Digital LTI Systems: Classification

EE 453 / CE 352
Saad Baig

LTI Systems Classification: FIR vs IIR

- We have seen that a discrete-time LTI system is completely characterized by its impulse response $h(n)$.
- It is also convenient to classify LTI systems into two types based on the duration of impulse response:
- Finite Impulse Response (FIR) Systems
  - These are the LTI systems whose impulse response has a finite duration.
- Infinite Impulse Response (IIR) Systems
  - These are the LTI systems whose impulse response has an infinite duration.

Finite Impulse Response (FIR) Systems

- If we focus on causal FIR systems:

$$h[n] = 0, \qquad\qquad n < 0 \text{ and } n \geq M$$

- The convolution formula for such a system reduces to:

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

Finite Impulse Response (FIR) Systems

- Useful Interpretation:
  - The output at any time n is simply a weighted linear combination of the input signal samples $x(n), x(n-1), \dots, x(n-M+1)$.
  - The system simply weighs, by the values of the impulse response $h(k), k = 0,1,\dots,M-1$, the most recent M signal samples and sums the resulting M products.
  - In forming the output at a certain instant, the system acts a window that views only the most recent M input signal samples and forgets all the prior input samples (i.e. $x(n-M), x(n-M-1), \dots$). Therefore, we say that the system has a finite memory of length M samples.
- The convolution formula provides a valid means of realization or implementation of the system. Such a realization involves additions, multiplications, and a finite number of memory locations.

Infinite Impulse Response (IIR) Systems

- If we focus on causal IIR systems:

$$h[n] = 0, \qquad\qquad n < 0$$

- The convolution formula for such a system reduces to:

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

Infinite Impulse Response (IIR) Systems

- Useful Interpretation
  - Now the system output at any time n is a weighted (by the impulse response $h(k)$) linear combination of the input signal samples at $x(n), x(n-1), x(n-2), \ldots.$
  - Since this weighted sum involves the present and all the past input samples, we say that the system has infinite memory.
- The practical implementation as implied by the convolution formula is clearly impossible, since it requires an infinite number of memory locations, multiplications, and additions.

How do you then implement a IIR system?

- There is a practically and computationally efficient means of implementing a family of IIR systems that makes use of difference equations.

- This family or subclass of IIR systems is very useful in a variety of practical implementation, including the implementation of digital filters.

Linear Constant Coefficient Difference Equations

- We will focus our attention on a family of LTI systems described by LCCDEs.
- Example: $y(n) = y(n-1) + x(n)$

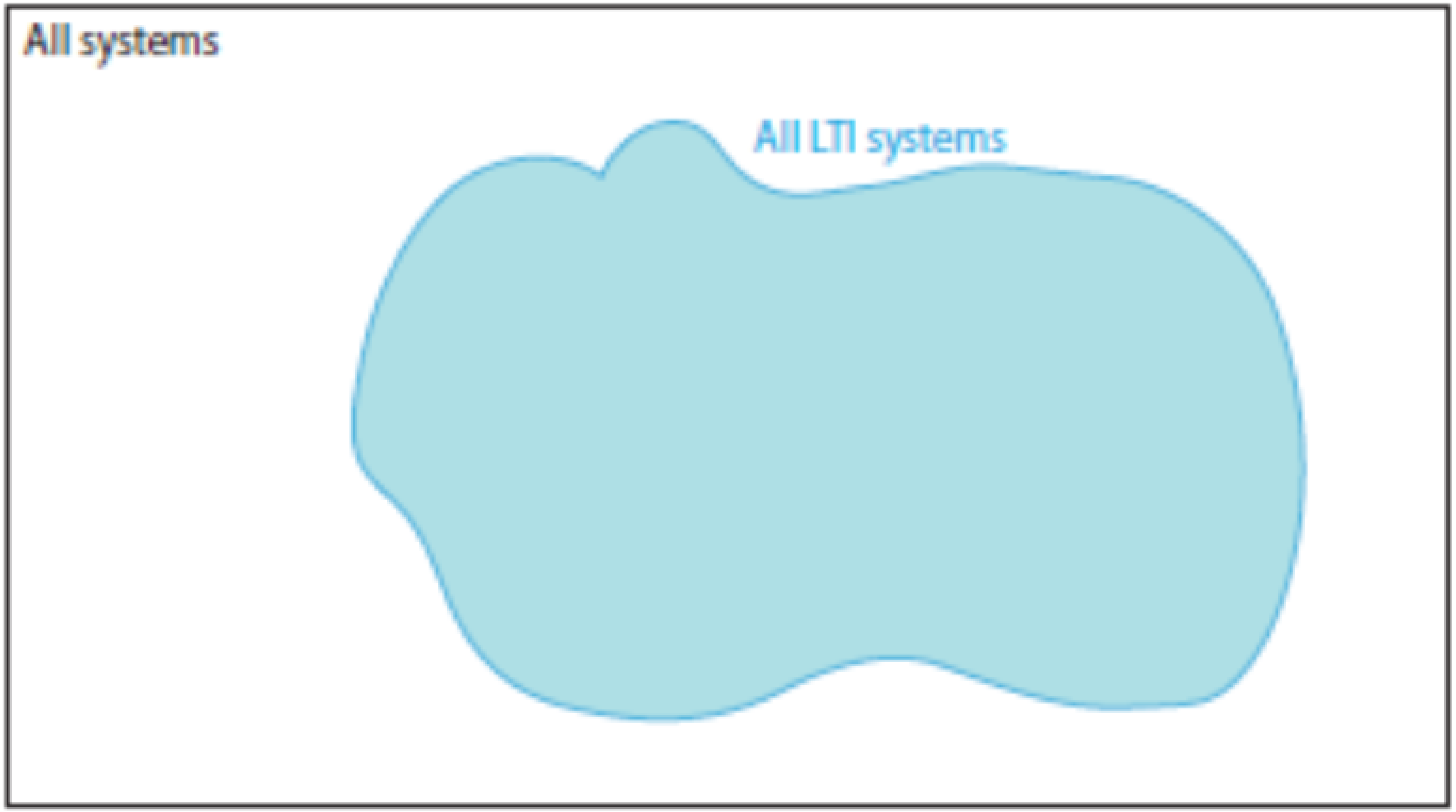
Initial conditions: at rest for $n < 0$, i.e., $y(-1) = 0$

