Digital Signal Processing

Class 7
02/11/2025

ENGR 71

- Class Overview
  - Discrete-Time Signals and Systems
- Assignments
  - Reading:

    Chapter 2: Discrete-Time Signals and Systems

    Begin reading Chapter 3
  - Lab 1 – Aliasing lab
    - Will be up on Moodle this afternoon

ENGR 71

- Lab 1-Aliasing Lab
  - Find a short piece of music to download
  - Subsample to demonstrate aliasing
  - Repeat the subsampling, but prior to subsampling, apply a low-pass anti-aliasing filter.
  - Compare the results
  - Mystery piece
- More details and sample code will be placed on Moodle: Lab 1

ENGR 71

- Homework 3
  - Problems: 2.9 (a), 2.17(a), 2.28(a & c),
    2.35, 2.46,
    C2.14(write your own code)
    C2.8 (use Matlab functions)

    Due Feb. 20

Class Information

- Topics in Discrete-Time Signals and Systems
  - Discrete-Time Signals
  - Discrete-Time Systems
  - Analysis of Linear Time-Invariant Systems
  - Description of Systems by Difference Equations
  - Implementation of Discrete-Time Systems
  - Correlation of Discrete-Time Systems

FIR and IIR System

- Finite Impulse Response systems (FIR)
  - For causal FIR system: impulse response lasts a finite number of steps $h(n) = 0, \quad n < 0 \text{ and } n \geq M$

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

    - Output is weighted sum of input values

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

    - Finite memory length
    - Not recursive

FIR and IIR System

- Infinite Impulse Response systems (IIR)
  - For causal system, $h(n)$ persists for n going to infinity

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

  - Infinite memory. Response depends on all previous inputs
  - Usually results from system with recursion where current output depends on previous outputs.

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

  - Could have something like moving average that goes on for ∞
    - But real physical systems are not infinite

Difference Equations

– LTI systems characterized by constant-coefficient difference equations

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

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

N'th order system corresponds to

N'th order difference equation

Difference Equations

– Example on Moodle page

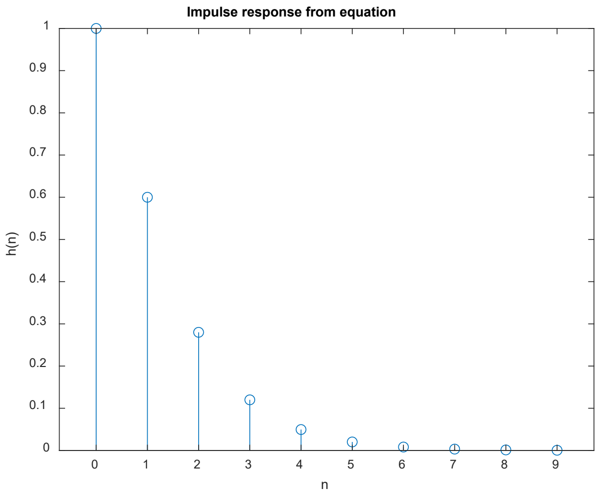
$$\lambda^n - 0.6\lambda^{n-1} + 0.08\lambda^{n-2} = 0$$

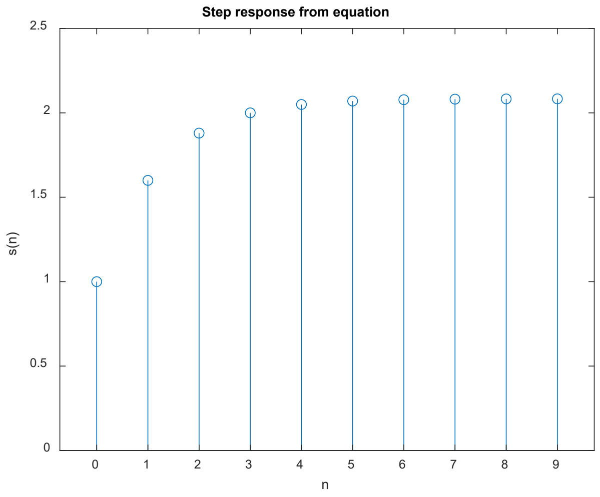
Impulse response: $h(n) = -\left(\frac{1}{5}\right)^n + 2\left(\frac{2}{5}\right)^n$

