CO2035

2. Discrete-Time Signal and System

anhpham (at) hcmut (dot) edu (dot) vn

Contents

Contents (con't)

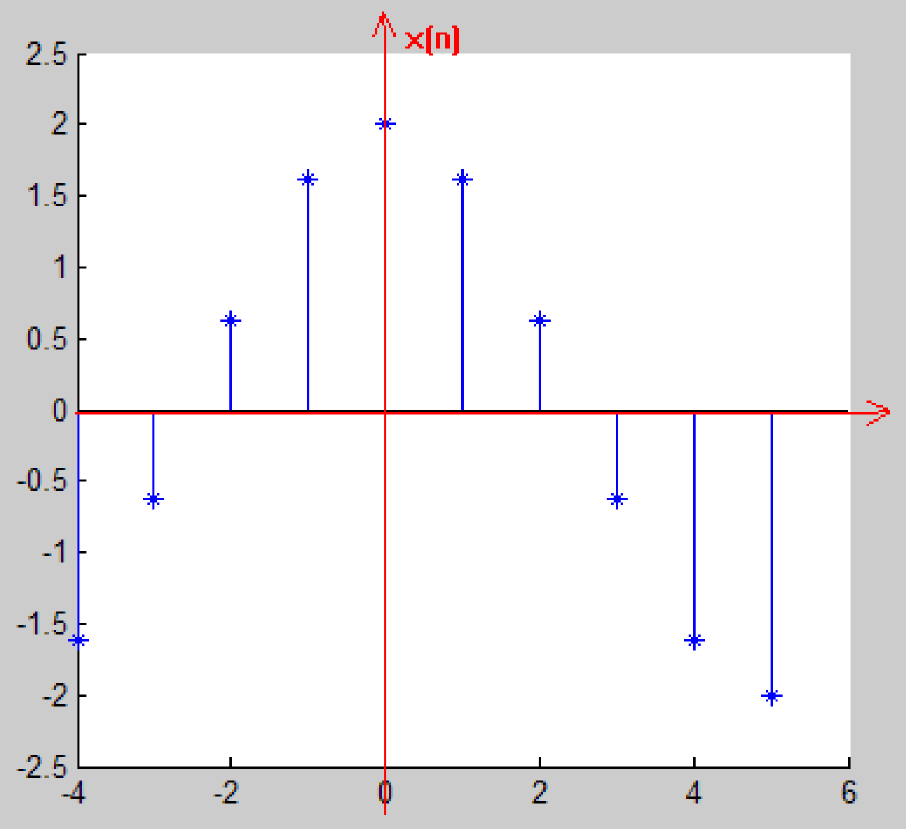
- **Difference Equations**
  - LTI Systems Characterized by Constant-Coefficient Difference Equations.
  - Solve the Constant-Coefficient Difference Equation.
  - Impulse Response of A Recursive LTI System

- **Implementation of Discrete-Time Systems**
  - Direct Form I Structure
  - Direct Form II Structure

Discrete-Time Signals

- Discrete-Time Signal x(n) is a function of an independent variable that is an integer (n € Z)
  - x(n) is not defined for non-integer values of n. It is incorrect to think that x(n) is equal to zero if n is not an integer.

- $x(n) = x_a(nT_s)$
  - x_a: corresponding analog signal
  - T_s: sampling cycle



Discrete-Time Signals

- Functional representation

$$x(n) = \begin{cases} 1, & \text{for } n = 1, 3 \\ 4, & \text{for } n = 2 \\ 0, & \text{elsewhere} \end{cases}$$

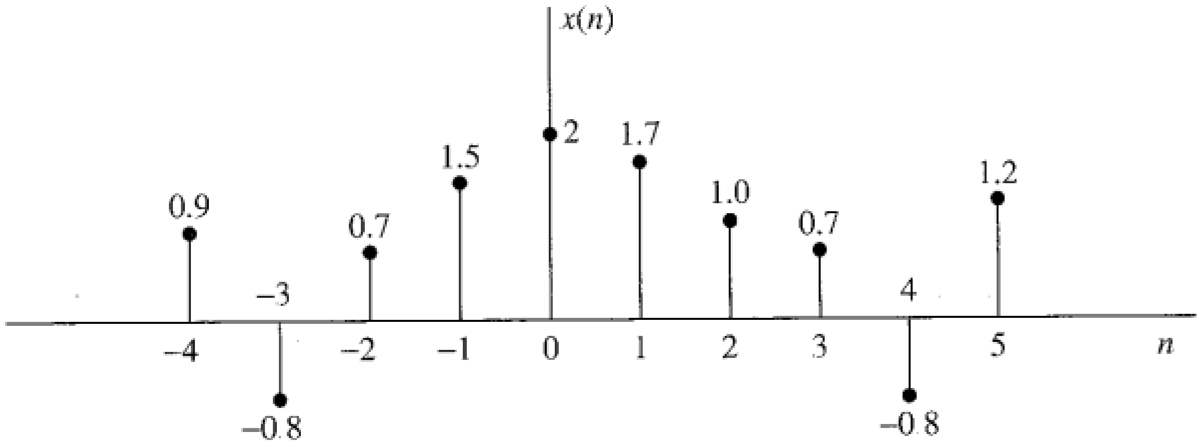
- Tabular representation

n	...	-2	-1	0	1	2	3	4	5	...
x(n)	...	0	0	0	1	4	1	0	0	...

- Sequence representation
  - The time origin (n=0) is indicated by symbol ↑ or *.

$$x(n) = \{\ldots 0, \underset{\uparrow}{0}, 1, 4, 1, 0, 0, \ldots\}$$

- Graphical representation



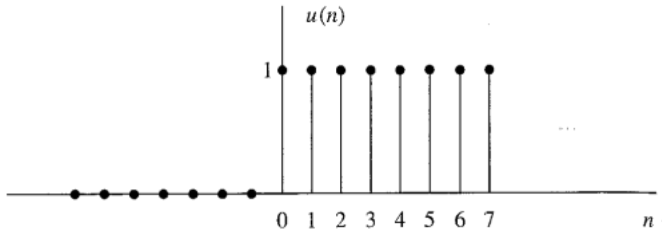
Elementary Discrete-Time Signals

- **Unit sample sequence (impulse)**

$$\delta(n) = \begin{cases} 1, & \text{for } n = 0 \\ 0, & \text{for } n \neq 0 \end{cases}$$

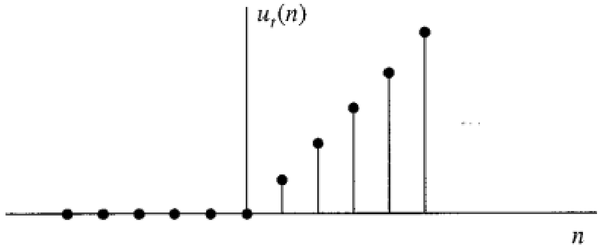
- **Unit step signal**

$$u(n) = \begin{cases} 1, & \text{for } n \geq 0 \\ 0, & \text{for } n < 0 \end{cases}$$



- **Unit ramp signal**

$$u_r(n) = \begin{cases} n, & \text{for } n \geq 0 \\ 0, & \text{for } n < 0 \end{cases}$$



- **Note:**

$$\begin{aligned} \delta(n) &= u(n) - u(n-1) = u_r(n+1) - 2u_r(n) + u_r(n-1) \\ u(n) &= u_r(n+1) - u_r(n) \end{aligned}$$

Exponential Signal

- Defined as
  - $x(n) = a^n, \forall n$
  - If a is real

    $\rightarrow$ x(n): real signal
  - If a is complex valued, it can be expressed as $a \equiv re^{j\theta}$

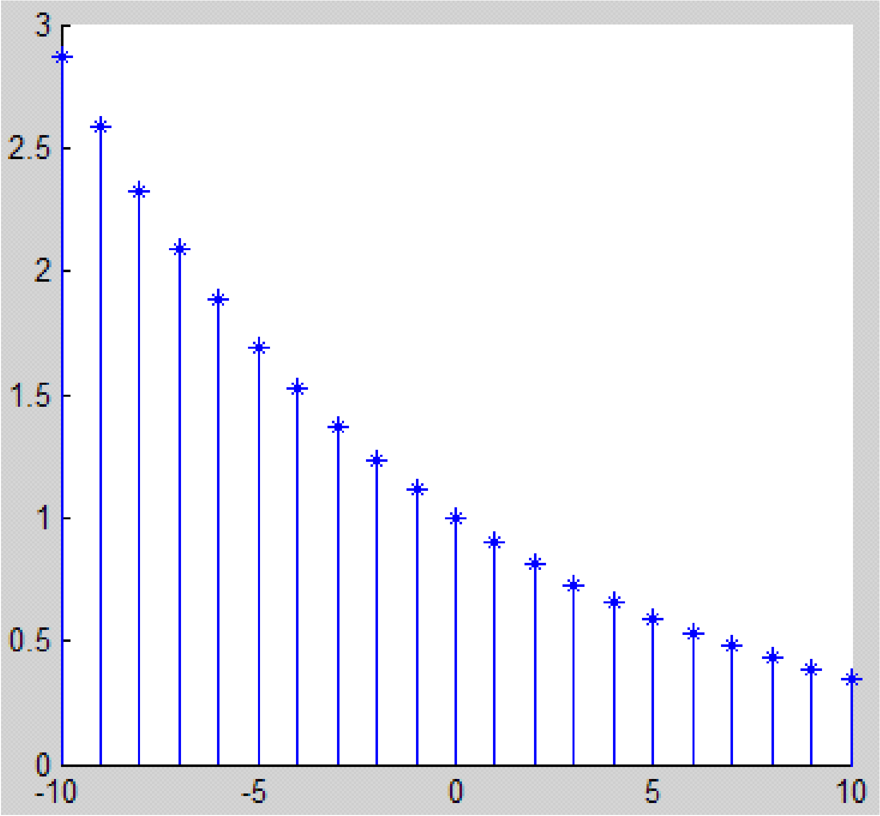
    $\rightarrow x(n) = r^n e^{j\theta n}$

    $= r^n(\cos\theta n + j\sin\theta n)$

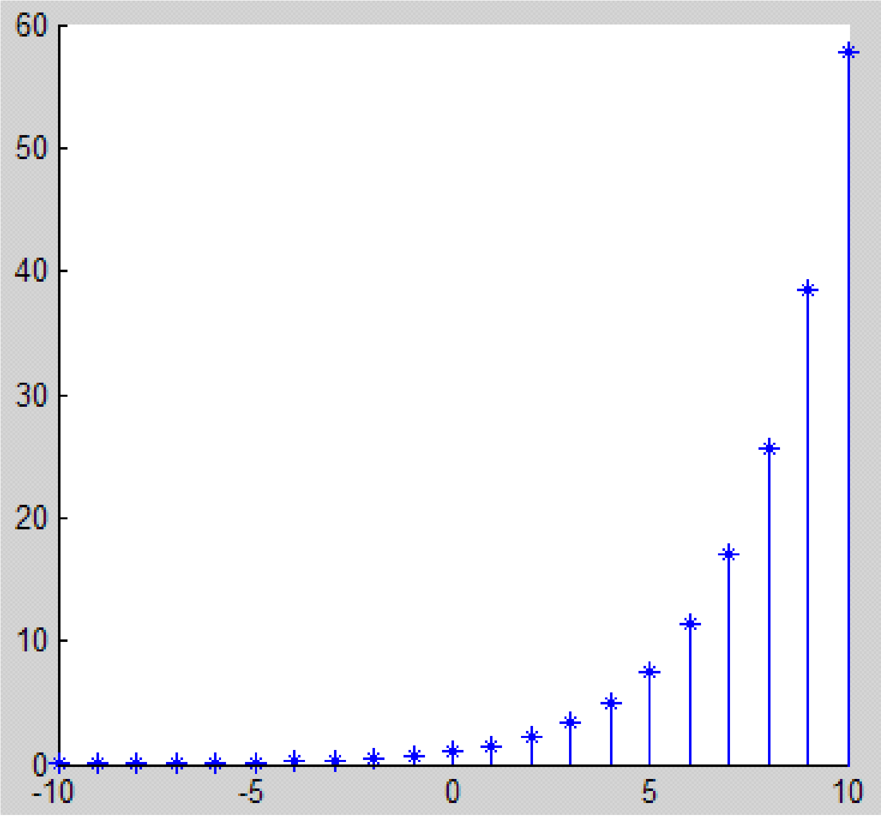
    - x(n) can be expressed in two forms

$$\begin{cases} x_R(n) = r^n cos\theta n \\ x_I(n) = r^n sin\theta n \end{cases} \qquad \begin{cases} |x(n)| = r^n \\ \angle x(n) = \theta n \end{cases}$$

Exponential Signal

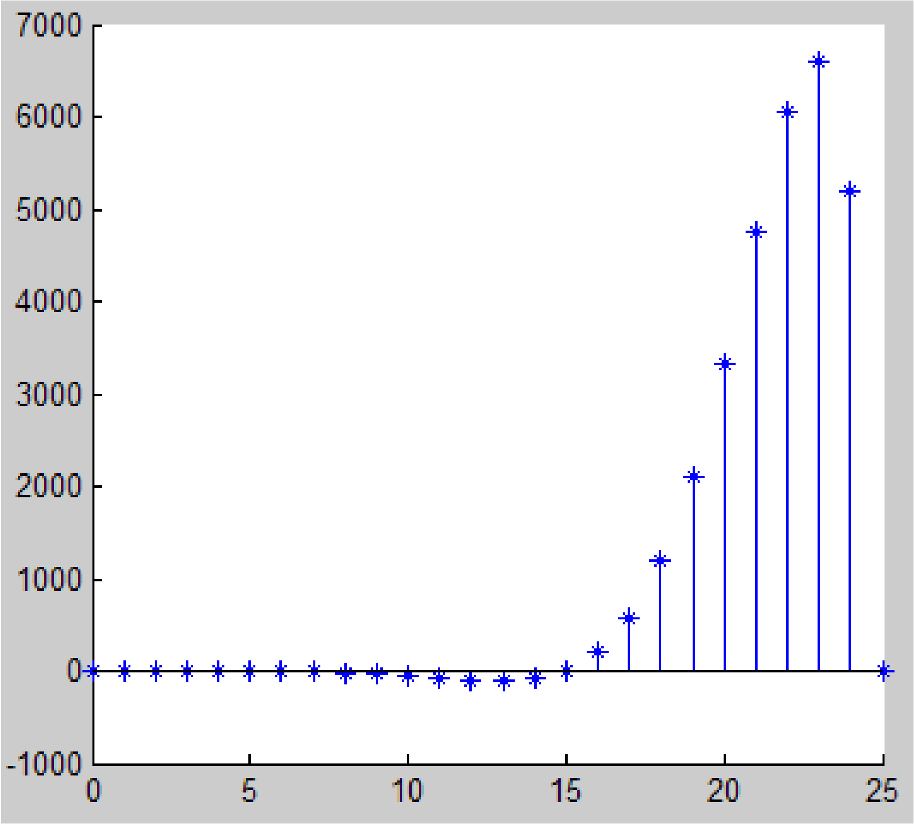


$x(n)=a^n$ (where a=0.9)

