Experiment 5 : Linear Filtering using OAM and OSM

Name	Prathamesh Mane
UID no. & Branch	**2022200078 (B1)**
Experiment No.	5

AIM:	To perform filtering of Long Data Sequence using Overlap Add Method and Overlap Save Method.
OBJECTIVE:	To Develop a function to implement Fast Overlap Add Algorithm and Overlap Save Algorithm
INPUT SPECIFICATION:	1. Length of first Signal L and Signal values 2. Length of impulse response of FIR filter Signal M and Signal values.
PROBLEM DEFINITION:	1. Take long input sequence x[n] and short length sequence h[n] 2. Find y[n] = x[n] * h[n] using FFT based Overlap Add Algorithm and Overlap Save Algorithm.

Case 1 : OAM
(Theoretical solution)

RESULT: case 1

Result analysis :

x[n] = { 1, 2, 3, 4, 5, 6, 1, 1, 1, 1, 1 , 1, 0, 1, 2, 3, 4, 5 }
h[n] = { 1, 1, 1 } Length M=3

Overlap Add Method For N=8, and M=3, Let L = 6

x1[n] = 1.00 2.00 3.00 4.00 5.00 6.00 0.00 0.00
y1[n] = 1.00 3.00 6.00 9.00 12.00 15.00 11.00 6.00

x2[n] = 1.00 1.00 1.00 1.00 1.00 1.00 0.00 0.00
y2[n] = 1.00 2.00 3.00 3.00 3.00 3.00 2.00 1.00

x3[n] = 0.00 1.00 2.00 3.00 4.00 5.00 0.00 0.00
y3[n] = 0.00 1.00 3.00 6.00 9.00 12.00 9.00 5.00

We have appended two zeros In the last to make it a length of ‘8’ hence the last two elements of y1[n] is added with the first two element and y2[n] and this goes on .This approach can help us to perform convolution of bigger input signal by breaking it into shorter sequence making the calculation simpler.

Hence y[n] = { 1, 3, 6, 9, 12, 15, 12, 8, 3, 3, 3, 2, 2, 3, 6, 9, 12, 9, 5 }

Code result :

```
Enter the length of signal x[n]: 18

Enter the values of signal x[n]: 1 2 3 4 5 6 1 1 1 1 1 1 0 1 2 3 4 5

Enter the length of impulse response h[n]: 3

Enter the values of impulse response h[n]: 1 1 1

x[n] =   1.00    2.00    3.00    4.00    5.00    6.00    1.00    1.00    1.00    1.00    1.00    1.00    0
    .00    1.00    2.00    3.00    4.00    5.00
h[n] =   1.00    1.00    1.00

Length of decomposed input signal: L = 6
Length of decomposed output signal: N = 8


x1[n] =   1.00    2.00    3.00    4.00    5.00    6.00    0.00    0.00
y1[n] =   1.00    3.00    6.00    9.00    12.00    15.00    11.00    6.00

x2[n] =   1.00    1.00    1.00    1.00    1.00    1.00    0.00    0.00
y2[n] =   1.00    2.00    3.00    3.00    3.00    3.00    2.00    1.00

x3[n] =   0.00    1.00    2.00    3.00    4.00    5.00    0.00    0.00
y3[n] =   0.00    1.00    3.00    6.00    9.00    12.00    9.00    5.00

Linear Convolution Output using Overlap Add Method:
y[n] =   1.00    3.00    6.00    9.00    12.00    15.00    12.00    8.00    3.00    3.00    3.00    3.00
     2.00    2.00    3.00    6.00    9.00    12.00    9.00    5.00
```

RESULT : case 2 :

Result analysis :

x[n] = { 6,12,7,14,8,16,9,18,1,2,3,4 }

h[n] = { 1, 1, 1 } Length M=3

Overlap Add Method For N=8, and M=3, Let L = 6

x1[n] = 6.00 12.00 7.00 14.00 8.00 16.00 0.00 0.00

y1[n] = 6.00 18.00 25.00 33.00 29.00 38.00 24.00 16.00

x2[n] = 9.00 18.00 1.00 2.00 3.00 4.00 0.00 0.00

y2[n] = 9.00 27.00 28.00 21.00 6.00 9.00 7.00 4.00

We have appended two zeros In the last to make it a length of ‘8’ hence the last two elements of y1[n] is added with the first two element and y2[n] and this goes on .This approach can help us to perform convolution of bigger input signal by breaking it into shorter sequence making the calculation simpler.

