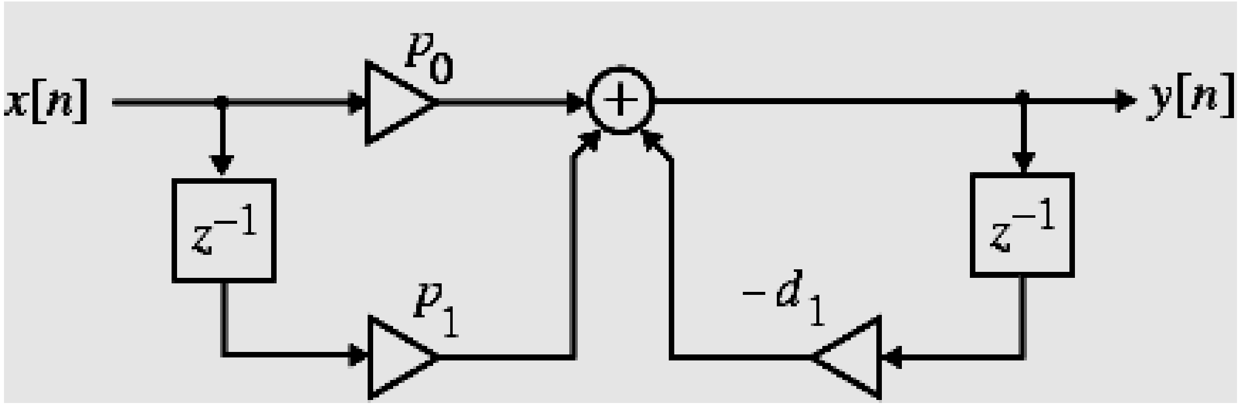
Digital Signal Processing

Digital Filter Structure

Wenhui Xiong
NCL
UESTC

Block Diagram Representation

- Computational algorithm by inspection
- Shows relation between the output and input
- Easy to derive “equivalent” block diagrams
- Easy to determine the hardware requirements
- Can be developed from the transfer function



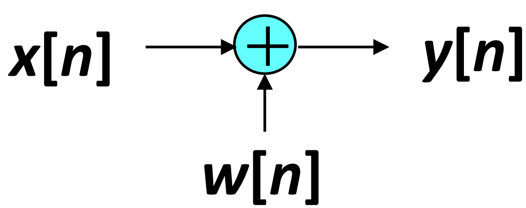
$y[n] = -d_1 y[n-1] + p_0 x[n] + p_1 x[n-1]$

Block Diagram Representation

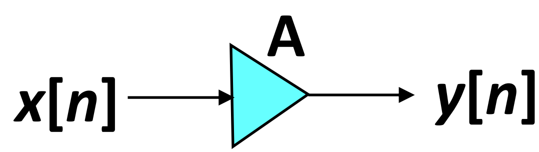
The calculation at each step requires:

- The previously calculated value
- The present value of the input
- The delayed value of the input