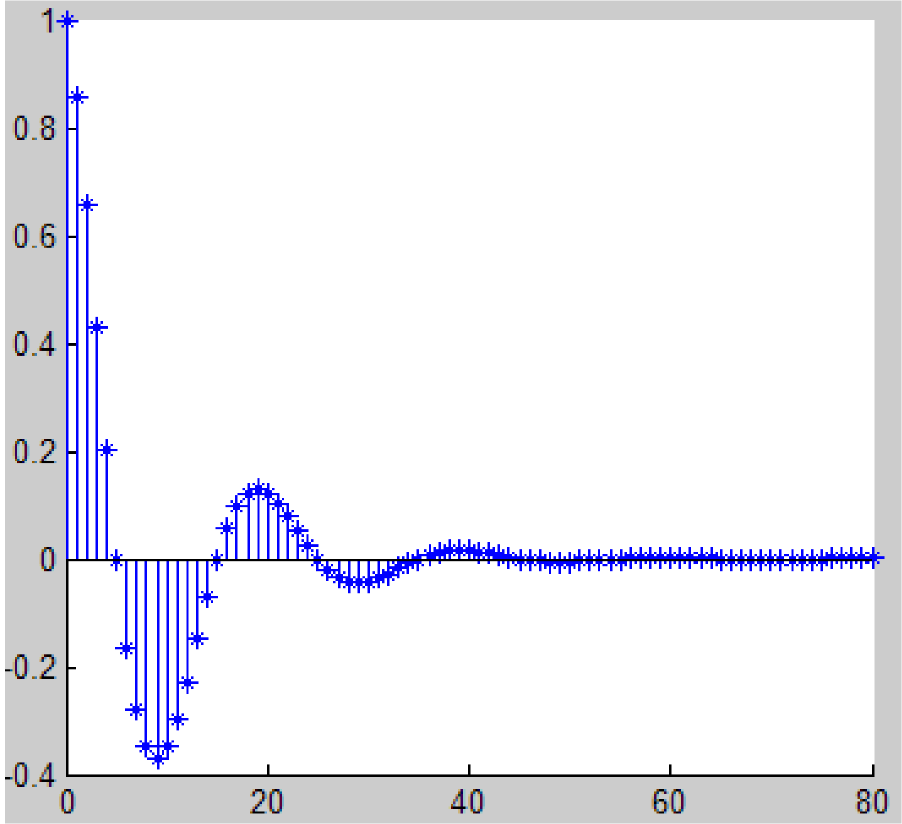


$x(n)=a^n$ (where a=1.5)

Exponential Signal



$x_r(n) = (1.5)^n \cos(\pi n/10)$



$x_r(n) = (0.9)^n \cos(\pi n/10)$

Classification of Discrete-Time Signals

- Energy Signal
- Power Signal
- Periodic Signal
- Aperiodic Signal

Energy Signal and Power Signal

- **The energy E_x of the signal x(n)**

$$\mathbf{E_x = \sum_{-\infty}^{+\infty} |x(n)|^2}$$

  - If E_x is finite ($0 < E_x < \infty$) → x(n): Energy signal

- **The average power P of the signal x(n)**

$$\mathbf{P_x = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x(n)|^2}$$

  - If P_x is finite ($0 < P_x < \infty$) → x(n): Power signal

- **The signal energy of x(n) over a finite interval [-N, N]**

$$\mathbf{E_N = \sum_{-N}^{N} |x(n)|^2}$$

  - The signal energy $E = \lim_{N\to\infty} E_N$
  - The signal power $P = \lim_{N\to\infty} \frac{1}{2N+1} E_N$

Periodic Signal

- A signal x(n) is periodic with a period N (N>0) if and only if
  - $x(n + N) = x(n), \forall n$
- The signal energy is
  - finite if
    - $0 \le n \le N - 1$
    - x(n) is finite
  - Infinite if
    - $-\infty \le n \le +\infty$
- The signal power is finite

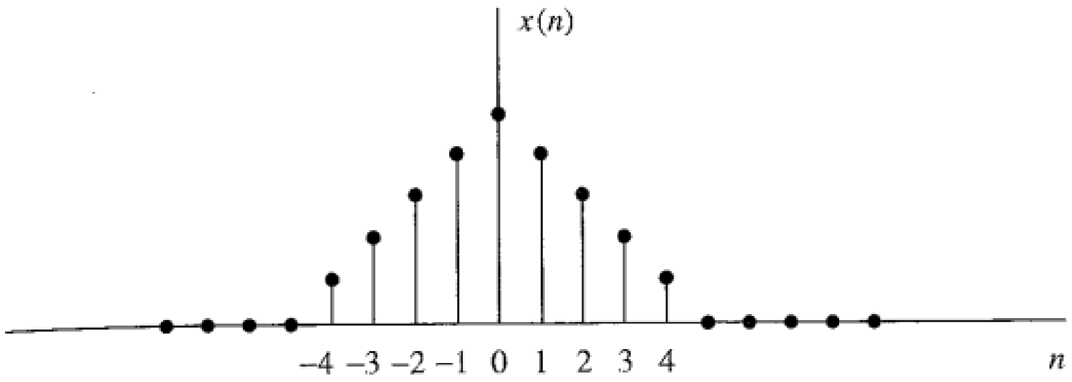
Periodic signals are power signals.

$$\mathbf{P = \frac{1}{N}\sum_{n-0}^{N-1}|x(n)|^2}$$

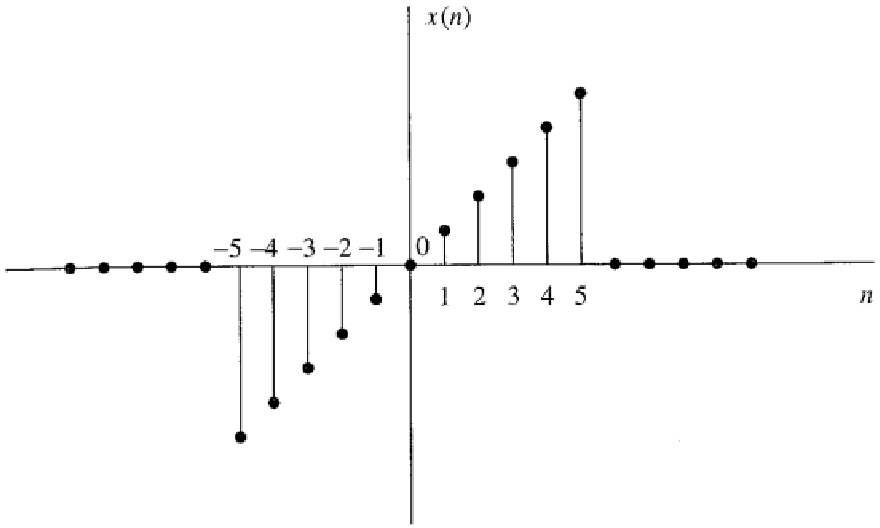
Signal Symmetry

x(n): real signal

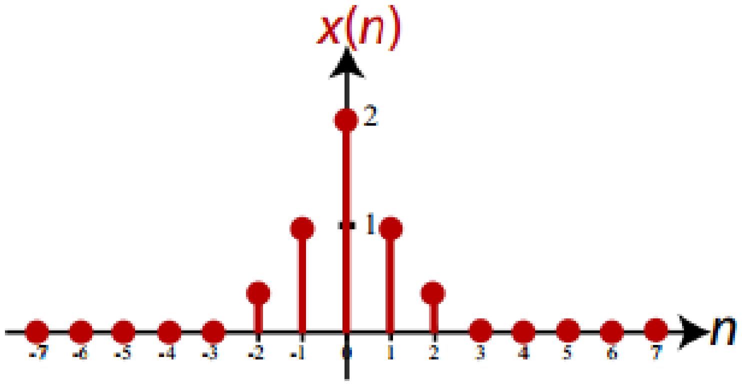
- Symmetric signal (even signal)
  - $x(n) = x(-n), \forall n$

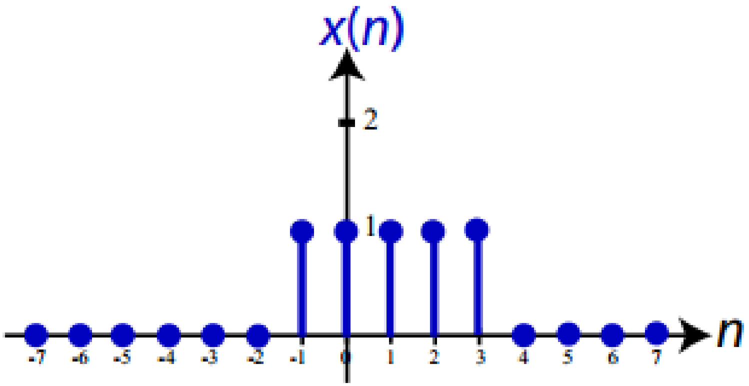


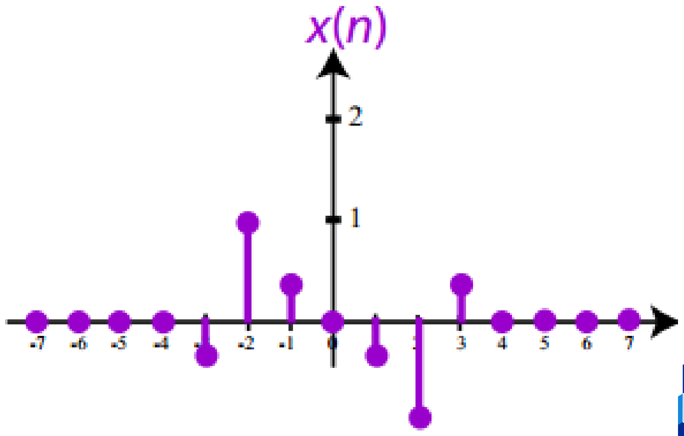
- Antisymmetric signal (odd signal)
  - $x(n) = -x(-n), \forall n$
- Any arbitrary signal can be expressed by the sum of two signal components
  - $x(n) = x_e(n) + x_o(n)$, where
  - Even signal component
    - $x_e(n) = (½)[x(n) + x(-n)]$
  - Odd signal component
    - $x_o(n) = (½)[x(n) - x(-n)]$



Quiz


x(n)
2
1
n
-7
-6
-5
-4
-3
-2
-1
0
1
2
3
4
5
6
7


x(n)
2
1
n
-7
-6
-5
-4
-3
-2
-1
0
1
2
3
4
5
6
7


x(n)
2
1
-7
-6
-5
-4
-2
-1
0
3
4
5
6
7

Simple Manipulations of Discrete-Time Signals

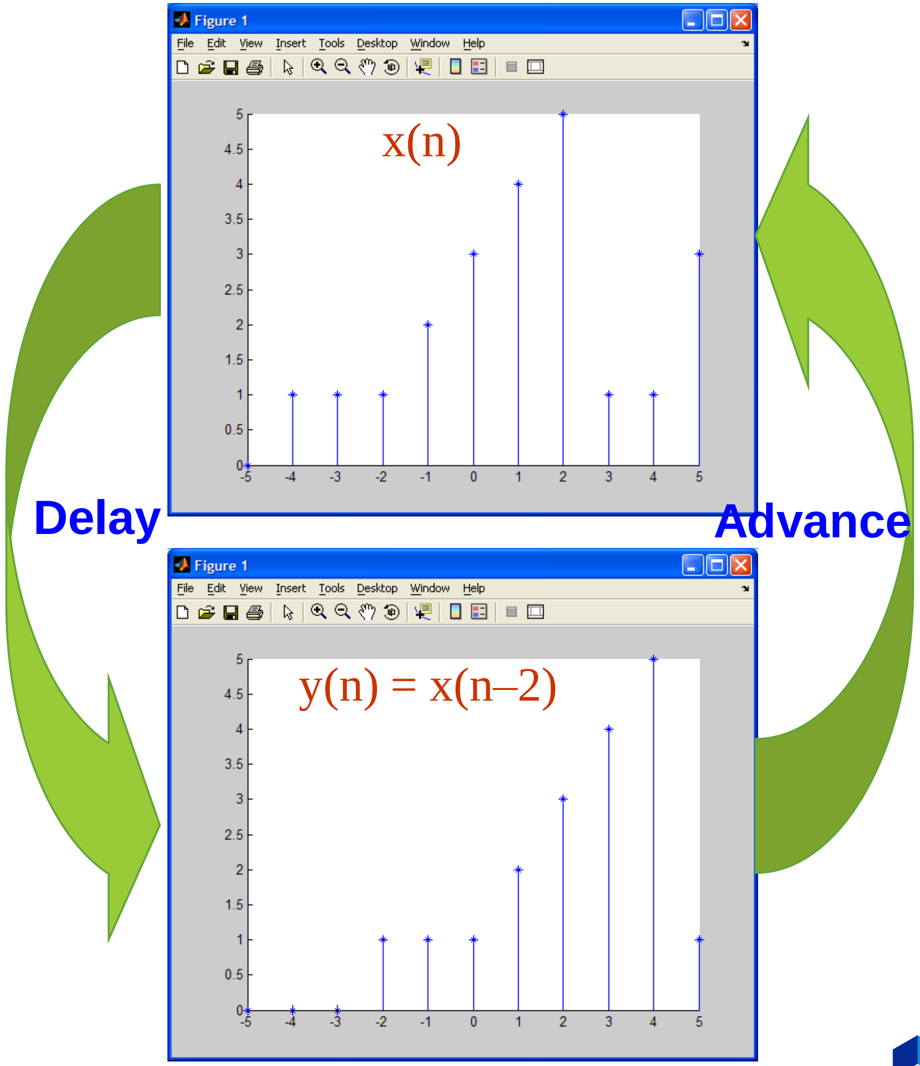
- **Transformation of the independent variable (time)**
  - Delay
  - Advance
  - Folding

- **Addition, Multiplication, and scaling of sequences**
  - Addition
  - Multiplication
  - Amplitude Scaling

Simple Manipulations of Discrete-Time Signals

- **Delay**: shifted in time by replacing n by n–k
  - $y(n) = x(n-k) \qquad \forall k > 0$
  - y(n) is the time shift result in a delay of the signal by k units of time.
  - Graphically, delay corresponds to **shifting the signal to the RIGHT on the time axis**.

