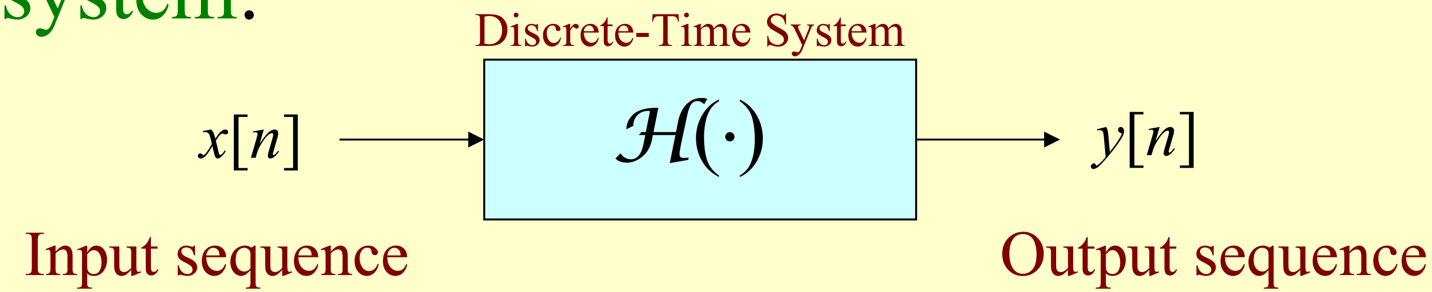
Discrete-Time Systems

- A discrete-time system processes a given **input sequence** $x[n]$ to generates an **output sequence** $y[n]$ with more desirable properties
- In most applications, the discrete-time system is a single-input, single-output system:

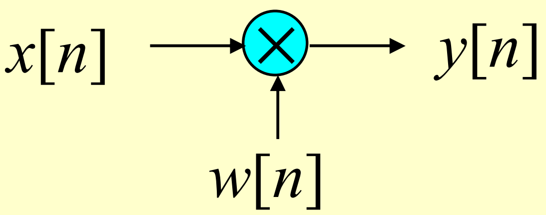


Discrete-Time Systems

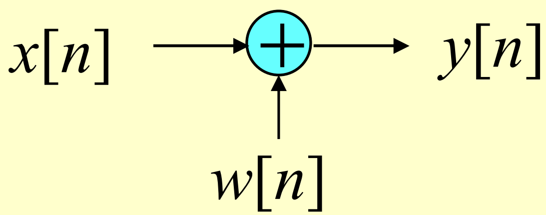
- Mathematically, the discrete-time system is characterized by an operator $\mathcal{H}(\cdot)$ that transforms the input sequence $x[n]$ into another sequence $y[n]$ at the output
- The discrete-time system may also have more than one input and/or more than one output

Discrete-Time Systems: Examples

- 2-input, 1-output discrete-time systems -

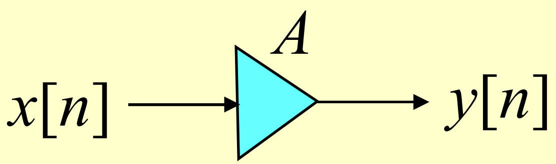


Modulator

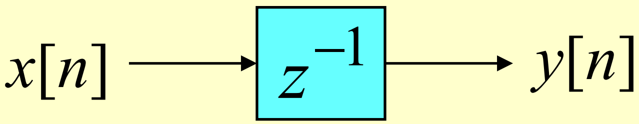


Adder

- 1-input, 1-output discrete-time systems -

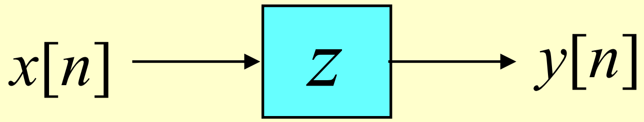


Multiplier



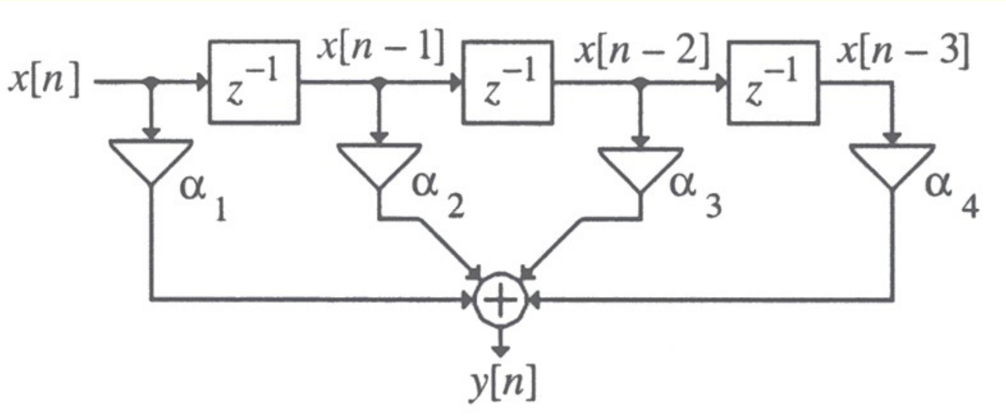
Unit Delay

Unit Advance



Discrete-Time Systems: Examples

- A more complex example of an one-input, one-output discrete-time system is shown below



Discrete-Time Systems: Examples

- **Accumulator -** $y[n] = \sum_{\ell=-\infty}^{n} x[\ell]$

$$= \sum_{\ell=-\infty}^{n-1} x[\ell] + x[n] = y[n-1] + x[n]$$

- The output $y[n]$ at time instant n is the sum of the input sample $x[n]$ at time instant n and the previous output $y[n-1]$ at time instant $n-1$, which is the sum of all previous input sample values from $-\infty$ to $n-1$
- The system cumulatively adds, i.e., it accumulates all input sample values

Discrete-Time Systems:Examples

- **Accumulator** - Input-output relation can also be written in the form

$$y[n] = \sum_{\ell=-\infty}^{-1} x[\ell] + \sum_{\ell=0}^{n} x[\ell]$$

$$= y[-1] + \sum_{\ell=0}^{n} x[\ell], \; n \geq 0$$

- The second form is used for a causal input sequence, in which case $y[-1]$ is called the **initial condition**

Discrete-Time Systems:Examples

- **M-point Moving-Average System** -

$$y[n] = \frac{1}{M} \sum_{k=0}^{M-1} x[n-k]$$

- Used in smoothing random variations in data
- In most applications, the data $x[n]$ is a bounded sequence
- ⟹ M-point average $y[n]$ is also a bounded sequence

Discrete-Time Systems:Examples

- If there is no bias in the measurements, an improved estimate of the noisy data is obtained by simply increasing M
- A direct implementation of the M-point moving average system requires $M-1$ additions, 1 division, and storage of $M-1$ past input data samples
- A more efficient implementation is developed next

Discrete-Time Systems:Examples

$$y[n] = \frac{1}{M}\left(\sum_{\ell=0}^{M-1} x[n-\ell]\right)$$

$$= \frac{1}{M}\left(\sum_{\ell=1}^{M-1} x[n-\ell] + x[n]\right)$$

$$= \frac{1}{M}\left(\sum_{\ell=1}^{M-1} x[n-\ell] + x[n] + x[n-M] - x[n-M]\right)$$

Discrete-Time Systems:Examples

$$y[n] = \frac{1}{M}\left(\sum_{\ell=1}^{M} x[n-\ell] + x[n] - x[n-M]\right)$$

$$= \frac{1}{M}\left(\sum_{\ell=0}^{M-1} x[n-1-\ell] + x[n] - x[n-M]\right)$$

$$= \frac{1}{M}\left(\sum_{\ell=0}^{M-1} x[n-1-\ell]\right) + \frac{1}{M}\big(x[n] - x[n-M]\big)$$

Hence

$$y[n] = y[n-1] + \frac{1}{M}\big(x[n] - x[n-M]\big)$$

Discrete-Time Systems:Examples

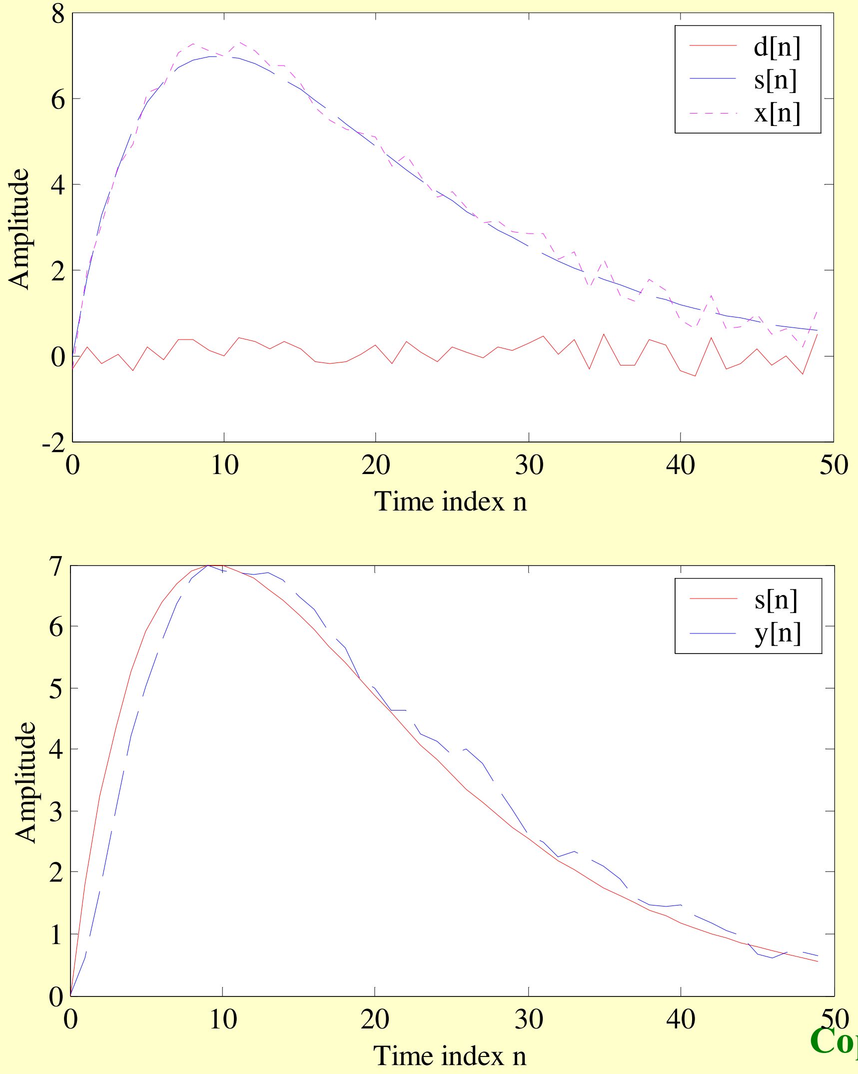
- Computation of the modified M-point moving average system using the recursive equation now requires 2 additions and 1 division
- An application: Consider

$$x[n] = s[n] + d[n],$$

where $s[n]$ is the signal corrupted by a noise $d[n]$

Discrete-Time Systems:Examples

$$s[n] = 2[n(0.9)^n],\ d[n]\ \text{- random signal}$$



Discrete-Time Systems:Examples

- **Exponentially Weighted Running Average Filter**

$$y[n] = \alpha y[n-1] + x[n], \quad 0 < \alpha < 1$$

- Computation of the running average requires only 1 addition, 1 multiplication and storage of the previous running average
- Does not require storage of past input data samples