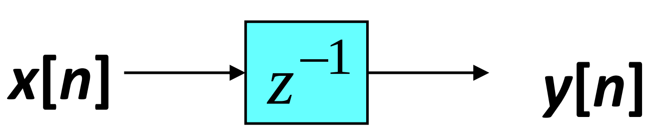
Basic Blocks



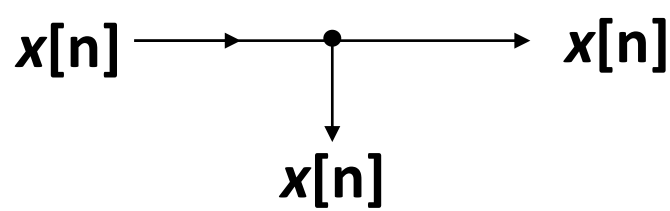
Adder



Multiplier



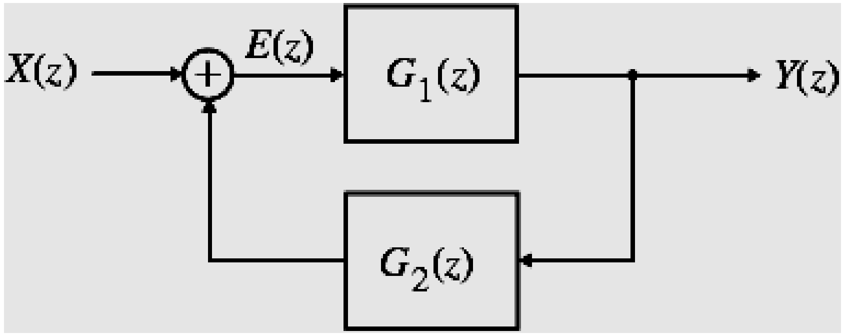
Unit delay



Pick-off node

Block Diagram Representation

Example



The output of the adder

$$E(z) = X(z) + G_2(z)Y(z)$$

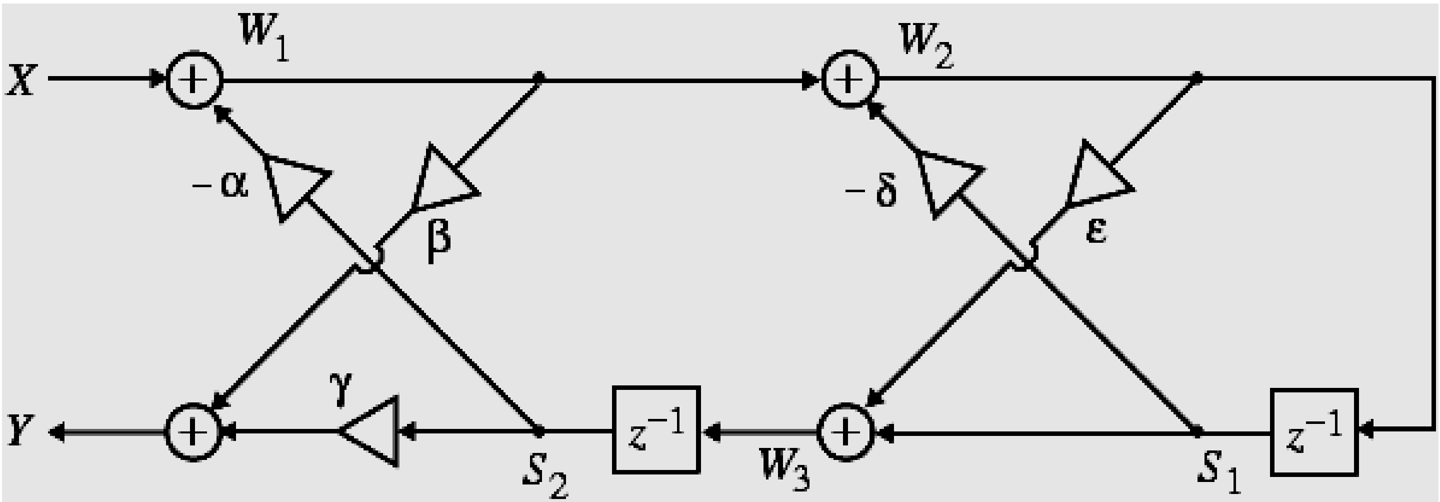
$$Y(z) = G_1(z)E(z)$$

$$Y(z) = G_1(z)\left(X(z) + G_2(z)Y(z)\right)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{G_1(z)}{1 - G_1(z)G_2(z)}$$

Block Diagram Representation

Example



$$W_1 = X - \alpha S_2 \qquad W_2 = W_1 - \delta S_1$$

$$S_1 = z^{-1} W_2 \qquad S_2 = z^{-1} W_3$$

$$W_3 = S_1 + \varepsilon W_2 \qquad Y = \beta W_1 - \gamma S_2$$

Block Diagram Representation

$$\boxed{W_2 = W_1 - \delta S_1 \quad S_1 = z^{-1} W_2} \quad W_3 = S_1 + \varepsilon W_2$$

$$W_2 = W_1 - \delta z^{-1} W_2 \qquad W_3 = z^{-1} W_2 + \varepsilon W_2$$

$$W_2 = \frac{1}{1 + \delta z^{-1}} W_1 \qquad W_3 = (z^{-1} + \varepsilon) W_2$$

$$W_3 = \frac{z^{-1} + \varepsilon}{1 + \delta z^{-1}} W_1$$

Block Diagram Representation

$$W_1 = X - \alpha S_2 \qquad S_2 = z^{-1} W_3 \qquad Y = \beta W_1 - \gamma S_2$$

$$W_1 = X - \alpha z^{-1} W_3 \qquad Y = \beta W_1 + \gamma z^{-1} W_3$$

$$W_3 = \frac{z^{-1} + \varepsilon}{1 + \delta z^{-1}} W_1$$

$$H(z) = \frac{Y}{X} = \frac{\beta + (\beta\delta + \gamma\varepsilon) z^{-1} + \gamma z^{-2}}{1 + (\delta + \alpha\varepsilon) z^{-1} + \alpha z^{-2}}$$

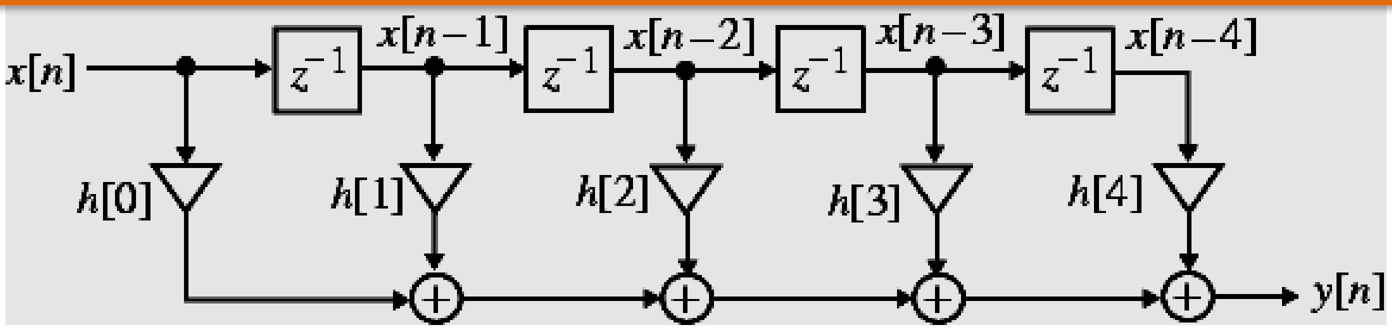
Note： *Delay-free loop:* feedback loops without any delay elements (physically impossible).