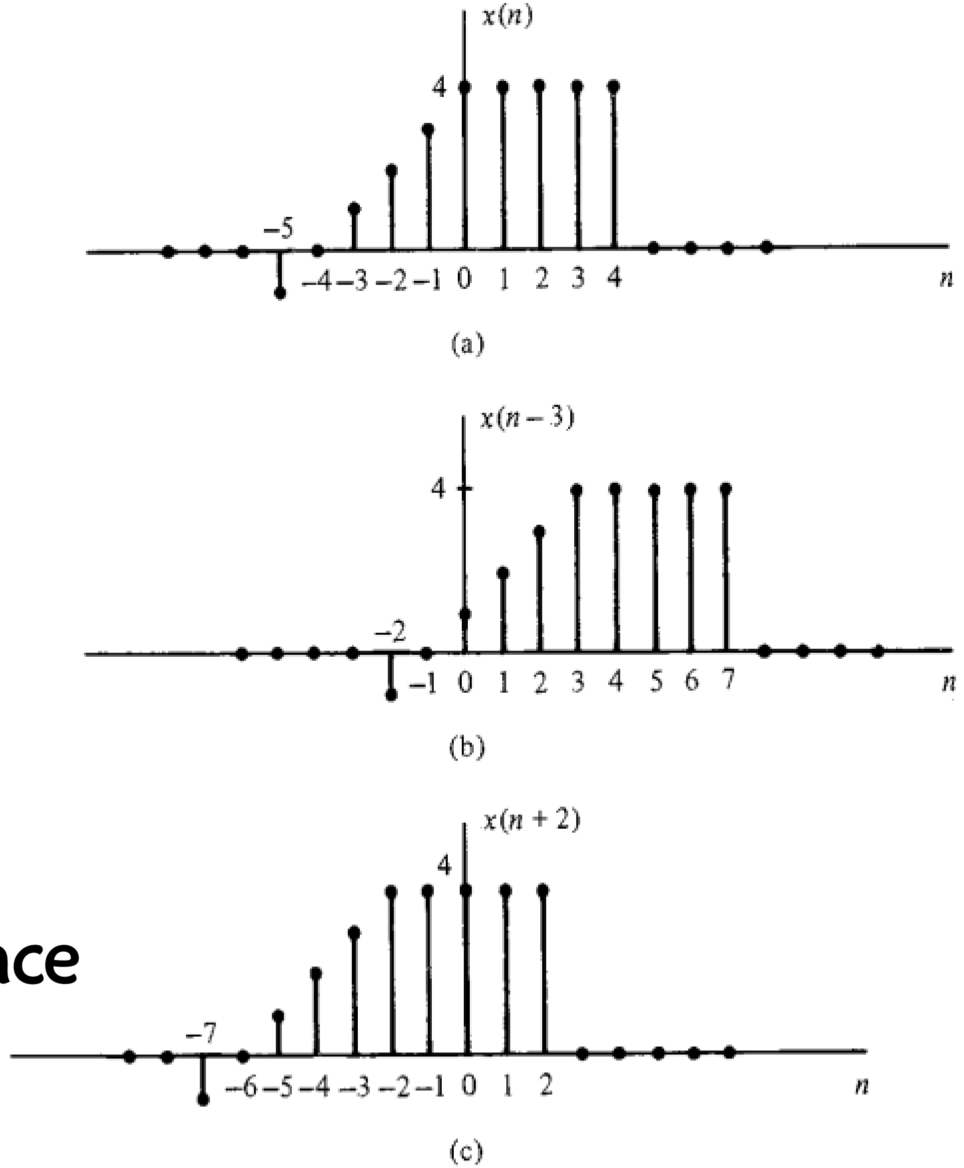
- **Advance**: shifted in time by replacing n by n+k
  - $y(n) = x(n+k) \qquad \forall k > 0$
  - y(n) is the time shift result in an advance of the signal by k units of time.
  - Graphically, advance implies **shifting the signal to the LEFT on the time axis**.



Example

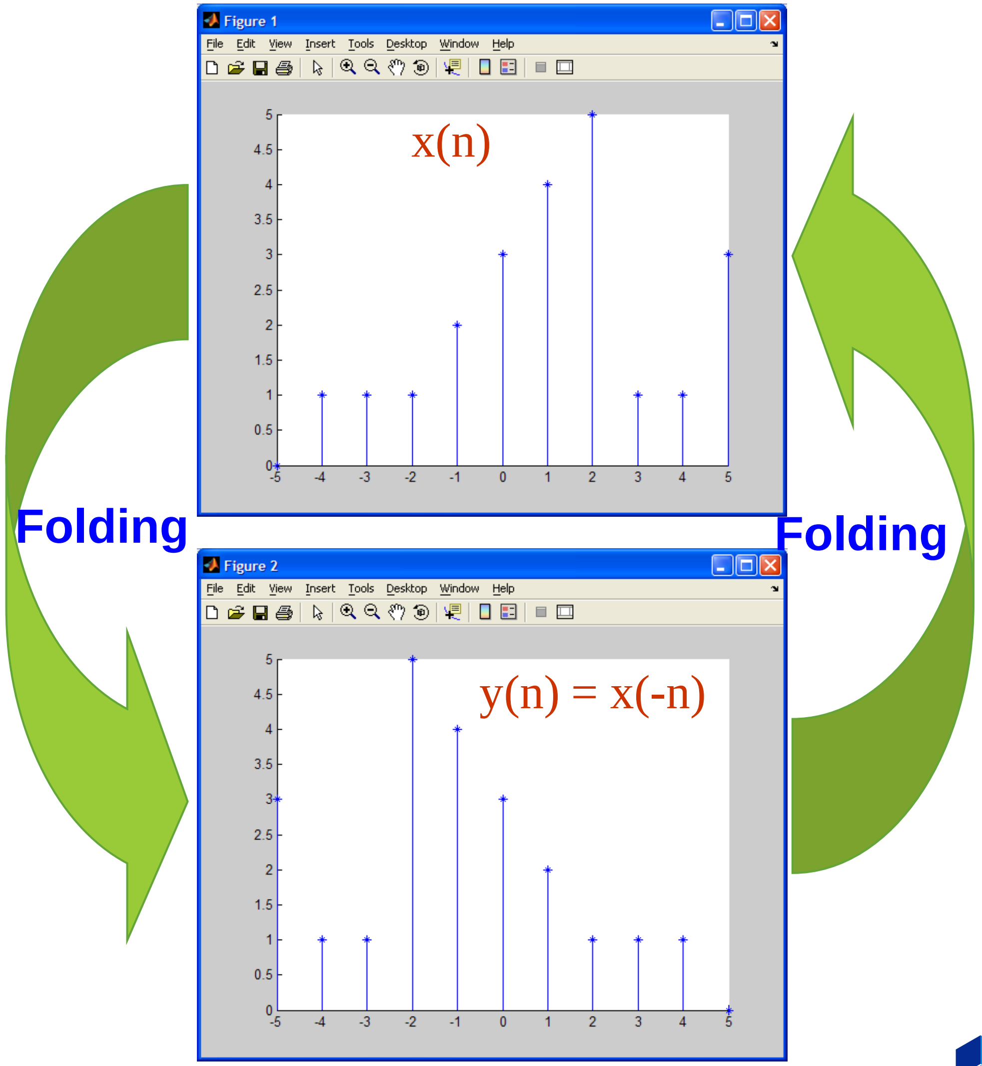
Delay

Advance

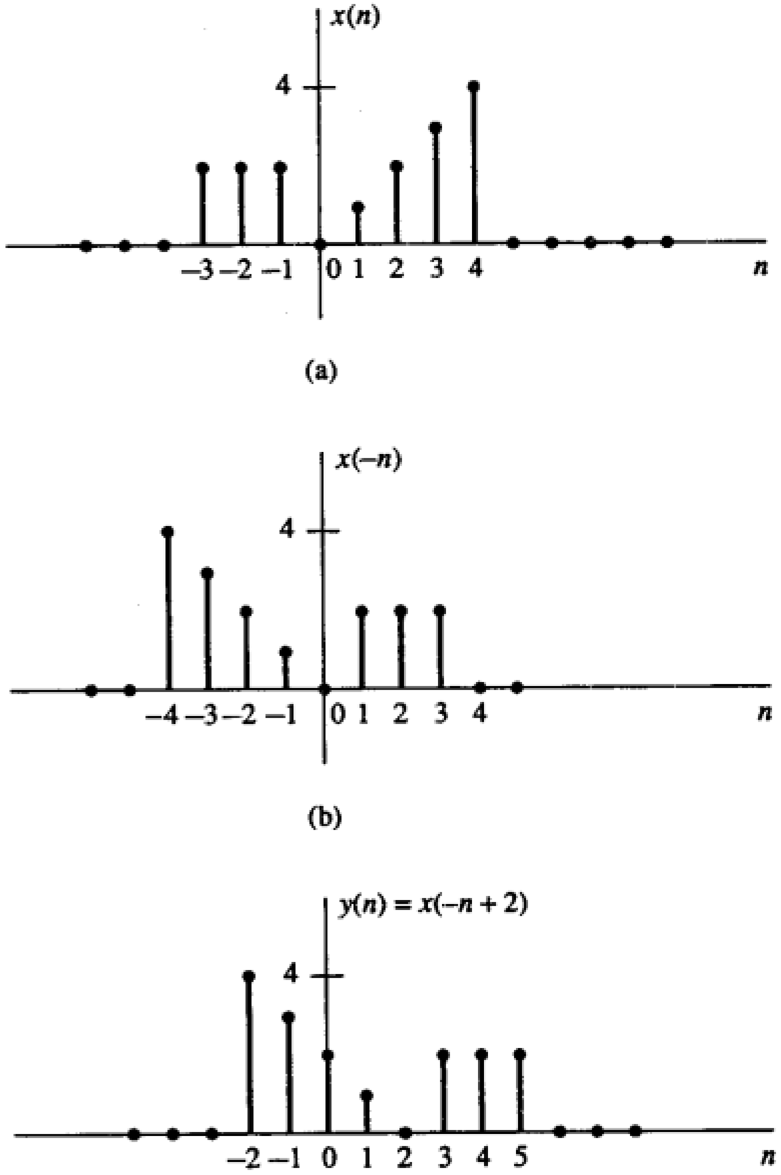


Simple Manipulations of Discrete-Time Signals

- **Folding**: replace n by –n
  - $y(n) = x(-n)$
  - y(n) is a folding or a reflection of the signal about the time origin n=0.

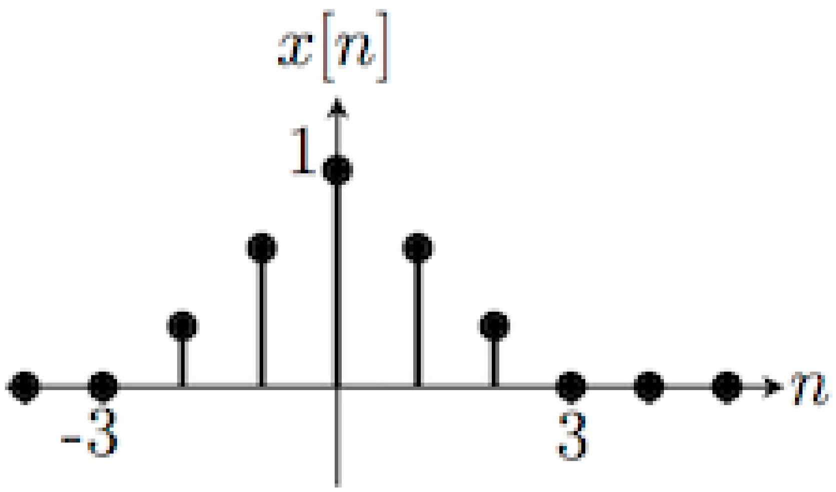


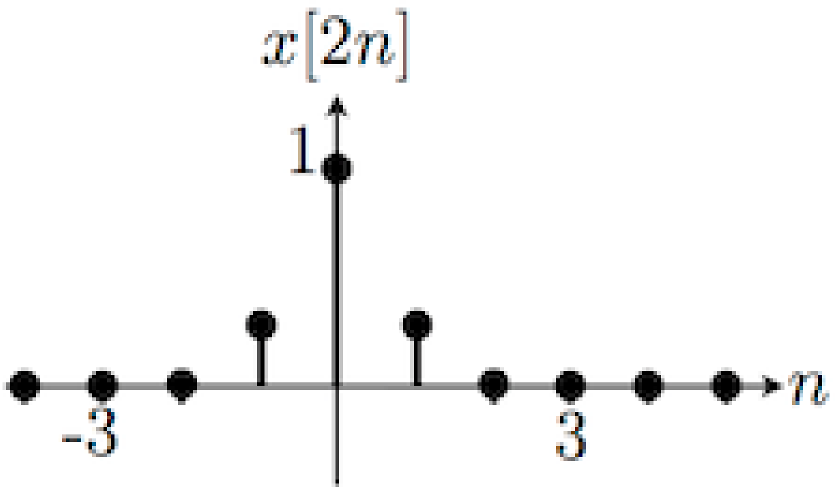
Example



Simple Manipulations of Discrete-Time Signals

- **Time scaling**: replace n by µn (µ € Z)
  - y(n) = x(µn)   where µ € Z
  - y(n) is the time scaling results of the signal x(n) with the coefficient µ





