Chapter 5 Finite-Length Discrete Transforms

- 5.1 Orthogonal Transforms
- 5.2 DFT
- 5.3 Relation Between the DTFT and the DFT
- 5.4 Circular Convolution
- 5.5Classifications of Finite-Length Sequences
- 5.6 DFT Symmetry Relations
- 5.7 DFT Properties
- 5.8 Fourier-Domain Filtering
- 5.9 Computation of the DFT of Real Sequences
- 5.10 Linear Convolution Using DFT

5.1 Orthogonal Transforms

Length-N time-domain sequence x[n]. The coefficient of its N-point orthogonal transform is X[k]

$$X[k]=\sum_{n=0}^{N-1}x[n]\Psi^{*}[k,n],\ 0\le k\le N-1$$ **analysis equation**

$$x[n]=\frac{1}{N}\sum_{k=0}^{N-1}X[k]\Psi[k,n],\ 0\le k\le N-1$$ **synthesis equation**

Set basis sequences $\Psi[k,n]=e^{j2\pi kn/N},0\le k\le N-1$ **orthogonal basis sequences, length-N sequences in both domains**

$\Psi[k,n]$ **orthogonal to each other**

$$\frac{1}{N}\sum_{n=0}^{N-1}\Psi[k,n]\Psi^{*}[l,n]=\begin{cases}1, & l=k\\ 0, & l\neq k\end{cases}$$

Parseval's theorem (orthogonality of basis sequence ensure the energy preservation

$$\sum_{n=0}^{N-1}|x[n]|^{2}=\frac{1}{N}\sum_{k=0}^{N-1}|X[k]|^{2}$$

$$X[k]=\sum_{n=0}^{N-1}x[n]e^{-j2\pi kn/N}\qquad 0\le k\le N-1$$

$$x[n]=\frac{1}{N}\sum_{n=0}^{N-1}X[k]e^{j2\pi kn/N}\qquad 0\le k\le N-1$$

§ 5.2 Discrete Fourier Transform (DFT)

1）Finite length sequence：

$$x[n], n = 0 \sim (M-1)$$ ——**length-M sequence**

N 点 DFT：

$M \le N$

$$\boxed{\begin{cases} X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}, & k = 0 \sim (N-1) \\ x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn}, & n = 0 \sim (N-1) \end{cases}}$$

2）数字谐波信号与数字基频 → 旋转因子Twiddle factor

$$W_N^{kn} = \exp\left(-\mathrm{j}\frac{2\pi k}{N} n\right) = \exp(-\mathrm{j}k\omega_0 n) \qquad \omega_0 = \frac{2\pi}{N}$$

数字谐波信号　　　　数字基频

§ 5.2.1 Discrete Fourier Transform (DFT)

- The **inverse discrete Fourier transform (IDFT)** is given by

$$x[n]=\frac{1}{N}\sum_{k=0}^{N-1}X[k]W_N^{-kn},\quad 0\le n\le N-1$$

To verify the above expression we multiply both sides of the above equation by W_N^{ln} and sum the result from n = 0 to n=N-1, resulting in

$$\sum_{n=0}^{N-1}x[n]W_N^{\ell n}=\sum_{n=0}^{N-1}\left(\frac{1}{N}\sum_{k=0}^{N-1}X[k]W_N^{-kn}\right)W_N^{\ell n}$$

$$=\frac{1}{N}\sum_{n=0}^{N-1}\sum_{k=0}^{N-1}X[k]W_N^{-(k-\ell)n}$$

$$=\frac{1}{N}\sum_{k=0}^{N-1}\sum_{n=0}^{N-1}X[k]W_N^{-(k-\ell)n}$$

§ 5.2.1 Discrete Fourier Transform (DFT)

$$\sum_{n=0}^{N-1} x[n] W_N^{\ell n} = \frac{1}{N} \sum_{k=0}^{N-1} \sum_{n=0}^{N-1} X[k] W_N^{-(k-\ell)n}$$

Making use of the identity

$$\sum_{n=0}^{N-1} W_N^{-(k-\ell)n} = \begin{cases} N, & \text{for } k-\ell = rN, \quad r \text{ an integer} \\ 0, & \text{otherwise} \end{cases}$$

Hence

$$\sum_{n=0}^{N-1} x[n] W_N^{\ell n} = X[\ell]$$

- **<u>Example5.1</u> - Consider the length-*N* sequence**

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

Its *N*-point DFT is given by

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} = x[0] W_N^0 = 1 \qquad 0 \le k \le N-1$$

- **<u>Example</u> - Consider the length-*N* sequence**

$$y[n] = \begin{cases} 1, & n = m \\ 0, & 0 \le n \le m-1,\ m+1 \le n \le N-1 \end{cases}$$

Its *N*-point DFT is given by

$$Y[k] = \sum_{n=0}^{N-1} y[n] W_N^{kn} = y[m] W_N^{km} = W_N^{km} \qquad 0 \le k \le N-1$$

<u>Example5.2</u> DFT of length-*N* sequence defined for 0 ≤n ≤N-1

$$g[n] = \cos(2\pi rn/N), \quad 0 \le r \le N-1$$

$$g[n] = \tfrac{1}{2}\left(e^{j2\pi rn/N} + e^{-j2\pi rn/N}\right)$$
$$= \tfrac{1}{2}\left(W_N^{-rN} + W_N^{rN}\right)$$

The *N*-point DFT of *g*[*n*] is thus given by

$$G[k] = \sum_{n=0}^{N-1} g[n] W_N^{kn} = \frac{1}{2}\left(\sum_{n=0}^{N-1} W_N^{-(r-k)n} + \sum_{n=0}^{N-1} W_N^{(r+k)n}\right), \quad 0 \le k \le N-1$$

as

$$\sum_{n=0}^{N-1} W_N^{-(k-\ell)n} = \begin{cases} N, & \text{for } k-\ell = rN, \ r \text{ an integer} \\ 0, & \text{otherwis} \end{cases}$$

$$G[k] = \begin{cases} N/2, & \text{for } k = r \\ N/2, & \text{for } k = N-r \\ 0, & \text{otherwise} \end{cases}$$

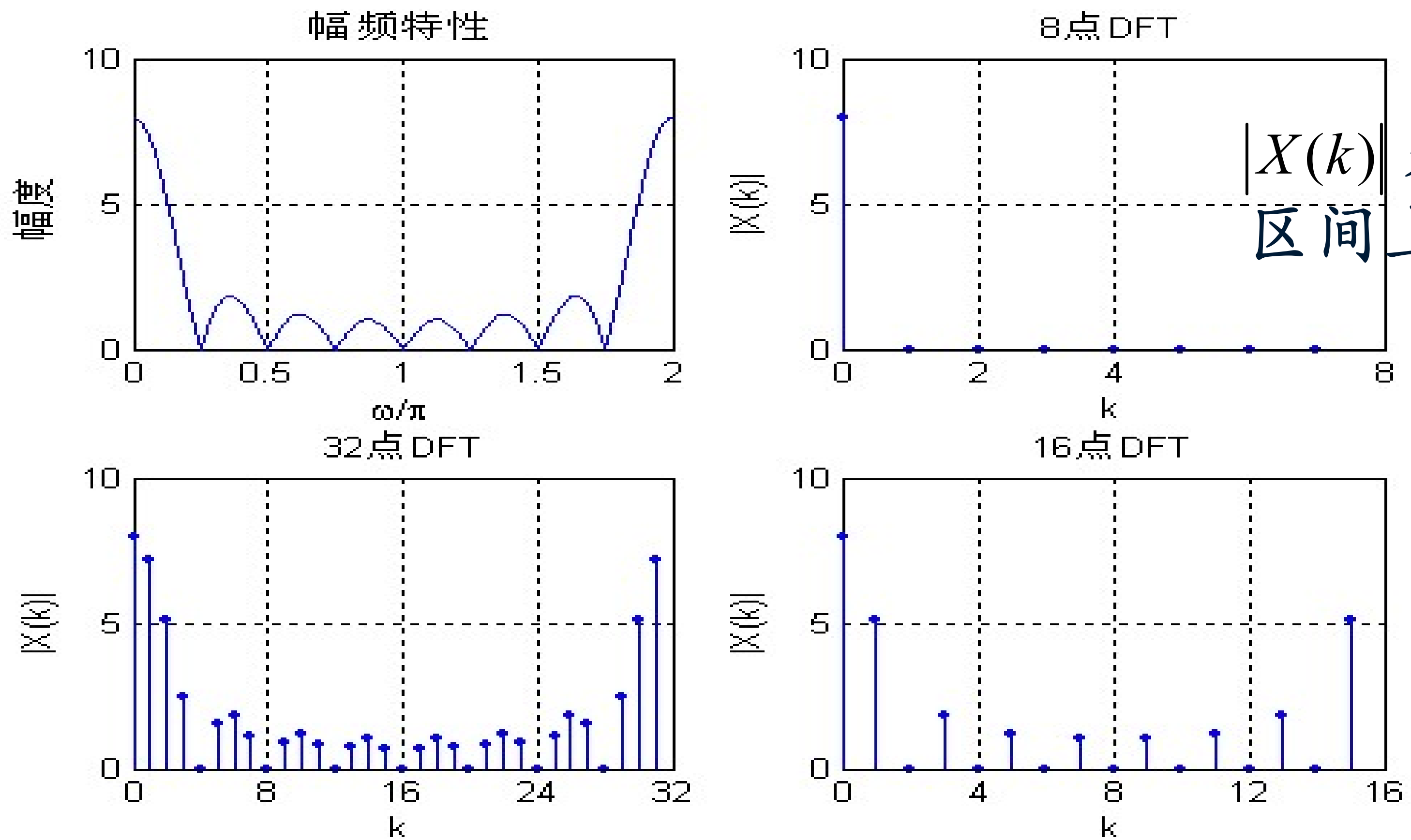
ExampleA： $x(n)=R_8(n)$， 8 point DFT and 16 point DFT of x(n)

8-point DFT:

$$X(k)=\sum_{n=0}^{7}R_8(n)W_8^{kn}=\sum_{n=0}^{7}\mathrm{e}^{-\mathrm{j}\frac{2\pi}{8}kn}=\begin{cases}8, & k=0\\ 0, & k=1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7\end{cases}$$

16-point DFT:

$$X(k)=\sum_{n=0}^{7}W_{16}^{kn}=\frac{1-W_{16}^{k8}}{1-W_{16}^{k}}=\frac{1-\mathrm{e}^{-\mathrm{j}\frac{2\pi}{16}8k}}{1-\mathrm{e}^{-\mathrm{j}\frac{2\pi}{16}k}}$$



$|X(k)|$ 是 $|X(e^{j\omega})|$ 在频率区间上的等间隔采样

5.2.3 Matrix Relations

- The DFT samples defined by

$$X[k]=\sum_{n=0}^{N-1} x[n]W_N^{kn}, \quad 0\le k\le N-1$$

and can be rewritten in matrix form as $\mathbf{X}=\mathbf{D}_N\mathbf{x}$

Where $X=[X[0]\ \ X[1]\ \ \dots\ X[N-1]]^T$

$x=[x[0]\ \ x[1]\ \ \dots\ x[N-1]]^T$

D_N is the N×N DFT matrix given by

$$D_N=\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N^1 & W_N^2 & \cdots & W_N^{(N-1)} \\ 1 & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & W_N^{(N-1)} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)^2} \end{bmatrix}$$

Matrix Relations

- Mathematically, from $\mathbf{X}=\mathbf{D}_N\mathbf{x}$, we can have

$$\mathbf{x} = \mathbf{D}_N^{-1}\mathbf{X}$$

- Likewise, the IDFT relation given by

$$x[n] = \frac{1}{N}\sum_{n=0}^{N-1} X[k] W_N^{-kn}, \quad 0 \le k \le N-1$$

can be expressed in matrix form as $x=D_N^{-1}X$,

where D_N^{-1} is the N×N IDFT matrix.