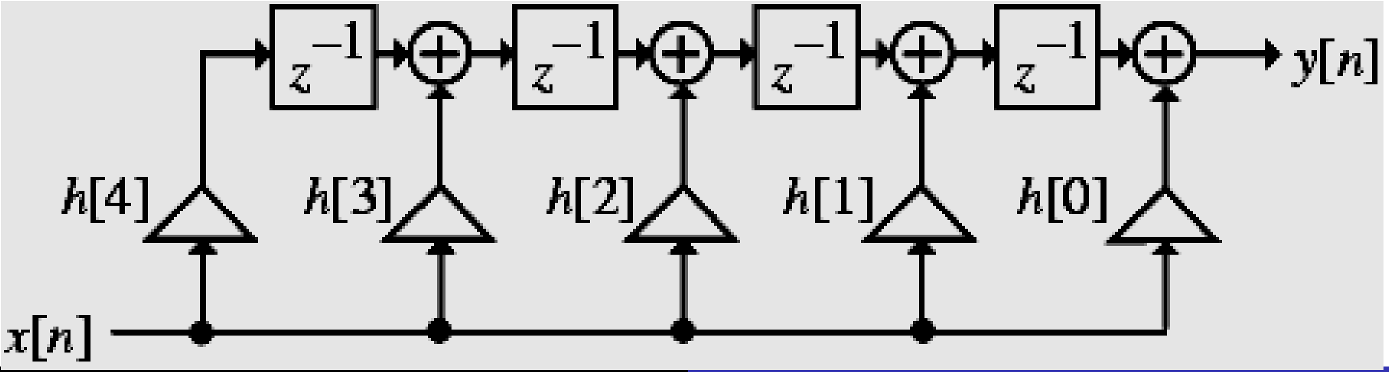
Equivalent Structures

Transpose Operation

- Reverse all paths
- Replace pick-off nodes by adders, and vice versa
- Interchange the input and output nodes

Example



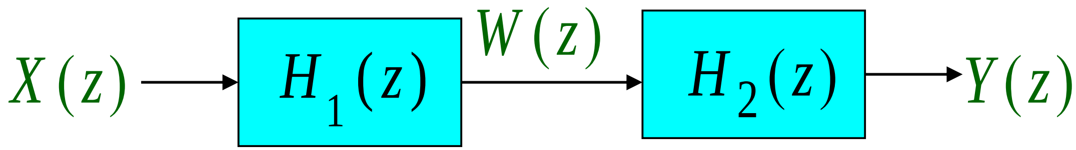


Equivalent Structures

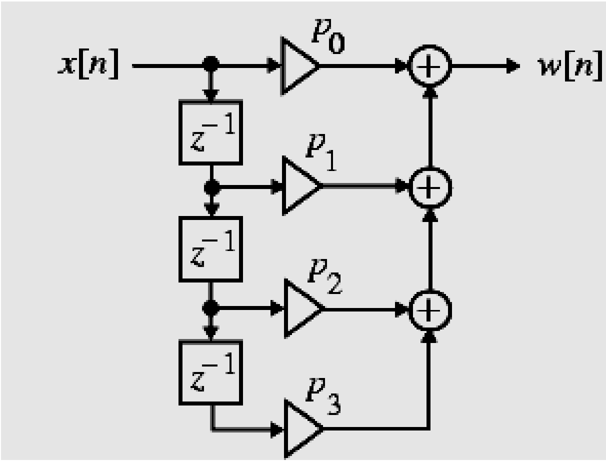
- There are infinite number of realization of the same system
- Each realization behaves the same—with float point number
- Fix point realizations behave differently
- Choose a structure that has the least quantization effects

IIR Filter Implementation

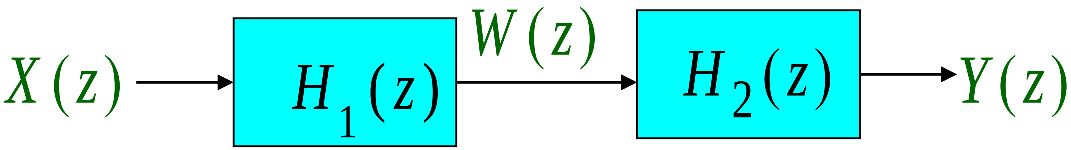
$$H(z)=\frac{P(z)}{D(z)}=\frac{p_0+p_1z^{-1}+p_2z^{-2}+p_3z^{-3}}{d_0+d_1z^{-1}+d_2z^{-2}+d_3z^{-3}}$$