Simple Manipulations of Discrete-Time Signals

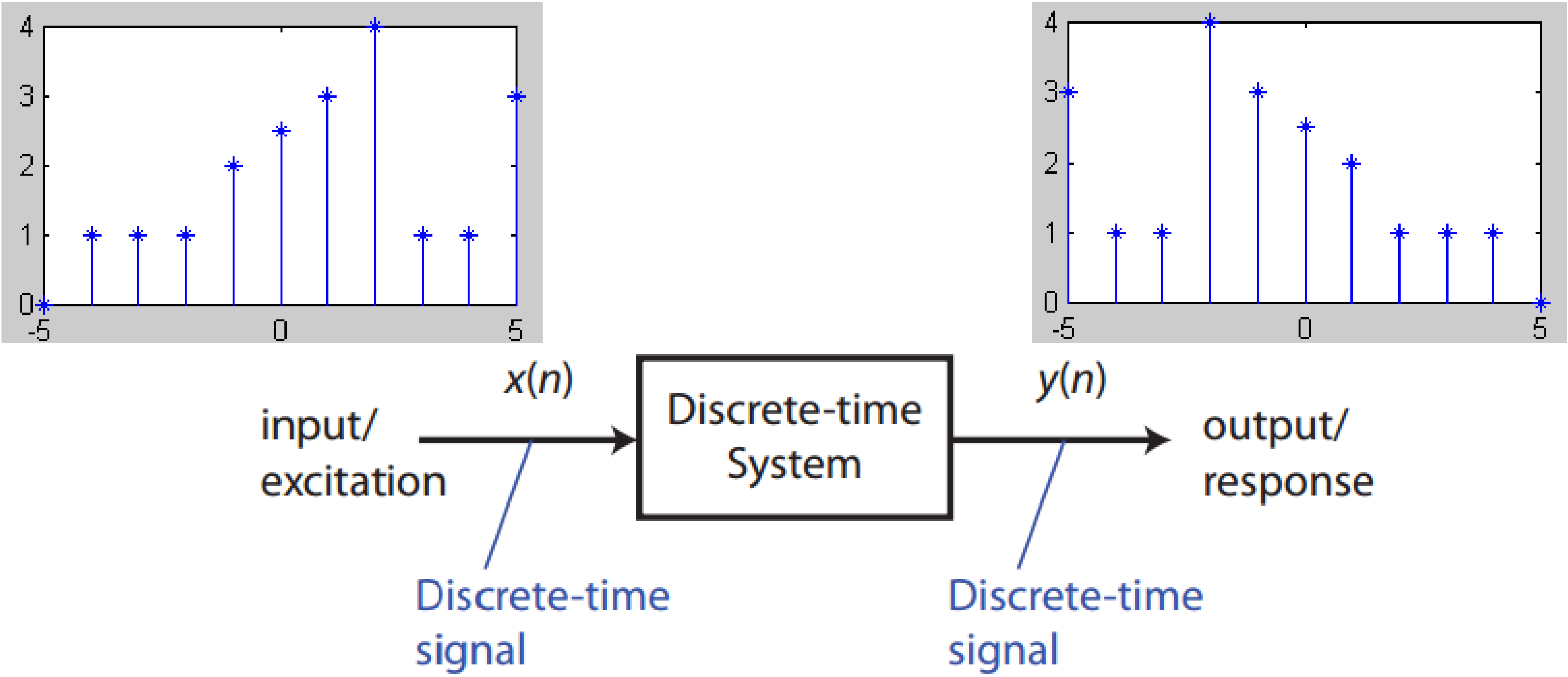
$x_1(n)$ và $x_2(n)$ n: $[-\infty,+\infty]$

- **Addition**
  - $y(n) = x_1(n) + x_2(n)$ n: $[-\infty,+\infty]$
- **Multiplication**
  - $y(n) = x_1(n).x_2(n)$ n: $[-\infty,+\infty]$
- **Amplitude Scaling**
  - $y(n) = ax_1(n)$ n: $[-\infty,+\infty]$

Exercise

- Given two digital signals $x_1 = \{1\ -1\hat{}\ \ 0\ \ 0\ \ 2\ -4\}$ and $x_2 = \{-2\ \ 3\hat{}\ \ 1\ \ 0\ -3\}$, determine
  - $y_1(n) = x_1(n\ -\ 2)$
  - $y_2(n) = x_2(-n+1)$
  - $y_3(n) = y_1(n) + y_2(n)$
  - $y_4(n) = y_1(n).y_2(n)$

Discrete-Time Systems



- Input-Output Description

$$y(n) = T[x(n)]$$

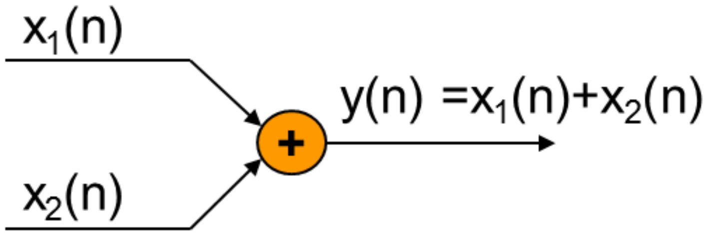
  - Exact structure of system is unknown or ignored.
  - Black-Box representation

$$x(n) \xrightarrow{T} y(n)$$

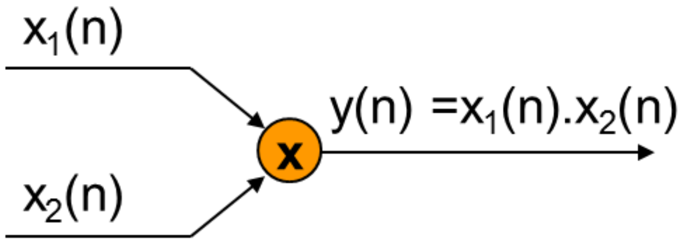
Discrete-Time Systems

- **Block Diagram Representation**
  - Interconnect basic blocks to describe the system.

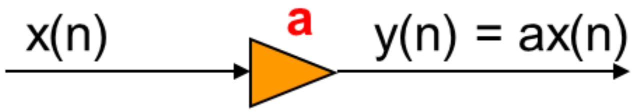
An Adder



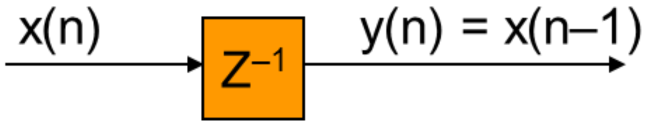
A Signal Multiplier



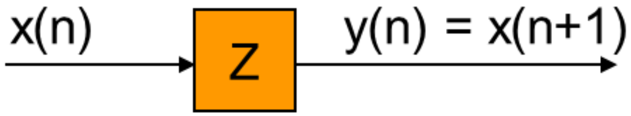
A Constant Multiplier



A Unit Delay Element



A Unit Advance Element

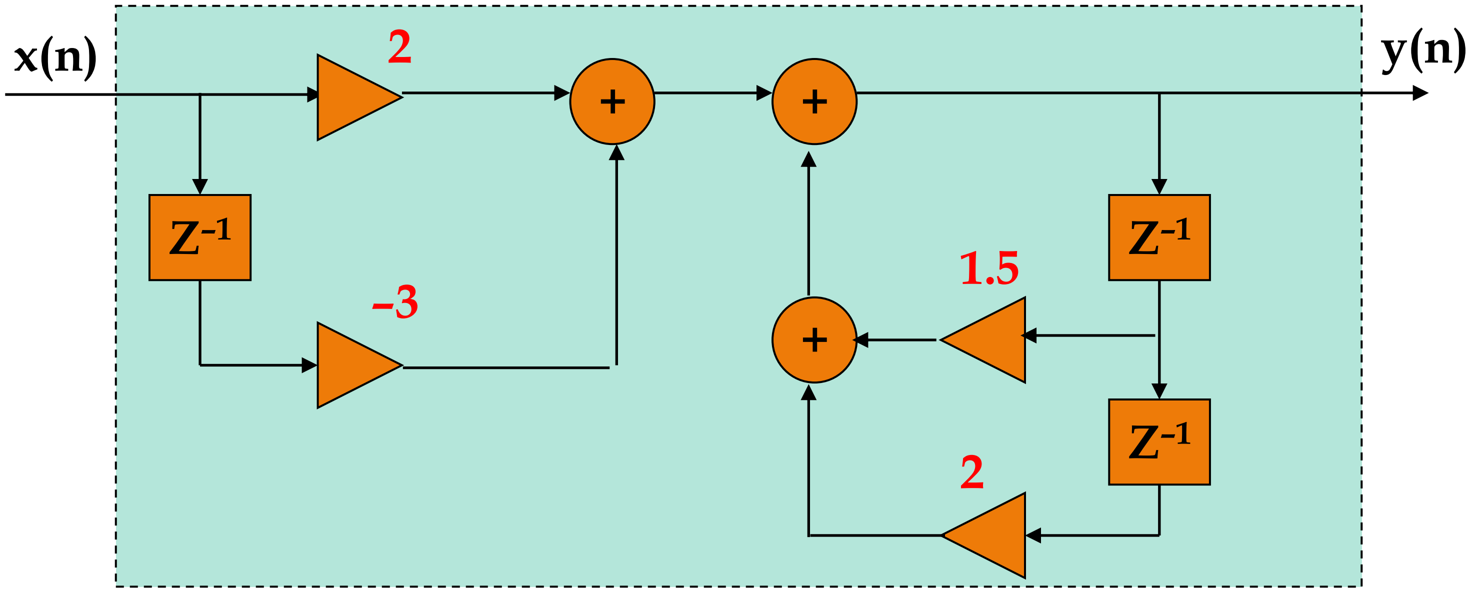


Example

- A system is given by Input-Output Description as follows

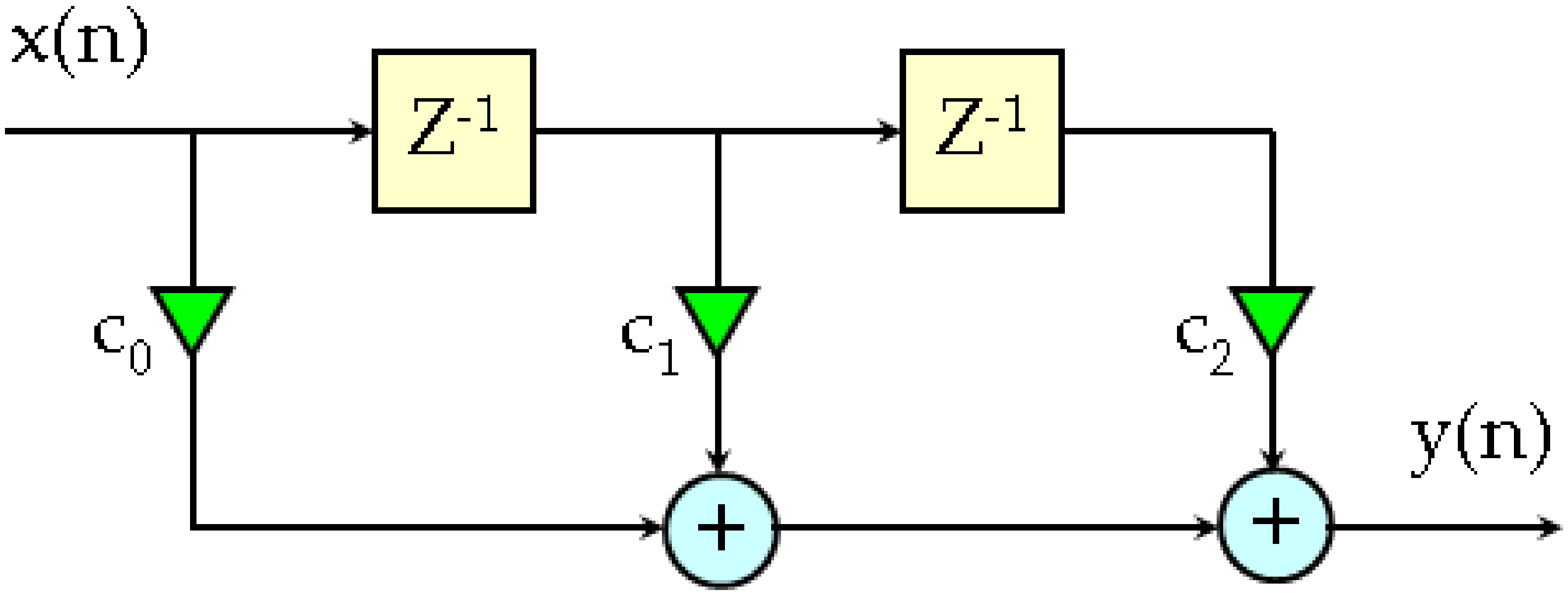
$$y(n) = 2x(n) - 3x(n-1) + 1.5y(n-1) + 2y(n-2)$$

- The corresponding block diagram representation of the above system is



Exercise

- Write the input-output description corresponding to the system represented by the block diagram as the following Figure.



Classification of Discrete-Time Systems

- Why is this so important?
  - mathematical techniques developed to analyze systems are often contingent upon the general characteristics of the systems being considered.

- For a system to possess a given property, the property must hold for **every** possible input to the system.
  - to disprove a property, need a single counter-example.
  - to prove a property, need to prove for the general case.

