Digital Signal Processing
Module 8
Fast Convolution using Overlap add and Save method

Objective:

To understand and apply the overlap add and overlap save methods to find the response of LSI systems.

Introduction:

Inspite of its computational advantages, there are some difficulties with the DFT approach for finding linear convolution. For example, if *x(n)* is *very long*, we must commit a significant amount of time computing very long DFTs and in the process accept very long processing delays. In some cases, it may even be possible that *x(n)* is *too long* to compute the DFT. The solution to these problems is to use *block convolution*, which involves segmenting the signal to be filtered, *x(n)*, into sections. Each section is then filtered with the FIR filter *h(n)*, and the filtered sections are pieced together to form the sequence *y(n)*. There are two block convolution techniques. The first is overlap-add, and the second is overlap-save.

Description:

Overlap Save Method:

Let the length of an input sequence be L_S and the length of an impulse response is M. In this method the input sequence is divided into blocks of data of size $N = L+M-1$. Each block consists of last $(M-1)$ data points of previous block followed by L new data points to form a data sequence of length $N = L + M - 1$. For first block of data the first $M - 1$ points are set to zero. Thus the blocks of data sequence are

$$x_1(n) = \{\underbrace{0, 0, 0, \ldots 0}_{(M-1)\text{ Zeros}}, x(0), x(n) \ldots, x(L-1)\}$$

$$x_2(n) = \{\underbrace{x(L-M+1), \ldots, x(L-1)}_{\text{Last }(M-1)\text{ data points from }x_1(n)}, \underbrace{x(L) \ldots, x(2L-1)}_{L\text{ new data points}}$$

$$x_3(n) = \{\underbrace{x(2L-M+1), \ldots, x(2L-1)}_{\text{Last }(M-1)\text{ data points from }x_2(n)}, \underbrace{x(2L) \ldots, x(3L-1)}_{L\text{ new data points}}\}$$

and so on.

Now the impulse response of the FIR filter is increased in length by appending $L-1$ zeros and an N-point circular convolution of $x_i(n)$ with $h(n)$ is computed.

$$\text{i.e.,} \quad y_i(n) = x_i(n) \circledN h(n)$$

In $y_i(n)$, the first $(M-1)$ points will not agree with the linear convolution of $x_i(n)$ and $h(n)$ because of aliasing, while the remaining points are identical to the linear convolution. Hence we discard the first $M-1$ points of the filtered section $x_i(n) \circledN h(n)$. The remaining points from successive sections are then abutted to construct the final filtered output.

For example, let the total length of the sequence $L_S = 15$ and the length of the impulse response is 3. Let the length of each block is 5.

Now the input sequence can be divided into blocks as

$$x_1(n) = \{\underbrace{0, 0,}_{M-1=2 \text{ zeros}} x(0), x(1), x(2)\}$$

$$x_2(n) = \{\underbrace{x(1), x(2)}, x(3), x(4), x(5)\}$$

$\downarrow$ Last two data points from previous block
$\uparrow$

$$x_3(n) = \{\overbrace{x(4), x(5)}, x(6), x(7), x(8)\}$$

$$x_4(n) = \{x(7), x(8), x(9), x(10), x(11)\}$$

$$x_5(n) = \{x(10), x(11), x(12), x(13), x(14)\}$$

$$x_6(n) = \{x(13), x(14), 0, 0, 0\}$$

Now we perform 5 point circular convolution of $x_i(n)$ and $h(n)$ by appending two zeros to the sequence $h(n)$. In the output block $y_i(n)$, first $M-1$ points are corrupted and must be discarded.

