Department of Electrical Engineering
University of Arkansas

ELEG 5173L Digital Signal Processing
Ch. 5 Digital Filters

Dr. Jingxian Wu

wuj@uark.edu

OUTLINE

- FIR and IIR Filters
- Filter Structures
- Analog Filters
- FIR Filter Design
- IIR Filter Design

FIR V.S. IIR

- **LTI discrete-time system**
  - Difference equation in time domain

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

  - Transfer function in z-domain

$$Y(z) = -\sum_{k=1}^{N} a_k Y(z) z^{-k} + \sum_{k=0}^{M} b_k X(z) z^{-k}$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

FIR V.S. IIR

- **Finite impulse response (FIR)**
  - difference equation in the time domain

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

  - Transfer function in the Z-domain

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{k=0}^{M} b_k z^{-k}$$

  - Impulse response

$$h(n) = [b_0, b_1, \cdots, b_M]$$

    - The impulse response is of finite length → finite impulse response

FIR V.S. IIR

- **Infinite impulse response (IIR)**
  - Difference equation in the time domain

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

  - Transfer function in the z-domain

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

  - Impulse response can be obtained through inverse-z transform, and it has infinite length

UNIVERSITY OF ARKANSAS

FIR V.S. IIR

- **Example**
  - Find the impulse response of the following system. Is it a FIR or IIR filter? Is it stable?

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

FIR V.S. IIR

- **Example**
  - Find the impulse response of the following system. Is it a FIR or IIR filter? Is it stable?

$$y(n) = x(n) - 2x(n-2) + 5x(n-3)$$

OUTLINE

- FIR and IIR Filters
- **Filter Structures**
- Analog Filters
- FIR Filter Design
- IIR Filter Design

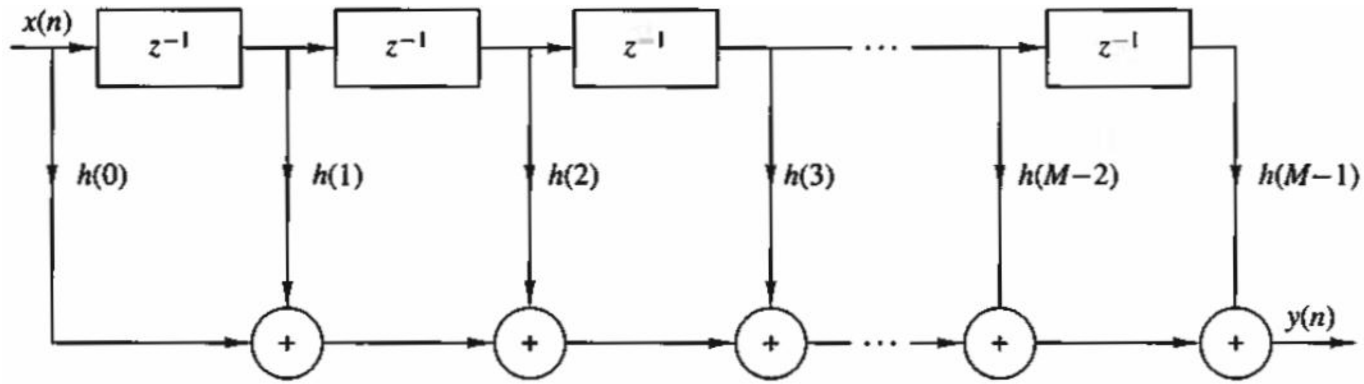
FILTER STRUCTURE: FIR

- **A FIR filter can be implemented in different structures**
  - Direct-form structure
  - Cascade-form structure
  - Lattice structure
  - Frequency sampling
  - ……

FILTER STRUCTURE: FIR

- **FIR: Direct-form**
  - Also called tapped-delay-line filter, or transversal filter

