DFT Properties

- Like the DTFT, the DFT also satisfies a number of properties that are useful in signal processing applications
- Some of these properties are essentially identical to those of the DTFT, while some others are somewhat different
- A summary of the DFT properties are given in tables in the following slides

Table 5.1: DFT Properties: Symmetry Relations

Length-N Sequence	N-point DFT
$x[n]$	$X[k]$
$x^*[n]$	$X^*[\langle -k\rangle_N]$
$x^*[\langle -n\rangle_N]$	$X^*[k]$
$\text{Re}\{x[n]\}$	$X_{\text{pcs}}[k] = \frac{1}{2}\{X[\langle k\rangle_N] + X^*[\langle -k\rangle_N]\}$
$j\,\text{Im}\{x[n]\}$	$X_{\text{pca}}[k] = \frac{1}{2}\{X[\langle k\rangle_N] - X^*[\langle -k\rangle_N]\}$
$x_{\text{pcs}}[n]$	$\text{Re}\{X[k]\}$
$x_{\text{pca}}[n]$	$j\,\text{Im}\{X[k]\}$

Note: $x_{\text{pcs}}[n]$ and $x_{\text{pca}}[n]$ are the periodic conjugate-symmetric and periodic conjugate-antisymmetric parts of $x[n]$, respectively. Likewise, $X_{\text{pcs}}[k]$ and $X_{\text{pca}}[k]$ are the periodic conjugate-symmetric and periodic conjugate-antisymmetric parts of $X[k]$, respectively.

$x[n]$ is a complex sequence

Table 5.2: DFT Properties: Symmetry Relations

Length-N Sequence	N-point DFT
$x[n]$	$X[k] = \text{Re}\{X[k]\} + j\,\text{Im}\{X[k]\}$
$x_{pe}[n]$	$\text{Re}\{X[k]\}$
$x_{po}[n]$	$j\,\text{Im}\{X[k]\}$
Symmetry relations	$X[k] = X^*[\langle -k\rangle_N]$ $\text{Re}\,X[k] = \text{Re}\,X[\langle -k\rangle_N]$ $\text{Im}\,X[k] = -\text{Im}\,X[\langle -k\rangle_N]$ $\lvert X[k]\rvert = \lvert X[\langle -k\rangle_N]\rvert$ $\arg X[k] = -\arg X[\langle -k\rangle_N]$

Note: $x_{pe}[n]$ and $x_{po}[n]$ are the periodic even and periodic odd parts of $x[n]$, respectively.

$x[n]$ is a real sequence

Table 5.3: DFT Theorems

Theorems	Length-N Sequence	N-point DFT
	$g[n]$ $h[n]$	$G[k]$ $H[k]$
Linearity	$\alpha g[n] + \beta h[n]$	$\alpha G[k] + \beta H[k]$
Circular time-shifting	$g[\langle n - n_o \rangle_N]$	$W_N^{kn_o} G[k]$
Circular frequency-shifting	$W_N^{-k_o n} g[n]$	$G[\langle k - k_o \rangle_N]$
Duality	$G[n]$	$N g[\langle -k \rangle_N]$
N-point circular convolution	$\sum_{m=0}^{N-1} g[m] h[\langle n - m \rangle_N]$	$G[k]H[k]$
Modulation	$g[n]h[n]$	$\frac{1}{N} \sum_{m=0}^{N-1} G[m] H[\langle k - m \rangle_N]$
Parseval's relation	$\sum_{n=0}^{N-1} \lvert x[n] \rvert^2 = \frac{1}{N} \sum_{k=0}^{N-1} \lvert X[k] \rvert^2$	

Operations on Finite-Length Sequences

- Consider the length-N sequence $x[n]$ defined for $0 \leq n \leq N-1$
- Its sample values are equal to zero for $\mathrm{n} < 0$ and $n \geq N$
- A time-reversal operation on $x[n]$ will result in a length-N sequence $x[-n]$ defined for $-(N-1) \leq n \leq 0$

Operations on Finite-Length Sequences

- Likewise, a linear time-shift of $x[n]$ by integer-valued M will result in a length-N sequence $x[n+M]$ no longer defined for $0 \le n \le N-1$
- Similarly, a convolution sum of two length-N sequences defined for $0 \le n \le N-1$ will result in a sequence of length $2N+1$ defined for $0 \le n \le 2N-2$

Circular Operations on Finite-Length Sequences

- Thus we need to define new type of time-reversal and time-shifting operations, and also new type of convolution operation for length-N sequences defined for $0 \le n \le N-1$ so that the resultant length-N sequences are also are in the range $0 \le n \le N-1$

The implicit periodicity of sequences manipulated with the DFT requires to substitute the zero-padded operation:

[1 2 3 4] --> [... 0 0 0 1 2 3 4 0 0 0 ...]

with a **circular** (i.e. periodic) extension of the sequence:

[1 2 3 4] --> [... 1 2 3 4 1 2 3 4 1 2 3 4 ...]

