# Experiment 5
# Overlap and Save Algorithm

## Aim

To execute overlap and save algorithm for linear filtering of a long data sequence.

## Theory

The input sequence $x(n)$ is often a very long sequence when real-time signal processing is done. We will list out the method for real-time filtering of long data sequences.

**Step 1:** Make use of input buffer of suitable size (say 128 or 256, i.e., some power of 2) to acquire the input signal data, as we want to make use of the FFT computation.

**Step 2:** When the input buffer is full, transfer the data to the next buffer, from where the data will be taken for processing.

**Step 3:** Use the data from the second buffer for filtering and in the background, let the input buffer be used for grabbing next samples.

**Step 4:** Allow the filtering operation to be completed when the input buffer is again full. This will then be termed as real-time processing.

For linear filtering of a long sequence, we make use of FFT to execute circular convolution of the two sequences. The long sequence is divided in small segments of size $L$. If the size of each block of data is $L$ and the size of filter length is $M$ then the convolved sequence will have length equal to $L + M - 1$. So, we have to append zeros at the end of each data block in the so as to make the processing block size of $L + M - 1$. We will use overlap and save algorithm.

***For detailed theory of overlap and save algorithm refer to Section 5.9 on fast convolution.***

## Experiment

Let the input sequence be

$$x(n) = [1\ 2\ 3\ 4\ 5\ 1\ 2\ 3\ 4\ 5\ 1\ 2\ 3\ 4\ 5]$$

Let the impulse response of the filter be

$$h(n) = [3\ 2\ 1]$$

We will use block size for the data as $L = 5$ and $M = 4$. We have to append three zeros at the beginning of first data block and four zeros to $h(n)$ so that the length of both is $L + M - 1 = 5 + 4 - 1 = 8$.

For subsequent blocks we append the last three samples of the previous block at the beginning of the next block. For the first block, we will append extra three zeros at the start.

The data blocks will be

$$x_1(n) = [0\ 0\ 0\ 1\ 2\ 3\ 4\ 5]$$

$$x_2(n) = [3\ 4\ 5\ 1\ 2\ 3\ 4\ 5]$$

$$x_3(n) = [3\ 4\ 5\ 1\ 2\ 3\ 4\ 5]$$

We have to take FFT of each block, multiply it with FFT of $h(n)$ and take IFFT of the multiplication output. This is the convolution result. The result of convolution of the data blocks is obtained using a MATLAB program.

The output convolved sequences obtained are

$$y_1(n) = [17\ 9\ 5\ 3\ 8\ 14\ 21\ 28]$$

$$y_2(n) = [26\ 27\ 31\ 20\ 17\ 19\ 21\ 28]$$

$$y_3(n) = [26\ 27\ 31\ 20\ 17\ 19\ 21\ 28]$$

We will now discard the first three output samples from each output sequence and save the non-overlapping outputs. First $M - 1$ samples of the first block are appended at the end by circular shifting. When we are appending $M - 1$ extra zeros at the start of the data block, the resulting output convolved sequence will also be circularly rotated by $M - 1$ samples (refer to Problem 9 in Chapter 5).

The result of convolution is

$$[3\ \ 8\ \ 14\ \ 21\ \ 28\ \ 20\ \ 17\ \ 19\ \ 21\ \ 28\ \ 20\ \ 17\ \ 19\ \ 21\ \ 28\ \ 17\ \ 9\ \ 5]$$

**Teaser** *The reader is encouraged to find the result of direct convolution of the two sequences and verify that the result is same as the result of overlap and save algorithm.*

The MATLAB program is as follows.

```
%overlap & save algorithm
clear all;
%a=[1 2 3 4 5 1 2 3 4 5 1 2 3 4 5];
a=[1 2 3 2 3 4 3 2 1 4 5 6 7 8 9];
b=[1 2 1 0 0 0 0 0];
c=conv(a,b);
disp(c);
b1=fft(b,8);
%a1=[0 0 0 1 2 3 4 5];
a1=[0 0 0 1 2 3 2 3];
a11=fft(a1,8);
```

```
for i=1:8,
   a111(i)=a11(i)*b1(i);
end
c1=ifft(a111,8);
disp(c1);
%a2=[3 4 5 1 2 3 4 5];
a2=[3 2 3 4 3 2 1 4];
a22=fft(a2,8);
for i=1:8,
   a222(i)=a22(i)*b1(i);
end
c2=ifft(a222,8);
disp(c2);
%a3=[3 4 5 1 2 3 4 5];
a3=[2 1 4 5 6 7 8 9];
a33=fft(a3,8);
for i=1:8,
   a333(i)=a33(i)*b1(i);
end
   c3=ifft(a333,8);
   disp(c3);
   %a4=[3 4 5 0 0 0 0 0];
   a4=[7 8 9 0 0 0 0 0];
   a44=fft(a4,8);
   for i=1:8,
      a444(i)=a44(i)*b1(i);
   end
   c4=ifft(a444,8);
   disp(c4);
   for i=1:5,
   d(i)=c1(3+i);
end
for i=6:10,
   d(i)=c2(i-2);
end
for i=11:15,
   d(i)=c3(i-7);
end
for i=16:20,
   d(i)=c4(i-12);
end
disp(d);
```

The result of MATLAB program is

3 8 14 21 28 20 17 19 21 28 20 17 19 21 28 17 9 5