Discrete-Time Systems:Examples

- For $0 < \alpha < 1$, the exponentially weighted average filter places more emphasis on current data samples and less emphasis on past data samples as illustrated below

$$y[n] = \alpha\big(\alpha y[n-2] + x[n-1]\big) + x[n]$$

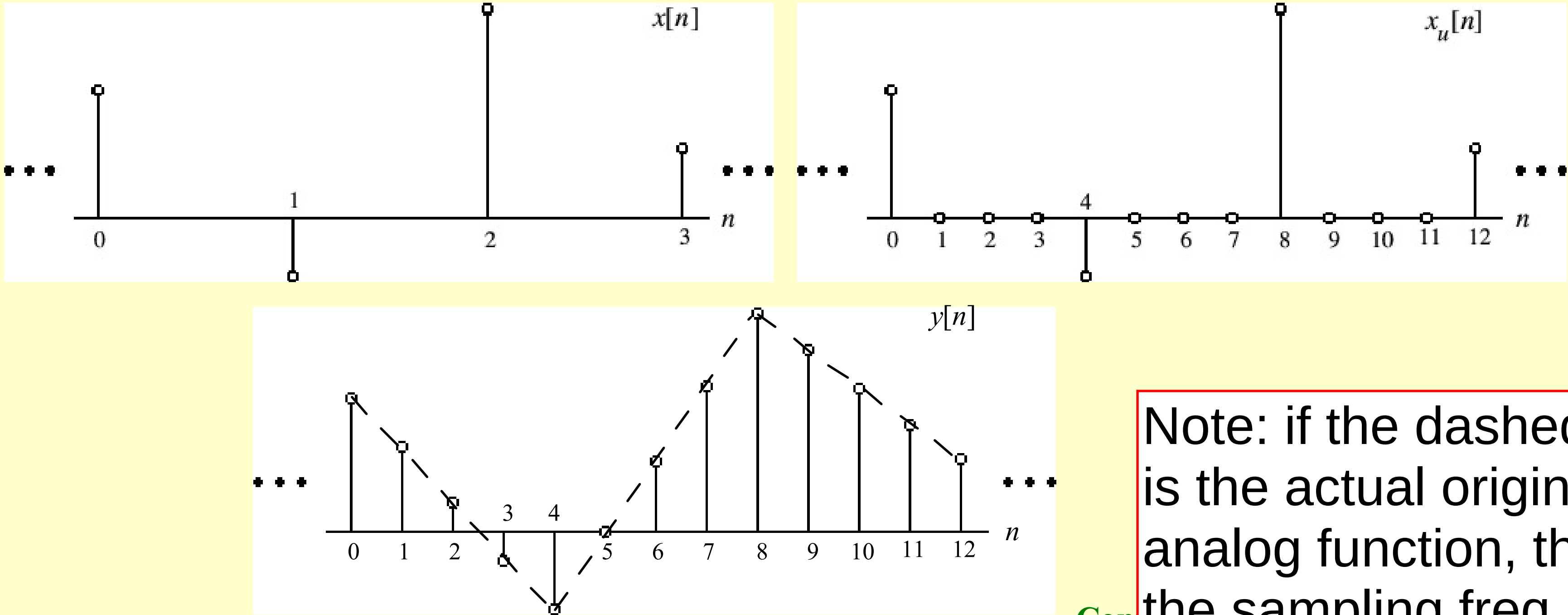
$$= \alpha^2 y[n-2] + \alpha x[n-1] + x[n]$$

$$= \alpha^2\big(\alpha y[n-3] + x[n-2]\big) + \alpha x[n-1] + x[n]$$

$$= \alpha^3 y[n-3] + \alpha^2 x[n-2] + \alpha x[n-1] + x[n]$$

Discrete-Time Systems:Examples

- **Linear interpolation** - Employed to estimate sample values between pairs of adjacent sample values of a discrete-time sequence
- **Factor-of-4 interpolation**



Note: if the dashed line is the actual original analog function, then the sampling freq. is TOO LOW

Cop

Discrete-Time Systems: Examples

- **Factor-of-2 interpolator** -

$$y[n] = x_u[n] + \frac{1}{2}\left(x_u[n-1] + x_u[n+1]\right)$$

- **Factor-of-3 interpolator** -

$$y[n] = x_u[n] + \frac{1}{3}\left(x_u[n-\boxed{2}] + x_u[n+2]\right)$$

$$+ \frac{2}{3}\left(x_u[n-\boxed{1}] + x_u[n+1]\right)$$

1 Note: DCgain = *L* (amplification factor of the 0-frequency input component)
Note: a formally correct solution would require more complex operators

Discrete-Time Systems: Examples

- **Factor-of-2 interpolator** –

Original (512×512)

Down−sampled (256×256)

Interpolated (512 × 512)

MATLAB

Discrete-Time Systems: Examples

Median Filter –

or equal to

- The median of a set of $(2K+1)$ numbers is the number such that K numbers from the set have values greater than this number and the other K numbers have values smaller
- Median can be determined by rank-ordering the numbers in the set by their values and choosing the number at the middle

Discrete-Time Systems: Examples

Median Filter –

- **Example**: Consider the set of numbers

$$\{2, \quad -3, \quad 10, \quad 5, \quad -1\}$$

- Rank-ordered set is given by

$$\{-3, \quad -1, \quad 2, \quad 5, \quad 10\}$$

- Hence,

$$\text{med}\{2, \quad -3, \quad 10, \quad 5, \quad -1\} = 2$$

Discrete-Time Systems: Examples

Median Filter –

- Implemented by sliding a window of odd length over the input sequence $\{x[n]\}$ one sample at a time
- Output $y[n]$ at instant n is the median value of the samples inside the window centered at n

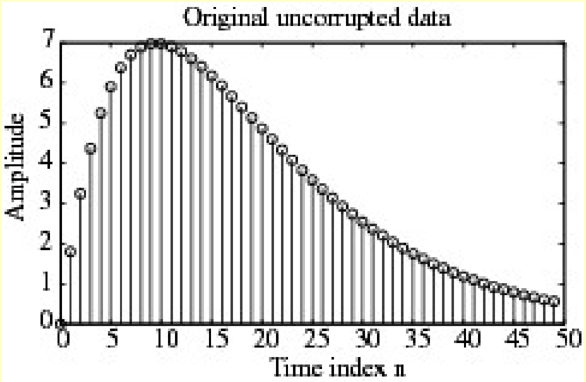
Discrete-Time Systems: Examples

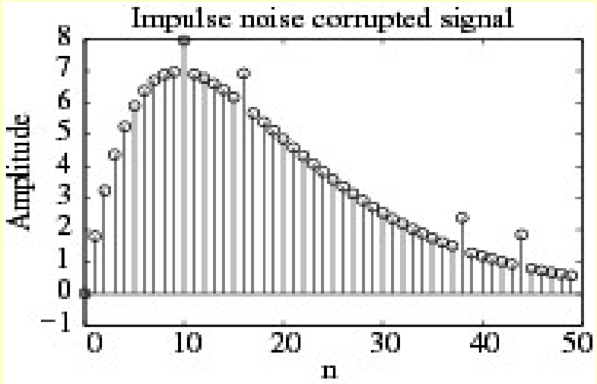
Median Filter –

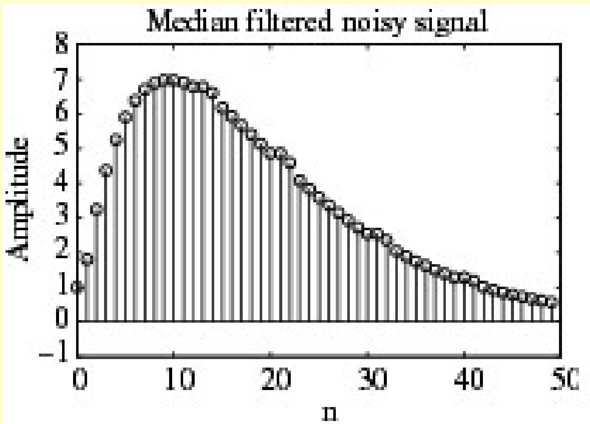
- Finds applications in removing additive $\boxed{\text{or substitutive}}$ random noise, which shows up as sudden large errors in the corrupted signal
- Usually used for the smoothing of signals corrupted by impulse noise

Discrete-Time Systems: Examples

Median Filtering Example –







Discrete-Time Systems: Classification

- Linear System
- Shift-Invariant System
- Causal System
- Stable System
- Passive and Lossless Systems

Linear Discrete-Time Systems

- **Definition** - If $y_1[n]$ is the output due to an input $x_1[n]$ and $y_2[n]$ is the output due to an input $x_2[n]$ then for an input

$$x[n] = \alpha\, x_1[n] + \beta\, x_2[n]$$

  the output is given by

$$y[n] = \alpha\, y_1[n] + \beta\, y_2[n]$$

- Above property must hold for any arbitrary constants α and β, and for all possible inputs $x_1[n]$ and $x_2[n]$

Linear Discrete-Time Systems

- **Accumulator** $y_1[n] = \sum_{\ell=-\infty}^{n} x_1[\ell], \quad y_2[n] = \sum_{\ell=-\infty}^{n} x_2[\ell]$

  For an input

$$x[n] = \alpha\, x_1[n] + \beta\, x_2[n]$$

  the output is

$$y[n] = \sum_{\ell=-\infty}^{n} \left(\alpha\, x_1[\ell] + \beta\, x_2[\ell]\right)$$

$$= \alpha \sum_{\ell=-\infty}^{n} x_1[\ell] + \beta \sum_{\ell=-\infty}^{n} x_2[\ell] = \alpha\, y_1[n] + \beta\, y_2[n]$$

- Hence, the above system is **linear**

Linear Discrete-Time Systems

- The outputs $y_1[n]$ and $y_2[n]$ for inputs $x_1[n]$ and $x_2[n]$ are given by

causal

$$y_1[n] = y_1[-1] + \sum_{\ell=0}^{n} x_1[\ell]$$

$$y_2[n] = y_2[-1] + \sum_{\ell=0}^{n} x_2[\ell]$$

- The output $y[n]$ for an input $\alpha\, x_1[n] + \beta\, x_2[n]$ is given by

$$y[n] = y[-1] + \sum_{\ell=0}^{n} (\alpha\, x_1[\ell] + \beta\, x_2[\ell])$$

Linear Discrete-Time Systems

- Now $\alpha\, y_1[n] + \beta\, y_2[n]$

$$= \alpha\Big(y_1[-1] + \sum_{\ell=0}^{n} x_1[\ell]\Big) + \beta\Big(y_2[-1] + \sum_{\ell=0}^{n} x_2[\ell]\Big)$$

$$= (\alpha\, y_1[-1] + \beta\, y_2[-1]) + \Big(\alpha \sum_{\ell=0}^{n} x_1[\ell] + \beta \sum_{\ell=0}^{n} x_2[\ell]\Big)$$

- Thus $y[n] = \alpha\, y_1[n] + \beta\, y_2[n]$ if

$$y[-1] = \alpha\, y_1[-1] + \beta\, y_2[-1]$$

27

Linear Discrete-Time System

- For the accumulator with a causal input to be **linear** the condition

$$y[-1] = \alpha\, y_1[-1] + \beta\, y_2[-1]$$

  must hold for all initial conditions $y[-1]$, $y_1[-1]$, $y_2[-1]$, and all constants α and β
- This condition cannot be satisfied unless the accumulator is initially at rest with zero initial condition

Nonlinear Discrete-Time System

- The median filter described earlier is a nonlinear discrete-time system
- To show this, consider a median filter with a window of length 3
- Output of the filter for an input

$$\{x_1[n]\} = \{3, \quad 4, \quad 5\}, 0 \le n \le 2$$

is

$$\{y_1[n]\} = \{3, \quad 4, \quad 4\}, 0 \le n \le 2$$

29

Nonlinear Discrete-Time System

- Output for an input

$$\{x_2[n]\} = \{2, \quad -1, \quad -1\}, 0 \le n \le 2$$

is

$$\{y_2[n]\} = \{0, \quad -1, \quad -1\}, 0 \le n \le 2$$

- However, the output for an input

$$\{x[n]\} = \{x_1[n] + x_2[n]\}$$

is

$$\{y[n]\} = \{3, \quad 4, \quad 3\}$$

Nonlinear Discrete-Time System

- Note: $\{y_1[n]+y_2[n]\}=\{3, \quad 3, \quad 3\}\neq\{y[n]\}$
- Hence, the median filter is a nonlinear discrete-time system
- The second form of the accumulator with non-zero initial condition is another example

Shift-Invariant System

- For a shift-invariant system, if $y_1[n]$ is the response to an input $x_1[n]$, then the response to an input

$$x[n] = x_1[n - n_o]$$

is simply

$$y[n] = y_1[n - n_o]$$

where n_o is any positive or negative integer

- The above relation must hold for any arbitrary input and its corresponding output

Shift-Invariant System

- In the case of sequences and systems with indices n related to discrete instants of time, the above property is called **time-invariance** property
- Time-invariance property ensures that for a specified input, the output is independent of the time the input is being applied

