Realization of digital filters

Introduction to Digital Signal Processing

DEPT. of EE

Sogang University

Direct Realization of IIR Filters

Causal, rational, stable rational function

$$H(z) = \boxed{\frac{B(z)}{A(z)}} = \frac{b[0] + b[1]z^{-1} + b[2]z^{-2} + \cdots + b[M]z^{-M}}{a[0] + a[1]z^{-1} + a[2]z^{-2} + \cdots + a[N]z^{-N}}$$

$$y[n] = \sum_{k=0}^{M} b[k]x[n-k] - \sum_{k=1}^{N} a[k]y[n-k] \qquad a[0] = 1$$

Direct realization

Direct forms use coefficients a[k] and b[k] directly

Direct Form 1: $\rightarrow \boxed{B(z)} \rightarrow \boxed{\frac{1}{A(z)}} \rightarrow$

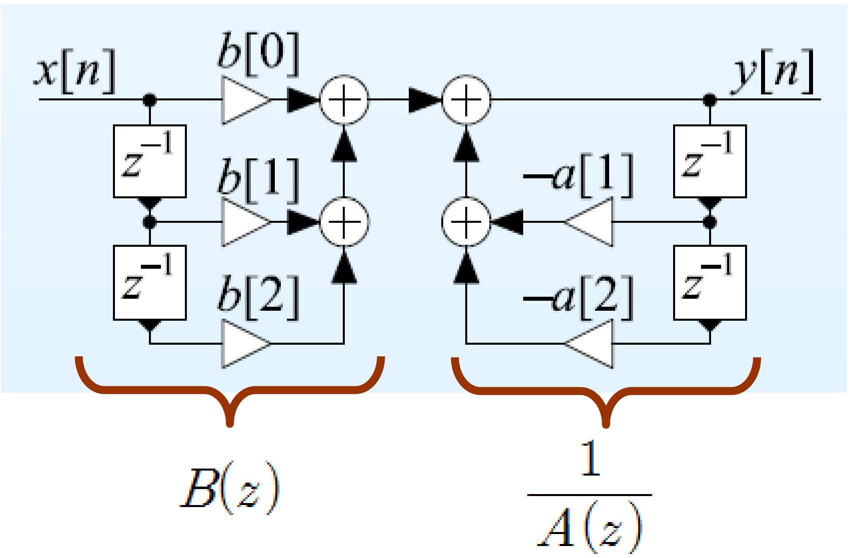
Direct Form 2: $\rightarrow \boxed{\frac{1}{A(z)}} \rightarrow \boxed{B(z)} \rightarrow$

Direct Realization of IIR Filters

Direct Form 1

$$H(z)=\frac{B(z)}{A(z)}=\frac{b[0]+b[1]z^{-1}+b[2]z^{-2}+\cdots+b[M]z^{-M}}{a[0]+a[1]z^{-1}+a[2]z^{-2}+\cdots+a[N]z^{-N}} \qquad a[0]=1$$

$$y[n]=\sum_{k=0}^{M}b[k]x[n-k]-\sum_{k=1}^{N}a[k]y[n-k]$$



Direct Realization of IIR Filters

Direct Form 2

$$H(z)=\frac{B(z)}{A(z)}=\frac{b[0]+b[1]z^{-1}+b[2]z^{-2}+\cdots+b[M]z^{-M}}{a[0]+a[1]z^{-1}+a[2]z^{-2}+\cdots+a[N]z^{-N}} \qquad a[0]=1$$

$$y[n]=\sum_{k=0}^{M}b[k]x[n-k]-\sum_{k=1}^{N}a[k]y[n-k]$$

$$U(z)=\frac{1}{A(z)}X(z) \quad \Rightarrow \quad Y(z)=B(z)U(z)$$

$$X(z)=A(z)U(z)$$

$$y(n)=\sum_{k=1}^{N}b_k u(n-k)$$

$$\sum_{k=0}^{N}a_k u(n-k)=x(n)$$

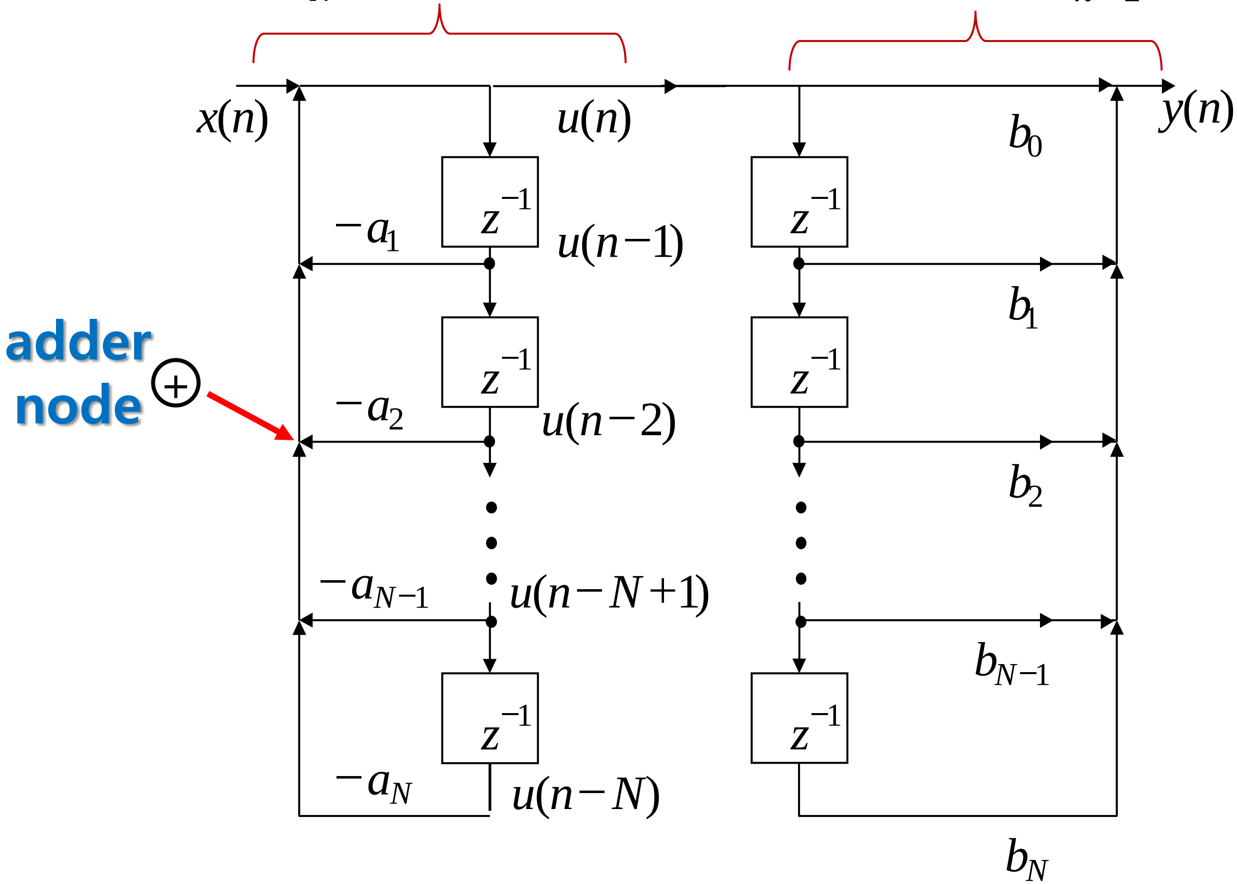
$$\Leftrightarrow \; u(n)=-a_1u(n-1)-\cdots-a_Nu(n-N)+x(n)$$