Modulo Operation

- The time-reversal operation on a finite-length sequence is obtained using the modulo operation
- Let $0,1,\ldots,N-1$ be a set of N positive integers and let m be any integer
- The integer r obtained by evaluating

  m modulo N

  is called the residue

Modulo Operation

- The residue r is an integer with a value between 0 and $N-1$
- The modulo operation is denoted by the notation $\langle m \rangle_N = m$ modulo N
- If we let $r = \langle m \rangle_N$ then $r = m + \ell N$ where ℓ is a positive or negative integer necessary to make $m + \ell N$ an integer between 0 and $N-1$

If m>0, it is the remainder of the integer division of m by N

Modulo Operation

- **Example** – For $N = 7$ and $m = 25$, we have

$$r = 25 + 7\ell = 25 - 7 \times 3 = 4$$

Thus, $\langle 25 \rangle_7 = 4$

- **Example** – For $N = 7$ and $m = -15$, we get

$$r = -15 + 7\ell = -15 + 7 \times 3 = 6$$

Thus, $\langle -15 \rangle_7 = 6$

Circular Time-Reversal Operation

- The circular time-reversal version $\{y[n]\}$ of a length-N sequence $\{x[n]\}$ defined for $0 \le n \le N-1$ is given by $\{y[n]\} = \{x[\langle -n \rangle_N]\}$
- **Example** – Consider

$$\{x[n]\} = \{x[0],\ x[1],\ x[2],\ x[3],\ x[4]\}$$

Its circular time-reversed version is given by $\{y[n]\} = \{x[\langle -n \rangle_5]\}$

$$= \{x[0],\ x[4],\ x[3],\ x[2],\ x[1]\}$$

Circular Shift of a Sequence

- The time shifting operation for a finite-length sequence, called circular shift operation, is defined using the modulo operation
- Let $x[n]$ be a length-N sequence defined for $0 \leq n \leq N-1$
- Its circularly shifted version $x_c[n]$, shifted n_o by samples, is given by

$$x_c[n] = x[\langle n - n_o \rangle_N]$$

Circular Shift of a Sequence

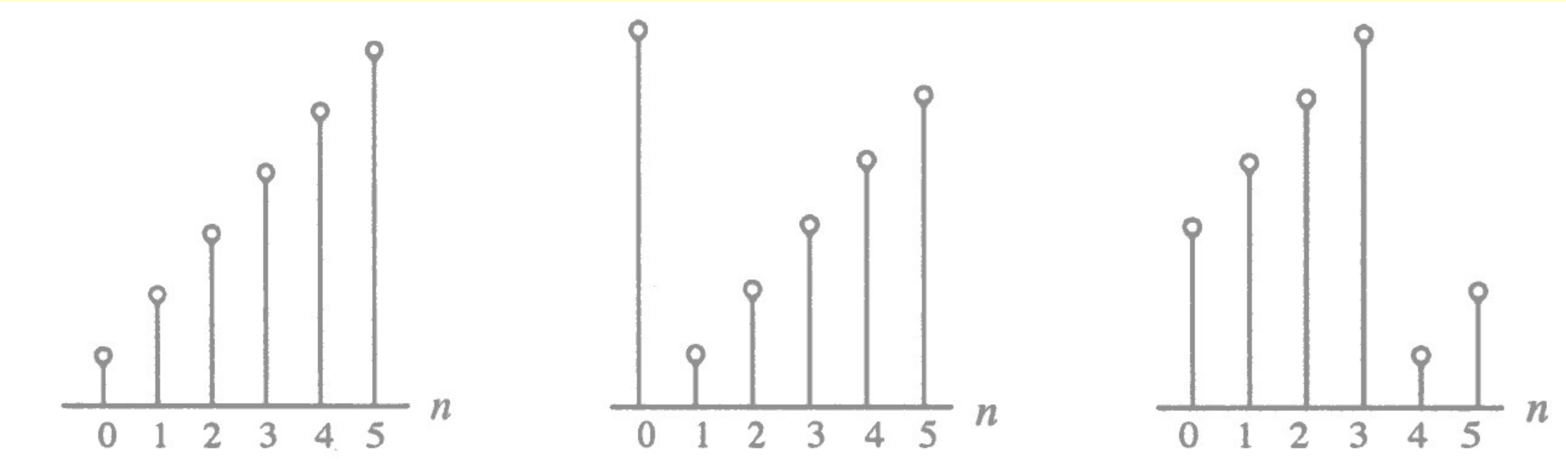
$$x_c[n] = x[\langle n - n_o \rangle_N]$$