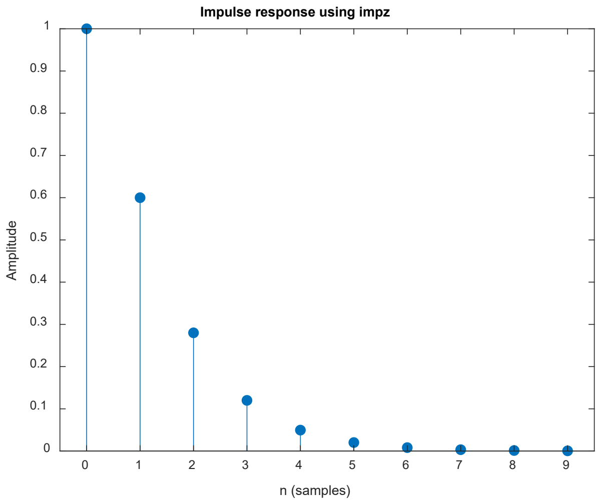
Step response: $s(n) = \sum_{k=1}^{n} h(n-k) = \frac{1}{12}\left[25 + \frac{1}{5^n}\left(3 - 2^{n+4}\right)\right]$

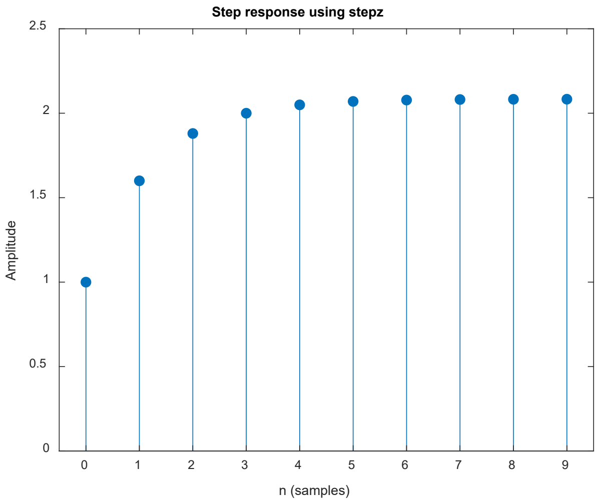
Also, Matlab commands using *b*'s and *a*'s from difference equation:
impz(b,a) → impz(1,[1, -0.6, 0.08]) for this system
stepz(b,a) → stepz(1,[1, -0.6, 0.08])

Difference Equations









System Implementation

- Implementation of Discrete-Time System
  - We will discuss two forms
    - Direct form I
    - Direct form II

System Implementation

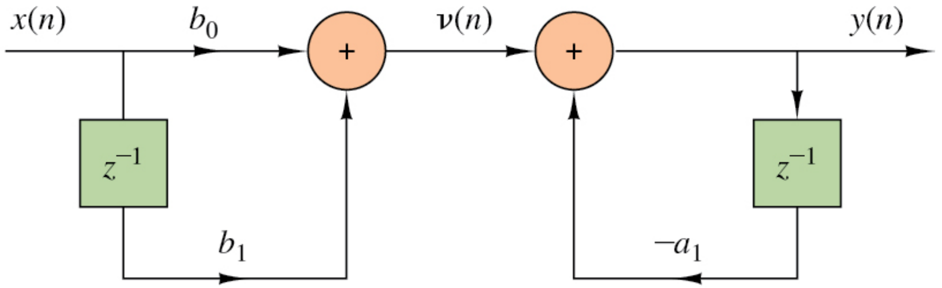
– Direct form I

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

- Uses separate delays for both input and output
- Cascade of two LTI systems

$$v(n) = \sum_{k=0}^{M} b_k x_k(n-k) \quad , \quad y(n) = v(n) - \sum_{k=1}^{N} a_k y_k(n-k)$$