Direct Realization of IIR Filters

Direct Form 2

$$u(n) = x(n) - a_1 u(n-1) - a_2 u(n-2) - \cdots - a_N u(n-N)$$

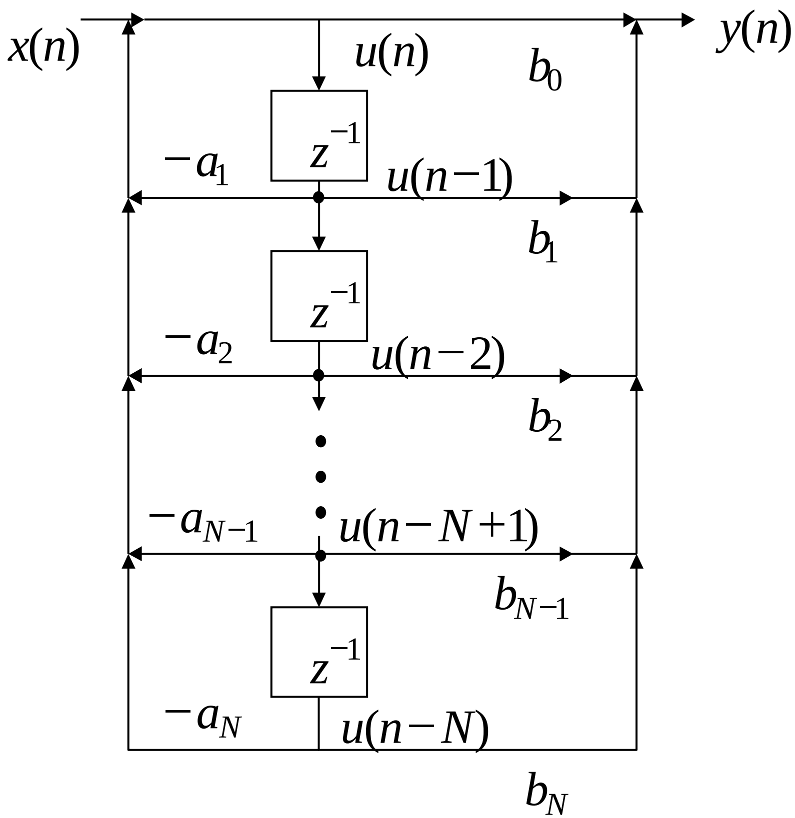
$$y(n) = \sum_{k=1}^{N} b_k u(n-k) \quad (M = N)$$

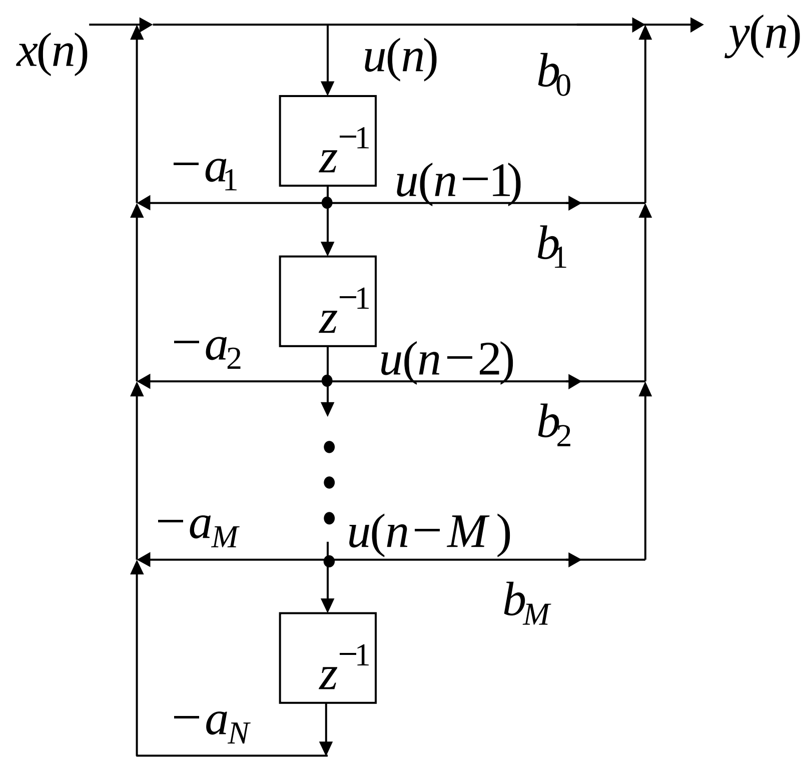


Direct Realization of IIR Filters

Direct Form 2 [Canonical Form]

A digital filter structure is said to be canonic if the number of delays is equal to the filter order





Direct Realization of IIR Filters

Direct form or direct realization (Canonical form)

- z^{-1} represents delays, i.e., memory devices like latches or registers
- use minimal possible number of delays (N delays)
- 2N+1 (or N+M+1) multipliers and 2N (or N+M) adders
- recursive realization.

