Digital Signal Processing

Ms. Ashwini

Department of Electronics and Communication.

Digital Signal Processing

Linear Filtering methods based on the DFT

Ms. Ashwini
Department of Electronics and Communication.

Overlap-Add Method

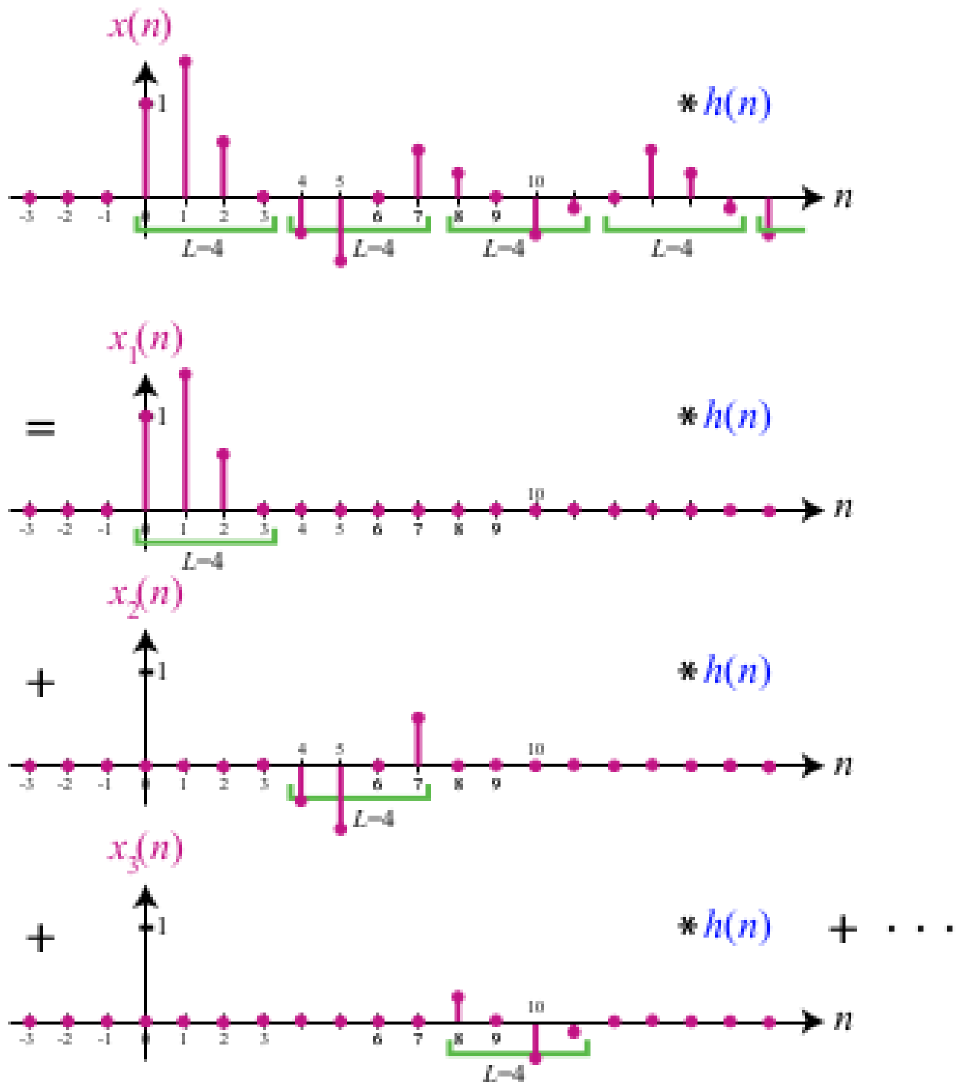
Deals with the following signal processing principles:

- The linear convolution of a discrete-time signal of length L and a discrete-time signal of length M produces a discrete-time convolved result of length $L + M - 1$.
- Addititvity:

$$(x_1(n)+x_2(n)) * h(n) = x_1(n) * h(n)+x_2(n) * h(n)$$

Signals, Linear Filtering methods based on the DFT

Overlap-Add Method



Input $x(n)$ is divided into non-overlapping blocks $x_m(n)$ each of length L.