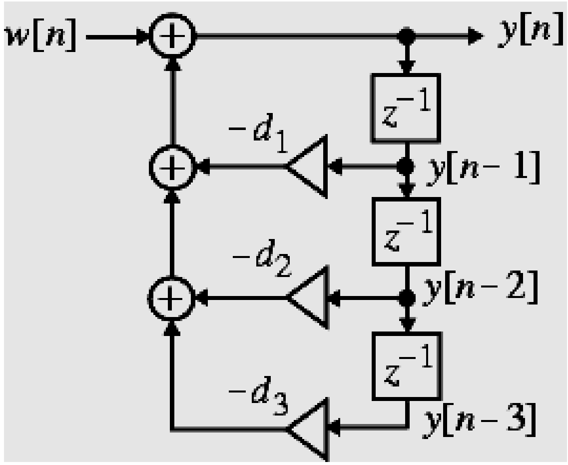


$$H_1(z)=P(z)=p_0+p_1z^{-1}+p_2z^{-2}+p_3z^{-3}$$



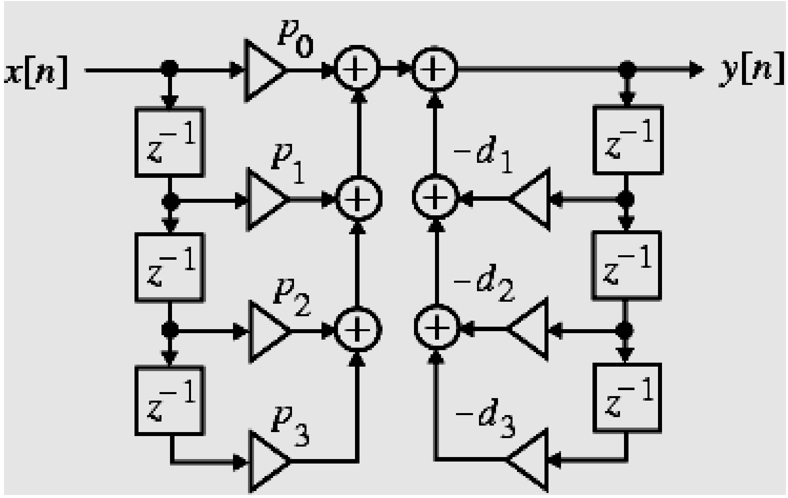
IIR Filter Implementation



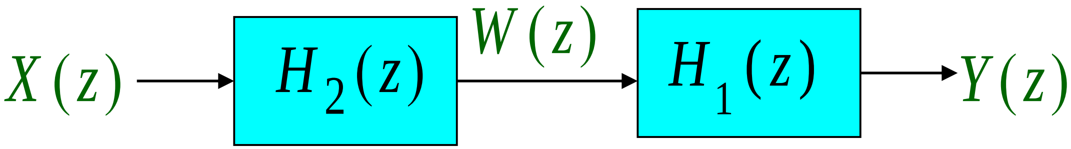


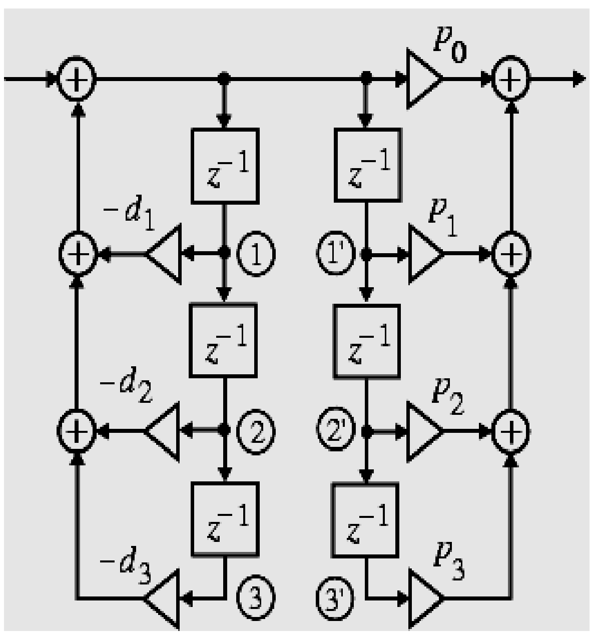
$$H_2(z) = \frac{1}{D(z)} = \frac{1}{d_0 + d_1 z^{-1} + d_2 z^{-2} + d_3 z^{-3}}$$

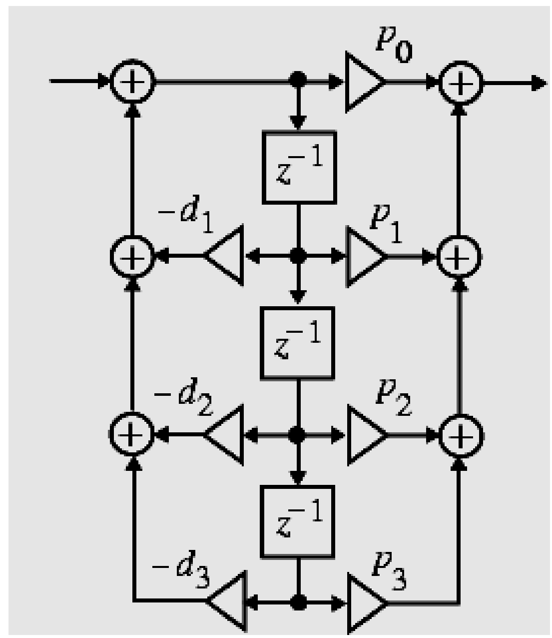
A cascade of $H_1(z)$ and $H_2(z)$ leads $H(z)$