Classification of Discrete-Time Systems

Common System Properties

- **Static vs. Dynamic**
- **Time-Invariant vs. Time-Variant**
- **Linear vs. Non-linear**
- **Causal vs. Non-causal**
- **Stable vs. Unstable**

Static vs. Dynamic Systems

- **A discrete-time system is called static or memoryless** if its output at any instant n depends only on the input sample at time n (not on the past or future sample of the input); **otherwise the system is said to be dynamic**.
  - **NO** Z^{-1} in block diagram representation
  - **NO** x(n–k) or y(n–k) in input-output description

- **Consider the general system**

$$\begin{aligned} y(n) &= \mathcal{T}[x(n-N), x(n-N+1), \cdots, x(n-1), x(n), x(n+1), \\ &\quad \cdots, x(n+M-1), x(n+M)], \qquad N, M > 0 \end{aligned}$$

  - For N=M=0, y(n)=T[x(n)] $\rightarrow$ the system is **static**.
  - 0< N, M < ∞ $\rightarrow$ the system is said to be **dynamic** with finite memory.
  - N=∞ (M =∞) $\rightarrow$ the system is said to have infinite memory.

Static vs. Dynamic Systems

- Example: static (memoryless) or not?
  - Y ▸ $y(n) = A\,x(n),\ A \neq 0$
  - Y ▸ $y(n) = A\,x(n) + B,\ A, B, \neq 0$
  - Y ▸ $y(n) = x(n)\cos(\frac{\pi}{25}(n - 5))$
  - N ▸ $y(n) = x(-n)$
  - N ▸ $y(n) = x(n+1)$
  - N ▸ $y(n) = \frac{1}{1 - x(n+2)}$
  - Y ▸ $y(n) = e^{3x(n)}$
  - N ▸ $y(n) = \sum_{k=-\infty}^{n} x(k)$

Time-Invariant vs. Time-Variant Systems

- **Time-Invariant System**
  - input-output characteristics do not change with time
  - **Definition**
    - A relaxed system T is time-invariant or (shift invariant) if and only if

$$x(n) \xrightarrow{T} y(n) \Rightarrow x(n-k) \xrightarrow{T} y(n-k) \qquad \forall x(n), \forall k$$

    - In general, we can write the output as

$$y(n,k) = T[x(n-k)]$$

- **Time-Variant System**
  - The system does not satisfy the above definition.

Time-Invariant vs. Time-Variant Systems

- Example 1
  - The system is described by the input-output equation

$$y(n) = T[x(n)] = x(n) - x(n-1)$$

  - If the input is delayed by k units in time and applied to the system, then the output will be

$$y(n, k) = x(n-k) - x(n-k-1)$$

  - On the other hand, if we delay y(n) by k units in time, we obtain

$$y(n-k) = x(n-k) - x(n-k-1)$$

  - Obviously, y(n, k) and y(n – k) are identical. Therefore, the system is time-invariant.

Time-Invariant vs. Time-Variant Systems

- Example 2
  - The system is described by the input-output equation

$$y(n) = T[x(n)] = nx(n)$$

  - If the input is delayed by k units in time and applied to the system, then the output will be

$$y(n, k) = nx(n - k)$$

  - On the other hand, if we delay y(n) by k units in time, we obtain

$$y(n - k) = (n - k)x(n - k)$$

  - Obviously, y(n, k) and y(n – k) are **different** ($y(n,k) \neq y(n-k)$). Therefore, the system is **time-variant**.

Time-Invariant vs. Time-Variant Systems

- **Quiz:** time-invariant or not?

- Y — $y(n) = A\,x(n),\ A \neq 0$
- Y — $y(n) = A\,x(n) + B,\ A, B, \neq 0$
- N — $y(n) = x(n)\cos(\frac{\pi}{25} n)$
- N — $y(n) = x(-n)$
- Y — $y(n) = x(n+1)$
- Y — $y(n) = \frac{1}{1-x(n+2)}$
- Y — $y(n) = e^{3x(n)}$
- Y — $y(n) = \sum_{k=-\infty}^{n} x(k)$

Linear vs. Non-Linear Systems

- **Linear System**
  - Obey superposition principle
  - **Definition**
    - A system is linear if and only if:

$$T[a_1x_1(n) + a_2x_2(n)] = a_1T[x_1(n)] + a_2T[x_2(n)] \quad \forall a_i, \forall x_i(n)$$

  - **Homogeneity**
    - Let $a_2 = 0 \rightarrow T[a_1x_1(n)] = a_1T[x_1(n)]$
  - **Additivity**
    - Let $a_1 = a_2 = 1 \rightarrow T[x_1(n) + x_2(n)] = T[x_1(n)] + T[x_2(n)]$
- **Non-Linear System**
  - The system does not obey superposition principle

Linear vs. Non-Linear Systems

- **Note:**

Linearity = **Homogeneity + Additivity**

  - If a system is **not homogeneous**, it is **not linear**.
  - If a system is **not additive**, it is **not linear**.

Linear vs. Non-Linear Systems

- **Example 1**
  - The system is described by the input-output equation

$$y(n) = T[x(n)] = nx(n)$$

  - For two input sequences $x_1(n)$ and $x_2(n)$, the corresponding output are:

$$y_1(n) = nx_1(n) \qquad (I)$$
$$y_2(n) = nx_2(n) \qquad (II)$$

  - A linear combination of the two input sequences in the output

$$y_3(n) = T[a_1x_1(n)+a_2x_2(n)] = n[a_1x_1(n)+a_2x_2(n)] = na_1x_1(n) + na_2x_2(n)$$

  - On the other hand, a linear combination of the two outputs (I)&(II) results in the output.

$$a_1y_1(n)+a_2y_2(n)=a_1nx_1(n) + a_2nx_2(n)$$

  - Obviously, the system **obeys** superposition principle. Therefore, the system is **Linear**.

Linear vs. Non-Linear Systems

- **Example 2**
  - The system is described by the input-output equation

$$y(n) = T[x(n)] = x^2(n)$$

  - For two input sequences $x_1(n)$ and $x_2(n)$, the corresponding output are:

$$y_1(n) = x_1^2(n) \qquad (I)$$
$$y_2(n) = x_2^2(n) \qquad (II)$$

  - A linear combination of the two input sequences in the output

$$y_3(n) = T[a_1x_1(n)+a_2x_2(n)] = [a_1x_1(n)+a_2x_2(n)]^2 \qquad (III)$$

  - On the other hand, a linear combination of the two outputs (I)&(II) results in the output.

$$a_1y_1(n)+a_2y_2(n)=a_1x_1^2(n) + a_2x_2^2(n) \qquad (IV)$$

  - From (III) & (IV), the system **does not** obey superposition principle. Therefore, the system is **Non-Linear**.

Linear vs. Non-Linear Systems

- **Quiz:** Linear or not?

Y	$y(n) = A\,x(n),\ A \neq 0$
N	$y(n) = A\,x(n) + B,\ A, B, \neq 0$
Y	$y(n) = x(n)\cos(\frac{\pi}{25}n)$
Y	$y(n) = x(-n)$
Y	$y(n) = x(n+1)$
N	$y(n) = \frac{1}{1-x(n+2)}$
N	$y(n) = e^{3x(n)}$
Y	$y(n) = \sum_{k=-\infty}^{n} x(k)$

Causal vs. Noncausal Systems

- **Causal System**
  - **Definition**
    - A system T is said to be causal if the output of the system at any time n [i.e. y(n)] **depends only on present and past inputs** [i.e. x(n), x(n – 1), x(n – 2) …]. In mathematical term, the output of a causal system satisfies an equation of the form

$$y(n) = F[x(n), x(n-1), x(n-2), \ldots]$$

- **Noncausal System**
  - The system is said to be Noncausal if the output of the system does not abbey the above definition.

Causal vs. Noncausal Systems

- **Quiz:** Causal or not?

- Y — $y(n) = A\,x(n),\ A \neq 0$
- Y — $y(n) = A\,x(n) + B,\ A, B, \neq 0$
- Y — $y(n) = x(n)\cos(\frac{\pi}{25}(n+1))$
- N — $y(n) = x(-n)$
- N — $y(n) = x(n+1)$
- N — $y(n) = \frac{1}{1-x(n+2)}$
- Y — $y(n) = e^{3x(n)}$
- Y — $y(n) = \sum_{k=-\infty}^{n} x(k)$

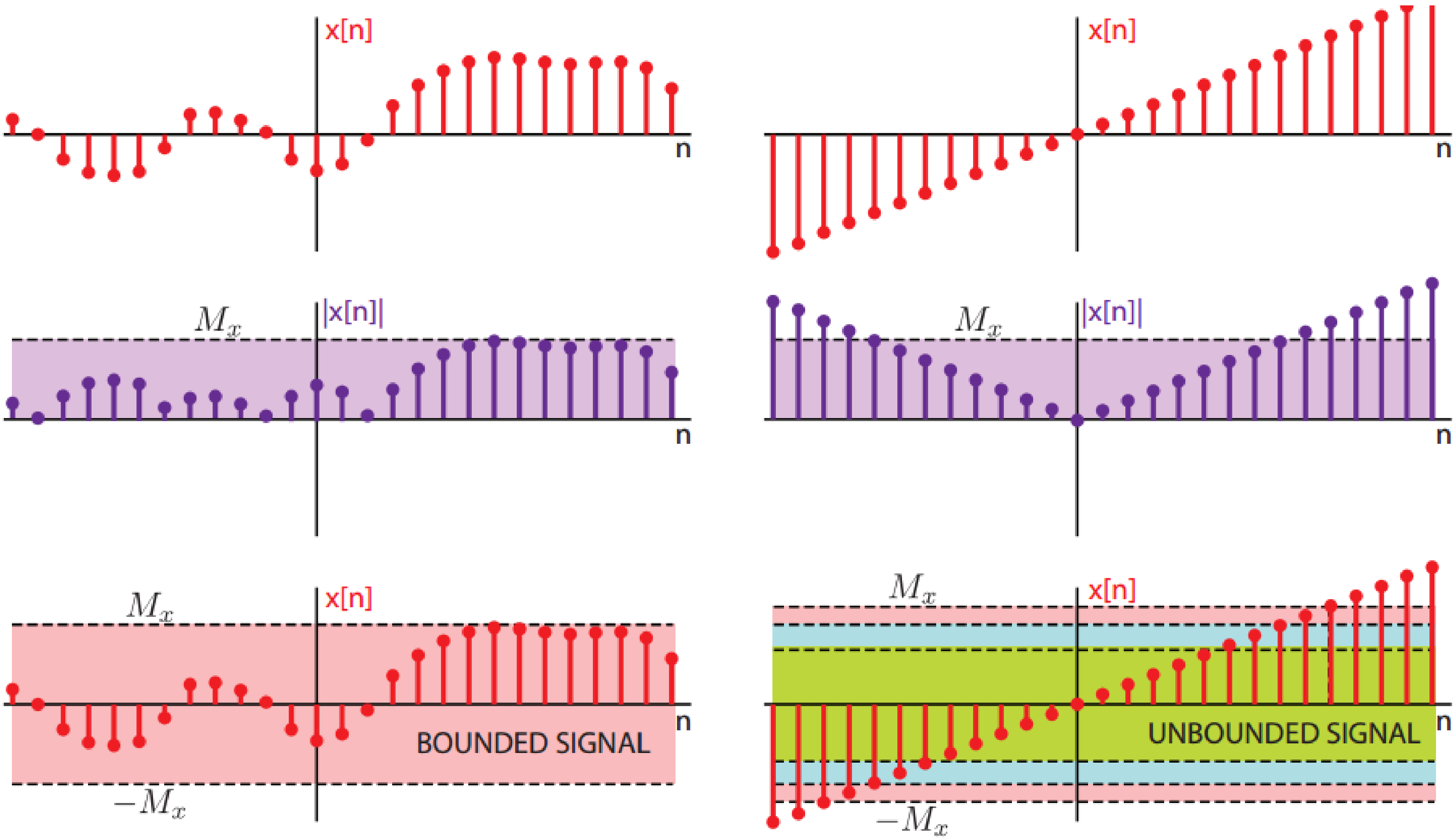
Stable vs. Unstable Systems

- **Stable System**
  - **BIBO:** Bounded Input-Bounded Output
  - **Definition**
    - A relaxed system is said to be BIBO Stable if and only if every bounded input produces a bounded output.

$$\forall x(n): |x(n)| \le M_x < \infty \qquad \rightarrow |y(n)| = |T[x(n)]| \le M_y < \infty$$

- **Unstable System**
  - A system is said to be unstable if it does not satisfy the above definition.

Discrete-Time Bounded Signals



Stable vs. Unstable Systems

- **Quiz:** Stable or not?
  - Y — $y(n) = A\,x(n),\ A \neq 0$
  - Y — $y(n) = A\,x(n) + B,\ A, B, \neq 0$
  - Y — $y(n) = x(n)\cos(\frac{\pi}{25}n)$
  - Y — $y(n) = x(-n)$
  - Y — $y(n) = x(n+1)$
  - N — $y(n) = \frac{1}{1-x(n+2)}$
  - Y — $y(n) = e^{3x(n)}$
  - N — $y(n) = \sum_{k=-\infty}^{n} x(k)$

Final Remarks

- For a system to possess a given property, the property must hold for **every** possible **input** and **parameter** of the system.
  - To disprove a property, need a **single counter-example.**
  - To prove a property, need to **prove for the general case.**

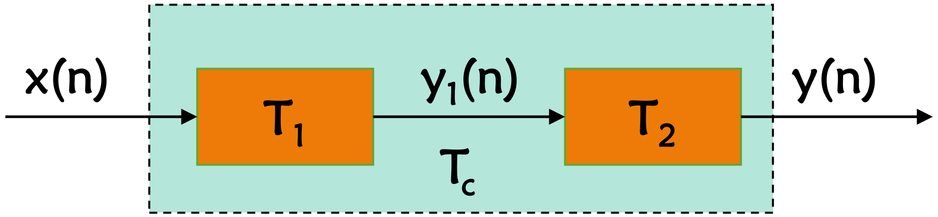
In-Class Problems

- Investigate all the properties of the following systems
  - $y_1(n) = x(n) + nx(n + 1)$
  - $y_2(n) = x(2n)$

Interconnection of Discrete-Time Systems

- Discrete-time systems can be interconnected to form larger systems.

