Discrete-Time Signals and Systems

Professor Deepa Kundur

University of Toronto

Discrete-Time Signals and Systems

Reference:

Sections 2.1-2.5 of

John G. Proakis and Dimitris G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 4th edition, 2007.

2.1 Discrete-Time Signals

Elementary Discrete-Time Signals

1. unit sample sequence (a.k.a. Kronecker delta function):

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

2. unit step signal:

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

3. unit ramp signal:

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

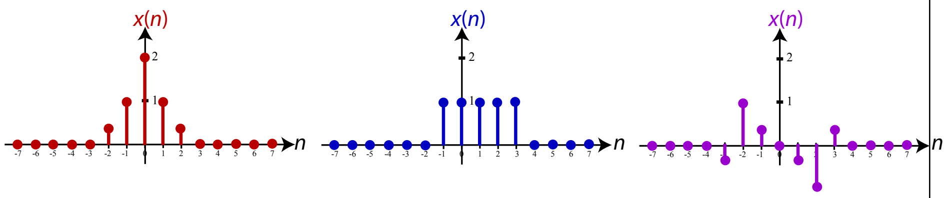
Note:

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

Signal Symmetry

Even signal: $x(-n) = x(n)$

Odd signal: $x(-n) = -x(n)$



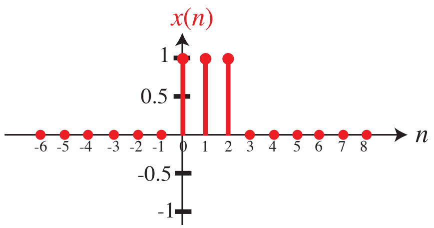
Signal Symmetry

Even signal component: $x_e(n) = \frac{1}{2}[x(n) + x(-n)]$

Odd signal component: $x_o(n) = \frac{1}{2}[x(n) - x(-n)]$

Note: $x(n) = x_e(n) + x_o(n)$

Signal Symmetry

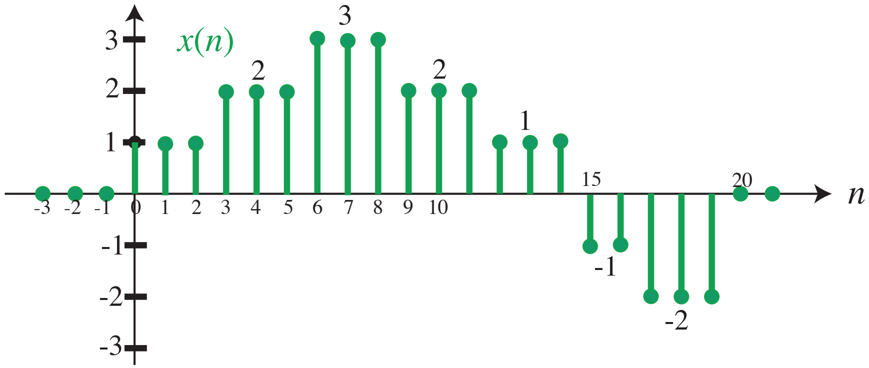


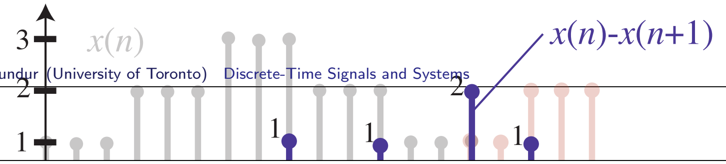
Simple Manipulation of Discrete-Time Signals

- Transformation of independent variable:
  - time shift: $n \rightarrow n - k$, $k \in \mathbb{Z}$
    - Question: what if $k \notin \mathbb{Z}$?
  - time scale: $n \rightarrow \alpha n$, $\alpha \in \mathbb{Z}$
    - Question: what if $\alpha \notin \mathbb{Z}$?
- Additional, multiplication and scaling:
  - amplitude scaling: $y(n) = Ax(n)$, $-\infty < n < \infty$
  - sum: $y(n) = x_1(n) + x_2(n)$, $-\infty < n < \infty$
  - product: $y(n) = x_1(n)x_2(n)$, $-\infty < n < \infty$

Simple Manipulation of Discrete-Time Signals

Find $x(n) - x(n+1)$.