Each input block $x_m(n)$ is individually filtered as it is received to produce the output block $y_m(n)$.

Overlap-Add Method

- makes use of the N-DFT and N-IDFT where: $N = L + M - 1$
  - Thus, zero-padding of $x(n)$ and $h(n)$ that are of length $L, M < N$ is required.
  - The actual implementation of the DFT/IDFT will use the FFT/IFFT for computational simplicity.

Overlap-Add Method

Let $x_m(n)$ have support $n = 0, 1, \ldots, L-1$.
Let $h(n)$ have support $n = 0, 1, \ldots, M-1$.

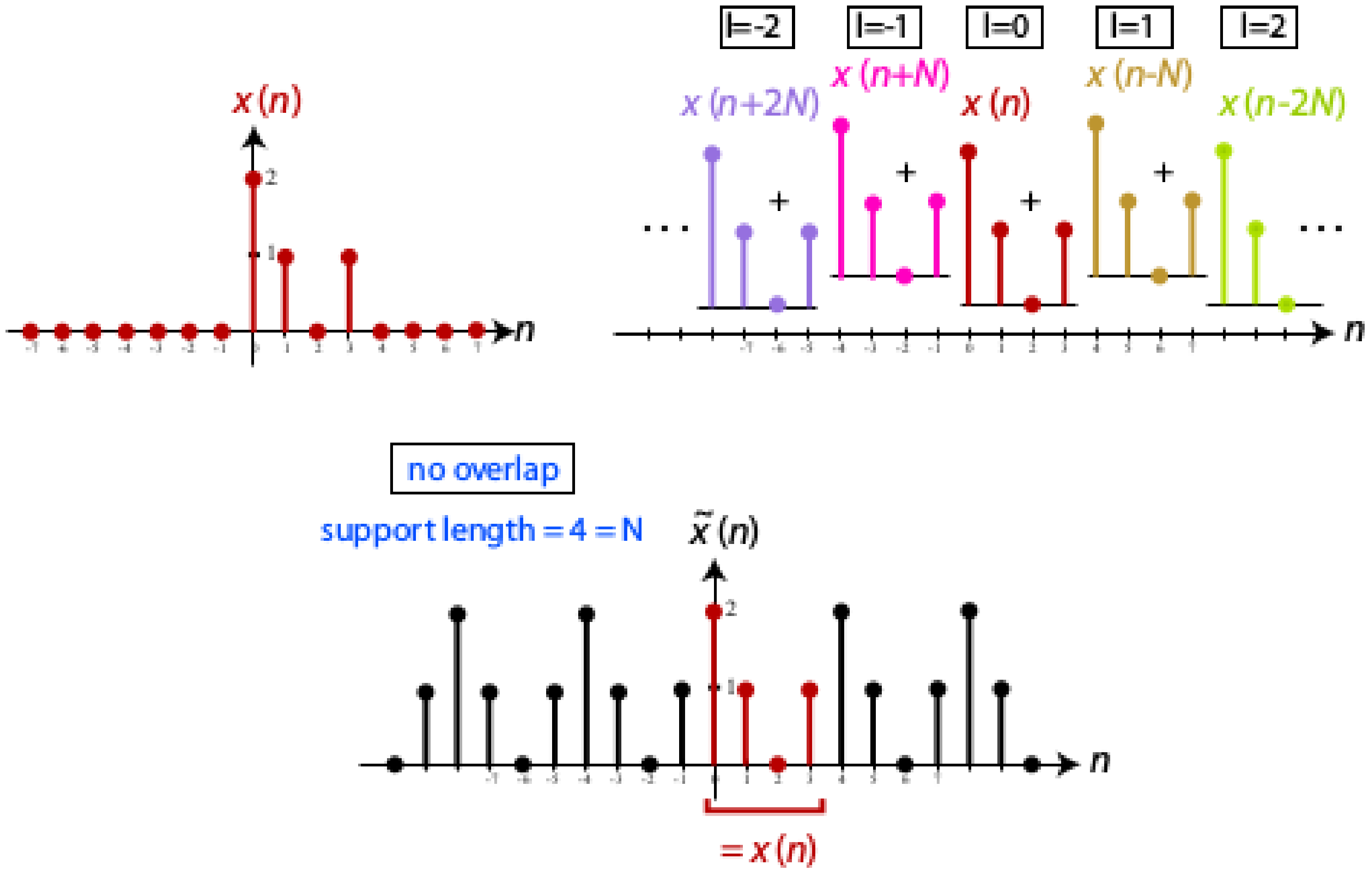
We set $N \geq L + M - 1$ (the length of the linear convolution result) and zero pad $x_m(n)$ and $h(n)$ to have support $n = 0, 1, \ldots, N-1$.