- $x_c[n]$ is also a length-N sequence defined for $0 \le n \le N-1$
- For $n_o > 0$ (right circular shift), the above equation implies

$$x_c[n] = \begin{cases} x[n - n_o], & \text{for } n_o \le n \le N-1 \\ x[N - n_o + n], & \text{for } 0 \le n < n_o \end{cases}$$

Circular Shift of a Sequence

- Illustration of the concept of a circular shift



$x[n]$

$x[\langle n-1\rangle_6]$ $= x[\langle n+5\rangle_6]$

$x[\langle n-4\rangle_6]$ $= x[\langle n+2\rangle_6]$

right

left

Circular Shift of a Sequence

- As can be seen from the previous figure, a right circular shift by n_o is equivalent to a left circular shift by $N - n_o$ sample periods
- A circular shift by an integer number n_o greater than N is equivalent to a circular shift by $\langle n_o \rangle_N$

Circular Convolution

- This operation is analogous to linear convolution, but with a subtle difference
- Consider two length-N sequences, $g[n]$ and $h[n]$, respectively
- Their linear convolution results in a length-$(2N-1)$ sequence $y_L[n]$ given by

$$y_L[n] = \sum_{m=0}^{N-1} g[m]h[n-m], \quad 0 \leq n \leq 2N-2$$

Circular Convolution

- In computing $y_L[n]$ we have assumed that both length-N sequences have been zero-padded to extend their lengths to $2N-1$
- The longer form of $y_L[n]$ results from the time-reversal of the sequence $h[n]$ and its linear shift to the right
- The first nonzero value of $y_L[n]$ is $y_L[0] = g[0]h[0]$, and the last nonzero value is $y_L[2N-2] = g[N-1]h[N-1]$

Circular Convolution

- To develop a convolution-like operation resulting in a length-N sequence $y_C[n]$, we need to define a circular time-reversal, and then apply a circular time-shift
- Resulting operation, called a **circular convolution**, is defined by

$$y_C[n] = \sum_{m=0}^{N-1} g[m]h[\langle n-m \rangle_N], \quad 0 \le n \le N-1$$

Circular Convolution

- Since the operation defined involves two length-N sequences, it is often referred to as an N-point circular convolution, denoted as

$$y[n] = g[n] \circledN h[n]$$

- The circular convolution is commutative, i.e.

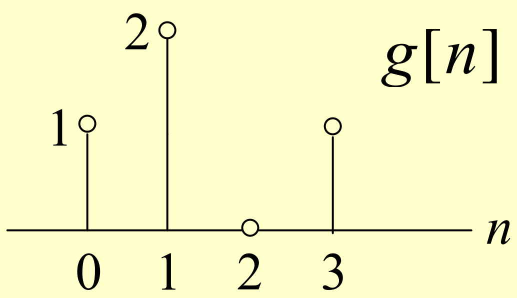
$$g[n] \circledN h[n] = h[n] \circledN g[n]$$

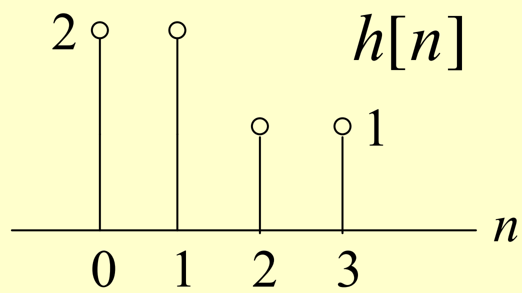
Circular Convolution

- Example - Determine the 4-point circular convolution of the two length-4 sequences:

$$\{g[n]\} = \{\underset{\uparrow}{1} \quad 2 \quad 0 \quad 1\}, \quad \{h[n]\} = \{\underset{\uparrow}{2} \quad 2 \quad 1 \quad 1\}$$

as sketched below





Circular Convolution

- The result is a length-4 sequence $y_C[n]$ given by

$$y_C[n] = g[n] \circledast_4 h[n] = \sum_{m=0}^{3} g[m]h[\langle n-m\rangle_4],$$