  Transient response → steady-state response
- Initialization of filter

  Memories are initialized either

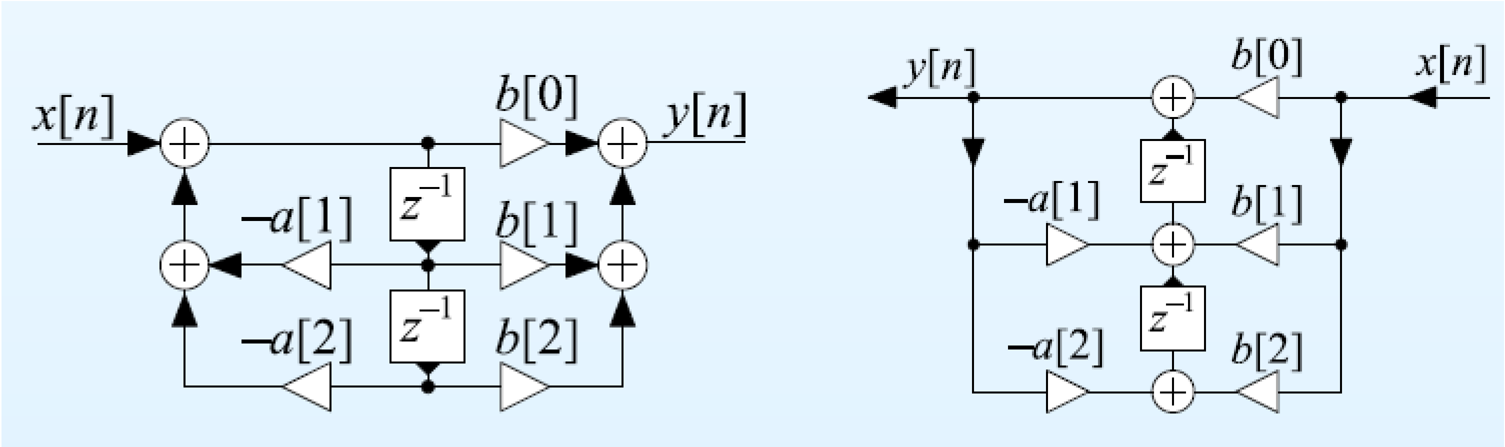
  to zero or

  to other proper values.

Direct Realization of IIR Filters

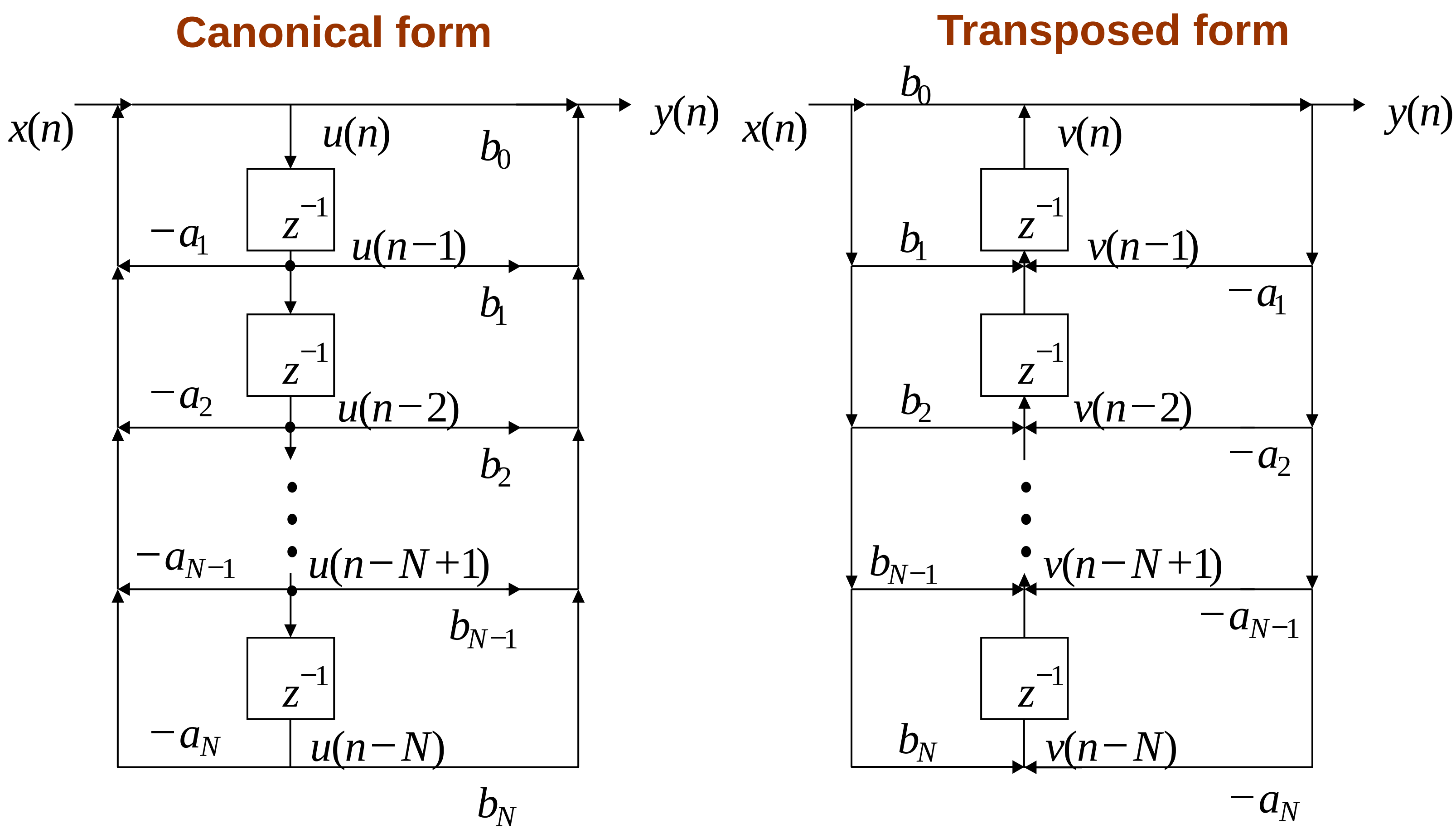
Transposed Form: Any block diagram can be converted into an equivalent a transposed form

