# Chapter 2
# Discrete-Time Signals in the Time Domain

# Chapter 2
# Discrete-Time Signals in the Time Domain

- 2.1 Time-Domain Representation
- 2.2 Operations on Sequences
- 2.3 Operations on Finite-Length Sequences
- 2.4 Typical Sequences and Sequence Representation
- 2.5 The Sampling Process

# § 2.1 Discrete-Time Signals: Time-Domain Representation

- **In digital signal processing, signals represented as sequences of numbers, called samples**
- **Sample value of a typical signal or sequence denoted as $x[n]$ with $n$ being an integer in the range $-\infty \leq n \leq \infty$**
- **x[n] defined only for integer values of $n$ and undefined for noninteger values of $n$**
- **Discrete-time signal represented by $\{x[n]\}$**

# § 2.1 Discrete-Time Signals: Time-Domain Representation

- **Discrete-time signal may also be written as a sequence of numbers inside braces:**

$$\{x[n]\}=\{\ldots,-0.2,\underset{\uparrow}{2.17},1.1,0.2,-3.67,2.9,\ldots\}$$

- **The arrow is placed under the sample at time index n = 0**
- **In the above, x[-1]= -0.2, x[0]=2.17, x[1]=1.1, etc.**

# § 2.1 Discrete-Time Signals: Time-Domain Representation

- **Here, *n*-th sample is given by**

  $$x[n]=x_a(t)\,|_{t=nT}=x_a(nT),\quad n=\ldots,-2,-1,0,1,\ldots$$

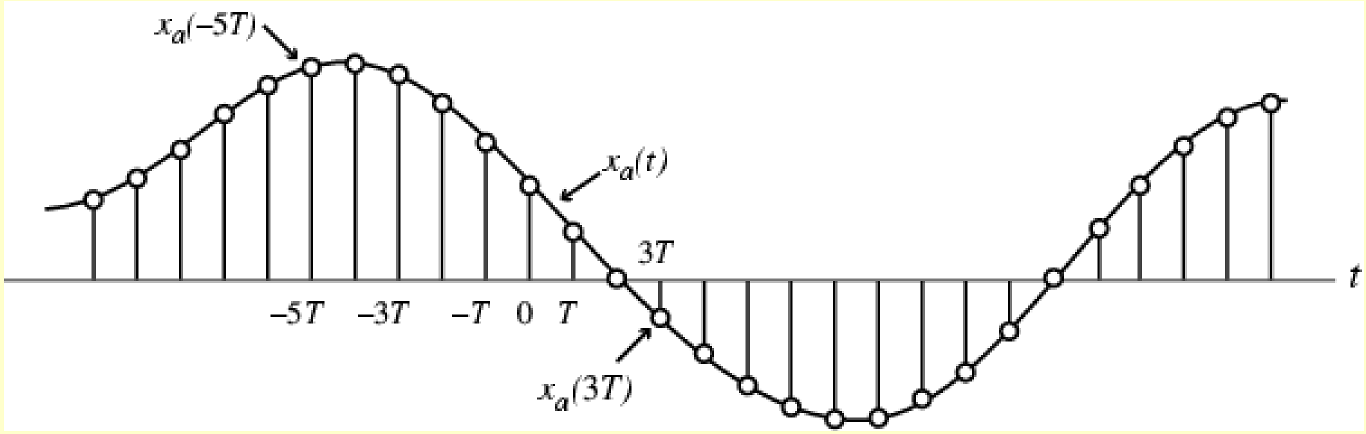
- **The spacing T between two consecutive samples is called the sampling interval or sampling period**
- **Reciprocal of sampling interval T, denoted as $F_T$ , is called the sampling frequency:**

  $$F_T=1/T$$

- **Unit of sampling frequency is cycles per second, or hertz (Hz), if T is in seconds**

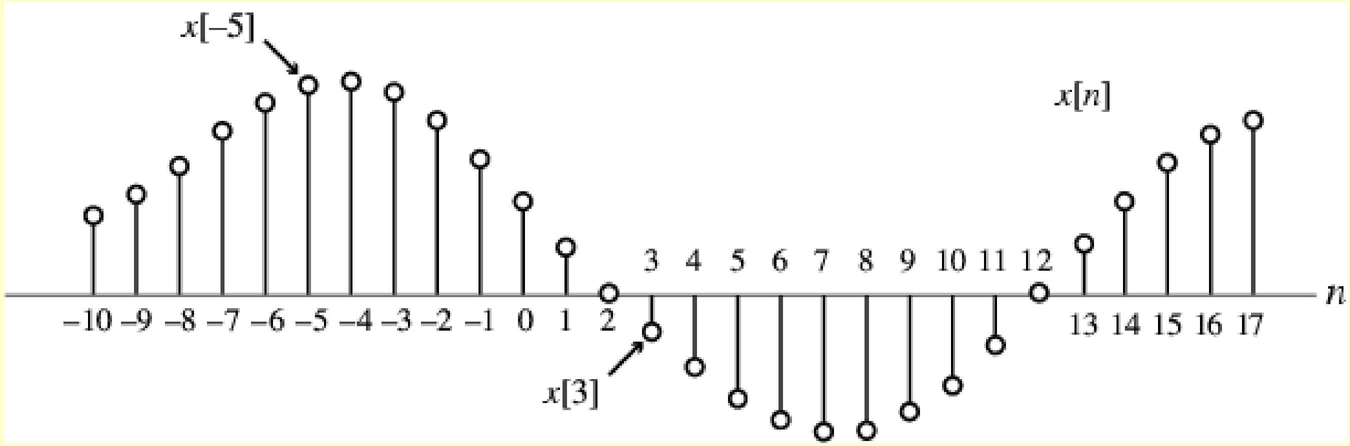
# § 2.1 Discrete-Time Signals: Time-Domain Representation

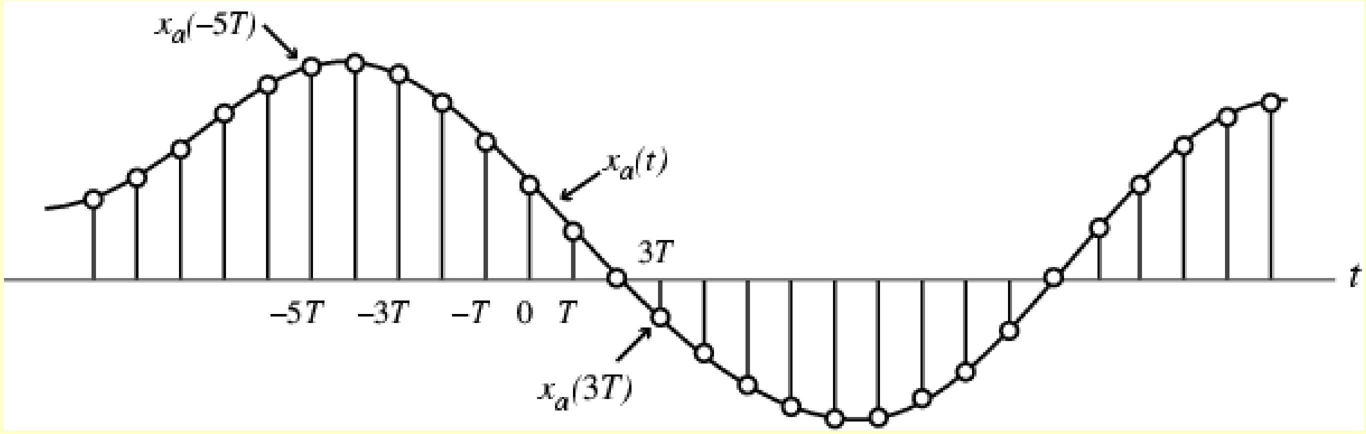
**A discrete-time sequence {*x*[*n*]} generated by periodically sampling a continuous-time signal $x_a(t)$ at uniform intervals of time**



# § 2.1 Discrete-Time Signals: Time-Domain Representation

- **Graphical representation of a discrete- time signal with real-valued samples :**



# § 2.1 Discrete-Time Signals: Time-Domain Representation

- **$\{x[n]\}$ is a real sequence, if the $n$-th sample $x[n]$ is real for all values of $n$**
- **Otherwise, $\{x[n]\}$ is a complex sequence**
- **A complex sequence $\{x[n]\}$ can be written as $\{x[n]\}=\{x_{re}[n]\}+j\{x_{im}[n]\}$ where $x_{re}$ and $x_{im}$ are the real and imaginary parts of $x[n]$**

# § 2.1 Discrete-Time Signals: Time-Domain Representation

- Example - $\{x[n]\} = \{\cos 0.25n\}$ is a real sequence
- $\{y[n]\} = \{e^{j0.3n}\}$ is a complex sequence
- We can write

$$\{y[n]\} = \{\cos 0.3n + j \sin 0.3n\}$$

$$= \{\cos 0.3n\} + j\{\sin 0.3n\}$$

where $\{y_{re}[n]\} = \{\cos 0.3n\}$

$$\{y_{im}[n]\} = \{\sin 0.3n\}$$

# § 2.1 Discrete-Time Signals: Time-Domain Representation

- **The complex conjugate sequence of {x[n]} is given by $\{x^*[n]\}=\{x_{re}[n]\} - j\{x_{im}[n]\}$**
- Example -

$$\{w[n]\} = \{\cos 0.3n\} - j\{\sin 0.3n\} = \{e^{-j0.3n}\}$$

is the complex conjugate sequence of $\{y[n]\}$

- That is,

$$\{w[n]\} = \{y^*[n]\}$$

- **Often the braces are ignored to denote a sequence if there is no ambiguity**

# § 2.1 Discrete-Time Signals: Time-Domain Representation

- **Two types of discrete-time signals:**

  **- Sampled-data signals in which samples are continuous-valued**

  **- Digital signals in which samples are discrete-valued**

- **Signals in a practical digital signal processing system are digital signals obtained by quantizing the sample values either by rounding or truncation**

# § 2.1 Discrete-Time Signals: Time-Domain Representation

- A discrete-time signal may be a finite-length or an infinite-length sequence
- Finite-length (also called finite-duration or finite-extent) sequence is defined only for a finite time interval: $N_1 \leq n \leq N_2$

  where $-\infty < N_1$ and $N_2 < \infty$ with $N_1 \leq N_2$
- Length or duration of the above finite-length sequence is $N = N_2 - N_1 + 1$

# § 2.1 Discrete-Time Signals: Time-Domain Representation

- Example - $x[n] = n^2, -3 \le n \le 4$ is a finite-length sequence of length $4-(-3)+1=8$

