Chapter 2

Discrete-time signals in the time domain

§ 2.1 Discrete-Time Signals

- Discrete-time signal in its most basic form is defined at equally spaced discrete values of time, with the signal amplitude at these discrete times taking a continuous value
- It is represented as sequences of numbers, called samples(样本)
- 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

§ 2.1.1 Time-Domain Representation

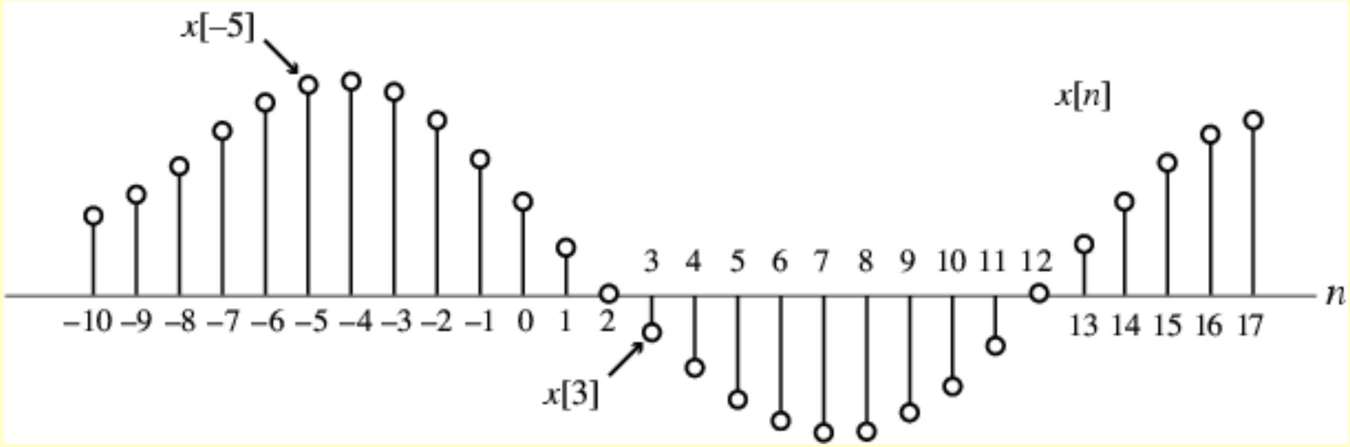
- Discrete-time signal represented by {x[n]}
- Discrete-time signal may also be written as a sequence of numbers inside braces(括号):

$$\{x[n]\}=\{\ldots,-0.2,\underset{\uparrow}{2.2},1.1,0.2,-3.7,2.9,\ldots\}$$

- The arrow(箭头) is placed under the sample at time index n = 0
- In the above, x[-1]= -0.2, x[0]=2.2, x[1]=1.1, etc.

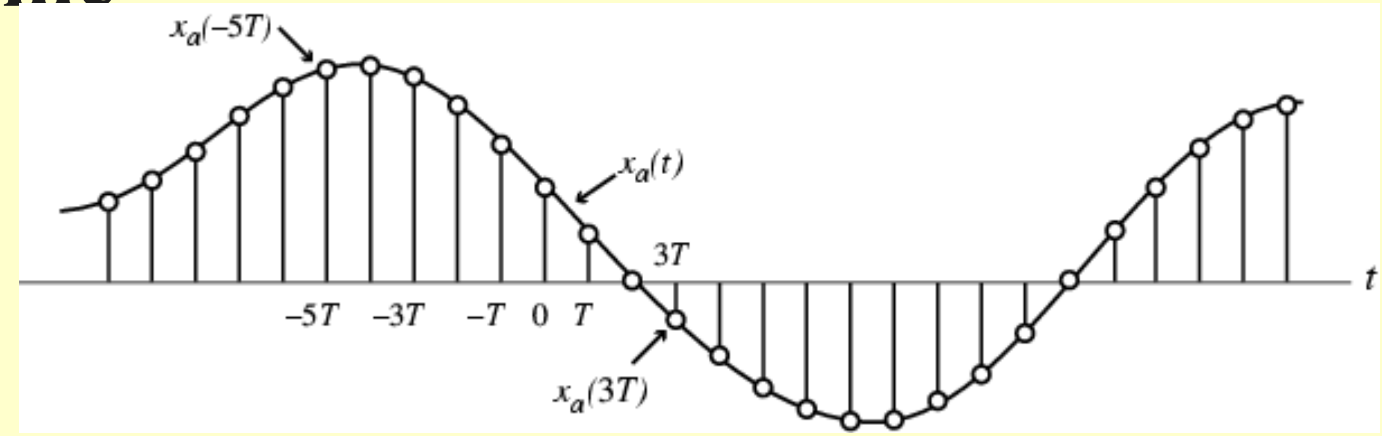
§ 2.1.1 Time-Domain Representation

- **Graphical(图形的) representation of a discrete-time signal with real-valued samples is as shown below:**



§ 2.1.1 Time-Domain Representation

- **In some applications, a discrete-time sequence $\{x[n]\}$ may be generated by periodically sampling a continuous-time signal $x_a(t)$ at uniform(统一的) intervals(间隔) of time**



§ 2.1.1 Time-Domain Representation

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

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

- **The spacing T between two consecutive(连续的) samples is called the sampling interval(抽样间隔) or sampling period(抽样周期)**

- **Reciprocal(倒数) of sampling interval T, denoted as F_T , is called the sampling frequency(抽样频率):**

  $$F_T=1/T$$

§ 2.1.1 Time-Domain Representation

- **Unit of sampling frequency is cycles per second, or hertz (Hz), if T is in seconds**
- **Whether or not the sequence $\{x[n]\}$ has been obtained by sampling, the quantity x[n] is called the n-th sample of the sequence**
- **{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**

§ 2.1.1 Time-Domain Representation

- 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 $x[n]$
- The **complex conjugate(复共轭)** sequence of $\{x[n]\}$ is given by $\{x^*[n]\}=\{x_{re}[n]\} - j\{x_{im}[n]\}$
- Often the braces are ignored to denote a sequence if there is no ambiguity

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

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

§ 2.1.1 Time-Domain Representation

- Example -

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

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

- That is,

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

§ 2.1.1 Time-Domain Representation

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

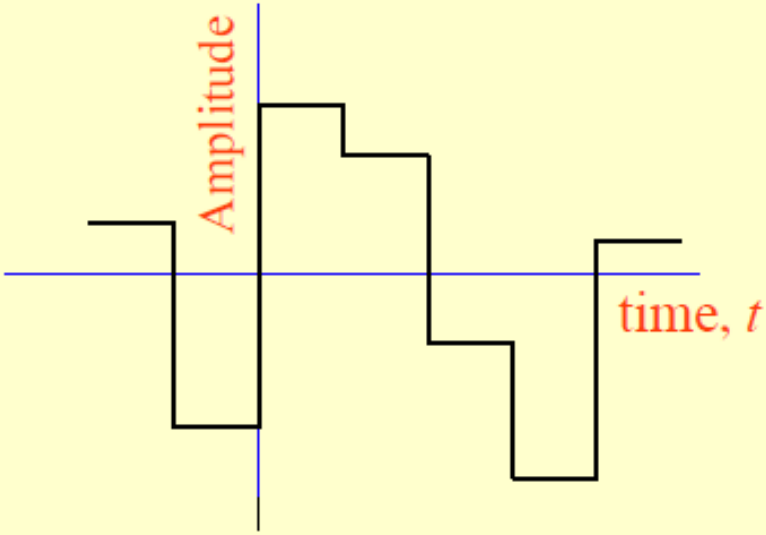
  - Sampled-data signals in which samples are continuous-valued

  - Digital signals in which samples are discrete-valued

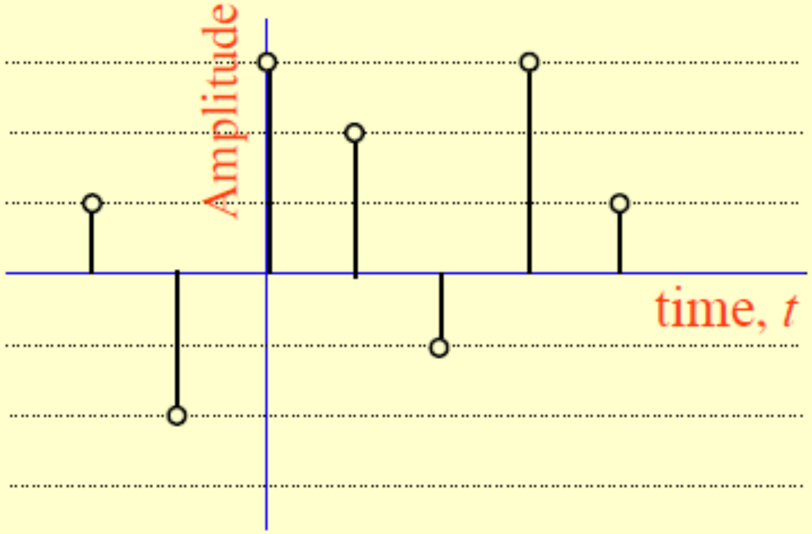
- **Signals in a practical digital signal processing system are digital signals obtained by quantizing the sample values either by rounding(舍入，凑整) or truncation(切尾)**

§ 2.1.1 Time-Domain Representation

Example -



Boxedcar signal



Digital signal

§ 2.1.1 Time-Domain Representation

Length of a discrete-time signal

- A discrete-time signal may be a **finite-length** or an **infinite-length sequence**
- Finite-length 有限范围 (also called **finite-duration** or **finite-extent**) sequence is defined 有限时宽 only for a finite time interval: $N_1 \le n \le N_2$ where $-\infty < N_1$ and $N_2 < \infty$ with $N_1 \le N_2$
- Length or duration of the above finite-length sequence is $N = N_2 - N_1 + 1$

§ 2.1.1 Time-Domain Representation

Length of a discrete-time signal

- **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

§ 2.1.1 Time-Domain Representation

Length of a discrete-time signal

- 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**

§ 2.1.1 Time-Domain Representation
Length of a discrete-time signal

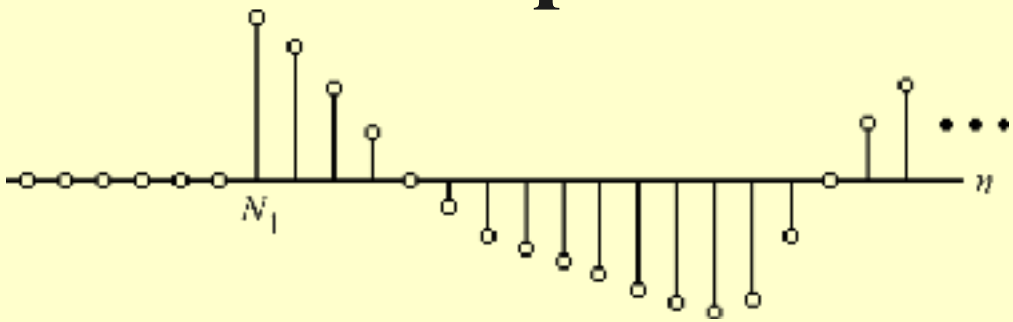
- Example -

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

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

§ 2.1.1 Time-Domain Representation
Length of a discrete-time signal

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

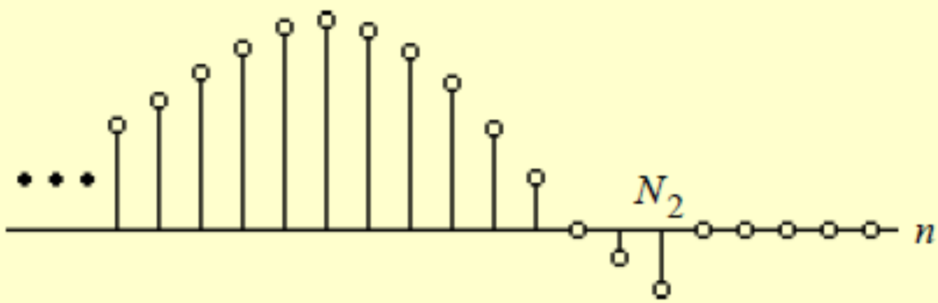


A right-sided sequence

•If $N_1 \geq 0$, a right-sided sequence is called a <u>causal(因果的)</u> sequence

§ 2.1.1 Time-Domain Representation
Length of a discrete-time signal

- 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**

§ 2.1.2 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**

$x[n] \longrightarrow$ **Discrete-time system** $\longrightarrow y[n]$

Input sequence **Output sequence**

§ 2.1.2 Operations on Sequences

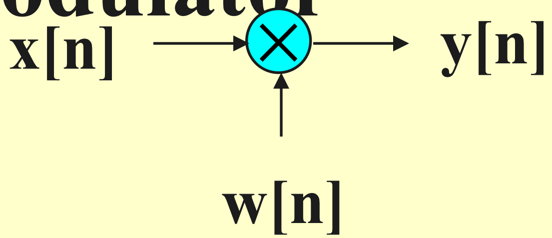
- **For example: the input may be a signal corrupted(污染) with additive noise**
- **Discrete-time system is designed to generate an output by removing the noise component from the input**
- **In most cases, the operation defining a particular discrete-time system is composed of some basic operations**

Elementary Operations

- **Product (modulation) operation:**

调制

Modulator

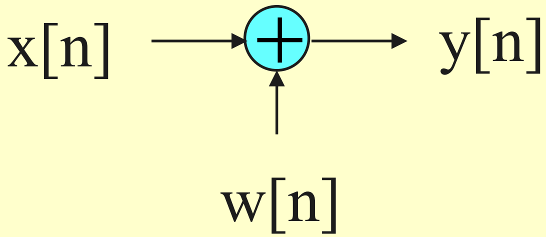


$y[n]=x[n].w[n]$

- An application is in forming a finite-length sequence from an infinite-length sequence by multiplying the latter with a finite-length sequence called an **window sequence**
- The process is called **windowing(加窗)**

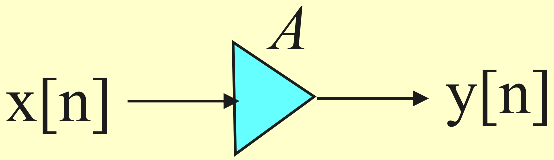
Elementary Operations

- **Addition operation:**
  - **Adder**



$$\mathbf{y[n]=x[n]+w[n]}$$

- **Multiplication operation**
  - **Multiplier**



$$\mathbf{y[n]=A.x[n]}$$

Elementary Operations

- **Time-shifting operation, where N is an integer**
- **If N > 0, it is delaying operation**

–Unit delay

$$x[n] \longrightarrow \boxed{z^{-1}} \longrightarrow y[n]$$

y[n]=x[n-1]