$$0 \le n \le 3$$

- From the above we observe

$$y_C[0] = \sum_{m=0}^{3} g[m]h[\langle -m\rangle_4]$$

$$= g[0]h[0] + g[1]h[3] + g[2]h[2] + g[3]h[1]$$

$$= (1\times 2) + (2\times 1) + (0\times 1) + (1\times 2) = 6$$

Circular Convolution

- Likewise $y_C[1] = \sum_{m=0}^{3} g[m]h[\langle 1-m \rangle_4]$

$$= g[0]h[1] + g[1]h[0] + g[2]h[3] + g[3]h[2]$$

$$= (1\times 2) + (2\times 2) + (0\times 1) + (1\times 1) = 7$$

$$y_C[2] = \sum_{m=0}^{3} g[m]h[\langle 2-m \rangle_4]$$

$$= g[0]h[2] + g[1]h[1] + g[2]h[0] + g[3]h[3]$$

$$= (1\times 1) + (2\times 2) + (0\times 2) + (1\times 1) = 6$$

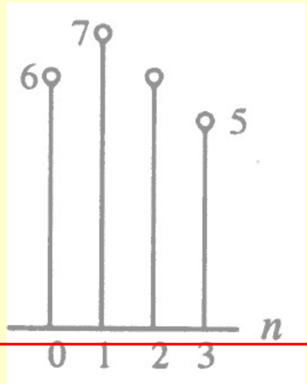
Circular Convolution

and

$$y_C[3] = \sum_{m=0}^{3} g[m]h[\langle 3-m\rangle_4]$$

$$= g[0]h[3] + g[1]h[2] + g[2]h[1] + g[3]h[0]$$

$$= (1\times 1) + (2\times 1) + (0\times 2) + (1\times 2) = 5$$

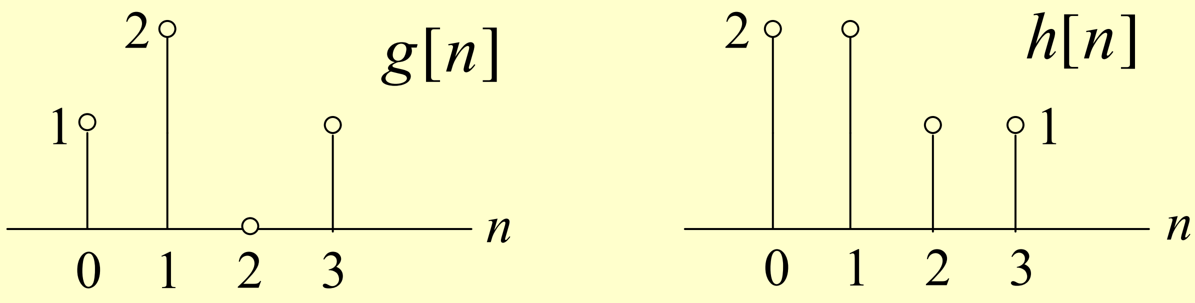


$y_C[n]$

- The circular convolution can also be computed using a DFT-based approach as indicated in Table 5.3

Circular Convolution

- Example - Consider the two length-4 sequences repeated below for convenience:



- The 4-point DFT $G[k]$ of $g[n]$ is given by

$$G[k] = g[0] + g[1]e^{-j2\pi k/4} + g[2]e^{-j4\pi k/4} + g[3]e^{-j6\pi k/4}$$
$$= 1 + 2e^{-j\pi k/2} + e^{-j3\pi k/2}, \quad 0 \le k \le 3$$

Circular Convolution

- Therefore $G[0] = 1 + 2 + 1 = 4,$

$$G[1] = 1 - j2 + j = 1 - j,$$
$$G[2] = 1 - 2 - 1 = -2,$$
$$G[3] = 1 + j2 - j = 1 + j$$

- Likewise,

$$H[k] = h[0] + h[1]e^{-j2\pi k/4}$$
$$+ h[2]e^{-j4\pi k/4} + h[3]e^{-j6\pi k/4}$$
$$= 2 + 2e^{-j\pi k/2} + e^{-j\pi k} + e^{-j3\pi k/2}, \quad 0 \le k \le 3$$

Circular Convolution

- Hence, $H[0] = 2 + 2 + 1 + 1 = 6,$
  $H[1] = 2 - j2 - 1 + j = 1 - j,$
  $H[2] = 2 - 2 + 1 - 1 = 0,$
  $H[3] = 2 + j2 - 1 - j = 1 + j$
- The two 4-point DFTs can also be computed using the matrix relation given earlier

Circular Convolution

$$\begin{bmatrix} G[0] \\ G[1] \\ G[2] \\ G[3] \end{bmatrix} = \mathbf{D}_4 \begin{bmatrix} g[0] \\ g[1] \\ g[2] \\ g[3] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 1-j \\ -2 \\ 1+j \end{bmatrix}$$

$$\begin{bmatrix} H[0] \\ H[1] \\ H[2] \\ H[3] \end{bmatrix} = \mathbf{D}_4 \begin{bmatrix} h[0] \\ h[1] \\ h[2] \\ h[3] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 2 \\ 2 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 6 \\ 1-j \\ 0 \\ 1+j \end{bmatrix}$$