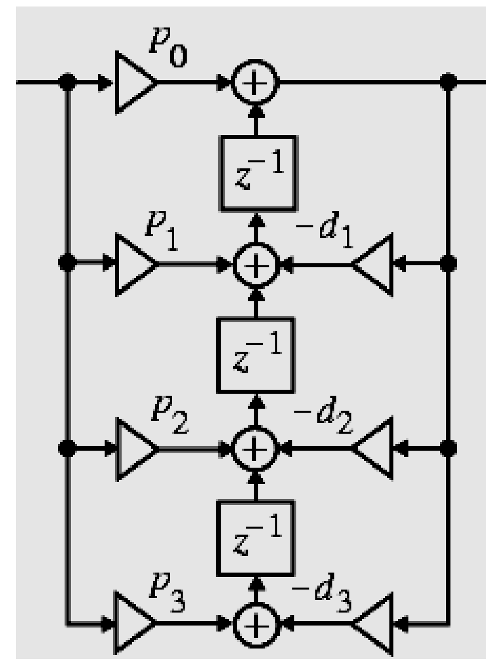


Direct Form I (non-canonic)









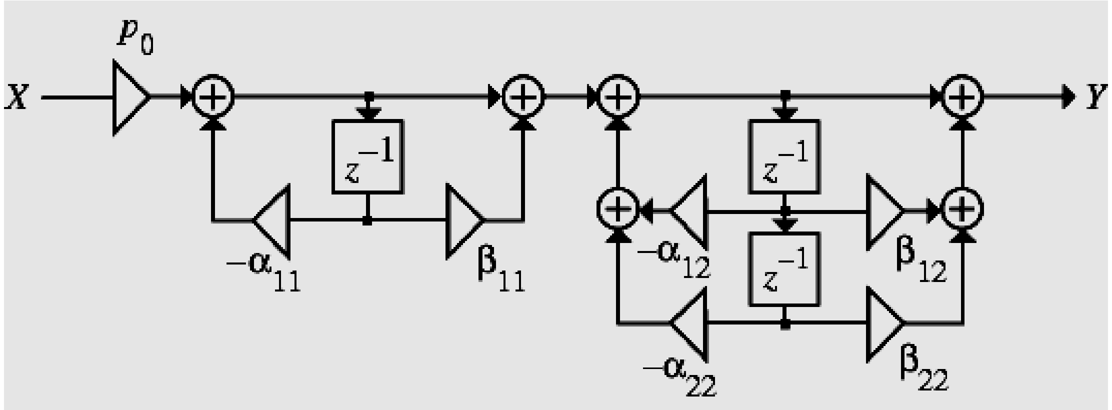
Canonic Form

IIR Filter Implementation

Factorize the polynomial into the product polynomials

$$H(z) = p_0 \prod \left(\frac{1 + \beta_{1k} z^{-1} + \beta_{2k} z^{-2}}{1 + \alpha_{1k} z^{-1} + \alpha_{2k} z^{-2}} \right)$$

$$H(z) = p_0 \left(\frac{1 + \beta_{11} z^{-1}}{1 + \alpha_{11} z^{-1}} \right) \left(\frac{1 + \beta_{12} z^{-1} + \beta_{22} z^{-2}}{1 + \alpha_{12} z^{-1} + \alpha_{22} z^{-2}} \right)$$

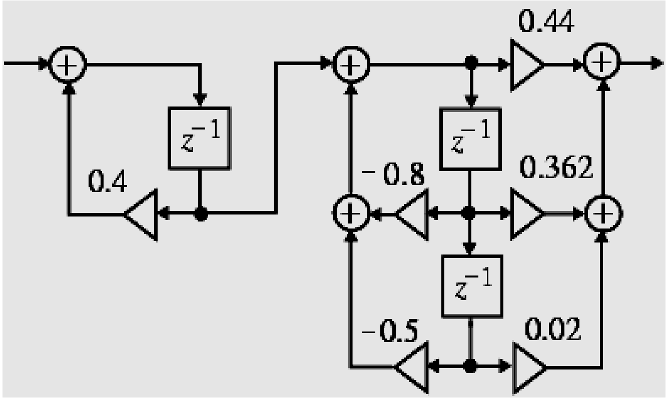


IIR Filter Implementation

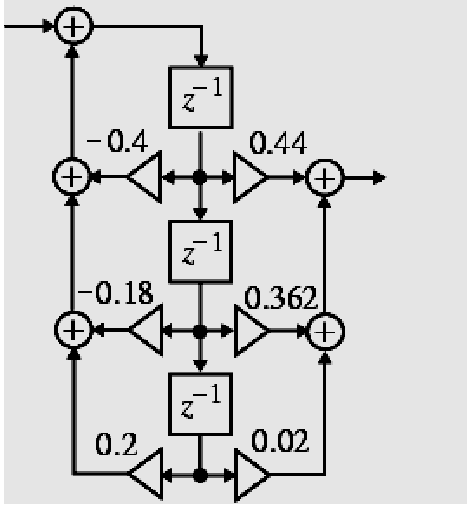
Example

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

$$=\frac{z^{-1}}{1-0.4z^{-1}}\frac{0.44+0.362z^{-1}+0.02z^{-2}}{1+0.8z^{-1}+0.5z^{-2}}$$