If N < 0, it is an advance operation

-Unit advance

$$x[n] \longrightarrow \boxed{z} \longrightarrow y[n]$$

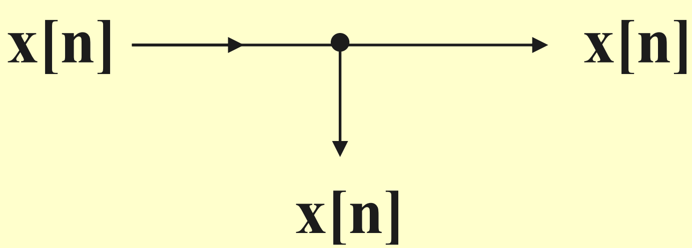
y[n]=x[n+1]

Elementary Operations

- Time-reversal (<u>folding</u>)(时间反转，反折) operation:

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

- Branching(分支) operation: Used to provide multiple copies of a sequence



Elementary Operations

- **Example** - Consider the two following sequences of length 5 defined for $0 \leq n \leq 4$:

  $\{a[n]\}=\{3\ 4\ 6\ {-9}\ 0\}$

  $\{b[n]\}=\{2\ {-1}\ 4\ 5\ {-3}\}$

- New sequences generated from the above two sequences by applying the basic operations are as follows:

Elementary operations

{c[n]}={a[n].b[n]}={6 –4 24 –45 0}

{d[n]}={a[n]+b[n]}={5 3 10 –4 -3}

{e[n]}=(3/2){a[n]}={4.5 6 9 –13.5 0}

- As pointed out by the above example, operations on two or more sequences can be carried out if all sequences involved are of same length and defined for the same range of the time index n

Elementary operations

- However if the sequences are not of same length, in some situations, this problem can be circumvented(避免) by appending(添加) zero-valued samples to the sequence(s) of smaller lengths to make all sequences have the same range of the time index
- Example - Consider the sequence of length 3 defined for $0 \leq n \leq 2$: $\{f[n]\}=\{-2, 1, -3\}$

Elementary Operations

- **We cannot add the length-3 sequence to the length-5 sequence {a[n]} defined earlier**
- **We therefore first append {f[n[} with 2 zero-valued samples resulting in a length-5 sequence $\{f_e[n[\}=\{-2\ 1\ –3\ 0\ 0\}$**
- **Then**

$$\{g[n]\}=\{a[n]\}+\{f[n]\}=\{1\ 5\ 3\ –9\ 0\}$$

Elementary Operations

- We cannot add the length-3 sequence $\{f[n]\}$ to the length-5 sequence $\{a[n]\}$ defined earlier
- 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

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

Elementary Operations

Ensemble Averaging 总体均值

- A very simple application of the addition operation in improving the quality of measured data corrupted by an additive random noise 污染
- In some cases, actual uncorrupted data vector **s** remains essentially the same from one measurement to next

Elementary Operations

- While the additive noise vector is random and not reproducible 可再生的
- Let $\mathbf{d}_i$ denote the noise vector corrupting the *i*-th measurement of the uncorrupted data vector $\mathbf{s}$:

$$\mathbf{x}_i = \mathbf{s} + \mathbf{d}_i$$

Elementary Operations

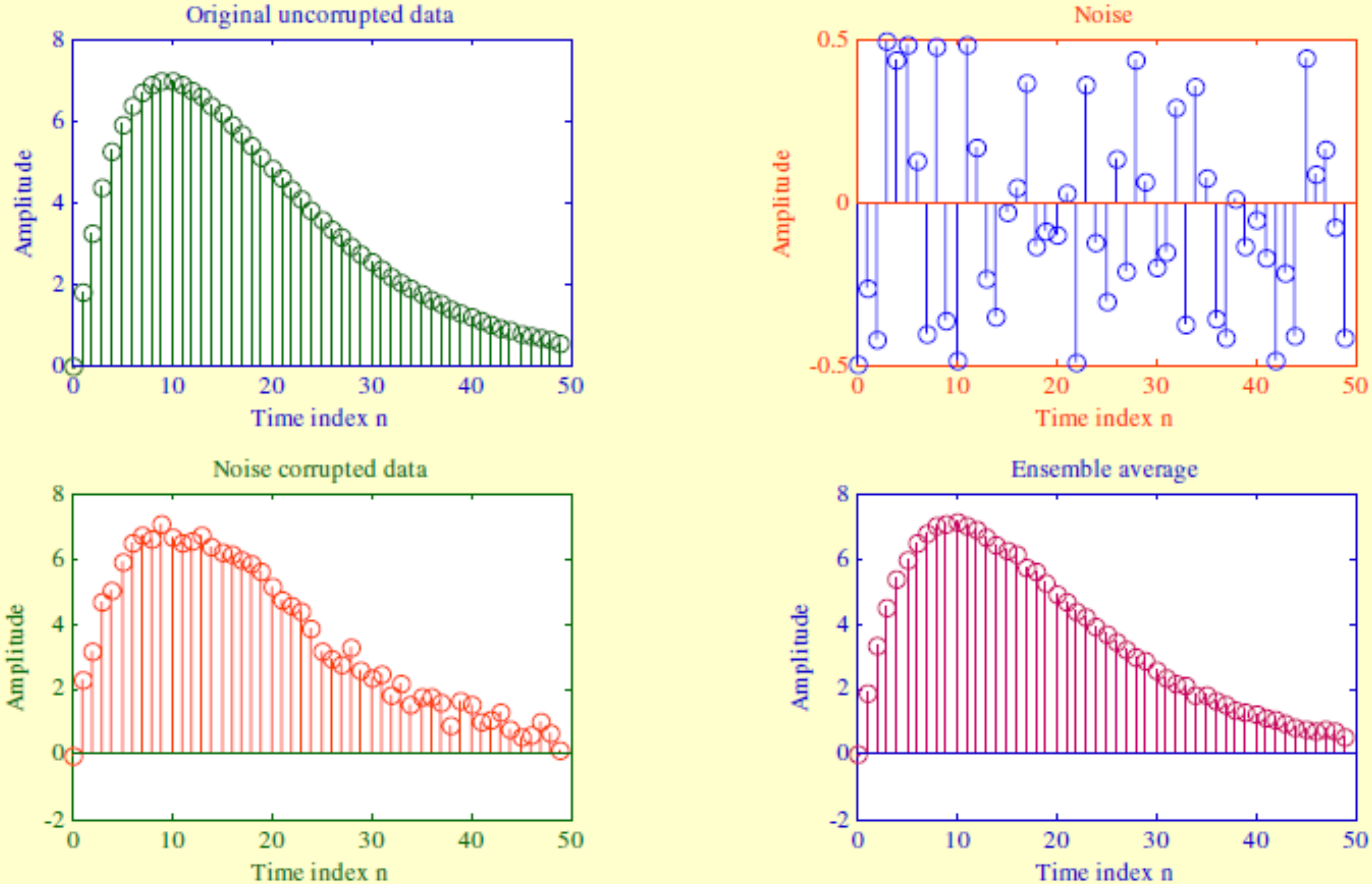
- The average data vector, called the ensemble average, obtained after K measurements is given by

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

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

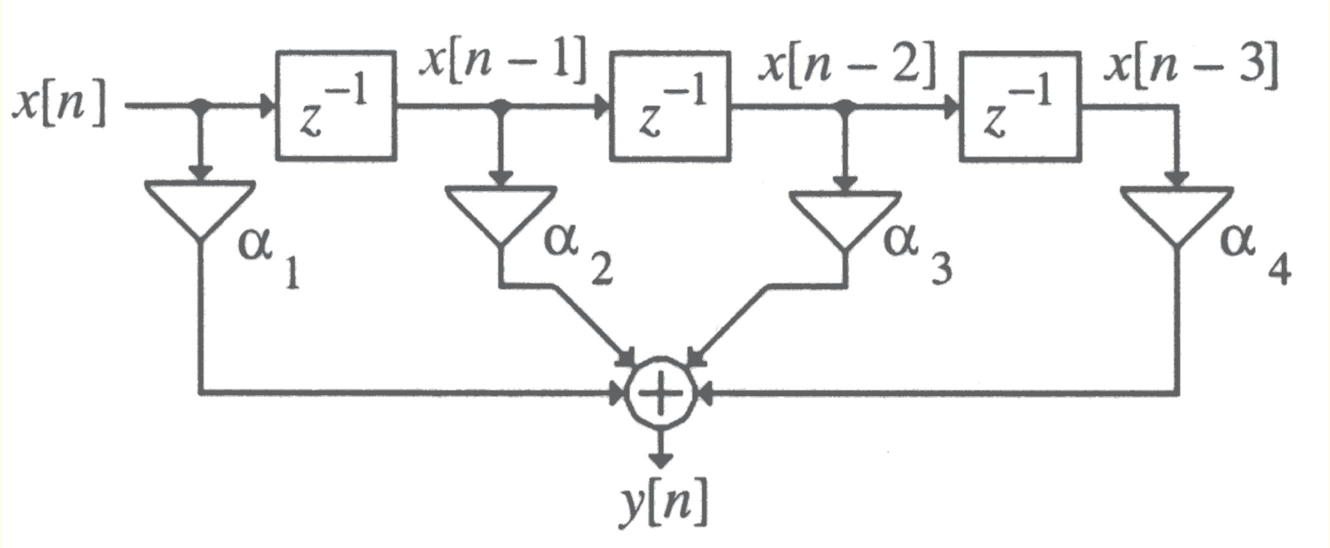
Elementary Operations

- Example



Combination of Operations

➢ **<u>Example</u> -**



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

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

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

Convolution Sum

- We illustrate the convolution operation for the following two sequences:

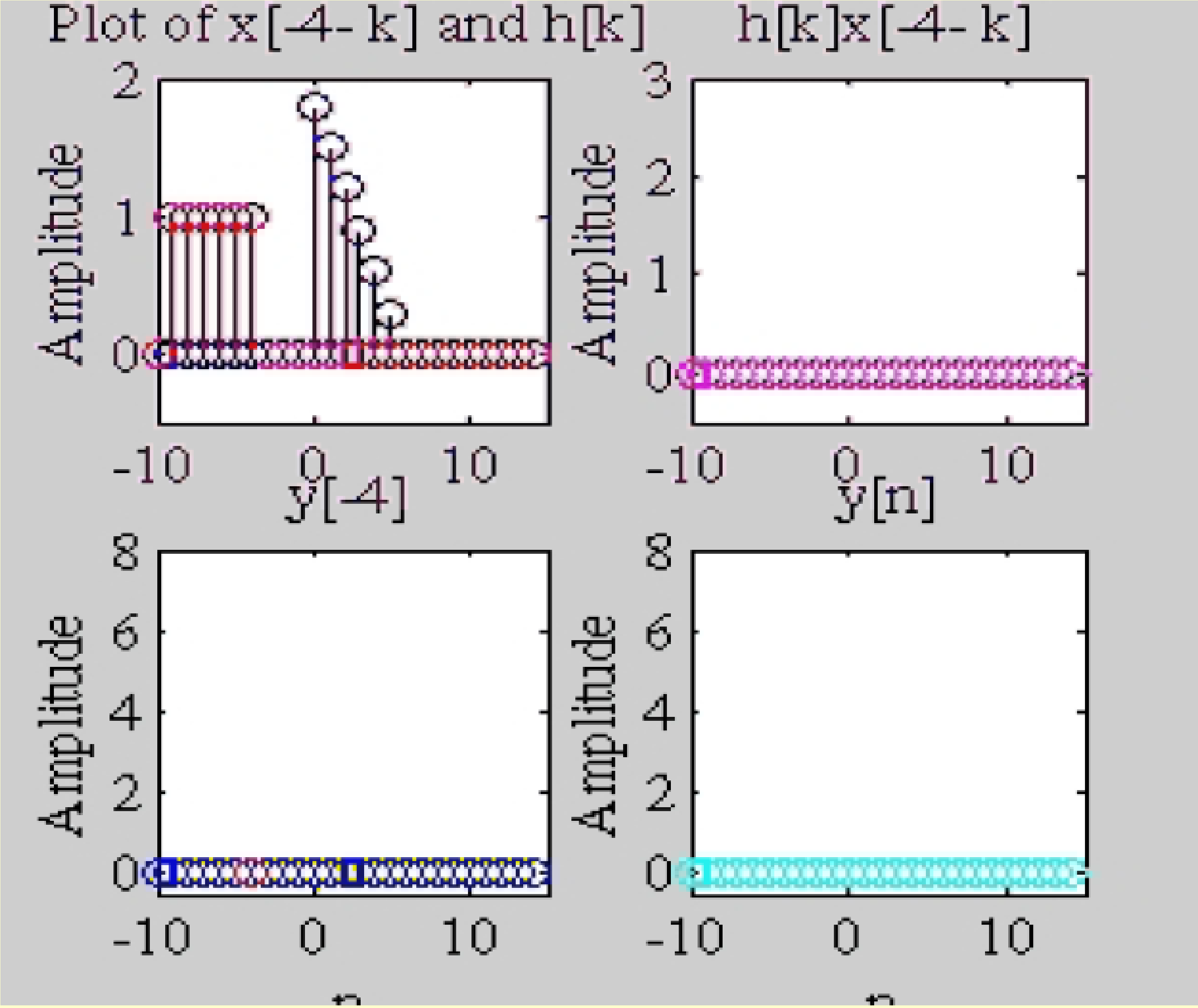
$$x[n] = \begin{cases} 1, & 0 \le n \le 5 \\ 0, & \text{otherwise} \end{cases}$$

$$h[n] = \begin{cases} 1.8 - 0.3n, & 0 \le n \le 5 \\ 0, & \text{otherwise} \end{cases}$$

