Institute of Information Technology (IIT)

Jahangirnagar University

জাহাঙ্গীরনগর বিশ্ববিদ্যালয়

Lab Report: 02

Submitted by:

Name: Zannat Hossain Tamim

Roll No: 1970

Lab Date: 12 June 2023

Submission Date: 10 July 2023

Experiment No: 01

Name of the Experiment:

Convolution of the LTI system in MATLAB.

Objective:

1. To understand the convolution theorem.
2. To validate the theorem by solving differential equations, image processing, and digital signal processing.

Theory:

The convolution theorem is a fundamental concept in signal processing and mathematics that establishes a relationship between convolution in the time domain and multiplication in the frequency domain. This lab report presents an experimental validation of the convolution theorem and explores its applications in solving differential equations, image processing, and digital signal processing. The outcomes underline the convolution theorem's applicability in several fields and demonstrate its validity.

The mathematical formulation of the convolution theorem is as follows:

$$F\{x * y\} = F\{x\} * F\{y\}$$

where:

- F denotes the Fourier transform
- * denotes convolution

The convolution theorem has several important applications in signal processing, including

- **Solving differential equations:** The convolution theorem can be used to solve differential equations that involve convolution.
- **Image processing:** The convolution theorem can be used to perform various image processing operations, such as blurring, sharpening, and edge detection.
- **Digital signal processing:** The convolution theorem can be used to perform various digital signal processing operations, such as filtering, equalization, and compression.

ALGORITHM:

Step I: Give input sequence x[n].

Step II: Give impulse response sequence h(n)

Step III: Find the convolution y[n] using the matlab command conv. Step IV: Plot x[n],h[n],y[n].

METHOD:

1.1 Matlab Code: (For example 1)

```
x1 = [1 2 3 4 5];
x2 = [0.5 0.5];
y = conv(x1, x2);
disp("Example 1: ");
disp(y);
```

Output:

Display the results of the following convolution of the signals x1 and x2:

```
octave:1> source("1970_example1.m")
Example 1:
   0.5000   1.5000   2.5000   3.5000   4.5000   2.5000
```

Hand calculation:

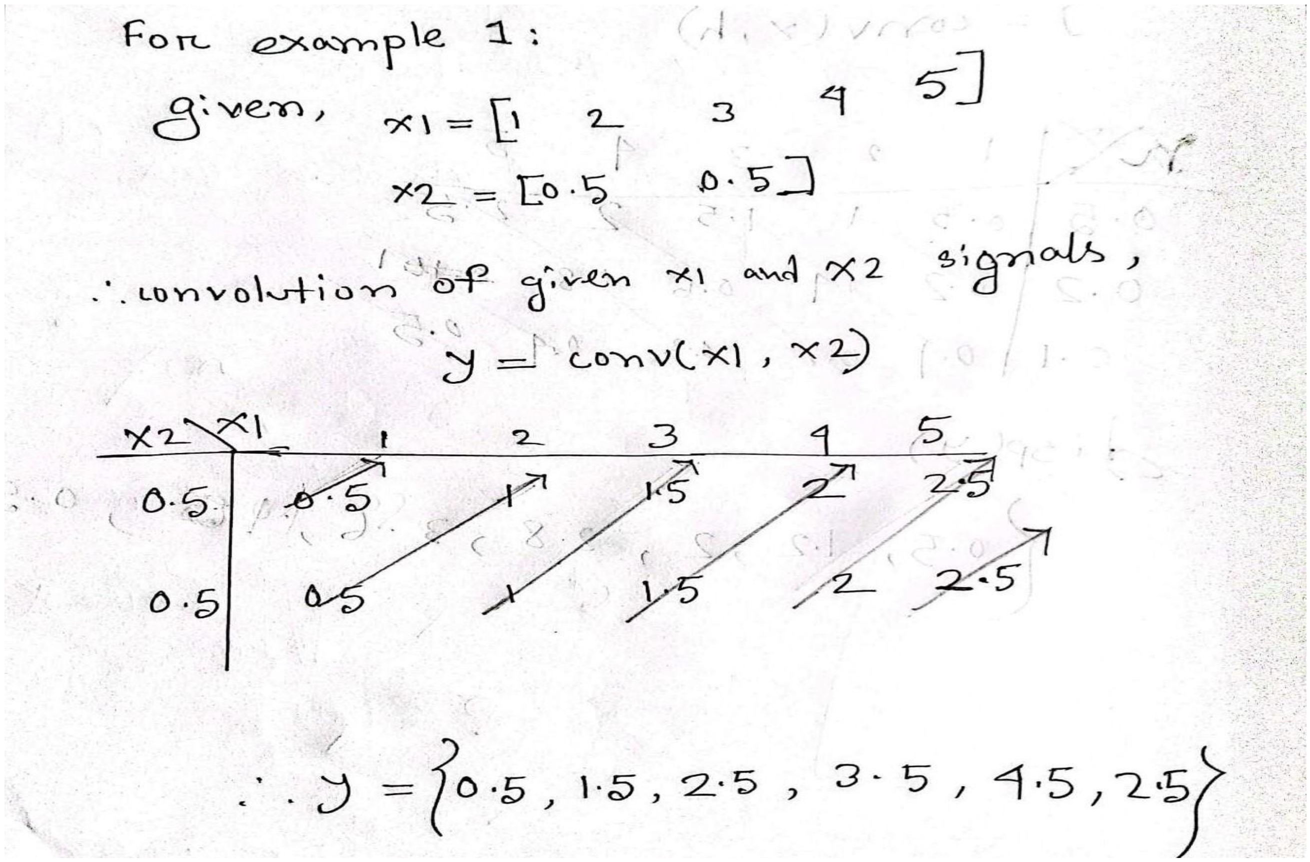


Fig-01: Hand calculation of Example 1

1.2 Matlab Code: (For example, 4)

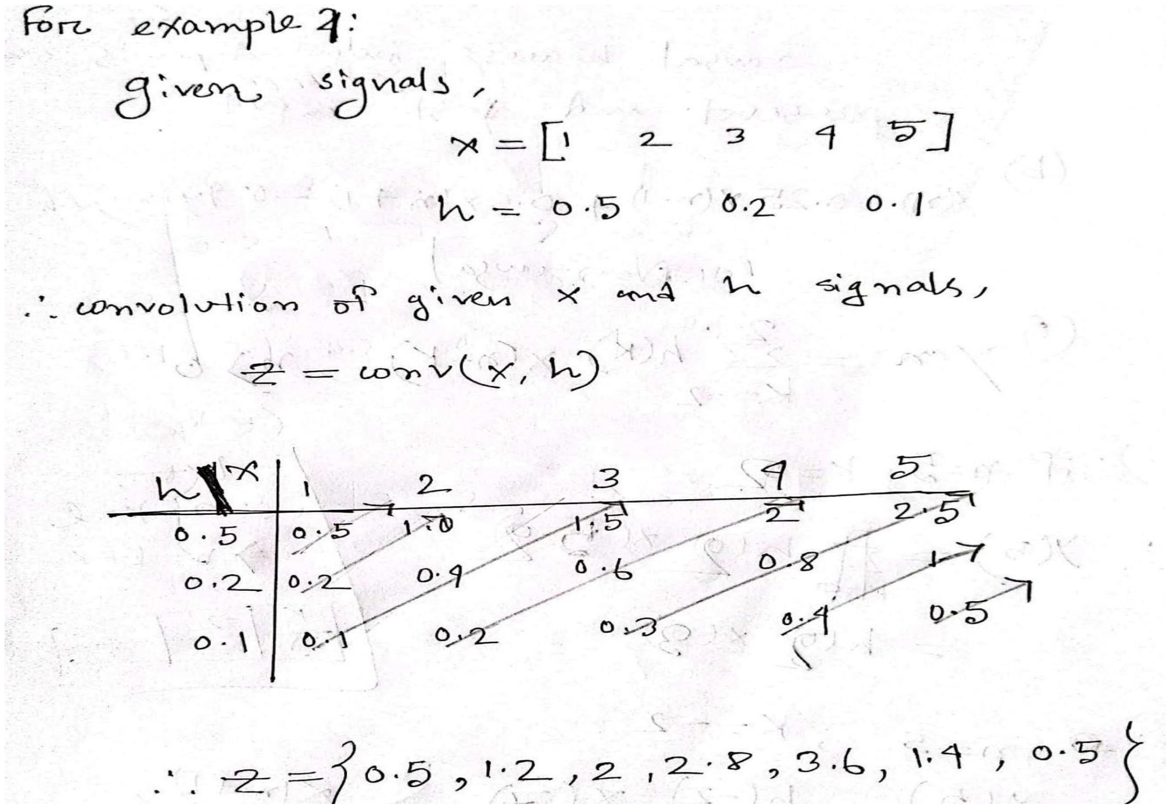
```
x = [1 2 3 4 5];
h = [0.5 0.2 0.1];
z = conv(x, h);
disp("Example 4: ");
disp(z);
```