Cascade form



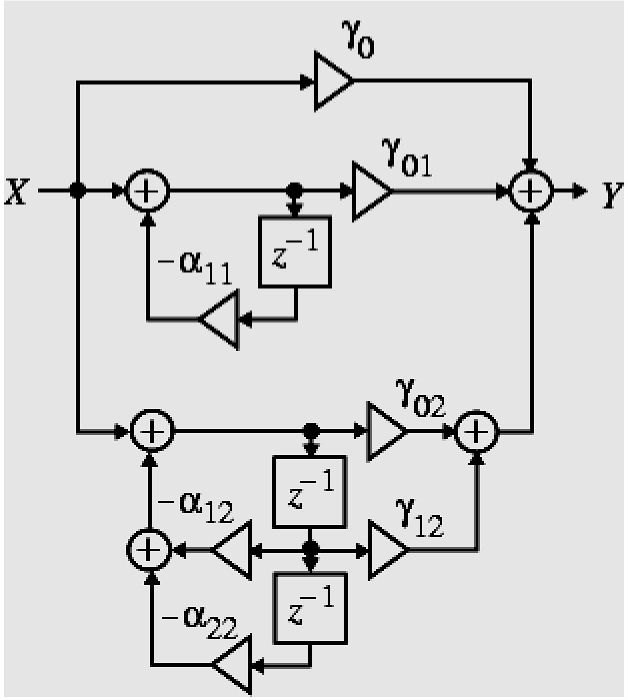
Direct form II

IIR Filter Implementation

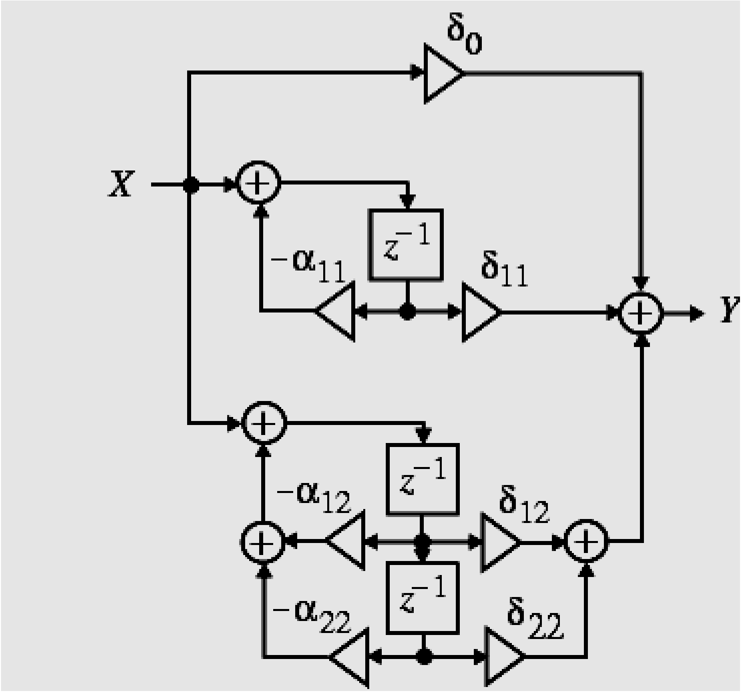
Factorize the polynomial into the sum polynomials

$$H(z) = \gamma_0 + \sum_k \left(\frac{\gamma_{0k} + \gamma_{1k} z^{-1}}{1 + \alpha_{1k} z^{-1} + \alpha_{2k} z^{-2}} \right)$$

$$H(z) = \delta_0 + \sum_k \left(\frac{\delta_{0k} z^{-1} + \delta_{1k} z^{-2}}{1 + \alpha_{1k} z^{-1} + \alpha_{2k} z^{-2}} \right)$$



Parallel form I



Parallel form II

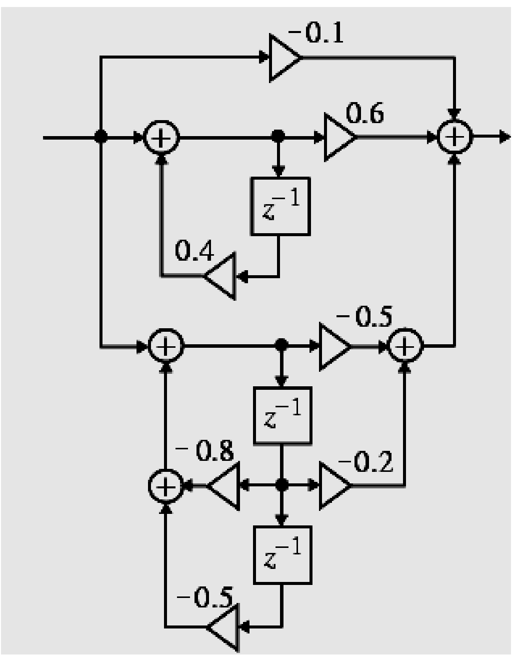
IIR Filter Implementation

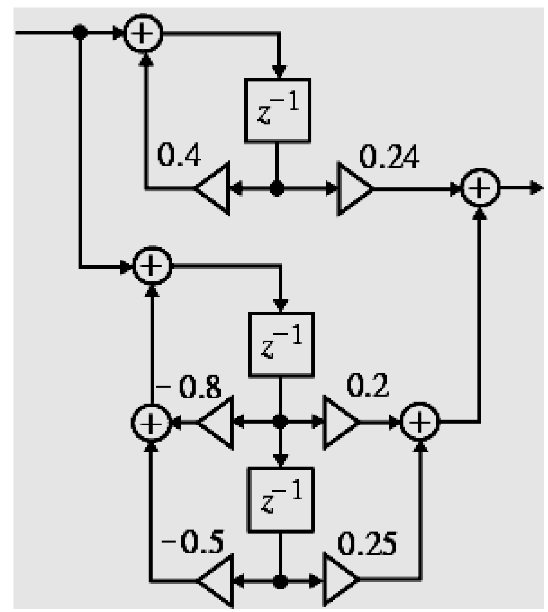
Example

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

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

$$H(z)=\frac{0.24z^{-1}}{1-0.4z^{-1}}+\frac{0.2z^{-1}+0.25z^{-1}}{1+0.8z^{-1}+0.5z^{-2}}$$