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



Easy to get from difference equation

$$v(n) = b_0 x(n) + b_1 x(n-1)$$

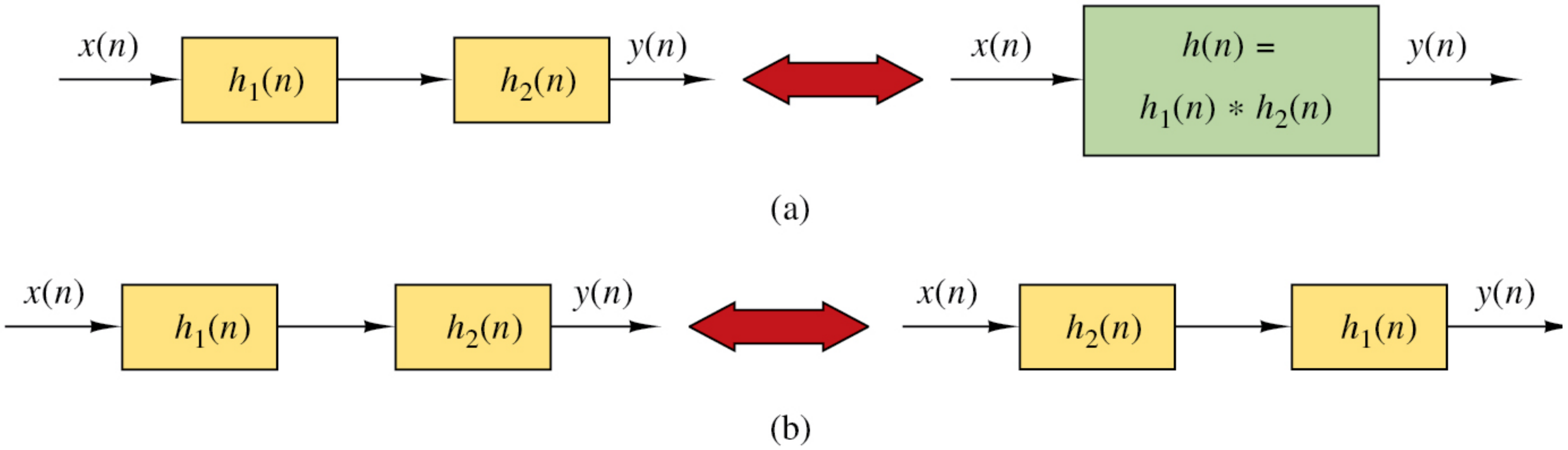
$$y(n) = v(n) - a_1 y(n-1)$$

System Implementation

- Direct form II

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

- The order of cascaded LTI systems doesn't matter because of the commutative property of convolution
- Cascade of two LTI systems $h_1(n) * h_2(n) = h_2(n) * h_1(n)$



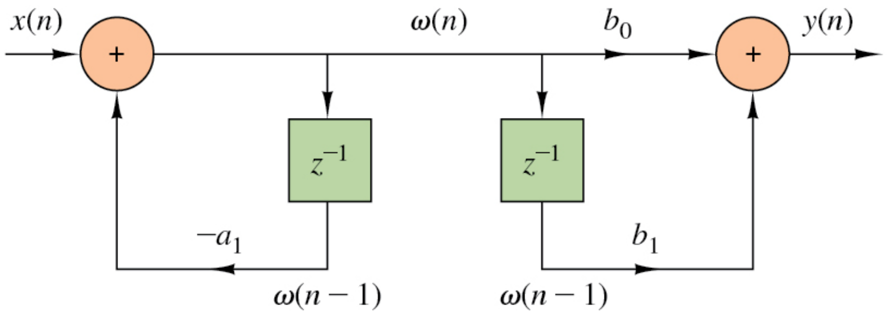
System Implementation

– Direct form II

- Input-output relationship is same for interchanged systems
- Cascaded interchanged systems:

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

$$\text{Example: } y(n) = -a_1 y(n-1) + b_0 x(n) + b_1 (n-1) \quad \left(N = 1, M = 1\right)$$

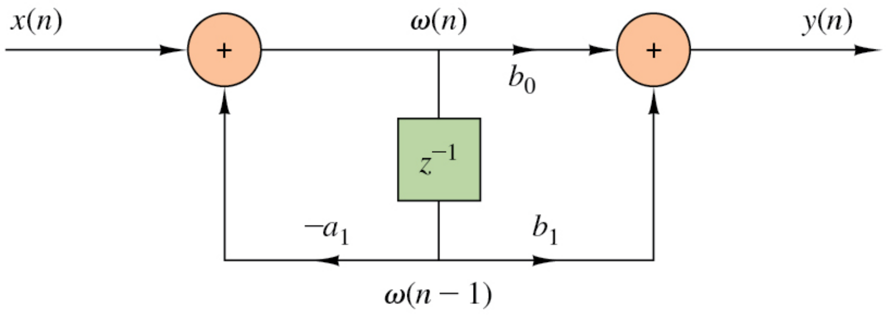


$$w(n) = x(n) - a_1 w(n-1)$$

$$y(n) = b_0 w(n) + b_1 w(n-1)$$

System Implementation

- Direct form II
  - Notice that $w(n)$ is shifted in both subsystems
  - Just use one shifter:

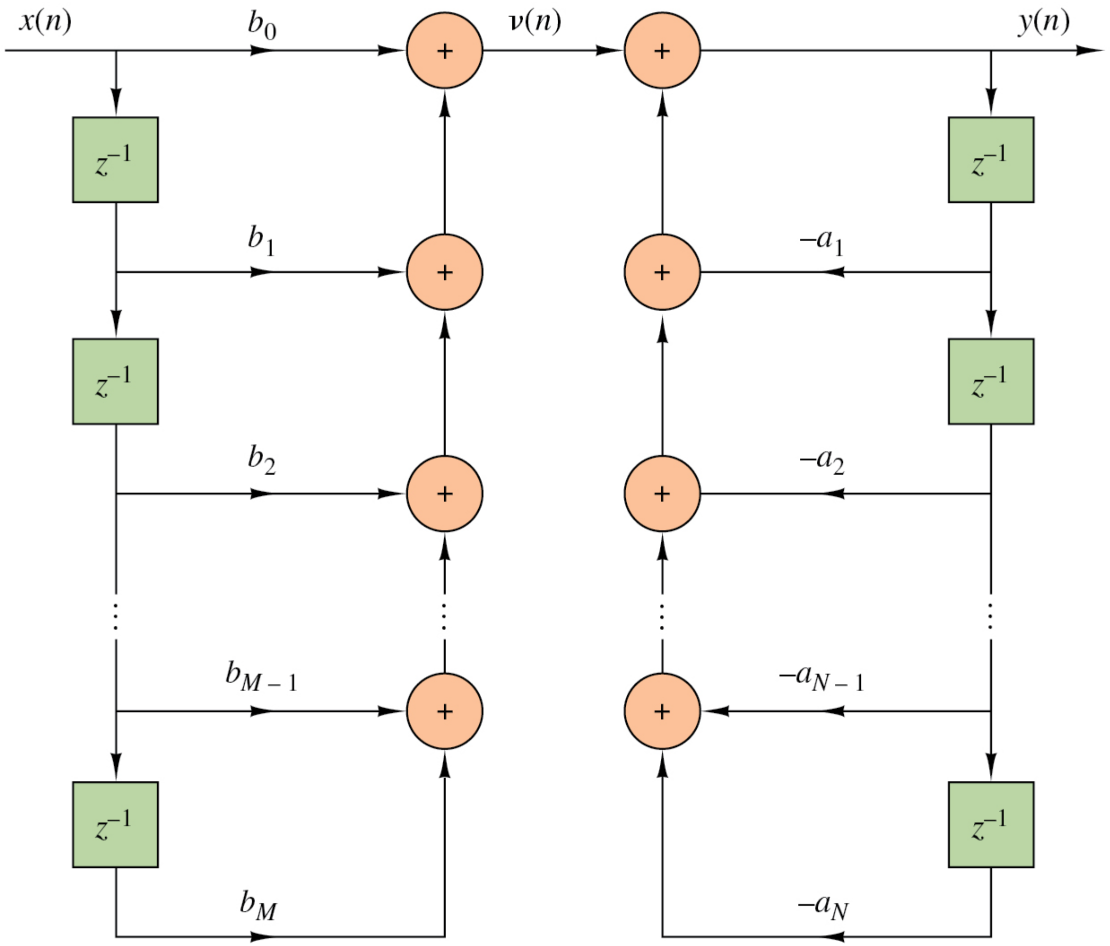


$$w(n) = x(n) - a_1 w(n-1)$$

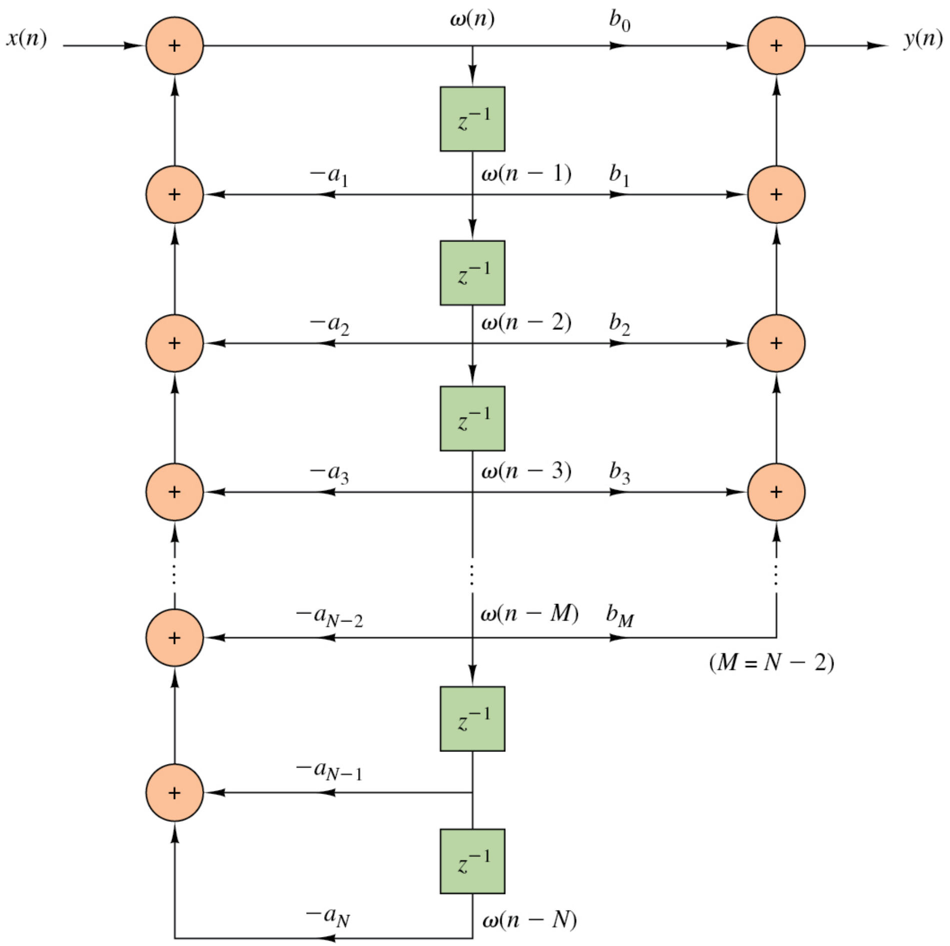
$$y(n) = b_0 w(n) + b_1 w(n-1)$$

System Implementation

Direct form I



Direct form II



System Implementation

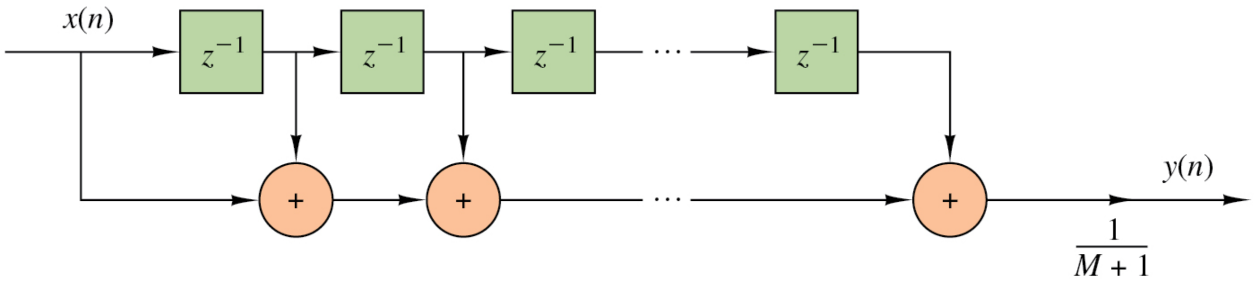
- Number of operations:
  - Direct form I:
    - Number of Multiplications: $N+M+1$ (example: 1+1+1 =3)
    - Number of delays: $M+N$ (example: 1+1 = 2)
  - Direct form II
    - Number of Multiplications: $N+M+1$ (example: 1+1+1 =3)
    - Number of delays: $M+N-1$ (example: 1+1 = 2)

System Implementation

– If N=0: Weighted Moving Average Filter