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



Order of the filter: *M*

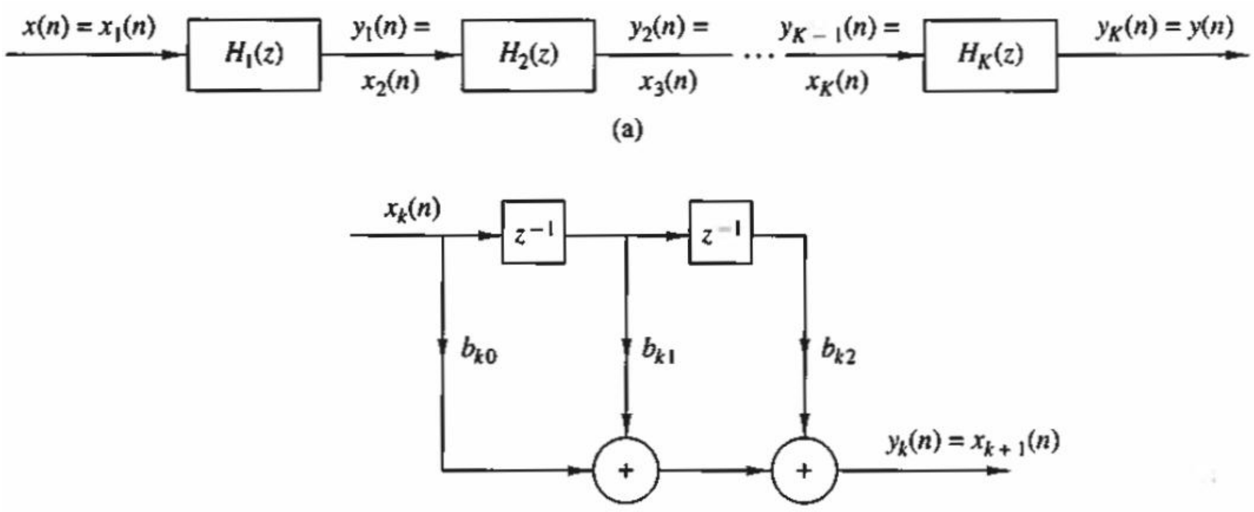
FILTER STRUCTURE: FIR

- **FIR: Cascade-form**
  - Factor the transfer function into the product of second-order FIR systems

$$H(z) = H_1(z)H_2(z)\cdots H_K(z)$$

- Where

$$H_k(z) = b_{k0} + b_{k1}z^{-1} + b_{k2}z^{-2}$$



FILTER STRUCTURE: FIR

- **Example**
  - Draw the direct-form and cascade-form structures of the following filter

$$H(z) = (1 - 2z^{-1} + 3z^{-2})(2 + 5z^{-2})$$

FILTER STRUCTURE: IIR

- **An IIR filter can be implemented in various structures**
  - Direct-form I
  - Direct-form II
  - Transposed structure
  - Cascade-form structure
  - Parallel-form structure
  - Lattice structure
  - ……

FILTER STRUCTURE: IIR

- **IIR: Direct-form I**

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}} = H_1(z) H_2(z)$$

$$H_1(z) = \sum_{k=0}^{M} b_k z^{-k}$$

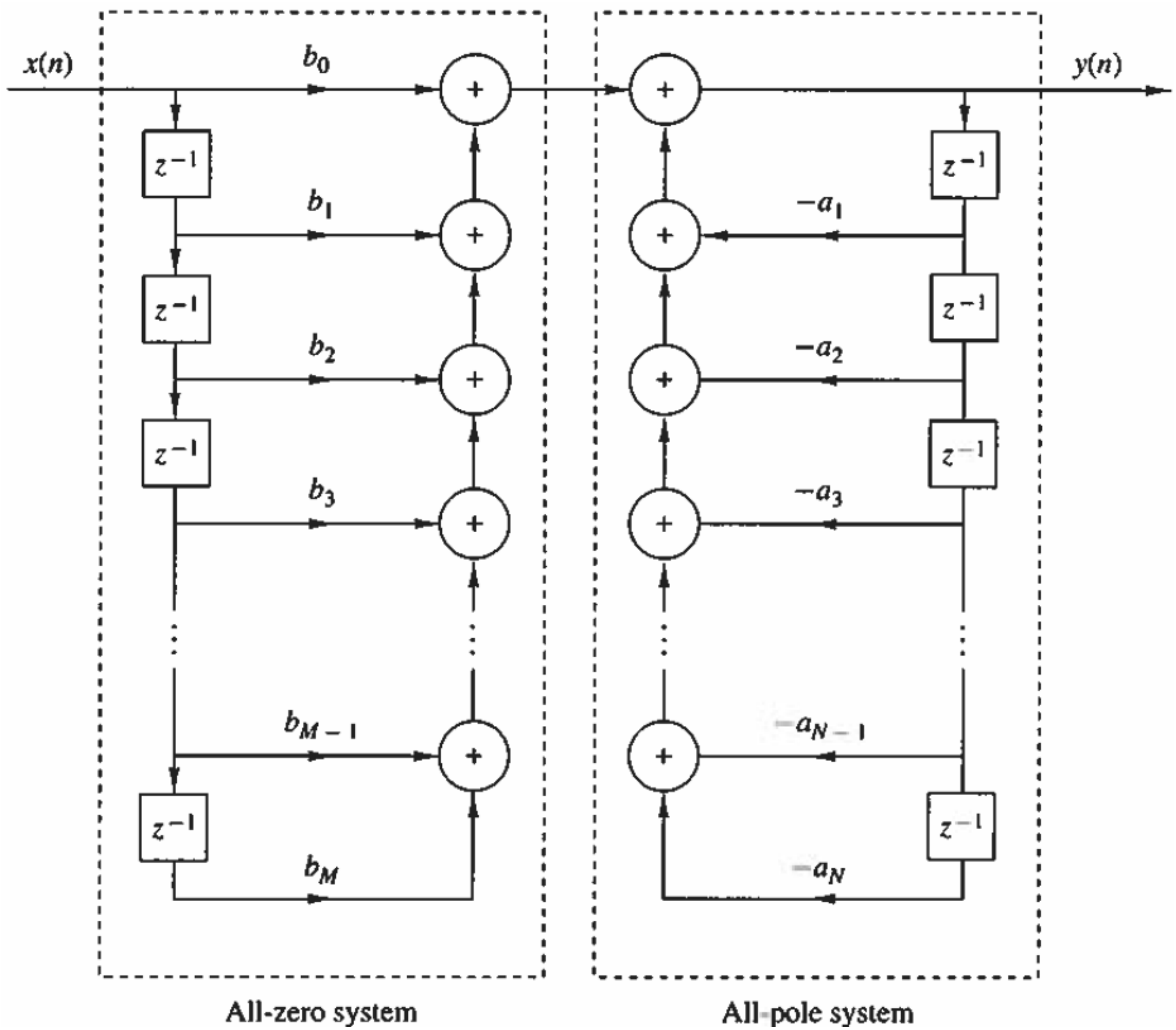
$$H_2(z) = \frac{1}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