General expression for N^{th} order LCCDE:

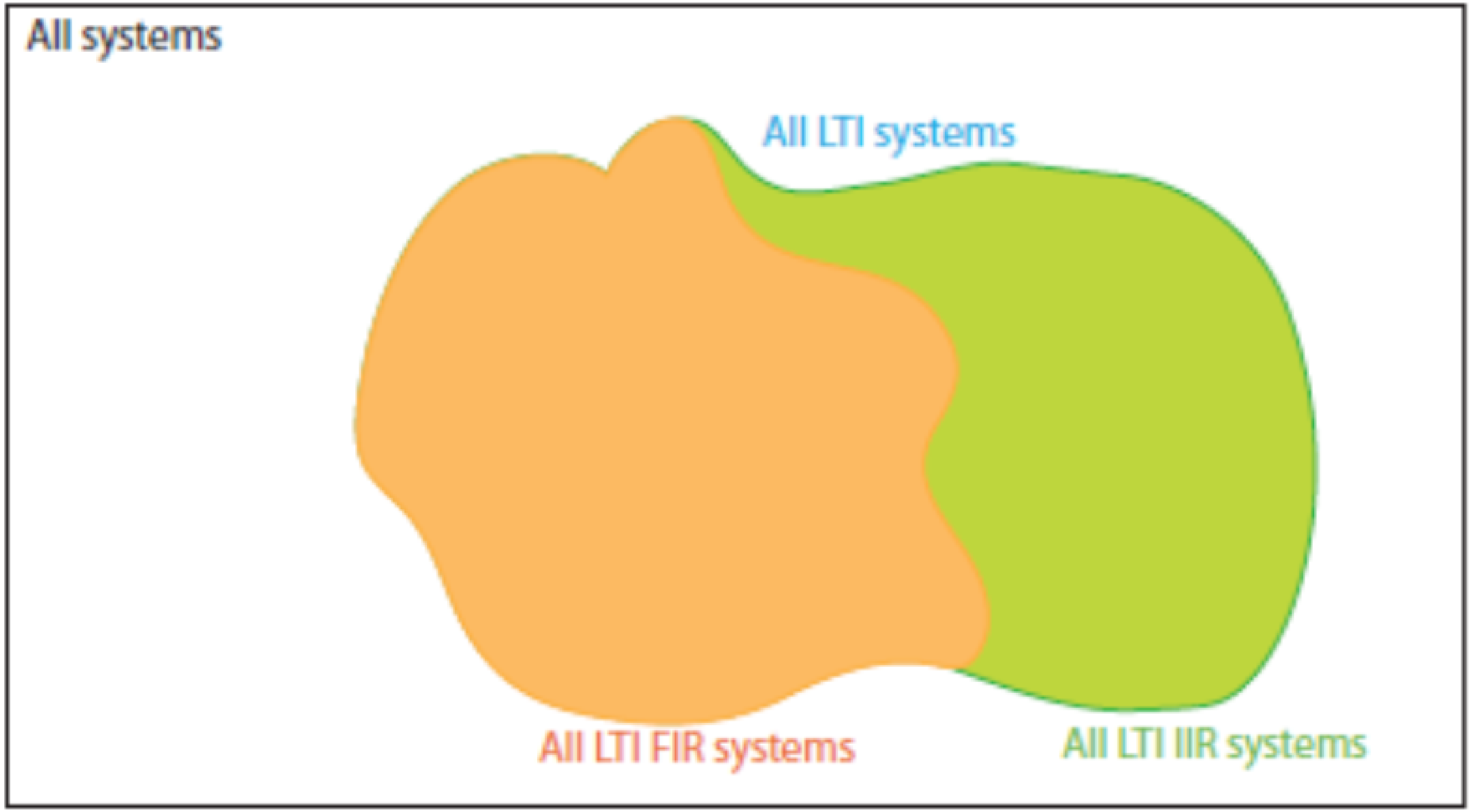
$$\sum_{k=0}^{N} a_k y(n-k) = \sum_{k=0}^{M} b_k x(n-k) \qquad a_0 \triangleq 1$$