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

- Nonrecursive, FIR $h(n) = \left[b_0, b_1, \ldots, b_M\right]$
- Weighted moving average of inputs
- If just averaging, weights are all 1/(M+1)



System Implementation

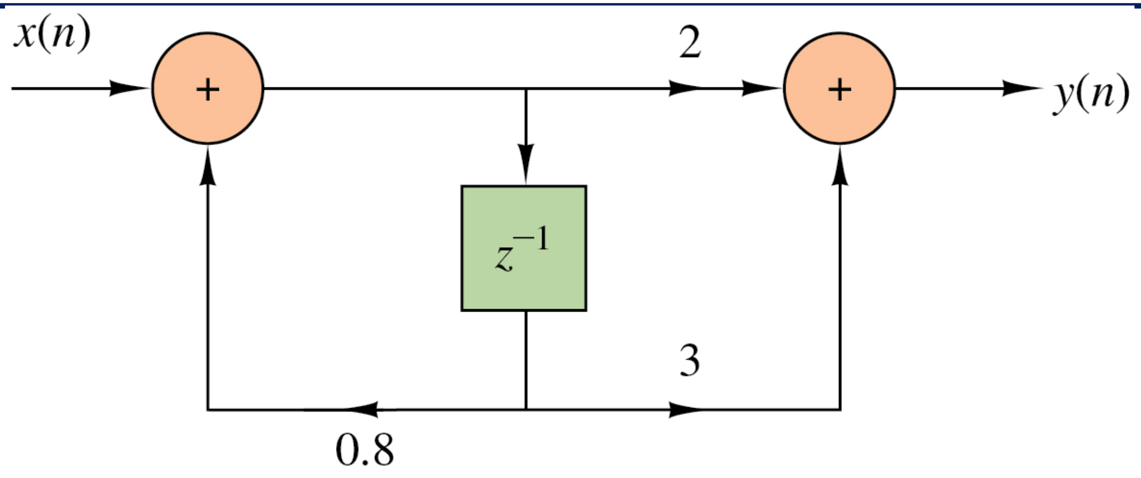
- Advantages of direct form I
  - Simplicity
  - Better stability in finite precision arithmetic
  - Lower round-off error for coefficient quantization
- Disadvantages of direct form 1
  - Twice as many delays (memory requirement is higher)
  - More sensitive to overflow in fixed-point arithmetic
  - Increased computational load for high order filters
    - Due to twice as many delays