IIR Filter Implementation

Matlab Realization

- Factorization of the transfer function with p2sos(z, p, k)
  - output: coefficients of each 2nd-order transfer function $H(z)$

$$sos = \begin{bmatrix} p_{01} & p_{11} & p_{21} & d_{01} & d_{11} & d_{21} \\ p_{02} & p_{12} & p_{22} & d_{02} & d_{12} & d_{22} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_{0L} & p_{1L} & p_{2L} & d_{0L} & d_{1L} & d_{2L} \end{bmatrix}$$

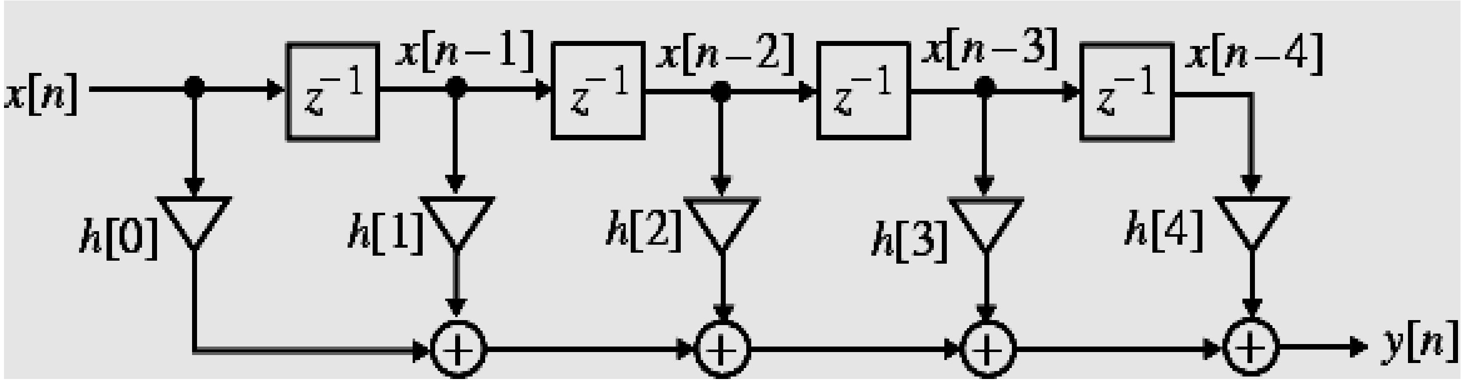
$$H(z) = \prod_{i=1}^{L} H_i(z) = \prod_{i=1}^{L} \frac{p_{0i} + p_{1i} z^{-1} + p_{2i} z^{-2}}{d_{0i} + d_{1i} z^{-1} + d_{2i} z^{-2}}$$

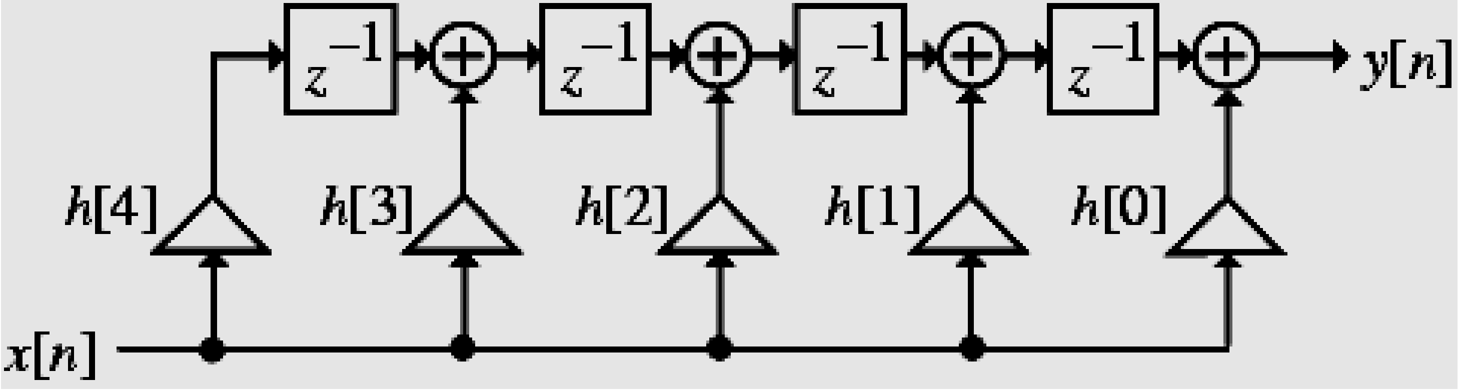
- Factorization of the transfer using residue(B,A) to get the parallel form implementation

