolution Using DFT

```
% Linear convolution using DFT

clc; clear all; close all;

x = [1 2];
h = [2 1];

x1 = [x zeros(1, length(h)-1)];
h1 = [h zeros(1, length(x)-1)];

X = fft(x1);
H = fft(h1);

y = X .* H;
y1 = ifft(y);

disp('the linear convolution of the given sequence')
disp(y1)
```

Output:

```
the linear convolution of the given sequence
     2     5     2
```

Result:

- Linear convolution was successfully obtained using direct, circular, and DFT-based methods.
- The outputs from circular convolution and DFT-based convolution matched the direct convolution result after proper zero-padding.

Conclusion:

This experiment verified that linear convolution of discrete-time signals can be performed using different methods such as direct convolution, circular convolution, and DFT-based convolution. All methods produced identical results when implemented correctly.