Hence y[n] = { 6.00 ,18.00 ,25.00 ,33.00 ,29.00 ,38.00 ,33.00, 43.00 ,28.00 ,21.00 ,6.00 ,9.00 ,7.00 ,4.00 }

Code result :

```
Enter the length of signal x[n]: 12

Enter the values of signal x[n]: 6 12 7 14 8 16 9 18 1 2 3 4

Enter the length of impulse response h[n]: 3

Enter the values of impulse response h[n]: 1 1 1

x[n] =   6.00    12.00    7.00    14.00    8.00    16.00    9.00    18.00    1.00    2.00    3.00    4.00
h[n] =   1.00    1.00    1.00

Length of decomposed input signal: L = 6
Length of decomposed output signal: N = 8


x1[n] =   6.00    12.00    7.00    14.00    8.00    16.00    0.00    0.00
y1[n] =   6.00    18.00    25.00    33.00    29.00    38.00    24.00    16.00

x2[n] =   9.00    18.00    1.00    2.00    3.00    4.00    0.00    0.00
y2[n] =   9.00    27.00    28.00    21.00    6.00    9.00    7.00    4.00

Linear Convolution Output using Overlap Add Method:
y[n] =   6.00    18.00    25.00    33.00    29.00    38.00    33.00    43.00    28.00    21.00    6.00    9
    .00    7.00    4.00
```

Case 2 : OSM
(Theoretical solution)

RESULT : case 1 :

Result analysis :

x[n] = { 1, 2, 3, 4, 5, 6, 1, 1, 1, 1, 1 , 1, 0, 1, 2, 3, 4, 5 }
h[n] = { 1, 1, 1 } Length M=3

Overlap save Method For N=8, and M=3, Let L = 6

x[n] = 0.00 0.00 1.00 2.00 3.00 4.00 5.00 6.00
y[n] = 11.00 6.00 1.00 3.00 6.00 9.00 12.00 15.00

x[n] = 5.00 6.00 1.00 1.00 1.00 1.00 1.00 1.00
y[n] = 7.00 12.00 12.00 8.00 3.00 3.00 3.00 3.00

x[n] = 1.00 1.00 0.00 1.00 2.00 3.00 4.00 5.00
y[n] = 10.00 7.00 2.00 2.00 3.00 6.00 9.00 12.00

x[n] = 4.00 5.00 0.00 0.00 0.00 0.00 0.00 0.00
y[n] = 4.00 9.00 9.00 5.00 0.00 0.00 0.00 0.00

We have added two zeros In the beginning and repeated the last two elements in the next sequences to make it a length of ‘8’ hence the first two elements of y1[n] is discarded , and first two element of y2[n] is also discarded and this goes on .This approach can help us to perform convolution of bigger input signal by breaking it into shorter sequence making the calculation simpler.

Hence y[n] = { 1, 3, 6, 9, 12, 15, 12, 8, 3, 3, 3, 2, 2, 3, 6, 9, 12, 9, 5 }

Code result :

```
Enter the length of x[n]: 18

Enter the values of x[n]: 1 2 3 4 5 6 1 1 1 1 1 1 0 1 2 3 4 5

Enter the length of h[n]: 3

Enter the values of h[n]: 1 1 1

x[n] =   1.00    2.00    3.00    4.00    5.00    6.00    1.00    1.00    1.00    1.00    1.00    1.00    0
    .00    1.00    2.00    3.00    4.00    5.00
h[n] =   1.00    1.00    1.00

Length of decomposed input signal: L = 6
Length of decomposed output signal: N = 8

x[n] =   0.00    0.00    1.00    2.00    3.00    4.00    5.00    6.00
y[n] =   11.00    6.00    1.00    3.00    6.00    9.00    12.00    15.00

x[n] =   5.00    6.00    1.00    1.00    1.00    1.00    1.00    1.00
y[n] =   7.00    12.00    12.00    8.00    3.00    3.00    3.00    3.00

x[n] =   1.00    1.00    0.00    1.00    2.00    3.00    4.00    5.00
y[n] =   10.00    7.00    2.00    2.00    3.00    6.00    9.00    12.00

x[n] =   4.00    5.00    0.00    0.00    0.00    0.00    0.00    0.00
y[n] =   4.00    9.00    9.00    5.00    0.00    0.00    0.00    0.00

Linear Convolution Output using Overlap Save Method:
y[n] =   1.00    3.00    6.00    9.00    12.00    15.00    12.00    8.00    3.00    3.00    3.00    3.00
     2.00    2.00    3.00    6.00    9.00    12.00    9.00    5.00
```