33

Shift-Invariant System

- Example - Consider the up-sampler

$$x[n] \longrightarrow \boxed{\uparrow L} \longrightarrow x_u[n]$$

with an input-output relation given by

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

Shift-Invariant System

- For an input $x_1[n] = x[n - n_o]$ the output $x_{1,u}[n]$ is given by

$$x_{1,u}[n] = \begin{cases} x_1[n/L], & n = 0, \pm L, \pm 2L, \ldots.. \\ 0, & \text{otherwise} \end{cases}$$

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

Shift-Invariant System

- However from the definition of the up-sampler

$$x_u[n-n_o]$$
$$= \begin{cases} x[(n-n_o)/L], & n = n_o, n_o \pm L, n_o \pm 2L, ... \\ 0, & \text{otherwise} \end{cases}$$
$$\neq x_{1,u}[n]$$

- Hence, the up-sampler is a time-varying system

Linear Time-Invariant System

- **Linear Time-Invariant (LTI) System** -
  A system satisfying both the linearity and the time-invariance property
- LTI systems are mathematically easy to analyze and characterize, and consequently, easy to design
- Highly useful signal processing algorithms have been developed utilizing this class of systems over the last several decades

Causal System

- In a **causal system**, the n_o-th output sample $y[n_o]$ depends only on input samples $x[n]$ for $n \leq n_o$ and does not depend on input samples for $n > n_o$
- Let $y_1[n]$ and $y_2[n]$ be the responses of a causal discrete-time system to the inputs $x_1[n]$ and $x_2[n]$, respectively

Causal System

- Then

$$x_1[n] = x_2[n] \text{ for } n < N$$

implies also that

$$y_1[n] = y_2[n] \text{ for } n < N$$

- For a causal system, changes in output samples do not precede changes in the input samples

39

Causal System

- Examples of causal systems:

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

$$y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2]$$
$$+ a_1 y[n-1] + a_2 y[n-2]$$

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

- Examples of noncausal systems:

$$y[n] = x_u[n] + \frac{1}{2}(x_u[n-1] + x_u[n+1])$$

$$y[n] = x_u[n] + \frac{1}{3}(x_u[n-\boxed{2}] + x_u[n+2])$$
$$+ \frac{2}{3}(x_u[n-\boxed{1}] + x_u[n+1])$$

Causal System

- A noncausal system can be implemented as a causal system by delaying the output by an appropriate number of samples
- For example a causal implementation of the factor-of-2 interpolator is given by

$$y[n] = x_u[n-1] + \frac{1}{2}(x_u[n-2] + x_u[n])$$

Stable System

- There are various definitions of stability
- We consider here the **bounded-input, bounded-output (BIBO) stability**
- If $y[n]$ is the response to an input $x[n]$ and if

$$|x[n]| \le B_x \quad \text{for all values of } n$$

then

$$|y[n]| \le B_y \quad \text{for all values of } n$$

Stable System

- Example - The M-point moving average filter is BIBO stable:

$$y[n] = \frac{1}{M}\sum_{k=0}^{M-1} x[n-k]$$

- For a bounded input $|x[n]| \le B_x$ we have

$$|y[n]| = \left|\frac{1}{M}\sum_{k=0}^{M-1} x[n-k]\right| \le \frac{1}{M}\sum_{k=0}^{M-1} |x[n-k]|$$

$$\le \frac{1}{M}(MB_x) = B_x$$

Passive and Lossless Systems

- A discrete-time system is defined to be **passive** if, for every finite-energy input $x[n]$, the output $y[n]$ has, at most, the same energy, i.e.

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

- For a **lossless** system, the above inequality is satisfied with an equal sign for every input

Passive and Lossless Systems

- Example - Consider the discrete-time system defined by $y[n] = \alpha\, x[n-N]$ with N a positive integer
- Its output energy is given by

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

- Hence, it is a passive system if $\left|\alpha\right| < 1$ and is a lossless system if $\left|\alpha\right| = 1$

Impulse and Step Responses

- The response of a discrete-time system to a unit sample sequence $\{\delta[n]\}$ is called the **unit sample response** or simply, the **impulse response**, and is denoted by $\{h[n]\}$
- The response of a discrete-time system to a unit step sequence $\{\mu[n]\}$ is called the **unit step response** or simply, the **step response**, and is denoted by $\{s[n]\}$

Impulse Response

- Example - The impulse response of the system

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

is obtained by setting $x[n]=\delta[n]$ resulting in

$$h[n]=\alpha_1\delta[n]+\alpha_2\delta[n-1]+\alpha_3\delta[n-2]+\alpha_4\delta[n-3]$$

- The impulse response is thus a finite-length sequence of length 4 given by

$$\{h[n]\}=\{\underset{\uparrow}{\alpha_1}, \quad \alpha_2, \quad \alpha_3, \quad \alpha_4\}$$

note: coefficients = impulse response
NOT TRUE for recursive systems

Impulse Response

- Example - The impulse response of the discrete-time accumulator

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

is obtained by setting $x[n] = \delta[n]$ resulting in

$$h[n] = \sum_{\ell=-\infty}^{n} \delta[\ell] = \mu[n]$$

48

Impulse Response

- Example - The impulse response $\{h[n]\}$ of the factor-of-2 interpolator

$$y[n] = x_u[n] + \frac{1}{2}(x_u[n-1] + x_u[n+1])$$

- is obtained by setting $x_u[n] = \delta[n]$ and is given by

$$h[n] = \delta[n] + \frac{1}{2}(\delta[n-1] + \delta[n+1])$$

- The impulse response is thus a finite-length sequence of length 3:

$$\{h[n]\} = \{0.5, \underset{\uparrow}{1} \quad 0.5\}$$

Time-Domain Characterization of LTI Discrete-Time System

- **Input-Output Relationship** - A consequence of the linear, time-invariance property is that an LTI discrete-time system is completely characterized by its impulse response
- Knowing the impulse response one can compute the output of the system for any arbitrary input

Time-Domain Characterization of LTI Discrete-Time System

- Let $h[n]$ denote the impulse response of a LTI discrete-time system
- We compute its output $y[n]$ for the input:

$$x[n] = 0.5\delta[n+2] + 1.5\delta[n-1] - \delta[n-2] + 0.75\delta[n-5]$$

- As the system is linear, we can compute its outputs for each member of the input separately and add the individual outputs to determine $y[n]$

Time-Domain Characterization of LTI Discrete-Time System

- Since the system is time-invariant

input output

$$\delta[n+2] \rightarrow h[n+2]$$

$$\delta[n-1] \rightarrow h[n-1]$$

$$\delta[n-2] \rightarrow h[n-2]$$

$$\delta[n-5] \rightarrow h[n-5]$$

Time-Domain Characterization of LTI Discrete-Time System

- Likewise, as the system is linear

input output

$$0.5\delta[n+2] \rightarrow 0.5h[n+2]$$

$$1.5\delta[n-1] \rightarrow 1.5h[n-1]$$

$$-\delta[n-2] \rightarrow -h[n-2]$$

$$0.75\delta[n-5] \rightarrow 0.75h[n-5]$$

- Hence because of the linearity property we get

$$y[n] = 0.5h[n+2] + 1.5h[n-1] \\ -h[n-2] + 0.75h[n-5]$$

Time-Domain Characterization of LTI Discrete-Time System

- Now, any arbitrary input sequence $x[n]$ can be expressed as a linear combination of delayed and advanced unit sample sequences in the form

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

- The response of the LTI system to an input $x[k]\delta[n-k]$ will be $x[k]h[n-k]$

Time-Domain Characterization of LTI Discrete-Time System

- Hence, the response $y[n]$ to an input

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

is given by

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

which can be alternately written as

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

Convolution Sum

- The summation

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

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

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

Convolution Sum

- **Properties -**
- Commutative property:

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

- Associative property :

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

- Distributive property :

$$x[n] \circledast (h[n] + y[n]) = x[n] \circledast h[n] + x[n] \circledast y[n]$$

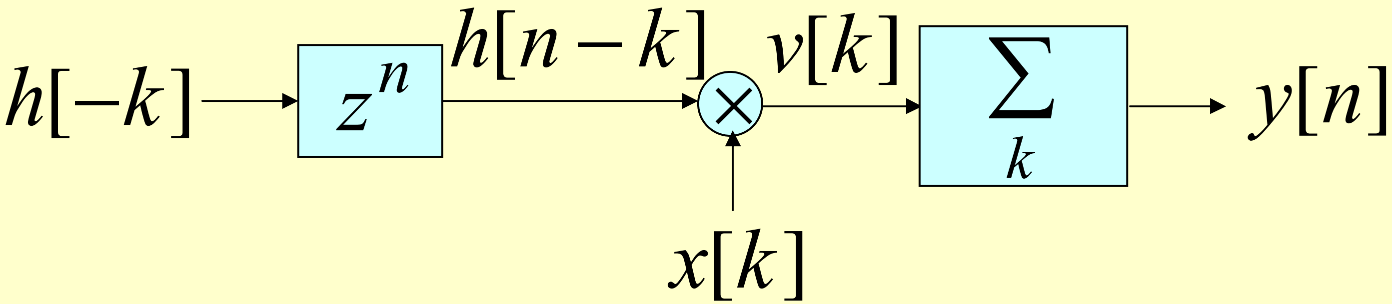
You sure?

Convolution Sum

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

Convolution Sum

- **Schematic Representation -**



- The computation of an output sample using the convolution sum is simply a sum of products
- Involves fairly simple operations such as additions, multiplications, and delays

Convolution Sum

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

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

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

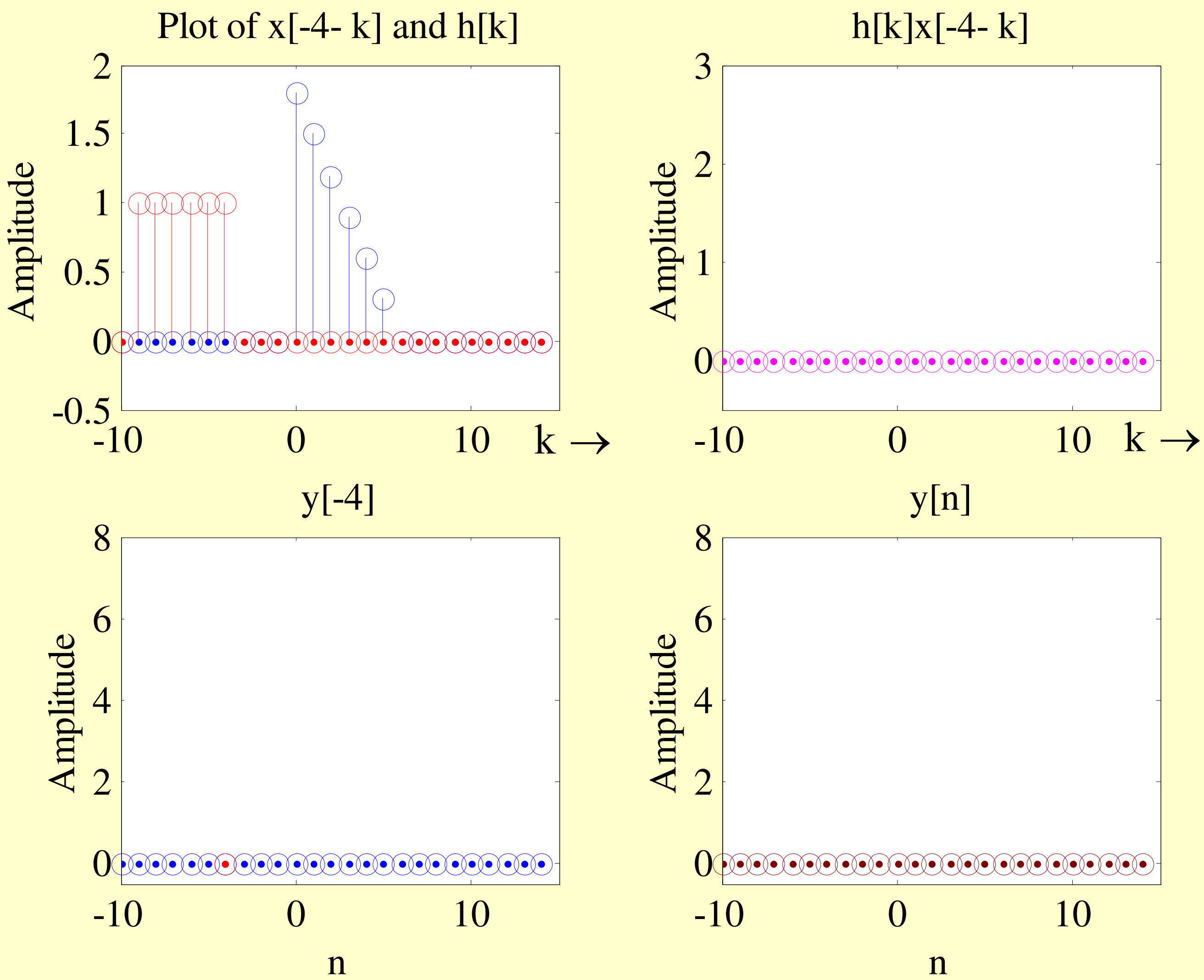
- Figures on the next several slides the steps involved in the computation of