Matrix Relations

where

$$D_N^{-1} = \frac{1}{N}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N^{-1} & W_N^{-2} & \cdots & W_N^{-(N-1)} \\ 1 & W_N^{-2} & W_N^{-4} & \cdots & W_N^{-2(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & W_N^{-(N-1)} & W_N^{-2(N-1)} & \cdots & W_N^{-(N-1)^2} \end{bmatrix}$$

Note: $\mathbf{D_N^{-1} = D_N^{*}/N}$

5.2.4 DFT Computation Using MATLAB

- The functions to compute the DFT and the IDFT are FFT and IFFT
- These functions make use of FFT algorithms which are computationally highly efficient compared to the direct computation
- Programs 5_2and 5_4(p.194) illustrate the use of these functions

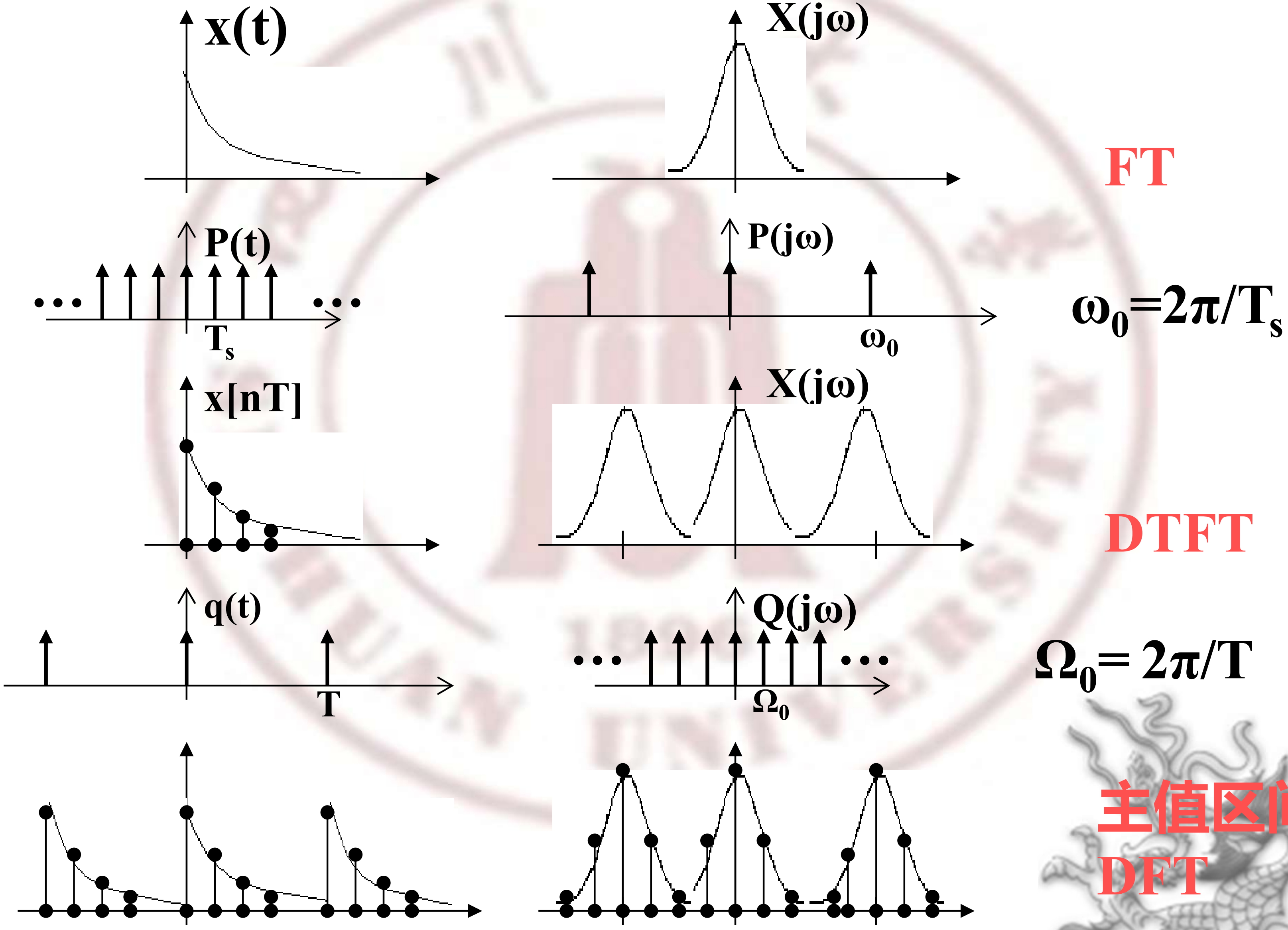
§ 5.3 Relation Between the DTFT and the DFT

Time domain		Frequency domain
Continue aperiodical	$\xleftarrow{\ }$ FT $\xrightarrow{\ }$	Continue aperiodical
Periodical	$\xleftarrow{\ }$ FST $\xrightarrow{\ }$	discrete spectrum
Discrete	$\xleftarrow{\ }$ DTFT $\xrightarrow{\ }$	periodical spectrum
Discrete periodical	$\xleftarrow{\ }$ DFT $\xrightarrow{\ }$	periodical discrete

- **DTFT is discrete in time domain. Its spectrum is periodical, but continue**

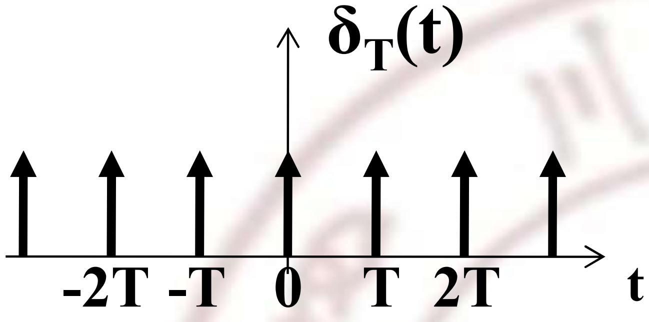
Computer could only process digital signals in both sides, that means the signals in both sides must be both discrete and periodical.

Make a signal discrete and periodical



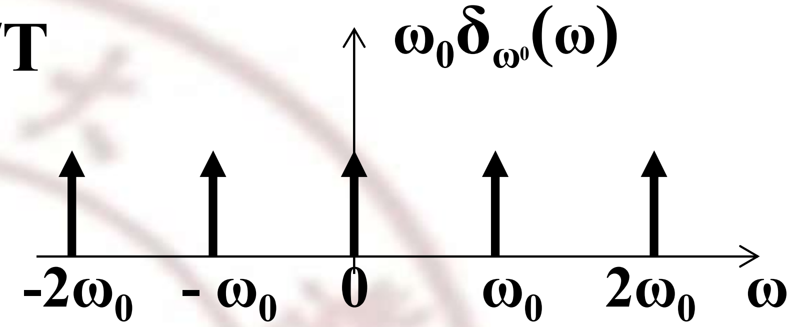
Typical DFT Pair

$\delta_T(t) \longleftrightarrow \omega_0\delta\omega_0(\omega)$



$\omega_0 = 2\pi/T$

$\longleftrightarrow$



- In DFT, the signals in both sides are discrete, so it is the only transform pair which can be processed by computer.

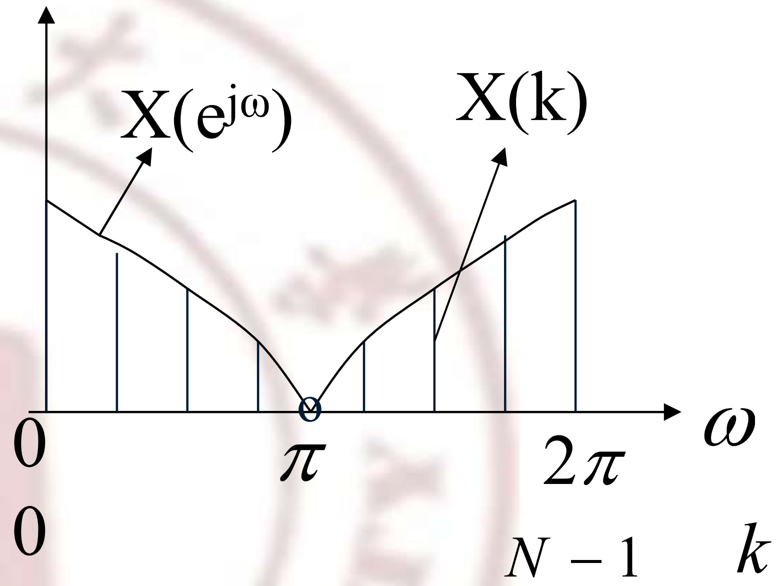
- The signals in both sides are periodical, so the processing could be in one period, which is important because

(1) the number of calculation is limited, which is necessary for computer;

(2) all of the signal information could be kept in one period, which is necessary for accurate processing.

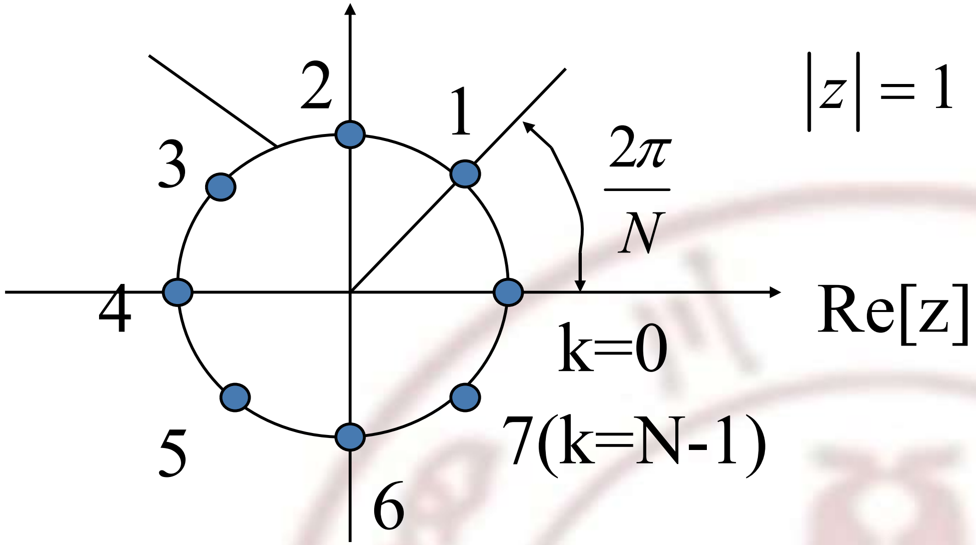
From DTFT to get DFT

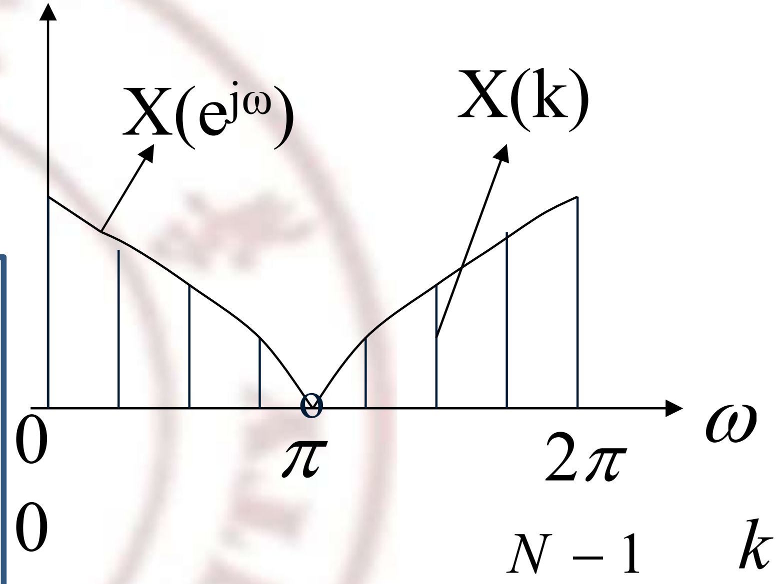
- DTFT: discrete in time domain & continuous in frequency domain



DFT : Sampling the DTFT of sequences to get N frequency points to research.