1. Take N-DFT of $x_m(n)$ to give $X_m(k)$, $k = 0, 1, \ldots, N-1$.
2. Take N-DFT of $h(n)$ to give $H(k)$, $k = 0, 1, \ldots, N-1$.
3. Multiply: $Y_m(k) = X_m(k) \cdot H(k)$, $k = 0, 1, \ldots, N-1$.
4. Take N-IDFT of $Y_m(k)$ to give $y_m(n)$, $n = 0, 1, \ldots, N-1$.

Signals, Linear Filtering methods based on the DFT

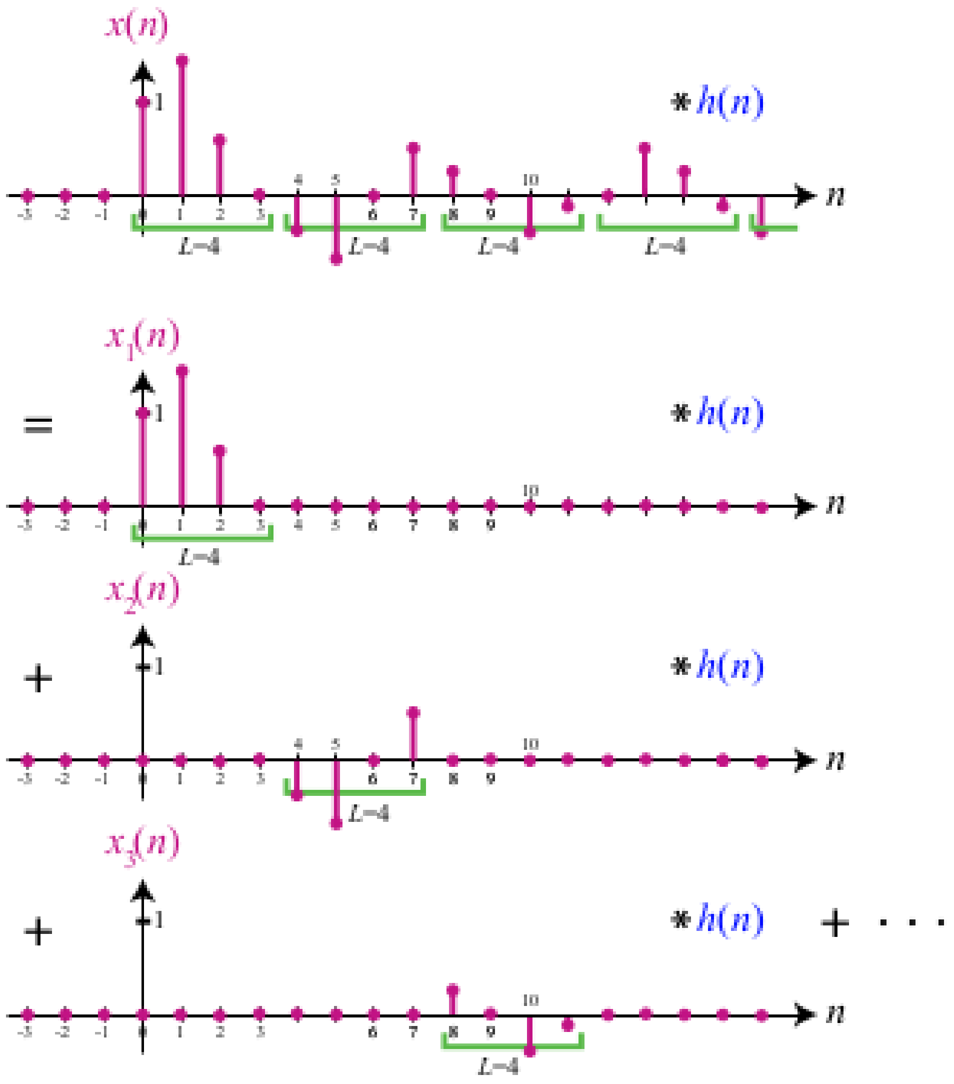
Overlap-Add Method