$\mathbf{D}_4$ is the 4-point DFT matrix

Circular Convolution

- If $Y_C[k]$ denotes the 4-point DFT of $y_C[n]$ then from Table 3.5 we observe

$$Y_C[k] = G[k]H[k], \;\; 0 \le k \le 3$$

- Thus

$$\begin{bmatrix} Y_C[0] \\ Y_C[1] \\ Y_C[2] \\ Y_C[3] \end{bmatrix} = \begin{bmatrix} G[0]H[0] \\ G[1]H[1] \\ G[2]H[2] \\ G[3]H[3] \end{bmatrix} = \begin{bmatrix} 24 \\ -j2 \\ 0 \\ j2 \end{bmatrix}$$

Circular Convolution

- A 4-point IDFT of $Y_C[k]$ yields

$$\begin{bmatrix} y_C[0] \\ y_C[1] \\ y_C[2] \\ y_C[3] \end{bmatrix} = \frac{1}{4}\mathbf{D}_4^* \begin{bmatrix} Y_C[0] \\ Y_C[1] \\ Y_C[2] \\ Y_C[3] \end{bmatrix}$$

$$= \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix} \begin{bmatrix} 24 \\ -j2 \\ 0 \\ j2 \end{bmatrix} = \begin{bmatrix} 6 \\ 7 \\ 6 \\ 5 \end{bmatrix}$$

Circular Convolution

- Example - Now let us extend the two length-4 sequences to length 7 by appending each with three zero-valued samples, i.e.

$$g_e[n] = \begin{cases} g[n], & 0 \le n \le 3 \\ 0, & 4 \le n \le 6 \end{cases}$$

$$h_e[n] = \begin{cases} h[n], & 0 \le n \le 3 \\ 0, & 4 \le n \le 6 \end{cases}$$

Circular Convolution

- We next determine the 7-point circular convolution of $g_e[n]$ and $h_e[n]$:

$$y[n] = \sum_{m=0}^{6} g_e[m] h_e[\langle n-m \rangle_7], \quad 0 \le n \le 6$$

- From the above $y[0] = g_e[0]h_e[0] + g_e[1]h_e[6]$

$$+ g_e[2]h_e[5] + g_e[3]h_e[4] + g_e[4]h_e[3]$$

$$+ g_e[5]h_e[2] + g_e[6]h_e[1] = \mathrm{g}[0]\mathrm{h}[0] = 1 \times 2 = 2$$

Circular Convolution

- Continuing the process we arrive at

$$y[1] = g[0]h[1] + g[1]h[0] = (1\times 2) + (2\times 2) = 6,$$

$$y[2] = g[0]h[2] + g[1]h[1] + g[2]h[0]$$

$$= (1\times 1) + (2\times 2) + (0\times 2) = 5,$$

$$y[3] = g[0]h[3] + g[1]h[2] + g[2]h[1] + g[3]h[0]$$

$$= (1\times 1) + (2\times 1) + (0\times 2) + (1\times 2) = 5,$$

$$y[4] = g[1]h[3] + g[2]h[2] + g[3]h[1]$$

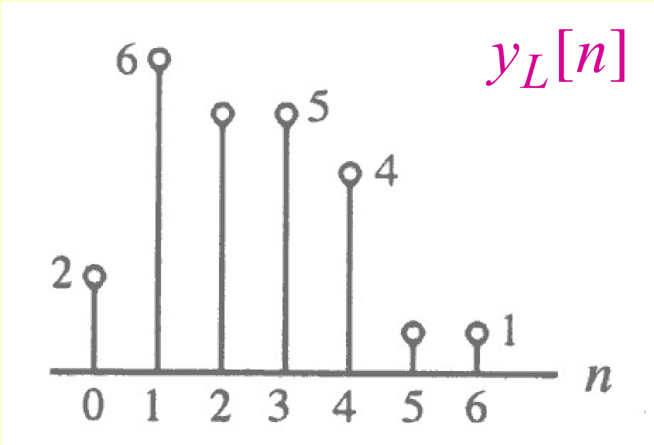
$$= (2\times 1) + (0\times 1) + (1\times 2) = 4,$$