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

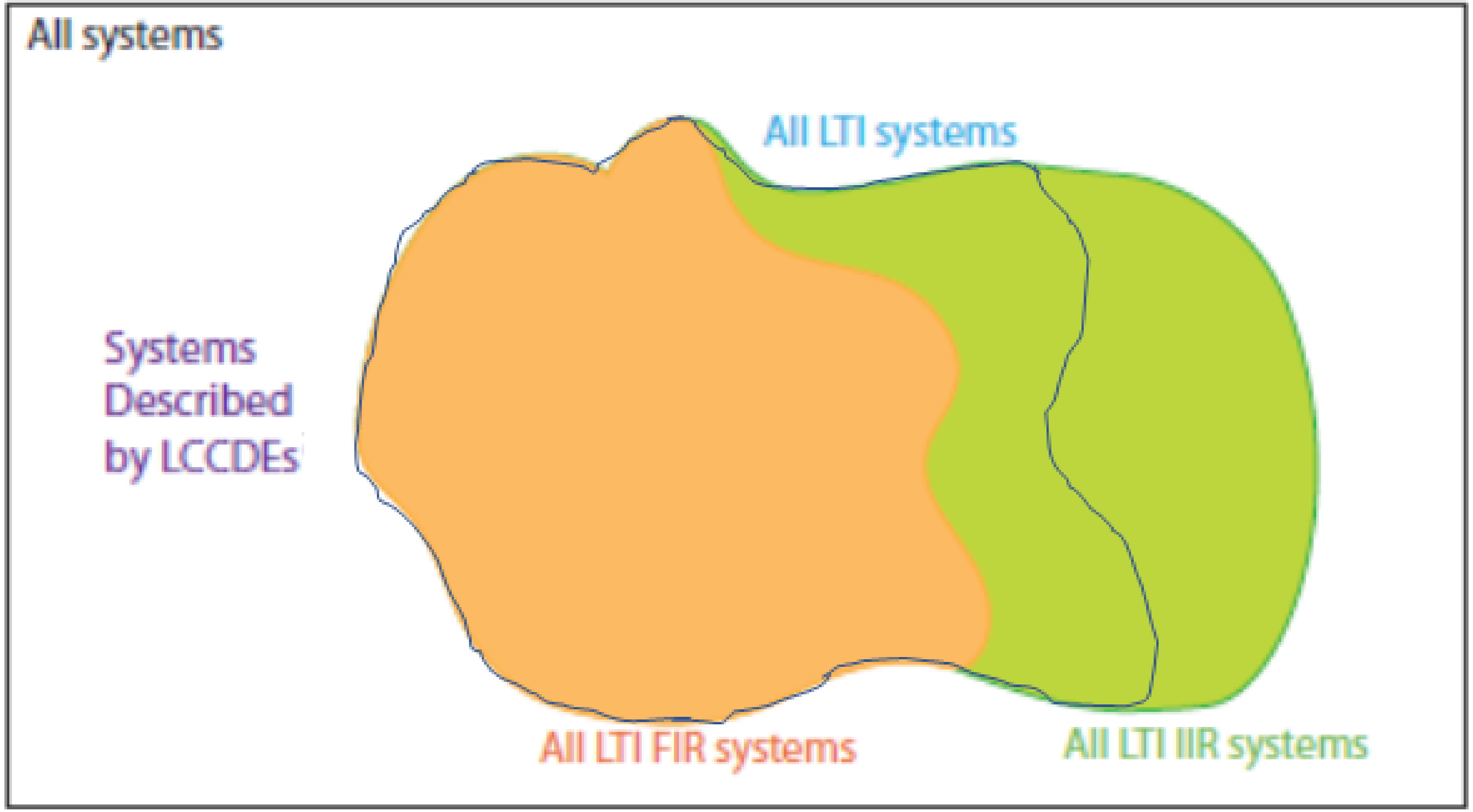
Linear Constant Coefficient Difference Equations (LCCDEs)