- **2 basic interconnections**
  - **Cascade**

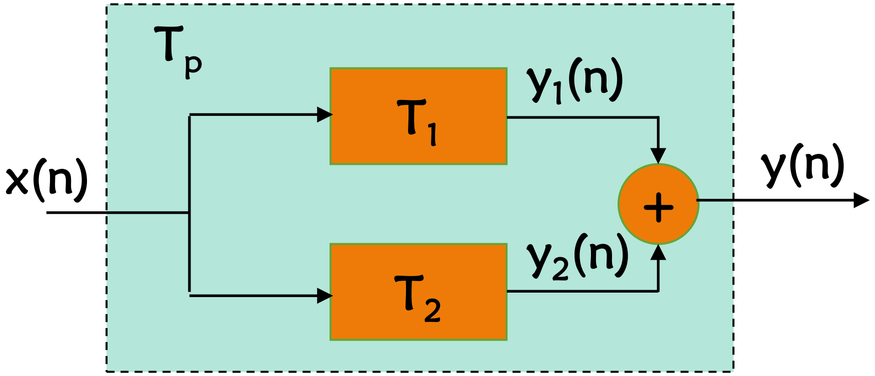


$$\left.\begin{array}{l} y_1(n) = T_1[x(n)] \\ y(n) = T_2[y_1(n)] \end{array}\right\} \qquad y(n) \begin{array}{l} = T_2[T_1[x(n)]] \\ = T_c[x(n)] \end{array} \quad \text{với } T_c \equiv T_2T_1$$

    - $T_2T_1 \neq T_1T_2$
    - **If both T_1 and T_2 are linear and time-invariant (LTI).**
      - $\mathbf{T_c = T_2T_1}$: time-invariant system
      - $\mathbf{T_2T_1 = T_1T_2}$

  - **Parallel**

$$\begin{aligned} y(n) &= T_1[x(n)] + T_2[x(n)] \\ &= (T_1+T_2)[x(n)] \\ &= T_p[x(n)] \qquad \text{where } T_p \equiv T_1+T_2 \end{aligned}$$



Analysis of Discrete-Time LTI Systems

- **Techniques for the Analysis of Linear System**

  1. Directly solve the input-output equation of the system.

  2. **Decompose or resolve the input signal into a sum of elementary signals** that are selected so that the response of the system to each signal component is predetermined.

  - Then, using the **linearity**, the response of the system to the given input signals are the summation of the responses of the system to each elementary signals.

- **Example**
  - Decompose the input signal
    - where $y_k(n) = T[x_k(n)]$

$$x(n) = \sum_k c_k x_k(n)$$

$$\begin{aligned} y(n) &= T[x(n)] \\ &= T[\sum_k c_k x_k(n)] \\ &= \sum_k c_k T[x_k(n)] \end{aligned}$$

$$\Rightarrow y(n) = \sum_k c_k y_k(n)$$

Resolution of A Discrete-Time Signal Into Impulses

- Resolution of A Discrete-Time Signal Into Impulses
  - Select the elementary signals
    - $x_k(n) = \delta(n-k)$
  - And
    - $x(n)\delta(n-k) = x(k)\delta(n-k) \quad \forall k$
  - Sum all the product sequences, the result will be a sequence equal to sequence x(n)

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k)$$

  - Example

$$x(n) = \{2 \quad 4 \quad \underset{\uparrow}{3} \quad 1\}$$

$$x(n) = 2\delta(n+2) + 4\delta(n+1) + 3\delta(n) + \delta(n-1)$$

Response of LTI Systems

- The response **y(n,k)** of the system to the input unit sample sequence at n=k is denoted **h(n,k)**
  - $y(n, k) \equiv h(n, k) = T[\delta(n-k)]$ $\quad -\infty < k < \infty$
    - n: time index
    - k: position of corresponding impulse

- If the impulse at the input is scaled by an amount $c_k = x(k)$, the response of the system is also correspondingly scaled by $c_k h(n, k) = x(k)h(n, k)$

The Convolution Sum

The Convolution Sum

$$\begin{aligned} y(n) &= T[x(n)] \\ &= T[\sum_{k=-\infty}^{\infty} x(k)\delta(n-k)] \\ &= \sum_{k=-\infty}^{\infty} x(k)T[\delta(n-k)] \\ &= \sum_{k=-\infty}^{\infty} x(k)h(n,k) \end{aligned}$$

- For LTI system, if h(n) = T[δ(n)] then h(n–k) = T[δ(n–k)]

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k)$$

The Convolution Sum

- Procedure to determine the response of the system at time instant n_0.

$$y(n_0) = \sum_{k=-\infty}^{\infty} x(k)h(n_0 - k)$$

1. **Folding**: $h(k) \rightarrow h(-k)$
2. **Shifting**: $h(-k) \rightarrow h(-k + n_0)$: shifting $h(-k)$ n_0 units to the **RIGHT** or **LEFT** if n_0 is **positive** or **negative** respectively.
3. **Multiplication**: $v_{n0}(k) = x(k)\, h(-k + n_0)$
4. **Summation**: sum all the sequences $v_{n0}(k)$

The Convolution Sum

- **Example**
  - The impulse response of a LTI system is

$$h(n) = \{1, \underset{\uparrow}{2}, 1, -1\}$$

  - Determine the response of the system to the input signal

$$x(n) = \{\underset{\uparrow}{1}, 2, 3, 1\}$$

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k)$$

The Convolution Sum

- In the convolution equation, if replacing m=n–k (i.e. k=n–m), we obtain

$$y(n) = \sum_{m=-\infty}^{\infty} x(n-m)h(m) = \sum_{k=-\infty}^{\infty} x(n-k)h(k)$$

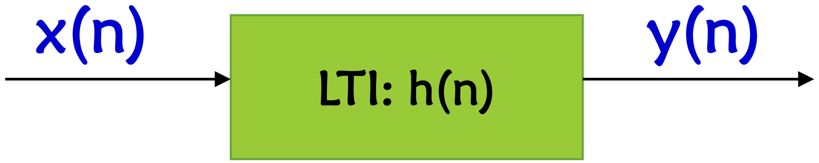
  - Although, the above output y(n) and the result of convolution sum are identical. They are in different arrangement.
  - If

$$\left.\begin{aligned} v_n(k) &= x(k)h(n-k) \\ w_n(k) &= x(n-k)h(k) \end{aligned}\right\} \quad v_n(k) = w_n(n-k)$$

$$\Rightarrow y(n) = \sum_{k=-\infty}^{\infty} v_n(k) = \sum_{k=-\infty}^{\infty} w_n(n-k)$$

The Convolution Sum

- **Summary**



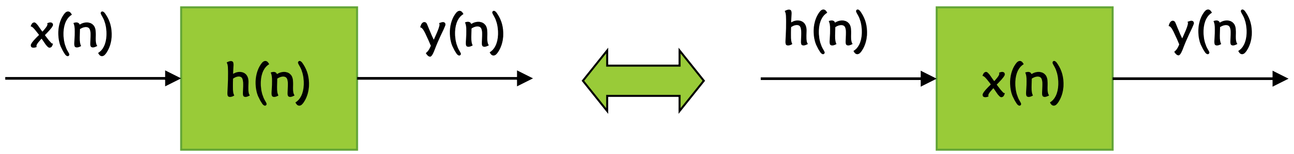
h(n) : The impulse response of the LTI system

$$\begin{aligned} y(n) &= x(n) * h(n) \\ &= \sum_{k=-\infty}^{\infty} x(k)h(n-k) \end{aligned}$$

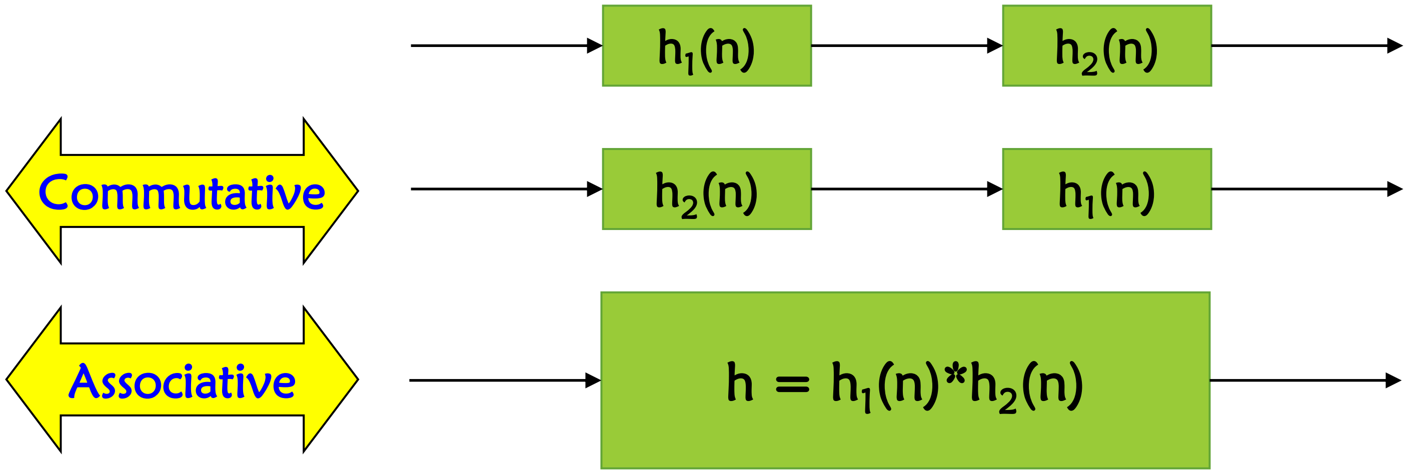
$$\begin{aligned} y(n) &= h(n) * x(n) \\ &= \sum_{k=-\infty}^{\infty} x(n-k)h(k) \end{aligned}$$

Properties of Convolution

- **Commutative** $x(n)*h(n) = h(n)*x(n)$



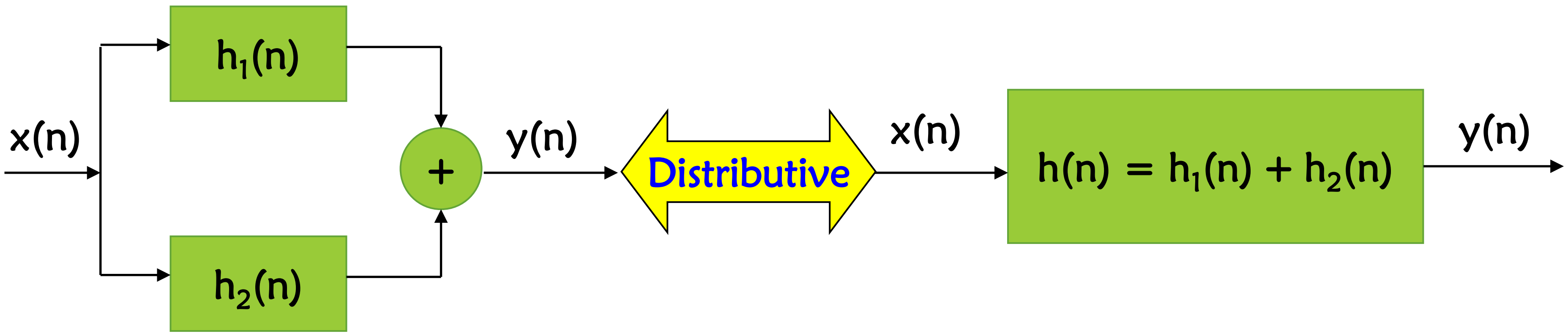
- **Associative** $[x(n)*h_1(n)]*h_2(n) = x(n)*[h_1(n)*h_2(n)]$



Properties of Convolution

- **Distributive**

$$x(n)*[h_1(n) + h_2(n)] = x(n)*h_1(n) + x(n)*h_2(n)$$



- **Example**: Determine the response of the following systems using convolution.
  - $x(n) = a^n u(n)$ and $h(n) = b^n u(n)$ for two cases **a=b** and **a≠b**
  - $x(n) = \{\ldots 0, 1^{*}, 2, 1, 1, 0\ldots\}$ and $h(n)=\delta(n)-\delta(n-1)+\delta(n-4)+\delta(n-5)$

Finite vs. Infinite Impulse Response

- **FIR** (Finite-duration Impulse Response)
  - $h(n) = 0 \quad \forall n: n < 0 \text{ and } n \geq M$

$$y(n) = \sum_{k=0}^{M-1} h(k)x(n-k)$$

  - An FIR system has a finite memory of length-M samples.

- **IIR** (Infinite-duration Impulse Response)
  - For a causal system

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k)$$

  - An IIR system has an infinite memory.

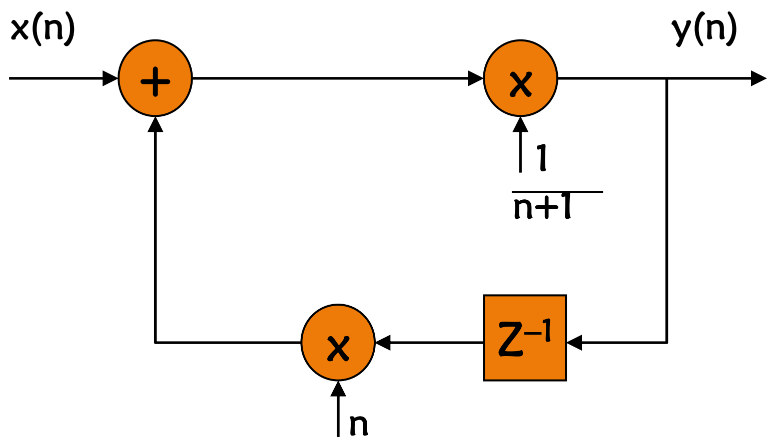
Recursive Discrete-Time Systems

- The cumulative average of a signal x(n) in the interval 0≤k≤n.

$$y(n) = \frac{1}{n+1}\sum_{k=0}^{n} x(k)$$

  - The computation y(n) requires the storage of all the input samples x(k) for 0≤k≤n ⇒ **since n is increasing, our memory requirements grow linearly with time**.