FILTER STRUCTURE: IIR

- **IIR: Direct-form I**

$$H_1(z) = \sum_{k=0}^{M} b_k z^{-k}$$

$$H_2(z) = \frac{1}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$



FILTER STRUCTURE: IIR

- **IIR: Direct-form II**
  - Put $H_2(z)$ before $H_1(z)$

$$H(z) = H_2(z)H_1(z)$$

- Difference equation of $H_2(z) = \frac{1}{1+\sum_{k=1}^{N} a_k z^{-k}}$

$$w(n) = -\sum_{k=1}^{N} a_k w(n-k) + x(n)$$

- Difference equation of $H_1(z) = \sum_{k=0}^{M} b_k z^{-k}$

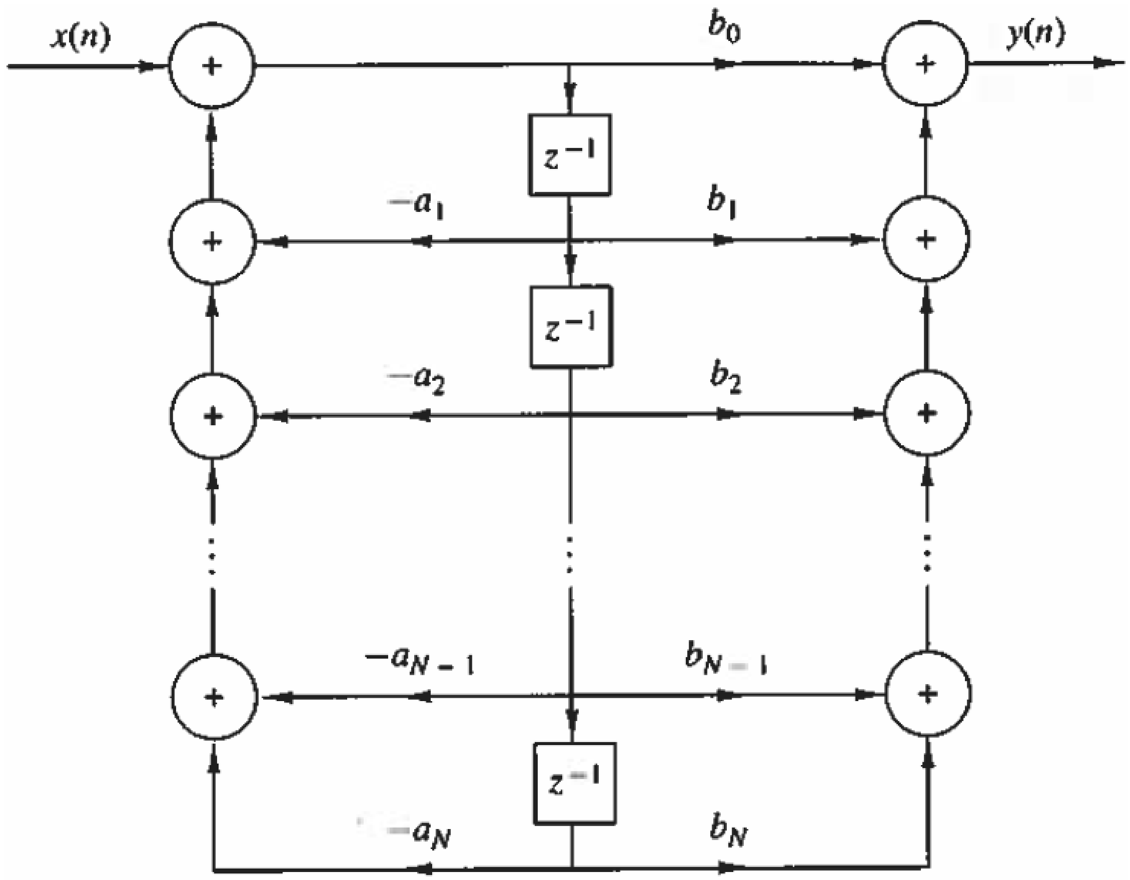
$$y(n) = \sum_{k=0}^{M} b_k w(n-k)$$

- Both involve the delay of w(n)