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

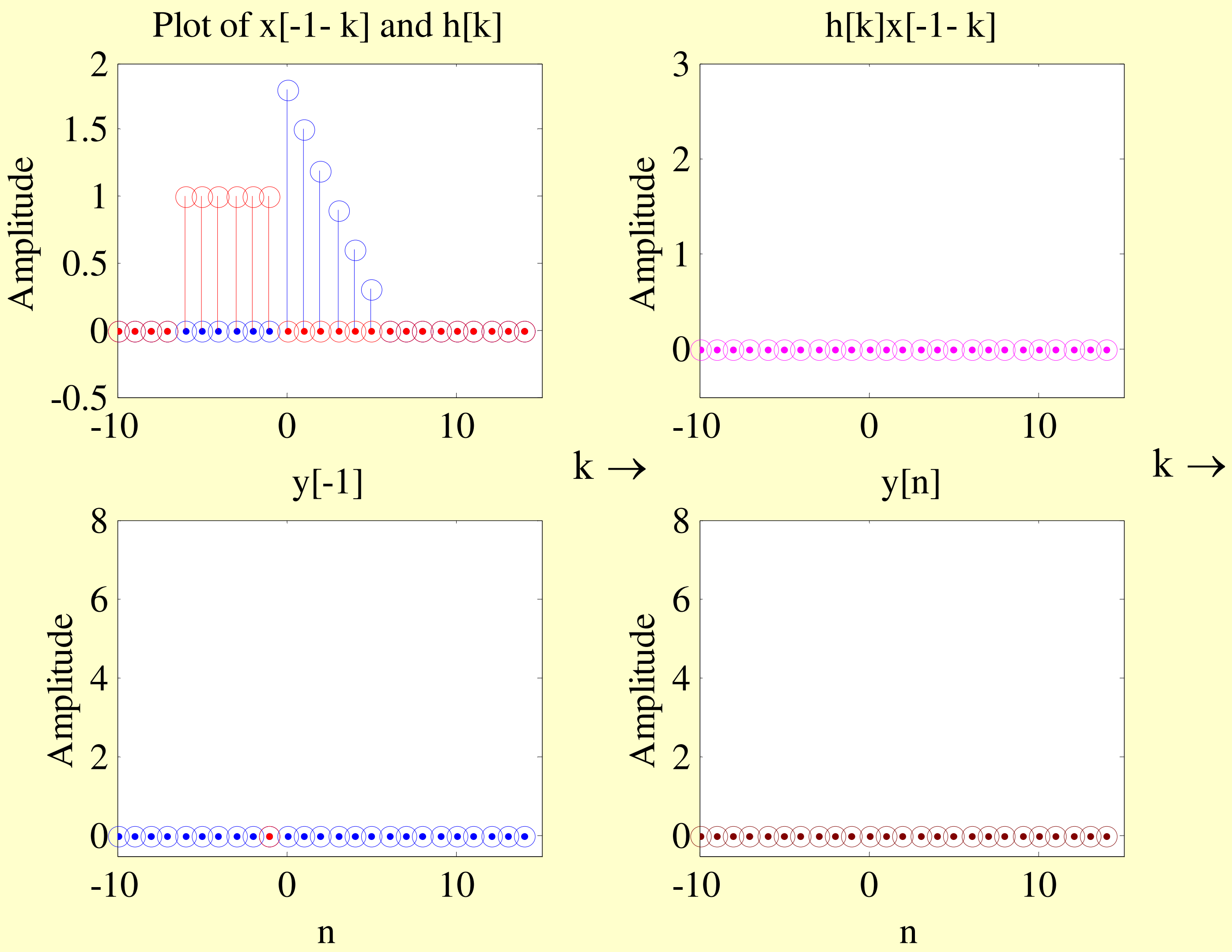
Convolution Sum

k

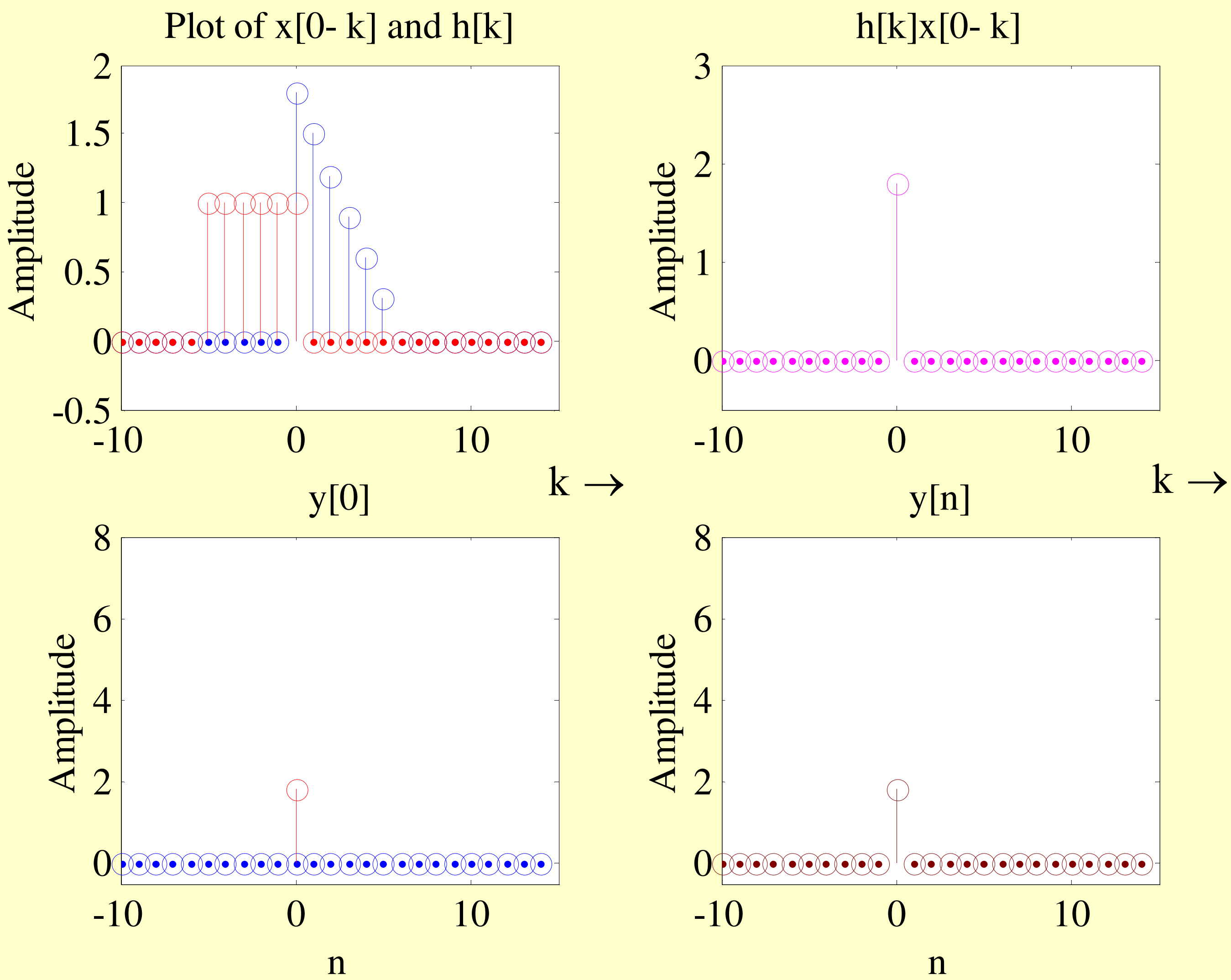
n



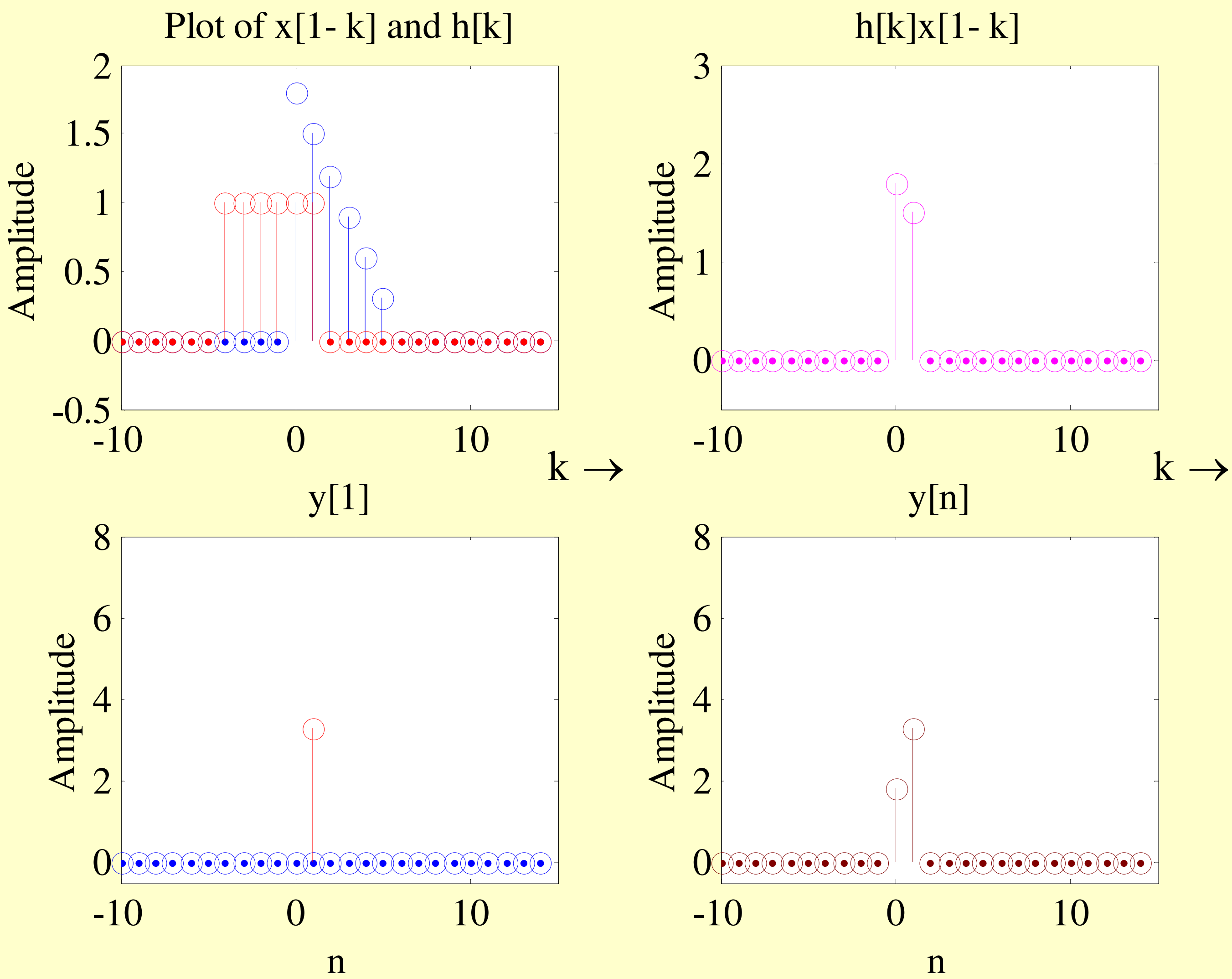
Convolution Sum



Convolution Sum

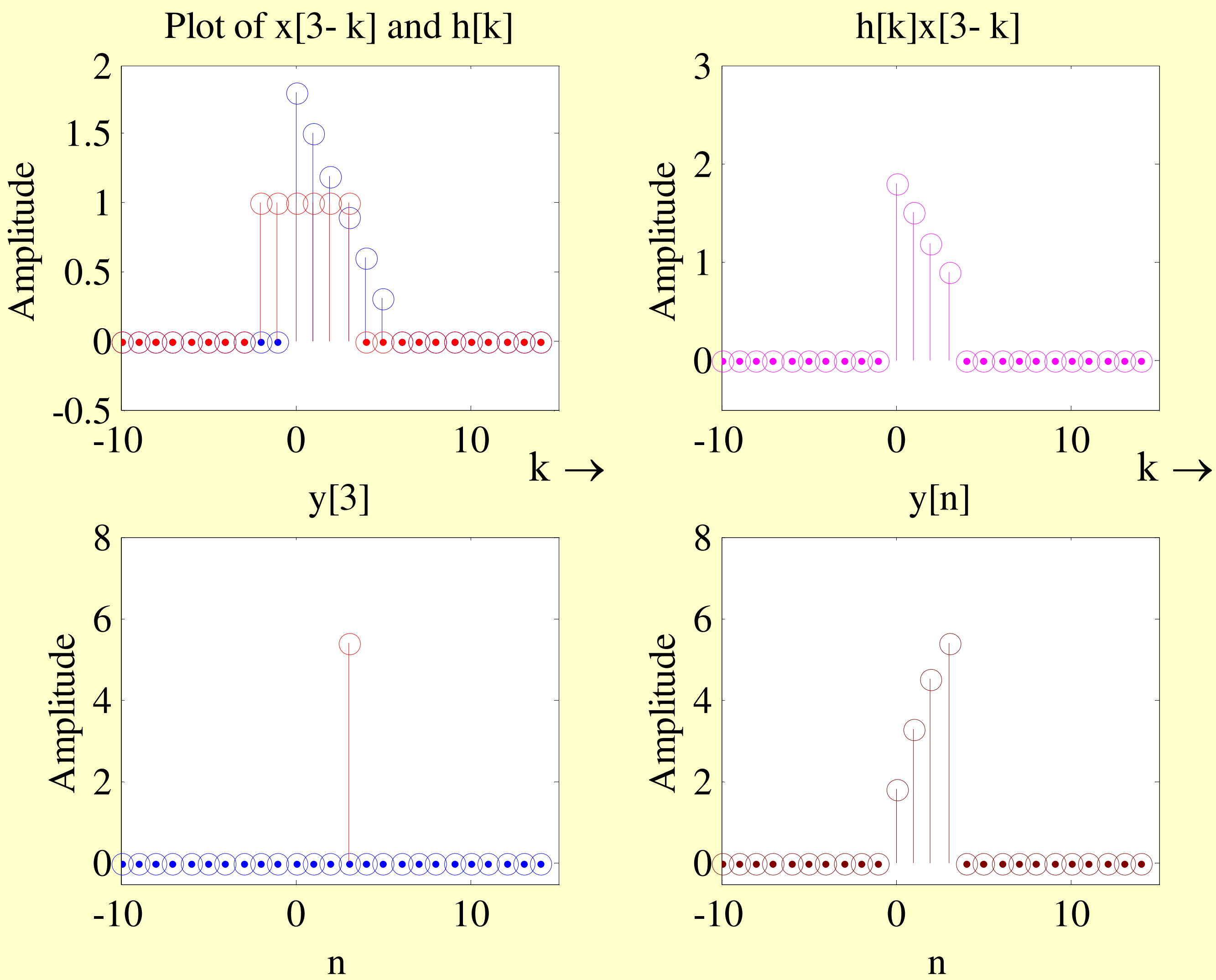


Convolution Sum

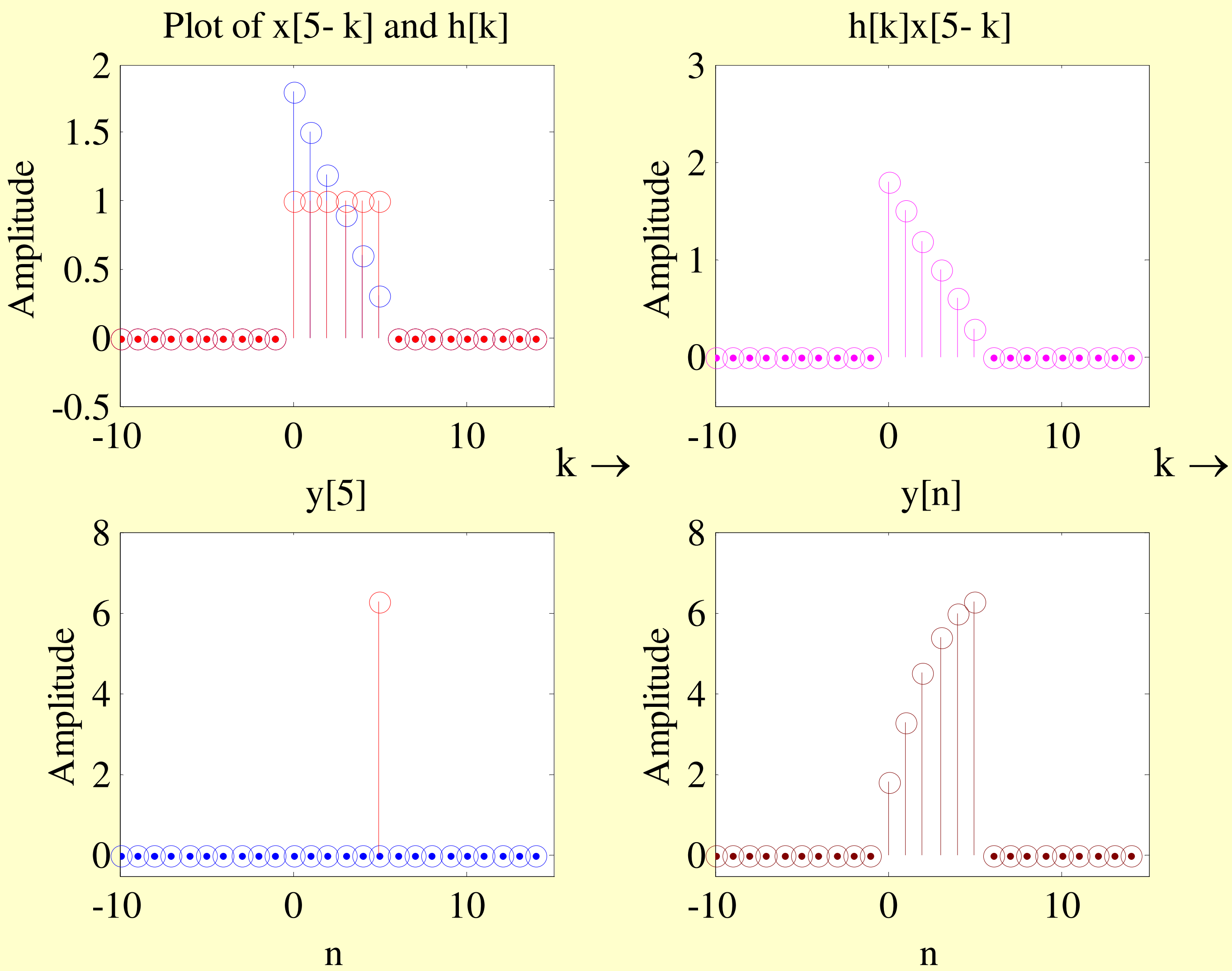


45

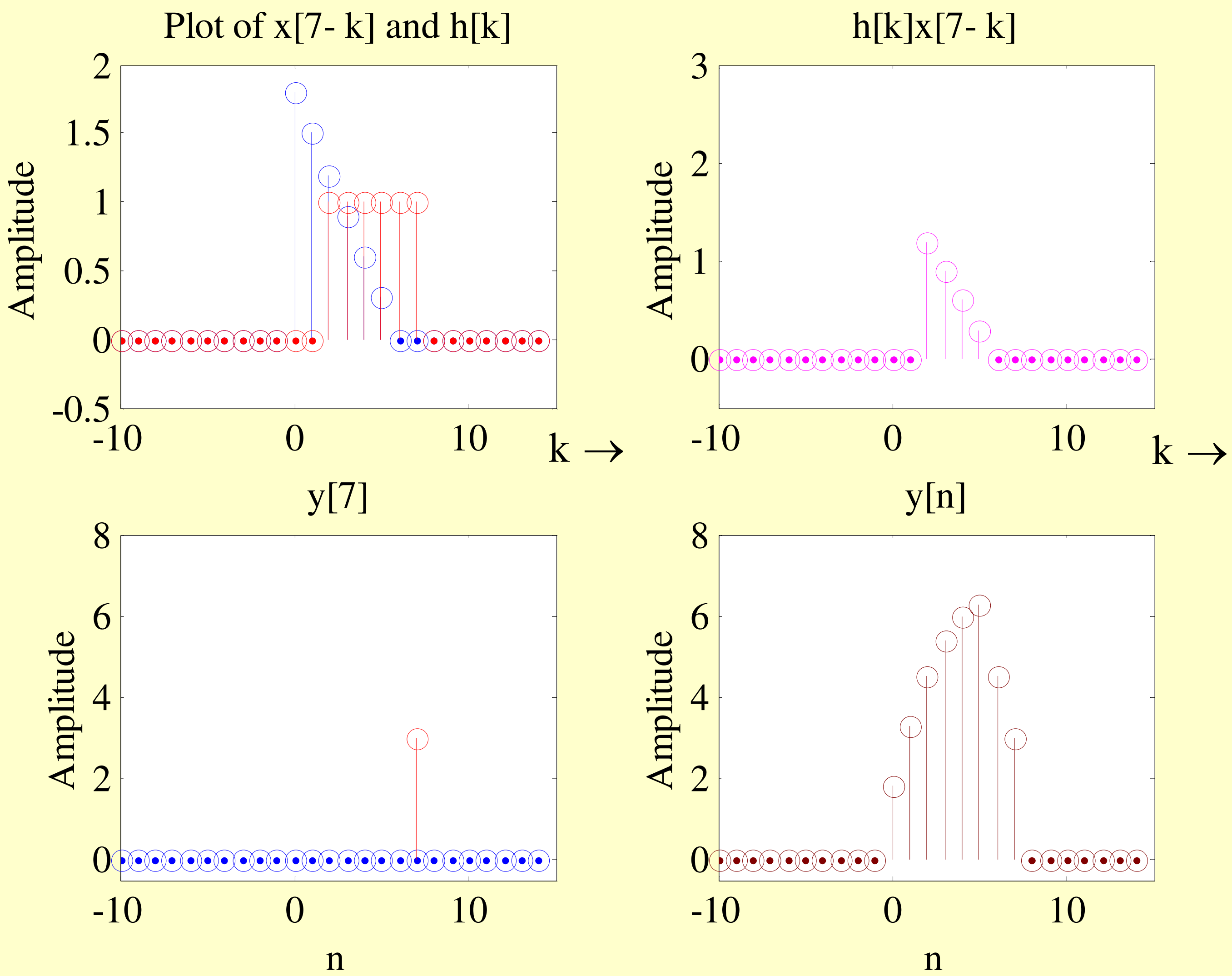
Convolution Sum



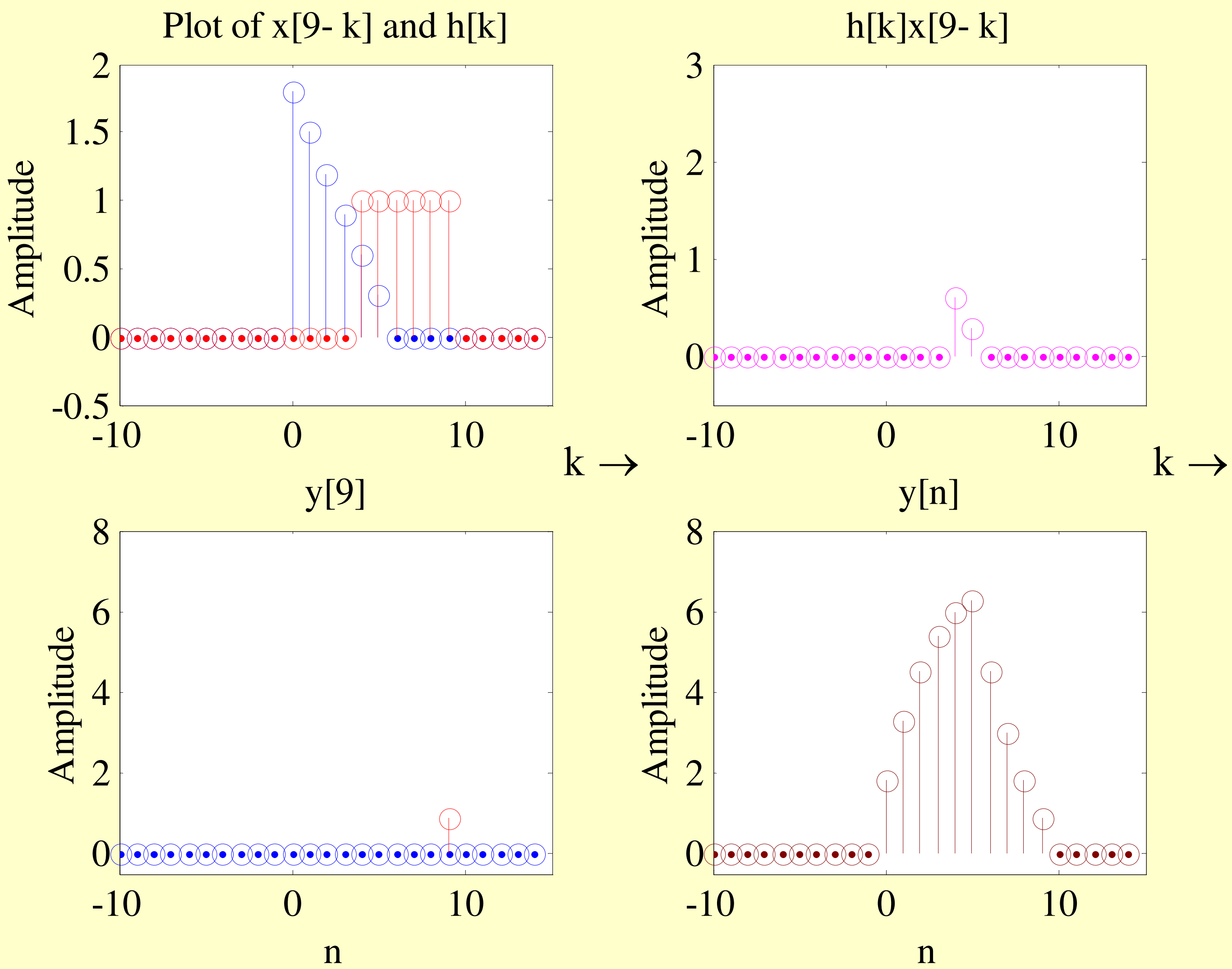
Convolution Sum



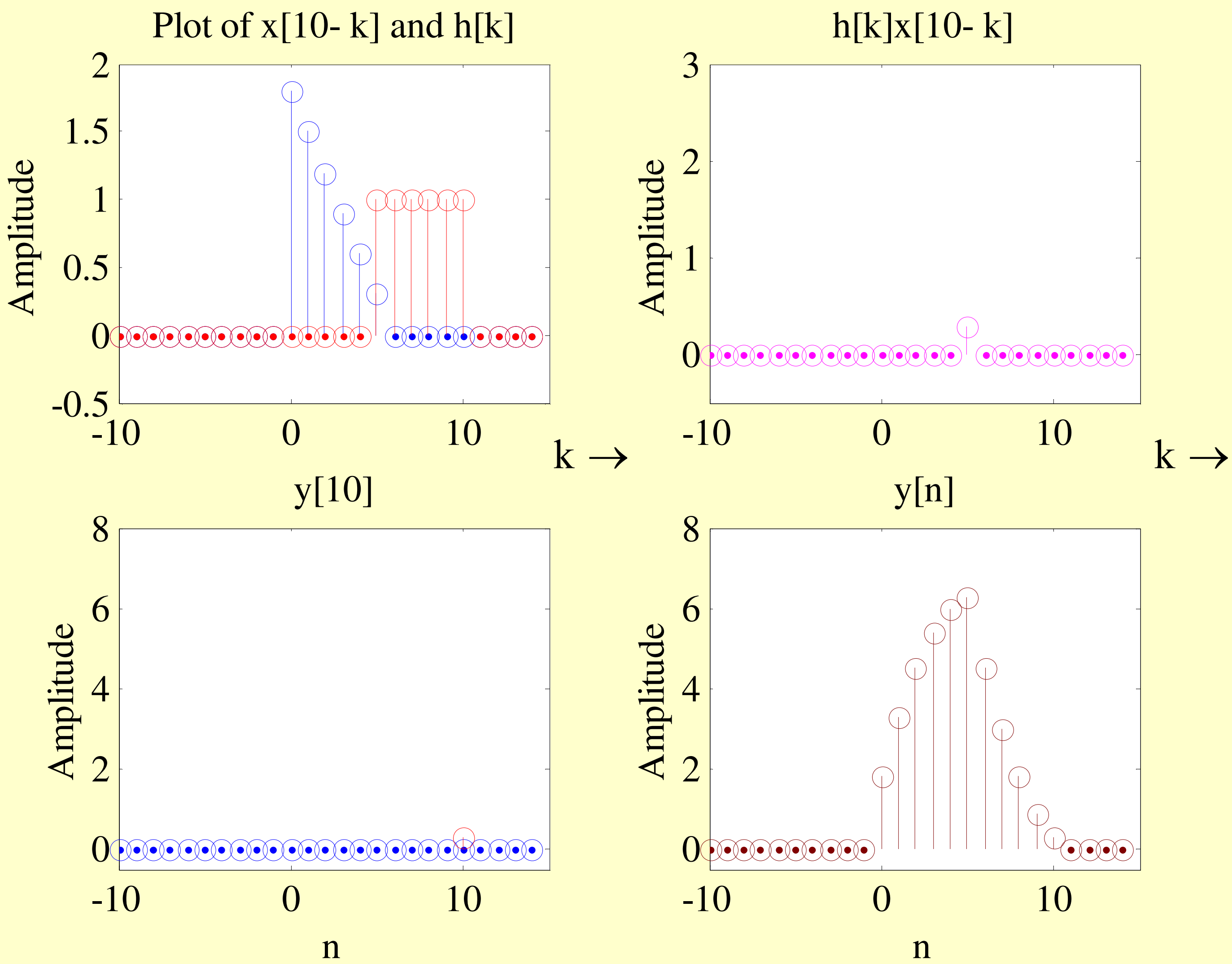
Convolution Sum