$y[n] = \cos 0.4n$ is an infinite-length sequence

# § 2.1 Discrete-Time Signals: Time-Domain Representation

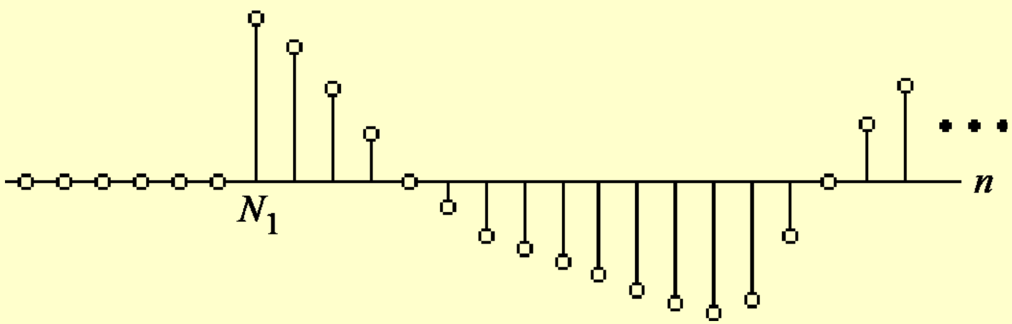
- A length-$N$ sequence is often referred to as an $N$**-point sequence**
- The length of a finite-length sequence can be increased by **zero-padding**, i.e., by **appending** it **with zeros**
  - Example -

$$x_e[n] = \begin{cases} n^2, & -3 \le n \le 4 \\ 0, & 5 \le n \le 8 \end{cases}$$

is a finite-length sequence of length 12 obtained by zero-padding $x[n] = n^2, -3 \le n \le 4$ with 4 zero-valued samples

# § 2.1 Discrete-Time Signals: Time-Domain Representation

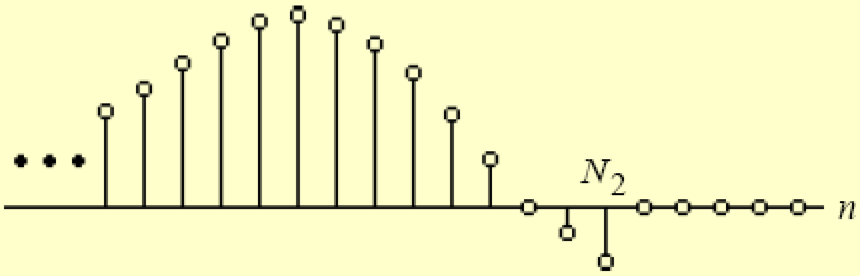
- **A** **right-sided sequence** ***x*[*n*] has zero-valued samples for n < $N_1$**




A right-sided sequence

**•If $N_1 \geq 0$, a right-sided sequence is called a** **causal sequence**

- A **left-sided sequence** $x[n]$ has zero-valued samples for $n > N_2$



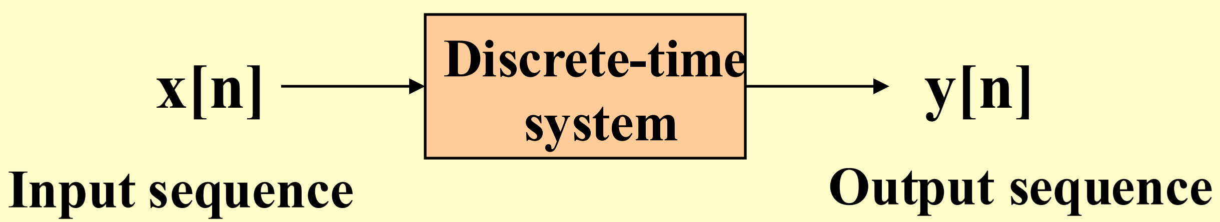

A left-sided sequence

- If $N_2 \leq 0$, a left-sided sequence is called a **anti-causal sequence**

- A **two-sided sequence** is defined for both negative and possitive values of $n$ .

# § 2.2 Operations on Sequences

- **A single-input, single-output discrete-time system operates on a sequence, called the input sequence, according some prescribed rules and develops another sequence, called the output sequence, with more desirable properties.**

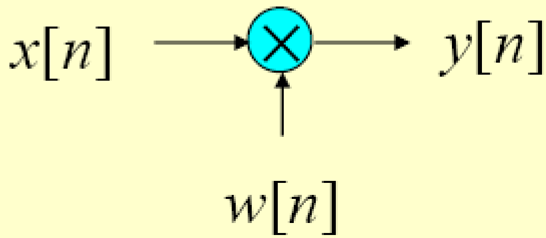


# § 2.2 Operations on Sequences

- **For example, the input may be a signal corrupted with additive noise**
- **Discrete-time system is designed to generate an output by removing the noise component from the input**
- **In most cases, the operation defining a particular discrete-time system is composed of some basic operations**

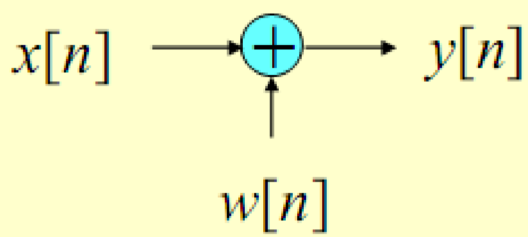
# § 2.2.1 Basic Operations

- **Product (modulation)** operation:
  - Modulator



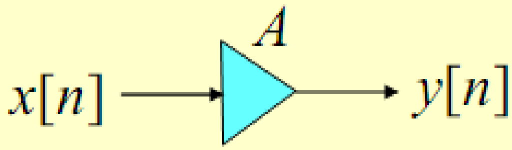

$$y[n] = x[n] \cdot w[n]$$

- **Addition** operation:
  - Adder




$$y[n] = x[n] + w[n]$$

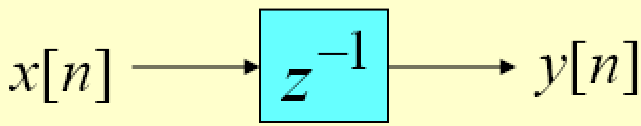
- **Multiplication** operation
  - Multiplier




$$y[n] = A \cdot x[n]$$

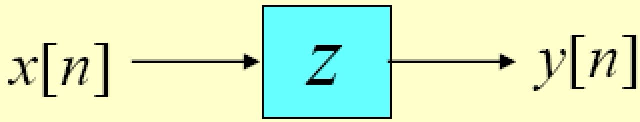
# § 2.2.1 Basic Operations

- **Time-shifting** operation: $y[n] = x[n-N]$
  where $N$ is an integer
- If $N > 0$, it is **delaying** operation
  - Unit delay




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

- If $N < 0$, it is an **advance** operation




$y[n] = x[n+1]$

  - Unit advance

# § 2.2.1 Basic Operations

- **Time-reversal (folding) operation:**

$$y[n]=x[-n]$$

# § 2.2.1 Basic Operations

- Example - Consider the sequence of length 3 defined for $0 \le n \le 2$: $\{f[n]\} = \{-2 \quad 1 \quad -3\}$
  $\{a[n]\} = \{3 \quad 4 \quad 6 \quad -9 \quad 0\}$ for $0 \le n \le 4$

- We therefore first append $\{f[n]\}$ with 2 zero-valued samples resulting in a length-5 sequence $\{f_e[n]\} = \{-2 \quad 1 \quad -3 \quad 0 \quad 0\}$
- Then

$$\{g[n]\} = \{a[n]\} + \{f_e[n]\} = \{1 \quad 5 \quad 3 \quad -9 \quad 0\}$$

# § 2.2.1 Basic operations

**Example:**

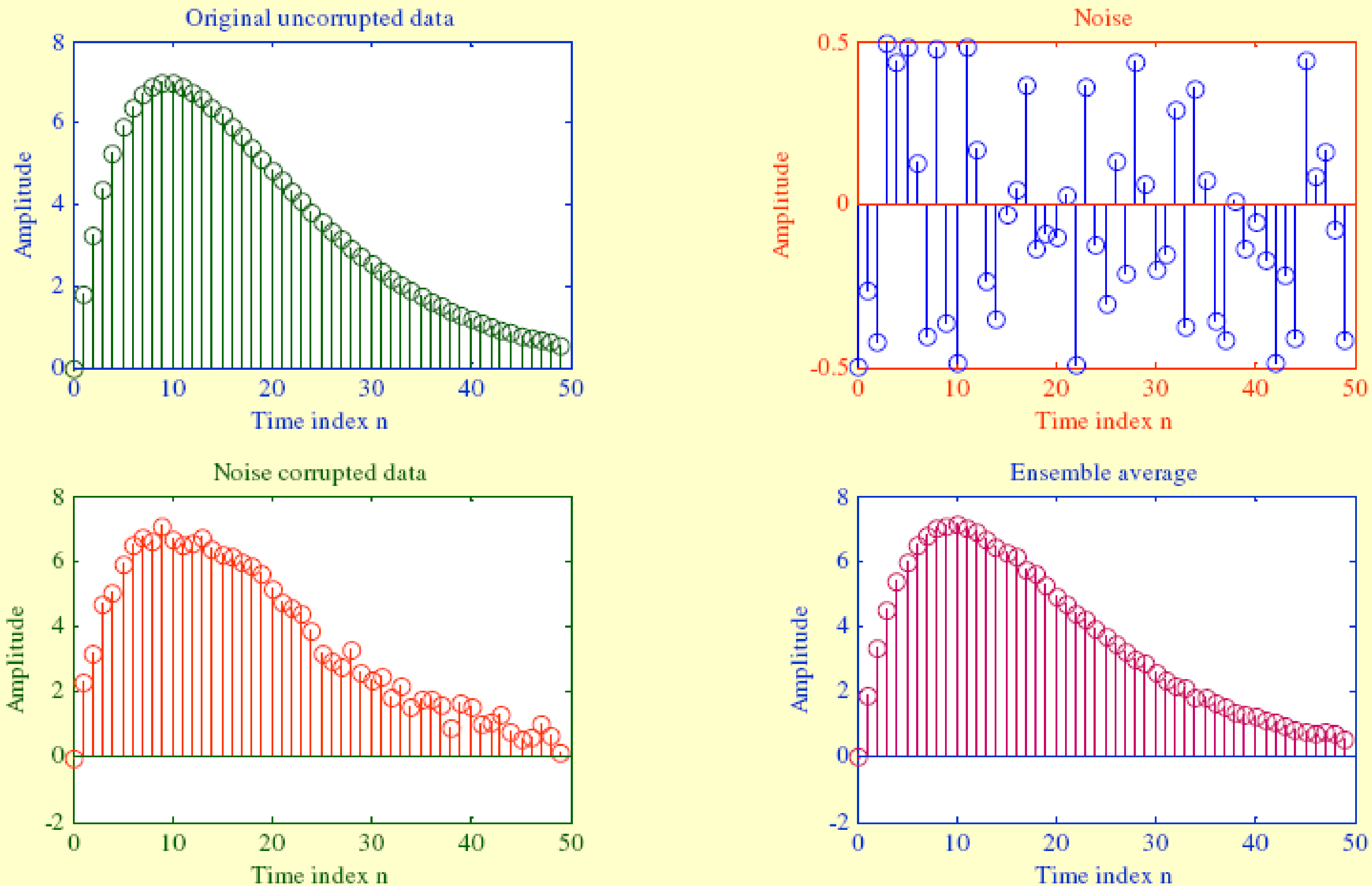
$$\underbrace{\mathbf{x}_i}_{\text{Measured data vector}} = \underbrace{\mathbf{s}}_{\text{Uncorrupted data vector}} + \underbrace{\mathbf{d}_i}_{\text{Noise vector}}$$