$$y_1(n) = x_1(n) \textcircled{N} h(n) = \{\underbrace{y_1(0), y_1(1)}_{\text{discard}}, y_1(2), y_1(3), y_1(4)\}$$

$$y_2(n) = x_2(n) \textcircled{N} h(n) = \{\underbrace{y_2(0), y_2(1)}_{\text{discard}}, y_2(2), y_2(3), y_2(4)\}$$

$$y_3(n) = x_3(n) \textcircled{N} h(n) = \{\underbrace{y_3(0), y_3(1)}_{\text{discard}}, y_3(2), y_3(3), y_3(4)\}$$

$$y_4(n) = x_4(n) \textcircled{N} h(n) = \{\underbrace{y_4(0), y_4(1)}_{\text{discard}}, y_4(2), y_4(3), y_4(4)\}$$

$$y_5(n) = x_5(n) \textcircled{N} h(n) = \{\underbrace{y_5(0), y_5(1)}_{\text{discard}}, y_5(2), y_5(3), y_5(4)\}$$

$$y_6(n) = x_6(n) \textcircled{N} h(n) = \{\overbrace{y_6(0), y_6(1)}, y_6(2), y_6(3), 0\}$$

The output blocks are abutted together to get

$$y(n) = \{y_1(2), y_1(3), y_1(4), y_2(2), y_2(3), y_2(4), y_3(2), y_3(3), y_3(4), y_4(2), y_4(3), y_4(4), y_5(2), y_5(3), y_5(4), y_6(2), y_6(3), \}$$