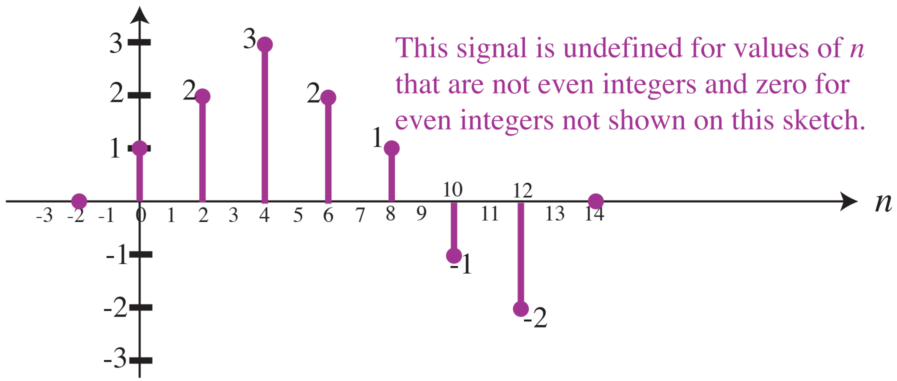
Simple Manipulation of Discrete-Time Signals I

Find $x(\frac{3}{2}n + 1)$.

n	$\frac{3n}{2} + 1$	$x(\frac{3n}{2} + 1)$
< -1	$< -\frac{1}{2}$	0 if $\frac{3n}{2} + 1$ is an integer; undefined otherwise
-1	$-\frac{1}{2}$	undefined
0	1	$x(1) = 1$
1	$\frac{5}{2}$	undefined
2	4	$x(4) = 2$
3	$\frac{11}{2}$	undefined
4	7	$x(7) = 3$
5	$\frac{17}{2}$	undefined
6	10	$x(10) = 2$
7	$\frac{23}{2}$	undefined
8	13	$x(13) = 1$
9	$\frac{29}{2}$	undefined
10	16	$x(16) = -1$
11	$\frac{35}{2}$	undefined
12	19	$x(19) = -2$
> 12	> 19	0 if $\frac{3n}{2} + 1$ is an integer; undefined otherwise

Simple Manipulation of Discrete-Time Signals

Graph of $x(\frac{3}{2}n + 1)$.



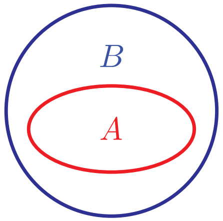
2.2 Discrete-Time Systems

Terminology: Implication

If "A" then "B" Shorthand: $A \implies B$

Example 1:
it is snowing $\implies$ it is at or below freezing temperature

Example 2:
$\alpha \geq 5.2 \implies \alpha$ is positive
Note: For both examples above, $B \not\Longleftarrow A$



Terminology: Equivalence

If "A" then "B" Shorthand: $A \implies B$

and

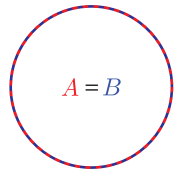
If "B" then "A" Shorthand: $B \implies A$

can be rewritten as

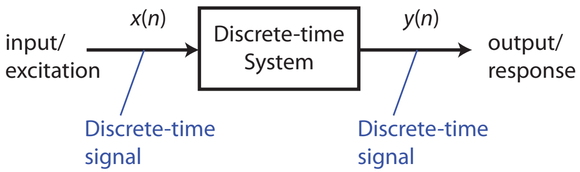
"A" if and only if "B" Shorthand: $A \iff B$

We can also say:

- A is EQUIVALENT to B
- $A = B$



Terminology: Input-Output Description



- Input-output description (exact structure of system is unknown or ignored):
$$y(n) = \mathcal{T}[x(n)]$$
- "black box" representation:
$$x(n) \xrightarrow{\mathcal{T}} y(n)$$

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.	nonlinear
causal	vs.	non-causal
stable	vs.	unstable systems
$\vdots$		$\vdots$

Static vs. Dynamic

- Static system (a.k.a. memoryless): the output at time n depends only on the input sample at time n; otherwise the system is said to be dynamic
- a system is static iff (if and only if)

$$y(n) = \mathcal{T}[x(n), n]$$

for every time instant n.