- The average data vector, called the ensemble average, obtained after $K$ measurements is given by

$$\mathbf{x}_{ave} = \frac{1}{K}\sum_{i=1}^{K}\mathbf{x}_i = \frac{1}{K}\sum_{i=1}^{K}(\mathbf{s}+\mathbf{d}_i) = \mathbf{s} + \frac{1}{K}\sum_{i=1}^{K}\mathbf{d}_i$$

- For large values of $K$, $\mathbf{x}_{ave}$ is usually a reasonable replica of the desired data vector

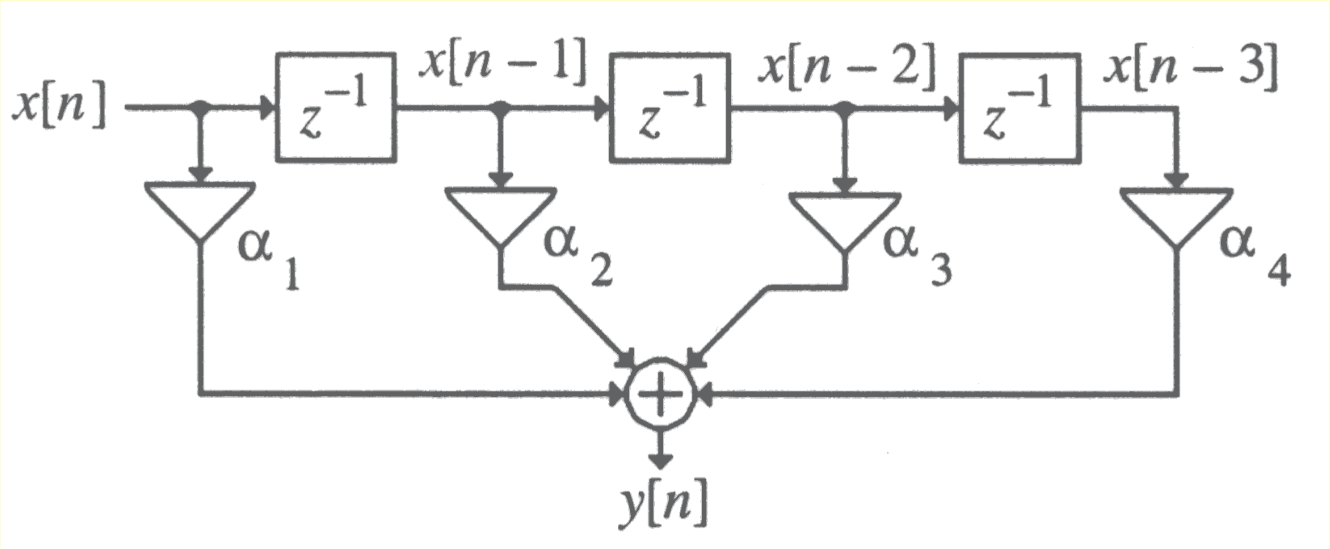
# • Example



Original uncorrupted data
Amplitude
Time index n
Noise
Amplitude
Time index n
Noise corrupted data
Amplitude
Time index n
Ensemble average
Amplitude
Time index n

# § 2.2.2 Combinations of Basic Operations

- **<u>A Simple Example</u>:**




$$y[n]=\alpha_1 x[n]+\alpha_2 x[n-1]+\alpha_3[n-2]+\alpha_4 x[n-3]$$

# § 2.2.3 Convolution Sum

- **The summation**

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = \sum_{k=-\infty}^{\infty} x[n-k]h[n]$$

**is called the convolution sum of the sequences $x[n]$ and $h[n]$ and represented compactly as**

$$y[n] = x[n] \circledast h[n]$$

# § 2.2.3 Convolution Sum

- **Interpretation -**
- 1) Time-reverse $x[k]$ to form $x[-k]$
- 2) Shift $x[-k]$ to the right by $n$ sampling periods if $n > 0$ or shift to the left by $n$ sampling periods if $n < 0$ to form $x[n-k]$
- 3) Form the product $v[k] = h[n]x[n-k]$
- 4) Sum all samples of $v[k]$ to develop the $n$-th sample of $y[n]$ of the convolution sum

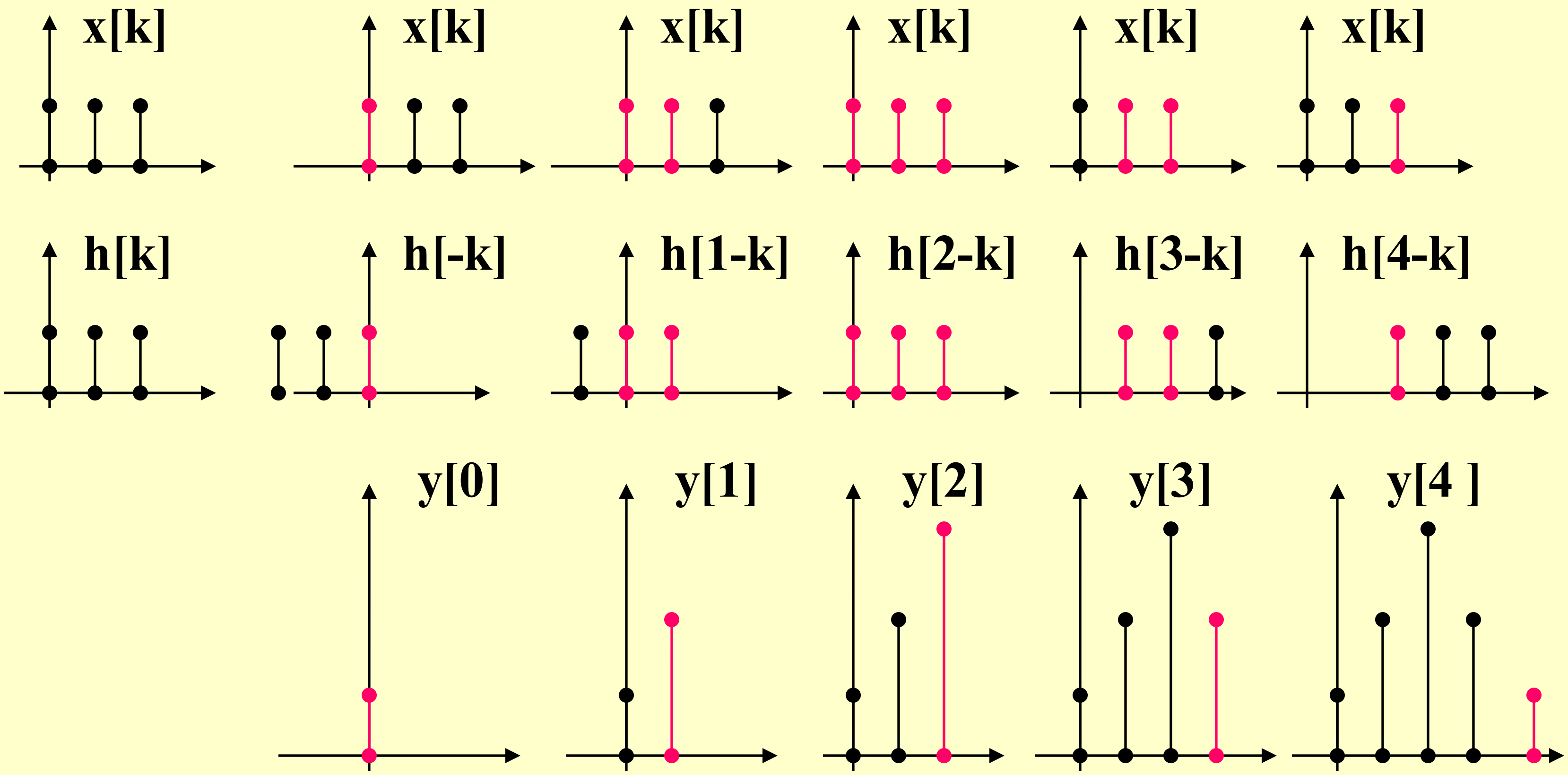
# § 2.2.3 Convolution Sum

- **<u>Example:</u> Develop the sequence y[n] generated by the convolution of the sequences x[n] and h[n] :**

  **x[n] = h[n] = δ[n] + δ[n-1] + δ[n-2]**

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

$$x[n] = h[n] = \delta[n] + \delta[n-1] + \delta[n-2]$$




$$y[n]=\delta[n]+2\delta[n-1]+3\delta[n-2]+2\delta[n-3]+\delta[n-4]$$

## § 2.2.3 Convolution Sum

- Note: The sum of indices of each sample product inside the convolution sum is equal to the index of the sample being generated by the convolution operation
- For example, the computation of $y[3]$ in the previous example involves the products $x[0]h[3]$, $x[1]h[2]$, $x[2]h[1]$, and $x[3]h[0]$
- The sum of indices in each of these products is equal to 3

# § 2.2.3 Convolution Sum

- In the example considered the convolution of a sequence $\{x[n]\}$ of length 5 with a sequence $\{h[n]\}$ of length 4 resulted in a sequence $\{y[n]\}$ of length 8
- In general, if the lengths of the two sequences being convolved are $M$ and $N$, then the sequence generated by the convolution is of length $M+N-1$

# § 2.2.3 Convolution Sum

- **In Matlab, the M-file conv implements the convolution sum of two finite-length sequences**
- **If a=[-2 0 1 -1 3]**