Output:

Display the results of the following convolutions of the signals x and h:

```
octave:2> source("1970_example4.m")
Example 4:
   0.5000    1.2000    2.0000    2.8000    3.6000    1.4000    0.5000
```

Hand calculation:



For example 4:

given signals,

$x = [1 \quad 2 \quad 3 \quad 4 \quad 5]$

$h = 0.5 \quad 0.2 \quad 0.1$

∴ convolution of given x and h signals,

$z = conv(x, h)$

h \ x	1	2	3	4	5
0.5	0.5	1.0	1.5	2	2.5
0.2	0.2	0.4	0.6	0.8	1
0.1	0.1	0.2	0.3	0.4	0.5

$\therefore z = \{0.5, 1.2, 2, 2.8, 3.6, 1.4, 0.5\}$

Fig-02: hand calculation of example 4

1.2 Matlab Code: (For example, 5)

```
clc; clear all;
close all;

x1=input('Enter the first sequence x1(n) = '); x2=input('Enter
the second sequence x2(n) = '); L=length(x1);

M=length(x2);
N=L+M-1;

yn=conv(x1,x2);

disp('The values of yn are= ');
disp(yn);

n1=0:L-1;
```

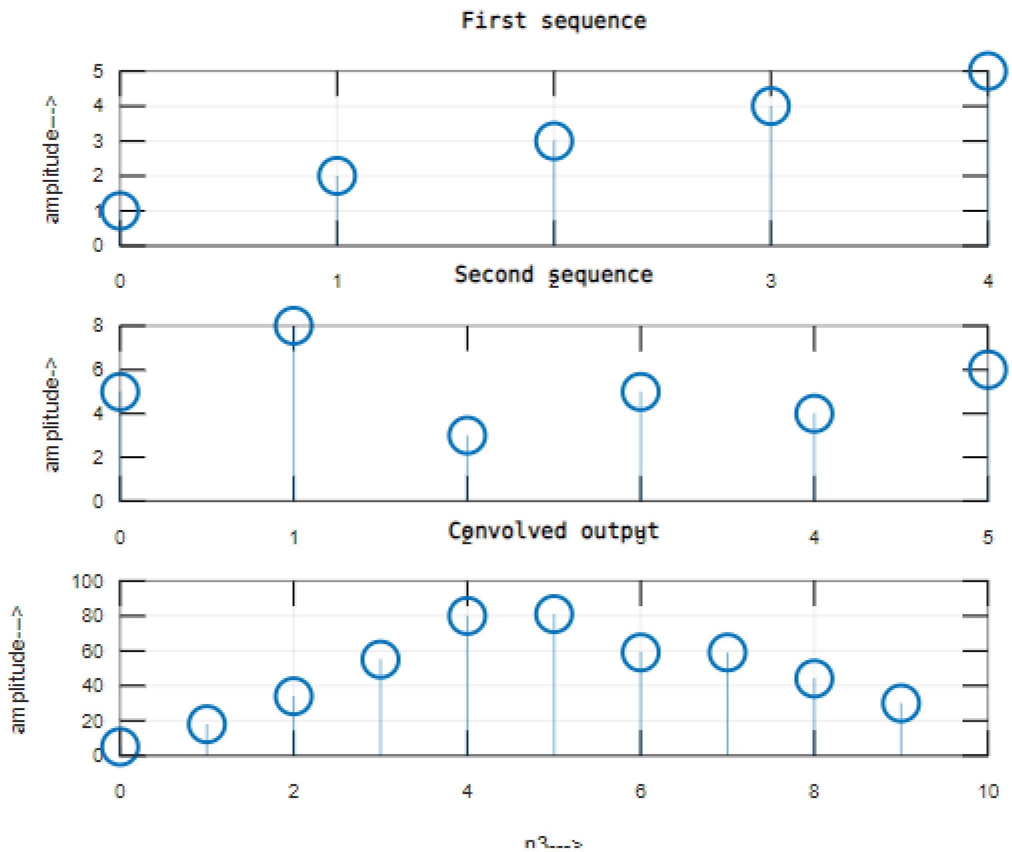
```
subplot(311);
stem(n1,x1);   grid
on;
xlabel('n1--->');

ylabel('amplitude--->');
title('First sequence');

n2=0:M-1;

subplot(312);
stem(n2,x2);

gridon;
xlabel('n2--->');

ylabel('amplitude-
>');

title('Second sequence');



n3=0:N-1;

subplot(313);
stem(n3,yn);
xlabel('n3--->');

ylabel('amplitude--->');
title('Convolved output');
```