Static vs. Dynamic

- 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 $N = M = 0$, $y(n) = \mathcal{T}[x(n)]$, the system is static.
  - For $0 < N, M < \infty$, the system is said to be dynamic with finite memory of duration $N + M + 1$.
  - For either N and/or M equal to infinite, the system is said to have infinite memory.

Static vs. Dynamic

Examples: memoryless or not?

- $y(n) = A\, x(n),\ A \neq 0$
- $y(n) = A\, x(n) + B,\ A, B, \neq 0$
- $y(n) = x(n) \cos(\frac{\pi}{25}(n-5))$
- $y(n) = x(-n)$
- $y(n) = x(n+1)$
- $y(n) = \frac{1}{1 - x(n+2)}$
- $y(n) = e^{3x(n)}$
- $y(n) = \sum_{k=-\infty}^{n} x(k)$

Ans: Y, Y, Y, N, N, N, Y, N

Time-invariant vs. Time-variant Systems

- Time-invariant system: input-output characteristics do not change with time
- a system is time-invariant iff

$$x(n) \xrightarrow{\mathcal{T}} y(n) \implies x(n-k) \xrightarrow{\mathcal{T}} y(n-k)$$

for every input $x(n)$ and every time shift k.

Time-invariant vs. Time-variant Systems

Examples: time-invariant or not?

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

Ans: Y, Y, N, N, Y, Y, Y, Y

Linear vs. Nonlinear Systems

- Linear system: obeys superposition principle
- a system is linear iff

$$\mathcal{T}[a_1\, x_1(n) + a_2\, x_2(n)] = a_1\, \mathcal{T}[x_1(n)] + a_2\, \mathcal{T}[x_2(n)]$$

for any arbitrary input sequences $x_1(n)$ and $x_2(n)$, and any arbitrary constants a_1 and a_2

Linear Systems: Homogeneity

A system is linear iff

$$\mathcal{T}[a_1\, x_1(n) + a_2\, x_2(n)] = a_1\, \mathcal{T}[x_1(n)] + a_2\, \mathcal{T}[x_2(n)]$$

- Homogeneity: Let $a_2 = 0$.

$$\mathcal{T}[a_1\, x_1(n)] = a_1\, \mathcal{T}[x_1(n)]$$

$$x(n) \xrightarrow{\mathcal{T}} y(n) \implies a_1 x(n) \xrightarrow{\mathcal{T}} a_1 y(n)$$

for any constant a_1.

Linear Systems: Additivity

A system is linear iff

$$\mathcal{T}[a_1\, x_1(n) + a_2\, x_2(n)] = a_1\, \mathcal{T}[x_1(n)] + a_2\, \mathcal{T}[x_2(n)]$$

- Additivity: Let $a_1 = a_2 = 1$.

$$\mathcal{T}[x_1(n) + x_2(n)] = \mathcal{T}[x_1(n)] + \mathcal{T}[x_2(n)]$$

$$\begin{array}{c} x_1(n) \xrightarrow{\mathcal{T}} y_1(n) \\ x_2(n) \xrightarrow{\mathcal{T}} y_2(n) \end{array} \implies x_1(n) + x_2(n) \xrightarrow{\mathcal{T}} y_1(n) + y_2(n)$$

for <u>any</u> input sequences $x_1(n)$ and $x_2(n)$.

Linear Systems: Additivity

Therefore:

Linearity = Homogeneity + Additivity

Need both!
If a system is <u>not</u> homogeneous, it is <u>not</u> linear.
If a system is <u>not</u> additive, it is <u>not</u> linear.

Linear vs. Nonlinear Systems

Examples: linear or not?

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

Ans: Y, N, Y, Y, Y, N, N, Y

Causal vs. Noncausal Systems

- Causal system: output of system <u>at any time</u> n depends only on present and past inputs
- a system is causal iff

$$y(n) = \mathcal{T}\left[x(n), x(n-1), x(n-2), \ldots\right]$$

<u>for all</u> n

Causal vs. Noncausal Systems

Examples: causal or not?

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

Ans: Y, Y, Y, N, N, N, Y, Y

Stable vs. Unstable Systems

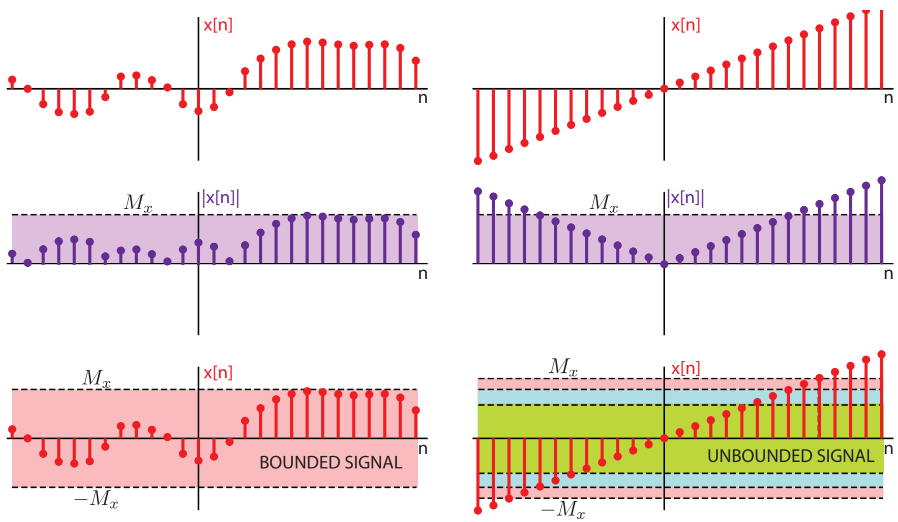
- Bounded Input-Bounded output (BIBO) Stable: every bounded input produces a bounded output

- a system is BIBO stable iff

$$|x(n)| \leq M_x < \infty \implies |y(n)| \leq M_y < \infty$$

for all n.

Discrete-Time Bounded Signals



Stable vs. Unstable Systems

Examples: stable or not?

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

Ans: Y, Y, Y, Y, Y, N, Y, N

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

2.3 Analysis of Dst-Time Linear Time-Invaraint Systems

The Convolution Sum

Recall:

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

The Convolution Sum

Let the response of a linear time-invariant (LTI) system to the unit sample input $\delta(n)$ be $h(n)$.

$$\begin{aligned}
\delta(n) &\xrightarrow{\mathcal{T}} h(n) \\
\delta(n-k) &\xrightarrow{\mathcal{T}} h(n-k) \\
\alpha\ \delta(n-k) &\xrightarrow{\mathcal{T}} \alpha\ h(n-k) \\
x(k)\ \delta(n-k) &\xrightarrow{\mathcal{T}} x(k)\ h(n-k) \\
\sum_{k=-\infty}^{\infty} x(k)\delta(n-k) &\xrightarrow{\mathcal{T}} \sum_{k=-\infty}^{\infty} x(k)h(n-k) \\
x(n) &\xrightarrow{\mathcal{T}} y(n)
\end{aligned}$$

The Convolution Sum

Therefore,

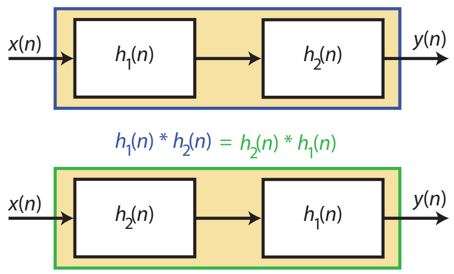
$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) = x(n) * h(n)$$

for any LTI system.

Properties of Convolution

Associative and Commutative Laws:

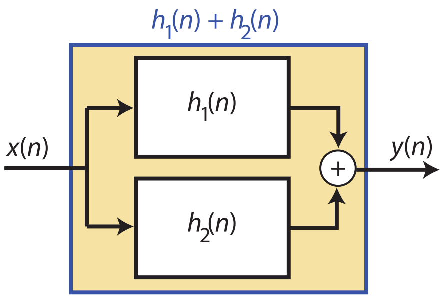
$$\begin{aligned} x(n) * h(n) &= h(n) * x(n) \\ [x(n) * h_1(n)] * h_2(n) &= x(n) * [h_1(n) * h_2(n)] \end{aligned}$$



Properties of Convolution

Distributive Law:

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



Causality and Convolution

For a causal system, $y(n)$ only depends on present and past inputs values. Therefore, for a causal system, we have:

$$\begin{aligned} y(n) &= \sum_{k=-\infty}^{\infty} h(k)x(n-k) \\ &= \sum_{k=-\infty}^{-1} h(k)x(n-k) + \sum_{k=0}^{\infty} h(k)x(n-k) \\ &= \sum_{k=0}^{\infty} h(k)x(n-k) \end{aligned}$$

where $h(n) = 0$ for $n < 0$ to ensure causality.

2.4 Discrete-time Systems Described by Difference Equations

Finite vs. Infinite Impulse Response

For causal LTI systems, $h(n) = 0$ for $n < 0$.

Finite impulse response (FIR):

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

Infinite impulse response (IIR):

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

How would one realize these systems?

Finite vs. Infinite Impulse Response

Implementation: Two classes

Finite impulse response (FIR):

$$\left. y(n) = \sum_{k=0}^{M-1} h(k)x(n-k) \quad \right\} \text{nonrecursive systems}$$

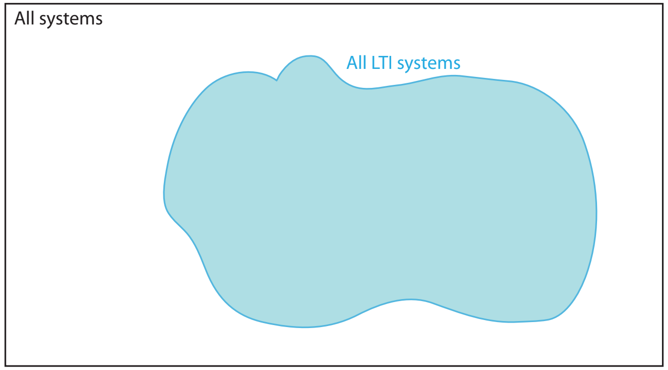
Infinite impulse response (IIR):

$$\left. y(n) = \sum_{k=0}^{\infty} h(k)x(n-k) \quad \right\} \text{recursive systems}$$

System Realization

There is a practical and computationally efficient means of implementing all FIR and a family of IIR systems that makes use of . . .

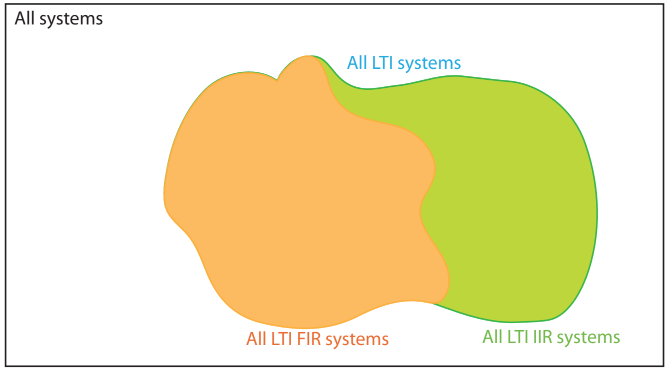
. . . difference equations.



System Realization

There is a practical and computationally efficient means of implementing all FIR and a family of IIR systems that makes use of ...

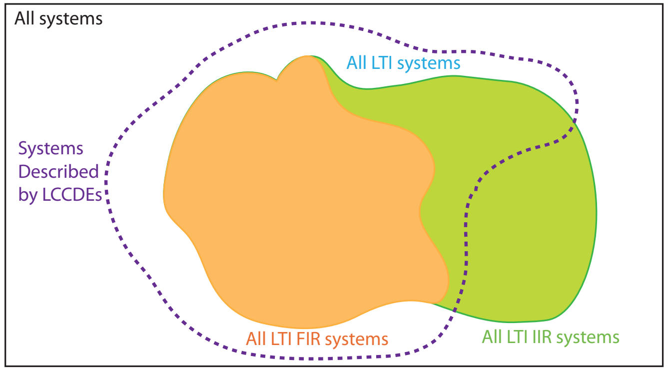
...difference equations.



System Realization

There is a practical and computationally efficient means of implementing all FIR and a family of IIR systems that makes use of ...

...difference equations.



System Realization

General expression for Nth-order LCCDE:

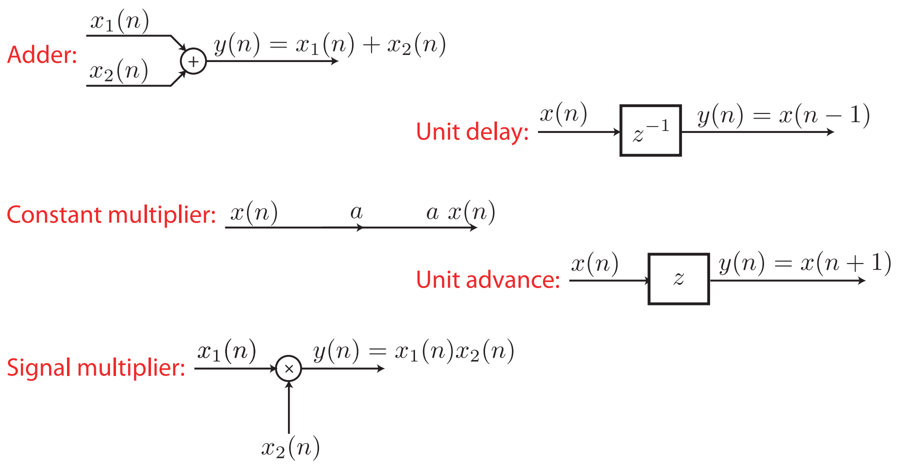
$$\sum_{k=0}^{N} a_k y(n-k) = \sum_{k=0}^{M} b_k x(n-k) \qquad a_0 \triangleq 1$$

Initial conditions: $y(-1), y(-2), y(-3), \ldots, y(-N)$.

Need: (1) constant scale, (2) addition, (3) delay elements.

2.5 Implementation of Discrete-time Systems

Building Block Elements



FIR System Realization

Finite Impulse Response Systems and Nonrecursive Implementation

FIR System Realization: Example

- Consider a 5-point local averager:

$$y(n) = \frac{1}{5} \sum_{k=n-4}^{n} x(k) \quad n = 0, 1, 2, \ldots$$

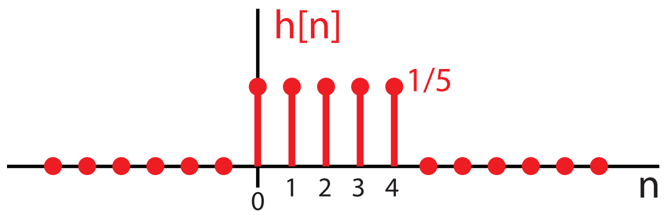
- The impulse response is given by:

$$\begin{aligned} h(n) &= \frac{1}{5} \sum_{k=n-4}^{n} \delta(k) \\ &= \frac{1}{5}\delta(n-4) + \frac{1}{5}\delta(n-3) + \frac{1}{5}\delta(n-2) + \\ &\qquad \frac{1}{5}\delta(n-1) + \frac{1}{5}\delta(n) \end{aligned}$$

FIR System Realization: Example

- Consider a 5-point local averager:

$$y(n) = \frac{1}{5} \sum_{k=n-4}^{n} x(k) \quad n = 0, 1, 2, \ldots$$



Indeed FIR!