- **Reverse direction of each interconnection (branch)**
- **Reverse direction of each multiplier**
- **Change junctions to adders and vice versa**
- **Interchange the input and output signals**



Direct Realization of IIR Filters

Transposed Direct From 2



Parallel Realization of IIR Filters

Rational transfer function

$$H(z)=\frac{b(z)}{a(z)}=\frac{b[0]+b[1]z^{-1}+b[2]z^{-2}+\cdots+b[q]z^{-q}}{a[0]+a[1]z^{-1}+a[2]z^{-2}+\cdots+a[p]z^{-p}}$$

- $q > p$: FIR + IIR

$$H(z)=\frac{b(z)}{a(z)}=c_0+\cdots+c_{q-p}z^{-(q-p)}+\sum_{i=1}^{p}\frac{A_i}{1-\alpha_i z^{-1}}$$

※ Note: If a_i is complex, then $a^*{}_i$ is also a pole and $A_i=A_i^*$

➔ $$\frac{(A_i+A_i^*)-(A_i\alpha_i^*+A_i^*\alpha_i)z^{-1}}{1-(\alpha_i+\alpha_i^*)z^{-1}+\alpha_i\alpha_i^*z^{-2}}$$

Parallel Realization of IIR Filters

Rational transfer function

- $p \geq q$

※ **Order =** N_1 **(real poles) +** $2N_2$ **(complex poles)**

$$\Rightarrow \quad H(z) = c_0 + \sum_{i=1}^{N_1} \frac{f_i}{1 + e_i z^{-1}} + \sum_{i=1}^{N_2} \frac{f_{N_1+2i-1} + f_{N_1+2i} z^{-1}}{1 + e_{N_1+2i-1} z^{-1} + e_{N_1+2i} z^{-2}}$$

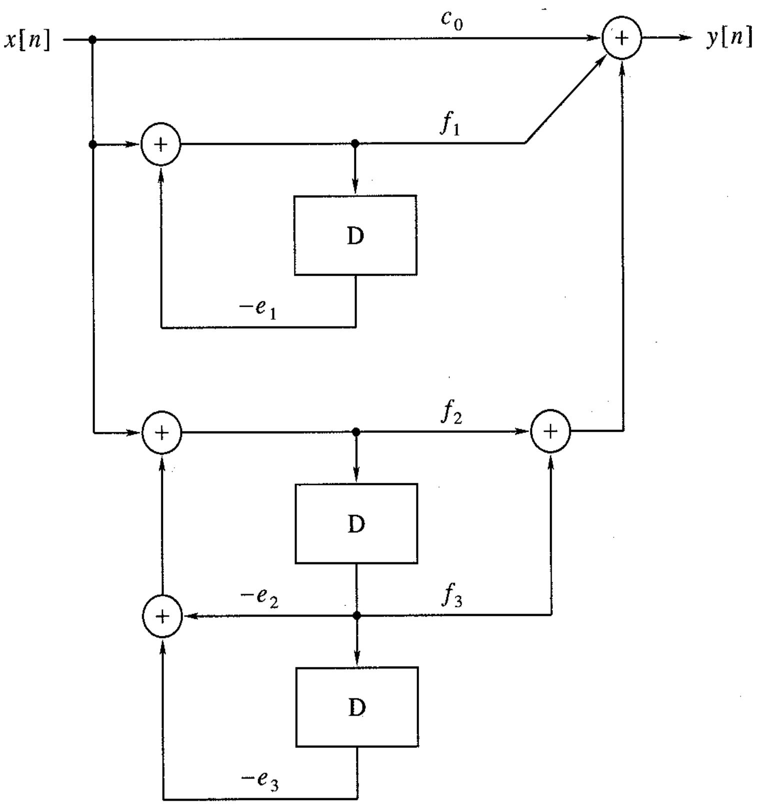
where real numbers $\{e_i,\ f_i,\ 1 \leq i \leq N\}$

depend on $\{A_i,\ \alpha_i,\ 1 \leq i \leq N\}$

Parallel Realization of IIR Filters

Parallel realization

- Ex: $N_1 = 1,\ N_2 = 1 \ \Rightarrow\ N = 3$



$$H(z) = c_0 + \sum_{i=1}^{N_1} \frac{f_i}{1 + e_i z^{-1}} + \sum_{i=1}^{N_2} \frac{f_{N_1+2i-1} + f_{N_1+2i} z^{-1}}{1 + e_{N_1+2i-1} z^{-1} + e_{N_1+2i} z^{-2}}$$

Parallel Realization of IIR Filters

Parallel realization

- $p = q$:

  direct and parallel forms require the same multipliers and adders.

- $p > q$:
  - ✓ Direct form requires $p+q+1$ multiplications and $(p+q)$ additions.
  - ✓ Parallel form requires $2p+1$ multiplications and additions.