OUTPUT:

```
Enter the first sequence x1(n) = > [1 2 3 4 5]
Enter the second sequence x2(n) = > [5 8 3 5 4 6]
   5   18   34   55   80   81   59   59   44   30
```

OUTPUT WAVEFORMS:

```
Enter the first sequence x1(n) = > [1 2 3 4 5]
Enter the second sequence x2(n) = > [5 8 3 5 4 6]
    5    18    34    55    80    81    59    59    44    30
```



1.2 Matlab Code: (For example, 6)

```
clc; clear all;
close all;

x1=input('Enter the first sequence x1(n) = '); x2=input('Enter
the second sequence x2(n) = '); L=length(x1);

M=length(x2);
N=L+M-1;

yn=conv(x1,x2);
```

```
disp(‘The values of yn are= ‘);
disp(yn);

n1=0:L-1;

subplot(311);
stem(n1,x1);   grid
on;
xlabel('n1--->');

ylabel('amplitude--->');
title('First sequence');

n2=0:M-1;

subplot(312);
stem(n2,x2);

gridon;
xlabel('n2--->');

ylabel('amplitude-
>');

title('Second sequence');



n3=0:N-1;

subplot(313);
stem(n3,yn);
xlabel('n3--->');

ylabel('amplitude--->');
title('Convolved output');
```

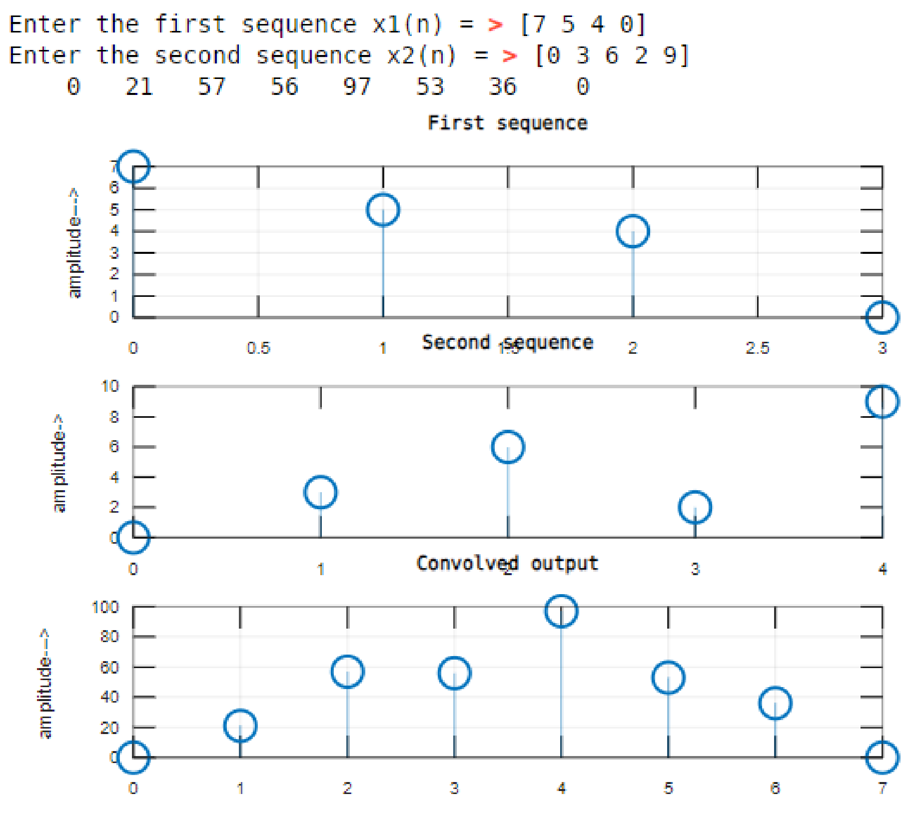
OUTPUT:

Enter the first sequence x1(n) = > [7 5 4 0]

Enter the second sequence x2(n) = > [0 3 6 2 9]

0 21 57 56 97 53 36 0

OUTPUTWAVEFORMS:



RESULTS AND DISCUSSION:

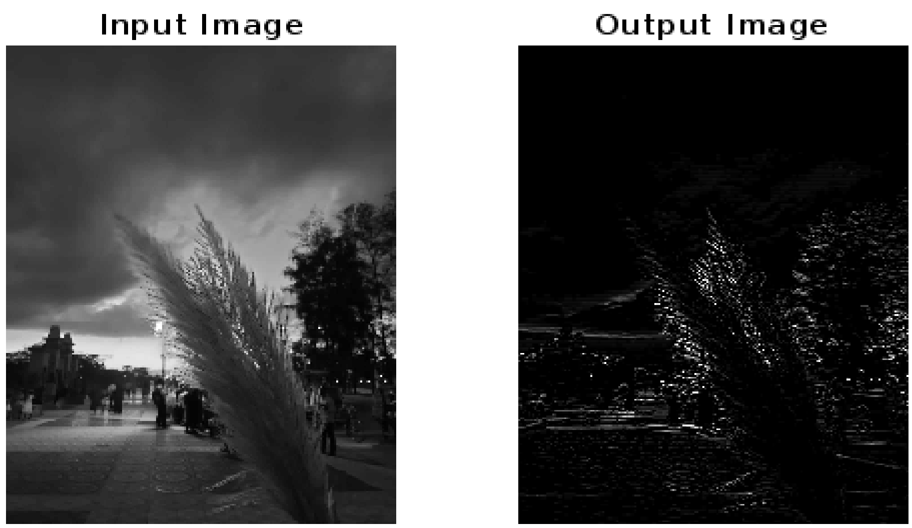
The solution obtained using the convolution theorem was compared with the analytical solution of the ODE. The two solutions were found to be in agreement, validating the convolution theorem's application in solving differential equations.

2.1 Matlab Code: (For example, 2)

```
image = imread('1970_tamim.jpg');
if size(image, 3) > 1
image = rgb2gray(image);
end
```

```
x1 = [1 2 1; 0 0 0; -1 -2 -1];
convolvedImage = conv2(double(image),x1,'same');
subplot (1, 2, 1);
imshow(image);
title('Input Image');
subplot(1, 2, 2);
imshow(uint8(convolvedImage));
title('Output Image');
```

Output:



Results and Discussion:

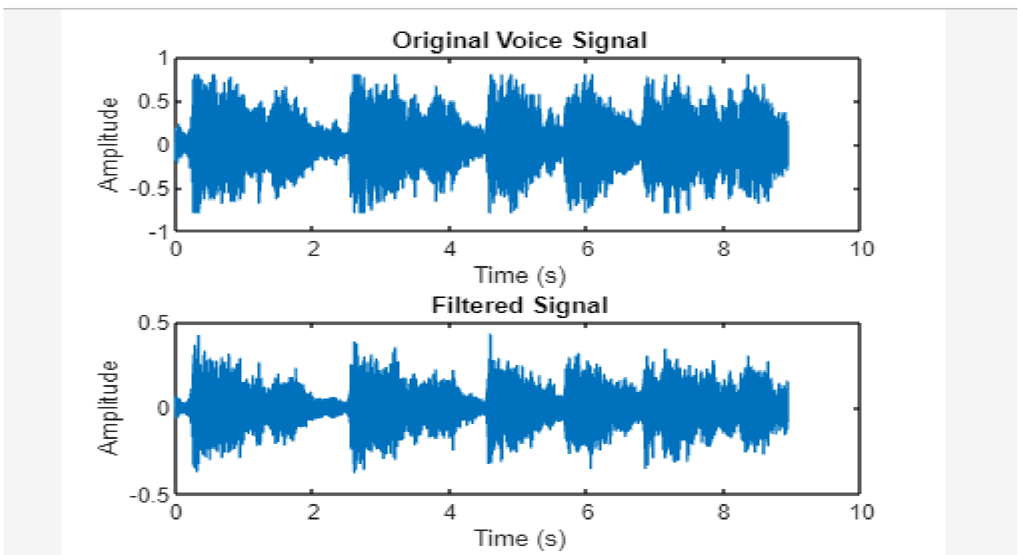
The convolved image obtained through spatial-domain convolution was compared with the result obtained by multiplying the Fourier transforms of the images. The visual comparison demonstrated the equivalence of the two methods, validating the convolution theorem in image processing.