  **b=[1 2 0 -1]**

  **then conv(a,b) yields**

  **[-2 -4 1 3 1 5 1 -3]**

# § 2.2.4 Sampling Rate Alteration

- **Sampling Rate Alteration --**

  **Employed to generate a new sequence $y[n]$ with a sampling rate $F_T'$ higher or lower than that of the sampling rate $F_T$ of a given sequence $x[n]$**

- **Sampling rate alteration ratio is** $$R = \frac{F_T'}{F_T}$$
- **If $R > 1$, *the process called interpolation***
- **If $R < 1$, *the process called decimation***

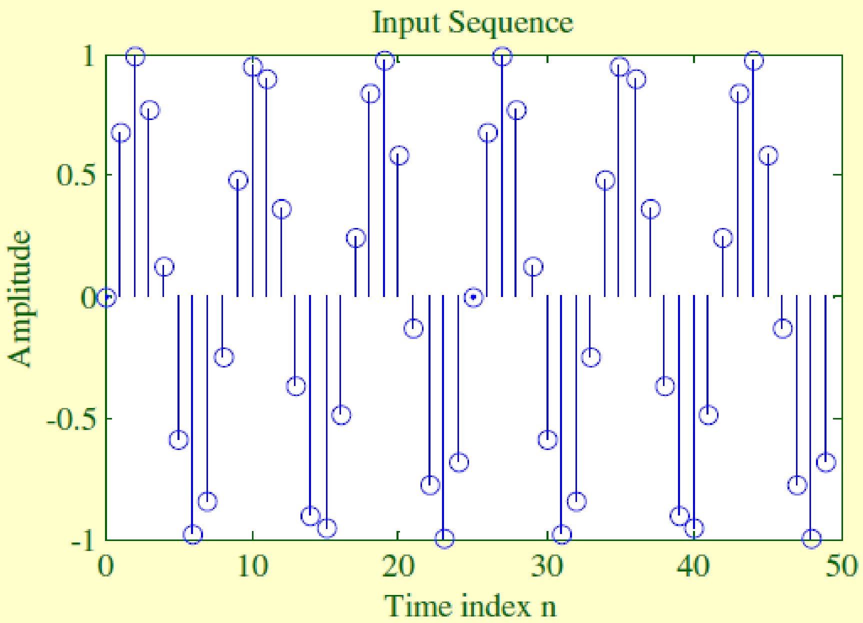
# § 2.2.4 Sampling Rate Alteration

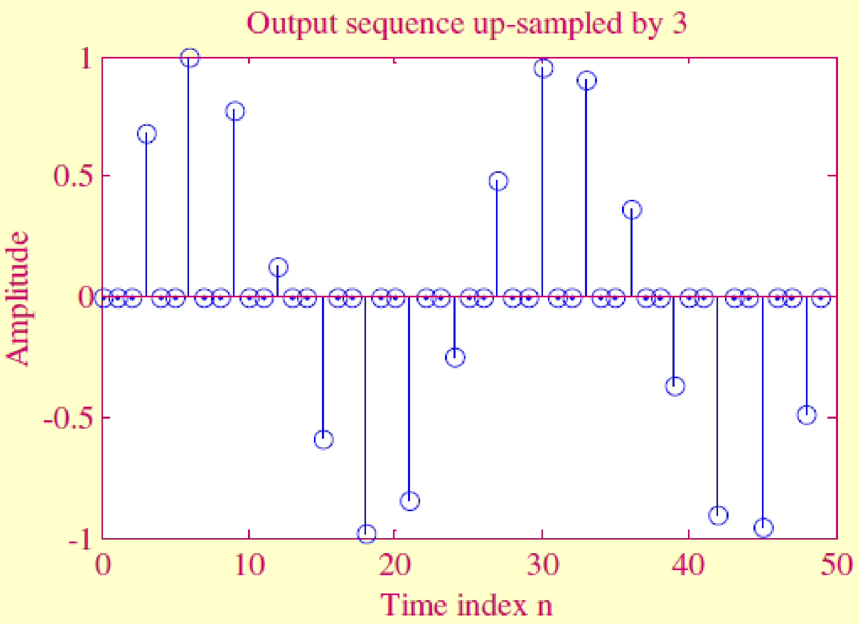
- In **up-sampling by an integer factor** $L > 1$, **L-1 equidistant zero-valued samples are inserted by the up-sampler between each two consecutive samples of the input sequence** $x[n]$**:**

$$x_u[n] = \begin{cases} x(n/L) & n = 0, \pm L, \pm 2L, \ldots \\ 0 & otherwise \end{cases}$$

# § 2.2.4 Sampling Rate Alteration

- An example of the up-sampling operation



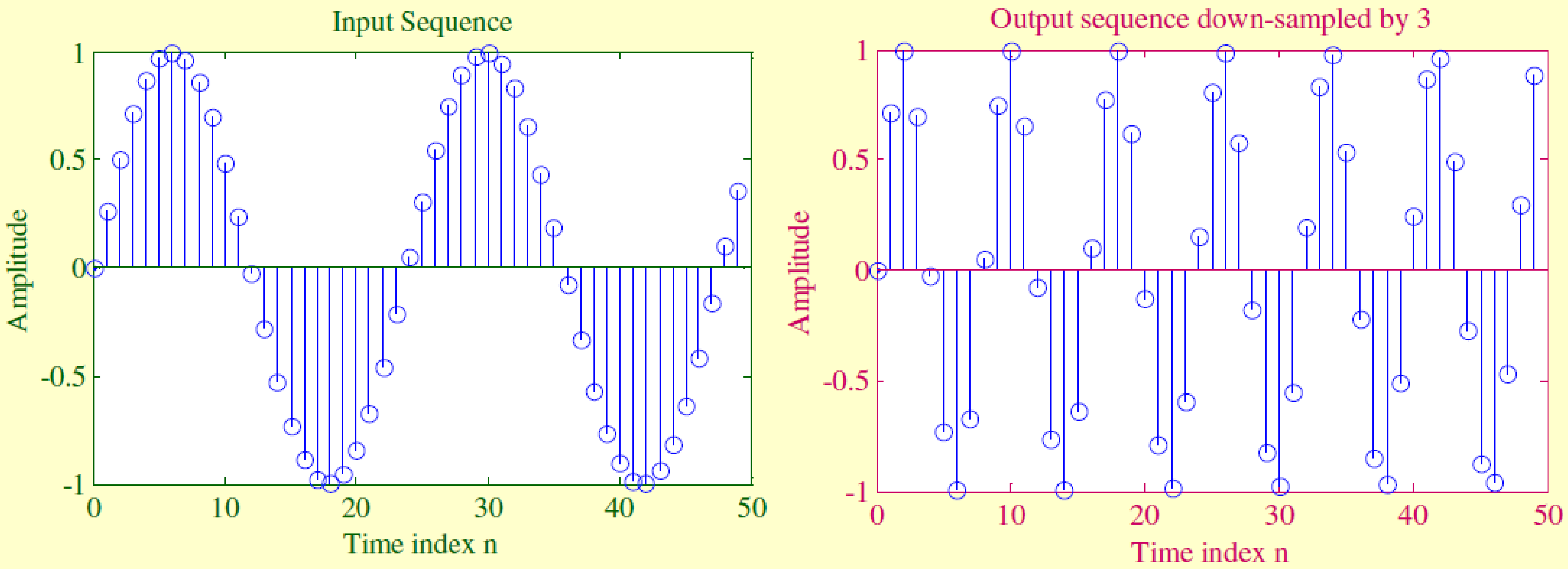
# § 2.2.4 Sampling Rate Alteration

- **In down-sampling by an integer factor** $M > 1$**, every** $M$-th **samples of the input sequence are kept and** $M$**-1 in-between samples are removed:**

$$y[n]=x[nM]$$

# § 2.2.4 Sampling Rate Alteration

- An example of the down-sampling operation



# 2.3 Operations on Finite-Length Sequences

## 2.3.1 Circular Time-Reversal Operation

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

$$m \ \text{modulo} \ N$$

**is called the residue**

- **The residue $r$ is an integer with a value between 0 and $N$-$1$**

# § 2.3.1 Circular Time Time-Reversal Operation

- **The modulo operation is denoted by the notation**

$$\langle m \rangle_N = m \text{ modulo } N$$

- **If we let $r = \langle m \rangle_N$ then $r = m + lN$ where $l$ is a positive or negative integer chosen to make $m+lN$ an integer between 0 and $N-1$**

# § 2.3.1 Circular Time Time-Reversal 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, $\langle -15\rangle_7 = 6$

# § 2.3.1 Circular Time Time-Reversal Operation

- The circular time-reversal version $\{y[n]\}$ of a length-N sequence $\{x[n]\}$ defined for $0 \leq n \leq N\text{-}1$ 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 $\{y[n]\}=\{x[\langle -n\rangle_5]\}$

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

# § 2.3.2 Circular Shift of a Sequence

- The **circular shift operation** for a finite- length sequence is defined using the modulo operation
- Let $x[n]$ be a length-$N$ sequence defined for

  $0 \leq n \leq N\text{-}1$
- Its circularly shifted version $x_c[n]$, shifted $n_0$ by samples, is given by