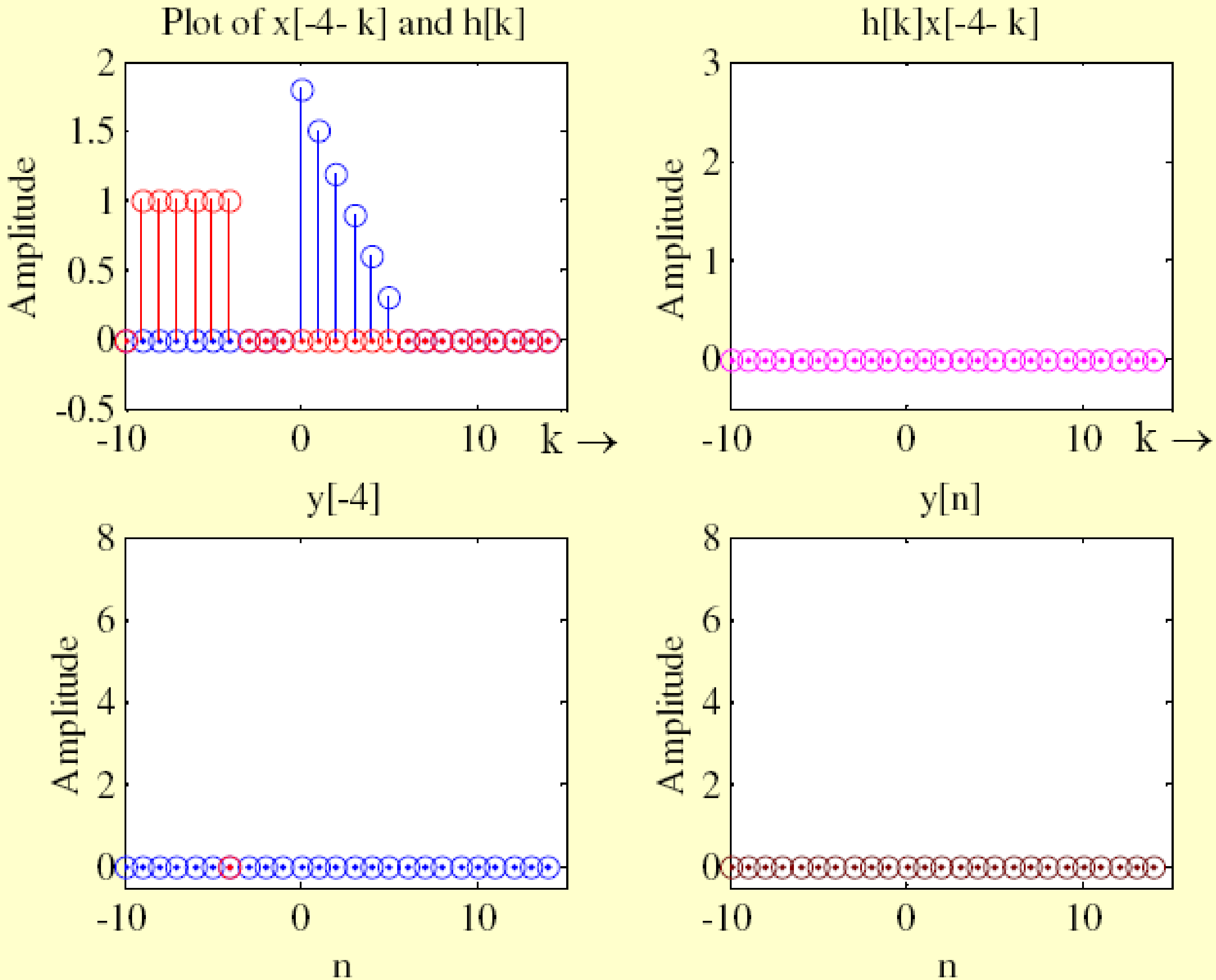
- Figures on the next several slides the steps involved in the computation of

