



## DIGITAL SIGNAL PROCESSING LAB

Debagnik Kar

Roll: 1804373, ETC -06, EC - 3096

## Index Page

Experiment No.	Aim of the Experiment	Page No.	Date of Experiment	Date of Submission
01	To write a function for finding circular convolution and correlation. Prove that autocorrelation is an even function and $R_{xy}(n) = R_{yx}(-n)$	2-6	23/12/2020	25/12/2020
02	To solve a given differential equation to find its impulse and step responses. Plot the response and verify the solution	7-9	06/01/2021	13/01/2021
03	To find the DTFT of an arbitrary sequence and prove the convolution property	10-14	13/01/2021	20/01/2021
04	To write a function to find DFT and IDFT and use it to find circular and linear convolution of given sequences.	15-20	20/01/2021	27/01/2021
05	To perform sectional convolution using overlap add and overlap save method.	21-26	27/01/2021	24/02/2021
06	To design FIR LPF, HPF and BPF using windowing techniques.	27-36	17/02/2021	24/02/2021
07	To Study various parameters required for designing an IIR filter and design LPF, HPF and BPF Butterworth filters for specific cut-off frequencies.	37-	24/02/2021	03/03/2021
08	To study up sampling and down sampling of the sequences and study the frequency response			
09	Open Ended – I			
10	Open Ended - II			

<b>Experiment Number</b>	05
<b>Date of Experiment</b>	27/01/2021
<b>Date of Submission</b>	24/02/2021
<b>Name of the Student</b>	Debagnik Kar
<b>Roll Number</b>	1804373
<b>Section</b>	ETC - 06

**Aim of The Experiment :-**

To perform sectional convolution using overlap add and overlap save method.

**Software Required :-**

- MATLAB R2018a

**Theory :-**

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).

Two methods that make linear convolution look like circular convolution are overlap-save and overlap-add.

**Overlap-Add Method**

The overlap-add method is used to break long signals into smaller segments for easier processing. FFT convolution uses the overlap-add method together with the Fast Fourier Transform, allowing signals to be convolved by multiplying their frequency spectra.

- The overlap-add method allows us to calculate the convolution of very long sequences.
- The overlap-add method breaks a long sequence,  $x(n)$ , into signals of shorter length and calculates the convolution of each block independently.

**Overlap-Save Method**

- If the input frame size is  $N$  and the filter length is  $M$ , then a length  $N$  FFT and IFFT are used.
- As a result, samples of the output are invalid due to time aliasing.
- The overlap-save method writes out the good samples and uses a hop size of  $M$ , thus recomputing the time-aliased output samples in the previous frame.

**Code :-**

```

%Overlapadd.m
%Written by Debagnik Kar 1804373
clc
clear all
close all
x=input('enter The sequence x')
h=input('enter the sequence h')
lx=length(x)
M=length(h)
N=2*M-1
L=N-M+1
x=[x,zeros(1,N-1)]
h=[h,zeros(1,N-M)]
k=floor(lx/M)+1
p=zeros(k,N)
for i=1:k
    p(i,1:L)=x((i-1)*L+1:(i-1)*L+L)
end
y=zeros(k,N)
for i=1:k
    y(i,:)=cconv(p(i,:),h,N)
end
for i=2:k
    y(i,1:M-1)= y(i,1:M-1)+y(i-1,L+1:N)
end
z=y(:,1:L)
a=z'
subplot(311)
stem(x)
title('x')
subplot(312)
stem(h)
title('h')
subplot(313)
stem(a(:))
title('overlap add')

%Overlapssave.m
%Written by Debagnik Kar 1804373
clc
close all
clear all
x = input('Enter sequence')
h = input('Enter an impulse response')
lx = length(x)
M = length(h)
N = 2*M-1
L = N-M+1
x = [x,zeros(1,N)]
h = [h,zeros(1,N-M)]

```

```

k = floor(lx/M)+1
p = zeros(k,N)
for i=1:k
    p(i,L:N)=x((i-1)*L+1:(i-1)*L+L)
end
for i=2:k
    p(i,1:M-1)=p(i-1,L+1:N)
end
y=zeros(k,N)
for i=1:k
    y(i,:)=cconv(p(i,:),h,N)
end
z = y(:,L:N)
u = z'
subplot(311)
stem(x)
title('x')
subplot(312)
stem(h)
title('h')
subplot(313)
stem(u(:))
title('Overlap Save')