<u>Definition</u> - The simplest relation between a length-N sequence x[n], defined for $0 \le n \le N-1$, and its DTFT $X(e^{j\omega})$ is obtained by uniformly sampling $X(e^{j\omega})$ on the ω-axis between $0 \le \omega \le 2\pi$ at $\omega_k = 2\pi k/N$, $0 \le k \le N-1$





$$X(k)=X(z)\Big|_{z=e^{j\frac{2\pi}{N}k}},\quad k=0, 1, 2, ..., N-1$$

$$X(k)=X(e^{j\omega})\Big|_{\omega=\frac{2\pi}{N}k},\quad k=0, 1, 2, ..., N-1$$

结论:

(1)序列的N点DFT是序列傅里叶变换在频率区间[0, 2π]上的N点等间隔采样，采样间隔为2π /N。

(2)序列的N点DFT是序列的Z变换在单位圆上的N点等间隔采样，频率采样间隔为2π /N。

§ 5.3.1 Relation with the DTFT

- **From the definition of the DTFT have**

$$X[k] = X(e^{j\omega})\Big|_{\omega=2\pi k/N} = \sum_{n=0}^{N-1} x[n]e^{-j\omega n}\Big|_{\omega=2\pi k/N}$$

$$= \sum_{n=0}^{N-1} x[n]e^{-j2\pi kn/N}, \qquad 0 \le k \le N-1$$

  - **Note: X[k] is also a length-N sequence in the frequency domain**
  - **The sequence X[k] is called the discrete Fourier transform (DFT) of the sequence x[n]**
  - **Using the notation $W_N = e^{-j2\pi/N}$ the DFT is usually expressed as:**

$$X[k] = \sum_{n=0}^{N-1} x[n]W_N^{kn}, \; 0 \le k \le N-1$$

A1. Relation between DFT and DFS

$$\text{有限长序列} \xrightarrow{\text{周期延拓}} \text{周期序列} \xrightarrow{\text{取主值}} \text{主值区序列}$$

有限长序列 $x(n) \quad n = 0,1,2,\cdots M-1$

周期序列 $\tilde{x}_N(n) = \sum_{m=-\infty}^{\infty} x(n+mN) = x((n))_N$

$$0 \le n_0 \le N-1 \quad n = mN + n_0 \qquad ((n))_N = n_0$$

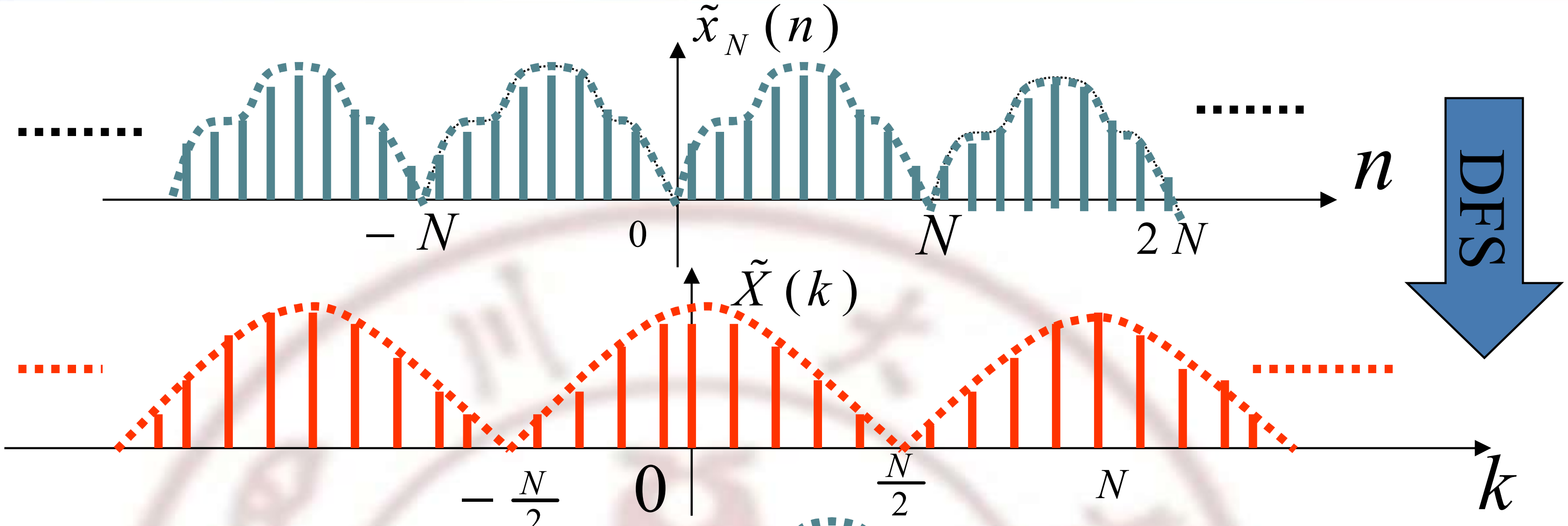
主值区间序列 $x_N(n) = \tilde{x}_N(n) R_N(n) \quad N \ge M, \quad x_N(n) = x(n)$

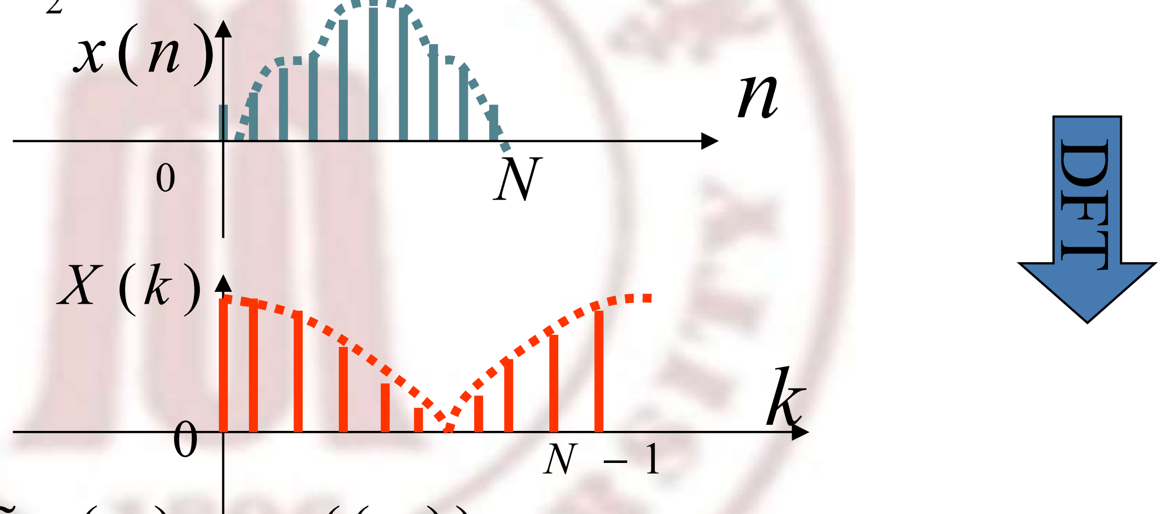
周期序列**DFS:** $\tilde{X}(k) = DFS[\tilde{x}_N(n)] = \sum_{n=0}^{N-1} \tilde{x}_N(n) W_N^{kn}$

$$= \sum_{n=0}^{M-1} x(n) W_N^{kn} \qquad -\infty < k < \infty$$

有限长序列的**DFT:**

$$X(k) = DFT[x(n)]_N = \sum_{n=0}^{N-1} x(n) W^{kn} = \sum_{n=0}^{M-1} x(n) W^{kn} \qquad 0 \le k \le N-1$$





$$DFS: \tilde{x}(n) \Leftrightarrow \tilde{X}(k)$$
$$DFT: x(n) \Leftrightarrow X(k)$$

$$\begin{cases} x(n) = \tilde{x}(n)R_N(n) & \tilde{x}_N(n) = x((n))_N \\ X(k) = \tilde{X}(k)R_N(k) & \tilde{X}(k) = X((k))_N \end{cases} \quad N \geq M$$

有限长序列x(n)的DFT变换X(k)，就是x(n)的周期延拓序列 $\tilde{x}(n)$ 的DFS系数 $\tilde{X}(k)$ 的主值序列

§ 5.3.2 Numerical Computation of the DTFT Using the DFT

- To numerical compute DTFT a length-N sequence x[n] using DFT. We can evaluate it at a dense grid of frequencies **$\omega_k = 2\pi k/M,\ 0 \le k \le M-1$.** Thus, compute M-point DFT M»N.