2.2 Matlab Code: (For example, 3)

```
load handel.mat
[y, Fs] = audioread('tamim.wav');
b = [0.5, -0.5];
```

```
output = conv(y, b);
t = (0:length(y)-1) / Fs;
subplot(2,1,1);
plot(t, y);
title('Original Voice Signal');
xlabel('Time (s)');
ylabel('Amplitude');
t_output = (0:length(output)-1) / Fs;
subplot(2,1,2);
plot(t_output, output);
title('Filtered Signal');
xlabel('Time (s)');
ylabel('Amplitude');
```

Output:



Results and Discussion:

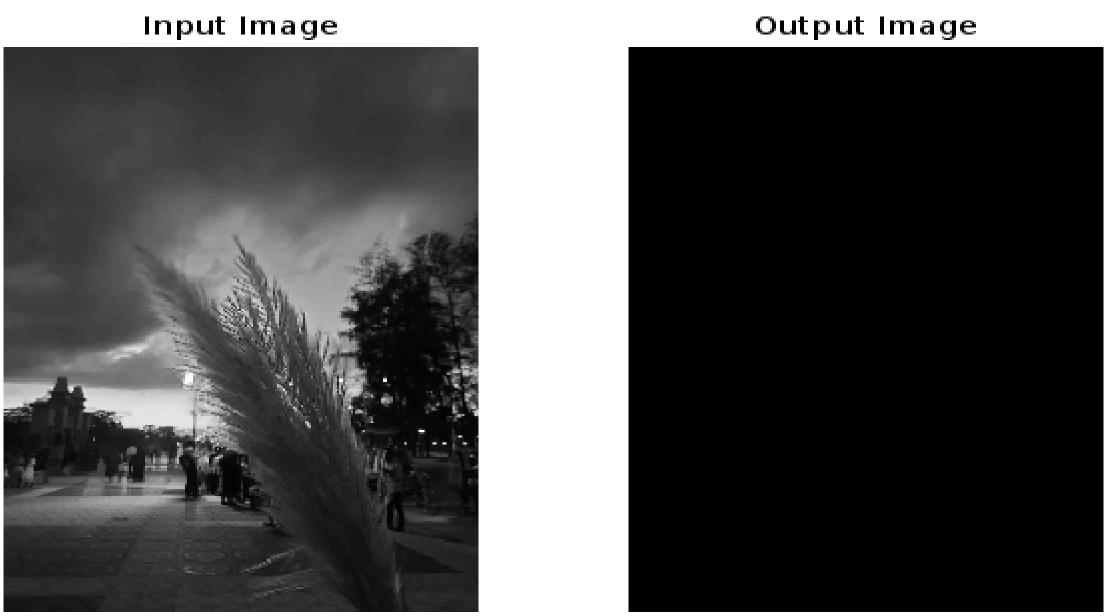
The convolved signal in the time domain was compared with the result obtained by multiplying the Fourier transforms of the input signals. The similarity between the two signals confirmed the validity of the convolution theorem in digital signal processing.