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

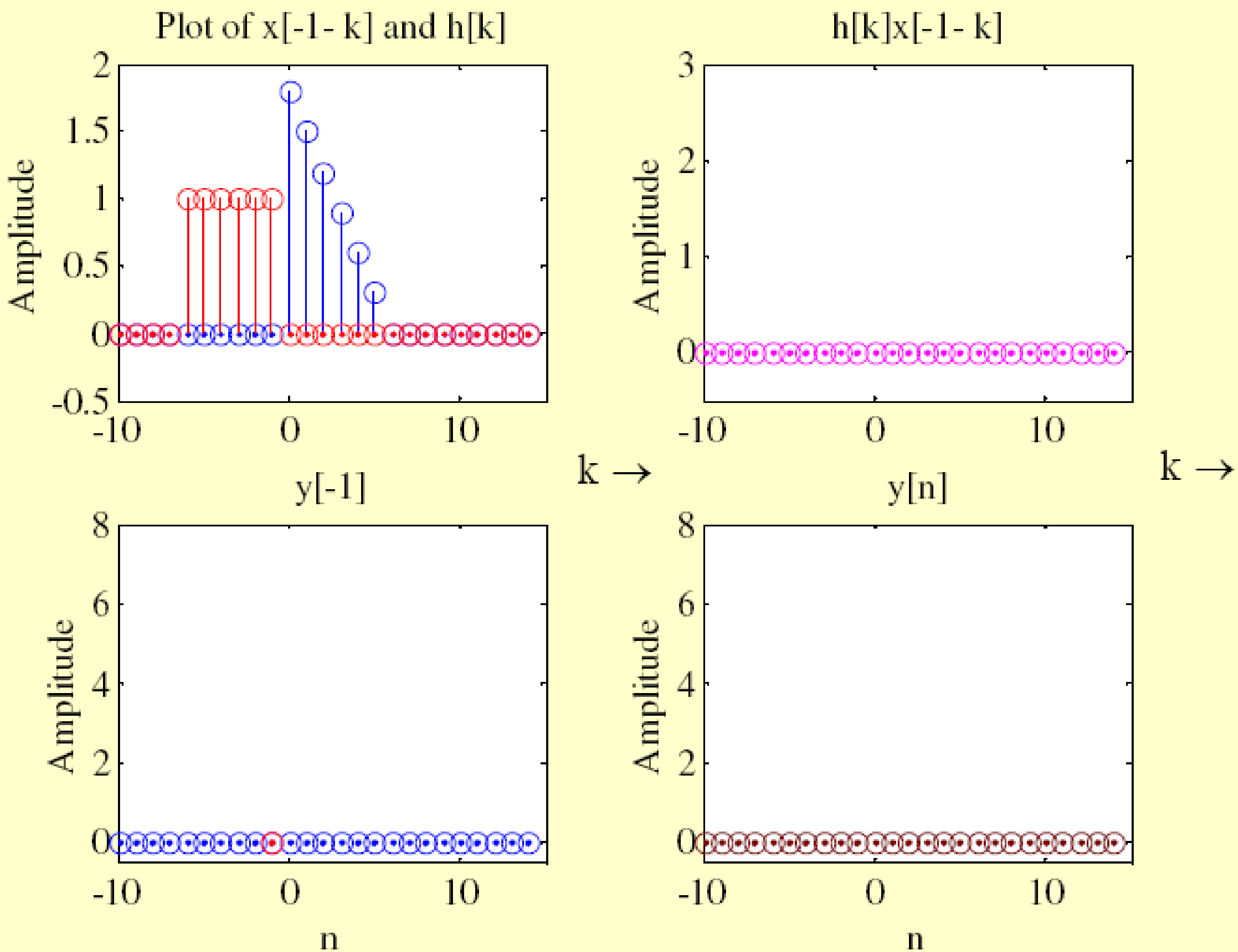
Convolution Sum



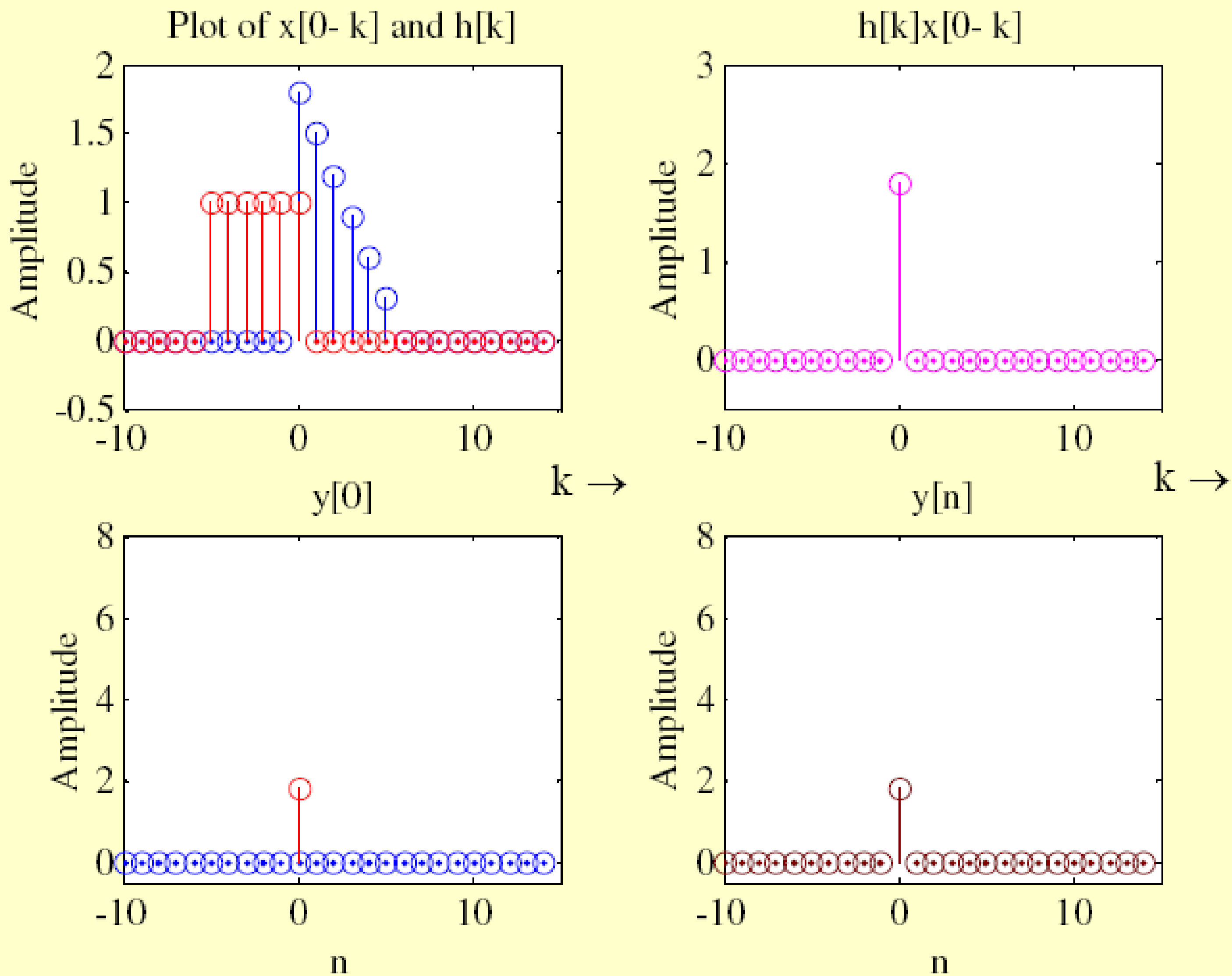
Convolution Sum



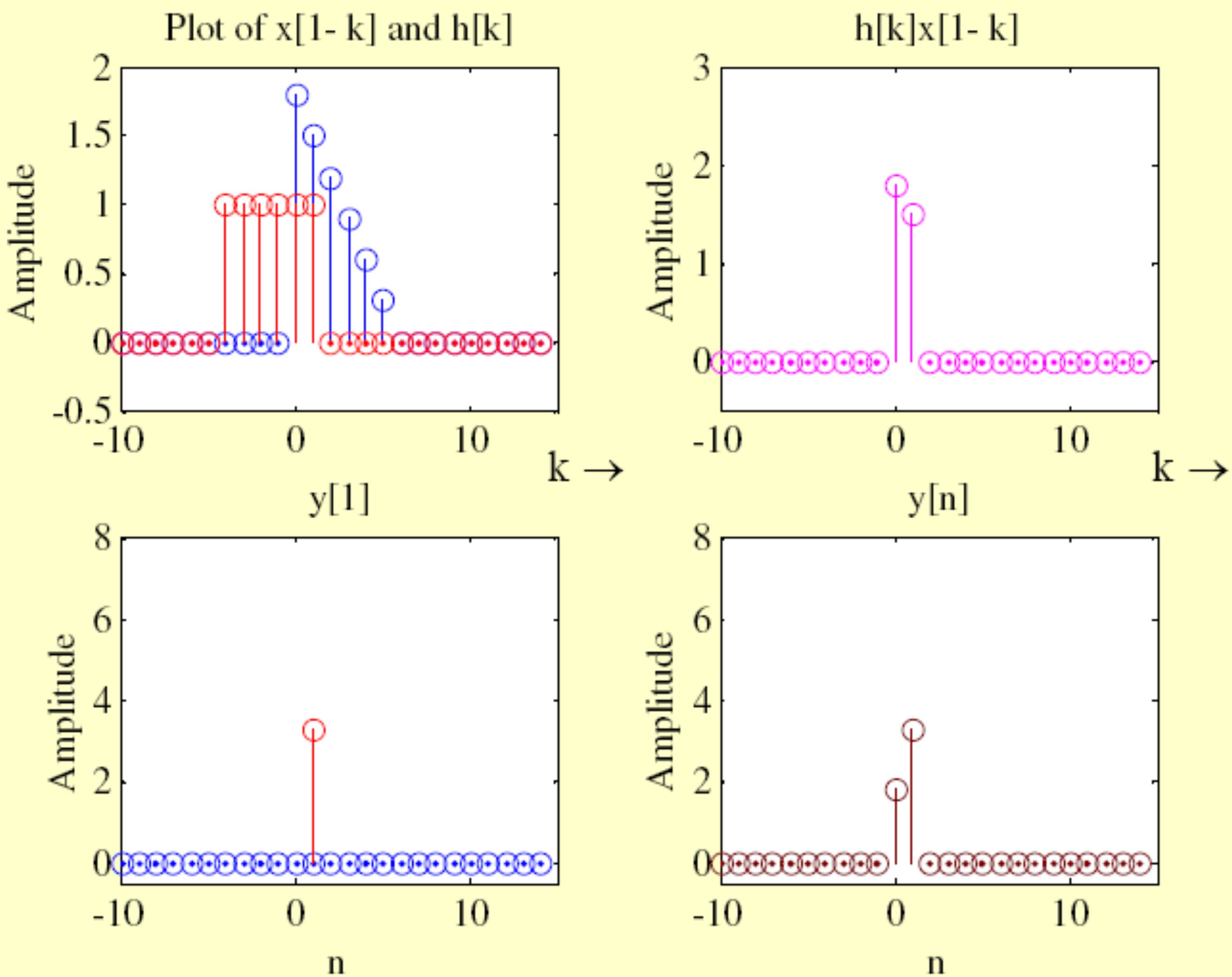
Convolution Sum



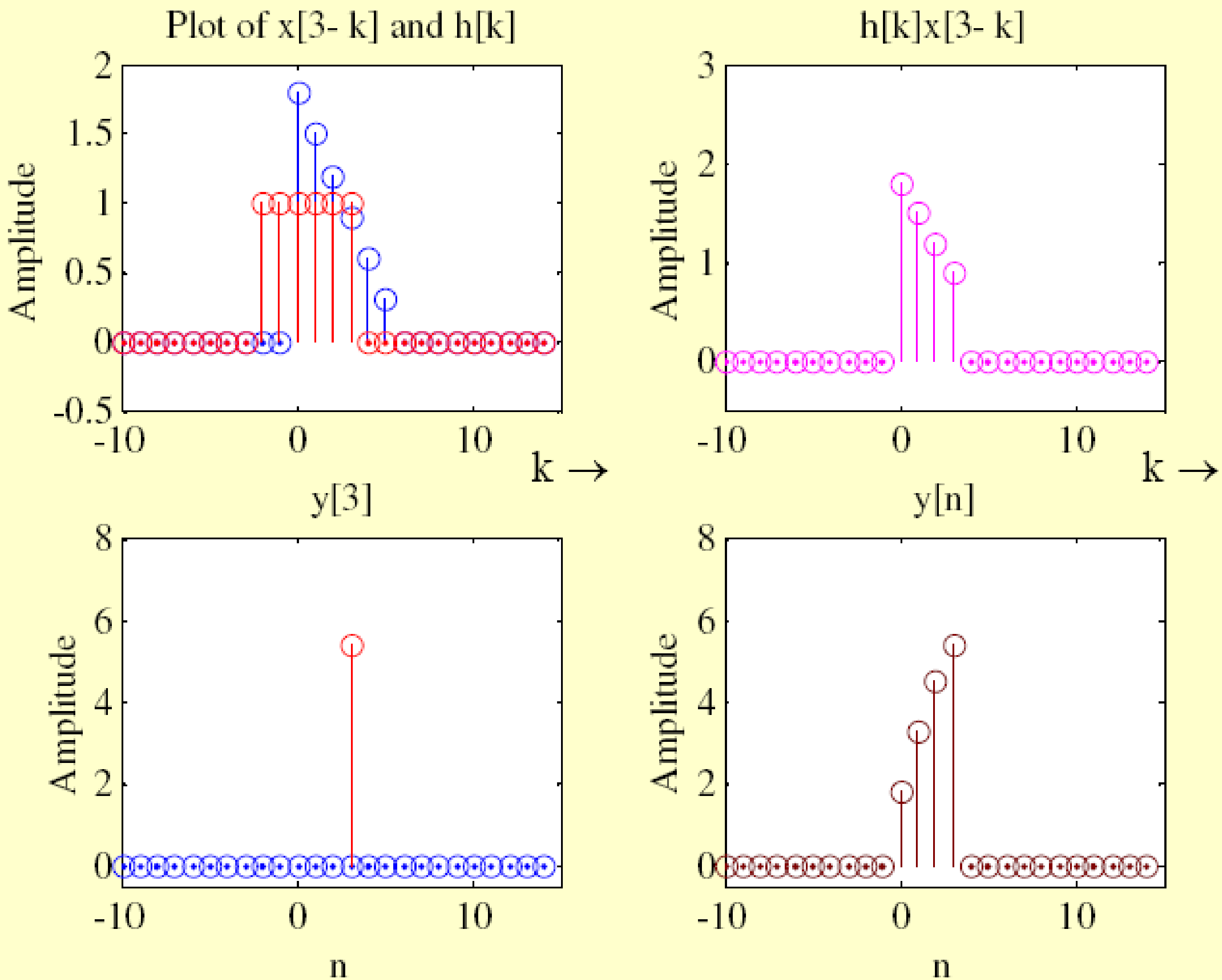
Convolution Sum



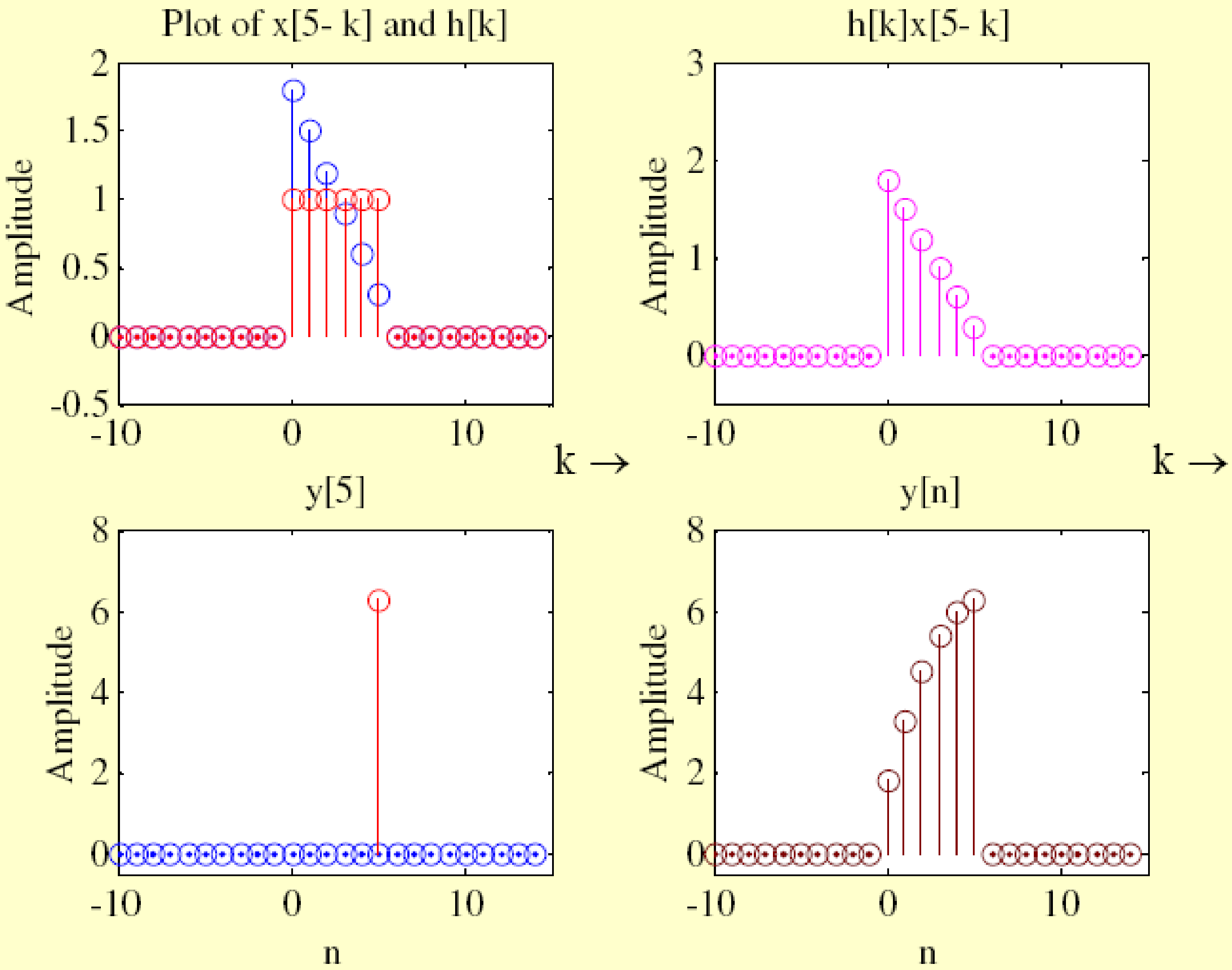
Convolution Sum



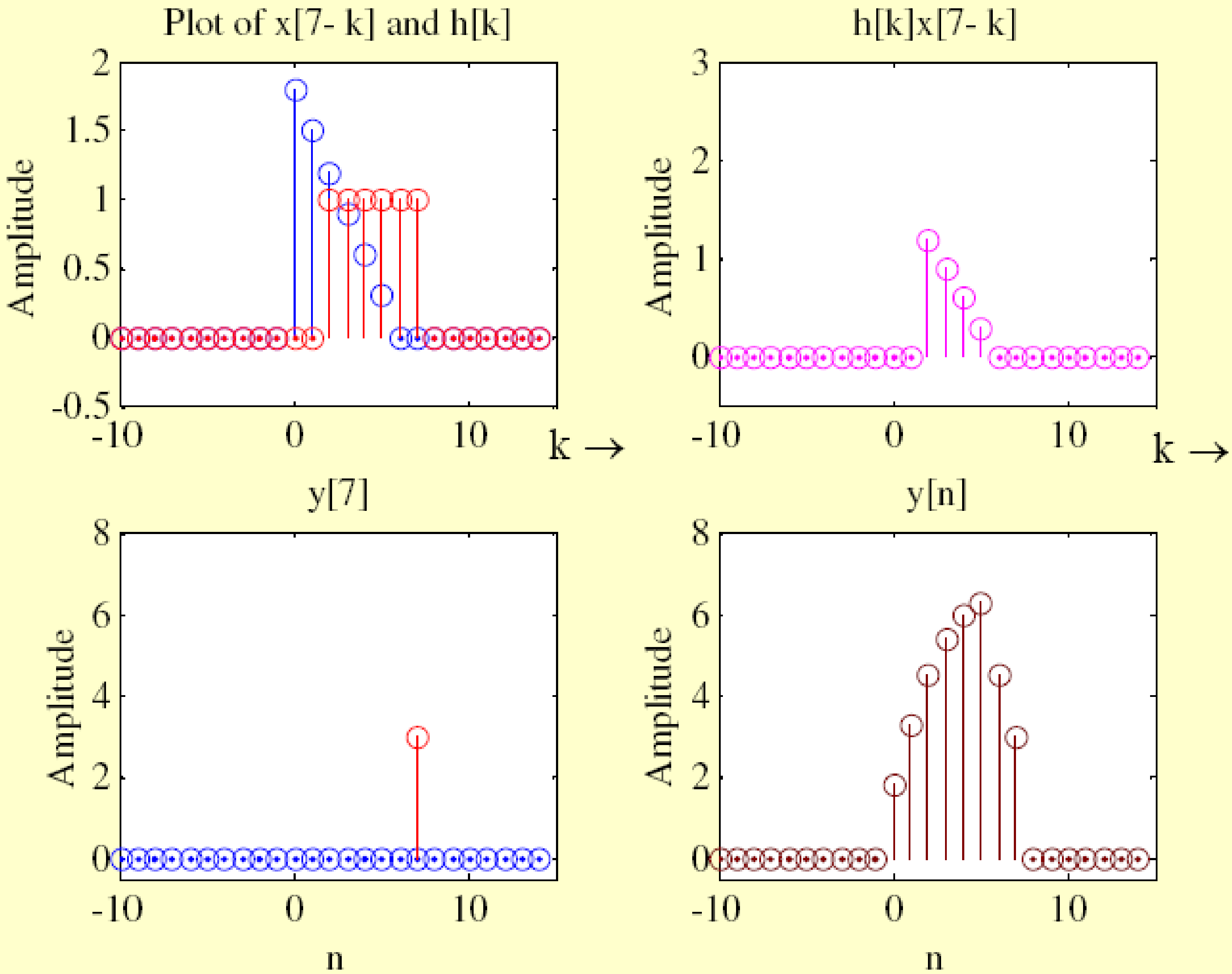
Convolution Sum



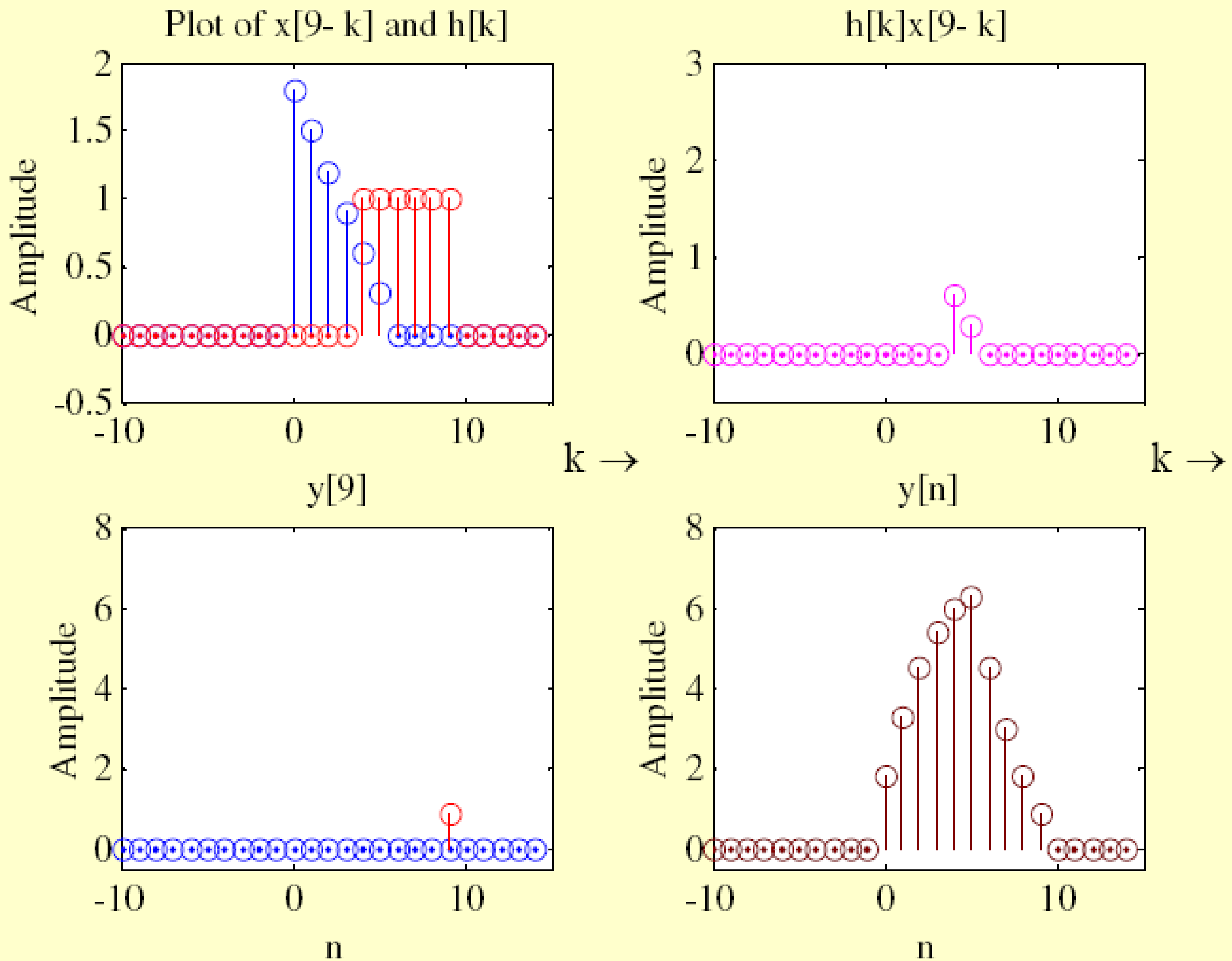
Convolution Sum



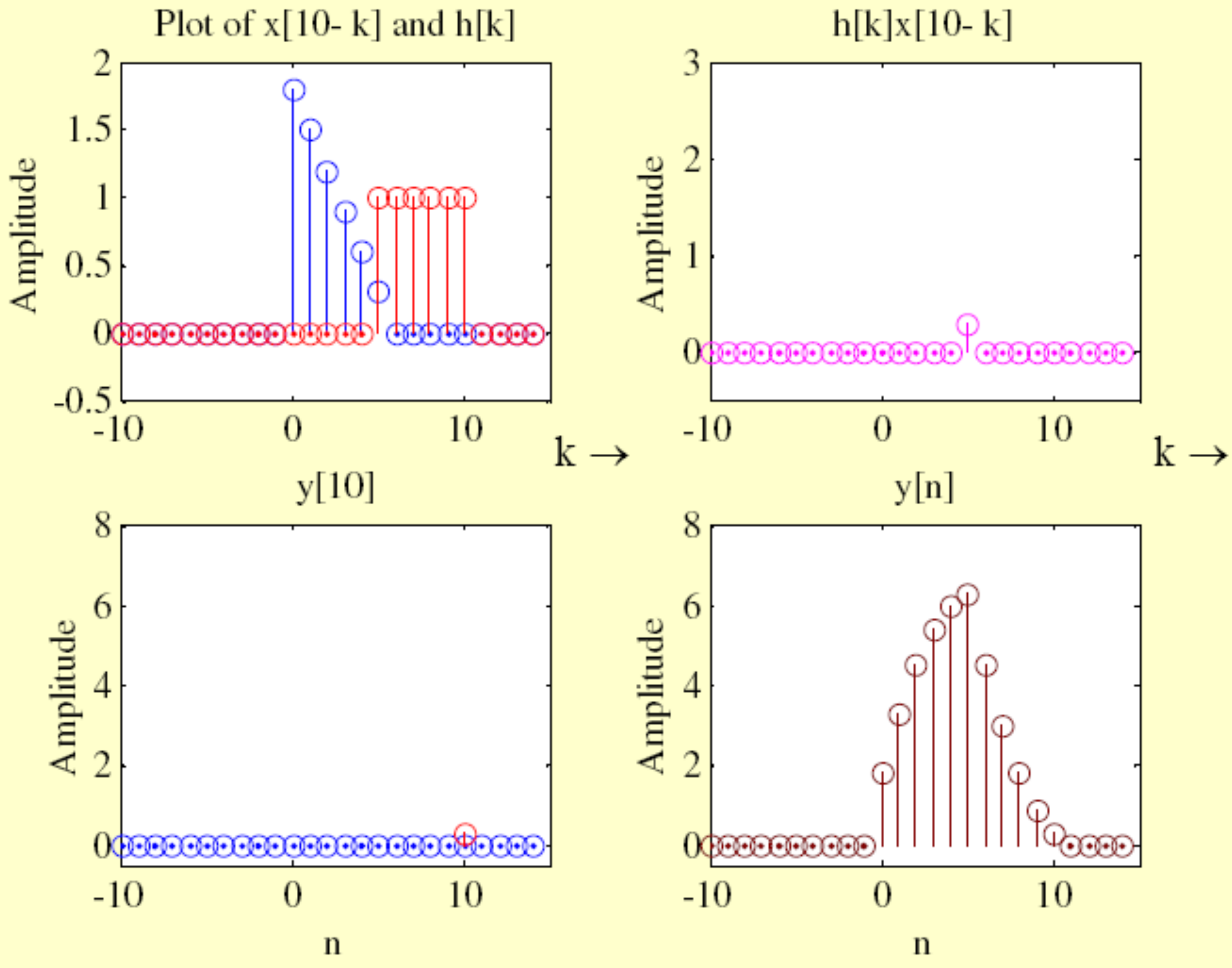
Convolution Sum



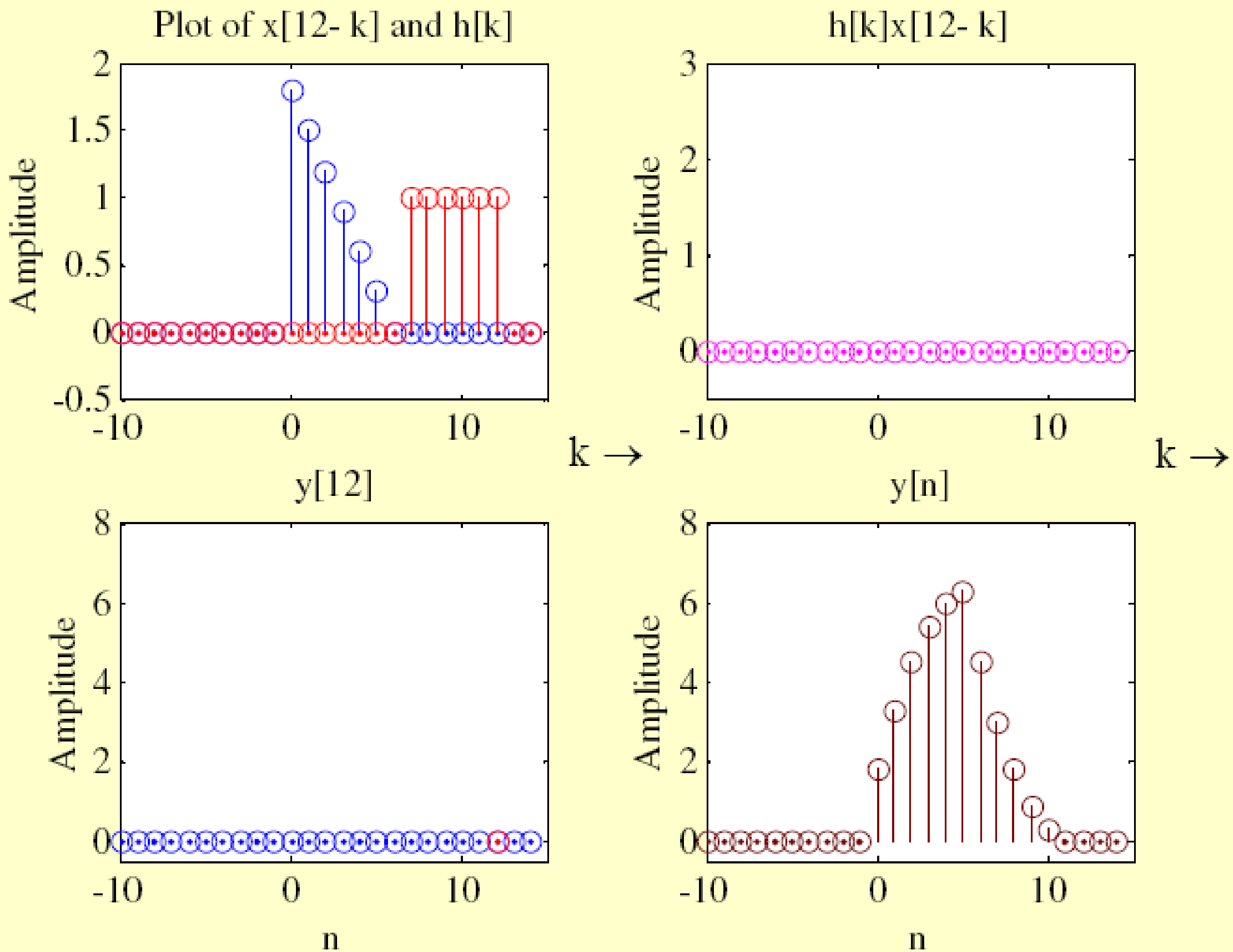
Convolution Sum



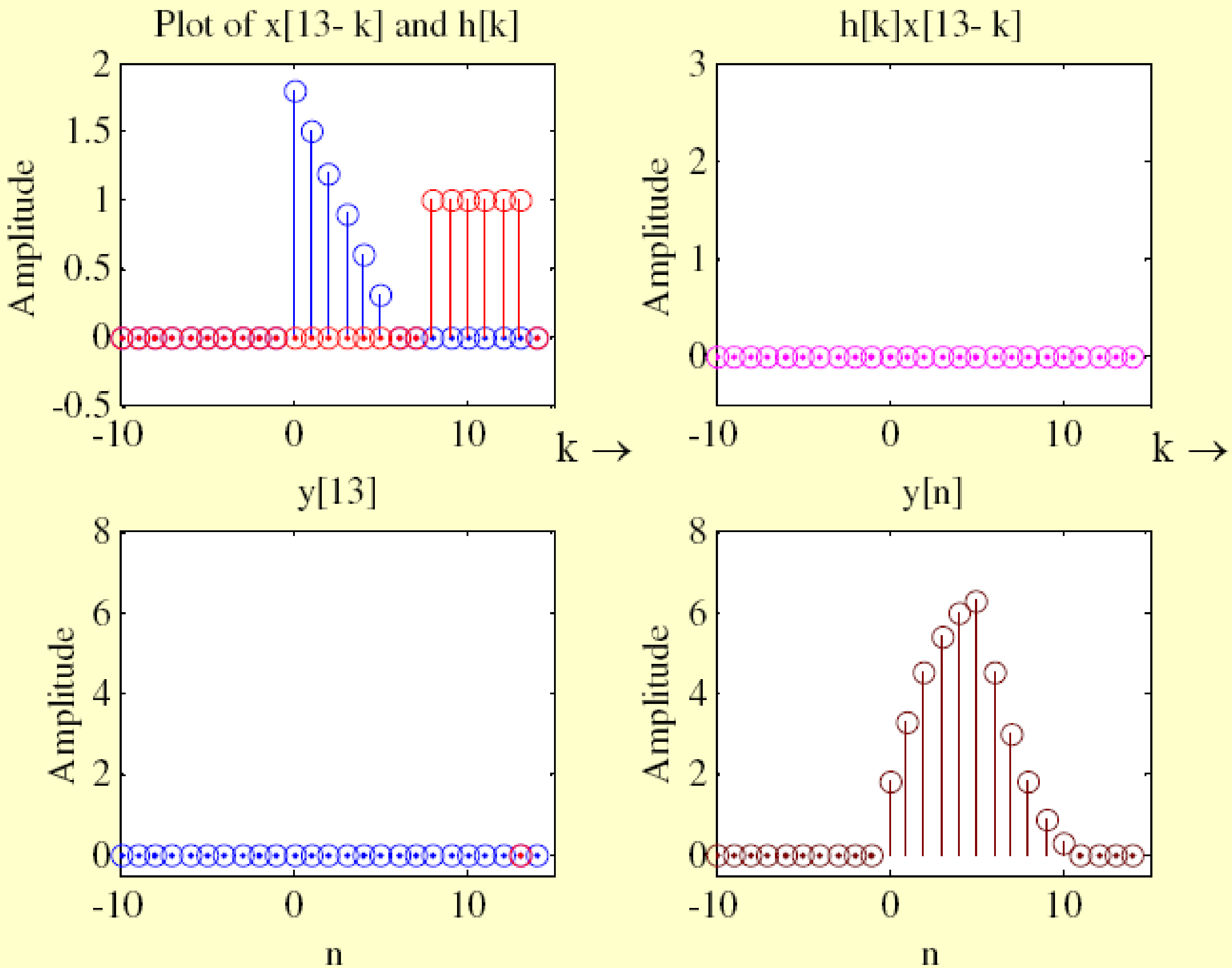
Convolution Sum



Convolution Sum



Convolution Sum



Convolution Sum

- The summation

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

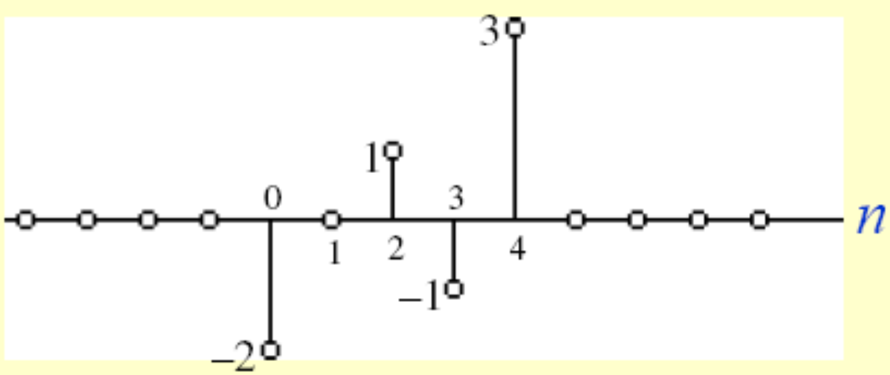