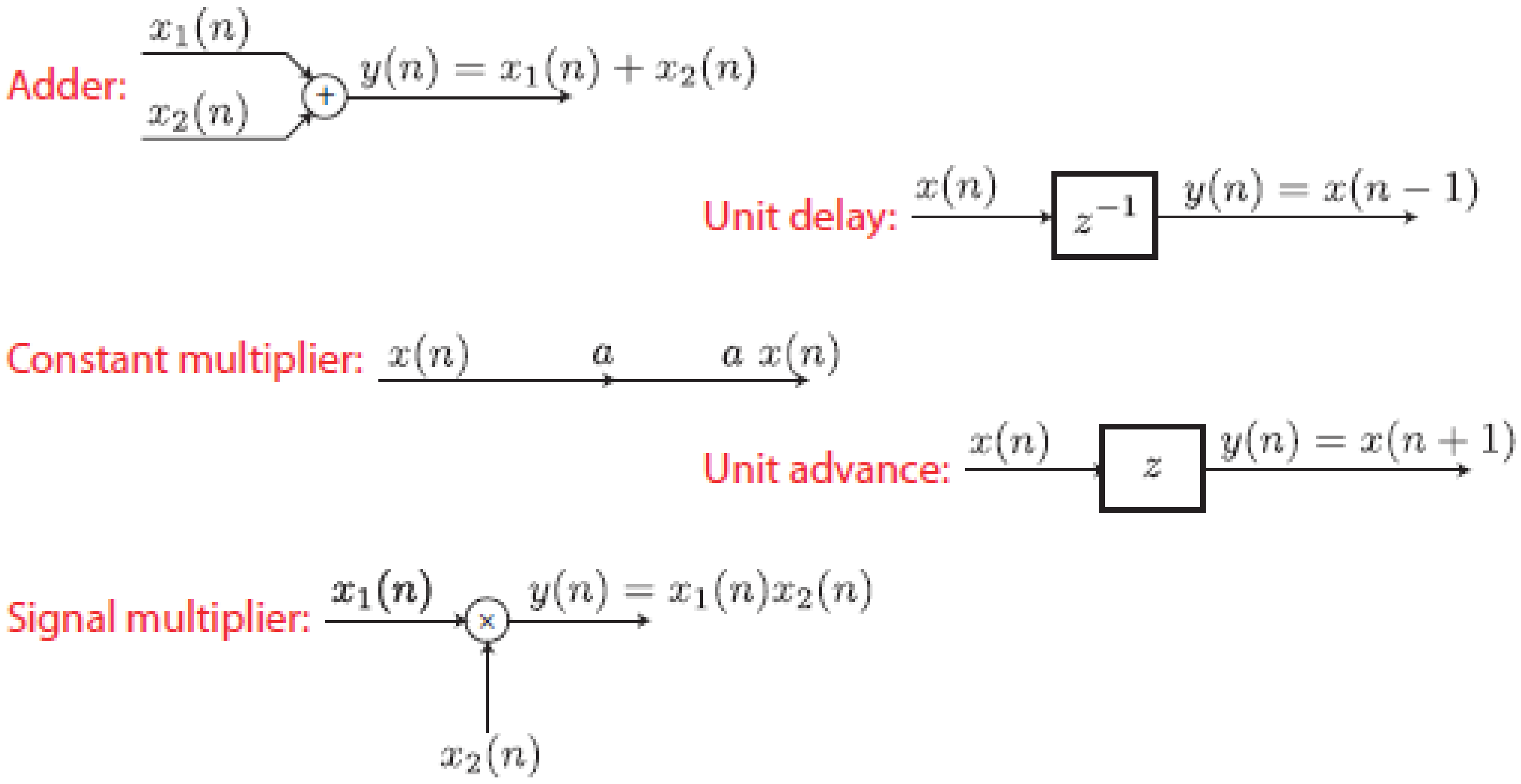
Linear Constant Coefficient Difference Equations (LCCDEs)



Linear Constant Coefficient Difference Equations (LCCDEs)



System Realization: Building Blocks



FIR System Realization

Finite Impulse Response Systems and Nonrecursive Implementation

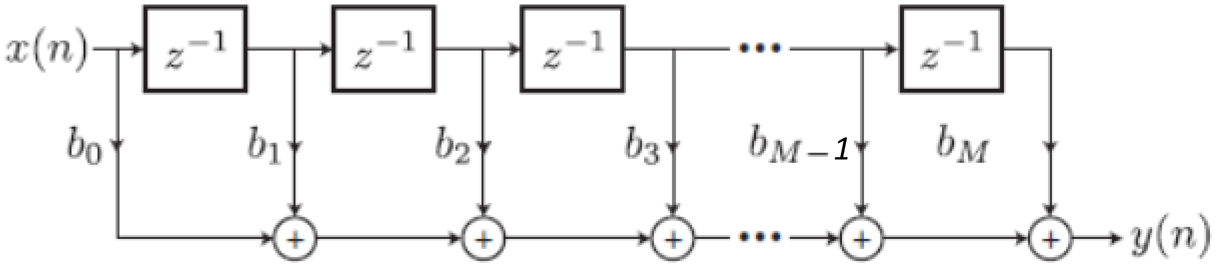
FIR System Realization: General

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

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



- Requires:
  - $M + 1$ multiplications
  - M additions
  - M memory elements

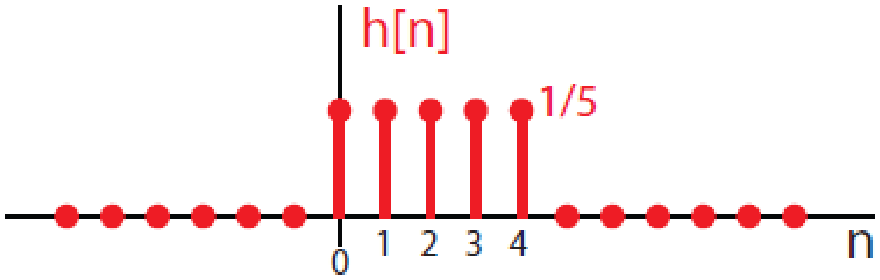
FIR System Realization: Example

- 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:

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

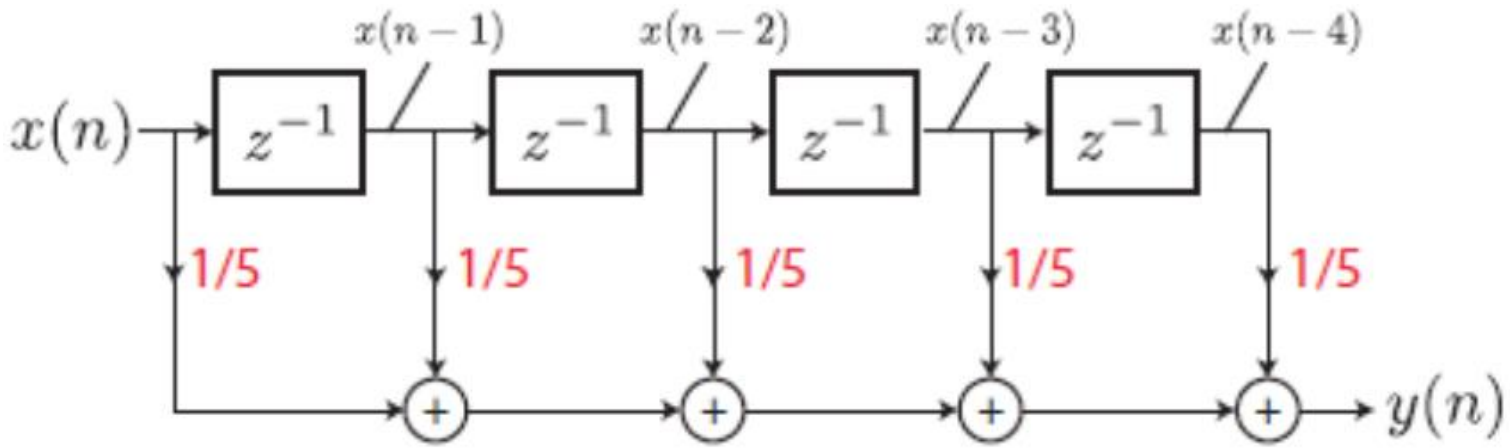


Indeed FIR!

FIR System Realization: Example

$$y(n) = \frac{1}{5}\sum_{k=n-4}^{n} x(k) = \sum_{k=n-4}^{n} \frac{1}{5}x(k)$$

$$\therefore\ y(n) = \frac{1}{5}x(n-4) + \frac{1}{5}x(n-3) + \frac{1}{5}x(n-2) + \cdots$$
$$\cdots \frac{1}{5}x(n-1) + \frac{1}{5}x(n)$$



IIR System Realization

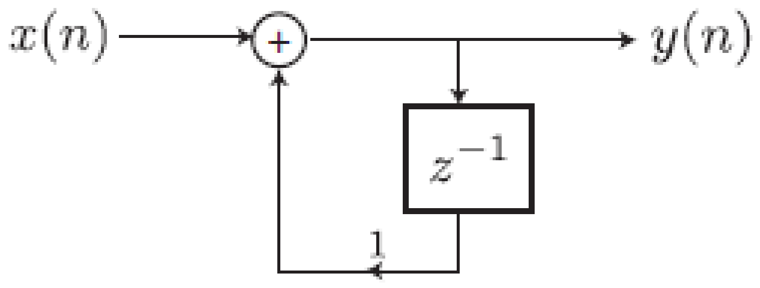
Infinite Impulse Response Systems and Recursive Implementation

IIR System Realization: Example

- Consider an accumulator:

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

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



recursive implementation

Realization: Direct Form I vs Direct Form II

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

is equivalent to the cascade of the following systems:

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

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

Realization: Direct Form I vs Direct Form II

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

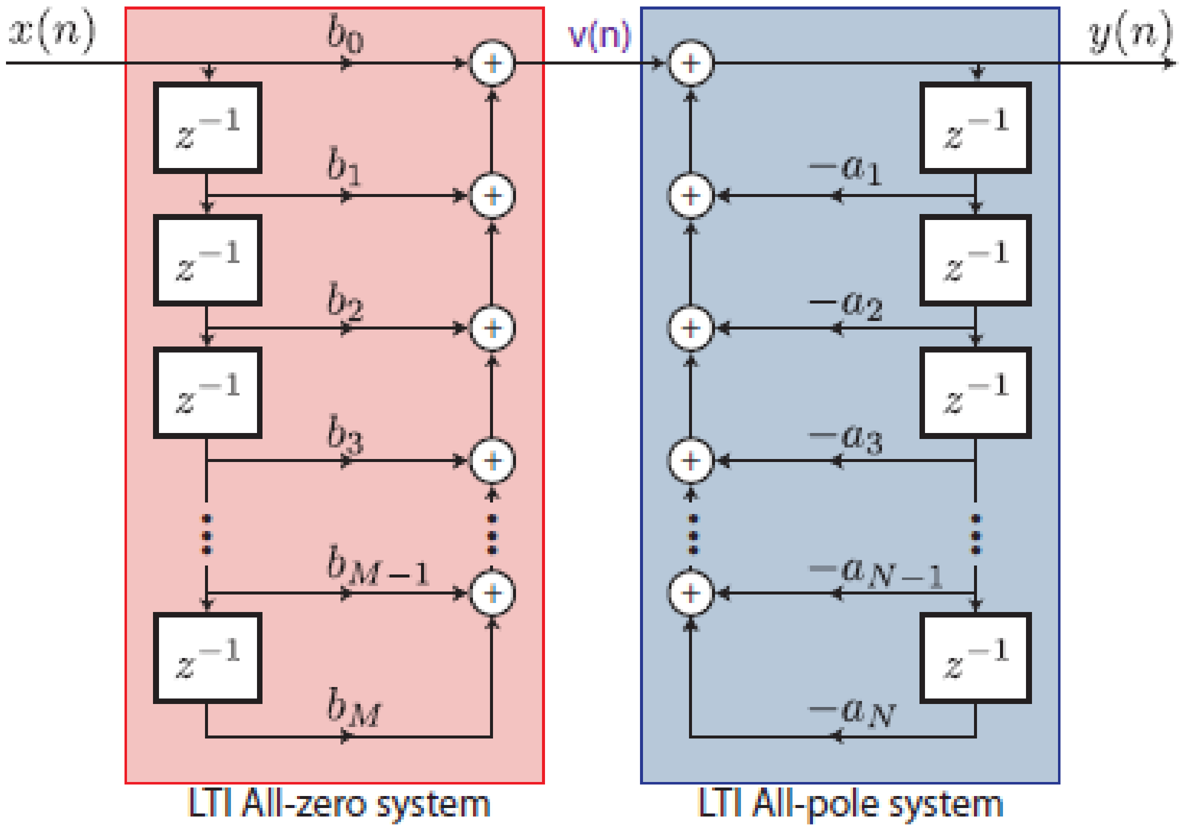
is equivalent to the cascade of the following systems:

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

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

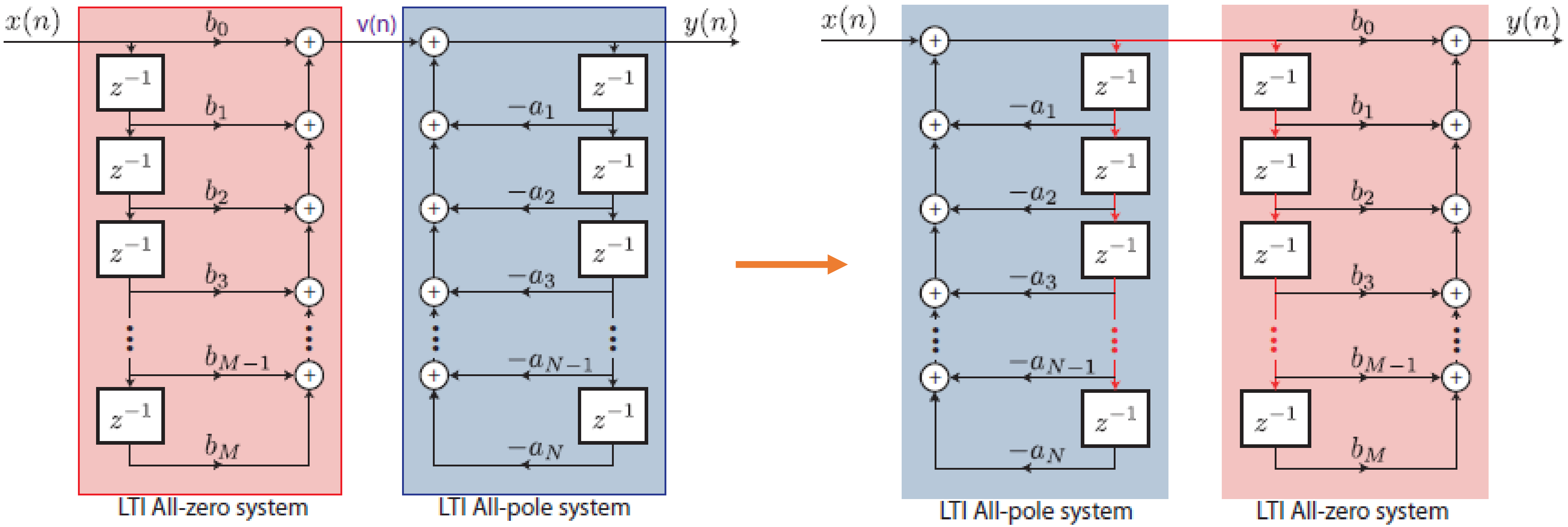
- Requires:
  - $M + N + 1$ multiplications
  - $M + N$ additions
  - $M + N$ memory locations