  - y(n) can be computed by using recursive method



$$(n+1)y(n) = \sum_{k=0}^{n-1} x(k) + x(n) = ny(n-1) + x(n)$$

$$\Rightarrow y(n) = \frac{n}{n+1} y(n-1) + \frac{1}{n+1} x(n)$$

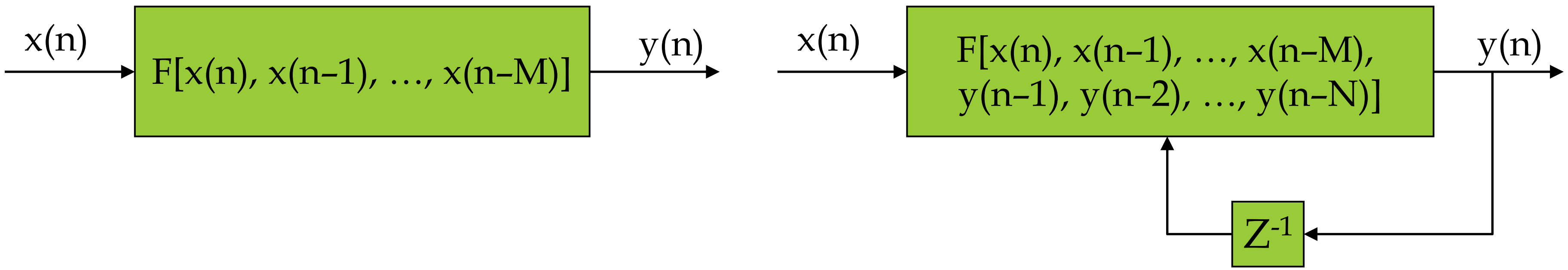
  - A system whose output y(n) at time n depends on any number of past output values y(n – 1), y(n – 2), …is called a **recursive system**.

Nonrecursive Discrete-Time Systems

- The system is **nonrecursive** if

$$y(n) = F[x(n), x(n-1), \ldots, x(n-M)]$$

- Recursive vs. Nonrecursive Systems



- **Notes**
  - If the system is recursive, to compute y(n), we first need to compute all previous (past) values y(0), y(1), … y(n – 1).
  - If the system is nonrecursive, we can compute the output y(n) immediately without having past values y(n – 1), y(n – 2), …
  - Recursive System = Sequential System
  - Nonrecursive system = Combinational System.

LTI Systems Characterized by Constant-Coefficient Difference Equations

- Restate the properties of linearity, time-invariance, and stability of the system described by constant-coefficient difference equations.

- For linear property
  - A system is linear if it satisfies the three following requirements

    1. The total response is equal to the sum of the zero-input and zero-state responses **[i.e., $y(n) = y_{zi}(n) + y_{zs}(n)$]**.

    2. The principle of superposition applies to the zero-state response (zero-state linear).

    3. The principle of superposition applies to the zero-input response (zero-input linear).

  - If the system does not satisfy one among three above conditions is non-linear.

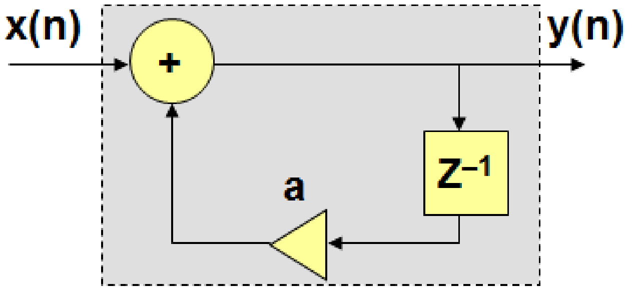
LTI Systems Characterized by Constant-Coefficient Difference Equations

- **Example:** determine if the recursive system defined by the difference equation.

  y(n) = ay(n – 1) + x(n)

- Condition 1

$$\left.\begin{aligned} y_{zs}(n) &= \sum_{k=0}^{n} a^k x(n-k) \qquad \forall n \ge 0 \\ y_{zi}(n) &= a^{n+1} y(-1) \qquad \forall n \ge 0 \end{aligned}\right\} \Rightarrow \mathbf{y(n) = y_{zs}(n) + y_{zi}(n)}$$



- Condition 2
  - Assume that $x(n) = c_1x_1(n) + c_2x_2(n)$

$$y_{zs}(n) = \sum_{k=0}^{n} a^k x(n-k) = \sum_{k=0}^{n} a^k [c_1x_1(n-k) + c_2x_2(n-k)]$$

$$= c_1 \sum_{k=0}^{n} a^k x_1(n-k) + c_2 \sum_{k=0}^{n} a^k x_2(n-k) = \mathbf{c_1 y_{zs}^{(1)} + c_2 y_{zs}^{(2)}}$$

- Condition 3
  - Assume $y(-1)=c_1y_1(-1) + c_1y_2(-1)$

$$y_{zi}(n) = a^{n+1} y(-1) = a^{n+1}[c_1y_1(-1) + c_2y_2(-1)]$$

$$= c_1a^{n+1}y_1(-1) + c_2a^{n+1}y_2(-1) = \mathbf{c_1 y_{zi}^{(1)}(-1) + c_2 y_{zi}^{(1)}(n)}$$

- Hence, **y(n) is linear**.

LTI Systems Characterized by Constant-Coefficient Difference Equations

- For Time-Invariant Property
  - a_k and b_k are constant $\rightarrow$ Time-Invariance
  - A recursive system characterized by Constant-Coefficient Difference Equations is Linear Time-Invariant.

- For Stable Property
  - The BIBO system is stable if and only if its output sequence y(n) is bounded for every bounded intput x(n).
  - Example: determine the range of values of the parameter a for which the given system $y(n) = ay(n – 1) + x(n)$ is stable.

Example

- Assume $|x(n)| \leq \mathrm{M_x} < \infty \quad \forall \mathrm{n} \ \geq \ 0$

$$y(n) = a^{n+1}y(-1) + \sum_{k=0}^{n} a^k x(n-k) \qquad \leq \left|a^{n+1}y(-1)\right| + \left|\sum_{k=0}^{n} a^k x(n-k)\right|$$

$$\leq |a|^{n+1}|y(-1)| + M_x \sum |a|^k$$

$$\leq |a|^{n+1}|y(-1)| + M_x \frac{1-|a|^{n+1}}{1-|a|} \equiv M_y$$

- If n is finite → $\mathrm{M_y}$ is finite
- When $\mathrm{n} \rightarrow \infty$, $\mathrm{M_y}$ is finite only if $|\mathrm{a}| < 1 \Rightarrow \mathrm{M_y} = \frac{\mathrm{M_x}}{1-|\mathrm{a}|}$
- Therefore, the system is stable when $|\mathrm{a}| < 1$

Solve Linear Constant-Coefficient Difference Equations

- The goal is to determine the output y(n) of the system given a specific input x(n) ($n \geq 0$), and a set of initial conditions.

- 2 methods
  - Indirect method: Z – Transform
  - Direct method

- **Direct method** (Solve the linear constant-coefficient difference equation)
  - Total solution: $y(n) = y_h(n) + y_p(n)$
    - $y_h(n)$ is known as homogeneous or complementary solution ($x(n) = 0$)
    - $y_p(n)$ is known as particular solution (depending on x(n))

The Homogeneous Solution of A Difference Equation

- Homogeneous Solution
  - Assume x(n)=0
    - Homogeneous Difference Equation: $$\sum_{k=0}^{N} a_k y(n-k) = 0$$
  - Assume the solution is in form of $y_h(n) = \lambda^n$, then we obtain the polynomial equation

$$\sum_{k=0}^{N} a_k \lambda^{(n-k)} = 0 \Leftrightarrow \lambda^{n-N}\left(\lambda^N + a_1\lambda^{N-1} + a_2\lambda^{N-2} + \cdots + a_{N-1}\lambda + a_N\right) = 0$$

  - The polynomial in parentheses is called the **characteristic polynomial** of the system.
  - In general, it has N roots, which we denote as $\lambda_1, \lambda_2, \ldots, \lambda_N$.
  - Let us assume that the roots are distinct. Then the most general solution to the homogeneous difference equation is

$$y_h(n) = C_1\lambda_1^n + C_2\lambda_2^n + \ldots + C_N\lambda_N^n$$

    - where C_1, C_2, ..., C_N are weighting coefficients which are determined from the initial conditions specified for the system.

Example 2.4.4

- Determine the homogeneous solution of the system described the first-order difference equation.

$$y(n) + a_1 y(n - 1) = x(n) \qquad (2.4.18)$$

- Solution. The assumed solution obtained by setting x(n)=0 is

$$y_h(n) = \lambda^n$$

  - when we substitute this solution in (2.4.18), we obtain [with x(n)=0]

$$\lambda^n + a_1 \lambda^{n-1} = 0$$
$$\lambda^{n-1}(\lambda + a_1) = 0$$
$$\lambda = -a_1$$

  - Therefore, the solution to the homogeneous difference equation is

$$y_h(n) = C\lambda^n = C(-a_1)^n \qquad (2.4.19)$$

Example 2.4.4 (cont)

- The zero-input response of the system can be determined from (2.4.18) and (2.4.19) [with x(n)=0], (2.4.18) yields

$$y(0) = -a_1 y(-1)$$

- On the other hand, from (2.4.19) we have

$$y_h(0) = C$$

- and hence the zero-input response of the system is

$$y_{zi}(n) = (-a_1)^{n+1} y(-1)\,, \qquad n \geq 0 \qquad \text{(2.4.20)}$$

The Particular Solution of A Difference Equation

- The particular solution $y_p(n)$ is required to satisfy the differece equation for the specific input sigal x(n), $n \geq 0$. In other words, $y_p(n)$ is any solution satifing

$$\sum_{k=0}^{N} a_k y_p(n-k) = \sum_{k=0}^{M} b_k x(n-k) \qquad a_0 \equiv 1$$

x(n)	$y_p(n)$
A	K
Am^n	KM^n
An^M	$K_0 n^M + K_1 n^{M-1} + \ldots + K_M$
$A^n n^M$	$A^n(K_0 n^M + K_1 n^{M-1} + \ldots + K_M)$
$A\cos\omega_0 n$ $A\sin\omega_0 n$	$K_1\cos\omega_0 n + K_2\sin\omega_0 n$

Example 2.4.6

- Determine the particular solution of the first-order difference equation

$$y(n) + a_1 y(n-1) = x(n), \qquad |a_1| < 1 \qquad (2.4.26)$$

- when the input x(n) is a unit step sequence, that is,

$$x(n) = u(n)$$

- **Solution**
  - Since the input sequence x(n) is a constant for $n \geq 0$, the form of the solution that we assume is also a constant. Hence the assumed solution of the difference equation to the forcing function x(n), called the **particular solution** of the difference equation, is

$$y_p(n) = Ku(n)$$

Example 2.4.6 (cont)

- where K is a scale factor determined so that (2.4.26) is satisfied. Upon substitution of this assumed solution into (2.4.26), we obtain

$$Ku(n) + a_1 Ku(n-1) = u(n)$$

- To determine K, we must evaluate this equation for any $n \geq 1$, where none of the terms vanish. Thus,

$$K + a_1 K = 1 \Rightarrow K = \frac{1}{1 + a_1}$$

- Therefore, the particular solution to the difference equation is

$$y_p(n) = \frac{1}{1 + a_1} u(n) \qquad \textbf{(2.4.27)}$$

The Total Solution of A Difference Equation

- The **linearity property** of the linear constant-coefficient difference equation allows us to **add** the **homogeneous solution** and the **particular solution** in order to obtain the **total solution**. Thus

$$\mathbf{y(n) = y_h(n) + y_p(n)}$$

- The resultant sum y(n) contains the constant parameters {Ci} embodied in the homogeneous solution compenent $y_h(n)$. **These constants can be determined to satisfy the initial conditions**.

Example 2.4.8

- Determine the total solution y(n), $n \geq 0$, to the difference equation.

$$y(n) + a_1 y(n-1) = x(n) \qquad \textbf{(2.4.28)}$$

  - when x(n) is a unit step sequence [i.e., x(n)=u(n)] and $y(-1)$ is the initial condition.

- **Solution**
  - from (2.4.19) of example 2.4.4, the homogeneous solution is

$$y_h(n) = C(-a_1)^n$$

  - and from (2.4.26) of example 2.4.6, the particular solution is

$$y_p(n) = \frac{1}{1 + a_1} u(n)$$

Example 2.4.8 (cont)

- Consequently, the total solution is

$$y_p(n) = C(-a_1)^n + \frac{1}{1+a_1}u(n), \qquad n \geq 0 \qquad (2.4.29)$$

- where the constant C is determined to satisfy the initial condition $y(-1)$.