is called the **convolution sum** of the sequences $x[n]$ and $h[n]$ and represented compactly as

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

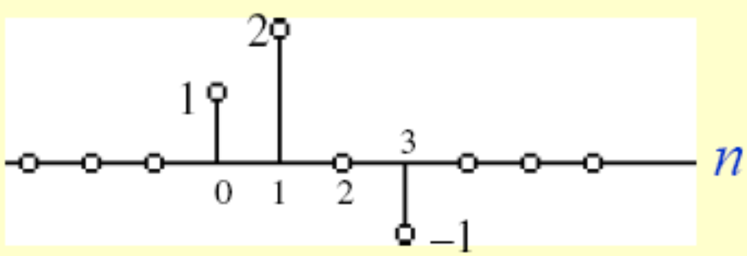
Convolution Sum

- Example - Develop the sequence $y[n]$ generated by the convolution of the sequences $x[n]$ and $h[n]$ shown below

$x[n]$



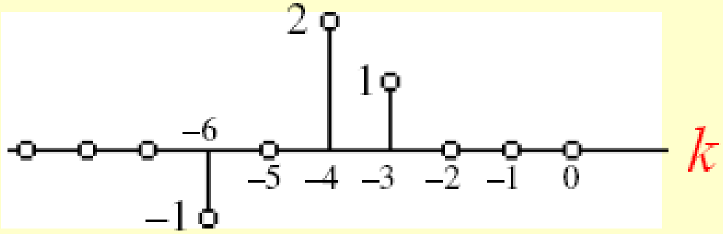
$h[n]$



Convolution Sum

- As can be seen from the shifted time-reversed version $\{h[n-k]\}$ for $n<0$, shown below for $n=-3$, for any value of the sample index k, the k-th sample of either $\{x[k]\}$ or $\{h[n-k]\}$ is zero

$h[-3-k]$

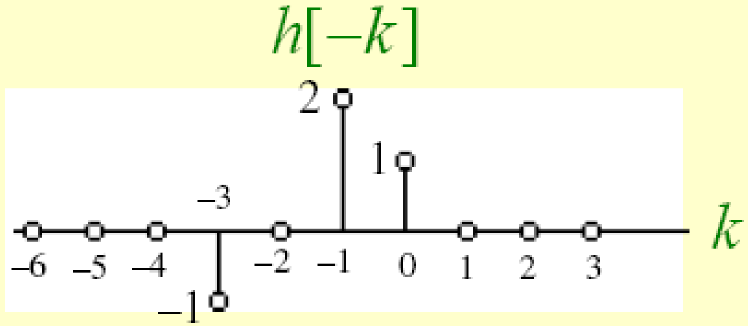


Convolution Sum

- As a result, for $n < 0$, the product of the k-th samples of $\{x[k]\}$ and $\{h[n-k]\}$ is always zero, and hence

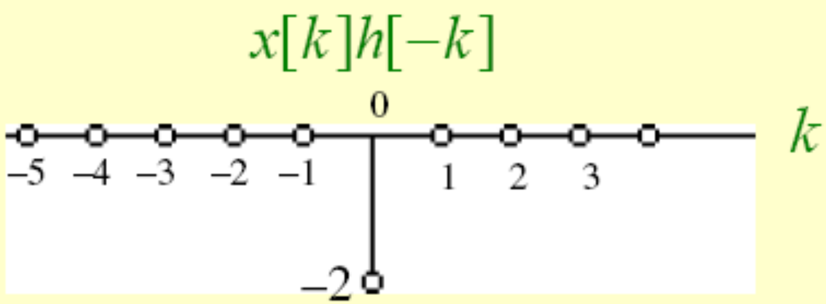
$$y[n] = 0 \quad \text{for} \;\; n < 0$$