Circular Convolution

$$y[5] = g[2]h[3] + g[3]h[2] = (0\times 1) + (1\times 1) = 1,$$

$$y[6] = g[3]h[3] = (1\times 1) = 1$$

- It can be seen from the above that $y[n]$ is precisely the sequence $y_L[n]$ obtained by a linear convolution of $g[n]$ and $h[n]$



Linear Convolution Using the DFT

- Linear convolution is a key operation in many signal processing applications
- Since a DFT can be efficiently implemented using FFT algorithms, it is of interest to develop methods for the implementation of linear convolution using the DFT

Linear Convolution of Two Finite-Length Sequences

- Let $g[n]$ and $h[n]$ be two finite-length sequences of length N and M, respectively
- Denote $L = N + M - 1$
- Define two length-L sequences

$$g_e[n] = \begin{cases} g[n], & 0 \le n \le N-1 \\ 0, & N \le n \le L-1 \end{cases}$$

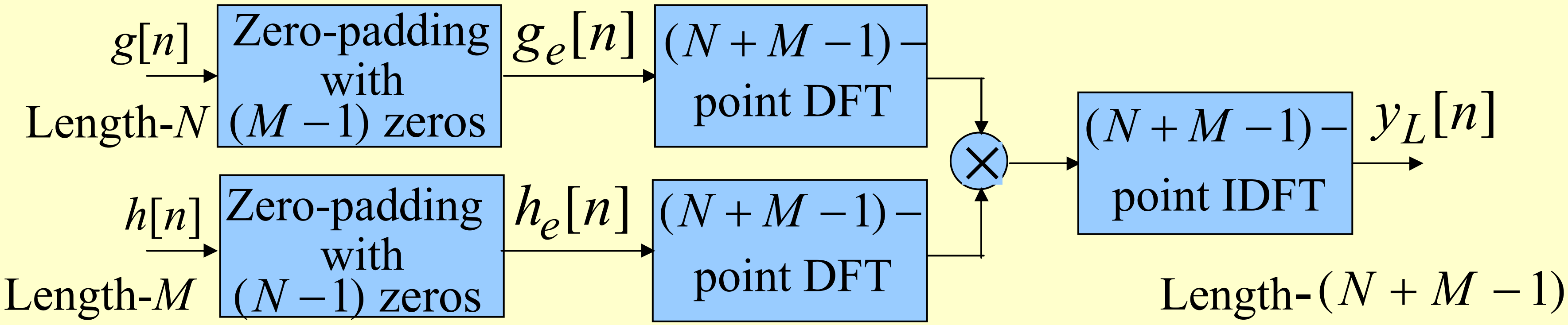
$$h_e[n] = \begin{cases} h[n], & 0 \le n \le M-1 \\ 0, & M \le n \le L-1 \end{cases}$$

Linear Convolution of Two Finite-Length Sequences

- Then

$$y_L[n] = g[n] \circledast h[n] = y_C[n] = g_e[n] \, \textcircled{L} \, h_e[n]$$

- The corresponding implementation scheme is illustrated below



Linear Convolution of a Finite-Length Sequence with an Infinite-Length Sequence

- We next consider the DFT-based implementation of

$$y[n] = \sum_{\ell=0}^{M-1} h[\ell]x[n-\ell] = h[n] \circledast x[n]$$

where $h[n]$ is a finite-length sequence of length M and $x[n]$ is an infinite length (or a finite length sequence of length much greater than M)

Overlap-Add Method

- We first segment $x[n]$, assumed to be a causal sequence here without any loss of generality, into a set of contiguous finite-length subsequences $x_m[n]$ of length N each:

$$x[n] = \sum_{m=0}^{\infty} x_m[n - mN]$$

where

$$x_m[n] = \begin{cases} x[n + mN], & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

Overlap-Add Method

- Thus we can write

$$y[n] = h[n] \circledast x[n] = \sum_{m=0}^{\infty} y_m[n - mN]$$

where

$$y_m[n] = h[n] \circledast x_m[n]$$

- Since $h[n]$ is of length M and $x_m[n]$ is of length N, the linear convolution $h[n] \circledast x_m[n]$ is of length $N + M - 1$

Overlap-Add Method

- As a result, the desired linear convolution $y[n] = h[n] \circledast x[n]$ has been broken up into a sum of infinite number of short-length linear convolutions of length $N + M - 1$ each: $y_m[n] = x_m[n] \circledast h[n]$
- Each of these short convolutions can be implemented using the DFT-based method discussed earlier, where now the DFTs (and the IDFT) are computed on the basis of $(N + M - 1)$ points

Overlap-Add Method

- There is one more subtlety to take care of before we can implement

$$y[n] = \sum_{m=0}^{\infty} y_m[n - mN]$$

using the DFT-based approach

- Now the first convolution in the above sum, $y_0[n] = h[n] \circledast x_0[n]$, is of length $N + M - 1$ and is defined for $0 \le n \le N + M - 2$