UNIVERSITY OF ARKANSAS

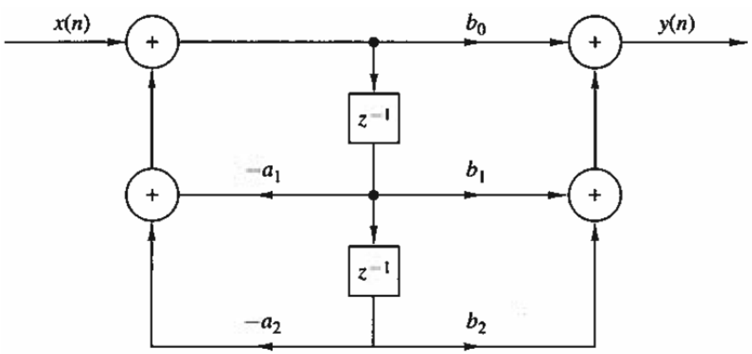
FILTER STRUCTURE: IIR

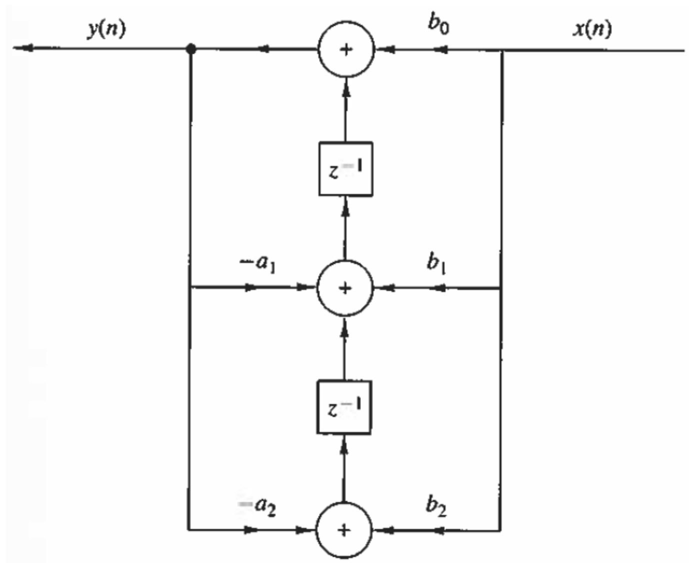
- **IIR: Direct-form II**



FILTER STRUCTURE: IIR

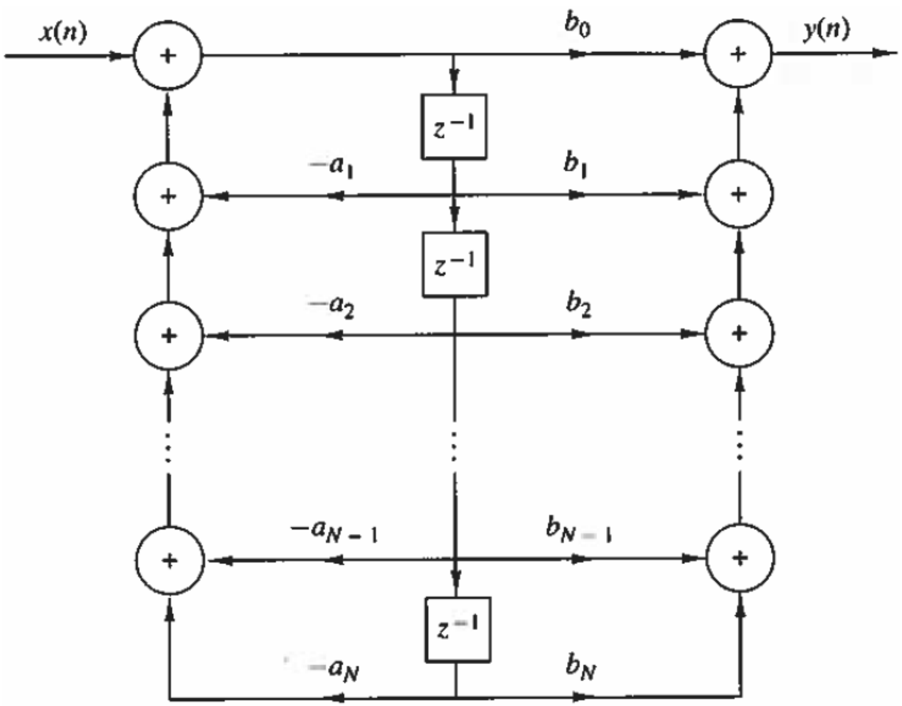
- **IIR: Transposed structure**
  - Transposition theorem: For a given filter structure, if we apply the following transposition: (1) reverse the direction of all signal flows; (2) replace all the adders with intersections, and vice versa; and (3) interchange the input and output, then the transposed system is the same as the original system.