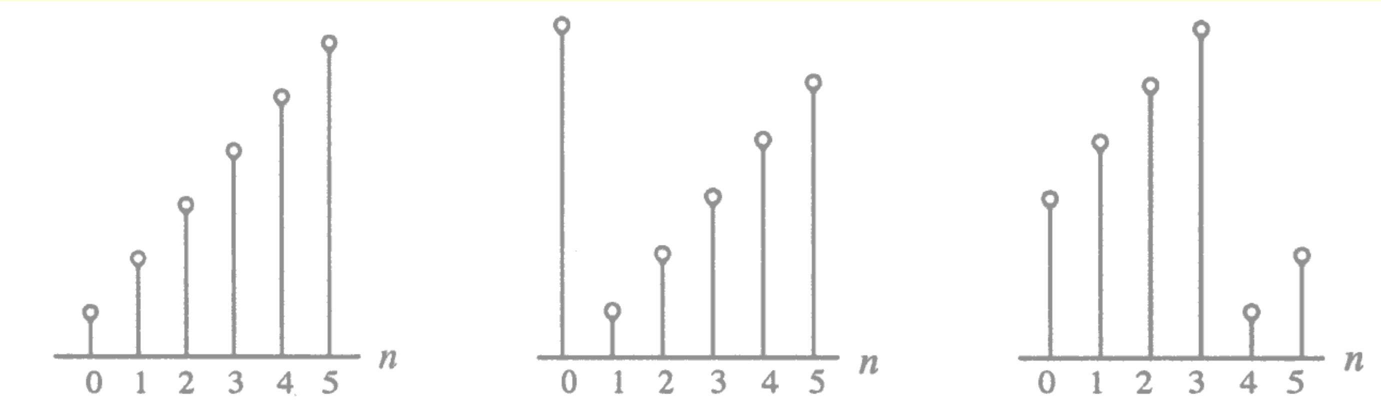
$$x_c[n]=x[<n-n_0>_N]$$

$x_c[n]$ is also a length-N sequence defined for

$0 \leq n \leq N\text{-}1$

# § 2.3.2 Circular Shift of a Sequence

- Illustration of the concept of a circular shift



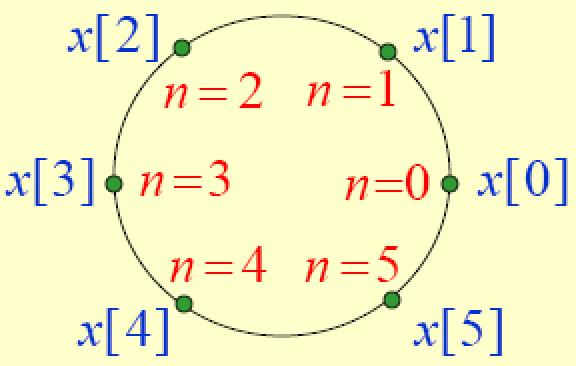

$x[n]$

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

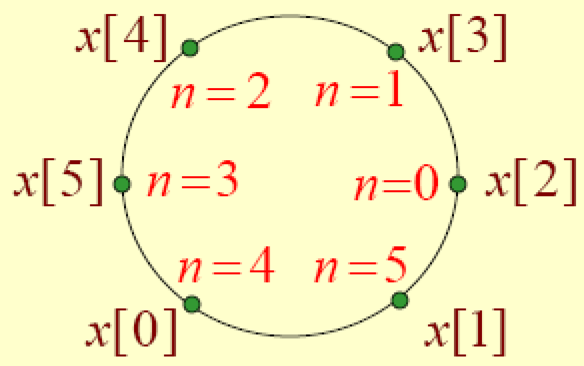
$x[\langle n-4 \rangle_6] = x[\langle n+2 \rangle_6]$

# § 2.3.2 Circular Shift of a Sequence

- **If the length-*N* sequence is displayed on a circle at *N* equally spaced points, then the circular shift operation can be viewed as a clockwise or anti-clockwise rotation of the sequence by $n_0$ sample spacings.**



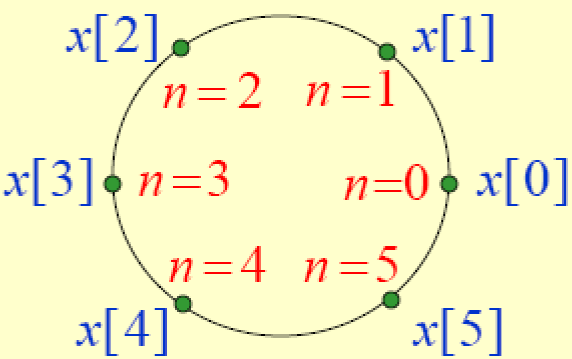

$x[n]$



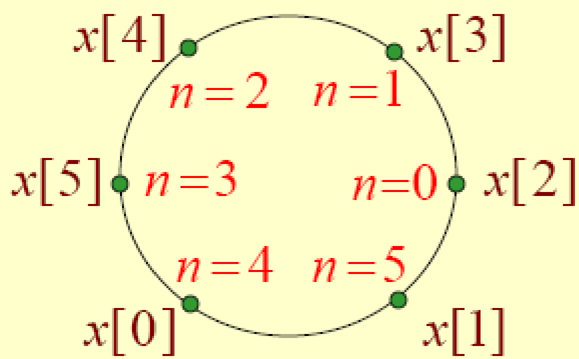

$x[\langle n-4\rangle_6] = x[\langle n+2\rangle_6]$

# § 2.3.2 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**

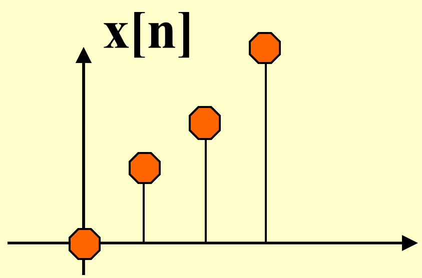


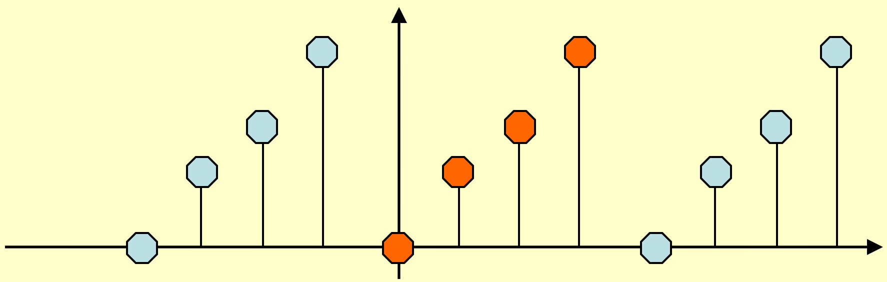

$x[n]$

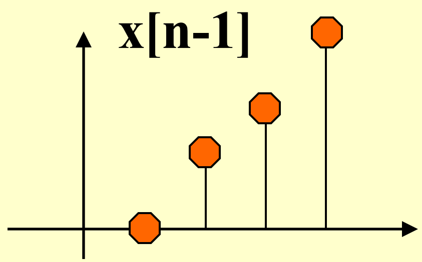



$x[\langle n-4\rangle_6] = x[\langle n+2\rangle_6]$

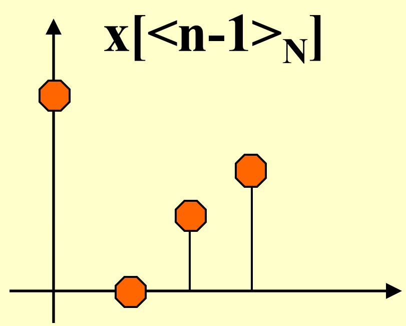
# § 2.3.2 Circular Shift of a Sequence

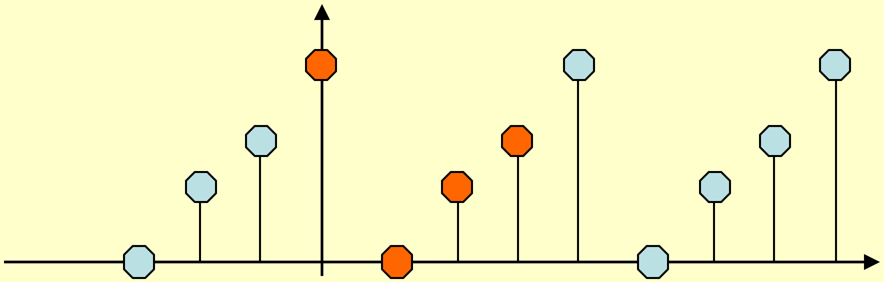





$$n=\langle 4\rangle_4=0$$



# § 2.3.3 Classification of Sequences

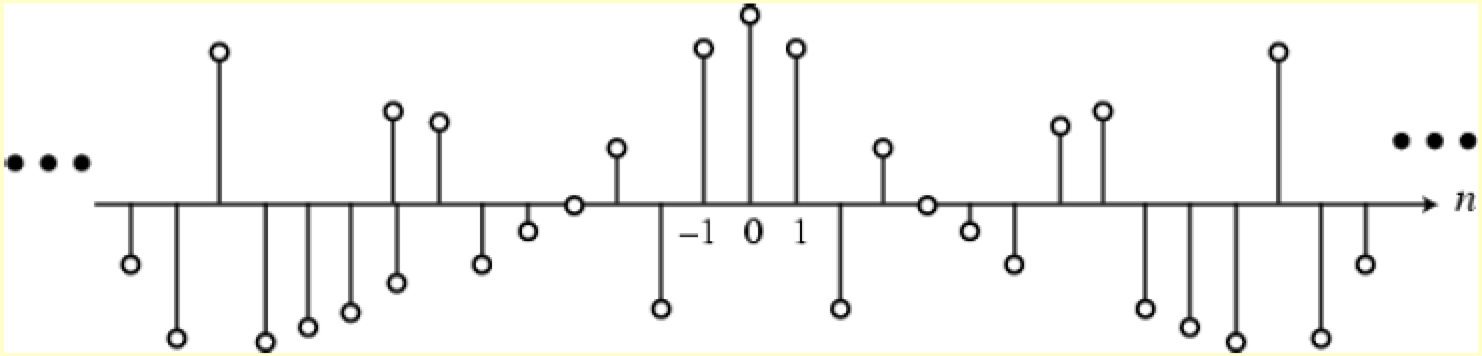
- There are several types of classification
- One classification is in terms of the number of samples defining the sequence
- Another classification is based on its symmetry with respect to time index $n = 0$
- Other classifications in terms of its other properties, such as periodicity, summability, energy and power

# § 2.3.3 Classification of Sequences

- **Conjugate-symmetric sequence**:

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

If $x[n]$ is real, then it is an **even sequence**



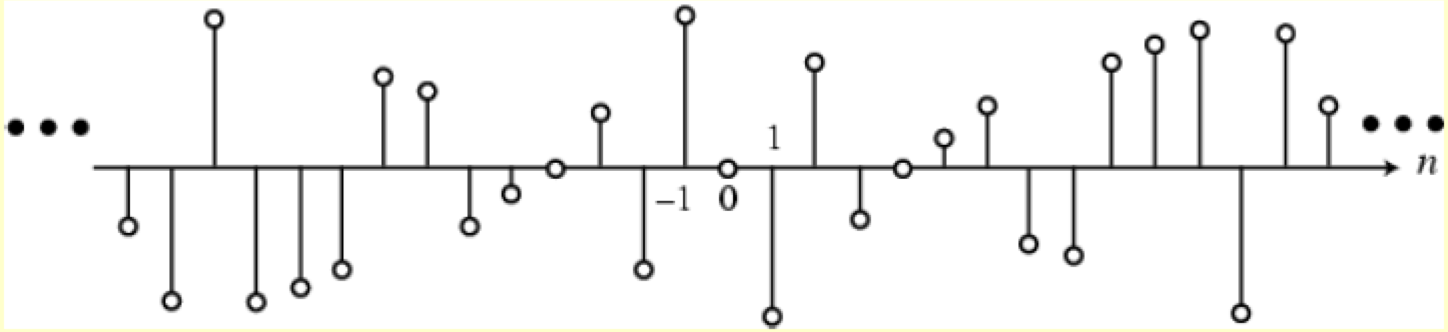

An even sequence

# § 2.3.3 Classification of Sequences

- **Conjugate-antisymmetric sequence**:

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

If $x[n]$ is real, then it is an **odd sequence**




An odd sequence

# § 2.3.3 Classification of Sequences

- It follows from the definition that for a conjugate-symmetric sequence $\{x[n]\}$, $x[0]$ must be a real number
- Likewise, it follows from the definition that for a conjugate anti-symmetric sequence $\{y[n]\}$, $y[0]$ must be an imaginary number
- From the above, it also follows that for an odd sequence $\{w[n]\}$, $w[0] = 0$

## § 2.3.3 Classification of Sequences

- Any complex sequence can be expressed as a sum of its conjugate-symmetric part and its conjugate-antisymmetric part:

$$x[n] = x_{cs}[n] + x_{ca}[n]$$

where

$$x_{cs}[n] = \frac{1}{2}\left(x[n] + x^*[-n]\right)$$

$$x_{ca}[n] = \frac{1}{2}\left(x[n] - x^*[-n]\right)$$

# § 2.3.3 Classification of Sequences

- Any real sequence can be expressed as a sum of its even part and its odd part:

$$x[n] = x_{ev}[n] + x_{od}[n]$$

where

$$x_{ev}[n] = \tfrac{1}{2}\left(x[n] + x[-n]\right)$$

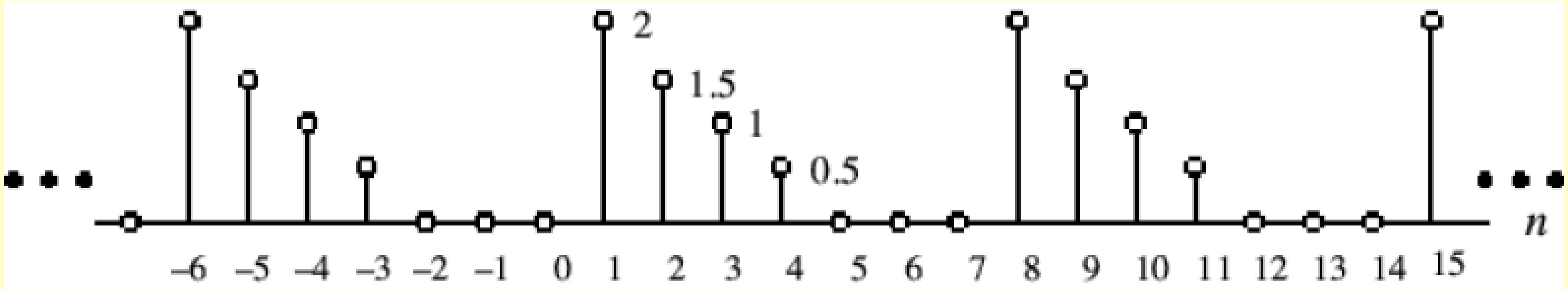
$$x_{od}[n] = \tfrac{1}{2}\left(x[n] - x[-n]\right)$$

# § 2.3.3 Classification of Sequences

- A sequence $\tilde{x}[n]$ satisfying $\tilde{x}[n] = \tilde{x}[n + kN]$ is called a **periodic sequence** with a **period** $N$ where $N$ is a positive integer and $k$ is any integer
- Smallest value of $N$ satisfying $\tilde{x}[n] = \tilde{x}[n + kN]$ is called the **fundamental period**

# § 2.3.3 Classification of Sequences

- <u>Example</u> -




- A sequence not satisfying the periodicity condition is called an **aperiodic sequence**

# § 2.3.3 Classification of Sequences

- Total **energy** of a sequence $x[n]$ is defined by

$$\mathcal{E}_{\mathrm{x}} = \sum_{n=-\infty}^{\infty} \left| x[n] \right|^2$$

- The **average power** of an aperiodic sequence is defined by

$$P_{\mathrm{x}} = \lim_{K\to\infty} \frac{1}{2K+1} \sum_{n=-K}^{K} \left| x[n] \right|^2$$

# § 2.3.3 Classification of Sequences

- An infinite energy signal with finite average power is called a **power signal**

  Example - A periodic sequence which has a finite average power but infinite energy

- A finite energy signal with zero average power is called an **energy signal**

  Example - A finite-length sequence which has finite energy but zero average power

# § 2.3.3 Classification of Sequences

- A sequence $x[n]$ is said to be **bounded** if

$$|x[n]| \leq B_x < \infty$$

- <u>Example</u> - The sequence $x[n] = \cos 0.3\pi n$ is a bounded sequence as

$$|x[n]| = |\cos 0.3\pi n| \leq 1$$

# § 2.3.3 Classification of Sequences

- A sequence $x[n]$ is said to be **absolutely summable** if

$$\sum_{n=-\infty}^{\infty} \left| x[n] \right| < \infty$$

- Example - The sequence

$$y[n] = \begin{cases} 0.3^n, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

is an absolutely summable sequence as

$$\sum_{n=0}^{\infty} \left| 0.3^n \right| = \frac{1}{1-0.3} = 1.42857 < \infty$$

## § 2.3.3 Classification of Sequences

- A sequence $x[n]$ is said to be **square-summable** if

$$\sum_{n=-\infty}^{\infty} \left| x[n] \right|^2 < \infty$$

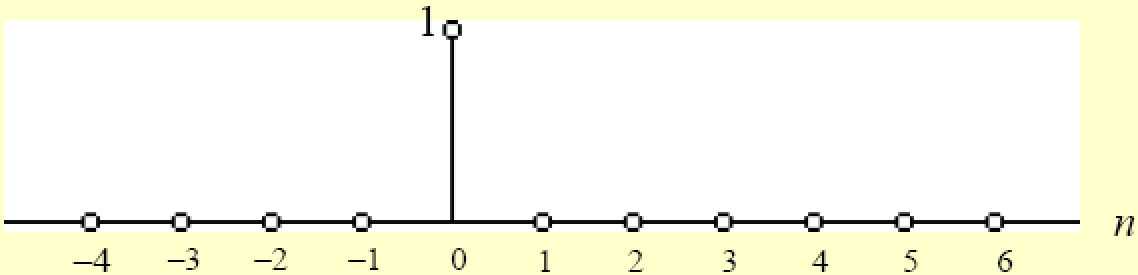
- Example - The sequence

$$h[n] = \frac{\sin 0.4n}{\pi n}$$

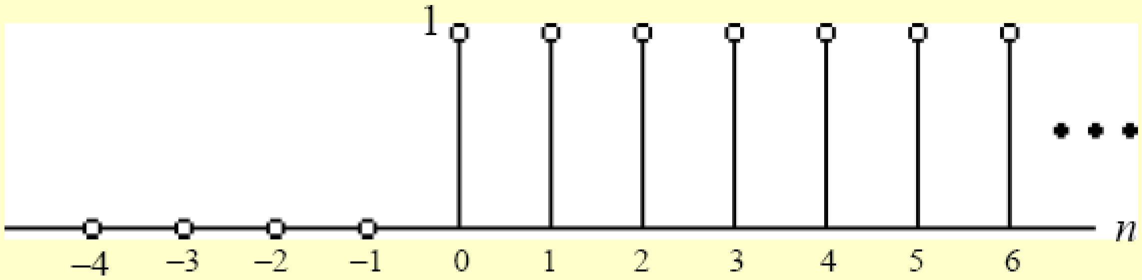
is square-summable but not absolutely summable

# § 2.4 Typical Sequences and Sequence Representation

- **Unit sample sequence** - $\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$




- **Unit step sequence** - $\mu[n] = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$



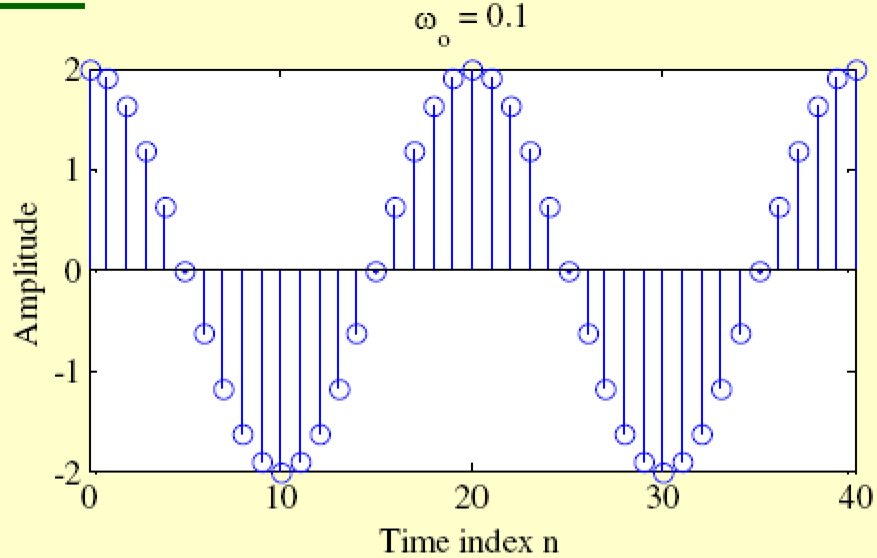
# § 2.4 Typical Sequences

- **Real sinusoidal sequence** -

$$x[n] = A\cos(\omega_o n + \phi)$$

where $A$ is the **amplitude**, $\omega_o$ is the **angular frequency**, and $\phi$ is the **phase** of $x[n]$

<u>Example</u> -



# § 2.4 Typical Sequences

- **Exponential sequence** -

$$x[n] = A\alpha^n, \quad -\infty < n < \infty$$

**where $A$ and $\alpha$ are real or complex numbers**

**If we write** $\alpha = e^{(\sigma_o + j\omega_o)}, \; A = |A| e^{j\phi},$

**then we can express**

$$x[n] = |A| e^{j\phi} e^{(\sigma_o + j\omega_o)n} = x_{re}[n] + j\, x_{im}[n],$$