Direct Form I



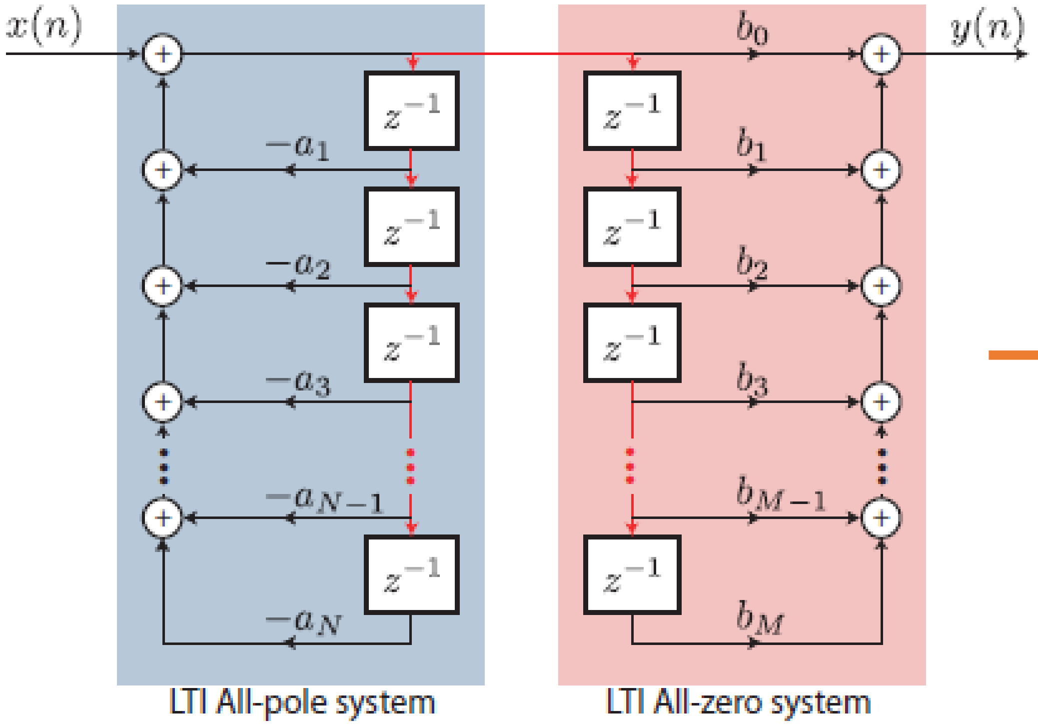
Realization: Direct Form I vs Direct Form II

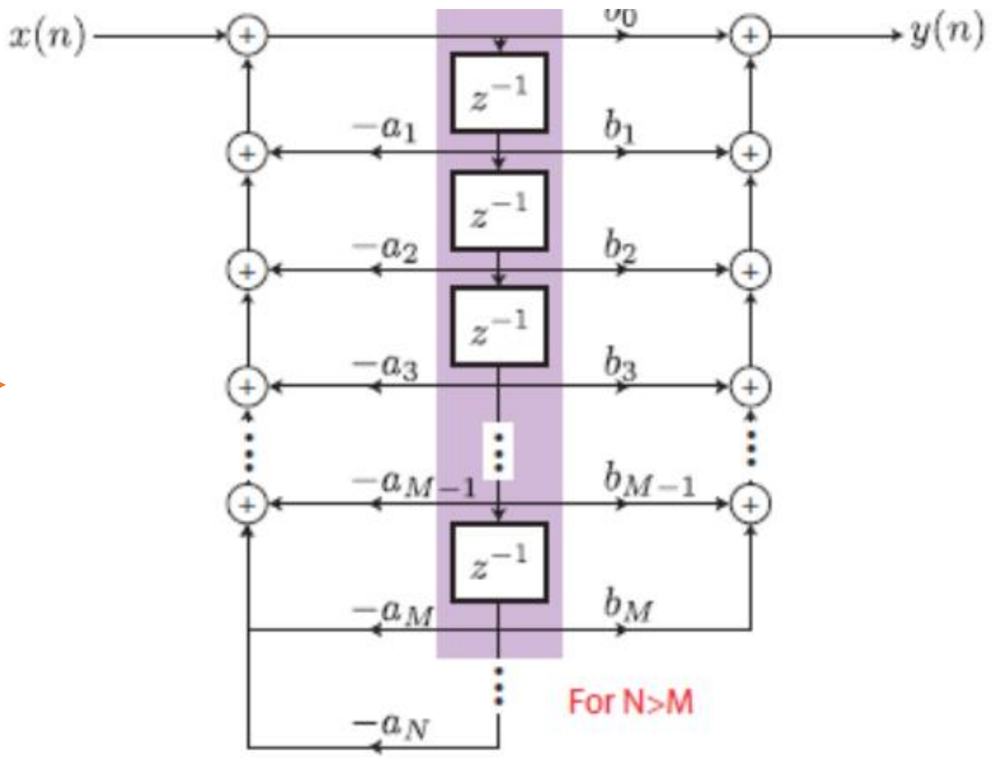
- The order of two cascaded LTI systems can be changed without changing the overall system response.



Realization: Direct Form I vs Direct Form II

- The delay elements can be merged since they have the same inputs and hence the same outputs.