Define a new sequence $x_e[n]$ by augmenting x[n] with M-N zero-valued samples

$$x_e[n] = \begin{cases} x[n] & 0 \le n \le N-1 \\ 0 & N \le n \le M-1 \end{cases}$$

$$X[e^{j\omega_k}] = \sum_{n=0}^{N-1} x[n] e^{-j\omega_k n} = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/M}$$

$$= \sum_{n=0}^{M-1} x_e[n] e^{-j2\pi kn/M}$$

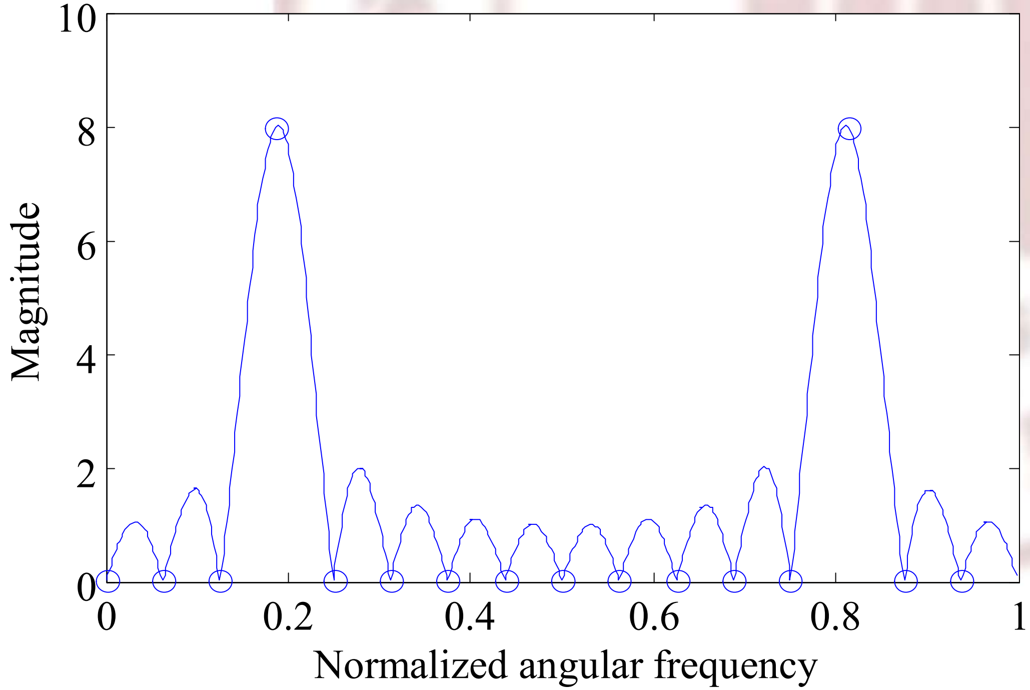
M-point DFT of the length-M sequence $x_e[n]$, M»N
M is an integer power of 2 (FFT requirement)

DFT Computation Using MATLAB

- **Example 5.5 compute the DFT and the DTFT of the sequence**

$$x[n] = \cos(6\pi n/16), \quad 0 \le n \le 15$$

$$G[k] = \begin{cases} N/2 = 8, & \text{for } k = r \quad = 3 \\ N/2 = 8, & \text{for } k = N - r = 13 \\ 0, & \text{otherwise} \end{cases}$$ example 5.2



Matlab first computes 512-point DFT(FFT), and then place the DFT (16points) sample on top of it.

○ indicates DFT samples

5.3.3~4 Frequency Sampling and interpolation

时域和频域的对偶原理 ⟷ 对时间序列$x(n)$的连续频谱函数在频域等间隔采样，则采样得到的离散频谱对应的时域序列必然是原时间序列$x(n)$ 的周期延拓序列。

而且仅对时域有限长序列，当满足频域采用定理时，才能由频域离散采样恢复原来的连续频谱函数（或原时间序列）。

时域采样 ⟶ 频域周期延拓　　时域周期延拓 ⟵ 频域采样

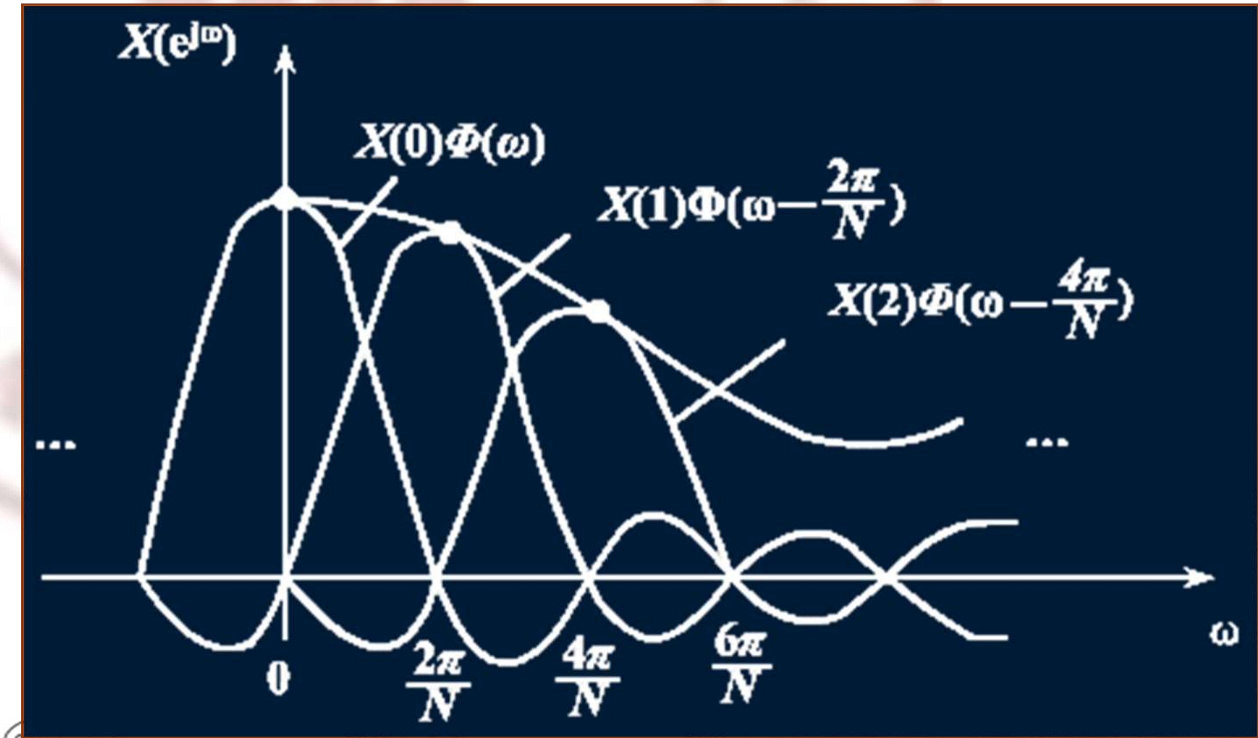
频域采样定理:
如果原序列x(n)长度为M，对 $X(e^{j\omega})$在频率区间[0，2π]上等间隔采样N点，得到 $\tilde{X}_N(k)$，则仅当采样点数N≥M时，才能由频域采样 $\tilde{X}_N(k)$ 恢复 $x(n)=\mathrm{IDFT}[X_N(k)]_N$ ，否则将产生时域混叠失真，不能由$\tilde{X}_N(k)$ 恢复原序列x(n)。

定理告诫我们，**只有当时域序列x(n)为有限长时，以适当的采样间隔对其频谱函数 $X(e^{j\omega})$ 采样，才不会丢失信息**。

内插公式： $$X(e^{j\omega})=\sum_{k=0}^{N-1}X(k)\varphi(\omega-\frac{2\pi}{N}k)$$

$$\varphi(\omega-\frac{2\pi}{N}k)=\begin{cases}1 & \omega=\frac{2\pi}{N}k=\omega_k \\ 0 & \omega=\frac{2\pi}{N}i=\omega_i \quad i\neq k\end{cases}$$

- 保证了各采样点上的值与原序列的频谱相同；
- 采样点之间为采样值与对应点的内插公式相乘，并叠加而成。



5.3.3~4 Frequency Sampling and interpolation

- 重构公式
- 重构函数——内插函数

$$\begin{cases} X[k]=\sum_{n=0}^{N-1} x[n]W_N^{kn}, & k=0\sim(N-1) \\ x[n]=\dfrac{1}{N}\sum_{k=0}^{N-1} X[k]W_N^{-kn}, & n=0\sim(N-1) \end{cases}$$

- Dirichlet or periodic sinc function 狄利克雷函数—周期sinc函数

$$D_K(\omega)=\frac{\sin(K\omega/2)}{K\sin(\omega/2)}=d(\omega,K)$$ 4π-周期Sinc函数

$$\mathrm{sinc}(t)=\frac{\sin(\pi t)}{\pi t} \Rightarrow \mathrm{sinc}\left(\frac{K\omega}{2\pi}\right)=\frac{\sin(K\omega/2)}{K\omega/2}$$

Dirichlet function-周期sinc函数

Sinc函数——非周期函数

$$d_k(\omega,K)=d\left(\omega-\frac{2\pi k}{K},K\right)$$ $k=0\sim K-1$

§ 5.4 Circular Convolution

Operations on Finite-Length Sequences

- **The DFT properties also show useful functions in signal processing applications.**
- **Two operations**
  - **(1) Circular Shift of a Sequence**
  - **(2) Circular Convolution**

Circular Shift of a Sequence

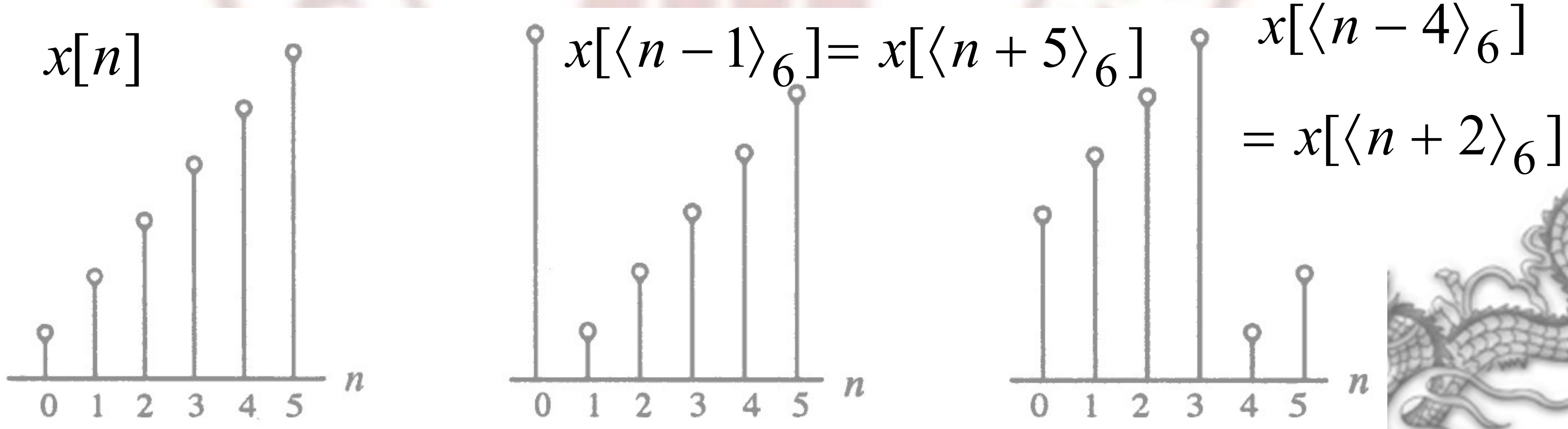
- **The desired shift, called the circular shift, is defined using a modulo operation:**