- In particular, suppose that we wish to obtain the zero-state response of the system described by the difference equation in (2.4.28). Then we set . To evaluate C, we evaluate (2.4.28) at n=0, obtaining

- Hence

$$y(0) + a_1 y(-1) = 1$$

$$y(0) = 1 - a_1 y(-1)$$

- On the other hand, (2.4.29) evaluated at n=0 yields

$$y(0) = C + \frac{1}{1+a_1}$$

Example 2.4.8 (cont)

- By equating these two relations, we obtain

$$C + \frac{1}{1 + a_1} = -a_1 y(-1) + 1 \Rightarrow C = -a_1 y(-1) + \frac{a_1}{1 + a_1}$$

- Finally, if we substitute this value of C into (2.4.9), we obtained

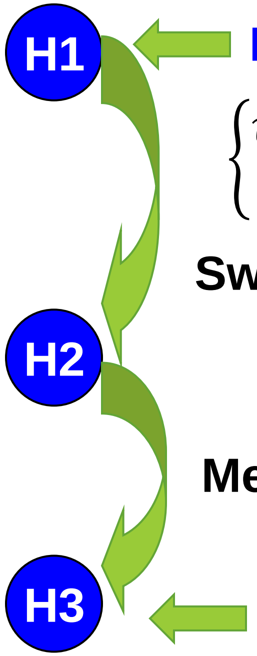
$$\begin{aligned} y(n) &= (-a_1)^{n+1} + \frac{1 - (-a_1)^{n+1}}{1 + a_1}, \qquad n \geq 0 \qquad \text{(2.4.30)} \\ &= y_{zi}(n) + y_{zs}(n) \end{aligned}$$

Structure for the Realization of LTI Systems

- Given first-order system

$y(n) = a_1 y(n-1) + b_0 x(n) + b_1 x(n-1)$

- Structures



Direct Form I Structure

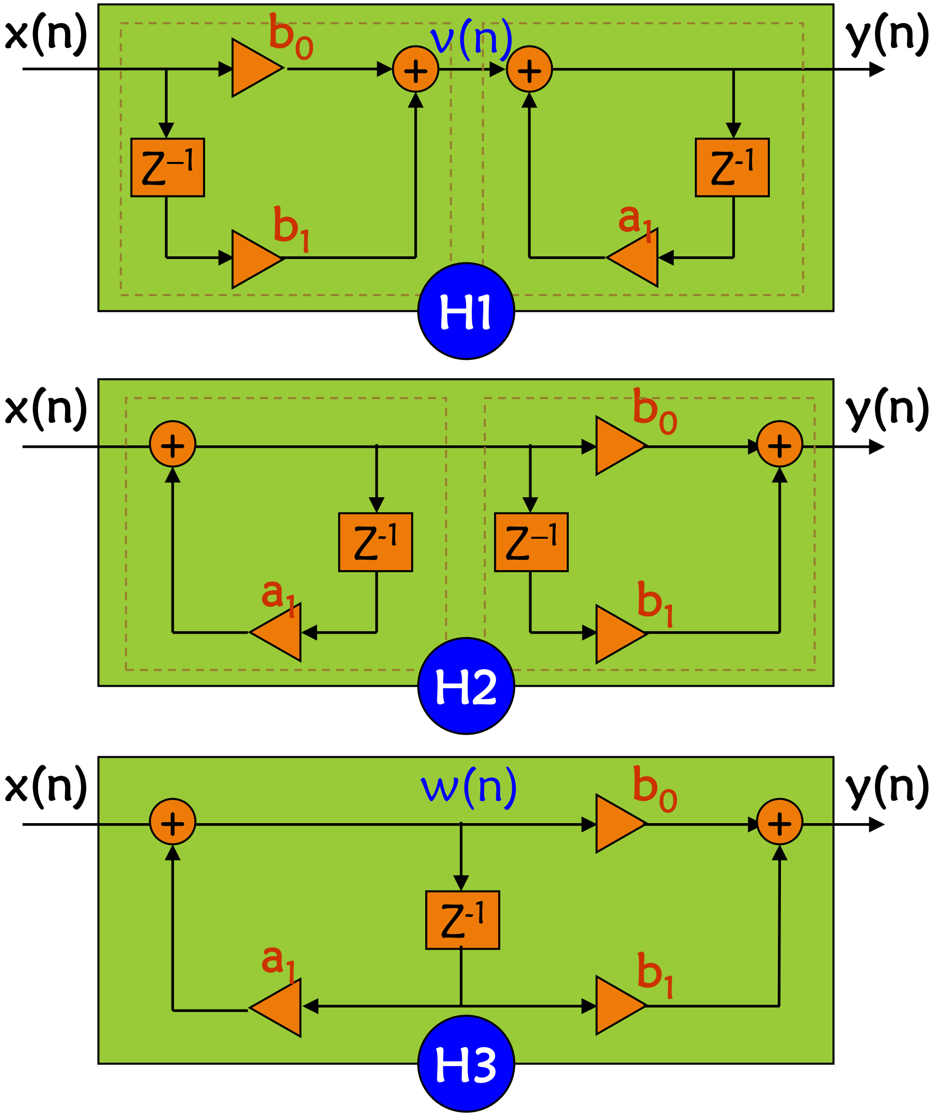
$$\begin{cases} v(n) = b_0 x(n) + b_1 x(n-1) \\ y(n) = a_1 y(n-1) + v(n) \end{cases}$$

Switch two sub-systems

Merge memories

Direct Form II Structure

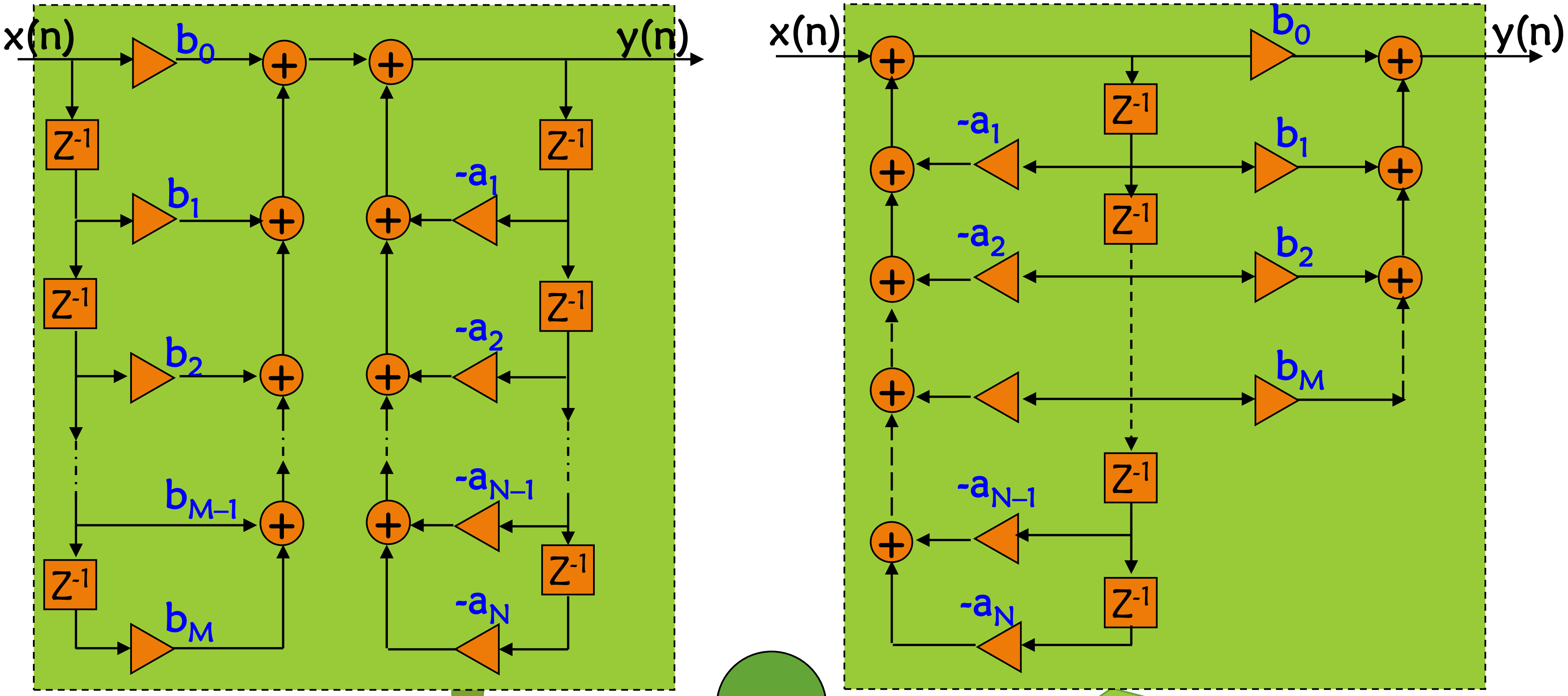
$$\begin{cases} w(n) = a_1 w(n-1) + x(n) \\ y(n) = b_0 w(n) + b_1 w(n-1) \end{cases}$$



Structure for the Realization of LTI Systems

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k)$$

Direct Form I Structure **Direct Form II Structure**



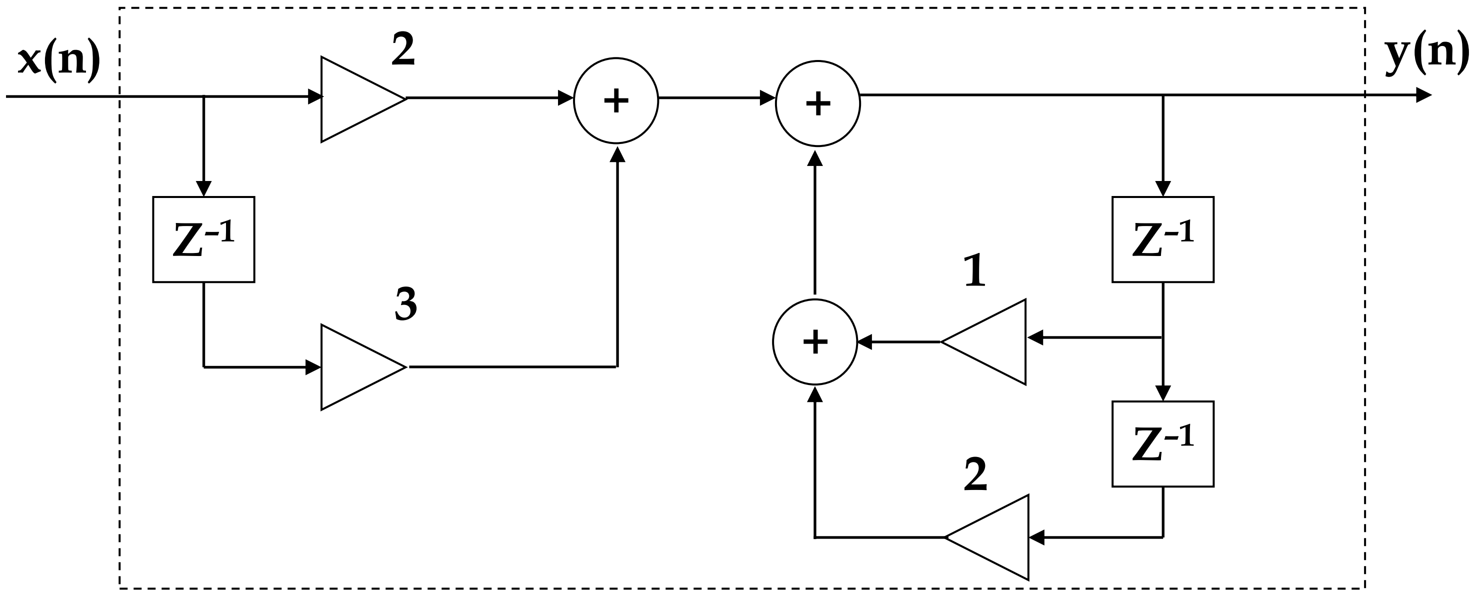
#memory: M+N

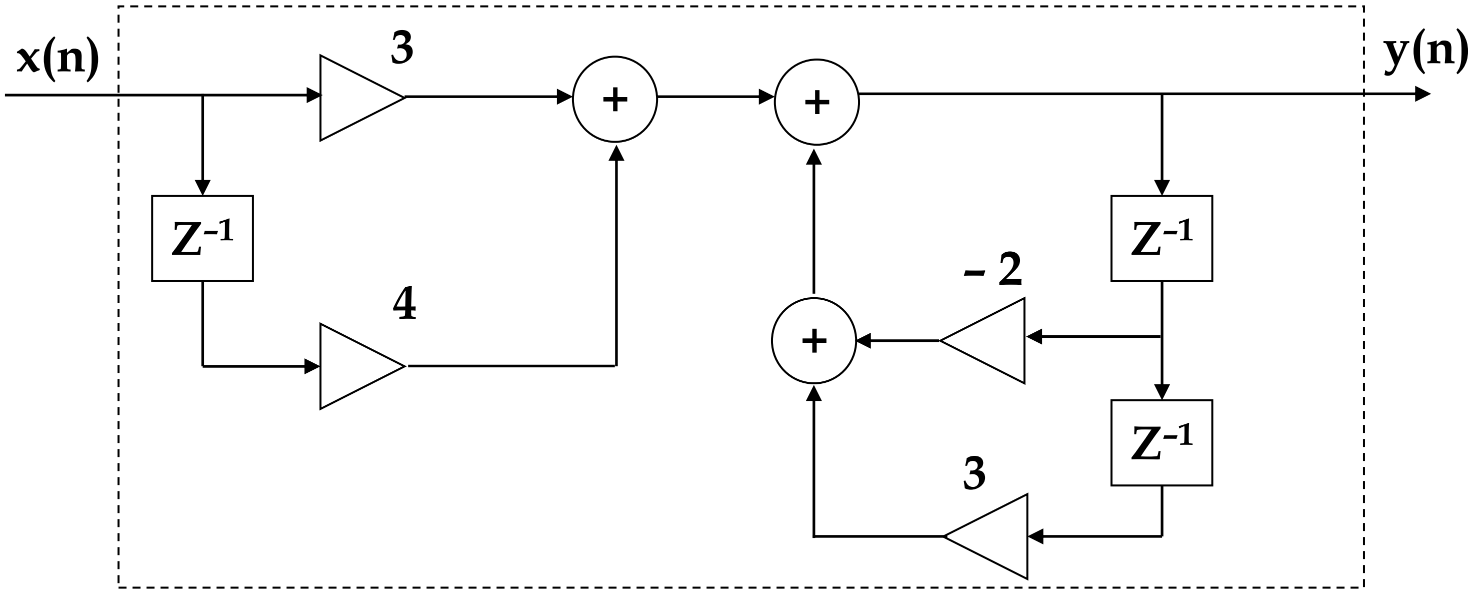
#memory: Max(M,N)

References

- Textbook "Digital Signal Processing: Principles, Algorithms, and Applications", 4th Edition, Prentice Hall.
  - John G. Proakis, Dimitris G. Manolakis

- Lecture Notes – Digital Signal Processing
  - Professor Deepa Kundur (University of Toronto)
  - http://www.comm.utoronto.ca/~dkundur/course/ece-455-digital-signal-processing/#lectures


x(n)
2
+
+
y(n)
Z-1
3
Z-1
1
+
Z-1
2


x(n)
3
+
+
y(n)
Z-1
4
Z-1
– 2
+
Z-1
3