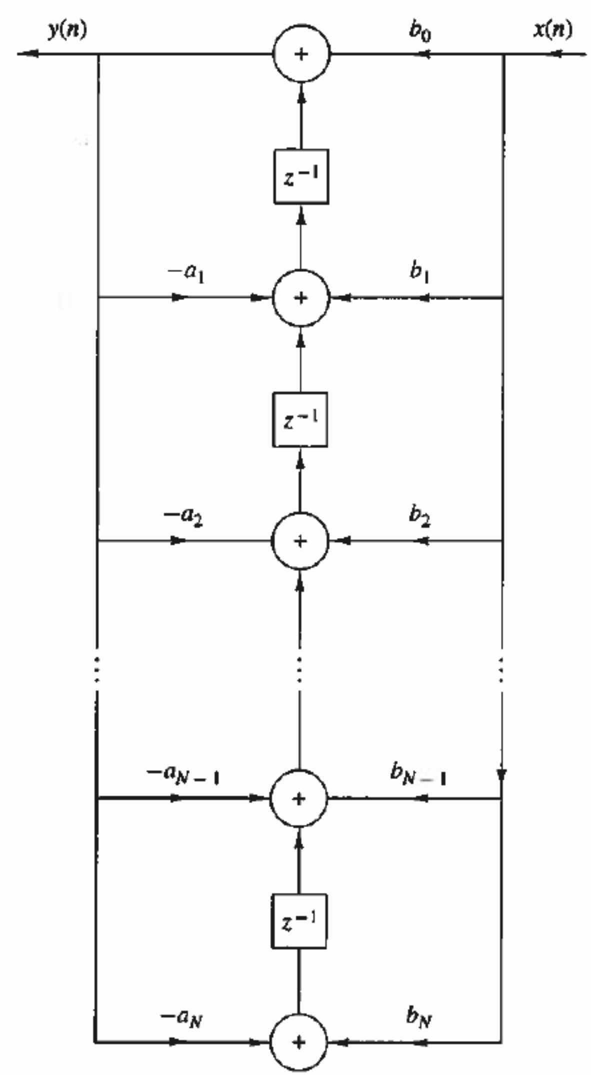




FILTER STRUCTURE: IIR

- **IIR: transposed structure of direct-form II**





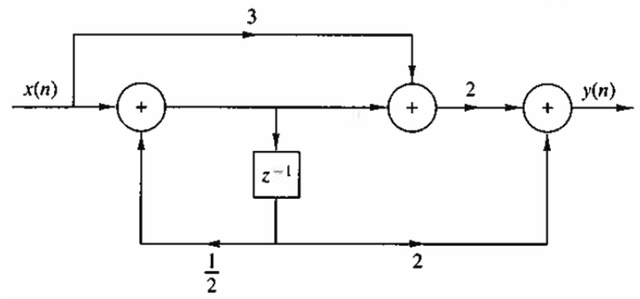
FILTER STRUCTURE

- **Example**
  - Draw the direct-form I, direct-form II, and transposed direct-form II of the filter

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

FILTER STRUCTURE

- **Example**
  - Find the difference equation and transfer function of the filter
  - Represent the filter in direct-form II

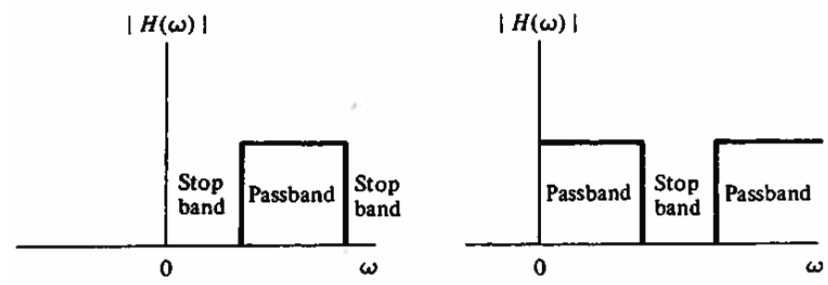


OUTLINE

- FIR and IIR Filters
- Filter Structures
- **Analog Filters**
- FIR Filter Design
- IIR Filter Design