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

For $n_0 > 0$ (right circular shift), the above equation implies

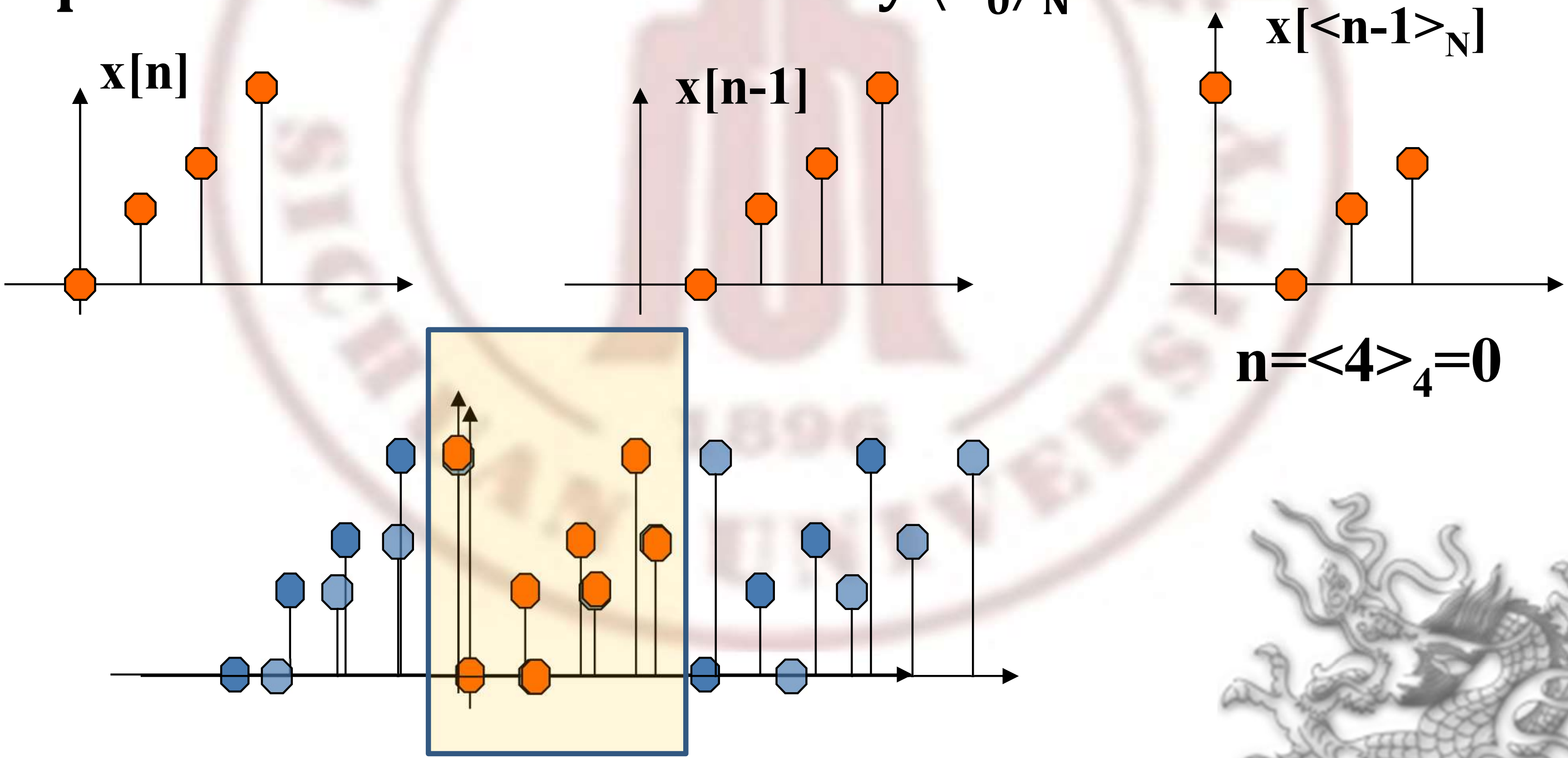
$$x_c[n] = \begin{cases} x[n - n_o], & \text{for } n_o \le n \le N-1 \\ x[N - n_o + n], & \text{for } 0 \le n < n_o \end{cases}$$

Illustration of the concept of a circular shift



Circular Shift of a Sequence

- **As can be seen from the previous figure, a right circular shift by n_0 is equivalent to a left circular shift by N-n_0 sample periods**
- **A circular shift by an integer number greater than *N* is equivalent to a circular shift by $\langle n_0 \rangle_N$**



§ 5.4 Circular Convolution

◈ Compare Linear convolution and circular convolution.

Consider two length-N sequences, g[n] and h[n].

- Linear convolution, results in a length(2N-1) sequence $y_L[n]$

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

The first nonzero value of $y_L[n]$ is $y_L[n]=g[0]h[0]$,

the last nonzero value is $y_L[2N-2]=g[N-1]h[N-1]$

- N-point circular convolution, results in a length-*N* sequence $y_C[n]$.

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

$$y[n] = g[n] \,Ⓝ\, h[n]$$

•commutative $\quad g[n] \,Ⓝ\, h[n] = h[n] \,Ⓝ\, g[n]$

§ 5.5Classifications of Finite-Length Sequences

- The definition of *circular conjugate-symmetric sequence* & *circular conjugate-antisymmetric sequence* .
  - (1) Circular conjugate-symmetry

$$x[n]=x^*[<-n>_N]=x^*[<N-n>_N],0\le n\le N-1$$

  - (2) Circular conjugate-antisymmetry

$$x[n]=-x^*[<-n>_N]$$
$$=-x^*[<N-n>_N],0\le n\le N-1$$

§ 5.5.1 Classification Based on Conjugate Symmetry

Any complex length-*N* sequence x[n] can be expressed as

$$x[n] = x_{cs}[n] + x_{ca}[n], 0 \le n \le N-1$$

$x_{cs}[n]$ is its circular conjugate-symmetric part, and $x_{ca}[n]$ its circular conjugate-antisymmetric part, defined by

$$x_{cs}[n] = \frac{1}{2}(x[n] + x^*[<-n>_N]), 0 \le n \le N-1$$

$$x_{ca}[n] = \frac{1}{2}(x[n] - x^*[<-n>_N]), 0 \le n \le N-1$$

- For a real sequence x[n], it can be expressed as

$$x[n] = x_{ev}[n] + x_{od}[n], 0 \le n \le N-1$$

$x_{ev}[n]$ is its circular even part, and $x_{ca}[n]$ is its circular odd part

$$x_{ev}[n] = \frac{1}{2}(x[n] + x[<-n>_N]) \quad x_{od}[n] = \frac{1}{2}(x[n] - x[<-n>_N])$$

§ 5.5.2 Classification Based on Geometric Symmetry

- Two types of geometric symmetries
  - **(1) Symmetric sequence**

$$x[n]=x[N-1-n]$$

  - **(2) Antisymmetric sequence**

$$x[n]=-x[N-1-n]$$

Since the length N of a sequence can be either even or odd, **four types of geometric symmetry are defined.**

Type1 -Symmetric impulse response with odd length.

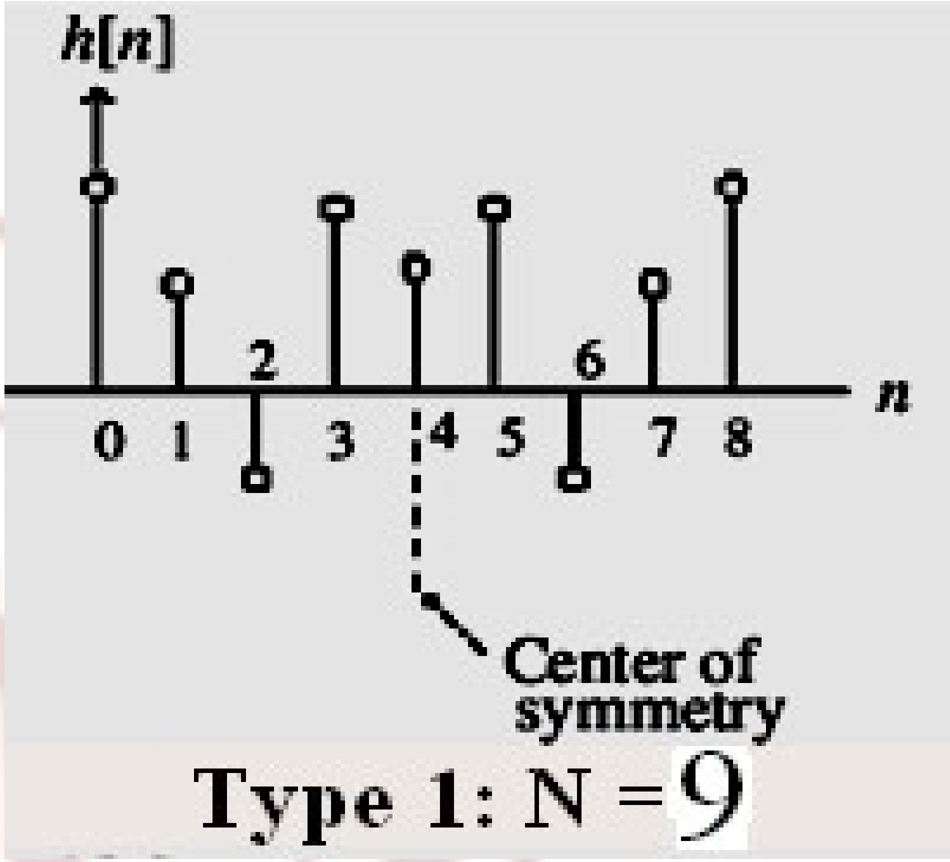
Type2 -Symmetric impulse response with even length.

Type3 -Antisymmetric impulse response with odd length.

Type4 - Antisymmetric impulse response with even length.

Question

- Assume a length 9 sequence which is Type 1 sequence.
- Try to compute it's phase function.



Consider the frequency response

$$X(e^{j\omega}) = x[0] + x[1]e^{-j\omega} + x[2]e^{-j2\omega} + x[3]e^{-j3\omega} + x[4]e^{-j4\omega}$$

$$+ x[5]e^{-j5\omega} + x[6]e^{-j6\omega} + x[7]e^{-j7\omega} + x[8]e^{-j8\omega}$$

- **Because of symmetry, we have x[0]=x[8], x[1] = x[7], x[2] = x[6], and x[3] = x[5]**

$$X\left(e^{j\omega}\right) = x[0](1+e^{-j8\omega}) + x[1](e^{-j\omega}+e^{-j7\omega})$$

$$+x[2](e^{-j2\omega}+e^{-j6\omega}) + x[3](e^{-j3\omega}+e^{-j5\omega}) + x[4]e^{-j4\omega}$$

$$= e^{-j4\omega}\{x[0](e^{j4\omega}+e^{-j4\omega}) + x[1](e^{j3\omega}+e^{-j3\omega})$$

$$+x[2](e^{j2\omega}+e^{-j2\omega}) + x[3](e^{j\omega}+e^{-j\omega}) + x[4]\}$$

$$X(e^{j\omega}) = e^{-j4\omega}\{2x[0]\cos(4\omega) + 2x[1]\cos(3\omega)$$

$$+2x[2]\cos(2\omega) + 2x[3]\cos(\omega) + x[4]\}$$

- **The quantity inside the braces is a real function of ω, and can assume positive or negative values in the range** $0 \le |\omega| \le \pi$
- **The phase function here is given by**

$$\theta(\omega) = -4\omega + \beta$$

where β is either 0 or π, and hence, it is a linear function of ω in the generalized sense