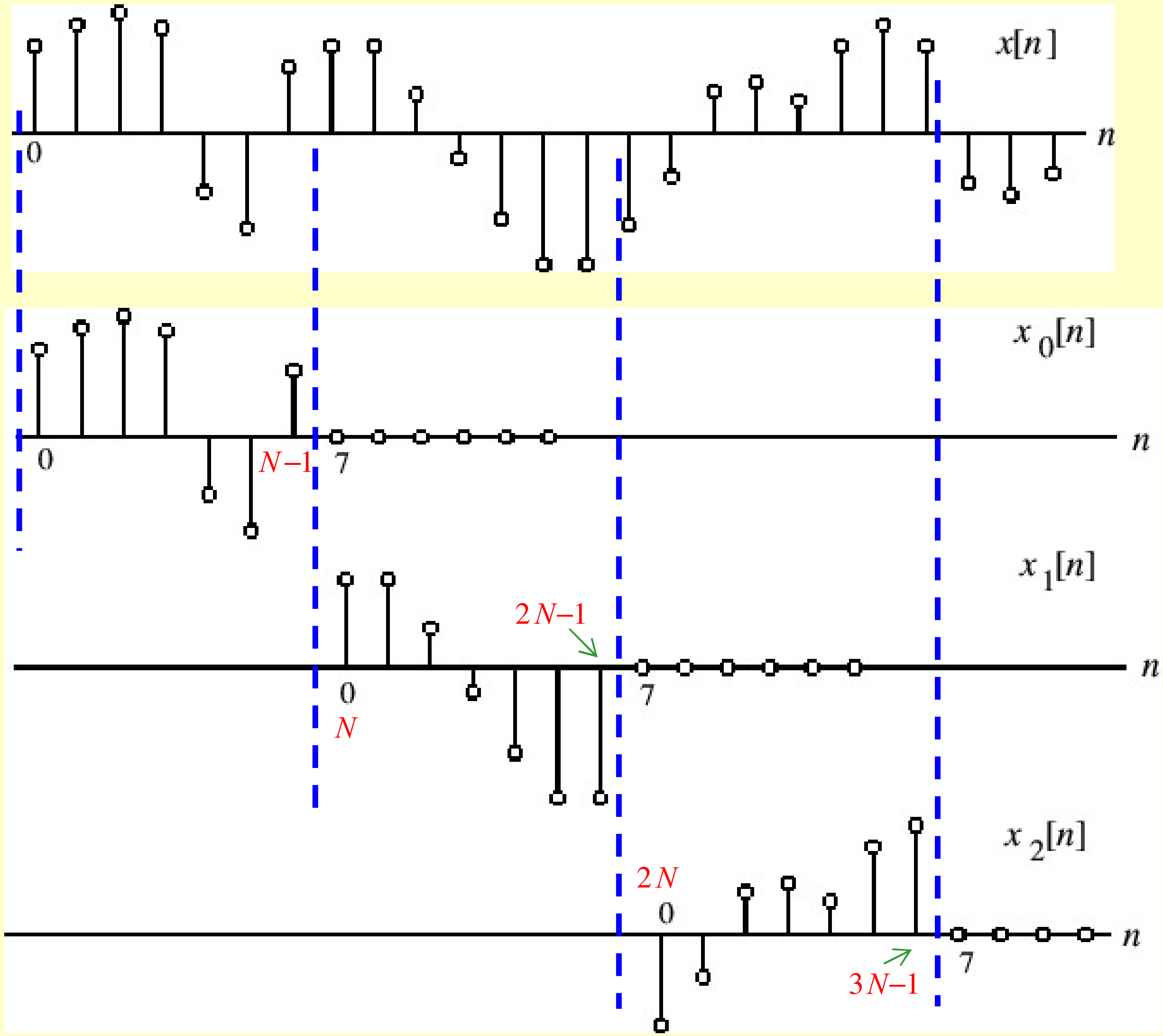
Overlap-Add Method

- The second short convolution $y_1[n] = h[n] \circledast x_1[n]$, is also of length $N+M-1$ but is defined for $N \le n \le 2N+M-2$
- ⇒ There is an overlap of $M-1$ samples between these two short linear convolutions
- Likewise, the third short convolution $y_2[n] = h[n] \circledast x_2[n]$, is also of length $N+M-1$ but is defined for $2N \le n \le 3N+M-2$