ANALOG FILTERS

- **Ideal filter**
  - Ideal analog filter



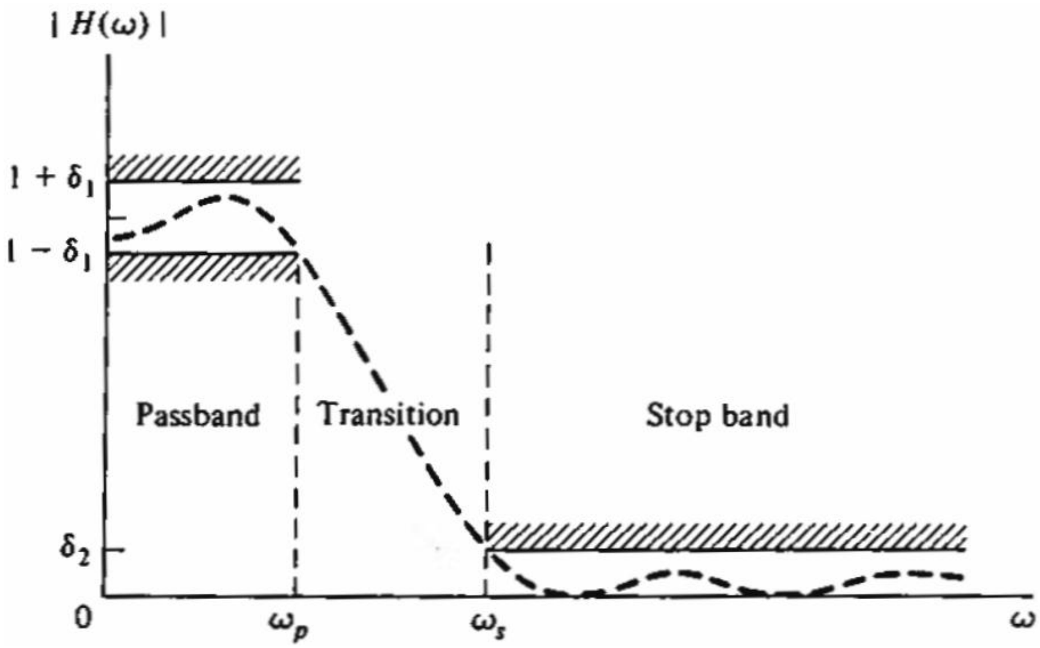
  - Ideal digital filter
  - In practice, we cannot implement filter characteristics with abrupt transitions
    - E.g. the impulse response of ideal low pass filter is sinc(t), which is non-causal.

ANALOG FILTERS

- **Practical low pass filter**
  - Passband cutoff frequency: ω_p
  - Stop band cutoff frequency: ω_s

$$1-\delta_1 \le |H(\omega)| \le 1+\delta_1, |\omega| \le \omega_p$$

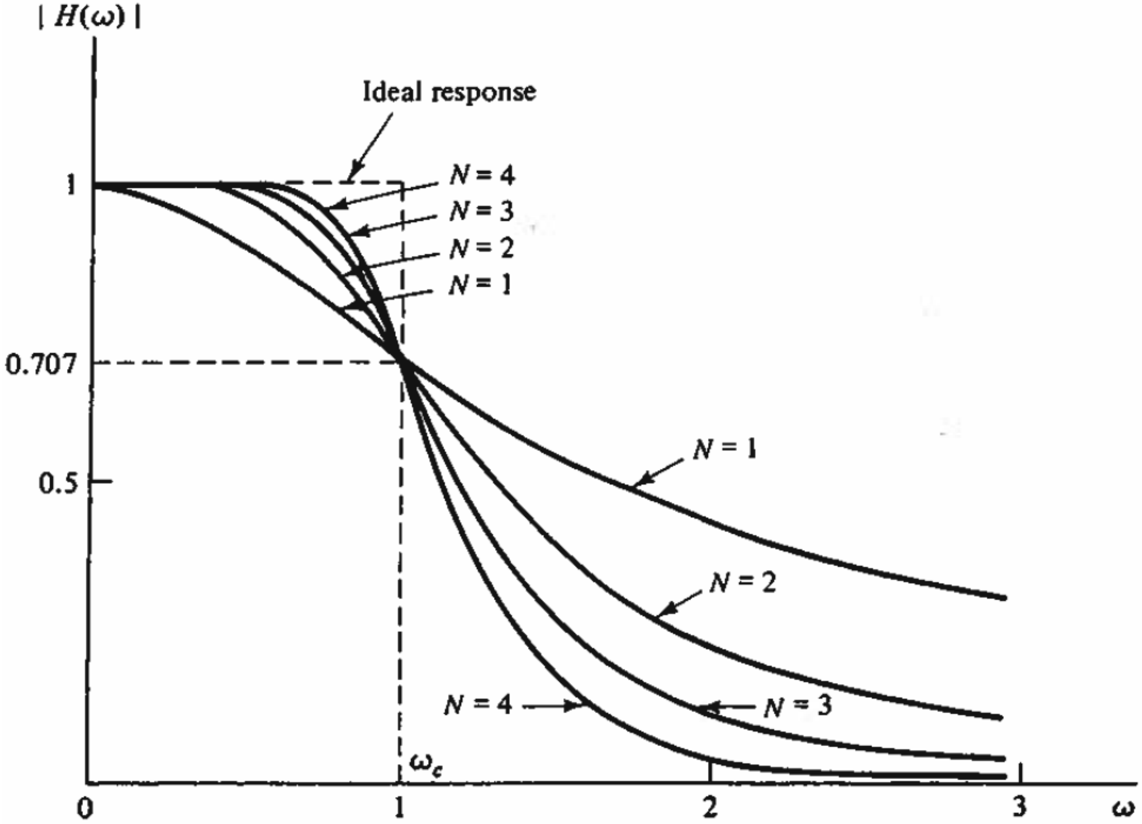
$$|H(\omega)| \le \delta_2, |\omega| \ge \omega_s$$