Length of linear convolution result = Length of DFT



Signals, Linear Filtering methods based on the DFT

Overlap-Add Method



Input $x(n)$ is divided into non-overlapping blocks $x_m(n)$ each of length L.

Each input block $x_m(n)$ is individually filtered as it is received to produce the output block $y_m(n)$.

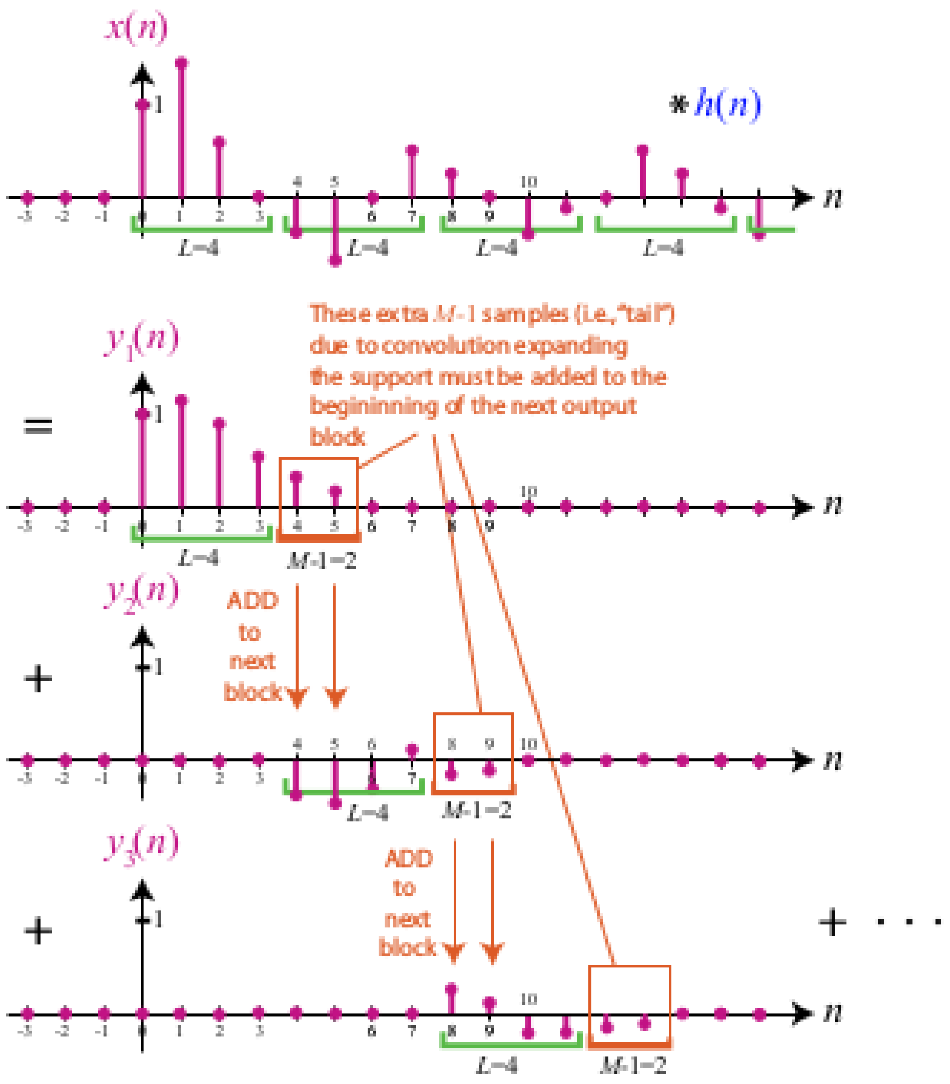
Overlap-Add Method

From the Addititvity property, since:

$$
\begin{aligned}
x(n) &= x_1(n) + x_2(n) + x_3(n) + \cdots = \sum_{m=1}^{\infty} x_m(n) \\
x(n) * h(n) &= (x_1(n) + x_2(n) + x_3(n) + \cdots) * h(n) \\
&= x_1(n) * h(n) + x_2(n) * h(n) + x_3(n) * h(n) + \cdots \\
&= \sum_{m=1}^{\infty} x_m(n) * h(n) = \sum_{m=1}^{\infty} y_m(n)
\end{aligned}
$$

Signals, Linear Filtering methods based on the DFT

Overlap-Add Method



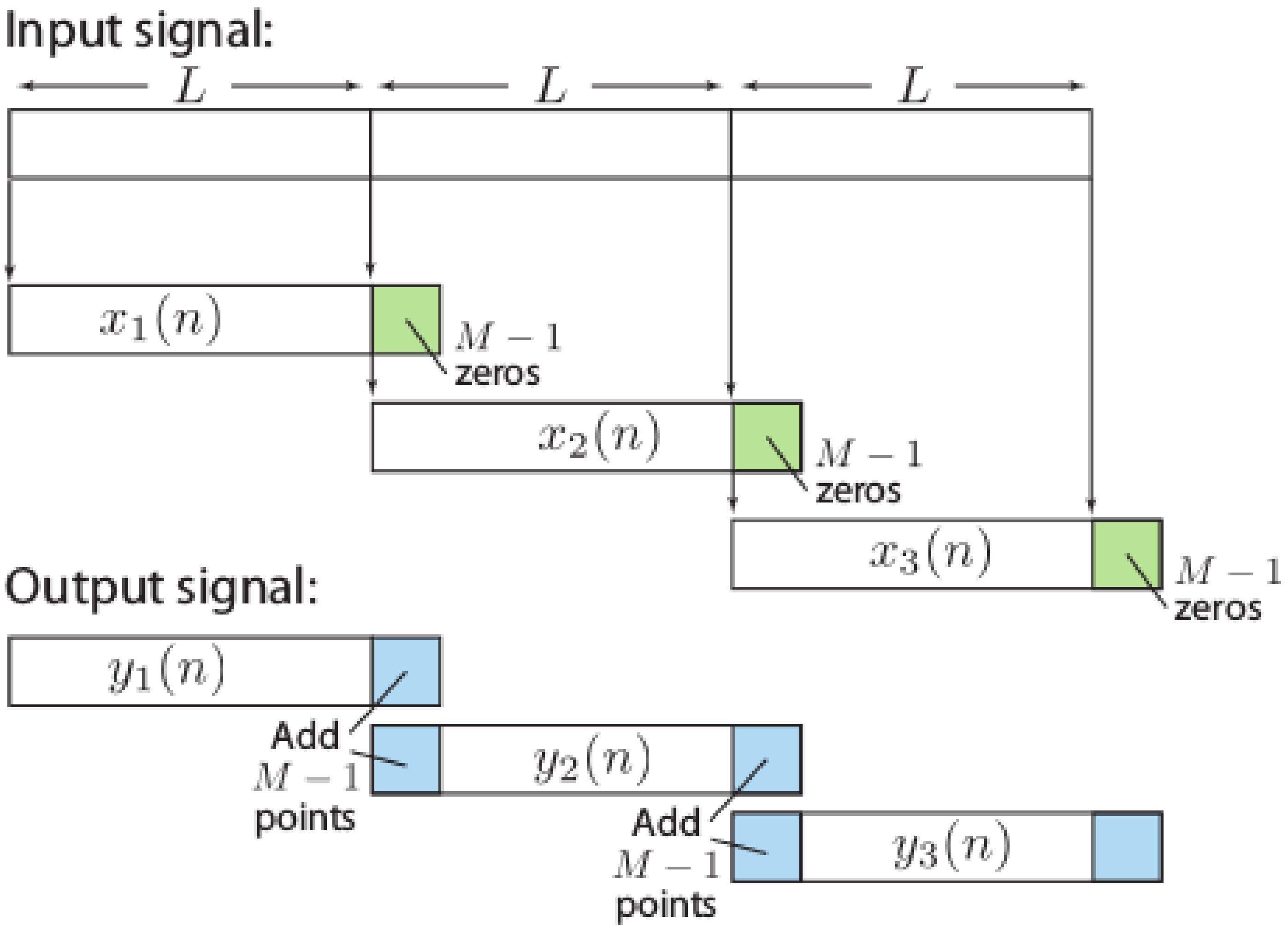
Output blocks $y_m(n)$ must be fitted together appropriately to generate:

$$y(n) = x(n) * h(n)$$

The support overlap amongst the $y_m(n)$ blocks must be accounted for.

Signals, Linear Filtering methods based on the DFT

Overlap-Add Method



Overlap-Add Method

1. Break the input signal $x(n)$ into non-overlapping blocks $x_m(n)$ of length L.
2. Zero pad $h(n)$ to be of length $N = L + M - 1$.
3. Take N-DFT of $h(n)$ to give $H(k)$, $k = 0, 1, \ldots, N - 1$.
4. For each block m:
   - 4.1 Zero pad $x_m(n)$ to be of length $N = L + M - 1$.
   - 4.2 Take N-DFT of $x_m(n)$ to give $X_m(k)$, $k = 0, 1, \ldots, N - 1$.
   - 4.3 Multiply: $Y_m(k) = X_m(k) \cdot H(k)$, $k = 0, 1, \ldots, N - 1$.
   - 4.4 Take N-IDFT of $Y_m(k)$ to give $y_m(n)$, $n = 0, 1, \ldots, N - 1$.
5. Form $y(n)$ by overlapping the last $M - 1$ samples of $y_m(n)$ with the first $M - 1$ samples of $y_{m+1}(n)$ and adding the result.

Overlap-Add Method

If you DO NOT overlap and add, but only append the output blocks $y_m(n)$ for $m = 1, 2, \ldots,$ then you will not get the true $y(n)$ sequence.

Overlap-Add Method

Signal x[n] (time domain):[3, -1, 0, 3, 2, 0, 1, 2, 1]

Filter h[n] (time domain): [1, -1, 1] M=3

If N=5

N = L + M - 1 = L + 3 - 1 = 5 Thus L=3

n	0	1	2	3	4	5	6	7	8	9	10
x[n]	3	-1	0	3	2	0	1	2	1	0	0
x_0[n]	3	-1	0	0	0	0	0	0	0	0	0
x_1[n-3]	0	0	0	3	2	0	0	0	0	0	0
x_2[n-6]	0	0	0	0	0	0	1	2	1	0	0

Signals, Linear Filtering methods based on the DFT

Overlap-Add Method

Computing y_0[n] Using Method 1: Fourier Transform

```
x_0[n] = [ 3, -1,  0,  0,  0]

h[n] = [ 1, -1,  1,  0,  0]

y_0[n] = IDFT(DFT(x_0[n]) . DFT(h[n]))

X_0[n] = DFT(x_0[n]) = [ 2, 2.691 +
0.951i, 3.809 + 0.588i, 3.809 - 0.588i,
2.691 - 0.951i]

H[n] = DFT(h[n]) = [ 1, -0.118 + 0.363i,
2.118 + 1.539i, 2.118 - 1.539i, -0.118 -
0.363i]

H[n] . X_0[n] = [ 2, -0.663 + 0.865i,
7.163 + 7.106i, 7.163 - 7.106i, -0.663 -
0.865i]

y_0[n] = [ 3, -4,  4, -1,  0]
```