Realization: Direct Form I vs Direct Form II

- General expression for Nth-order LCCDE:

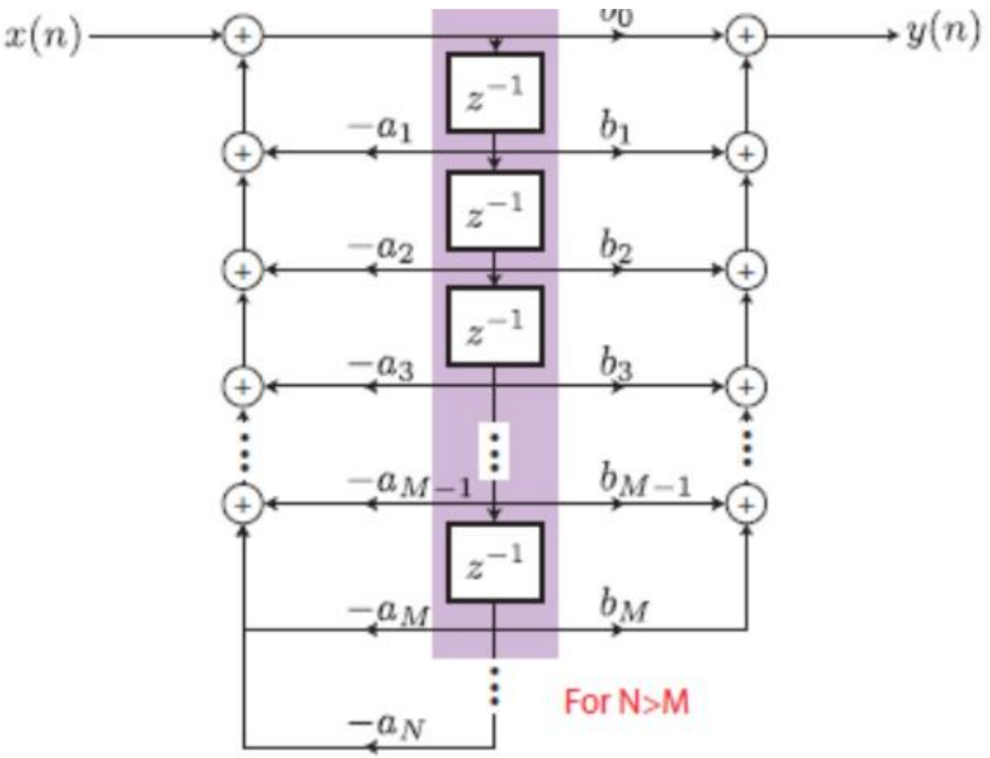
$$\sum_{k=0}^{N} a_k y(n-k) = \sum_{k=0}^{M} b_k x(n-k) \qquad a_0 \triangleq 1$$

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

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

- Requires:
  - $M + N + 1$ multiplications
  - $M + N$ additions
  - $\mathrm{Max}(M, N)$ memory locations

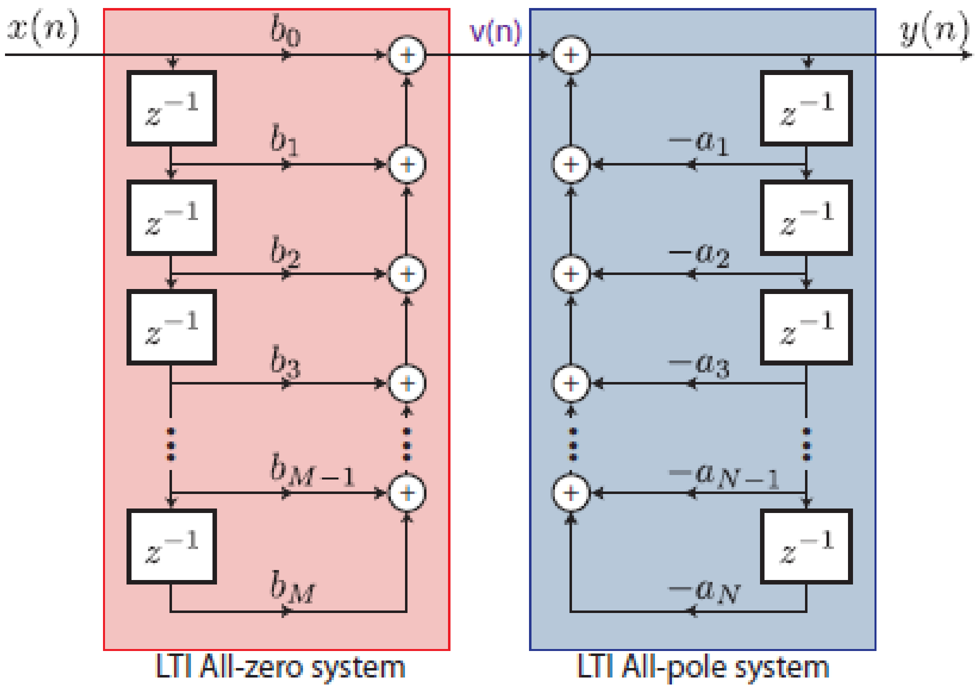
Direct Form II



Comparison of Direct Form I vs Direct Form II

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

Direct Form I



Direct Form II