System Implementation

- Advantages of direct form II
  - Half as many delays
    - Half the memory required for direct form 1
  - Less sensitive to overflow for fixed point arithmetic
  - Faster execution in some DSP processors
    - Optimized for direct form 2
  - More common in real-world implementations
    - Audio processing, communication systems, embedded processors
- Disadvantages of direct form 2
  - Less intuitive
  - Instability in finite precision arithmetic
  - Potential for higher round-off error in floating point implementations
- **Bottom Line: Use Direct form II**

System Implementation

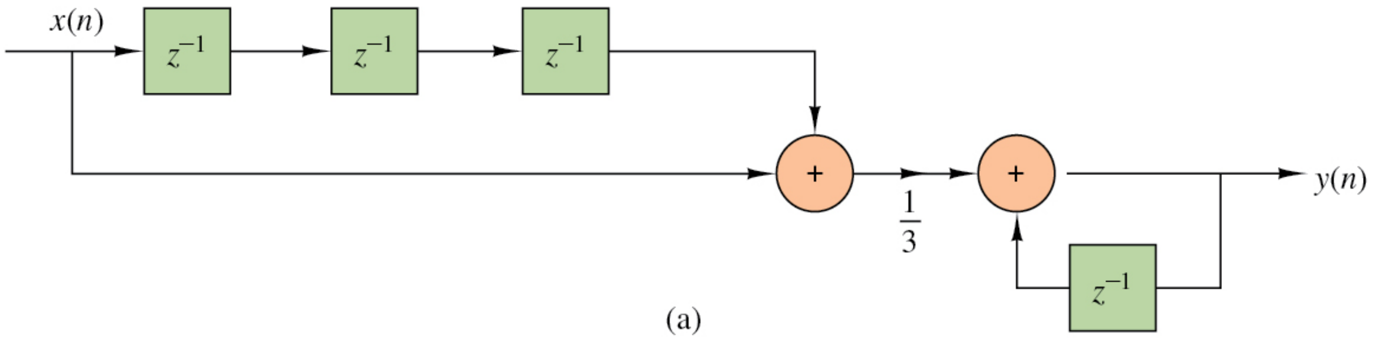
- Example:



- Convert to direct form 1
- Write difference equation
- Find impulse response

System Implementation

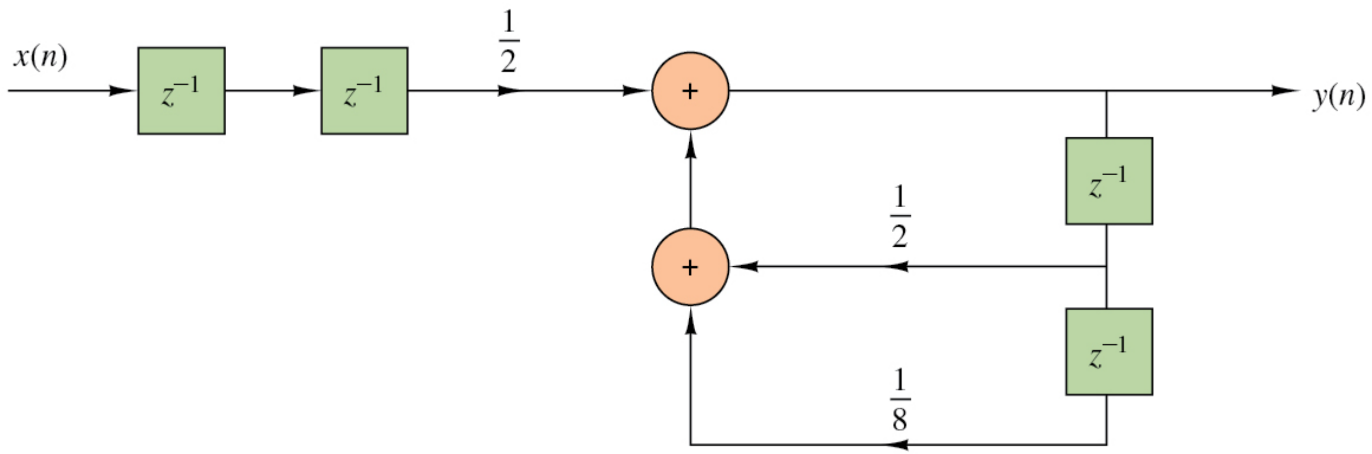
– Example:



(a)

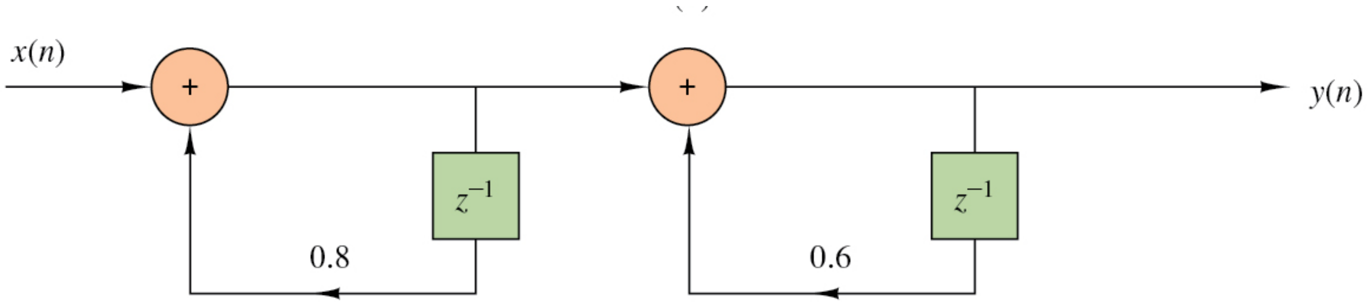
System Implementation

– Example:



System Implementation

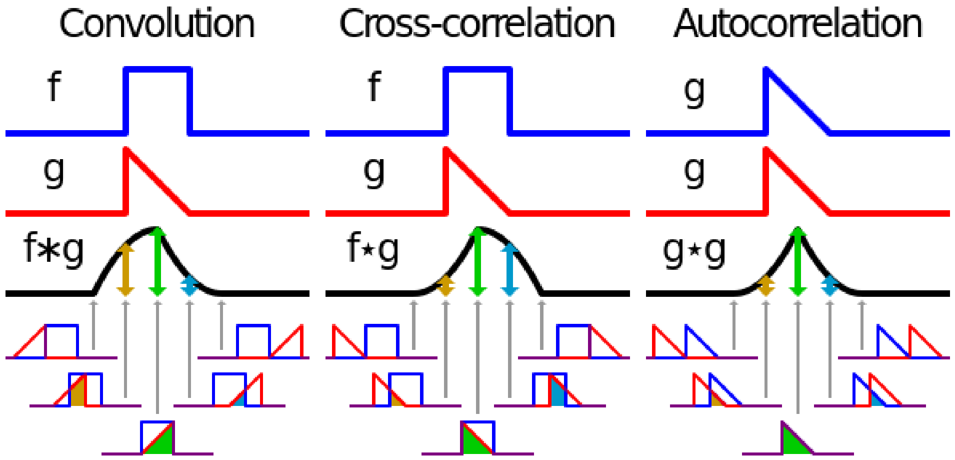
– Example:



(c)

Correlation

- A way to see how similar signals are
  - Correlation: $r_{xy}(n) = \sum_{k=-\infty}^{\infty} x(k)y(k-n)$
  - Compare this to convolution: $x(n) * y(n) = \sum_{k=-\infty}^{k} x(k)y(n-k)$



https://lpsa.swarthmore.edu/Convolution/CI.html

Correlation

- A way to see how similar signals are
  - Notation can be confusing.
    - Correlation looks at similarity of two signals as a function of lag (l), so book uses notation

$$r_{xy}(l) = \sum_{n=-\infty}^{\infty} x(n)y(n-l) \quad \text{where } l \text{ is the lag}$$

Moving average filters