- Number of delays are same.
- **Better sensitivity of the frequency response to finite word length (limited precision)**

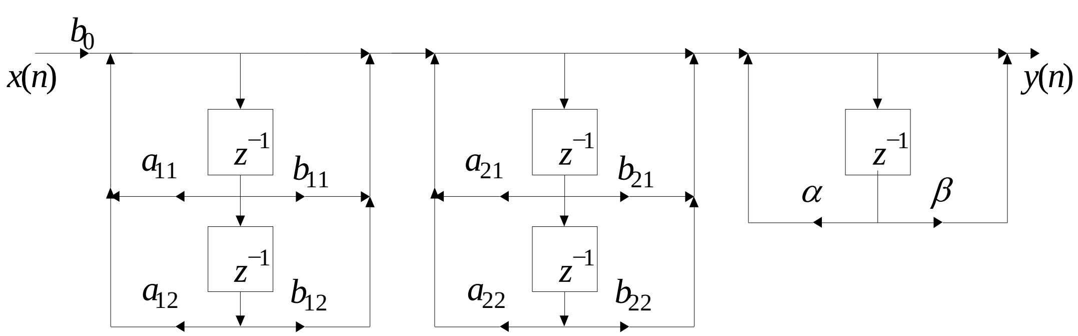
Cascade Realization of IIR Filters

Cascade realization

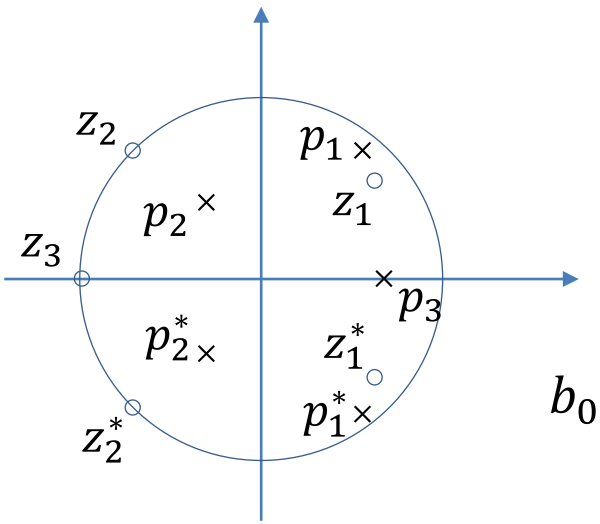
$$H(z) = b_0 \frac{1-\beta z^{-1}}{1-\alpha z^{-1}} \prod_{i=1}^{N} \frac{1+b_{i1}z^{-1}+b_{i2}z^{-2}}{1-a_{i1}z^{-1}-a_{i2}z^{-2}}$$

Biquad

- Can be used to realize any IIR filters.
- **Most widely used because less sensitive to quantization error**
- **The realization is not unique**
  - ✓ exist multiple ways of paring poles and zeros to make each biquads.
  - ✓ **multiple ways of ordering the biquads in the cascade connection**
- **Filtering performance can differ for different realization.**



Cascade Realization of IIR Filters



$$b_0 \left[\frac{(1-z_a z^{-1})}{(1-p_a z^{-1})}\frac{(1-z_a^* z^{-1})}{(1-p_a^* z^{-1})}\right]\left[\frac{(1-z_b z^{-1})}{(1-p_b z^{-1})}\frac{(1-z_b^* z^{-1})}{(1-p_b^* z^{-1})}\right]\frac{(1-z_b z^{-1})}{(1-p_b z^{-1})}$$

p_1	z_1	p_2	z_2
p_1	z_2	p_2	z_1
p_1	z_3	p_2	z_1
p_1	z_3	p_2	z_2

$\vdots$

Cascade Realization of IIR Filters

Pairing in cascade realization

1. **Select the pair of complex poles**

   1) nearest to the unit circles, pairing them with nearest complex zeros

   - ✓ **To make the mag response (MR) as flat as possible**
   - ✓ **Before the quantization errors accumulate too much, use poles nearest to the unit circle.**

   2) farthest to the unit circle with nearest complex zeros

   - ✓ **Poles near the unit circle make MR have the highest peaks and introduce most noise,**
   - ✓ **so place them last in the chain (minimize overflow)**

2. **Continue to pair the poles and zeros according to the same rule.**

   - ✓ **After complex zeros are exhausted, use real zeros**

3. **Pair the real poles.**

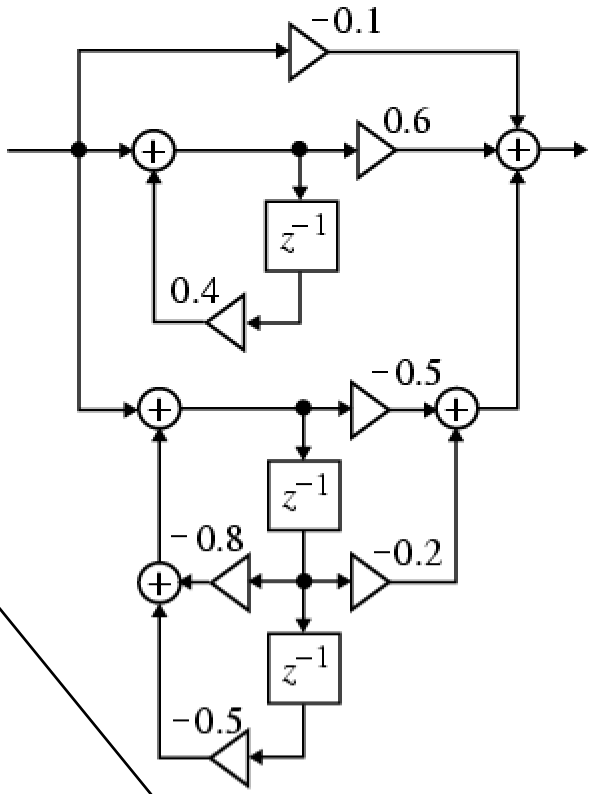
Realization of IIR Filters

Parallel realization 1

$$H(z)=\frac{0.44z^{-1}+0.362z^{-2}+0.02z^{-3}}{1+0.4z^{-1}+0.18z^{-2}-0.2z^{-3}}$$

A partial-fraction expansion in z^{-1} yields

$$H(z)=-0.1+\frac{0.6}{1-0.4z^{-1}}+\frac{-0.5-0.2z^{-1}}{1+0.8z^{-1}+0.5z^{-2}}$$



b=[0 .44 .362 .02]; a=[1 .4 .18 -.2];