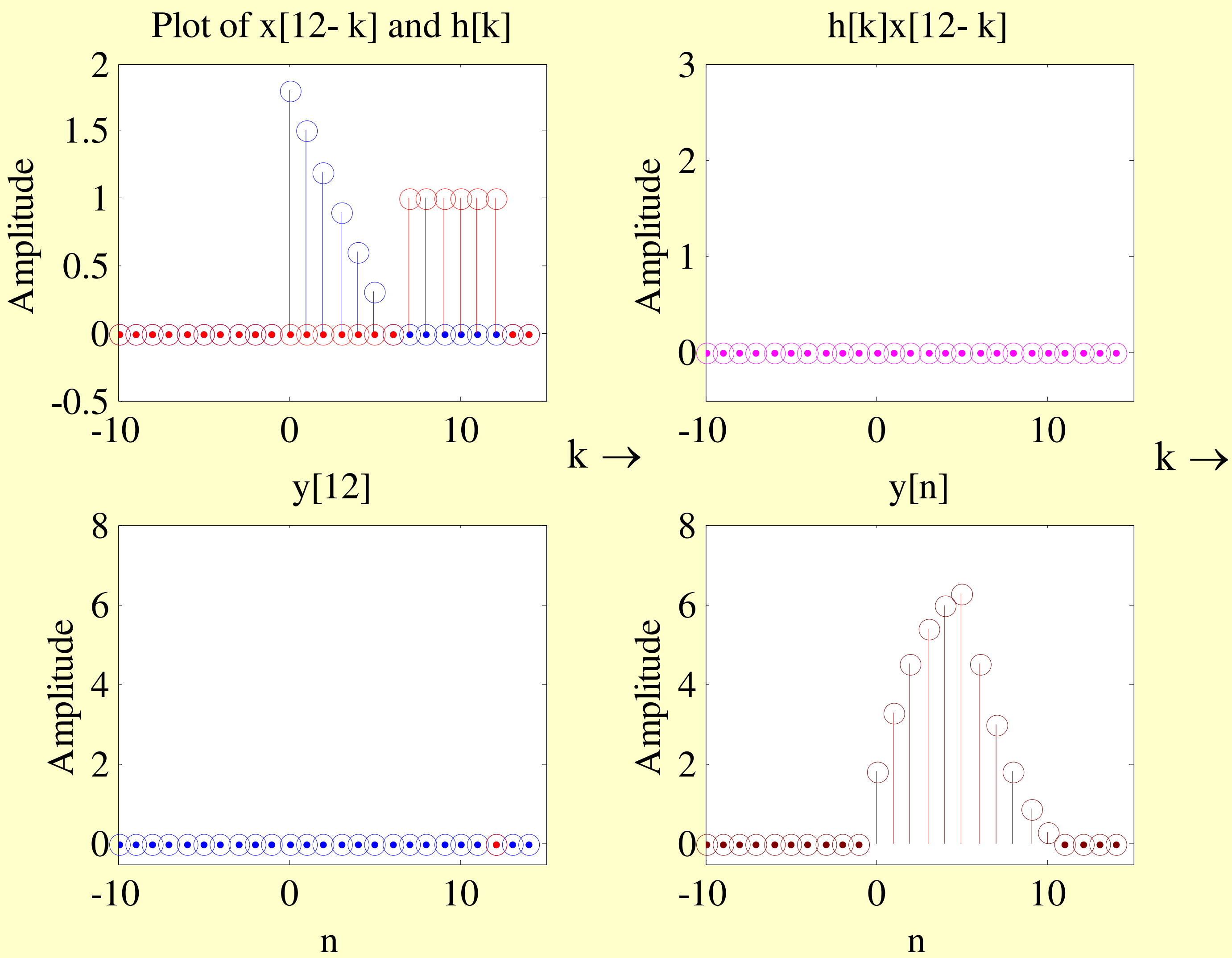
Convolution Sum



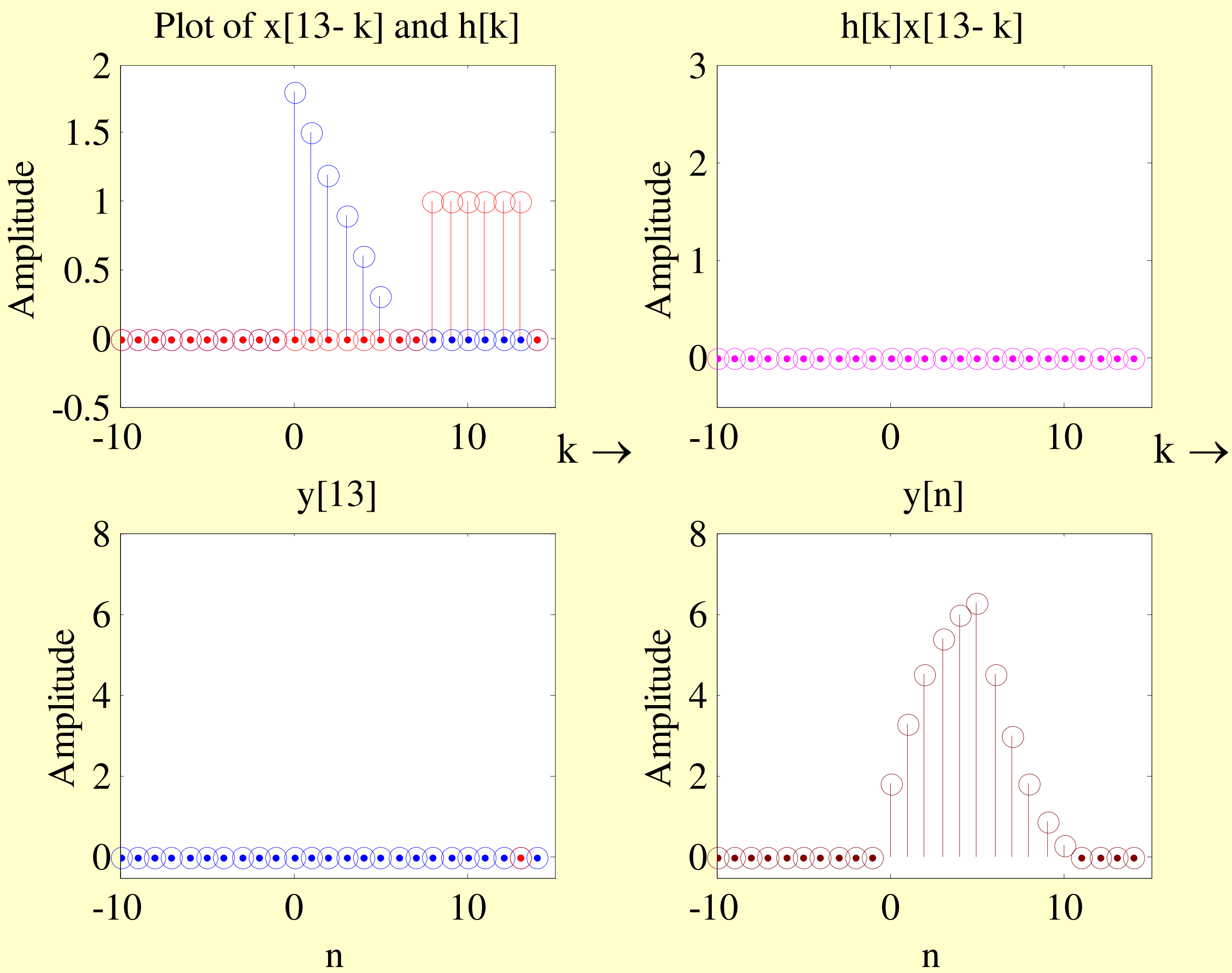
Convolution Sum



Convolution Sum

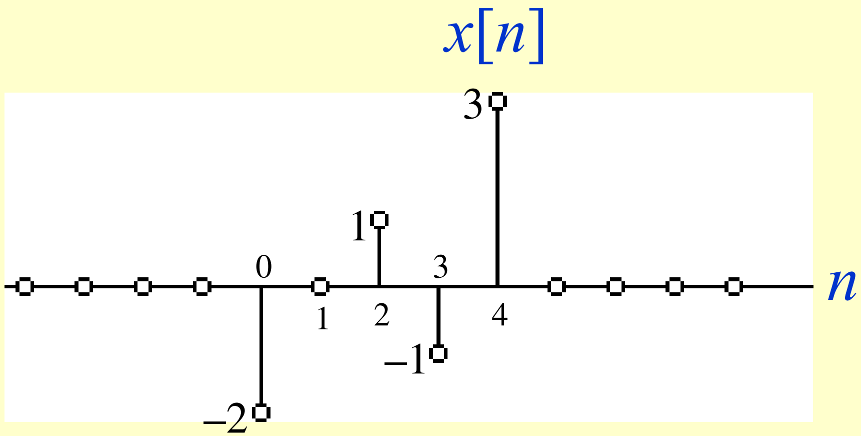


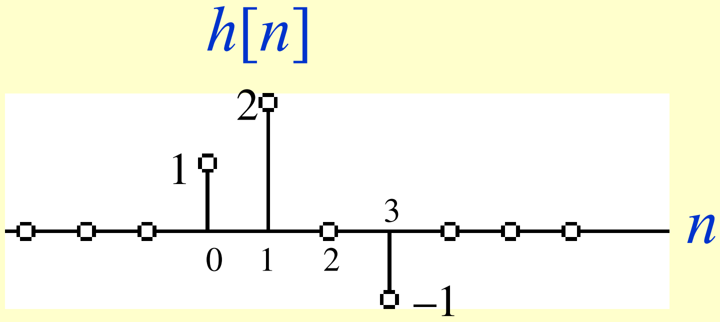
Convolution Sum



Convolution Sum

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

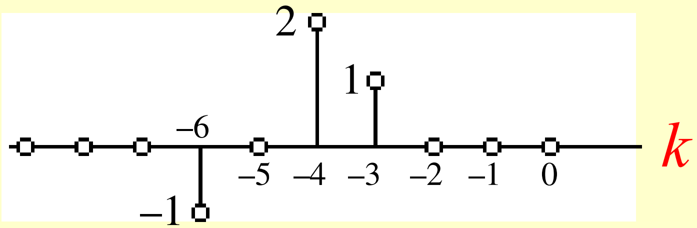




Convolution Sum

- 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

$h[-3-k]$

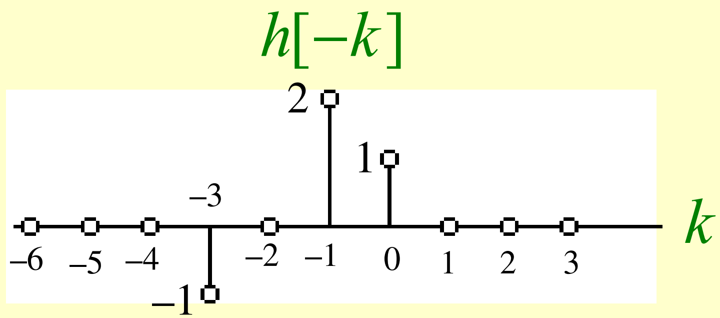


Convolution Sum