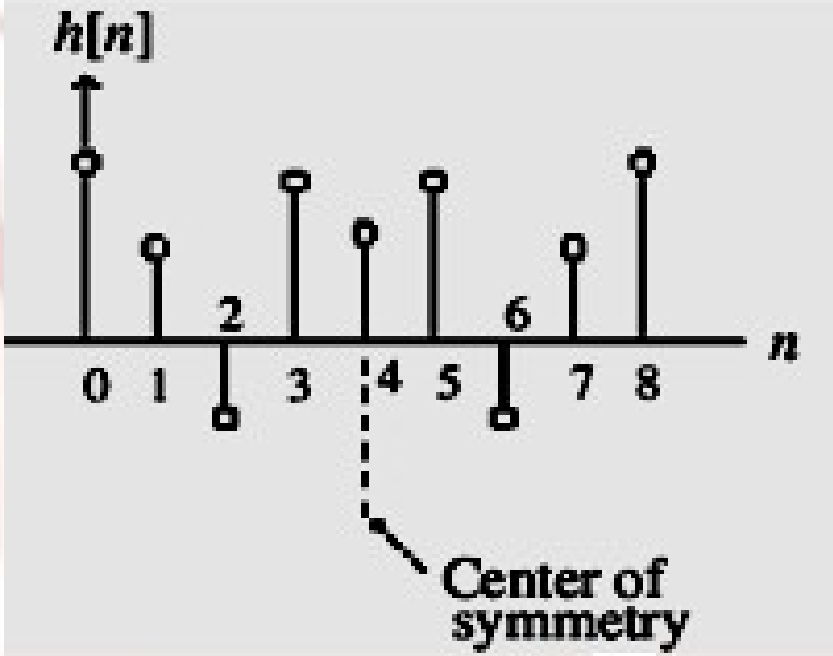
Type 1 -- Symmetric impulse response with odd length

- We get

$$\theta(\omega) = -4\omega + \beta$$

- In the general case, for Type 1 Linear phase sequences, the phase function is of the form

$$\theta(\omega) = -\frac{N-1}{2}\omega + \beta$$



Type 1: N = 9

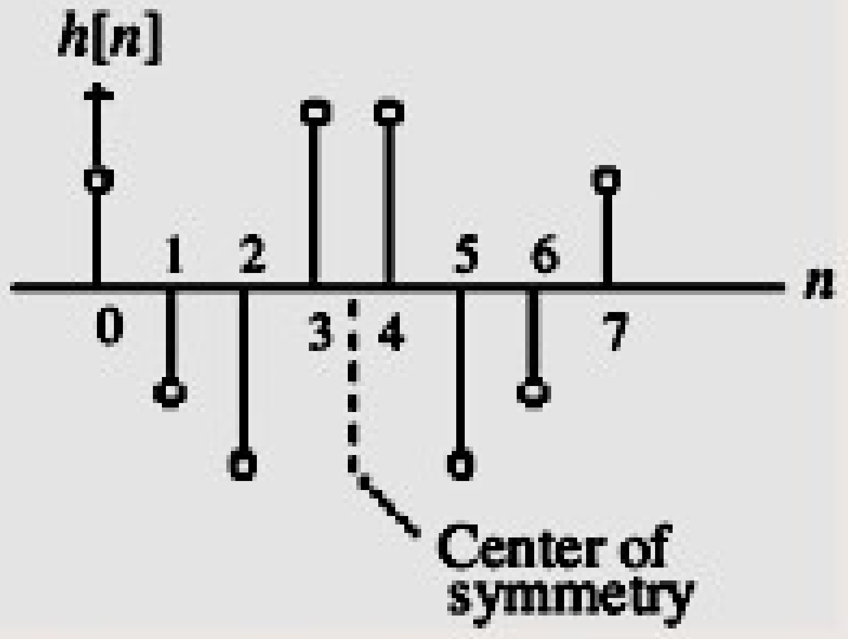
Type 2 -- Symmetric impulse response with even length

- For simplicity, assume length=8.
- We get

$$\theta(\omega) = -\frac{7}{2}\omega + \beta$$

- In the general case, for for Type 2 Linear phase sequences, the phase function is of the form

$$\theta(\omega) = -\frac{N-1}{2}\omega + \beta$$



Type 2: $N = 8$

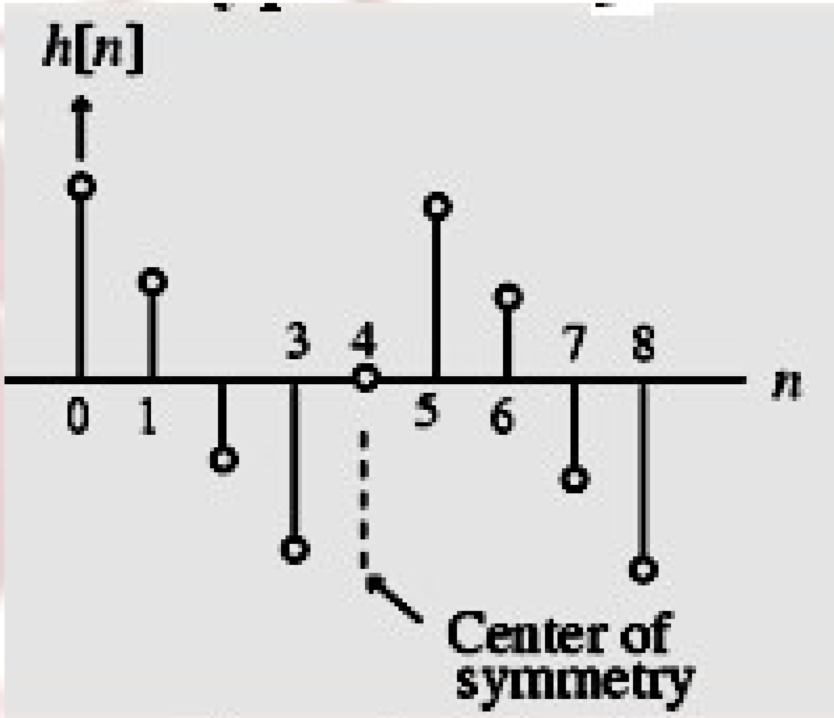
Type 3 -- Antisymmetric impulse response with odd length

- For simplicity, assume length=9.
- We get

$$\theta(\omega) = -4\omega + \frac{\pi}{2} + \beta$$

- In the general case, for for Type 3 Linear phase sequences, the phase function is of the form

$$\theta(\omega) = -\frac{N-1}{2}\omega + \frac{\pi}{2} + \beta$$



Type 3: N = 9

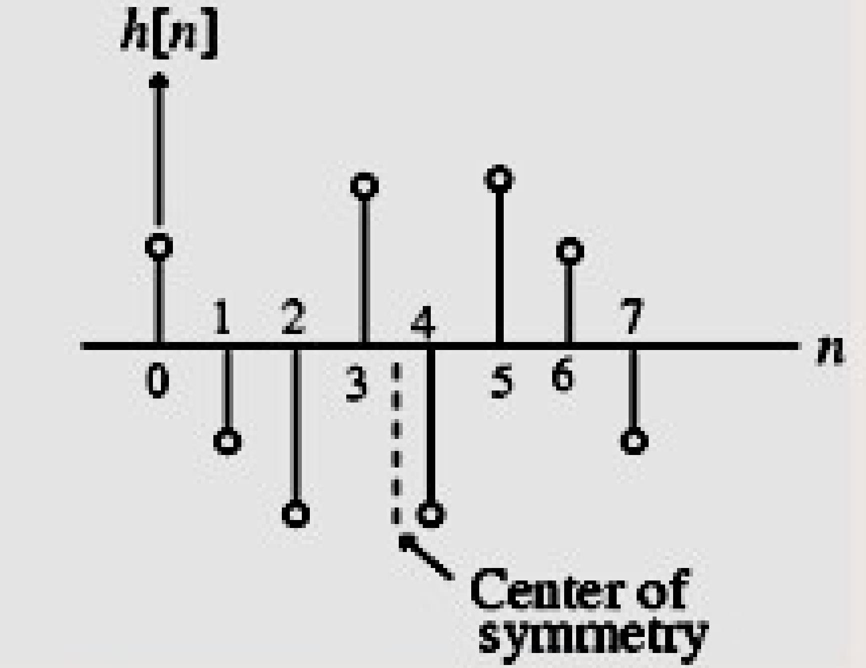
Type 4 -- Antisymmetric impulse response with even length

- For simplicity, assume length=8.
- We get

$$\theta(\omega) = -\frac{7}{2}\omega + \frac{\pi}{2} + \beta$$

- In the general case, for Type 4 Linear phase sequences, the phase function is of the form

$$\theta(\omega) = -\frac{N-1}{2}\omega + \frac{\pi}{2} + \beta$$



Type 4: N = 8

5.6 DFT Symmetry Relations

★ Even sequence、conjugate-symmetric sequence

实序列——偶序列　　　复序列——共轭对称序列

$$x[n] = x[-n] \qquad x[n] = x^*[-n]$$

★ Odd sequence、conjugate-antisymmetric sequence

实序列——奇序列　　　复序列——共轭反对称序列

$$x[n] = -x[-n] \qquad x[n] = -x^*[-n]$$

★ （任意）序列的对称分解 symmetric decomposition

实序列分解　　　复序列分解

$$x[n] = x_{\mathrm{EV}}[n] + x_{\mathrm{OD}}[n] \qquad x[n] = x_{\mathrm{CS}}[n] + x_{\mathrm{CA}}[n]$$

$$x_{\mathrm{EV}}[n] = \frac{x[n] + x[-n]}{2} \qquad x_{\mathrm{CS}}[n] = \frac{x[n] + x^*[-n]}{2}$$

$$x_{\mathrm{OD}}[n] = \frac{x[n] - x[-n]}{2} \qquad x_{\mathrm{CA}}[n] = \frac{x[n] - x^*[-n]}{2}$$

(1)Finite-length sequences——length-N sequence 有限长度序列的对称分解

★Periodic conjugate-symmetric sequence 周期共轭对称序列

$$x[n]=x^*[\langle -n\rangle_N]\leftarrow x_{\text{pcs}}[n] \qquad n=0\sim N-1$$

Real case: $x[n]=x[\langle -n\rangle_N]\leftarrow x_{\text{pev}}[n] \qquad n=0\sim N-1$

$x[0]=x[0]$ 周期偶（对称）序列

$x[1]=x[N-1]$

$x[2]=x[N-2]$

$\vdots$

$x[N-1]=x[1]$

★Periodic conjugate-antisymmetric sequence周期共轭反对称序列

$$x[n] = -x^*[\langle -n \rangle_N] \leftarrow x_{pca}[n] \qquad n = 0 \sim N-1$$

Real case： $x[n] = -x[\langle -n \rangle_N] \leftarrow x_{pod}[n] \quad n = 0 \sim N-1$

$x[0] = -x[0]$ 周期奇（对称）序列

$x[1] = -x[N-1]$

$x[2] = -x[N-2]$

$\vdots$

$x[N-1] = -x[1]$

★Symmetric decomposition有限长度序列的对称分解

·Complex case

$$x[n]=x_{\text{pcs}}[n]+x_{\text{pca}}[n]$$

$$x_{\text{pcs}}[n]=\frac{x[n]+x^{*}[\langle -n\rangle_N]}{2}$$

$$x_{\text{pca}}[n]=\frac{x[n]-x^{*}[\langle -n\rangle_N]}{2}$$

$$n=0\sim N-1$$

·Real case

$$x[n]=x_{\text{pev}}[n]+x_{\text{pod}}[n]$$

$$x_{\text{pev}}[n]=\frac{x[n]+x[\langle -n\rangle_N]}{2}$$

$$x_{\text{pod}}[n]=\frac{x[n]-x[\langle -n\rangle_N]}{2}$$

$$n=0\sim N-1$$

(2) 复序列DFT的共轭对称性

Length-N sequence $x[n] \Leftrightarrow X[k]$ N-point DFT

周期共轭对称性（Table5.1）

$x^*[n] \Leftrightarrow X^*[\langle -k \rangle_N]$ 即 $\mathrm{DFT}[x^*(n)] = X^*(N-k)$ $X(N) = X(0)$