[R, p, k] = residuez(b, a)

R =
-0.2500 - 0.0000i
-0.2500 + 0.0000i
0.6000

p =
-0.4000 + 0.5831i
-0.4000 - 0.5831i
0.4000

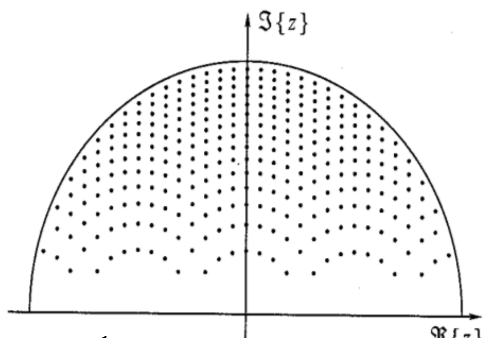
k =
-0.1000

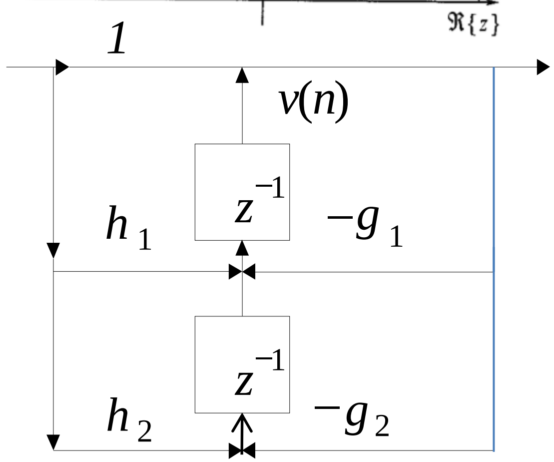
$$\frac{(A_i+A_i^*)-(A_i\alpha_i^*+A_i^*\alpha_i)z^{-1}}{1-(\alpha_i+\alpha_i^*)z^{-1}+\alpha_i\alpha_i^*z^{-2}}$$

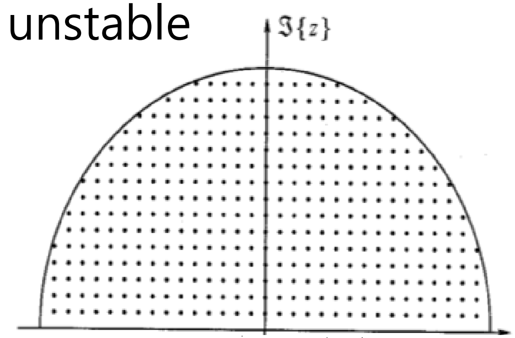
Coupled Cascade Realization

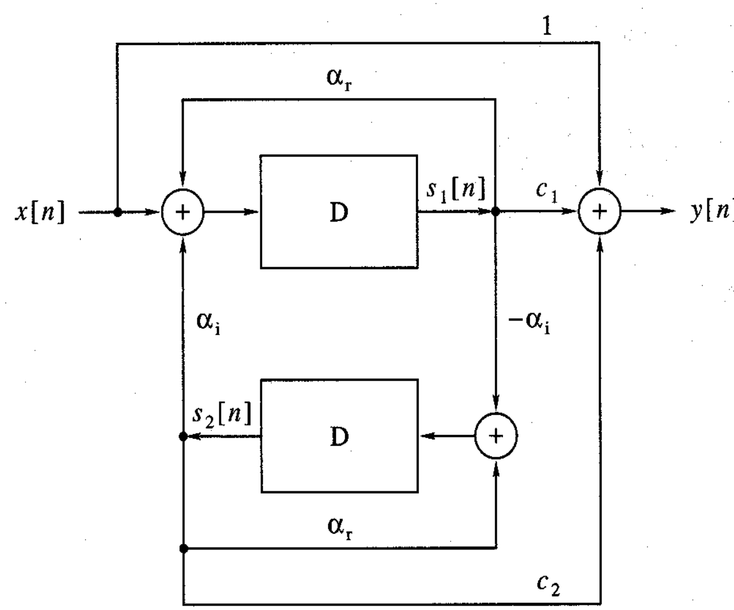
Coupled cascade realization

- When the word length is short and filter has poles near z= + - 1
- When quantization error can make the filter unstable
- $\alpha = \alpha_r + j\alpha_i$: complex pole



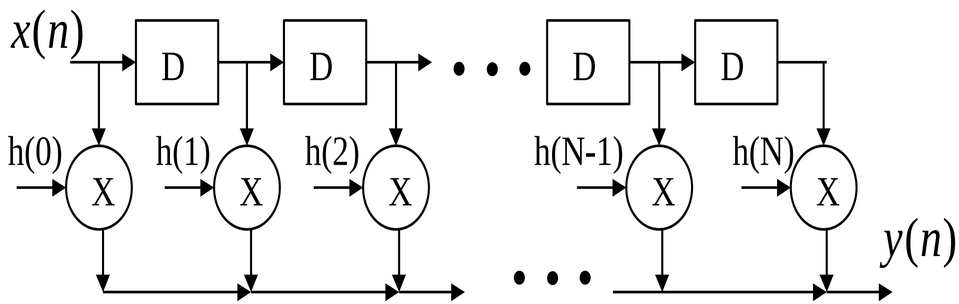






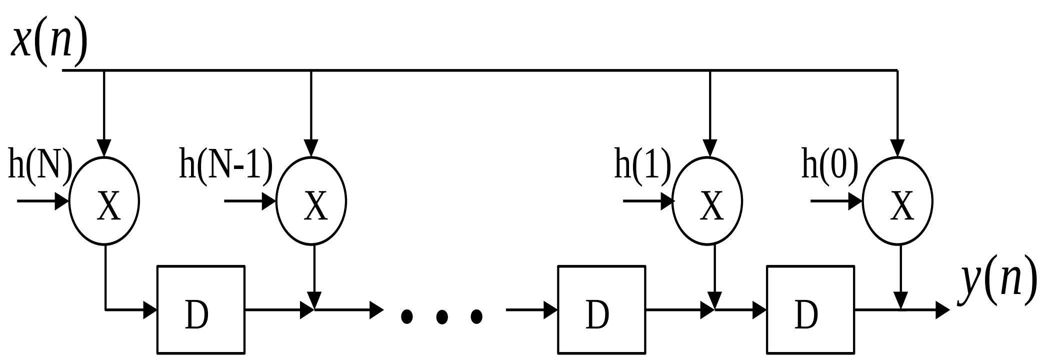
Direct Realization of FIR Filters

Standard form



- $y(n) = \sum_{k=0}^{N} h(k)x(n-k)$: N+1 Multipliers, N adders

Transposed form



Cascaded form is seldom used for FIR.