- Consider now the computation of $y[0]$
- The sequence $\{h[-k]\}$ is shown on the right



Convolution Sum

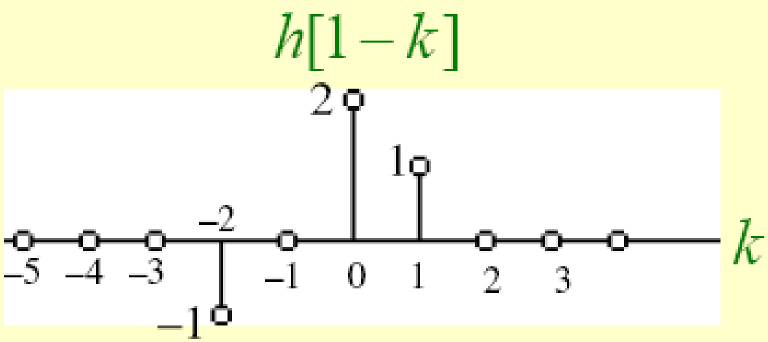
- The product sequence $\{x[k]h[-k]\}$ is plotted below which has a single nonzero sample $x[0]h[0]$ for $k = 0$

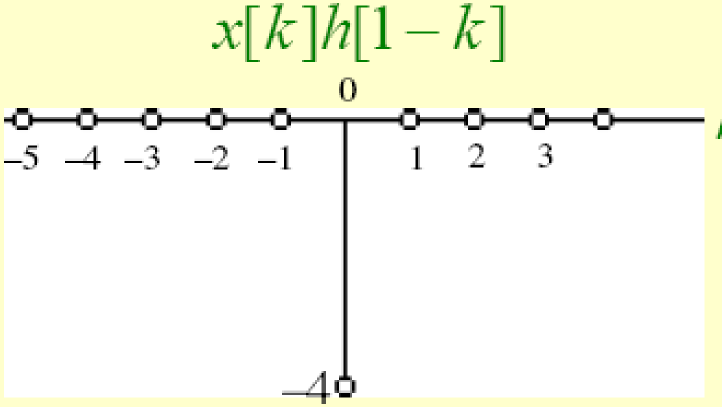


- Thus $y[0] = x[0]h[0] = -2$

Convolution Sum

- For the computation of y[1], we shift $\{h[-k]\}$ to the right by one sample period to form $\{h[1-k]\}$ as shown below on the left
- The product sequence $\{x[k]h[1-k]\}$ is shown below on the right



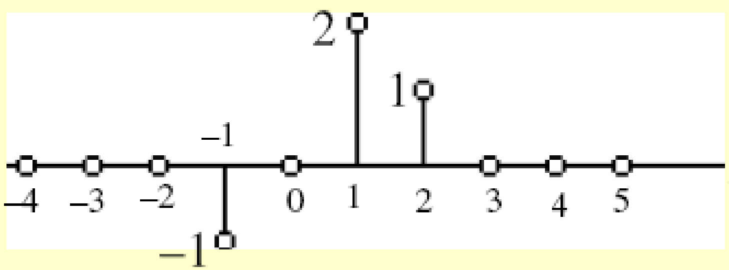


- Hence, $y[1]=x[0]h[1]+x[1]h[0]=-4+0=-4$

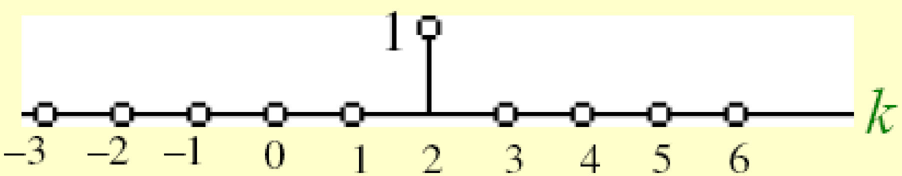
Convolution Sum

- To calculate $y[2]$, we form $\{h[2-k]\}$ as shown below on the left
- The product sequence $\{x[k]h[2-k]\}$ is plotted below on the right

$h[2-k]$



$x[k]h[2-k]$



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

Convolution Sum

- Continuing the process we get

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

$$y[4] = x[1]h[3] + x[2]h[2] + x[3]h[1] + x[4]h[0]$$
$$= 0 + 0 - 2 + 3 = 1$$

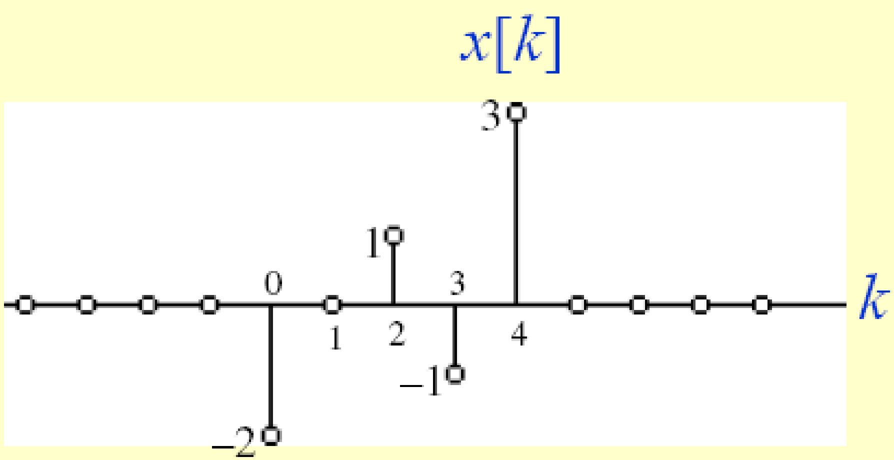
$$y[5] = x[2]h[3] + x[3]h[2] + x[4]h[1]$$
$$= -1 + 0 + 6 = 5$$

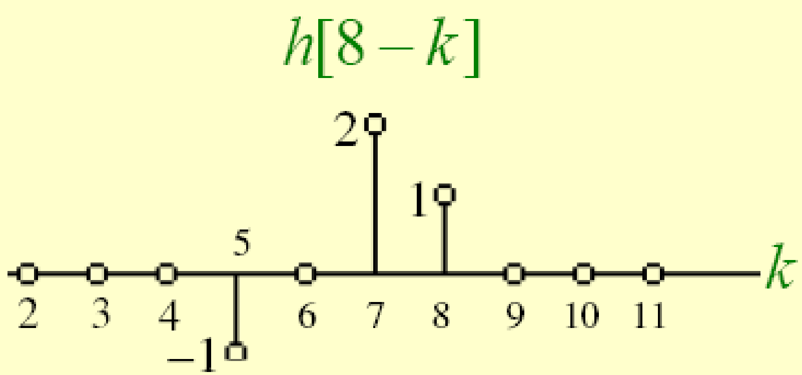
$$y[6] = x[3]h[3] + x[4]h[2] = 1 + 0 = 1$$

$$y[7] = x[4]h[3] = -3$$

Convolution Sum

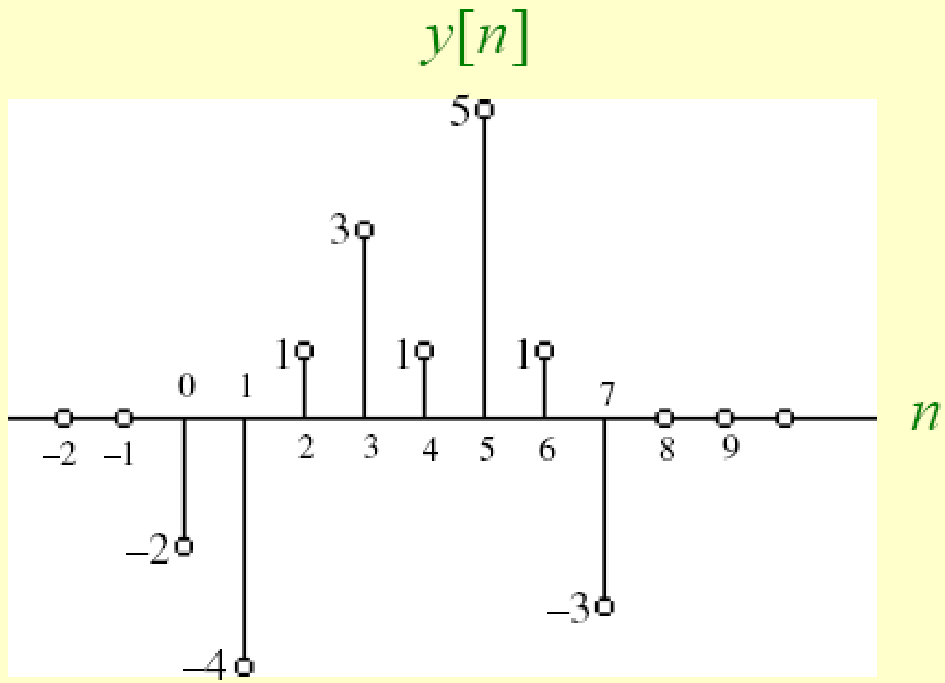
- From the plot of $\{h[n-k]\}$ for $n > 7$ and the plot of $\{x[k]\}$ as shown below, it can be seen that there is no overlap between these two sequences
- As a result $y[n] = 0$ for $n > 7$





Convolution Sum

- The sequence $\{y[n]\}$ generated by the convolution sum is shown below



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

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 $M + N - 1$

Operations on Finite-Length Sequences

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

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 $0 \le n \le N-1$

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

  $$m \text{ modulo } N$$

  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 chosen to make $m + \ell N$ an integer between 0 and $N-1$

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

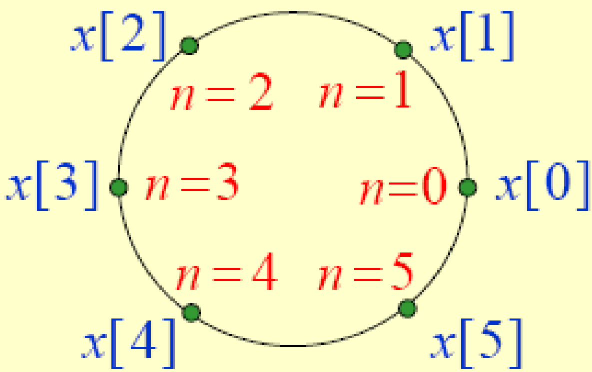
- The time shifting operation for a finite-length sequence, called circular shift operation, is defined using the modulo operation
- Let $x[n]$ be a length-N sequence defined for $0 \leq n \leq N-1$
- Its circularly shifted version $x_c[n]$, shifted n_o by samples, is given by

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

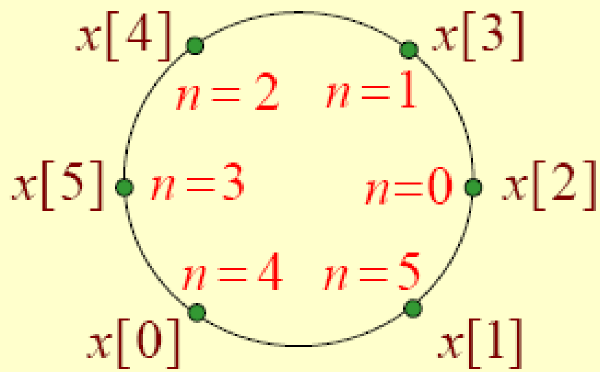
Circular Shift of a Sequence

- 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_o sample spacings as shown on the next slide

Circular Shift of a Sequence



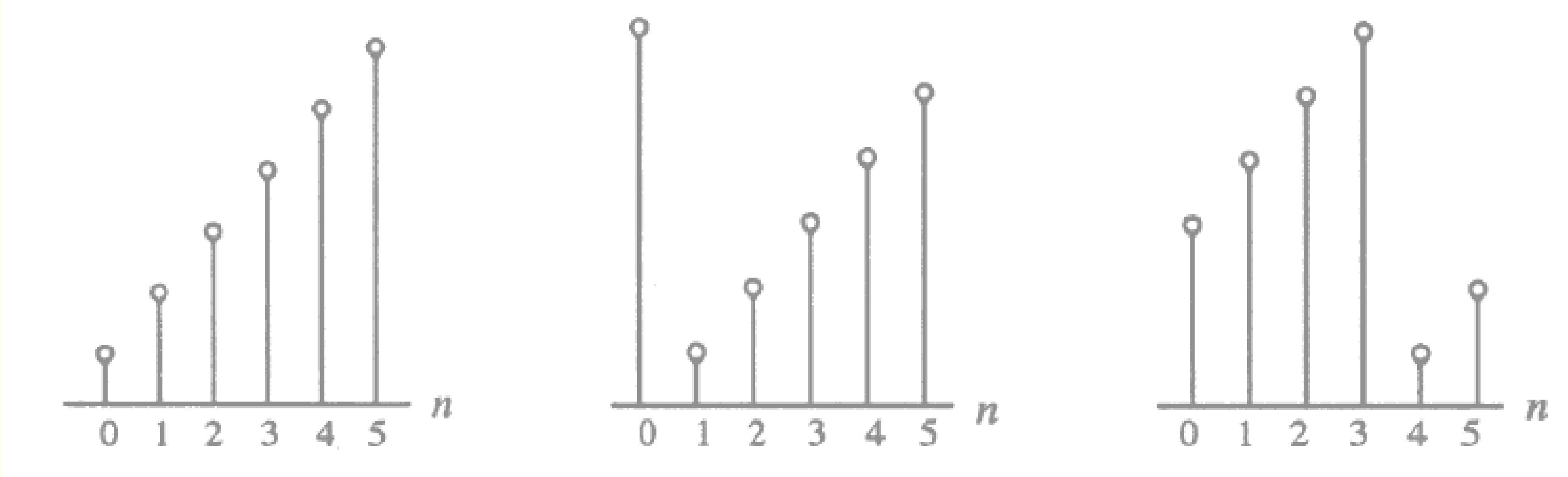
$$x[n]$$



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

Circular Shift of a Sequence

- Illustration of the concept of a circular shift



$x[n]$ $\qquad x[\langle n-1\rangle_6] = x[\langle n+5\rangle_6]$ $\qquad x[\langle n-4\rangle_6] = x[\langle n+2\rangle_6]$

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 $\langle n_o \rangle_N$

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

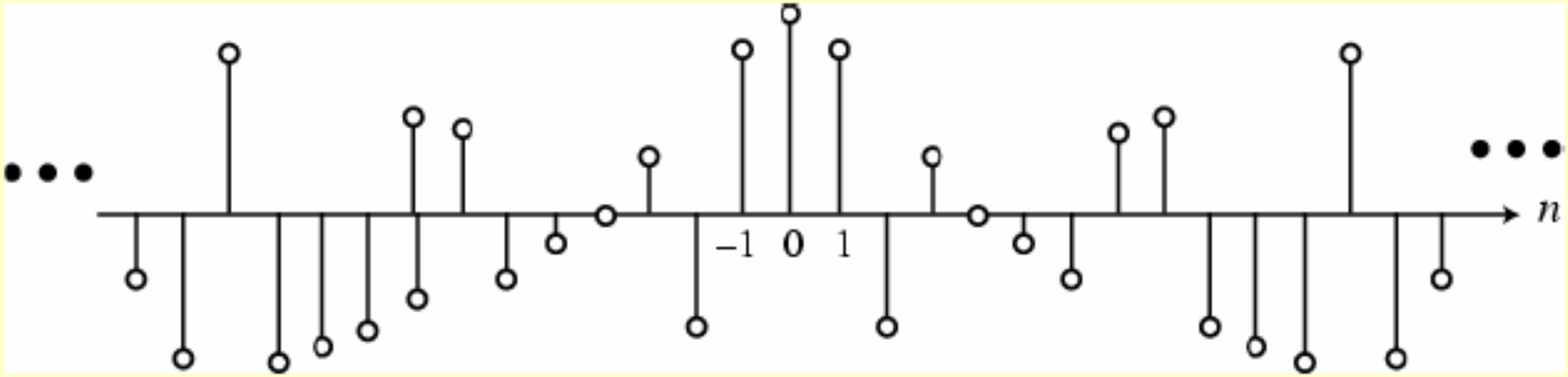
§ 2.1.3 Classification of Sequences
Based on Symmetry

共轭对称

- **Conjugate-symmetric sequence:**

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

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



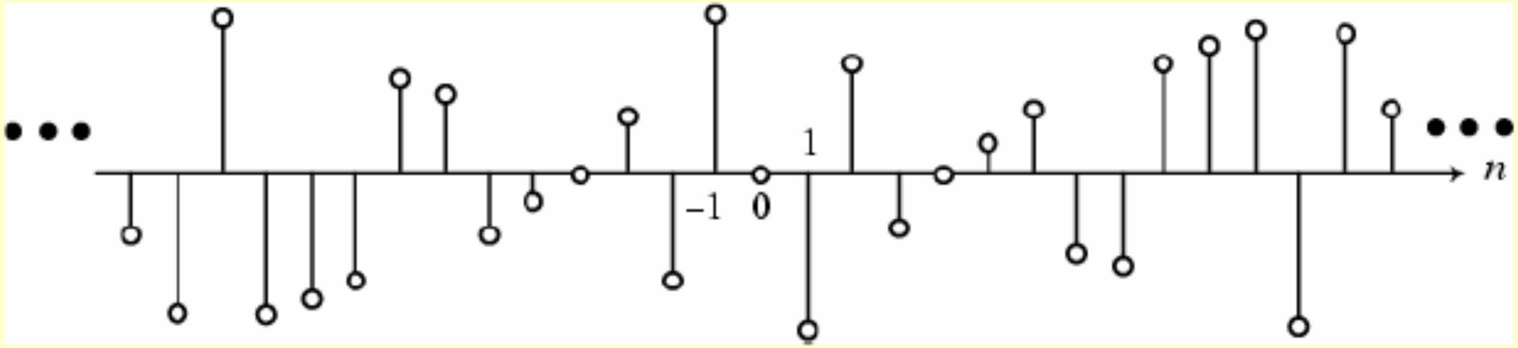
An even sequence

§ 2.1.3 Classification of Sequences Based on Symmetry

- **Conjugate-antisymmetric sequence**:

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

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



An odd sequence

§ 2.1.3 Classification of Sequences
Based on Symmetry

- 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 $\{w[n]\}$, $w[0] = 0$

§ 2.1.3 Classification of Sequences
Based on Symmetry

- 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

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

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

§ 2.1.3 Classification of Sequences
Based on Symmetry

Because of the time-reversal operation, the decomposition of a finite-length sequence into a sum of a conjugate-symmetric sequence and a conjugate-antisymmetric sequence is possible, if parent sequence is of odd length defined for a symmetric interval, $-M \leq 0 \leq M$.

§ 2.1.3 Classification of Sequences
Based on Symmetry

- Example - Consider the length-7 sequence defined for $-3 \le n \le 3$:

$$\{g[n]\} = \{0,\ 1+j4,\ -2+j3,\ \underset{\uparrow}{4-j2},\ -5-j6,\ -j2,\ 3\}$$

- Its conjugate sequence is then given by

$$\{g^*[n]\} = \{0,\ 1-j4,\ -2-j3,\ \underset{\uparrow}{4+j2},\ -5+j6,\ j2,\ 3\}$$

- The time-reversed version of the above is

$$\{g^*[-n]\} = \{3,\ j2,\ -5+j6,\ \underset{\uparrow}{4+j2},\ -2-j3,\ 1-j4,\ 0\}$$

§ 2.1.3 Classification of Sequences Based on Symmetry

- Therefore $\{g_{cs}[n]\} = \frac{1}{2}\{g[n] + g^*[-n]\}$

$$= \{1.5,\ 0.5+j3,\ -3.5+j4.5,\ \underset{\uparrow}{4},\ -3.5-j4.5,\ 0.5-j3,\ 1.5\}$$

- Likewise $\{g_{ca}[n]\} = \frac{1}{2}\{g[n] - g^*[-n]\}$

$$= \{-1.5,\ 0.5+j,\ 1.5-j1.5,\ \underset{\uparrow}{-j2},\ -1.5-j1.5,\ -0.5-j,\ 1.5\}$$

- It can be easily verified that $g_{cs}[n] = g^*_{cs}[-n]$ and $g_{ca}[n] = -g^*_{ca}[-n]$

§ 2.1.3 Classification of Sequences
Based on Symmetry

- 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

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

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

- **For a length-N sequence defined for 0≤n≤N-1, the above definitions of symmetry are not applicable.**

§ 2.1.3 Classification of Sequences

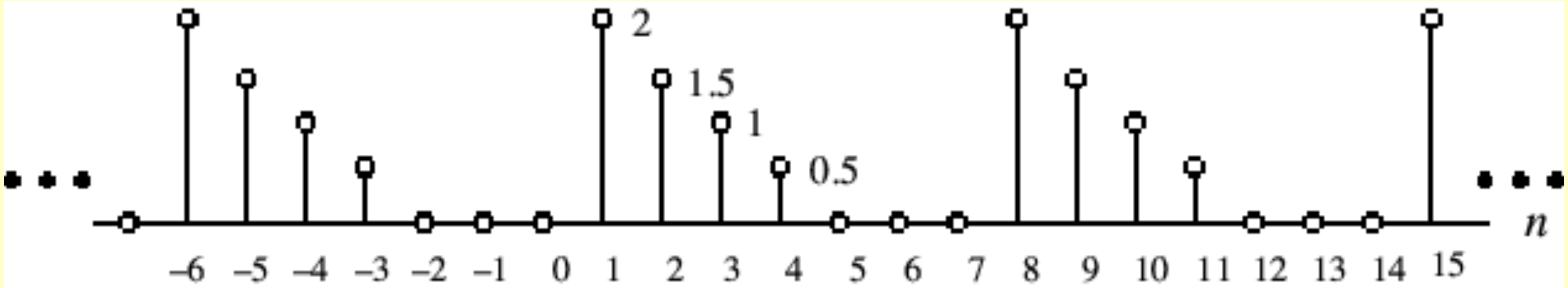
Periodic and Aperiodic Signals

- 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**

§ 2.1.3 Classification of Sequences

Periodic and Aperiodic Signals

- **<u>Example</u> -**



- **A sequence satisfying the periodicity condition is called an periodic sequence**

§ 2.1.3 Classification of Sequences

Energy and Power Signals

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

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

- **An infinite length sequence with finite sample values may or may not have finite energy**
- **A finite length sequence with finite sample values has finite energy**

Classification of Sequences: Energy and Power Signals

- **Example** – The infinite-length sequence

$$x[n] = \begin{cases} 1/n, & n \geq 1, \\ 0, & n \leq 0, \end{cases}$$

has an energy equal to

$$\mathcal{E}_x = \sum_{n=1}^{\infty} (1/n)^2$$

which converges to $\pi^2/6$, indicating that $x[n]$ has finite energy

Classification of Sequences: Energy and Power Signals

- **Example** – The infinite-length sequence

$$y[n] = \begin{cases} 1/\sqrt{n}, & n \geq 1, \\ 0, & n \leq 0, \end{cases}$$

has an energy equal to

$$\mathcal{E}_y = \sum_{n=1}^{\infty} (1/n)$$

which does not converge indicating that $y[n]$ has infinite energy

§ 2.1.3 Classification of Sequences
Energy and Power Signals

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

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

- **Define the energy of a sequence x[n] over a finite interval -K≤ n ≤ K as**

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

Classification of Sequences: Energy and Power Signals

- Then

$$P_x = \lim_{K \to \infty} \frac{1}{2K+1} \mathcal{E}_{x.K}$$

- The **average power** of a periodic sequence $\tilde{x}[n]$ with a period N is given by

$$P_x = \frac{1}{N} \sum_{n=0}^{N-1} \left| \tilde{x}[n] \right|^2$$

- The average power of an infinite-length sequence may be finite or infinite

§ 2.1.3 Classification of Sequences

Energy and Power Signals

- **<u>Example</u> - Consider the causal sequence defined by**

$$x[n]=\begin{cases}3(-1)^n, & n\geq 0\\ 0, & n<0\end{cases}$$

- **Note: $x[n]$ has infinite energy**
- **Its average power is given by**

$$P_x=\lim_{K\to\infty}\frac{1}{2K+1}\left(9\sum_{n=0}^{K}1\right)=\lim_{K\to\infty}\frac{9(K+1)}{2K+1}=4.5$$

§ 2.1.3 Classification of Sequences

Energy and Power Signals

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

  <u>Example</u> - 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**

  <u>Example</u> - A finite-length sequence which has finite energy but zero average power

§ 2.1.3 Classification of Sequences

bounded, absolutely summable and square-summable

➢ **A sequence $x[n]$ is said to be bounded(有界的) if**

$$\left|x[n]\right| \le B_x < \infty$$

- **Example - The sequence x[n]=cos(0.3πn) is a bounded sequence as**

$$\left|x[n]\right| = \left|\cos 0.3\pi n\right| \le 1$$

§ 2.1.3 Classification of Sequences

bounded, absolutely summable and squaresummable

- A sequence $x[n]$ is said to be **absolutely summable(绝对可和)** if

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

- **Example** - The sequence

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

is an absolutely summable sequence as

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

§ 2.1.3 Classification of Sequences

bounded, absolutely summable and squaresummable

- A sequence $x[n]$ is said to be **square-summable(平方可和)** if

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

  - **Example** - The sequence

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

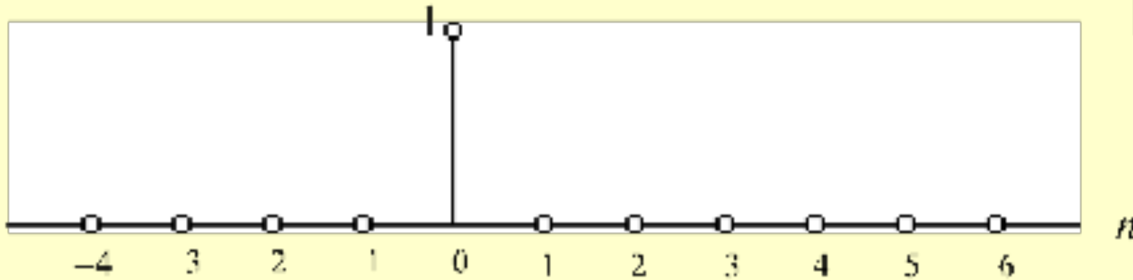
is square-summable but not absolutely summable

§ 2.2 Typical Sequences and Sequences Representation

§ 2.2.1 Some Basic Sequences

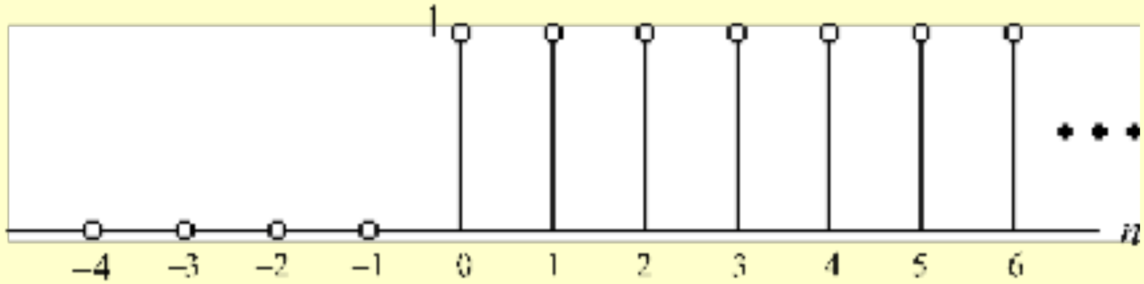
单位样本序列

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



单位阶跃序列

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



§ 2.2.1 Some Basic Sequences

正弦序列

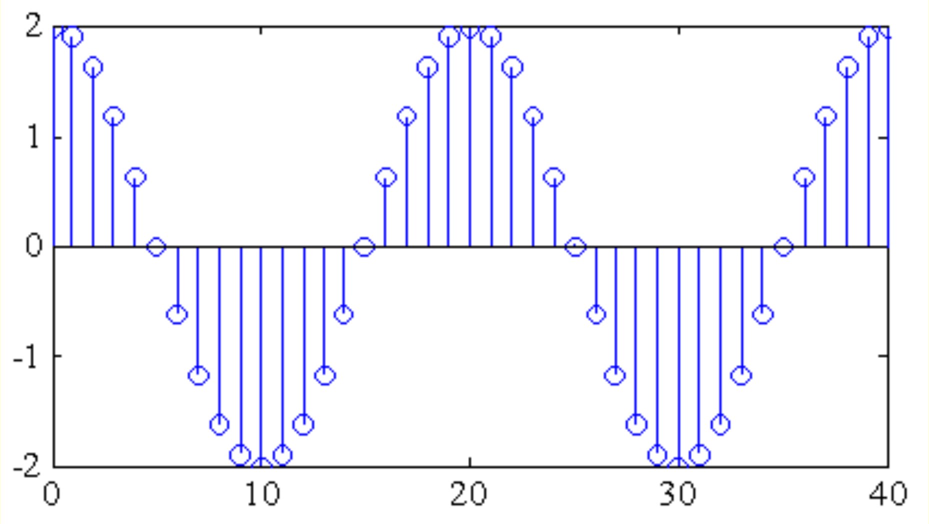
- Real sinusoidal sequence -

$$x[n]=A\cos(\omega_0 n+\varphi)$$

角频率

where A is the amplitude, ω_0 is the angular frequency, and φ is the phase of x[n]