或 $x^*[\langle -n \rangle_N] \Leftrightarrow X^*[k]$ $DFT[x^*(N-n)]_N = X^*(k)$

$$\mathrm{Re}\{x[n]\} \Leftrightarrow X_{\mathrm{pcs}}[k] = \tfrac{1}{2}\{X[k] + X^*[\langle -k \rangle_N]\}$$

$$\mathrm{j\,Im}\{x[n]\} \Leftrightarrow X_{\mathrm{pca}}[k] = \tfrac{1}{2}\{X[k] - X^*[\langle -k \rangle_N]\}$$

$$x_{\mathrm{pcs}}[n] \Leftrightarrow \mathrm{Re}\{X[k]\}$$

$$x_{\mathrm{pca}}[n] \Leftrightarrow \mathrm{j\,Im}\{X[k]\}$$

(3) 实序列DFT的共轭对称性

Length-N sequence $x[n] \Leftrightarrow X[k]$ **N-point DFT**

$$x_{\text{pcs}}[n] \Leftrightarrow \text{Re}\{X[k]\}$$

$$x_{\text{pca}}[n] \Leftrightarrow \text{j}\,\text{Im}\{X[k]\}$$

$$X(k) = X^*(N-k)$$

X[k]为共轭对称的

$$X[k] = X^*[\langle -k \rangle_N]$$

$$\text{Re}\{X[k]\} = \text{Re}\{X[\langle -k \rangle_N]\}$$

$$\text{Im}\{X[k]\} = -\text{Im}\{X[\langle -k \rangle_N]\}$$

$$|X[k]| = |X[\langle -k \rangle_N]|$$

$$\arg\{X[k]\} = -\arg\{X[\langle -k \rangle_N]\}$$

如果x(n)为实偶对称序列，
则X[k]为实偶对称；
如果x(n)为实奇对称序列，
则X[k]为纯虚奇对称

DFT Symmetric Relations

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

Table: 5.1

Note: x[n] is a complex sequence.

DFT Symmetric Relations

Length-N Sequencec	N-point DFT
$x[n]$	$X[k]=\mathrm{Re}\{X[k]\}+j\mathrm{Im}\{X[k]\}$
$x_{pe}[n]$	$\mathrm{Re}\{X[k]\}$
$x_{po}[n]$	$j\mathrm{Im}\{X[k]\}$
Symmetry relations	$X[k]=X^*[<-k>_N]$
	$\mathrm{Re}X[k]=\mathrm{Re}X[<-k>_N]$
	$\mathrm{Im}X[k]=-\mathrm{Im}X[<-k>_N]$
	$\lvert X[k]\rvert=\lvert X[<-k>_N]$
	$\arg X[k]=-\arg S[<-k>_N]$

Table： 5.2

Note: $x[n]$ is a real sequence.

§ 5.7 DFT Theorem

- **DFT Circular shifting Theorem—循环移位**
- ● **finite -length sequence circular shifting theorem**序列的循环移位
- ● **Circular Time-shifting Theorem**时域循环移位定理
- ● **Circular Frequency-shifting Theorem**频域循环移位定理

P. 215

- **DFT Circular Convolution Theorem—循环卷积**
- ● **finite -length sequence circular convolution theorem**有限长序列的循环卷积
- ● **Circular Convolution Theorem**时域循环卷积定理
- ● **Modulation Theorem** 时域调制定理（频域循环卷积定理）

P. 215~218

§ 5.7 DFT Properties

Type of Property	length-N sequence	N-point DFT
	$g[n]$	$G[k]$
	$h[n]$	$H[k]$
Linearity	$ag[n]+bh[n]$	$aG[k]+bH[k]$
Circular Time-shifting	$g[\langle n-n_0\rangle_N]$	$W_N^{kn_0}G[k]$
Frequency-shifting	$W_N^{-kn_0}g[n]$	$G[\langle k-k_0\rangle_N]$
Duality	$G[n]$	$N[g\langle -k\rangle_N]$
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	$\sum_{n=0}^{N-1} \lvert x[n]\rvert^2 = \frac{1}{N}\sum_{k=0}^{N-1} \lvert X[k]\rvert^2$	

§ 5.8 Fourier-Domain Filtering

- Often one is interested in removing the components of a finite-length discrete-time signal in one or more frequency bands.
- A straight forward approach to remove the unwanted components from a signal is to implement the filtering in the Fourier domain.
- Based on DTFT convolution theorem

$$\sum_{m=0}^{N-1} g[m]h[n-m] \longleftrightarrow G(e^{j\omega})H(e^{j\omega})$$

Example 5.14

§ 5.8 Fourier-Domain Filtering

- Practically, x[n] & h[n] are real-valued signal, the imaginary parts of the samples of the IDTFT of the product of their FT will be theoretically all zeros.
- The imaginary parts are very small numbers due to computational errors.
- The real part of the IDTFT, y[n] is kept as the filtered response.
- The Fourier-domain filtering is usually implemented using DFT.
- However, the DFT-based filtering will always lead to some small ripples in the filtered response.
- Why?

§ 5.9 Computation of the DFT of Real Sequences

- N-point DFTs of Two Real Sequences Using a Single N-Point DFT

- 2N-Point DFTs of a Real Sequences Using a Single N-Point DFT

§ 5.9.1 N-point DFTs of Two Real Sequences Using a Single N-Point DFT

- Two N-point DFTs can be computed using a single N-point DFT
- Define a complex length-N sequence

$$x(n)=g(n)+jh(n)$$

$$g(n)=\mathrm{Re}\{x(n)\}$$
$$h(n)=\mathrm{Im}\{x(n)\}$$

From DFT symmetry relation, we arrive that

$$G(k)=\frac{1}{2}\{X(k)+X^*(\langle -k\rangle_N)\}$$

$$H(k)=\frac{1}{2j}\{X(k)-X^*(\langle -k\rangle_N)\}$$

$$X^*(\langle -k\rangle_N)=X^*(\langle N-k\rangle_N)$$

X(k) is N-point DFT of x(n)

§ 5.9 .2 2N-point DFTs of a Real Sequences Using a Single N-Point DFT

- Let v(n) be a length-2N real sequence with a 2N-point DFT V(k)
- Define two length-N real sequences g(n) and h(n), G(k) and H(k) denote their N-point DFTs

$$g(n)=v(2n),\ h(n)=v(2n+1)\quad 0\leq n\leq N-1$$

$$\begin{aligned}
V(k) &= \sum_{n=0}^{2N-1} v(n)W_{2N}^{nk} \\
&= \sum_{n=0}^{N-1} v(2n)W_{2N}^{2nk} + \sum_{n=0}^{N-1} v(2n+1)W_{2N}^{(2n+1)k} \\
&= \sum_{n=0}^{N-1} g(n)W_{N}^{nk} + \sum_{n=0}^{N-1} h(n)W_{N}^{nk}W_{2N}^{k} \\
&= \sum_{n=0}^{N-1} g(n)W_{N}^{nk} + W_{2N}^{k}\sum_{n=0}^{N-1} h(n)W_{N}^{nk},\quad 0\leq k\leq 2N-1
\end{aligned}$$

$$V(k)=\sum_{n=0}^{N-1} g(n)W_N^{nk}+W_{2N}^{k}\sum_{n=0}^{N-1} h(n)W_N^{nk},\ 0\le k\le 2N-1$$

i.e.

$$V(k)=G(\langle k\rangle_N)+W_{2N}^{k}H(\langle k\rangle_N)\quad 0\le k\le 2N-1$$

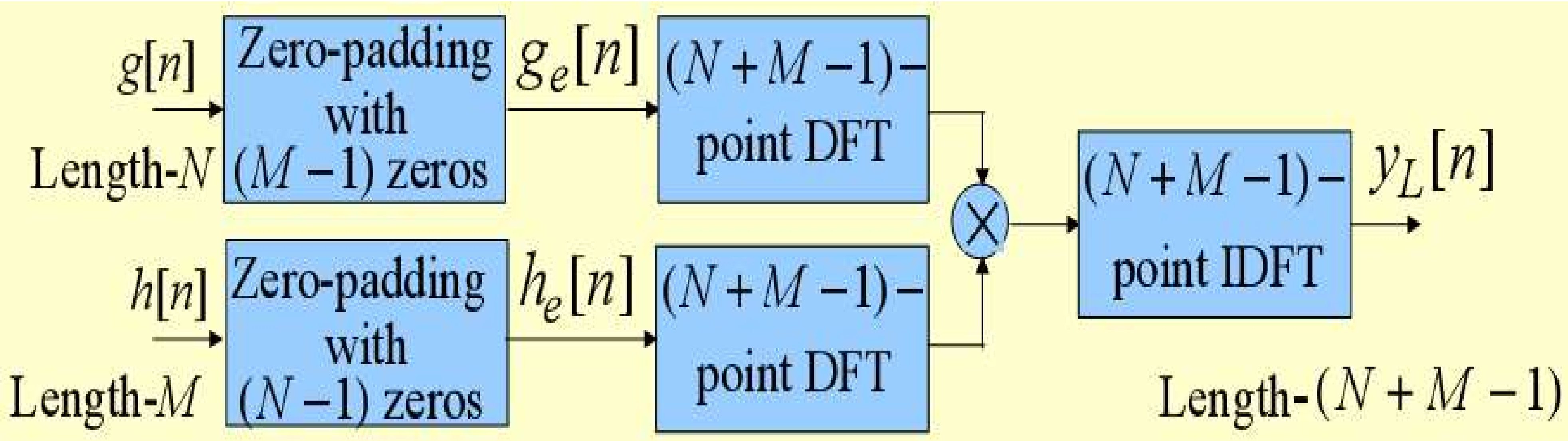
- Where the DFT of G(k) and H(k) can be compute by means of the method in 5.9.5 using a single N-Point DFT
- Example 5.16 P223

§ 5.10 Linear Convolution Using 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.**

- **Circular convolution can be computed using DFT**
- **For sequences x[n] with length N and y[n] with length N**
- **For what scenario, the result of the linear convolution is identical with that of the circular convolution?**
- **Linear convolution: $L = N + N - 1$**
- **Circular convolution: $L_c = N$**

- The corresponding implementation



Define two length-L sequences, L=N+M-1

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

$$\boxed{y_L[n]=g[n]*h[n]=y_C[n]=g_e[n]*h_e[n]}$$

Note: any value $L \ge N+M-1$ can be used. (In practice $L=2^v$, because DFT is computed using FFT)

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

- **Sometimes we need to perform a linear convolution of a finite-length sequence with a sequence that is of infinite length.**

  Such applications: The processing of a speech signal by an FIR filter

- **Objective: To develop computationally efficient DFT-based methods to implement the linear convolution.**

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

- 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).

- **How to realize it based on the method of the convolution of two finite-length sequences?**

Segmentation of the infinite-length sequence

Overlap-Add Method

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

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

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

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

$$y_m[n] = x_m[n] \textcircled{L}\ h[n]$$

- 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.