Direct Realization of FIR Filters

Structure for linear phase FIR filters

- can reduce the # of multiplications by half.

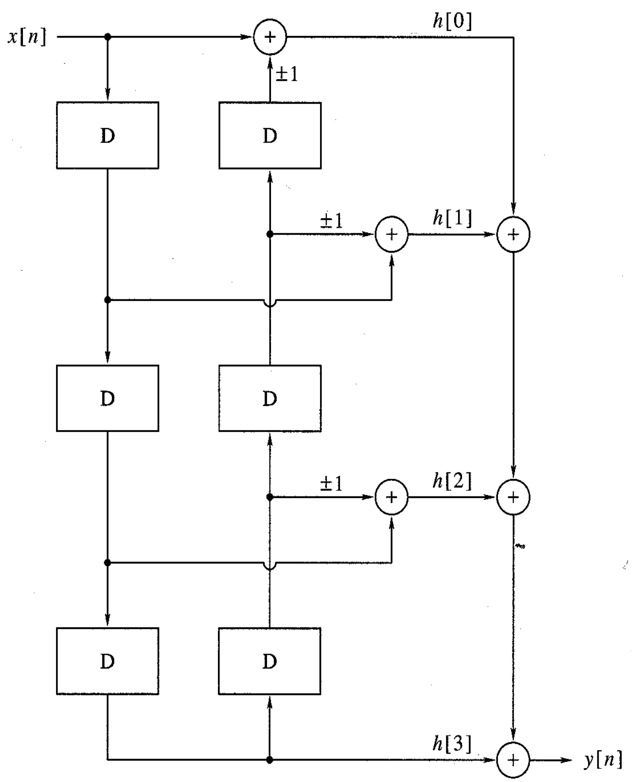


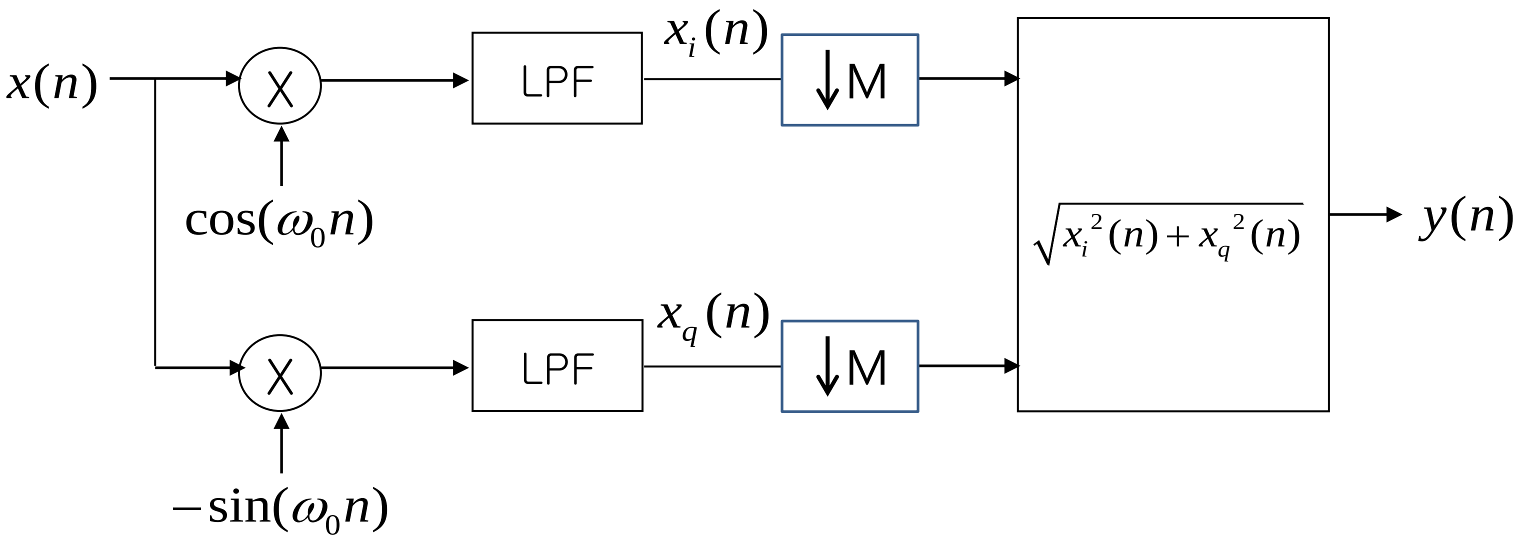
Figure 11.7 Direct realization of a symmetric or antisymmetric FIR filter.