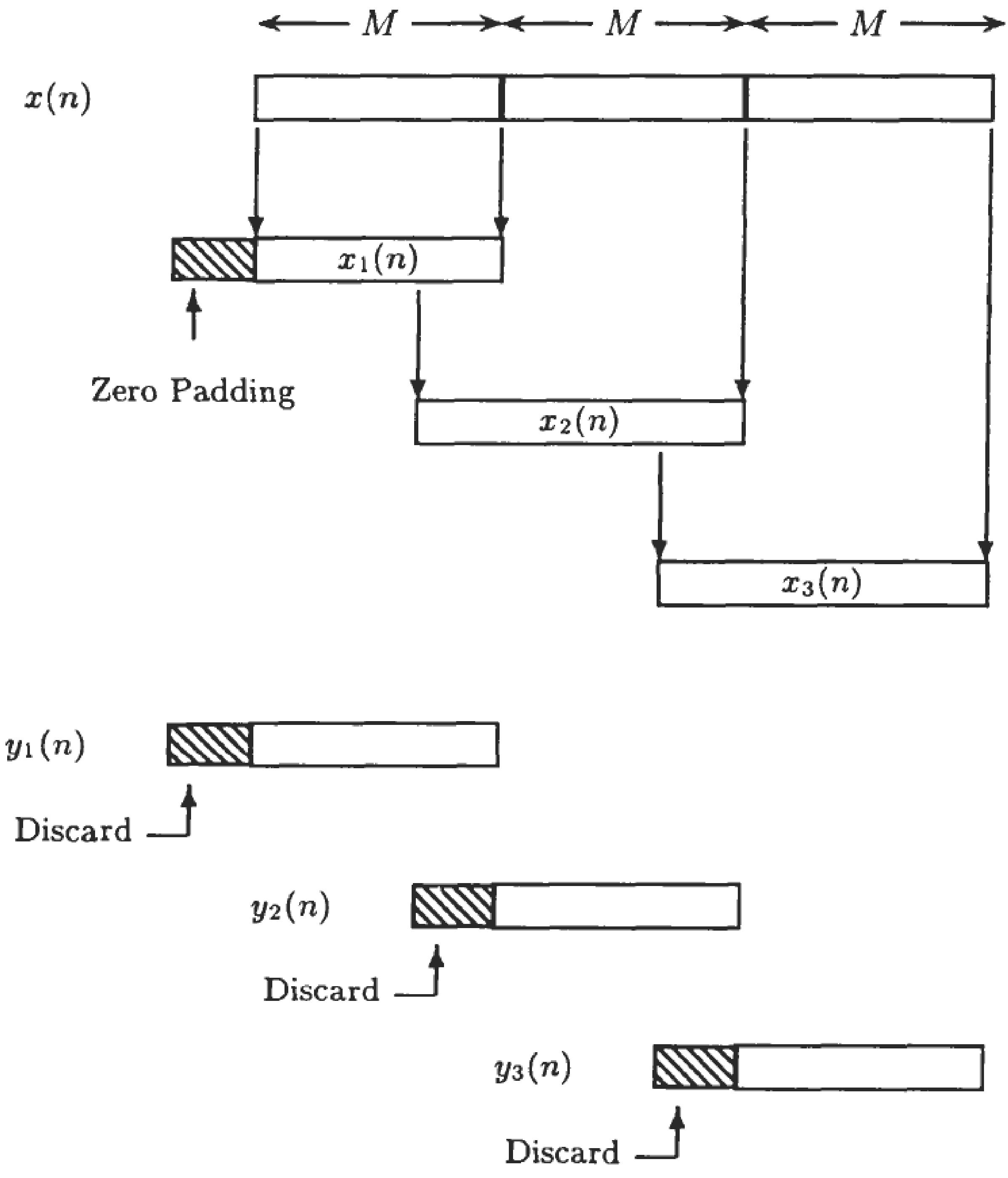

Figure 8.1. Illustration of Overlap-save method of block convolution

Overlap Add Method:

Let the length of the sequence be L_S and the length of the impulse response is M. The sequence is divided into blocks of data size having length L and $M-1$ zeros are appended to it to make the data size of $L+M-1$.

Thus the data blocks may be represented as

$$x_1(n) = \{x(0), x(1), \ldots, x(L-1), \underbrace{0, 0, \ldots}_{}\}$$

$\longrightarrow M-1$ zeros appended

$$x_2(n) = \{x(L), x(L+1), \ldots, x(2L-1), \overbrace{0, 0, \ldots}\}$$

$$x_3(n) = \{x(2L), x(2L+1), \ldots, x(3L-1), \underbrace{0, 0, \ldots}_{}\}$$

$\longrightarrow M-1$ zeros appended

Now $L-1$ zeros are added to the impulse response $h(n)$ and N-point circular convolution is performed. Since each data block is terminated with $M-1$ zeros, the last $M-1$ points from each output block must be overlapped and added to the first $M-1$ points of the succeeding block. Hence this method is called overlap-add method.

Let the output blocks are of the form

$$y_1(n) = \{y_1(0), y_1(1), \ldots, y_1(L-1), y_1(L), \ldots, y_1(N-1)\}$$
$$y_2(n) = \{y_2(0), y_2(1), \ldots, y_2(L-1), y_2(L), \ldots, y_2(N-1)\}$$
$$y_3(n) = \{y_3(0), y_3(1), \ldots, y_3(L-1), y_3(L), \ldots, y_3(N-1)\}$$

The output sequence is

$$y(n) = \{y_1 0, y_1(1), \ldots, y_1(L-1), y_1(L) + y_2(0), \ldots, y_1(N-1) + y_2(M-2),$$
$$y_2(M), \ldots, y_2(L) + y_3(0), y_2(L+1) + y_3(1), \ldots, y_3(N-1)\}$$