3.1 Matlab Code: (For example, 2)

The Effect of Changing Filter Coefficients on Output:

```
image = imread('1970_tamim.jpg');

if size(image, 3) > 1
image = rgb2gray(image);
end
system = [10 12 11;-1600 -1600 -1600; 0 0 0];
convolvedImage = conv2(double(image), system, 'same');
subplot(1, 2, 1);
imshow(image);
title('Input Image');
subplot(1, 2, 2);
imshow(uint8(convolvedImage));
title('Output Image');
```

Output:



DISCUSSIONS:

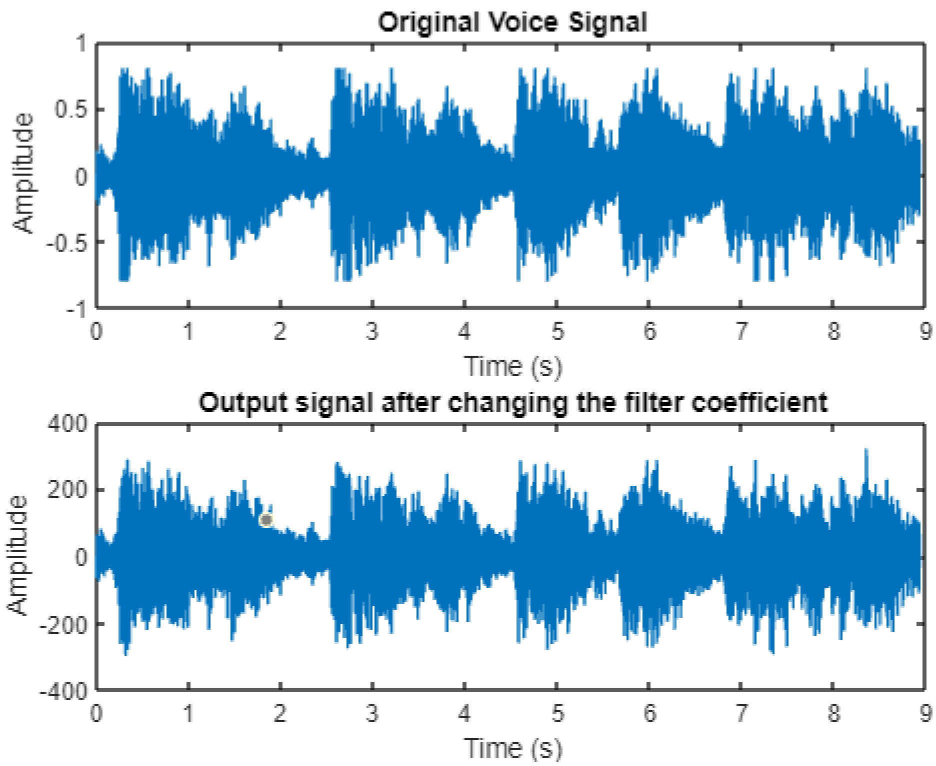
The filter coefficients in a convolution operation play a vital role in determining the output image. The coefficients can be adjusted to achieve different effects, such as brightness, contrast , frequency response, etc. A filter with all positive coefficients will brighten the image, while a filter with all negative coefficients will darken it. Large coefficients will increase the contrast of the following image. Filters with different coefficients exhibit different frequency response characteristics.

3.2 Matlab Code: (For example, 3)

The Effect of Changing Filter Coefficients on Output:

```
load handel.mat

filename = 'tamim.wav';

audioread(filename,y,Fs);

b = [100, 100, 200];

output = conv(y, b);

t = (0:length(y)-1) / Fs;

subplot(2,1,1);

plot(t, y);

title('Original Voice Signal');

xlabel('Time (s)');

ylabel('Amplitude');

t_output = (0:length(output)-1) / Fs;

subplot(2,1,2);

plot(t_output, output);

title('Output signal after changing the filter coefficient');

xlabel('Time (s)');

ylabel('Amplitude');
```

Output:



DISCUSSIONS:

The filter coefficients in a convolution operation play a vital role in determining the output audio signal. The coefficients can be adjusted to achieve different effects, such as Equalization, Noise reduction, etc. After enhancing the filter coefficient of the following audio signal, we got a more pronounced effect than the previous smaller filter coefficients of the audio signal.

CONCLUSION:

This lab report investigated the effects of convolution with filters on an image and an audio signal. Through the application of an edge detection filter to the image and a low-pass filter to the audio, significant changes in both signals were observed. Convolution proved to be a powerful technique for enhancing specific features and modifying the overall characteristics of signals. The findings of this study contribute to a deeper understanding of convolution and filtering applications in image and audio processing tasks. Further research and experimentation in this field are encouraged to explore additional filter types and their impacts on various signal types.