ANALOG FILTERS

- **Butterworth filter**

$$|H(\omega)|^2 = \frac{1}{1+\omega^{2N}}$$

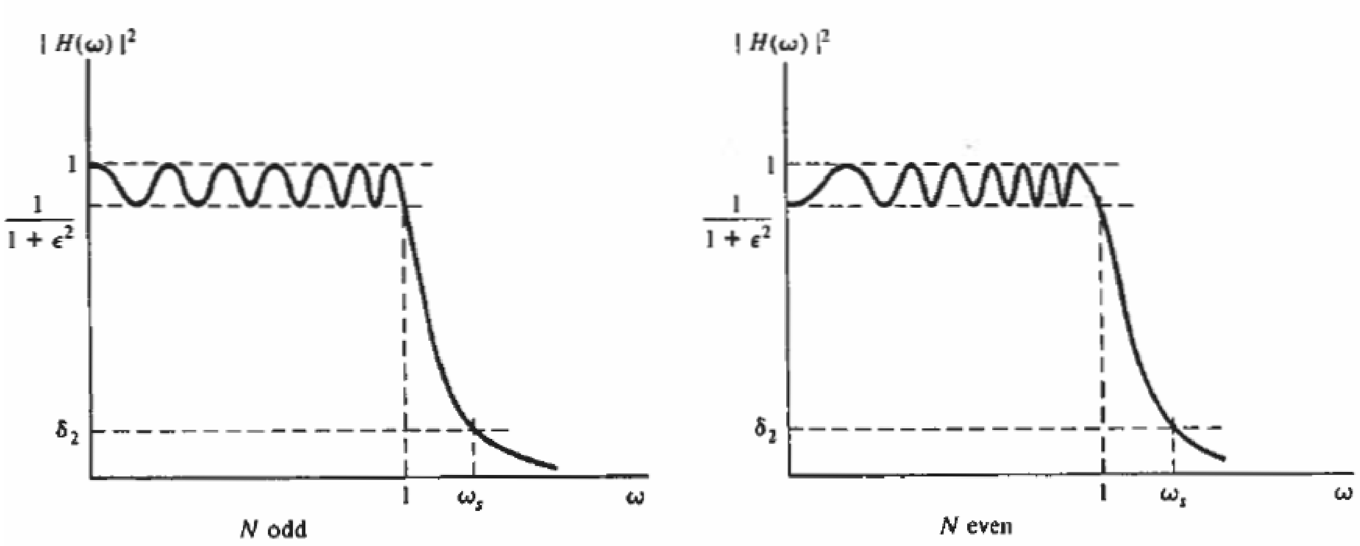


ANALOG FILTERS

- **The Chebyshev Filter**

$$|H(\omega)|^2 = \frac{1}{1+\varepsilon^2 C_N(\omega)}$$

$$C_N(\omega) = \begin{cases} \cos(N\cos^{-1}\omega), & |\omega| \le 1 \\ \cosh(N\cosh^{-1}\omega), & |\omega| > 1 \end{cases}$$



OUTLINE

- FIR and IIR Filters
- Filter Structures
- Analog Filters
- **FIR Filter Design**
- IIR Filter Design

FIR FILTER DESIGN

- **Linear phase filter**
  - A filter is called a linear phase filter if its transfer function can be expressed as

$$H(\Omega) = |H(\Omega)| e^{-jA\Omega}$$

    - Where A is a constant
    - The phase: $\theta = -A\Omega$, is a linear function of Ω
  - Example
    - Is the following filter a linear phase filter?

$$y(n) = x(n - n_0)$$

FIR FILTER DESIGN

- **Linear phase filter (Cont'd)**

$$H(\Omega) = |H(\Omega)| e^{-jA\Omega}$$

  - A linear phase filter introduces the same amount of delay, *A*, to all the frequencies (the same group delay)
    - If the phase is not linear, then frequency components at different frequencies will experience different delays → phase distortion
  - A subset of FIR filters can have linear phase
    - IIR filters do not have linear phase → IIR filter will introduce phase distortion
    - Not all FIR filters have linear phase