- 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

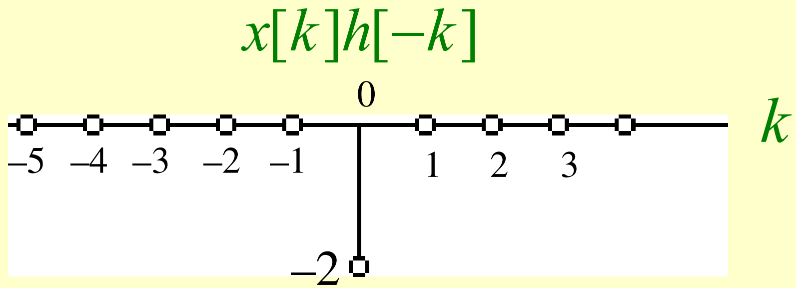
$$y[n] = 0 \qquad \text{for} \;\; n < 0$$

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



Convolution Sum

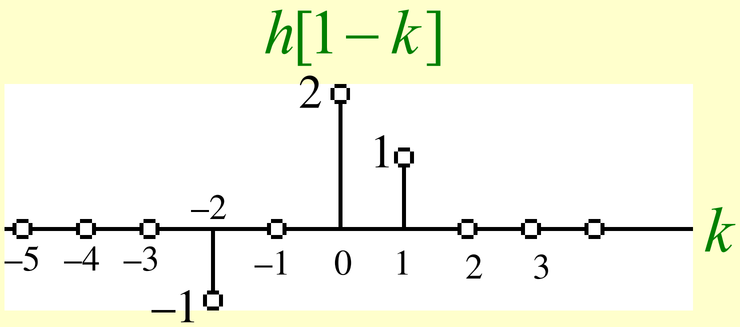
- The product sequence $\{x[k]h[-k]\}$ is plotted below which has a single nonzero sample $x[0]h[0]$ for $k = 0$

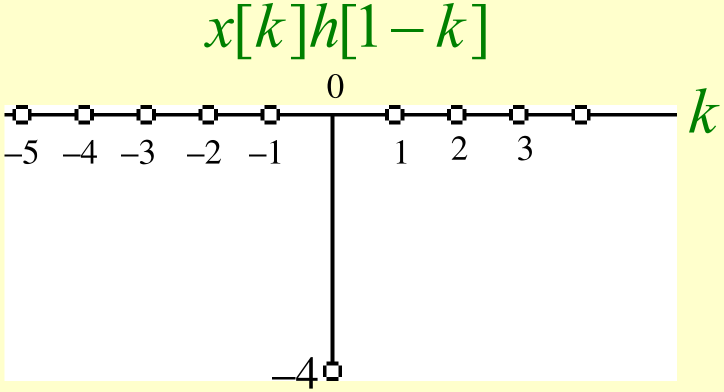


- Thus $y[0] = x[0]h[0] = -2$

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



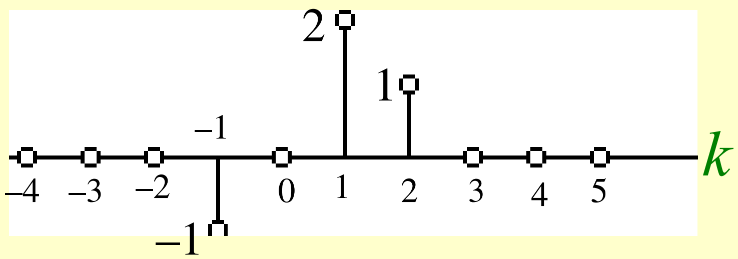


