

FOSIP EXPT-1

Name: Manish Shashikant Jadhav

UID: 2023301005

Branch :Comps B

Batch: B

Aim: The aim of this experiment is to study mathematical operations such as:

1. Linear Convolution,
 2. Circular Convolution, and
 3. Linear Convolution using Circular Convolution
-

1. Linear Convolution

Problem Definition: Find Linear Convolution of L point sequence $x[n]$ and M point sequence $h[n]$.

Experimentation

Case-1 : To find $y[n] = x[n] * h[n]$

Input: $x[n] = \{8,4,9,6,1\}$,

Length $L=5$

$h[n] = \{3,6,2,1\}$,

Length $M=4$

Output

```
Enter the length of x[n] ie L = 5
Enter the values of x[n]
8
4
9
6
1

Enter the length of h[n] ie M =
4
Enter the values of h[n]
3
6
2
1

x[n] =  8.00   4.00   9.00   6.00   1.00   0.00   0.00   0.00
h[n] =  3.00   6.00   2.00   1.00   0.00   0.00   0.00   0.00
y[n] = 24.00  60.00  67.00  88.00  61.00  27.00   8.00   1.00

...Program finished with exit code 0
Press ENTER to exit console.
```

$y[n] = \{24, 60, 67, 88, 61, 27, 8, 1\}$, Length $L = 8$

Length of Linear Convolution output signal is $N = 5 + 4 - 1 = 8$ That means, Length of Linear Convolution output signal is $N = L + M - 1$

Case 2:

$x[n] = \{1, 1, 0, 1, 1\}$

$h[n] = \{1, -2, -3, 4\}$

$y[n] = \{1, -1, -5, 2, 3, -5, 1, 4\}$

Output

```
Enter the length of x[n] ie L = 5
Enter the values of x[n]
1
1
0
1
1

Enter the length of h[n] ie M = 4
Enter the values of h[n]
1
-2
-3
4

x[n] =  1.00   1.00   0.00   1.00   1.00   0.00   0.00   0.00
h[n] =  1.00  -2.00  -3.00   4.00   0.00   0.00   0.00   0.00
y[n] =  1.00  -1.00  -5.00   2.00   3.00  -5.00   1.00   4.00
```

Analysis

Length of $x[n]$ (L) = 5

Length of $h[n]$ (M) = 4

$$\begin{aligned}\text{Length of } y[n] &= L + M - 1 \\ &= 5 + 4 - 1 \\ &= 9 - 1 \\ &= 8\end{aligned}$$

We conclude the following:

- Length of Linear Convolution output signal ($y[n]$) = Length of first input signal ($x[n]$) + Length of second input signal ($h[n]$) – 1
 - Adding zeros at the end of the input signal does NOT change the output for Linear Convolution, i.e. Linear Convolution always gives a unique answer.
 - In Linear convolution, if both the input signals are causal, then the resultant output signal is also causal.
-

2. Circular Convolution

Problem Definition: Find Circular Convolution of L point sequence $x[n]$ and M point sequence $h[n]$.

Experimentation

Case - 1: To find $y[n] = x[n] * h[n]$

Input: $x[n] = \{1, 3, 6, 4, 9, 8\}$,

Length $L = 6$

$h[n] = \{9, 7, 2\}$,

Length $M = 3$

Output:

$y[n] = \{88, 52, 77, 84, 112, 145\}$, Length $L = 6$

```
Enter the length of x[n] L = : 6
Enter the values of x[n] : 1
3
6
4
8
9
Enter the length of h[n] M = : 3
Enter the values of h[n] : 9
7
2

x[n] =   1.00   3.00   6.00   4.00   8.00   9.00
h[n] =   9.00   7.00   2.00   0.00   0.00   0.00
y[n] =  88.00  52.00  77.00  84.00  112.00  145.00

...Program finished with exit code 0
Press ENTER to exit console.
```

3. Linear Convolution using Circular Convolution

Problem Definition: Find Linear Convolution using Circular Convolution of L point sequence $x[n]$ and M point sequence $h[n]$.

Experimentation

Case 1: Length of input signals is Equal to the length of output signal of Simple Linear Convolution

$$x[n] = \{1, 1, 0, 1, 1\}$$

$$h[n] = \{1, -2, -3, 4\}$$

Theory

Original Length of $x[n]$ (L) = 5

Original Length of $h[n]$ (M) = 4

In Linear Convolution length of output signal $y[n] = L + M - 1 = 5 + 4 - 1 = 8$

So if we want to use Linear Convolution using Circular Convolution, we will have to do zero padding for the input signals. We will have to add zeros to the end of the signal such that the length of both the input signals will be equal to the length of the output signal in Linear Convolution.

Since length of $x[n] = 5$, to make its length = 8, we will pad signal $x[n]$ with $8-5 = 3$ zeros. Similarly length of $h[n] = 4$, to make its length = 8, we will pad signal $h[n]$ with $8-4 = 4$ zeros.