Example -



§ 2.2.1 Some Basic Sequences

指数序列

- **Exponential sequence -**

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

where A and α 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

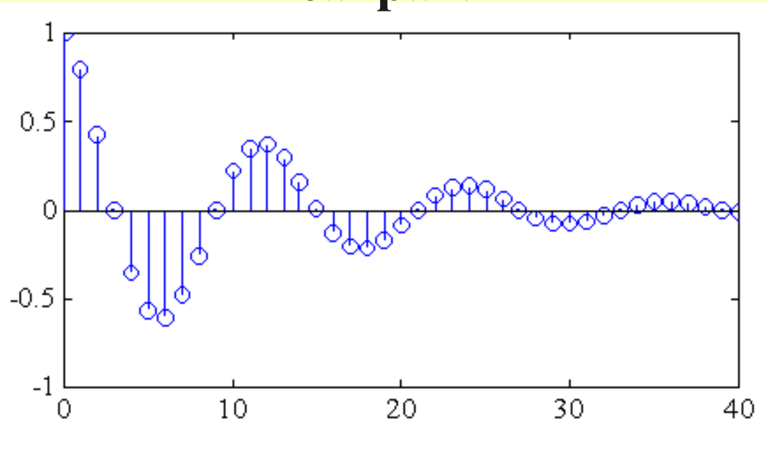
$$x_{re}[n] = |A| e^{\sigma_o n} \cos(\omega_o n + \phi),$$

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

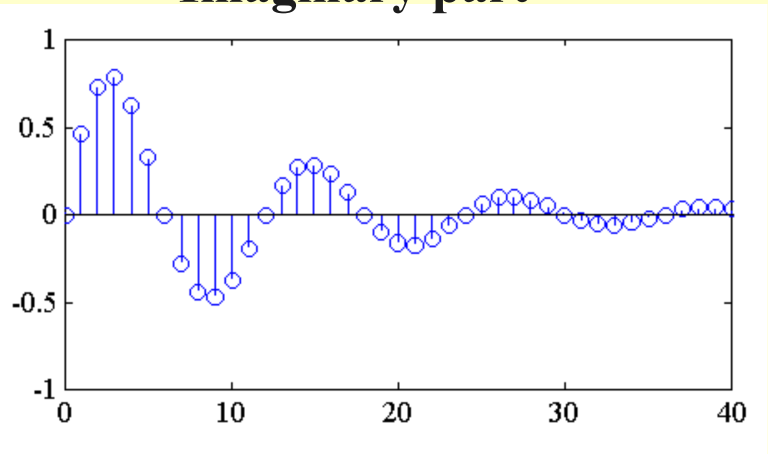
§ 2.2.1 Some Basic Sequences

- $x_{re}[n]$ and $x_{im}[n]$ of a complex exponential sequence are real sinusoidal sequences with constant (σ_0=0), 增长 growing (σ_0>0) , and 衰减 decaying (σ_0<0) amplitudes for n > 0

Real part



Imaginary part



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

§ 2.2.2 Sequence Generation Using Matlab

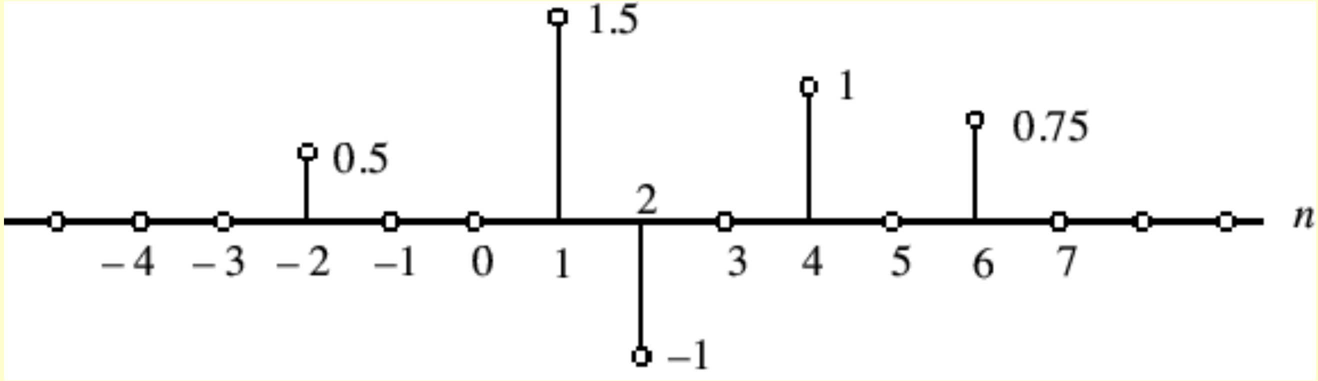
函数

- Matlab includes a number of **functions** that can be used for signal generation. Some of these functions of interest are

exp, sin, cos, square, sawtooth

§ 2.2.3 Representation of an Arbitrary Sequence

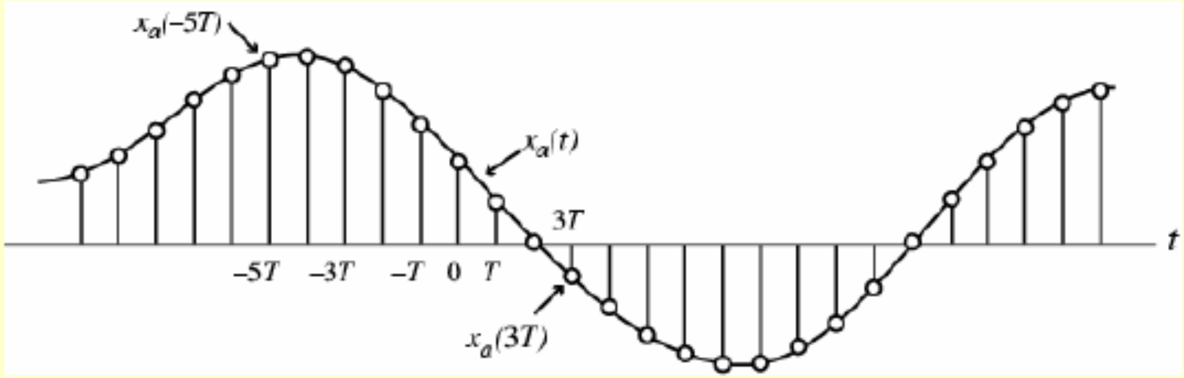
- An arbitrary sequence can be represented in the time-domain as a 加权和 **weighted sum** of some basic sequence and its delayed (advanced) versions



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

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

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

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

The Sampling Process

- Consider the continuous-time signal

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

- 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 $x[n]$

The Sampling Process

- If the unit of sampling period T is in seconds
- The unit of normalized digital angular frequency ω_o is radians/sample
- The unit of normalized analog angular frequency Ω_o is radians/second
- The unit of analog frequency f_o is hertz (Hz)

The Sampling Process

- The three continuous-time signals

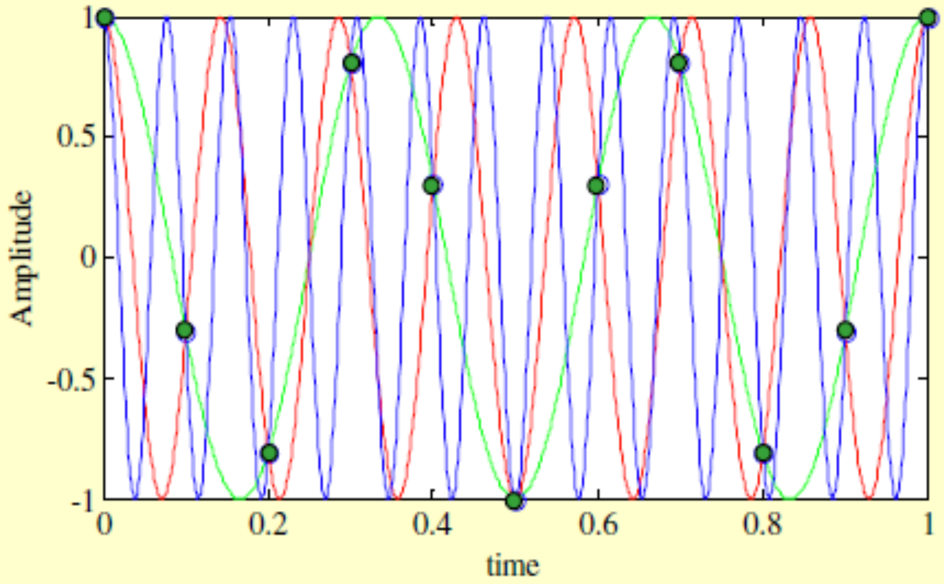
$$g_1(t) = \cos(6\pi t)$$
$$g_2(t) = \cos(14\pi t)$$
$$g_3(t) = \cos(26\pi t)$$

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

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

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

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

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**

The Sampling Process

- Since there are an infinite number of continuous-time signals that can lead to the same sequence when sampled periodically, additional conditions need to imposed so that the sequence $\{x[n]\} = \{x_a(nT)\}$ can uniquely represent the parent continuous-time signal $x_a(t)$
- In this case, $x_a(t)$ can be fully recovered from $\{x[n]\}$

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

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

$$v[n] = 6\cos(0.3\pi n) + 3\sin(1.5\pi n) + 2\cos(1.7\pi n)$$
$$+ 4\cos(2.5\pi n) + 10\sin(3.3\pi n)$$
$$= 6\cos(0.3\pi n) + 3\sin\big((2\pi - 0.5\pi)n\big) + 2\cos\big((2\pi - 0.3\pi)n\big)$$
$$+ 4\cos\big((2\pi + 0.5\pi)n\big) + 10\sin\big((4\pi - 0.7\pi)n\big)$$

The Sampling Process

$$= 6\cos(0.3\pi n) - 3\sin(0.5\pi n) + 2\cos(0.3\pi n) + 4\cos(0.5\pi n) - 10\sin(0.7\pi n)$$

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

- Note: $v[n]$ is composed of 3 discrete-time sinusoidal signals of normalized angular frequencies: 0.3π, 0.5π, and 0.7π

The Sampling Process

- Note: An identical discrete-time signal is also generated by uniformly sampling at a 200-Hz sampling rate the following continuous-time signals:

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

$$\begin{aligned} g_a(t) = {} & 2\cos(60\pi t) + 4\cos(100\pi t) + 10\sin(260\pi t) \\ & + 6\cos(460\pi t) + 3\sin(700\pi t) \end{aligned}$$

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 ω_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

The Sampling Process

- On the other hand, if $\Omega_T < 2\Omega_o$, the normalized digital angular frequency will foldover into a lower digital frequency $\omega_o = \langle 2\pi\Omega_o / \Omega_T \rangle_{2\pi}$ in the range $-\pi < \omega < \pi$ because of aliasing
- Hence, to prevent aliasing, the sampling frequency Ω_T should be greater than 2 times the frequency Ω_o of the sinusoidal signal being sampled

The Sampling Process

- Generalization: Consider an arbitrary continuous-time signal $x_a(t)$ composed of a weighted sum of a number of sinusoidal signals
- $x_a(t)$ can be represented uniquely by its sampled version $\{x[n]\}$ if the sampling frequency Ω_T is chosen to be greater than 2 times the highest frequency contained in $x_a(t)$

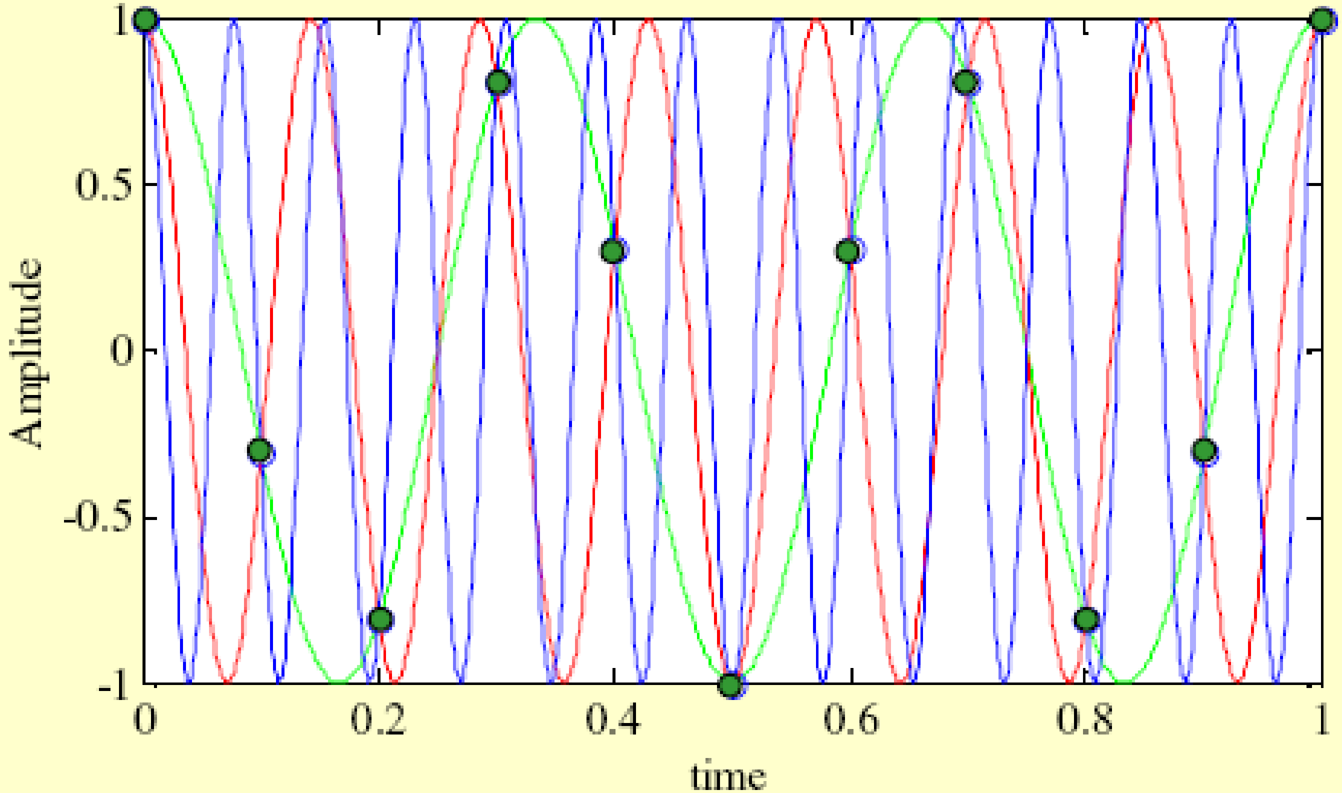
The Sampling Process

- The condition to be satisfied by the sampling frequency to prevent aliasing is called the **sampling theorem**
- A formal proof of this theorem will be presented later

Homework (M2.6, M2.7)

- Write a Matlab program to plot a continuous-time sinusoidal signal and its sampled version, and verify Figure 2.28. You need to use the *hold* function to keep both plots.
- Using the program developed in the previous problem, verify experimentally that the family of continuous-time sinusoids given by Eq.(2.65) lead to identical sampled signals.

Figure 2.28



Eq.(2.65)

$$x_{a,k}(t) = A\cos(\pm\Omega_0 t + \phi) + k\Omega_T t), k = 0, \pm 1, \pm 2, \ldots$$