Difference Equations and Implementation

Professor Deepa Kundur

University of Toronto

Difference Equations and Implementation

Reference:

Section 2.4 and 2.5 of

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

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):

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

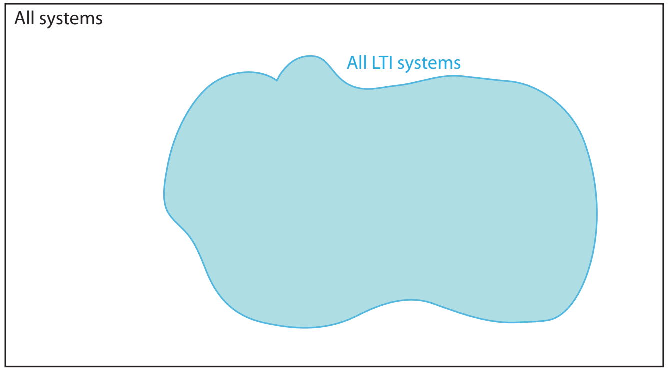
Infinite impulse response (IIR):

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

System Realization

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

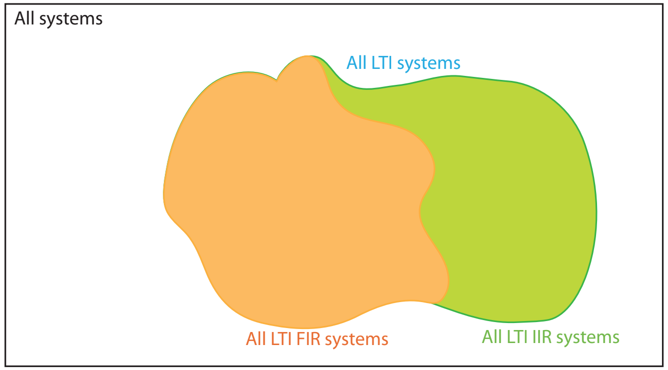
... difference equations.



System Realization

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

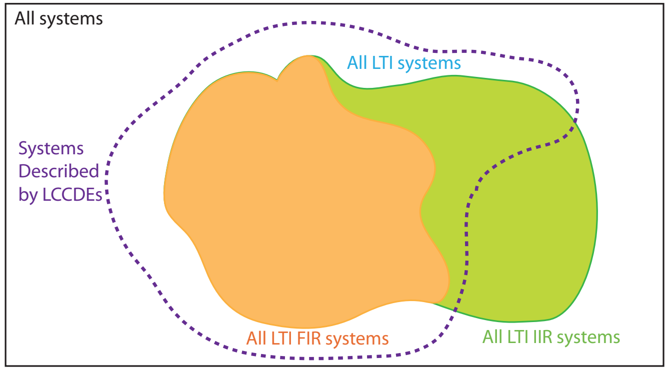
... difference equations.



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

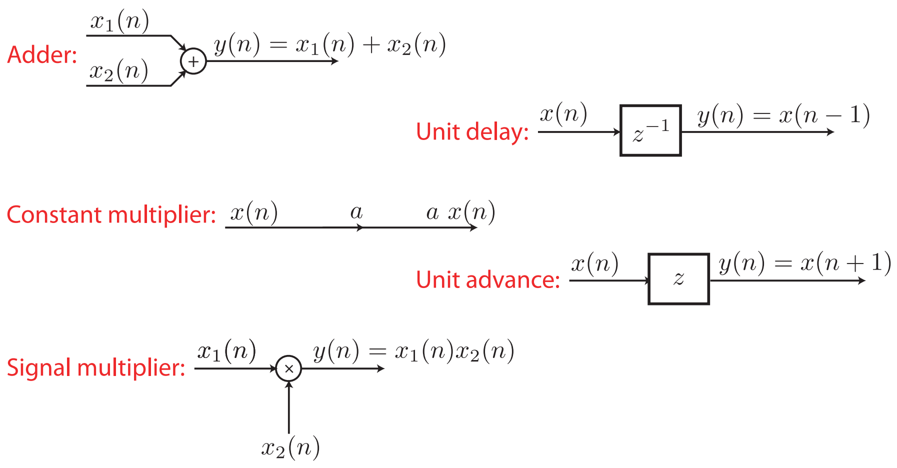
General expression for Nth-order LCCDE:

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

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

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

Building Block Elements



FIR System Realization

Finite Impulse Response Systems and Nonrecursive Implementation

FIR System Realization: Example

- Consider a 5-point local averager:

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

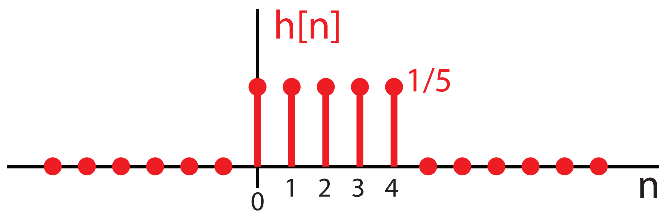
- The impulse response is given by:

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

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

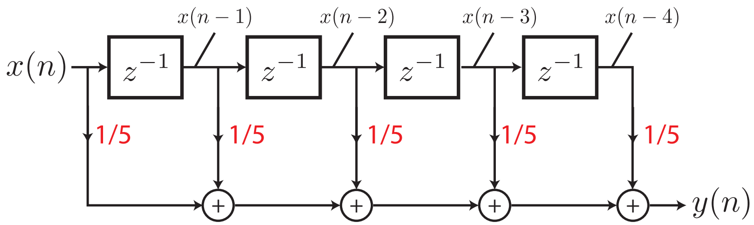
- Consider a 5-point local averager:

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

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

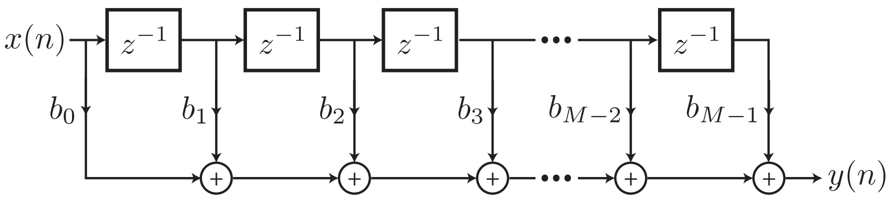
FIR System Realization: Example

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



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

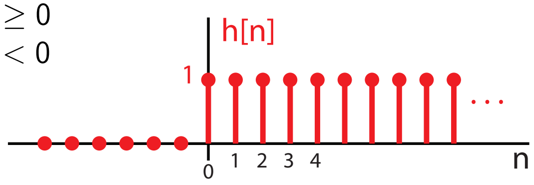
IIR System Realization: Example

- 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

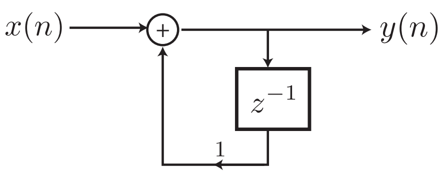
- Consider an accumulator:

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

- 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

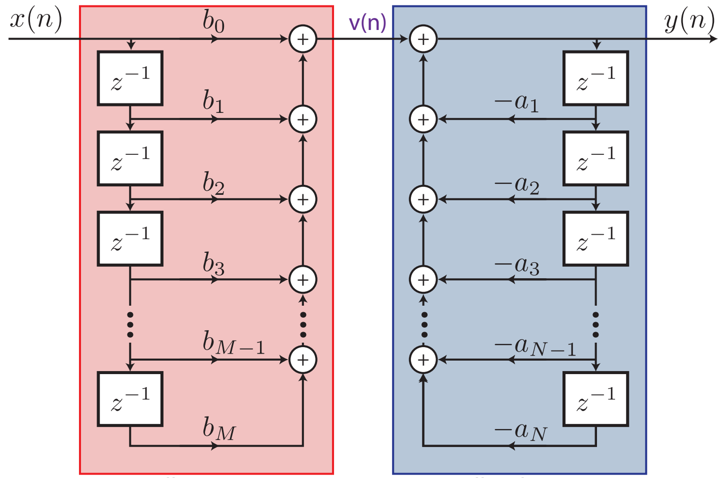
$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k)$$

is equivalent to the cascade of the following systems:

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

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

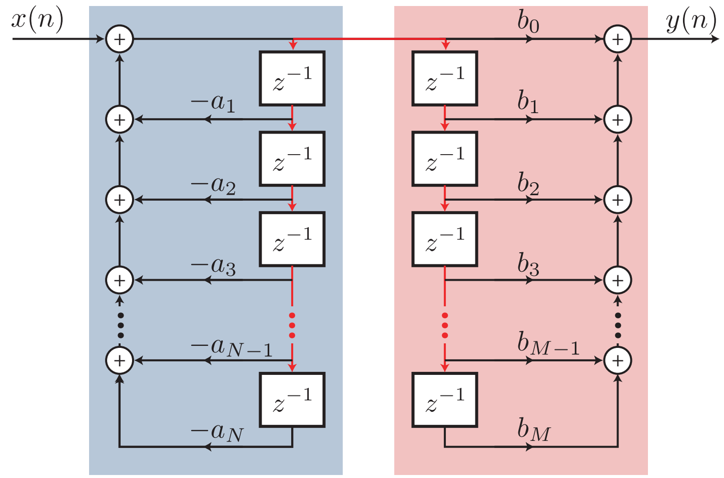
Direct Form I IIR Filter Implementation



LTI All-zero system LTI All-pole system

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

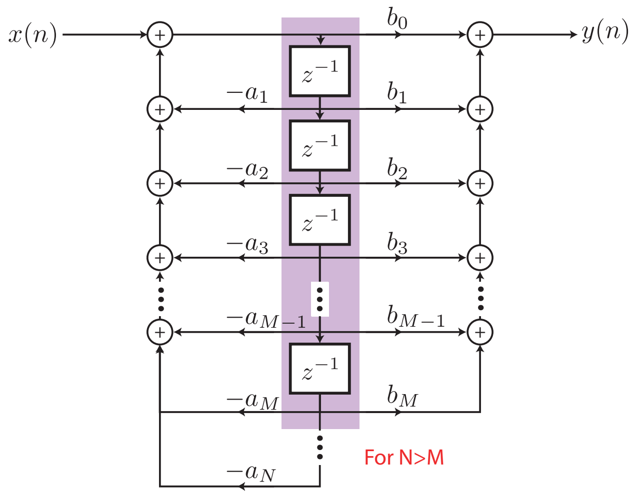
Direct Form II IIR Filter Implementation



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