Therefore final $x[n] = [1, 1, 0, 1, 1, 0, 0, 0]$

Therefore final $h[n] = [1, -2, -3, -4, 0, 0, 0, 0]$

And,

$$L = M = 8$$

Output

```
Enter the length of x[n] L = : 8
Enter the values of x[n] : 1
1
0
1
1
0
0
0
0
Enter the length of h[n] M = : 8
Enter the values of h[n] : 1
-2
-3
4
0
0
0
0
0
x[n] =  1.00   1.00   0.00   1.00   1.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
h[n] =  1.00  -2.00  -3.00   4.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
y[n] =  1.00  -1.00  -5.00   2.00   3.00  -5.00   1.00   4.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
```

$y[n] = \{1, -1, -5, 2, 3, -5, 1, 4\}$

Case 2: Length of input signals is GREATER than the length of output signal of Simple Linear Convolution

For Case 2, we will add two extra zeros to the already padded input signals to get length greater than the length of the output signal of Linear Convolution.

Output

```
Enter the length of x[n] L = : 10
Enter the values of x[n] : 1
1
0
1
1
0
0
0
0
0
0
Enter the length of h[n] M = : 10
Enter the values of h[n] : 1
-2
-3
4
0
0
0
0
0
0
0
x[n] =  1.00   1.00   0.00   1.00   1.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
h[n] =  1.00  -2.00  -3.00   4.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
y[n] =  1.00  -1.00  -5.00   2.00   3.00  -5.00   1.00   4.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00

...Program finished with exit code 0
Press ENTER to exit console.
```

Analysis

Final Length of $x[n]$ (L) = 8

Final Length of $h[n]$ (M) = 8

Final Length of $y[n] = 8$

In Circular Convolution length of output signal $y[n] = L = M = 8$

We can see that this $y[n]$ is the same as the one that we got by doing Linear Convolution in the first case.

We will get the same $y[n]$ if we add extra zeros to make the length greater than $L + M - 1$ of original signals = 8. So if we make length of $x[n] = 10$ and Length of $h[n] = 10$, we will get the same output signal $y[n]$ with 2 extra zeroes at the end.

We conclude the following:

- We will get correct output only when length of input signals with padding is greater than or equal to the length of output signal of simple Linear Convolution
- Length of output signal ($y[n]$) \geq Original Length of first input signal ($x[n]$) + Original Length of second input signal ($h[n]$) - 1

Conclusion :

1. Length of Linear Convolution output signal is $N = L + M - 1$
 - Where L is the length of first input signal.
 - M is the length of second input signal.
 - N is the length of linear convolution output signal.
 2. In Linear convolution if both the input signals are causal, then resultant output signal is also causal.
 3. To find Circular Convolution, Select $N = \text{MAX}(L, M)$
 - Where L is the length of first input signal
 - M is the length of second input signal
 4. To find Linear Convolution using Circular Convolution,
Select $N \geq L + M - 1$
 - Where L is the length of first input signal and M is the length of second input signal.
 5. Circular Convolution gives aliased output.
-

4. Application : Audio Signal Filtering

Problem Statement :

Filter the Audio Signal Captured in the presence of noise and improve the quality of sound

Algorithm :

1. Record Audio Signal in the presence of noise $\Rightarrow x[n]$.
2. Play the recorded signal $x[n]$ and observe the quality of sound.
3. Design FIR Low Pass Filter using MATLAB filter design Tool. Take $F_{pass} = 4000\text{Hz}$. $F_{stop} = 6000\text{Hz}$ $F_s = 44000$
4. Filter the audio signal $x[n]$ i.e. Perform Linear Convolution of $x[n]$ and $h[n]$ $\Rightarrow y[n]$
5. Play the filtered signal $y[n]$ and observe the quality of sound

• Matlab Code:

```
function Hd = DSP_Application1
    Fs = 44000; % Sampling Frequency
    ip_file = 'TextAudio.wav';
    [ip_audio,Fs] = audioread(ip_file);

    % Filter design parameters
    N = 100; % Order
    Fpass = 100; % Passband Frequency
    Fstop = 600; % Stopband Frequency
    Wpass = 1; % Passband Weight
    Wstop = 1; % Stopband Weight
    dens = 20; % Density Factor

    b = firpm(N, [0 Fpass Fstop Fs/2]/(Fs/2), [1 1 0 0], [Wpass Wstop], ...
        {dens});
    Hd = dfilt.dffir(b);

    op_audio = conv(ip_audio, b, 'same');

    op_file = 'filtered_op.wav';
    audiowrite(op_file, op_audio, Fs);

    sound(ip_audio, Fs);
    pause(length(ip_audio)/Fs);
    sound(op_audio, Fs);
    subplot(2,1,1);
    time_ip = (0:length(ip_audio)-1)/Fs;
```



```
time_op = (0:length(op_audio)-1)/Fs;  
plot(time_ip, ip_audio);  
title('Original Audio');  
subplot(2,1,2);  
plot(time_op, op_audio);  
title('Filtered Audio');  
end
```

Output:

