Discrete-Time Signals and Systems

Dr. 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.

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:

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

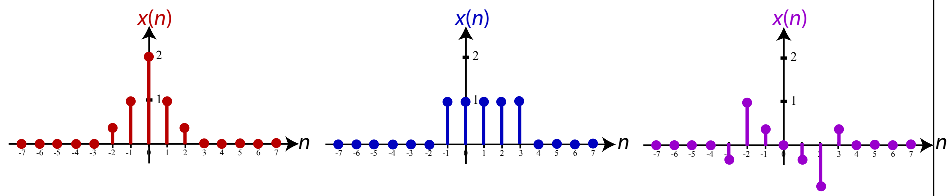
Note:

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

Signal Symmetry

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

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



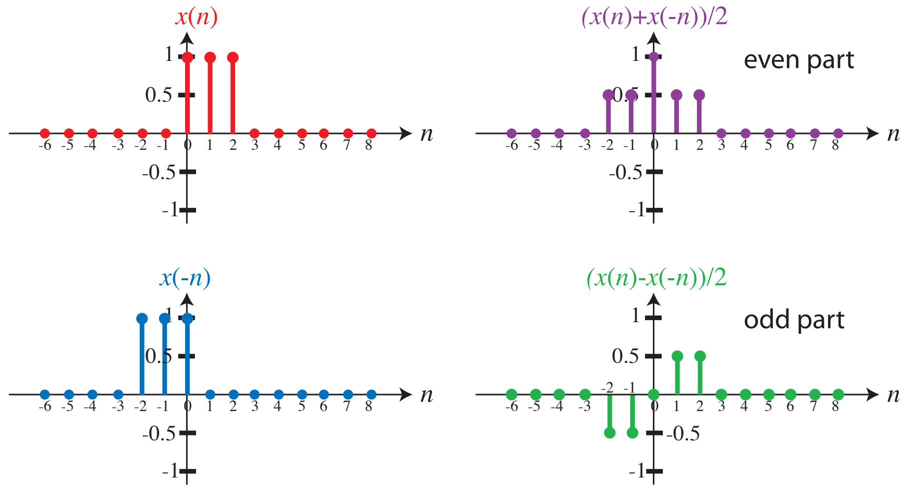
Signal Symmetry

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

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

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

Signal Symmetry

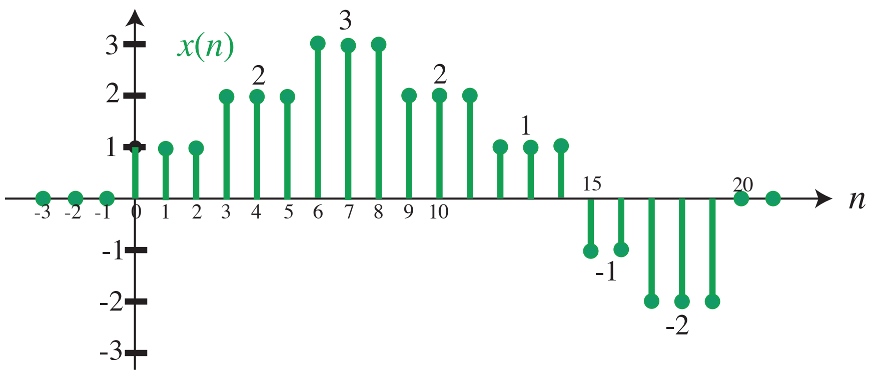


Simple Manipulation of Discrete-Time Signals

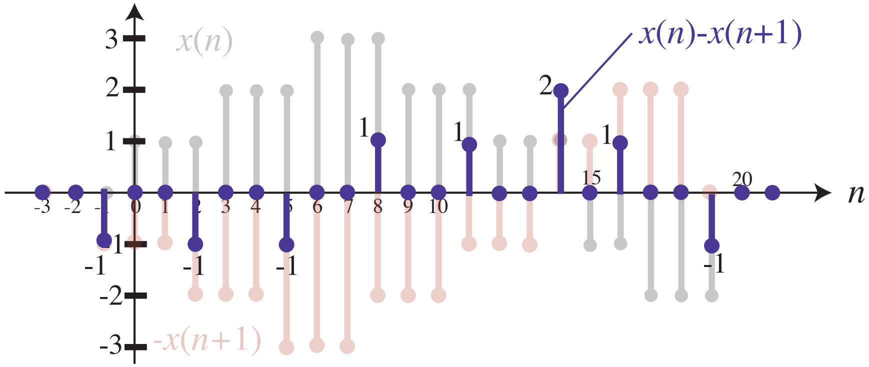
- Transformation of independent variable:
  - time shift: $n \rightarrow n - k$, $k \in \mathbb{Z}$
    - Question: what if $k \notin \mathbb{Z}$?
  - time scale: $n \rightarrow \alpha n$, $\alpha \in \mathbb{Z}$
    - Question: what if $\alpha \notin \mathbb{Z}$?
- Additional, multiplication and scaling: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

Simple Manipulation of Discrete-Time Signals I

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



Simple Manipulation of Discrete-Time Signals II



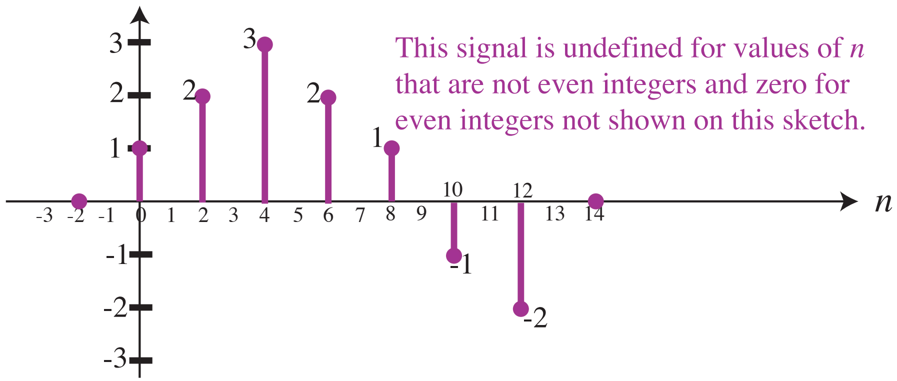
Simple Manipulation of Discrete-Time Signals I

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

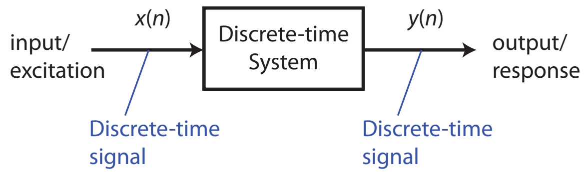
n	$\frac{3n}{2}+1$	$x(\frac{3n}{2}+1)$
< -1	$< -\frac{1}{2}$	0 if $\frac{3n}{2}+1$ is an integer; undefined otherwise
-1	$-\frac{1}{2}$	undefined
0	1	$x(1) = 1$
1	$\frac{5}{2}$	undefined
2	4	$x(4) = 2$
3	$\frac{11}{2}$	undefined
4	7	$x(7) = 3$
5	$\frac{17}{2}$	undefined
6	10	$x(10) = 2$
7	$\frac{23}{2}$	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 $x(\frac{3}{2}n+1)$.



Input-Output Description of Dst-Time Systems



- 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

Time-invariant vs. Time-variant Systems

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

- $y(n) = A\, x(n)$
- $y(n) = n\, x(n)$
- $y(n) = \sqrt{x(n)} + x^2(n-2)$
- $y(n) = x(-n)$
- $y(n) = x(n+1)$
- $y(n) = \frac{1}{1-x(n+2)}$
- $y(n) = e^{3x(n)}$

Ans: Y, N, Y, N, 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 vs. Nonlinear Systems

Examples: linear or not?

- $y(n) = A\, x(n)$
- $y(n) = n\, x(n)$
- $y(n) = \sqrt{x(n)} + x^2(n-2)$
- $y(n) = x(-n)$
- $y(n) = x(n+1)$
- $y(n) = \frac{1}{1-x(n+2)}$
- $y(n) = e^{3x(n)}$

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

Causal vs. Noncausal Systems

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

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

for all n

Causal vs. Noncausal Systems

Examples: causal or not?

- $y(n) = A\, x(n)$
- $y(n) = n\, x(n)$
- $y(n) = \sqrt{x(n)} + x^2(n-2)$
- $y(n) = x(-n)$
- $y(n) = x(n+1)$
- $y(n) = \frac{1}{1-x(n+2)}$
- $y(n) = e^{3x(n)}$

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

Stable vs. Unstable Systems

- Bounded Input-Bounded output (BIBO) Stable: every bounded input produces a bounded output
- a system is BIBO stable iff

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

for all n.

Stable vs. Unstable Systems

Examples: stable or not?

- $y(n) = A\,x(n)$
- $y(n) = n\,x(n)$
- $y(n) = \sqrt{x(n)} + x^2(n-2)$
- $y(n) = x(-n)$
- $y(n) = x(n+1)$
- $y(n) = \frac{1}{1-x(n+2)}$
- $y(n) = e^{3x(n)}$

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

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,

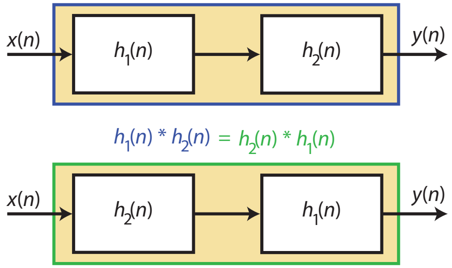
$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) = x(n) * h(n)$$

for any LTI system.