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 ≤ n ≤ N + M – 2**.

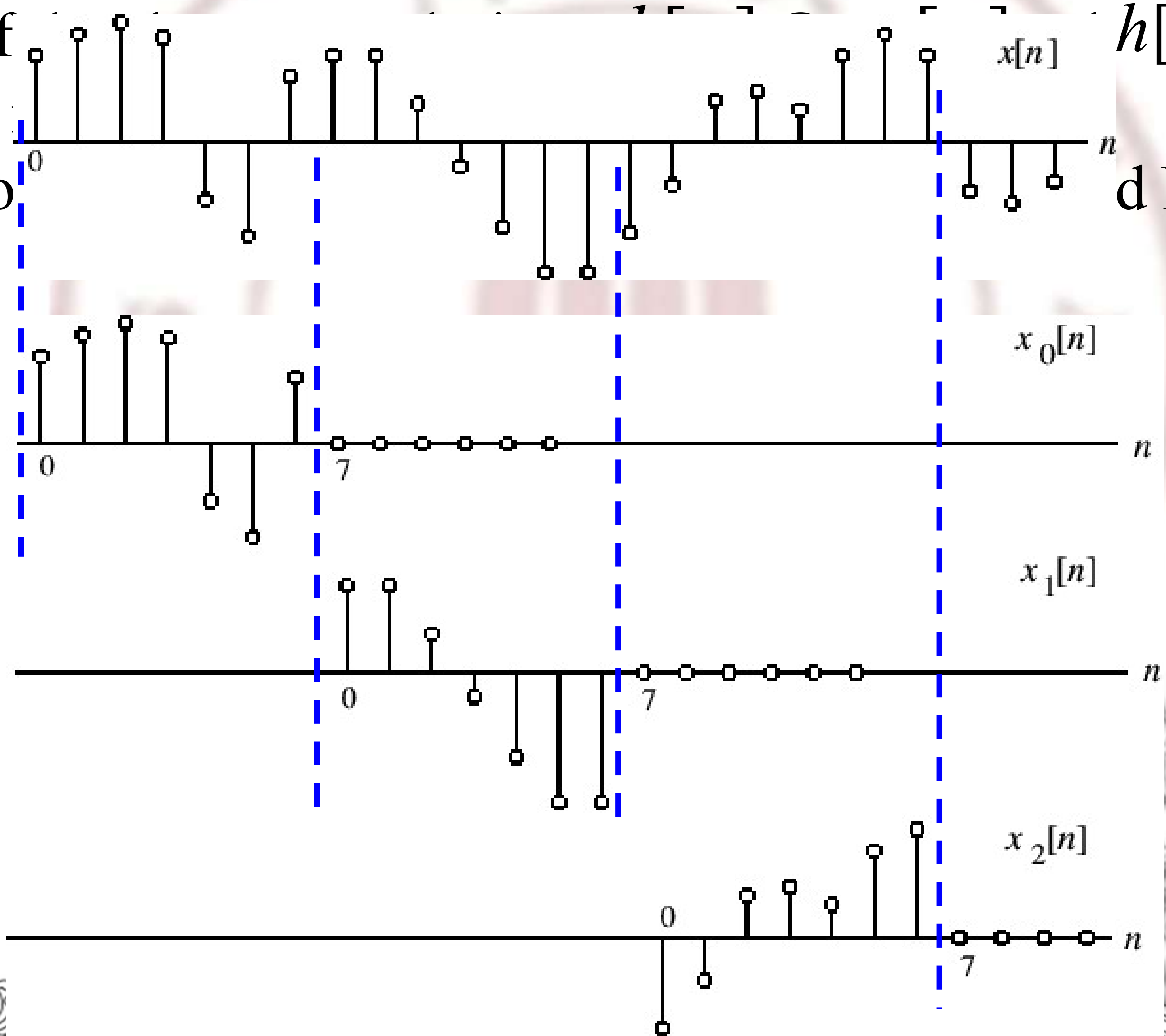
The second short convolution $y_1[n] = h[n] \circledast x_1[n]$ is also of length **N+M-1** but is defined for **N≤ n ≤ 2N + M – 2**.

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 ≤ n ≤3N + M – 2**.

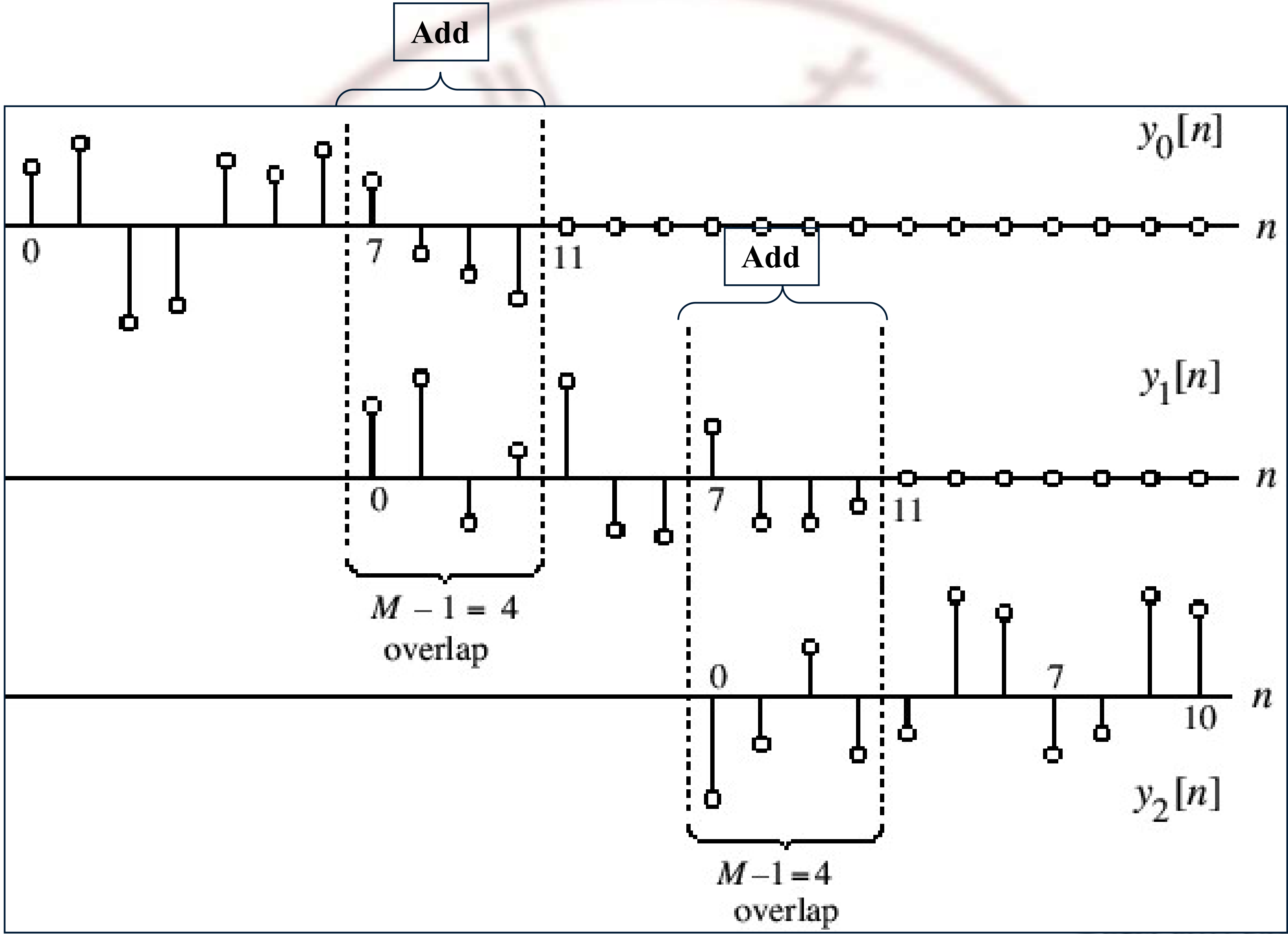
Note: overlap of M - 1 samples between two short linear convolutions.

Overlap-Add Method

- 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 … $h[n] \circledast x_{r-1}[n]$
  for rN≤ …
- This pro… d N = 7.



Overlap-Add Method



Overlap-Add Method

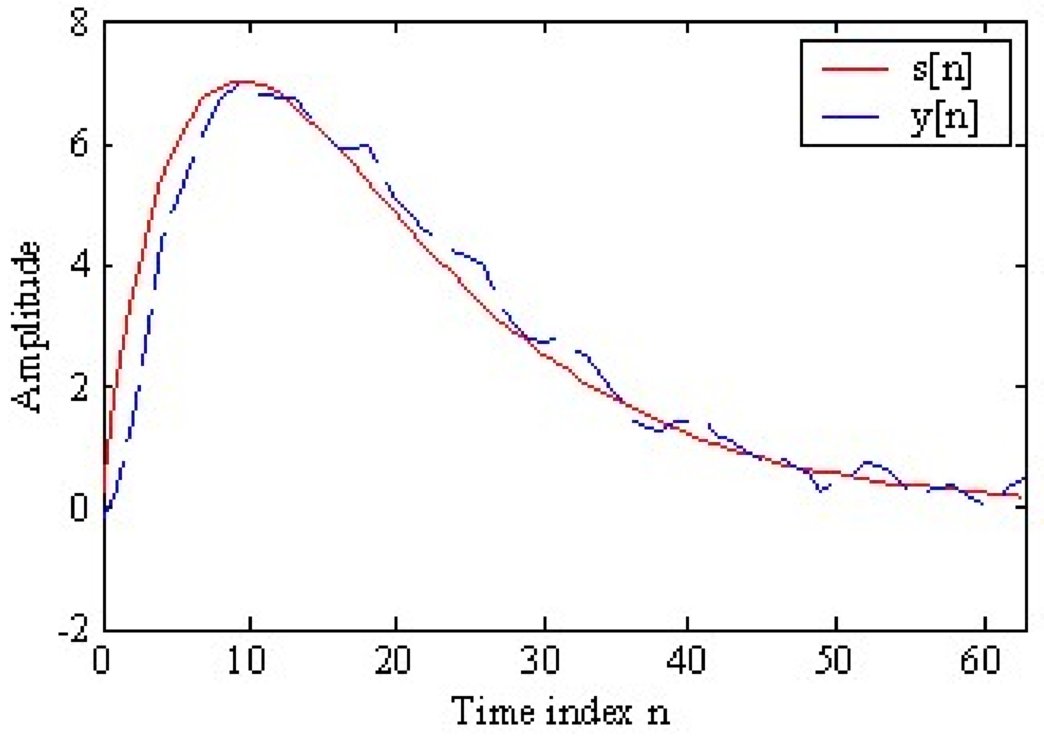
- Therefore, y[n] obtained by a linear convolution of x[n] and h[n] is given by

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

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.

Overlap-Add Method

- The function fftfilt can be used to implement the above method.
- Program 5-5 illustrates the use of fftfilt in the filtering of a noise-corrupted signal using a length-3 moving average filter.
- The plots generated by running this program is shown below



- 1) Exercises:

5.14, 5.20, 5.24, 5.29, 5.39, 5.45
5.49 5.68 5.79

- 2) Practice on MATLAB

-