RESULT : case 1 :

Result analysis :

x[n] = { 1, 2, 3, 4, 5, 6, 1, 1, 1, 1, 1 , 1, 0, 1, 2, 3, 4, 5 }
h[n] = { 1, 1, 1 } Length M=3

Overlap save Method For N=8, and M=3, Let L = 6

x[n] =   0.00   0.00   6.00   12.00   7.00   14.00   8.00   16.00
y[n] =   24.00   16.00   6.00   18.00   25.00   33.00   29.00   38.00

x[n] =   8.00   16.00   9.00   18.00   1.00   2.00   3.00   4.00
y[n] =   15.00   28.00   33.00   43.00   28.00   21.00   6.00   9.00

x[n] =   3.00   4.00   0.00   0.00   0.00   0.00   0.00   0.00
y[n] =   3.00   7.00   7.00   4.00   0.00   0.00   -0.00   0.00

We have added two zeros In the beginning and repeated the last two elements in the next sequences to make it a length of ‘8’ hence the first two elements of y1[n] is discarded , and first two element of y2[n] is also discarded and this goes on .This approach can help us to perform convolution of bigger input signal by breaking it into shorter sequence making the calculation simpler.

Hence y[n] = { 6.00 ,18.00 ,25.00 ,33.00 ,29.00 ,38.00 ,33.00, 43.00 ,28.00 ,21.00 ,6.00 ,9.00 ,7.00 ,4.00 }

Code result :

```
Enter the length of x[n]: 12

Enter the values of x[n]: 6 12 7 14 8 16 9 18 1 2 3 4

Enter the length of h[n]: 3

Enter the values of h[n]: 1 1 1

x[n] =   6.00    12.00    7.00    14.00    8.00    16.00    9.00    18.00    1.00    2.00    3.00    4.00
h[n] =   1.00    1.00    1.00

Length of decomposed input signal: L = 6
Length of decomposed output signal: N = 8

x[n] =   0.00    0.00    6.00    12.00    7.00    14.00    8.00    16.00
y[n] =   24.00    16.00    6.00    18.00    25.00    33.00    29.00    38.00

x[n] =   8.00    16.00    9.00    18.00    1.00    2.00    3.00    4.00
y[n] =   15.00    28.00    33.00    43.00    28.00    21.00    6.00    9.00

x[n] =   3.00    4.00    0.00    0.00    0.00    0.00    0.00    0.00
y[n] =   3.00    7.00    7.00    4.00    0.00    0.00    -0.00    0.00

Linear Convolution Output using Overlap Save Method:
y[n] =   6.00    18.00    25.00    33.00    29.00    38.00    33.00    43.00    28.00    21.00    6.00    9.
    .00    4.00
```

CONCLUSION:

1. The Overlap-Add and Overlap-Save methods provide an efficient and practical approach to evaluating the discrete convolution of long input sequences (x[n]) with finite-length impulse responses (h[n]), significantly reducing computational complexity.

2. These methods are specifically designed for FIR filters and are not suitable for IIR filters due to the infinite response nature of IIR filters, which complicates the segmentation and recombination process.

3. As block processing techniques, both Overlap-Add and Overlap-Save methods process data in segments, allowing for effective handling of long sequences in real-time applications while maintaining computational efficiency.

4. The Overlap-Add and Overlap-Save methods offer efficient ways to perform convolution on long sequences, leveraging the power of FFT.

	These techniques are essential in scenarios where direct convolution is impractical due to its high computational cost. By breaking the problem into smaller, manageable parts, these methods make real-time processing of signals possible in various applications, including audio processing, communications, and control systems.