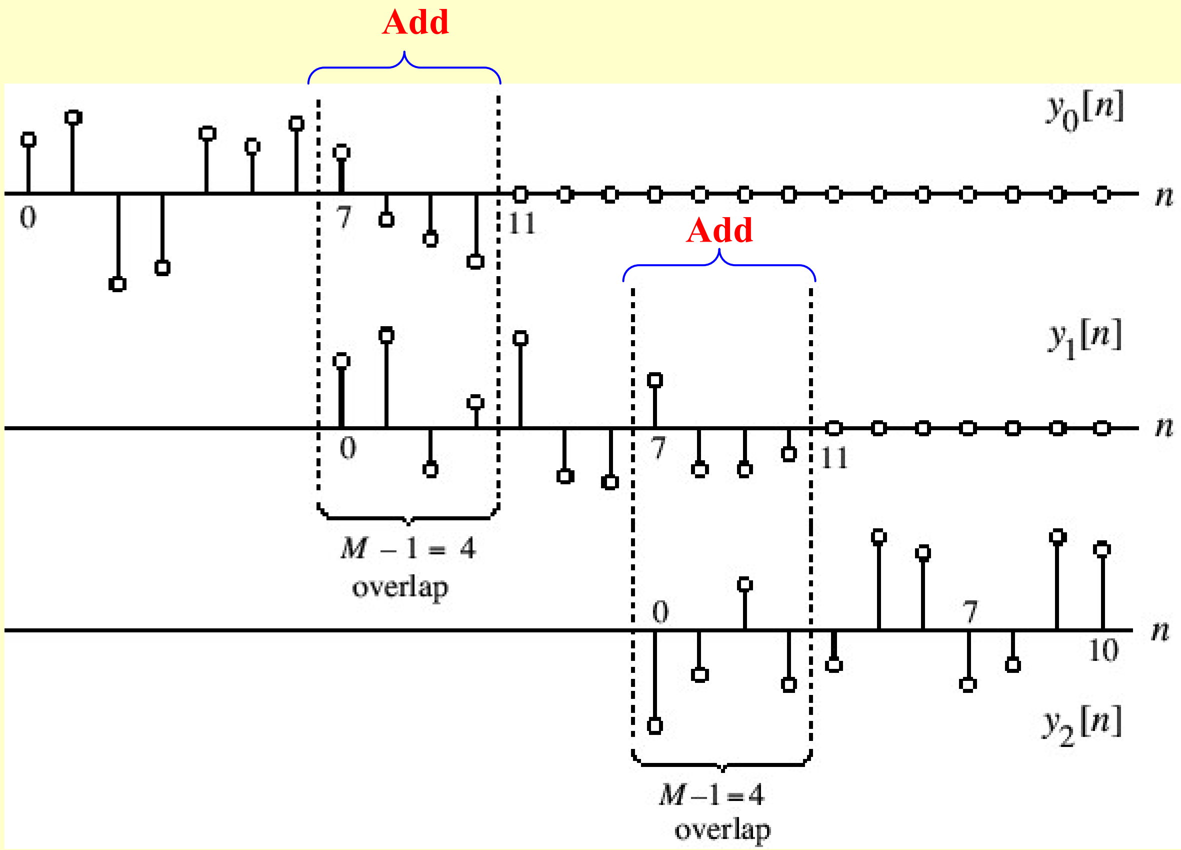
Overlap-Add Method

- Thus there is an overlap of $M-1$ samples between $h[n] \circledast x_1[n]$ and $h[n] \circledast x_2[n]$
- In general, there will be an overlap of $M-1$ samples between the samples of the short convolutions $h[n] \circledast x_{r-1}[n]$ and $h[n] \circledast x_r[n]$ for $(r-1)N \leq n \leq rN + M - 2$
- This process is illustrated in the figure on the next slide for $M = 5$ and $N = 7$

Overlap-Add Method



Overlap-Add Method



Overlap-Add Method

- Therefore, $y[n]$ obtained by a linear convolution of $x[n]$ and $h[n]$ is given by

$$
\begin{aligned}
y[n] &= y_0[n], & 0 \le n \le 6 \\
y[n] &= y_0[n] + y_1[n-7], & 7 \le n \le 10 \\
y[n] &= y_1[n-7], & 11 \le n \le 13 \\
y[n] &= y_1[n-7] + y_2[n-14], & 14 \le n \le 17 \\
y[n] &= y_2[n-14], & 18 \le n \le 20 \\
&\vdots
\end{aligned}
$$

Overlap-Add Method

- The above procedure is called the **overlap-add method** since the results of the short linear convolutions overlap and the overlapped portions are added to get the correct final result
- The function `fftfilt` can be used to implement the above method