FIR FILTER DESIGN

- **Linear phase filter**
  - If a length-N FIR filter is symmetric with the middle of the filter, i.e.

$$h(n) = h(N-1-n)$$

    - Then the filter has linear phase
  - Example

    h(n) = [1 2 3 3 2 1], h(n) = [1 2 3 4 3 2 1], h(n) = [2, 0, 0, 2]

  - Proof

FIR FILTER DESIGN

- **FIR filter design with window**
  - 1. Find the desired frequency response (DTFT) $H_d(\Omega)$
  - 2. Perform Inverse DTFT, find the impulse response $h_d(n)$
    - The impulse response usually has infinite length
  - 3. truncate $h_d(n)$ to finite length by multiplying it with a window function $w(n)$
    - E.g. rectangular window

$$w(n) = \begin{cases} 1, & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

$$h(n) = h_d\left[n-(N-1)/2\right]w(n)$$

- Different windows can be used to reduce the negative effects of truncation
  - Rectangular window, Hamming window, Hanning window, Bartlett window, Kaiser window, etc.
  - Hamming window

$$w(n) = \begin{cases} 0.54 - 0.46\cos\dfrac{2\pi n}{N-1}, & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

UNIVERSITY OF ARKANSAS

FIR FILTER DESIGN

- **Example**
  - Design a 9-point FIR filter to approximate a low pass filter with cut-off frequency $\Omega_c = 0.2\pi$

FIR FILTER DESIGN

- **Example**

```
% order of the filter
N = 9;

% cut-off frequency
Omega_c = 0.2*pi;

n = [-(N-1)/2:(N-1)/2];
% impulse response
hdn = Omega_c/pi*sinc(Omega_c/pi*n);

%wd = window(@rectwin, N).';
wd = window(@hamming, N).';

hn = hdn.*wd;
```

OUTLINE

- **FIR and IIR Filters**
- **Filter Structures**
- **Analog Filters**
- **FIR Filter Design**
- **IIR Filter Design**

IIR FILTER DESIGN

- **IIR Filter design methods**
  - 1. impulse invariance
  - 2. bilinear transformation
- **Method 1: Impulse invariance**
  - Directly sample the impulse response of an analog filter
  - If the impulse response and frequency response of the analog filter are

$$h_a(t) \qquad\qquad H_a(\omega)$$

- Then the impulse response and DTFT of the IIR filter are

$$h(n) = h_a(nT)$$

$$H(\Omega) = \frac{1}{T}\sum_{n=-\infty}^{+\infty} H_a\left(\frac{\Omega}{T} - \frac{2\pi n}{T}\right)$$

- It will cause spectrum aliasing if the analog filter has a bandwidth larger than $\frac{1}{2T}$

UNIVERSITY OF ARKANSAS

IIR FILTER DESIGN

- **Method 1: Impulse invariance**
  - Analog filter: ω, $s = \sigma + j\omega$,
  - Digital filter: Ω, $z = e^{j\Omega}$
  - Relationship between s and z

$$z = \exp(Ts)$$

$$s = \frac{1}{T}\log z$$

IIR FILTER DESIGN

- **Method 2: Bilinear transformation**
  - Use bilinear transformation to overcome aliasing

$$s = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}} \qquad z = \frac{1+(T/2)s}{1-(T/2)s}$$

$$\omega = \frac{2}{T}\tan\frac{\Omega}{2}$$

  - Procedure:
    - 1. From the digital filter specifications, find the corresponding analog filter specifications by using the bilinear transformation

$$\omega = \frac{2}{T}\tan\frac{\Omega}{2}$$

    - 2. Find the corresponding analog filter $H_a(s)$
    - 3. Find the equivalent digital filter

$$H(z) = H_a(s)\big|_{s=\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}}$$

IIR FILTER DESIGN

- **Example**
  - If the transfer function of an analog filter is

$$H(s) = \frac{1}{s+1}$$

    - Find the equivalent digital filter by using bilinear transformation. Assume the sampling frequency is 1000 Hz

IIR FILTER DESIGN

- **Example**

```
% analog Butterworth filter, in the form of zeros and poles
% cut-off frequency 1 rad/s
[z,p,k] = buttap(2); % creates a 2-pole Butterworth filter

% transfer function
[num,den] = zp2tf(z,p,k); % poles & zeros to transfer
function

% need to transform the cutoff frequency to 100 Hz (200
rad/sec)
wc = 200*pi;
[num,den] = lp2lp(num,den,wc);

% sampling frequency 1000 Hz
Fs = 1000;

% generate the digital filter with the bilinear
transformation
[numd,dend] = bilinear(num,den,Fs);

freqz(numd, dend)
```