FIR Filter Implementation

- Characterized by *N*+1 coefficients
- Require *N*+1 multipliers and *N* two-input adders

$$H(z) = \sum_{n=0}^{N} h[n] z^{-n}$$





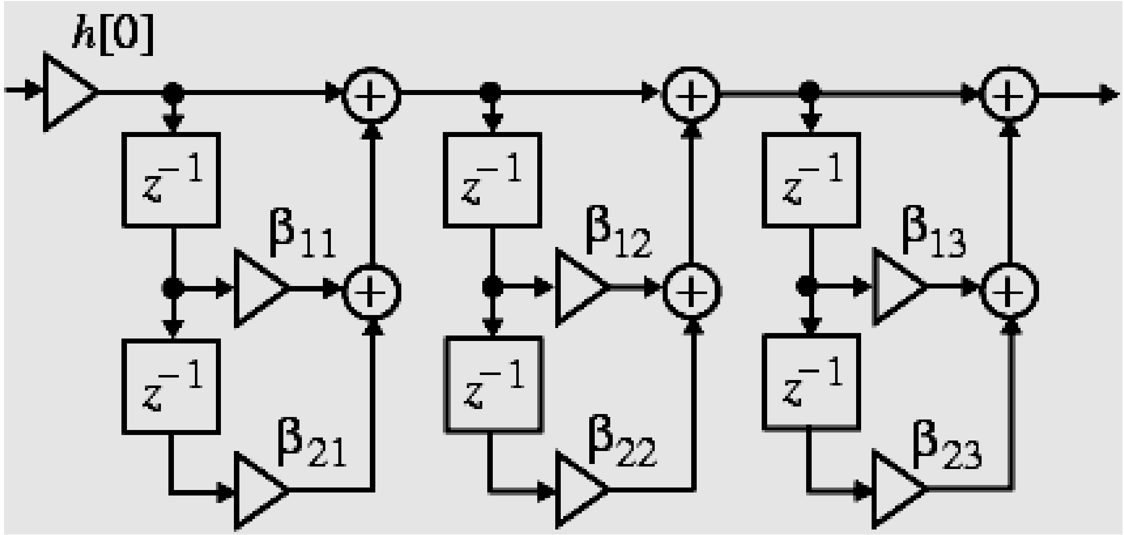
FIR Filter Implementation

FIR can also be realized as a cascade form

$$H(z) = h[0]\prod_{k=1}^{K}(1+\beta_{1k}z^{-1}+\beta_{2k}z^{-2})$$

$K = N/2$ if N is even,

$K = (N+1)/2$ if N is odd, with $\beta_{2K} = 0$



FIR Filter Implementation

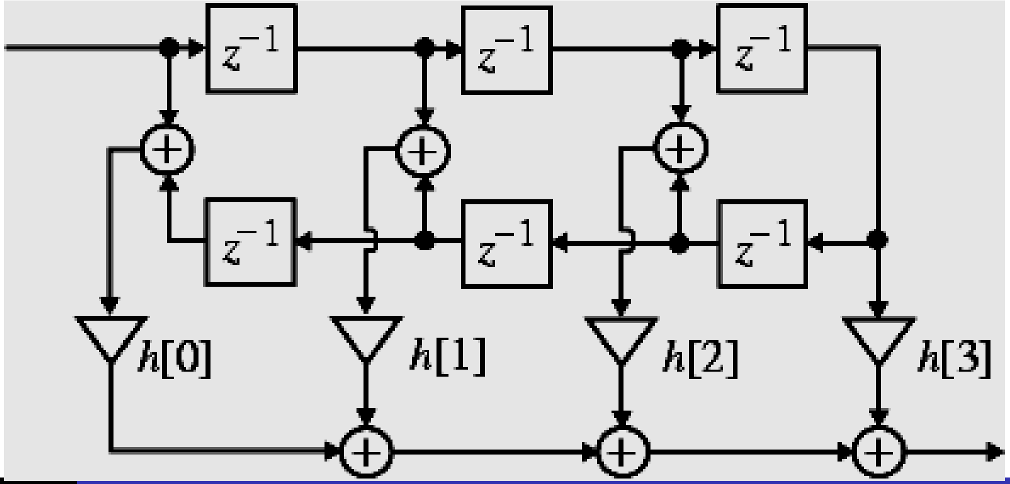
Linear Phase FIR: explore the symmetry (anti-symmetry)

Example

$$h[n] = h[M-n]$$

$$H(z) = h[0] + h[1]z^{-1} + h[2]z^{-2} + h[3]z^{-3} + h[2]z^{-4} + h[1]z^{-5} + h[0]z^{-6}$$

$$= h[0](1+z^{-6}) + h[1](z^{-1}+z^{-5}) + h[2](z^{-2}+z^{-4}) + h[3]z^{-3}$$



FIR Filter Implementation

Linear Phase FIR: explore the symmetry (anti-symmetry) property

Example

$$h[n] = h[M - n]$$

$$H[z] = h[0](1 + z^{-7}) + h[1](z^{-1} + z^{-6})$$
$$+ h[2](z^{-2} + z^{-5}) + h[3](z^{-3} + z^{-4})$$