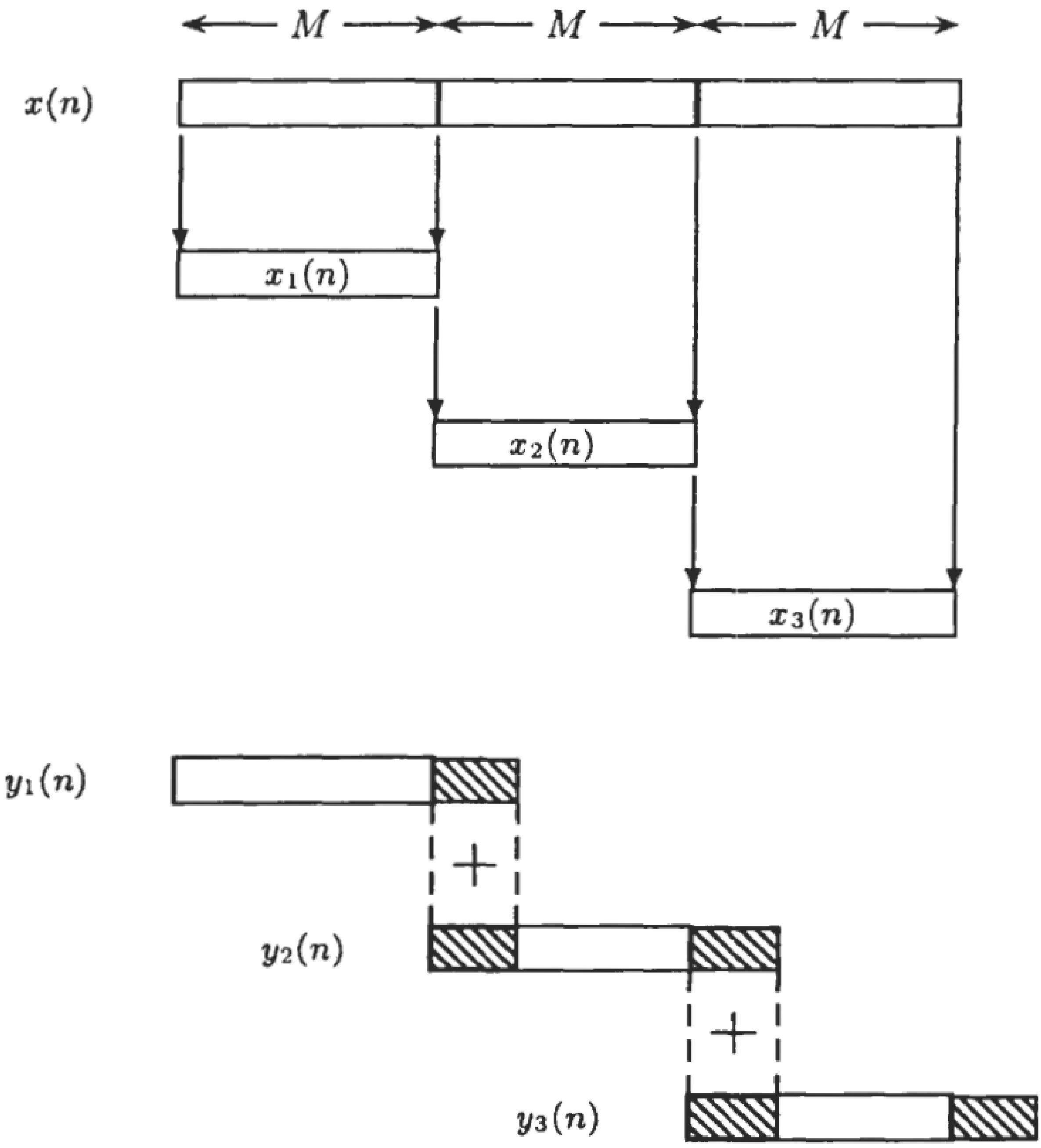

Figure 8.2. Partitioning a sequence into subsequences of length M for the overlap-add method of block convolution

Illustrative Examples:

Problem 1: Find the output y(n) of a filter whose impulse response is h(n)={1,1,1} and input signal x(n)={3,-1,0,1,3,2,0,1,2,1}using

i) Overlap - save method

ii) Overlap - add method

Solution:

(i) Overlap-save Method

The input sequence can be divided into blocks of data as follows.

$$x_1(n) = \{\underbrace{0,0}_{M-1=2 \text{ Zeros}} \quad \underbrace{3,-1,0}_{L=3 \text{ data points}}\}$$

$$x_2(n) = \{\underbrace{-1,0,}_{\substack{\text{Two datas}\\ \text{from previous}\\ \text{block}}} \quad \underbrace{1,3,2}_{\substack{\text{3new}\\ \text{data points}}}\}$$

$$x_3(n) = \{3,2,0,1,2\} \text{ and } x_4(n) = \{1,2,1,0,0\}$$

given $h(n) = \{1,1,1\}$

Increase the length of the sequence to $L + M - 1 = 5$ by adding two zeros.

i.e. $h(n) = \{1,1,1,0,0\}$

$$y_1(n) = x_1(n) \textcircled{N} h(n) = \{-1,0,3,2,2\}$$

$$y_2(n) = x_2(n) \textcircled{N} h(n) = \{4,1,0,4,6\}$$

$$y_3(n) = x_3(n) \textcircled{N} h(n) = \{6,7,5,3,3\}$$

$$y_4(n) = x_4(n) \textcircled{N} h(n) = \{1,3,4,3,1\}$$

Note: Circular convolution of the sequences left as an exercise to the students.

$$\boxed{\underbrace{-1,0,}_{\text{discard}}3,2,2}$$

$$\boxed{\underbrace{4,1,}_{\text{discard}}0,4,6}$$

$$\boxed{\underbrace{6,7,}_{\text{discard}}5,3,3}$$

$$\boxed{\underbrace{1,3,}_{\text{discard}}4,3,1}$$

$$y(n) = \{3,2,2,0,4,6,5,3,3,4,3,1\}$$

(ii) Overlap-Add method

Let the length of data block be 3. Two zeros are added to bring the length to five $(L + M - 1 = 5)$.

Therefore,

$$x_1(n) = \{3, -1, 0, 0, 0\}$$
$$x_2(n) = \{1, 3, 2, 0, 0\}$$
$$x_3(n) = \{0, 1, 2, 0, 0\}$$
$$x_4(n) = \{1, 0, 0, 0, 0\}$$
$$y_1(n) = x_1(n) \, (N) \, h(n) = \{3, 2, 2, -1, 0\}$$
$$y_2(n) = x_2(n) \, (N) \, h(n) = \{1, 4, 6, 5, 2\}$$
$$y_3(n) = x_3(n) \, (N) \, h(n) = \{0, 1, 3, 3, 2\}$$
$$y_4(n) = x_4(n) \, (N) \, h(n) = \{1, 1, 1, 0, 0\}$$

3, 2, 2, −1, 0
↕ ↕ add
1, 4, 6, 5, 2
↕ ↕ add
0, 1, 3, 3, 2
↕ ↕ add
1, 1, 1, 0, 0

$$y(n) = \{3, 2, 2, 0, 4, 6, 5, 3, 3, 4, 3, 1\}$$

Summary:

Therefore for long duration sequences the response of an LSI system can be found by using block convolution known as fast convolution using overlap - add and overlap - save methods faster than the methods using DFT and IDFT.

Assignment:

Problem 1: Find the output y(n) of a filter whose impulse response is h(n)={1,2} and input signal x(n)={1,2,-1,2,3,-2,-3,-1,1,1,2,-1}using

i) Overlap - save method
ii) Overlap - add method

Simulation:

%Overlap – Add method

```
clear
clc
x=input('enter x:');
h=input('enter h:');
subplot(3,1,1);
stem(0:length(x)-1,x);
xlabel('n---->');
ylabel('x(n)--->');
title('Input sequence');
subplot(3,1,2);
stem(0:length(h)-1,h);
xlabel('n---->');
ylabel('h(n)--->');
title('Impulse response sequence');
display('The fast convolution result using overlap-add method is:')
y=fftfilt(h,x)
subplot(3,1,3);
stem(0:length(y)-1,y);
xlabel('n---->');
ylabel('y(n)--->');
title('Fast Convolution using Overlap-Add method');
```

Input:

enter x:[3,-1,0,1,3,2,0,1,2,1]

enter h:[1 1 1]

Output:

The fast convolution result using overlap-add method is:

y = 3 2 2 0 4 6 5 3 3 4

References:

1. Digital Signal Processing, Principles, Algorithms and Applications – John G Proakis, Dimitris G Manolakis, Pearson Education / PHI, 2007
2. Discrete Time Signal Processing – A V Oppenheim and R W Schaffer, PHI, 2009
3. Digital Signal Processing – Monson H.Hayes – Schaum's Outlines, McGraw-Hill,1999
4. Fundamentals of Digital Signal Processing using Matlab – Robert J Schilling, Sandra L Harris, Thomson 2007.
5. Digital Signal processing – A Practical Approach, Emmanuel C Ifeachor and Barrie W Jervis, 2nd Edition, PE 2009
6. Digital Signal Processing – A Computer Based Approach, Sanjit K.Mitra, McGraw Hill,2nd Edition, 2001