**where**

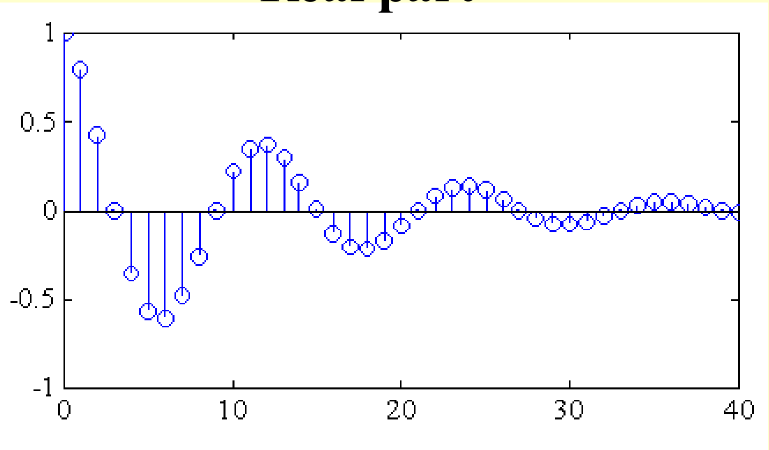
$$x_{re}[n] = |A| e^{\sigma_o n} \cos(\omega_o n + \phi)$$

$$x_{im}[n] = |A| e^{\sigma_o n} \sin(\omega_o n + \phi)$$

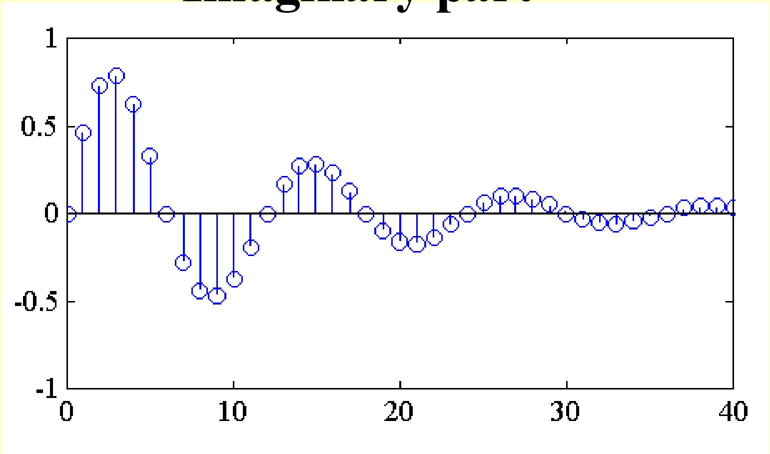
# § 2.4 Typical Sequences

- $x_{re}[n]$ and $x_{im}[n]$ of a complex exponential sequence are real sinusoidal sequences with constant ($\sigma_0$=0), growing ($\sigma_0$>0) , and decaying ($\sigma_0$<0) amplitudes for n > 0

**Real part**




**Imaginary part**



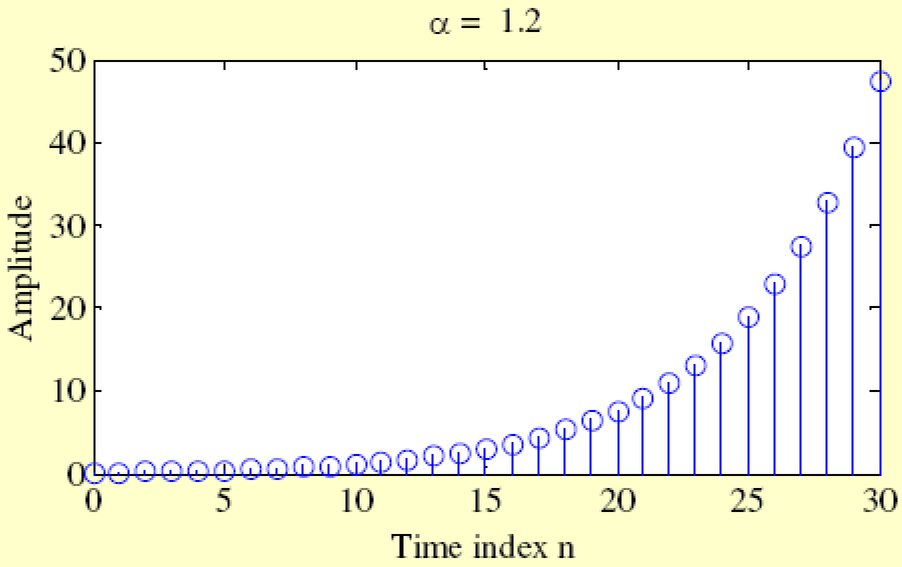

$$x[n] = \exp(-\tfrac{1}{12} + j\tfrac{\pi}{6})n$$

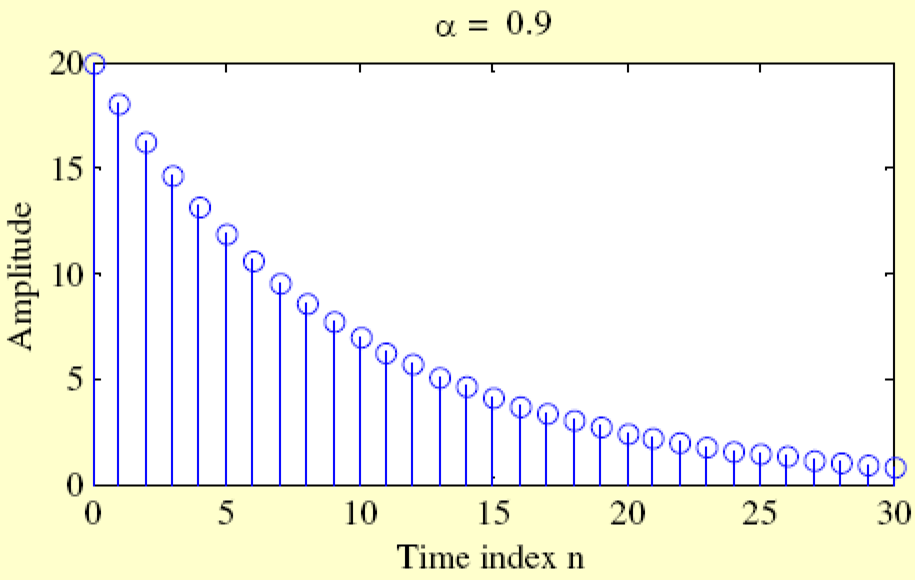
# § 2.4 Typical Sequences

- **Real exponential sequence -**

$$x[n] = A\alpha^n, \; -\infty < n < \infty$$

where $A$ and $\alpha$ are real numbers



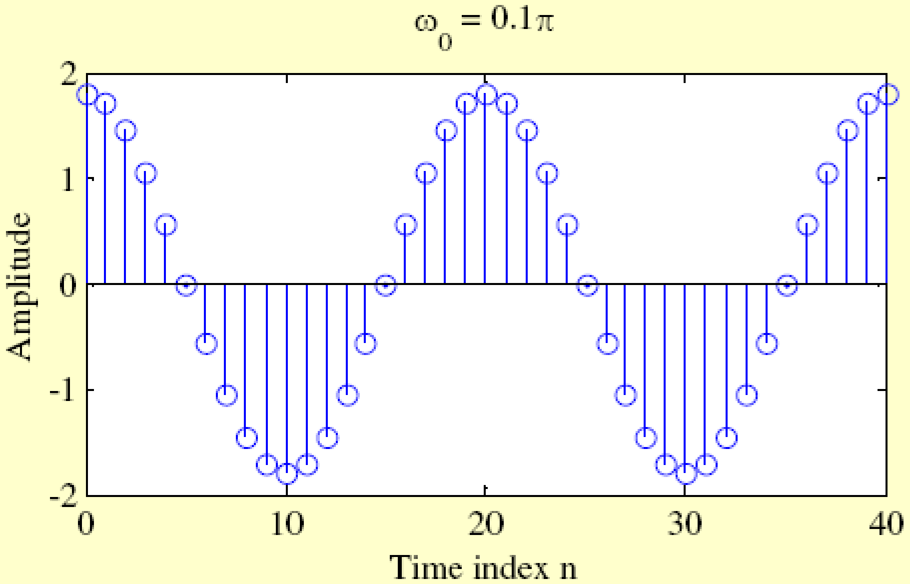
# § 2.4 Typical Sequences

- Sinusoidal sequence $A\cos(\omega_o n + \phi)$ and complex exponential sequence $B\exp(j\omega_o n)$ are periodic sequences of period $N$ if $\omega_o N = 2\pi r$ where $N$ and $r$ are positive integers
- Smallest value of $N$ satisfying $\omega_o N = 2\pi r$ is the **fundamental period** of the sequence

# § 2.4 Typical Sequences

- If $2\pi/\omega_o$ is a noninteger rational number, then the period will be a multiple of $2\pi/\omega_o$
- Otherwise, the sequence is **aperiodic**
- Example - $x[n] = \sin(\sqrt{3}n + \phi)$ is an aperiodic sequence

# § 2.4 Typical Sequences




- Here $\omega_o = 0.1\pi$
- Hence $N = \dfrac{2\pi r}{0.1\pi} = 20$ for $r = 1$

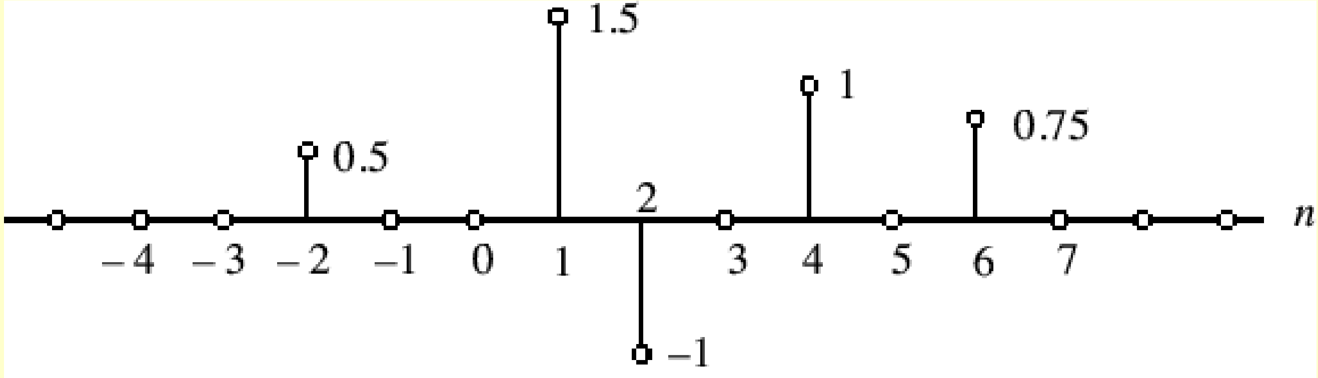
# § 2.4 Typical Sequences

- Property 1 - Consider $x[n] = \exp(j\omega_1 n)$ and $y[n] = \exp(j\omega_2 n)$ with $0 \le \omega_1 < \pi$ and $2\pi k \le \omega_2 < 2\pi(k+1)$ where $k$ is any positive integer
- If $\omega_2 = \omega_1 + 2\pi k$, then $x[n] = y[n]$
- Thus, $x[n]$ and $y[n]$ are indistinguishable