FIR System Realization: Example

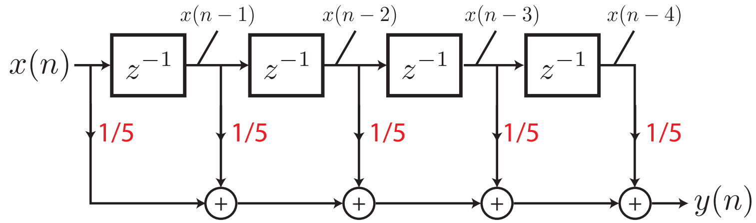
- Consider a 5-point local averager:

$$y(n) = \frac{1}{5}\sum_{k=n-4}^{n} x(k) \quad n = 0, 1, 2, \ldots$$

- Memory requirements stay constant; only need to store 5 values (4 last + present).
- fixed number of adders required

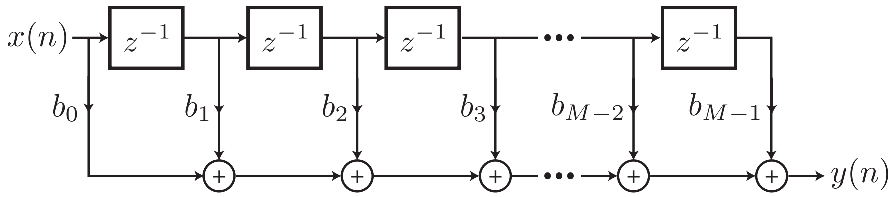
FIR System Realization: Example

$$\begin{aligned} y(n) &= \frac{1}{5}\sum_{k=n-4}^{n} x(k) = \sum_{k=n-4}^{n} \frac{1}{5}x(k) \\ \therefore\ y(n) &= \frac{1}{5}x(n-4) + \frac{1}{5}x(n-3) + \frac{1}{5}x(n-2) + \cdots \\ &\quad \cdots \frac{1}{5}x(n-1) + \frac{1}{5}x(n) \end{aligned}$$



FIR System Realization: General

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



Requires:

- M multiplications
- $M-1$ additions
- $M-1$ memory elements

FIR System Realization

Infinite Impulse Response Systems and Recursive Implementation

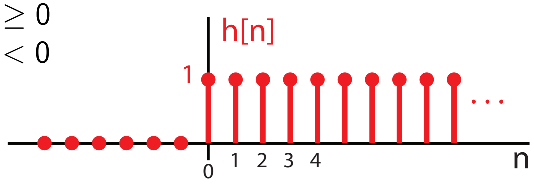
IIR System Realization: Example

- Consider an accumulator:

$$y(n) = \sum_{k=0}^{n} x(k) \quad n = 0, 1, 2, \ldots \qquad \text{for } y(-1) = 0.$$

- The impulse response is given by:

$$\begin{aligned} h(n) &= \sum_{k=0}^{n} \delta(k) = \delta(n) + \delta(n-1) + \delta(n-2) + \cdots \\ &= \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases} \end{aligned}$$



IIR System Realization: Example

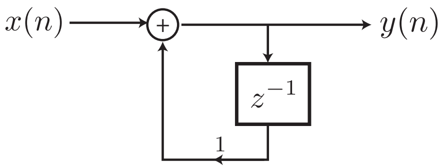
- Consider an accumulator:

$$y(n) = \sum_{k=0}^{n} x(k) \quad n = 0, 1, 2, \ldots \qquad \text{for } y(-1) = 0.$$

- IIR memory requirements seem to grow with increasing n!

IIR System Realization: Example

$$\begin{aligned} y(n) &= \sum_{k=0}^{n} x(k) \\ &= \sum_{k=0}^{n-1} x(k) + x(n) \\ &= y(n-1) + x(n) \\ \therefore\ y(n) &= y(n-1) + x(n) \end{aligned}$$