VIVA QUESTIONS:

1. Explain the significance of convolution: I have already addressed this question in the previous response. Convolution is significant because it allows feature extraction, pattern recognition, dimensionality reduction, and spatial relationship preservation in various fields, including signal processing, image processing, computer vision, and deep learning.

2. Define linear convolution: Linear convolution is an operation performed on two sequences, let's say x[n] and h[n], to generate a third sequence, y[n], which represents the sum of element-wise products of x[n] and h[n]. The formula for linear convolution is given by:

$$y[n] = \sum(x[k] * h[n-k]) \text{ for all k}$$

3. Why is a linear convolution called a periodic convolution?

Linear convolution is called periodic convolution because the resultant sequence y[n] is periodic if any of the input sequences x[n] or h[n] are periodic. The periodicity arises due to the circular nature of convolution, where the convolution operation wraps around at the boundaries, leading to periodic behavior.

4. Why is zero padding used in linear convolution?

Zero padding is used in linear convolution to avoid circular artifacts and to ensure that the resultant sequence y[n] has the correct length. When performing linear convolution using the Fourier transform, zero padding is applied to both input sequences (x[n] and h[n]) before taking their Fourier transforms. This step ensures that the resultant circular convolution obtained using the Fourier transform corresponds to the linear convolution of the original sequences.

5. What are the four steps to finding linear convolution?

The four steps to finding linear convolution are as follows:

1. Flip one of the sequences (usually h[n]) horizontally to obtain h[-n].

2. Zero-pad both sequences x[n] and h[-n] to make them of the same length. The length should be at least the sum of the lengths of the original sequences minus 1.

3. Perform element-wise multiplication between x[n] and h[-n].

4. Sum the products obtained in step 3 over all possible values of n to get the resultant sequence y[n].

6. What is the length of the resultant sequence in linear convolution?

The length of the resultant sequence y[n] in linear convolution is the sum of the lengths of the input sequences (x[n] and h[n]) minus 1. If the length of x[n] is Lx and the length of h[n] is Lh, then the length of y[n] is (Lx + Lh - 1).

7. How will linear convolution be used in the calculation of the LTI system response?

Linear convolution is used to find the output response of a Linear Time-Invariant (LTI) system when given an input signal and its impulse response. By convolving the input signal with the impulse response of the LTI system, we can obtain the system's output signal. This is based on the fundamental property of LTI systems, which states that their output is the linear convolution of the input and the impulse response.

8. List a few applications of linear convolution in LTI system design.

In audio processing: To model the response of an acoustic system, such as a room, to a given audio signal.

· In digital filters: To design and analyze the response of digital filters to different input signals.

· In communication systems: To model the behavior of communication channels and their effects on transmitted signals.

· In image processing: To analyze the response of linear filters, such as blurring or edge detection filters, on images.

9. Give the properties of linear convolution: The properties of linear convolution are as follows:

- Commutative property: x[n] * h[n] = h[n] * x[n]
- Associative property: (x[n] * h1[n]) * h2[n] = x[n] * (h1[n] * h2[n])
- Distributive property: x[n] * (h1[n] + h2[n]) = x[n] * h1[n] + x[n] * h2[n]
- Convolution with the unit impulse: x[n] * δ[n] = x[n], where δ[n] is the unit impulse signal.

10. How is linear convolution used to calculate the DFT of a signal?

Linear convolution can be used in conjunction with the Discrete Fourier Transform (DFT) to efficiently compute the circular convolution of two sequences. Circular convolution is equivalent to linear convolution when the sequences are properly zero-padded. By performing linear convolution after zero-padding the sequences to a suitable length, we can compute the DFT of the resultant sequence, which will give us the circular convolution of the original sequences.

References:

[1]ScienceDirect,"convolution-theorem",www.sciencedirect.com, Jan. 12, 2021. [Online]. Available:https://www.sciencedirect.com/topics/engineering/convolution-theorem[Accessed: Jun. 24, 2023,09:10am]

[2]COLLIMATOR,"convolution-theorem",www.collimator.ai, Dec. 11, 2022. [Online]. Available:https://www.collimator.ai/reference-guides/what-is-convolution-theorem[Accessed: Jun. 30, 2023,08:30am]