FIR Filter: Design and Implementation example

Design, verification, implementation, Test

- **SW model (gold standard): Regular MATLAB model**
- **HW simulation model: MATLAB model using the same number system as in the real HW design**
- **HW design: VHDL**

FIR Filter: Design and Implementation example



Real design challenge / practice :

- ❖ FIR LPF design → Implementation
- ❖ Rate converter design & Implementation
- ❖ Cos, sin waveform generator
- ❖ Square root function

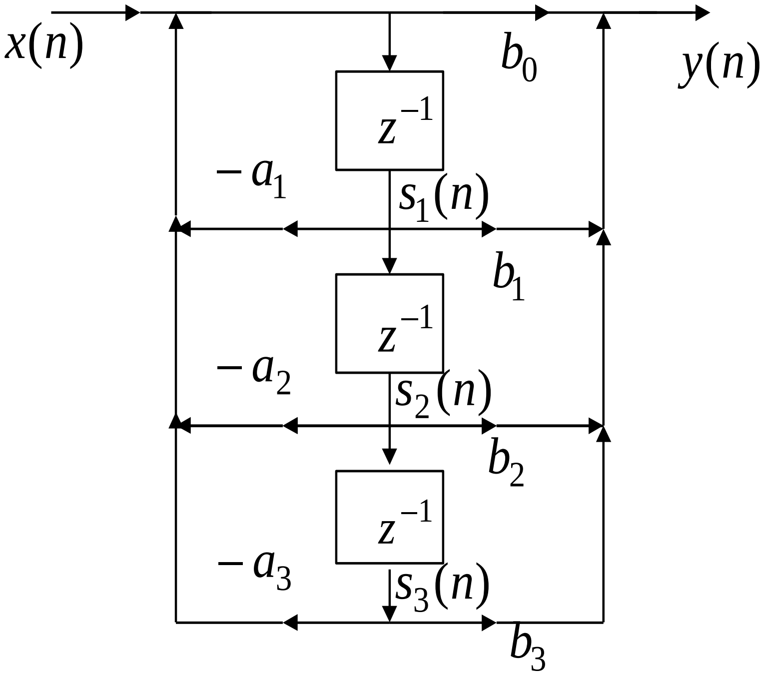
State-space Representation

State-space Concept

$$\begin{bmatrix} s_1(n+1) \\ s_2(n+1) \\ s_3(n+1) \end{bmatrix} = \begin{bmatrix} -a_1 & -a_2 & -a_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} s_1(n) \\ s_2(n) \\ s_3(n) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} x(n)$$

$$y(n) = [c_1 \quad c_2 \quad c_3] \begin{bmatrix} s_1(n) \\ s_2(n) \\ s_3(n) \end{bmatrix} + b_0 x(n)$$

$$c_k = b_k - b_0 a_k, \quad 1 \le k \le 3$$



State-space Representation

State-space Representation

- State equation

$$s[n+1] = A_1 s[n] + B_1 x[n] \quad s(n) \text{ : state vector}$$

- Output equation

$$y(n) = C_1 s[n] + D_1 x[n]$$

where
- State matrix $A_1 =$
- input matrix B_1

$$A_1 = \begin{bmatrix} -a_1 & -a_2 & \cdots & -a_{N-1} & -a_N \\ 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

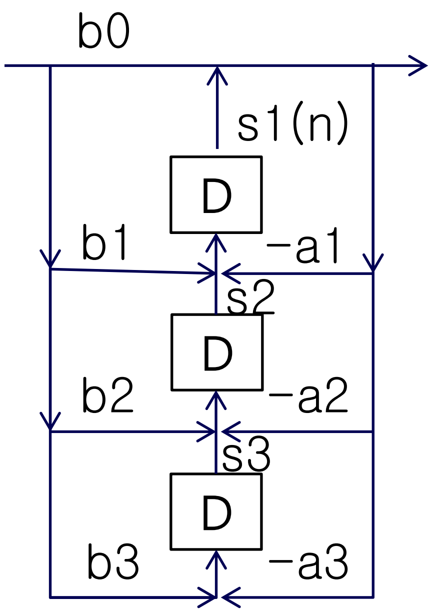
- output matrix $C_1 = [c_1 \quad c_2 \cdots c_N]$

- direct transmission matrix $D_1 = b_0$

State-space Representation

For transposed direct transformation

$$A_2 = A_1^t, \quad B_2 = C_1^t,$$
$$C_2 = B_1^t, \quad D_2 = D_1$$



- Two representations are dual

  or transpose to each other.

- Note:
  - A rational transfer function has an infinite number of state-space representations
  - You can assign the state variables in many different ways.
  - # of state vectors = # of memories

State-space Representation

Application of state space

$$s[n+1] = A\,s[n] + Bx[n]$$
$$y(n) = C\,s[n] + Dx[n]$$

- Impulse response for $s[n]=0,\ n\le 0$ and $x[n]=0,\ n<0$
  $x[n] = \delta[n]$

$$s[0]=0,\quad s[1]=B,\quad s[2]=AB,\quad s[3]=A^2B,\ \ldots$$

$$=>\ s[n] = \begin{cases} 0, & n\le 0 \\ A^{n-1}B, & n>0 \end{cases}$$

$$=>\ h[n] = \begin{cases} 0 & n<0 \\ D & n=0 \\ CA^{n-1}B & n>0 \end{cases}$$

- $h(n)$ can be obtained without using z-transform and partial fraction expansion.

State-space Representation

General system response when the system is relaxed.

$$s[n] = \begin{cases} 0, & n \le 0 \\ \sum_{k=0}^{n-1} A^{n-k-1} Bx[k] & n > 0 \end{cases} \qquad y[n] = \begin{cases} 0, & n \le 0 \\ Dx[0], & n = 0 \\ \sum_{k=0}^{n-1} C\ A^{n-k-1} Bx[k] + Dx[n] & n > 0 \end{cases}$$

Z-transform solution and Transfer function

$$z\,S(z) - z\,s[0] = A\ S(z) + BX(z)$$

$$\Rightarrow$$

$$(z\,I - A)\,S(z) = BX(z) + z\,s[0]$$

$$S(z) = (z\,I - A)^{-1}\{ BX(z) + z\,s[0]\}$$

$$\Rightarrow$$

$$Y(z) = [\,C(z\,I - A)^{-1}B + D]X(z) + C(z\,I - A)^{-1} z\,s[0]$$

$$\Rightarrow$$

$$H(z) = \frac{Y(z)}{X(z)} = C(z\,I - A)^{-1}B + D$$

$$\boxed{\begin{aligned} s[n+1] &= A\,s[n] + Bx[n] \\ y(n) &= C\,s[n] + Dx[n] \end{aligned}}$$