recursive implementation

Direct Form I vs. Direct Form II Realizations

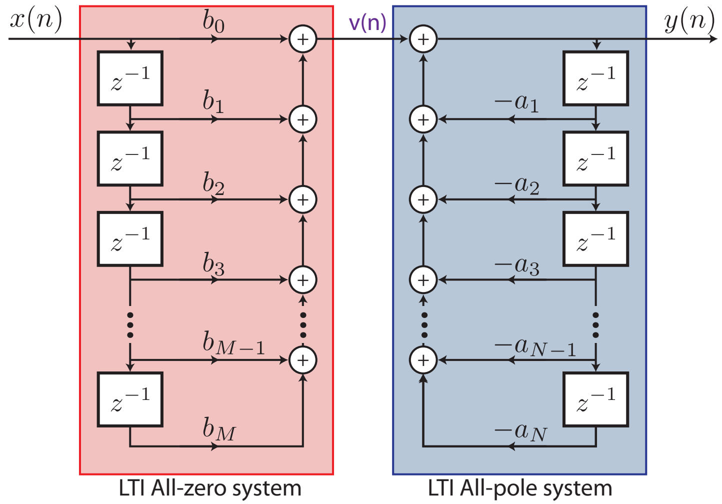
$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k)$$

is equivalent to the cascade of the following systems:

$$\underbrace{v(n)}_{\text{output 1}} = \sum_{k=0}^{M} b_k \underbrace{x(n-k)}_{\text{input 1}} \qquad \text{nonrecursive}$$

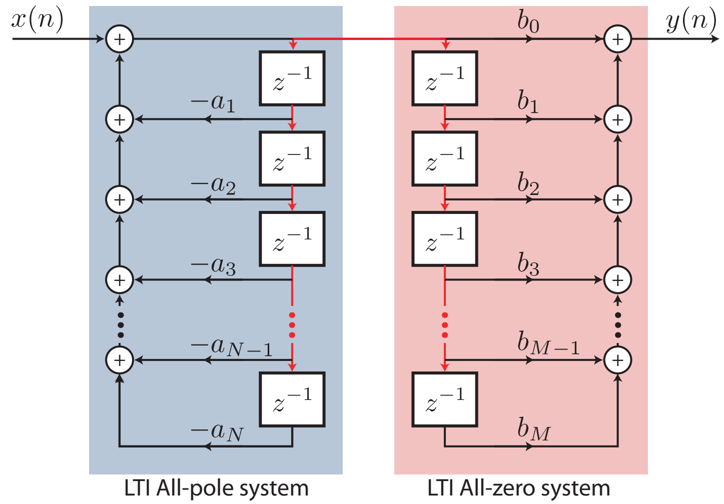
$$\underbrace{y(n)}_{\text{output 2}} = -\sum_{k=1}^{N} a_k y(n-k) + \underbrace{v(n)}_{\text{input 2}} \qquad \text{recursive}$$

Direct Form I IIR Filter Implementation



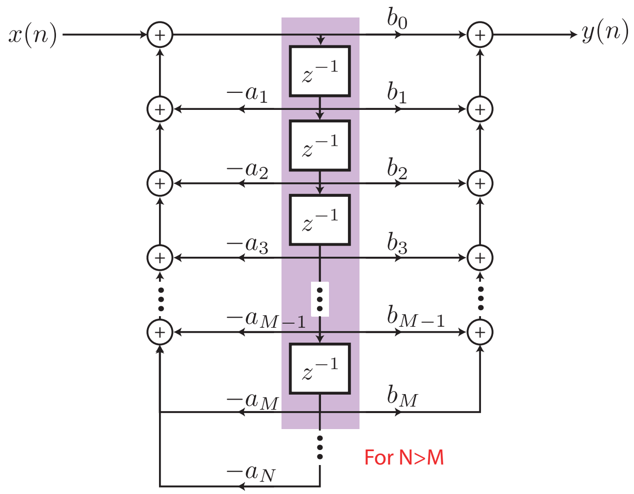
Requires: $M + N + 1$ multiplications, $M + N$ additions, $M + N$ memory locations

Direct Form II IIR Filter Implementation



Requires: $M + N + 1$ multiplications, $M + N$ additions, $M + N$ memory locations

Direct Form II IIR Filter Implementation



Requires: $M + N + 1$ multiplications, $M + N$ additions, $\max(M, N)$ memory locations