- Hence, $y[1] = x[0]h[1] + x[1]h[0] = -4 + 0 = -4$

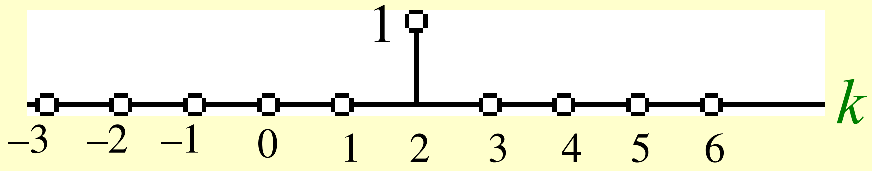
Convolution Sum

- 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

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

$$= 2 + 0 + 0 + 1 = 3$$

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

$$= 0 + 0 - 2 + 3 = 1$$

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

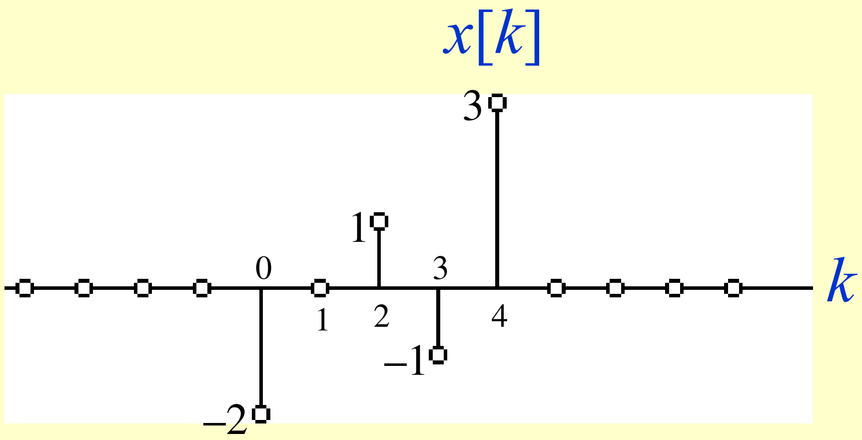
$$= -1 + 0 + 6 = 5$$

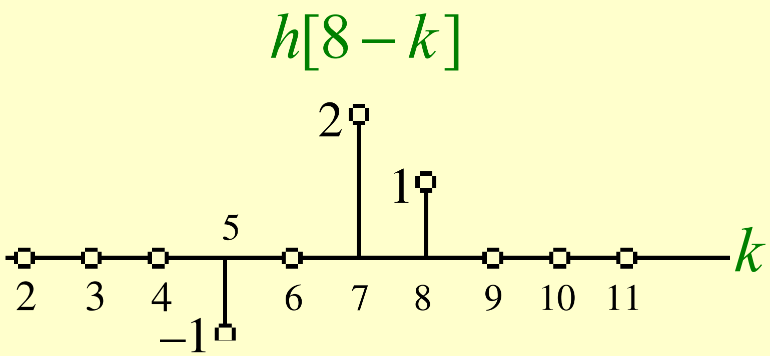
$$y[6] = x[3]h[3] + x[4]h[2] = 1 + 0 = 1$$