```

### **Graph/Output :-**

OVERLAP-ADD Method:

```

x =
    1    8    0    4    3    7    3    0    0    0    0
h =
    4   -1    6    0    0
p =
    1    8    0    0    0
    4    3    7    0    0
    3    0    0    0    0
y =
    4.0000  31.0000  -2.0000  48.0000  -0.0000
   64.0000   8.0000  49.0000  11.0000  42.0000
   23.0000  39.0000  18.0000   0.0000    0
z =
    4.0000  31.0000  -2.0000
   64.0000   8.0000  49.0000
   23.0000  39.0000  18.0000
a =
    4.0000  64.0000  23.0000
   31.0000   8.0000  39.0000
   -2.0000  49.0000  18.0000

```

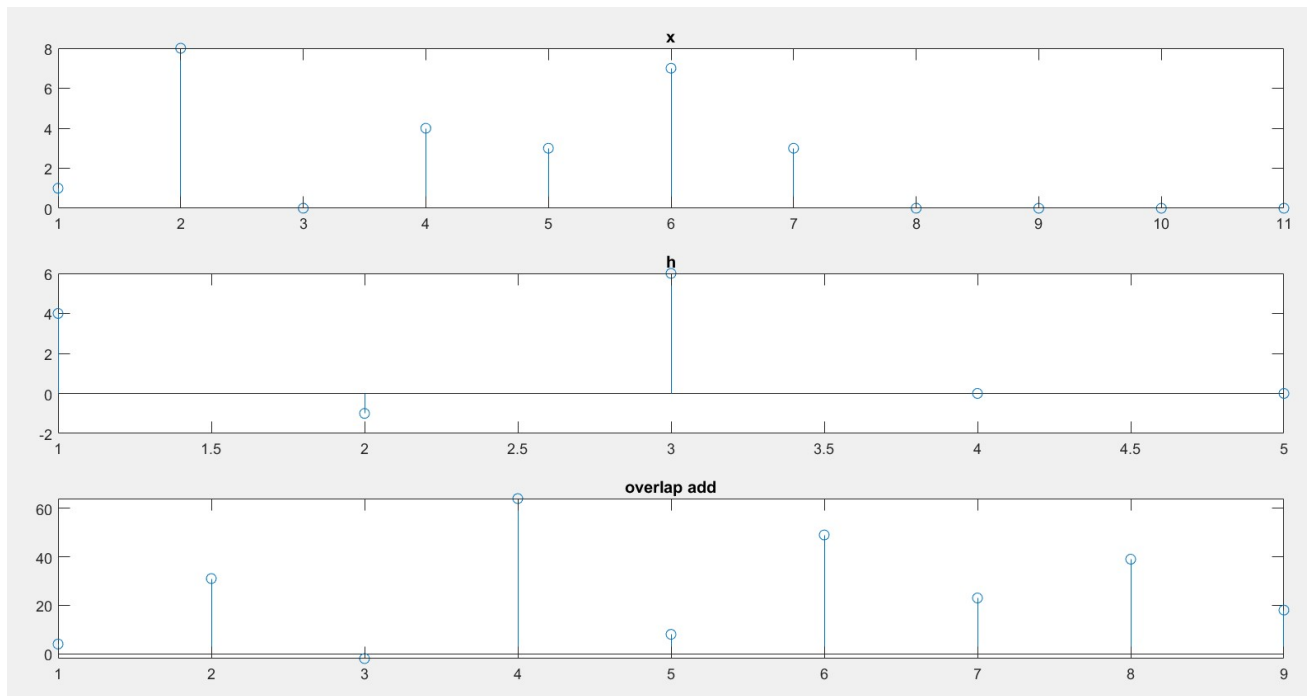


Fig 5.1: Linear convolution using Overlap-add method

### OVERLAP-SAVE Method:

$x =$

1 8 0 4 3 7 3 0 0 0 0 0

$h =$

4 -1 6 0 0

$k =$

3

$p =$

0 0 1 8 0

8 0 4 3 7

3 7 3 0 0

$y =$

48.0000 -0.0000 4.0000 31.0000 -2.0000

43.0000 34.0000 64.0000 8.0000 49.0000

12.0000 25.0000 23.0000 39.0000 18.0000

$z =$

4.0000 31.0000 -2.0000

64.0000 8.0000 49.0000

23.0000 39.0000 18.0000

$u =$

4.0000 64.0000 23.0000

31.0000 8.0000 39.0000

-2.0000 49.0000 18.0000

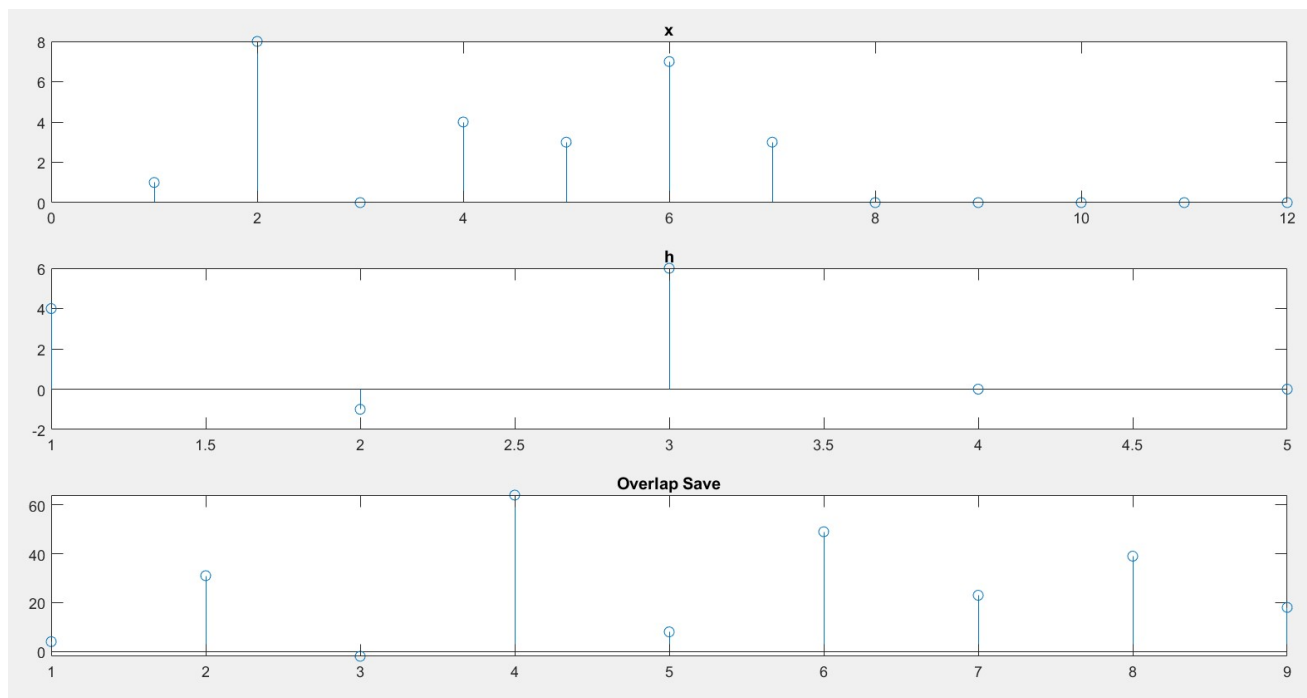


Fig 5.2: Linear convolution using overlap-save method

### **Discussion/Inference of the experiment :-**

In this experiment for the overlap add method we followed these steps to write the code

4. Broke the long sequence  $x(n)$ , into signals of length  $L$ .
5. Used the DFT-based method to calculate the convolution of each  $x_m(n)$  with  $h(n)$ .
6. Shifted each  $y_m(n)$  by  $mL$  samples and added the results together.

For the sequences

the first sequence [1 8 0 4 3 7 3]

the second sequence [4 -1 6]

The result was

[4 31 -2 48 0 64 8 49 11 42 23 39 18 0 0]

In this experiment for the overlap save method we followed these steps to write the code

3. cut the signal up into equal length segments with some overlap.

4. Then take the DFT of the segments and saves the parts of the convolution that correspond to the circular convolution.

For the sequences

the first sequence [1 8 0 4 3 7 3]

the second sequence [4 -1 6]

The result was

[4 31 -2 64 8 49 23 39 18]

**Conclusion :-**

In this experiment we learned about sectional convolution methods like overlap add and overlap save method and wrote codes to obtain linear convolution of given sequences using sectional convolution methods such as overlap add and overlap save. After successfully simulating the experiment we generated the required graphs using MATLAB whose behavior was in accordance with the theoretical concept.