Computing y_0[n] Using Method 2: Standard Convolution

```
x_0[n] = [ 3, -1,  0,  0,  0]

h[n] = [ 1, -1,  1,  0,  0]

          |  1  0  0  1 -1 |   |  3 |   |  3 |
          | -1  1  0  0  1 |   | -1 |   | -4 |
y_0[n] =  |  1 -1  1  0  0 | . |  0 | = |  4 |
          |  0  1 -1  1  0 |   |  0 |   | -1 |
          |  0  0  1 -1  1 |   |  0 |   |  0 |
```

Signals, Linear Filtering methods based on the DFT

Overlap-Add Method

PES UNIVERSITY ONLINE

Computing y_1[n] Using Method 1: Fourier Transform

```
x_1[n] = [ 3,  2,  0,  0,  0]

h[n] = [ 1, -1,  1,  0,  0]

y_1[n] = IDFT(DFT(x_1[n]) . DFT(h[n]))

X_1[n] = DFT(x_1[n]) = [ 5, 3.618 - 1.902i, 1.382 - 1.176i, 1.382 + 1.176i, 3.618 + 1.902i]

H[n] = DFT(h[n]) = [ 1, -0.118 + 0.363i, 2.118 + 1.539i, 2.118 - 1.539i, -0.118 - 0.363i]

H[n] . X_1[n] = [ 5, 0.264 + 1.539i, 4.736 - 0.363i, 4.736 + 0.363i, 0.264 - 1.539i]

y_1[n] = [ 3, -1,  1,  2,  0]
```

Computing y_1[n] Using Method 2: Standard Convolution

```
x_1[n] = [ 3,  2,  0,  0,  0]

h[n] = [ 1, -1,  1,  0,  0]

          |  1  0  0  1 -1 |   | 3 |   |  3 |
          | -1  1  0  0  1 |   | 2 |   | -1 |
y_1[n] =  |  1 -1  1  0  0 | . | 0 | = |  1 |
          |  0  1 -1  1  0 |   | 0 |   |  2 |
          |  0  0  1 -1  1 |   | 0 |   |  0 |
```

Signals, Linear Filtering methods based on the DFT

Overlap-Add Method

Computing y_2[n] Using Method 1: Fourier Transform

```
x_2[n] = [ 1,  2,  1,  0,  0]

h[n] = [ 1, -1,  1,  0,  0]

y_2[n] = IDFT(DFT(x_2[n]) . DFT(h[n]))

X_2[n] = DFT(x_2[n]) = [ 4, 0.809 - 2.49i, -0.309 - 0.225i, -0.309 + 0.225i, 0.809 + 2.49i]

H[n] = DFT(h[n]) = [ 1, -0.118 + 0.363i, 2.118 + 1.539i, 2.118 - 1.539i, -0.118 - 0.363i]

H[n] . X_2[n] = [ 4, 0.809 + 0.588i, -0.309 - 0.951i, -0.309 + 0.951i, 0.809 - 0.588i]

y_2[n] = [ 1,  1,  0,  1,  1]
```

Computing y_2[n] Using Method 2: Standard Convolution

```
x_2[n] = [ 1,  2,  1,  0,  0]

h[n] = [ 1, -1,  1,  0,  0]

         |  1  0  0  1 -1 |   | 1 |   | 1 |
         | -1  1  0  0  1 |   | 2 |   | 1 |
y_2[n] = |  1 -1  1  0  0 | . | 1 | = | 0 |
         |  0  1 -1  1  0 |   | 0 |   | 1 |
         |  0  0  1 -1  1 |   | 0 |   | 1 |
```

Overlap-Add Method

n	0	1	2	3	4	5	6	7	8	9	10
y_0[n]	3	-4	4	-1	0	0	0	0	0	0	0
y_1[n]	0	0	0	3	-1	1	2	0	0	0	0
y_2[n]	0	0	0	0	0	0	1	1	0	1	1
y[n]	3	-4	4	2	-1	1	3	1	0	1	1

THANK YOU

Ms. Ashwini

Department of Electronics and Communication

ashwinib@pes.edu

+91 80 6666 3333
Ext 741