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

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 $n > 7$

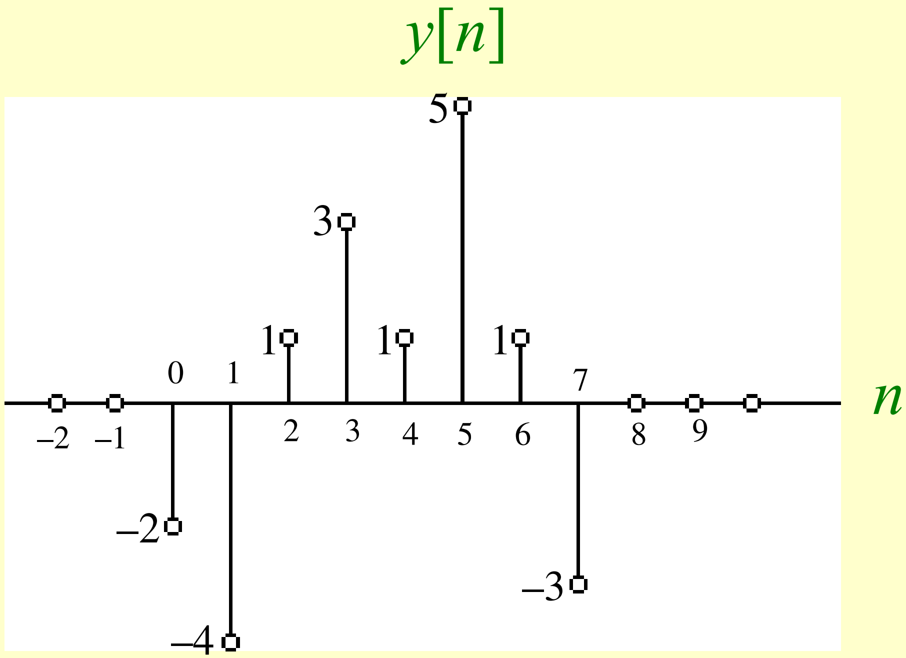




60

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 $M + N - 1$

Time-Domain Characterization of LTI Discrete-Time System

- In practice, if either the input or the impulse response is of finite length, the convolution sum can be used to compute the output sample as it involves a finite sum of products
- If both the input sequence and the impulse response sequence are of finite length, the output sequence is also of finite length

Time-Domain Characterization of LTI Discrete-Time System

- If both the input sequence and the impulse response sequence are of infinite length, convolution sum cannot be used to compute the output
- For systems characterized by an infinite impulse response sequence, an alternate time-domain description involving a finite sum of products will be considered

Convolution Using MATLAB

- The M-file `conv` implements the convolution sum of two finite-length sequences
- If $\mathtt{a=[-2\ \ 0\ \ 1\ \ -1\ \ 3]}$

  $\mathtt{b=[1\ \ 2\ \ 0\ \ -1]}$

  then `conv(a,b)` yields

$$\mathtt{[-2\ \ -4\ \ 1\ \ 3\ \ 1\ \ 5\ \ 1\ \ -3]}$$

Simple Interconnection Schemes

- Two simple interconnection schemes are:
- Cascade Connection
- Parallel Connection

Cascade Connection

$$\longrightarrow \boxed{h_1[n]} \longrightarrow \boxed{h_2[n]} \longrightarrow \;\equiv\; \longrightarrow \boxed{h_2[n]} \longrightarrow \boxed{h_1[n]} \longrightarrow$$

$$\equiv \longrightarrow \boxed{h_1[n] \circledast h_2[n]} \longrightarrow$$

- Impulse response $h[n]$ of the cascade of two LTI discrete-time systems with impulse responses $h_1[n]$ and $h_2[n]$ is given by

$$h[n] = h_1[n] \circledast h_2[n]$$

Cascade Connection

- Note: The ordering of the systems in the cascade has no effect on the overall impulse response because of the commutative property of convolution
- A cascade connection of two stable systems is stable
- A cascade connection of two passive (lossless) systems is passive (lossless)

Cascade Connection

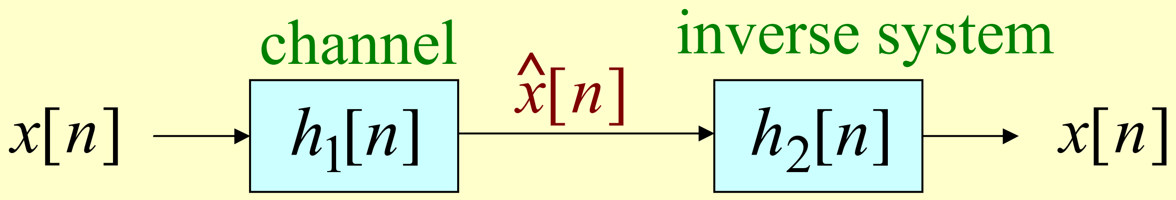
- An application is in the development of an **inverse system**
- If the cascade connection satisfies the relation

$$h_1[n] \circledast h_2[n] = \delta[n]$$

then the LTI system $h_1[n]$ is said to be the inverse of $h_2[n]$ and vice-versa

Cascade Connection

- An application of the inverse system concept is in the recovery of a signal $x[n]$ from its distorted version $\hat{x}[n]$ appearing at the output of a transmission channel
- If the impulse response of the channel is known, then $x[n]$ can be recovered by designing an inverse system of the channel



$$h_1[n] \circledast h_2[n] = \delta[n]$$

$$\begin{array}{ccccc} \mu(n) & & h_2(n) & & \delta(n) \\ \ldots\ 0\ 0\ 0\ \underset{\uparrow}{1}\ 1\ 1\ \ldots & \circledast & \ldots\ a\ b\ c\ \underset{\uparrow}{d}\ e\ f\ g\ \ldots & = & \ldots\ 0\ 0\ 0\ \underset{\uparrow}{1}\ 0\ 0\ 0\ \ldots \end{array}$$

- <u>Example</u> - Consider the discrete-time accumulator with an impulse response $\mu[n]$
- Its inverse system satisfy the condition

$$\mu[n] \circledast h_2[n] = \delta[n]$$

- It follows from the above that $h_2[n] = 0$ for $n < 0$ and

$$h_2[0] = 1$$

$$\sum_{\ell=0}^{n} h_2[\ell] = 0 \quad \text{for} \quad n \geq 1$$

Cascade Connection

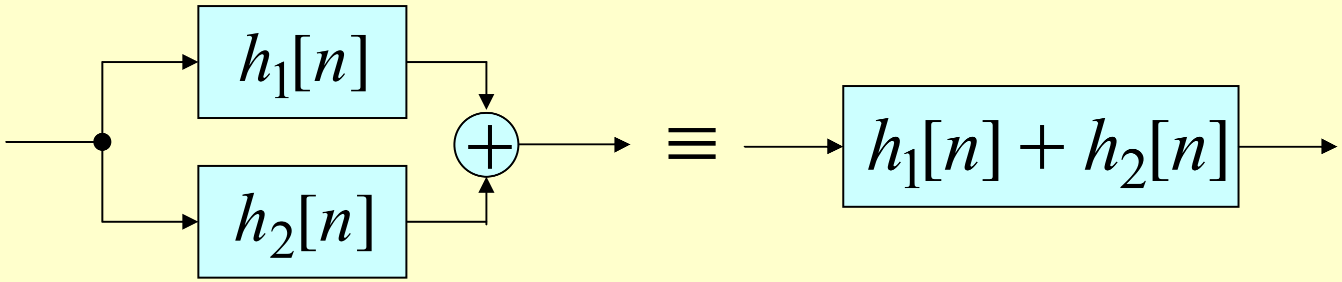
- Thus the impulse response of the inverse system of the discrete-time accumulator is given by

$$h_2[n] = \delta[n] - \delta[n-1] \quad \boxed{= \{\underset{\uparrow}{1}, -1\}}$$

which is called a **backward difference system**

$\boxed{\text{I/O relation is } y[n] = x[n] - x[n-1]}$

Parallel Connection



- Impulse response $h[n]$ of the parallel connection of two LTI discrete-time systems with impulse responses $h_1[n]$ and $h_2[n]$ is given by

$$h[n] = h_1[n] + h_2[n]$$

Simple Interconnection Schemes

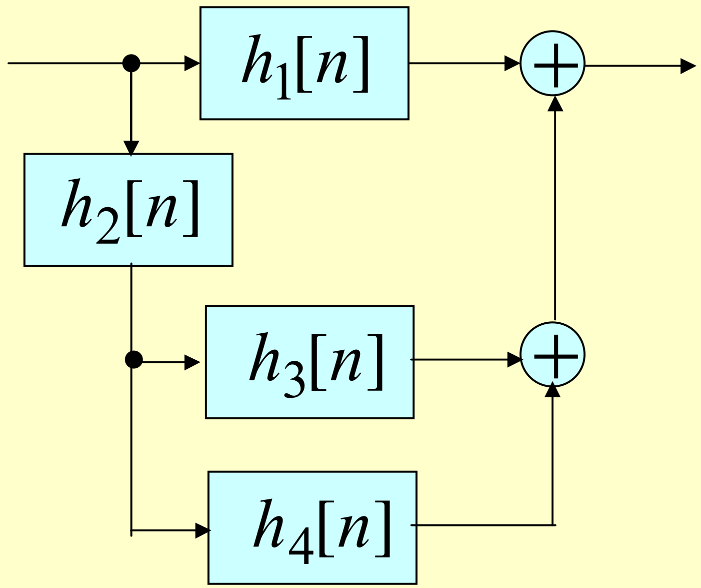
- Consider the discrete-time system where

$$h_1[n] = \delta[n] + 0.5\delta[n-1],$$

$$h_2[n] = 0.5\delta[n] - 0.25\delta[n-1],$$

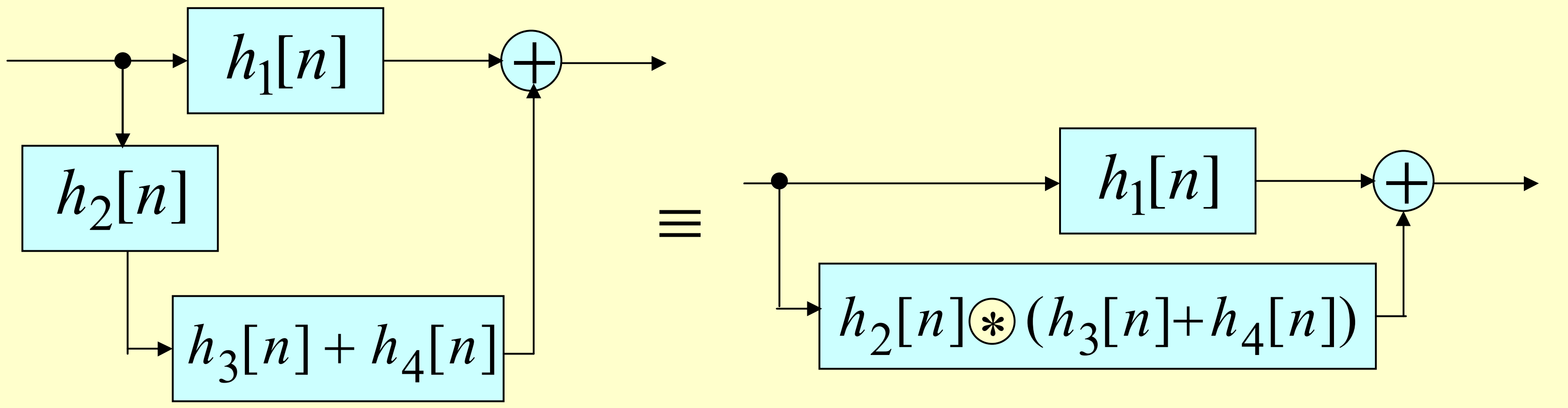
$$h_3[n] = 2\delta[n],$$

$$h_4[n] = -2(0.5)^n \mu[n]$$



Simple Interconnection Schemes

- Simplifying the block-diagram we obtain



Simple Interconnection Schemes

- Overall impulse response $h[n]$ is given by

$$h[n] = h_1[n] + h_2[n] \circledast (h_3[n] + h_4[n])$$
$$= h_1[n] + h_2[n] \circledast h_3[n] + h_2[n] \circledast h_4[n]$$

- Now,

$$h_2[n] \circledast h_3[n] = (\tfrac{1}{2}\delta[n] - \tfrac{1}{4}\delta[n-1]) \circledast 2\delta[n]$$
$$= \delta[n] - \tfrac{1}{2}\delta[n-1]$$

Simple Interconnection Schemes

$$h_2[n] \circledast h_4[n] = (\tfrac{1}{2}\delta[n] - \tfrac{1}{4}\delta[n-1]) \circledast \left(-2(\tfrac{1}{2})^n \mu[n]\right)$$

$$= -(\tfrac{1}{2})^n \mu[n] + \tfrac{1}{2}(\tfrac{1}{2})^{n-1}\mu[n-1]$$

$$= -(\tfrac{1}{2})^n \mu[n] + (\tfrac{1}{2})^n \mu[n-1]$$

$$= -(\tfrac{1}{2})^n \delta[n] = -\delta[n]$$

- Therefore

$$h[n] = \delta[n] + \tfrac{1}{2}\delta[n-1] + \delta[n] - \tfrac{1}{2}\delta[n-1] - \delta[n] = \delta[n]$$