# § 2.4 Typical Sequences

## 2.4.3 Representation of an Arbirary Sequences

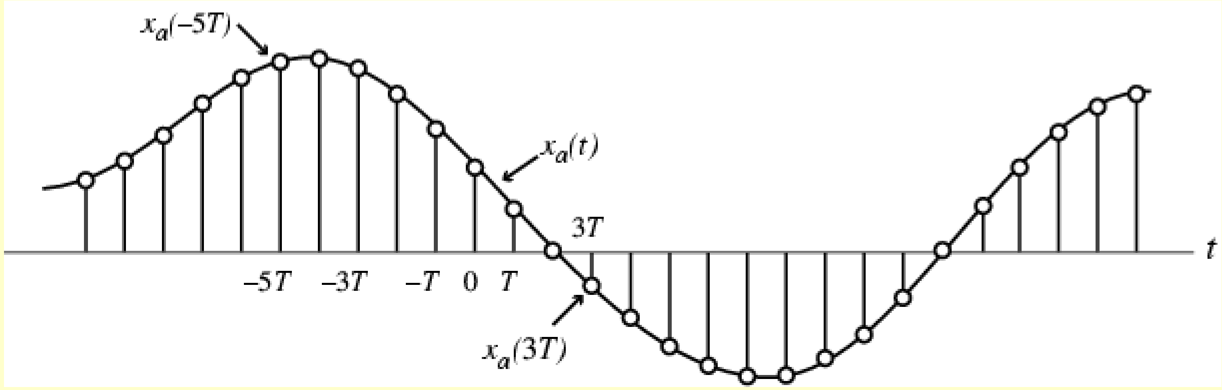
- An arbitrary sequence can be represented in the time-domain as a weighted sum of some basic sequence and its delayed (advanced) versions




$$x[n] = 0.5\delta[n+2] + 1.5\delta[n-1] - \delta[n-2]$$
$$+\delta[n-4] + 0.75\delta[n-6]$$

# § 2.5 The Sampling Process

- Often, a discrete-time sequence $x[n]$ is developed by uniformly sampling a continuous-time signal $x_a(t)$ as indicated below




- The relation between the two signals is

$$x[n] = x_a(t)\big|_{t=nT} = x_a(nT),\ n = \ldots, -2, -1, 0, 1, 2, \ldots$$

# § 2.5 The Sampling Process

- Time variable $t$ of $x_a(t)$ is related to the time variable $n$ of $x[n]$ only at discrete-time instants $t_n$ given by

$$t_n = nT = \frac{n}{F_T} = \frac{2\pi n}{\Omega_T}$$

with $F_T = 1/T$ denoting the sampling frequency and

$\Omega_T = 2\pi F_T$ denoting the sampling angular frequency

# § 2.5 The Sampling Process

- Consider the continuous-time signal

$$x_a(t) = A\cos(2\pi f_o t + \phi) = A\cos(\Omega_o t + \phi)$$

- The corresponding discrete-time signal is

$$x[n] = A\cos(\Omega_o nT + \phi) = A\cos(\frac{2\pi\Omega_o}{\Omega_T} n + \phi)$$

$$= A\cos(\omega_o n + \phi)$$

where $\omega_o = 2\pi\Omega_o / \Omega_T = \Omega_o T$

is the normalized digital angular frequency of $x[n]$

# § 2.5 The Sampling Process

- If the unit of sampling period T is in seconds
- The unit of normalized digital angular frequency $\omega_0$ is radians/sample
- The unit of normalized analog angular frequency $\Omega_0$ is radians/second
- The unit of analog frequency $f_0$ is hertz (Hz)

# § 2.5 The Sampling Process

- The three continuous-time signals

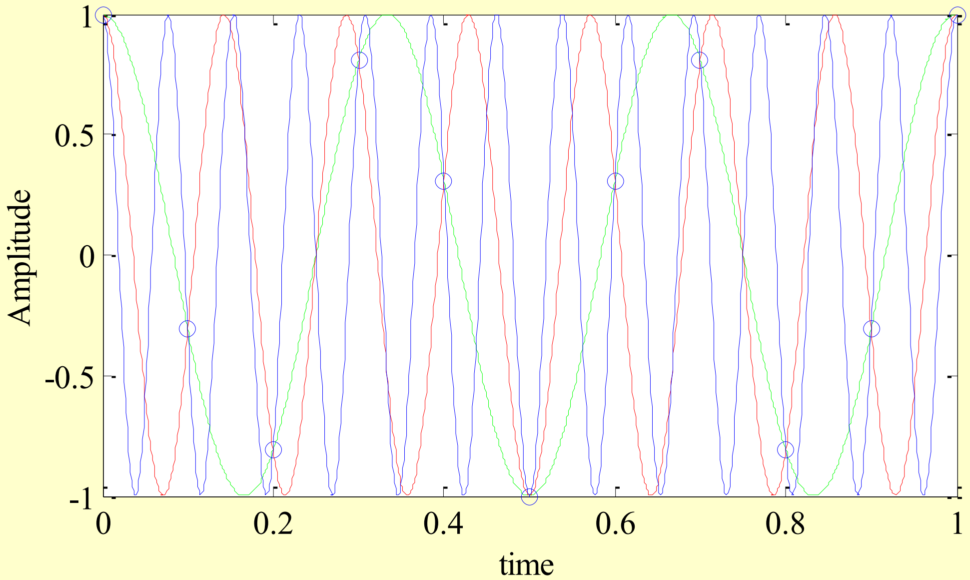
$$g_1(t) = \cos(6\pi t)$$
$$g_2(t) = \cos(14\pi t)$$
$$g_3(t) = \cos(26\pi t)$$

of frequencies 3 Hz, 7 Hz, and 13 Hz, are sampled at a sampling rate of 10 Hz, i.e. with $T = 0.1$ sec. generating the three sequences

$$g_1[n] = \cos(0.6\pi n) \qquad g_2[n] = \cos(1.4\pi n)$$
$$g_3[n] = \cos(2.6\pi n)$$

# § 2.5 The Sampling Process

- Plots of these sequences (shown with circles) and their parent time functions are shown below:




- **Note that each sequence has exactly the same sample value for any given n**

# § 2.5 The Sampling Process

- This fact can also be verified by observing that

$$g_2[n] = \cos(1.4\pi n) = \cos\big((2\pi - 0.6\pi)n\big) = \cos(0.6\pi n)$$

$$g_3[n] = \cos(2.6\pi n) = \cos\big((2\pi + 0.6\pi)n\big) = \cos(0.6\pi n)$$

- As a result, all three sequences are identical and it is difficult to associate a unique continuous-time function with each of these sequences

# § 2.5 The Sampling Process

- The above phenomenon of a continuous-time signal of higher frequency acquiring the identity of a sinusoidal sequence of lower frequency after sampling is called **aliasing**

# § 2.5 The Sampling Process

- Example - Determine the discrete-time signal $v[n]$ obtained by uniformly sampling at a sampling rate of 200 Hz the continuous-time signal

$$v_a(t) = 6\cos(60\pi t) + 3\sin(300\pi t) + 2\cos(340\pi t) + 4\cos(500\pi t) + 10\sin(660\pi t)$$

- Note: $v_a(t)$ is composed of 5 sinusoidal signals of frequencies 30 Hz, 150 Hz, 170 Hz, 250 Hz and 330 Hz

# § 2.5 The Sampling Process

- The sampling period is $T = \frac{1}{200} = 0.005$ sec
- The generated discrete-time signal $v[n]$ is thus given by

$$
\begin{aligned}
v[n] &= 6\cos(0.3\pi n) + 3\sin(1.5\pi n) + 2\cos(1.7\pi n) \\
&\qquad + 4\cos(2.5\pi n) + 10\sin(3.3\pi n) \\
&= 6\cos(0.3\pi n) + 3\sin\big((2\pi - 0.5\pi)n\big) + 2\cos\big((2\pi - 0.3\pi)n\big) \\
&\qquad + 4\cos\big((2\pi + 0.5\pi)n\big) + 10\sin\big((4\pi - 0.7\pi)n\big) \\
&= 6\cos(0.3\pi n) - 3\sin(0.5\pi n) + 2\cos(0.3\pi n) + 4\cos(0.5\pi n) \\
&\qquad - 10\sin(0.7\pi n) \\
&= 8\cos(0.3\pi n) + 5\cos(0.5\pi n + 0.6435) - 10\sin(0.7\pi n)
\end{aligned}
$$

# § 2.5 The Sampling Process

$$v[n] = 8\cos(0.3\pi n) + 5\cos(0.5\pi n + 0.6435) - 10\sin(0.7\pi n)$$

- Note: $v[n]$ is composed of 3 discrete-time sinusoidal signals of normalized angular frequencies: $0.3\pi$, $0.5\pi$, and $0.7\pi$
- Note: An identical discrete-time signal is also generated by uniformly sampling at a 200-Hz sampling rate the following continuous-time signals:

$$w_a(t) = 8\cos(60\pi t) + 5\cos(100\pi t + 0.6435) - 10\sin(140\pi t)$$

$$g_a(t) = 2\cos(60\pi t) + 4\cos(100\pi t) + 10\sin(260\pi t) + 6\cos(460\pi t) + 3\sin(700\pi t)$$

# § 2.5 The Sampling Process

- Recall $\omega_o = \dfrac{2\pi\Omega_o}{\Omega_T}$

- Thus if $\Omega_T > 2\Omega_o$, then the corresponding normalized digital angular frequency $\omega_o$ of the discrete-time signal obtained by sampling the parent continuous-time sinusoidal signal will be in the range $-\pi < \omega < \pi$
- $\Rightarrow$ No aliasing

# Homework

- **Problems**

  2.3, 2.4(b), 2.21(a),(c), 2.22, 2.26, 2.41, 2.43（只须做x[n] ）,2.51

- **MATLAB Exercise**

  M 2.1, M 2.2, M 2.4, M 2.6