Properties of Convolution

Associative and Commutative Laws:

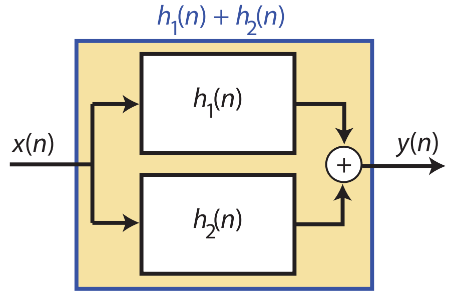
$$\begin{aligned} x(n) * h(n) &= h(n) * x(n) \\ [x(n) * h_1(n)] * h_2(n) &= x(n) * [h_1(n) * h_2(n)] \end{aligned}$$



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.

Finite-Impulse Reponse vs. Infinite Impulse Response

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? Two classes: recursive and nonrecursive.

Infinite Impulse Response System Realization

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

...difference equations.

Infinite Impulse Response System Realization

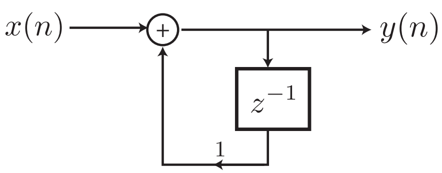
- Consider an accumulator:

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

- Memory requirements grow with increasing n!

Infinite Impulse Response System Realization

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



recursive implementation

Linear Constant-Coefficient Difference Equations

- Example for a linear constant-coefficient difference equation (LCCDE):

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

Initial conditions: at rest for $n < 0$; i.e., $y(-1) = 0$

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

Direct Form I vs. Direct Form II Realizations

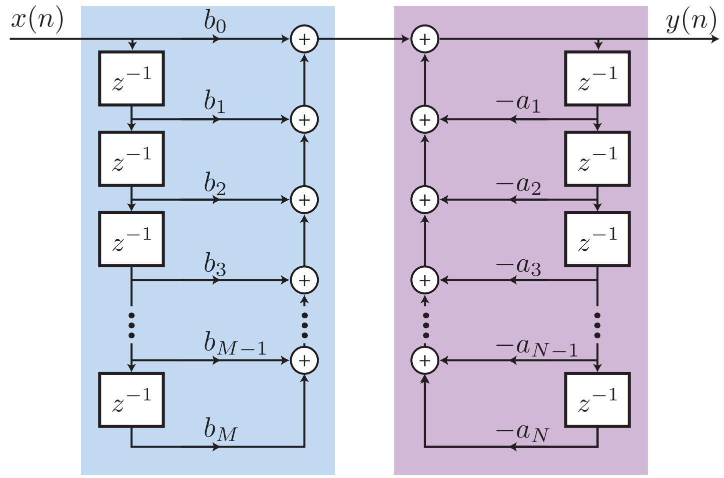
$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k)$$

is equivalent to the cascade of the following systems:

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

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

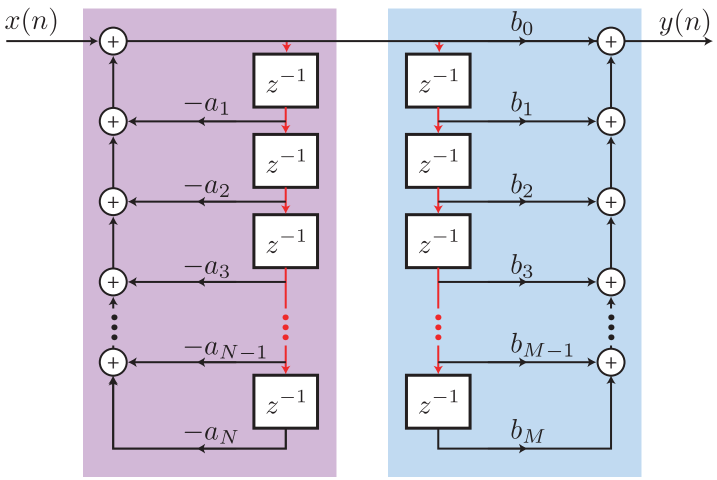
Direct Form I IIR Filter Implementation



LTI All-zero system LTI All-pole system

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

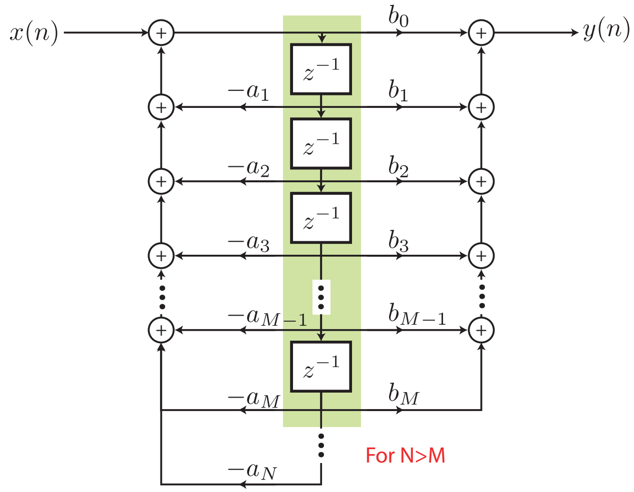
Direct Form II IIR Filter Implementation



LTI All-pole system LTI All-zero system

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 ■