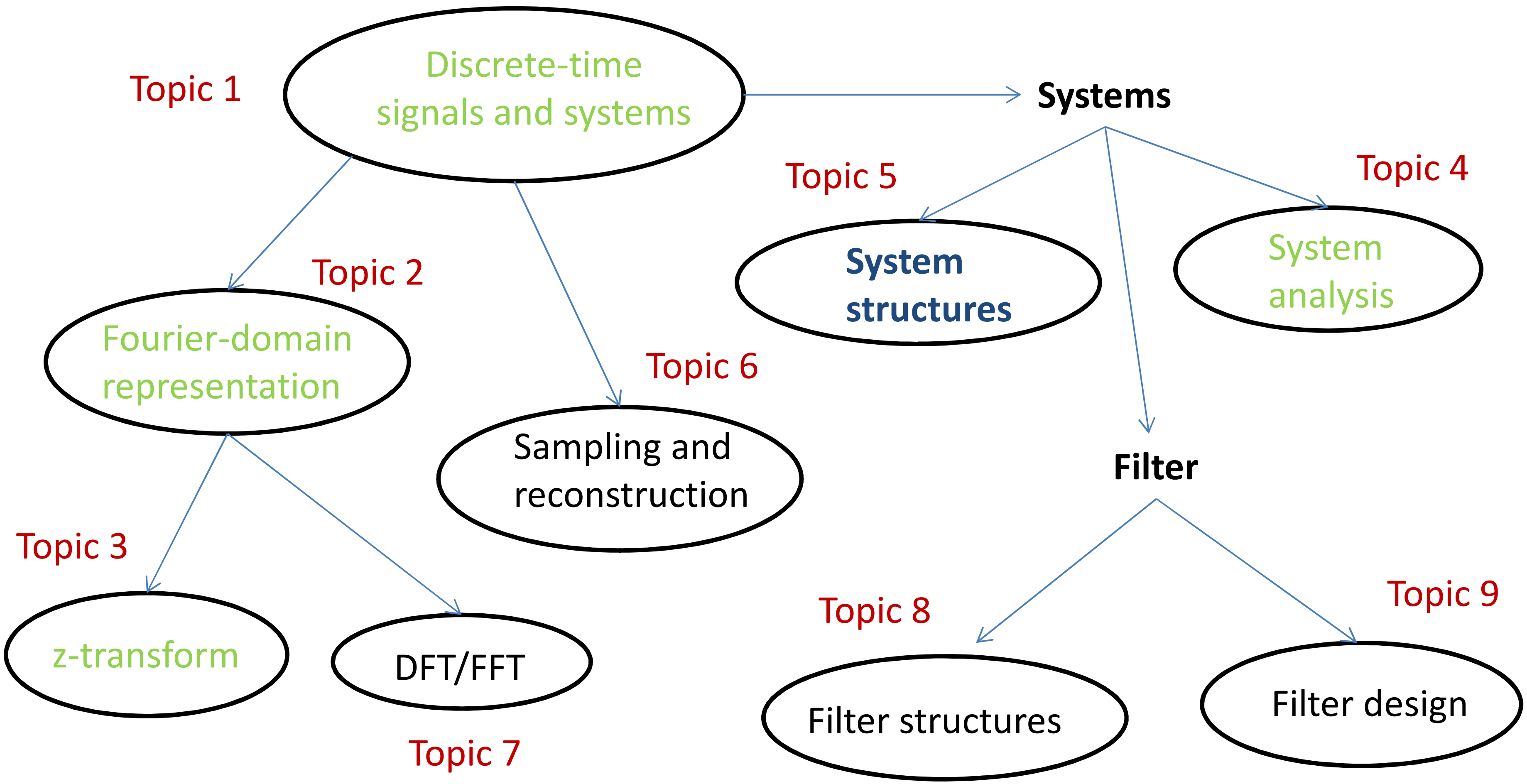
# Course at a glance



# System implementation

- LTI systems with rational system function e.g.

$$H(z) = \frac{b_0 + b_1 z^{-1}}{1 - a z^{-1}}, \quad |z| > |a|$$

- Impulse response

$$h[n] = b_0 a^n u[n] + b_1 a^{n-1} u[n-1]$$

- Linear constant-coefficient difference equation

$$y[n] - a y[n-1] = b_0 x[n] + b_1 x[n-1]$$

Three equivalent representations!

How to implement, i.e. convert to an algorithm or structure?

# System implementation

The input-output transformation $x[n] \rightarrow y[n]$ can be computed in different ways – each way is called an implementation

- An implementation is a specific description of its internal computational structure
- The choice of an implementation concerns with
  - computational requirements
  - memory requirements,
  - effects of finite-precision,
  - and so on

# System implementation

- Impulse response

$$h[n] = b_0 a^n u[n] + b_1 a^{n-1} u[n-1]$$

$$y[n] = x[n] * h[n]$$

is infinite-duration, impossible to implement in this way.

- However, linear constant-coefficient difference equation provides a means for recursive computation of the output

$$y[n] - ay[n-1] = b_0 x[n] + b_1 x[n-1]$$

$$y[n] = ay[\underline{n-1}] + \underline{b_0} x[n] \underline{+ b_1} x[n-1]$$

# Basic elements

- Implementation based on the recurrence formula derived from difference equation requires
  - adders
  - multipliers
  - memory for storing delayed sequence values

$$y[n] = ay[n-1] + b_0 x[n] + b_1 x[n-1]$$

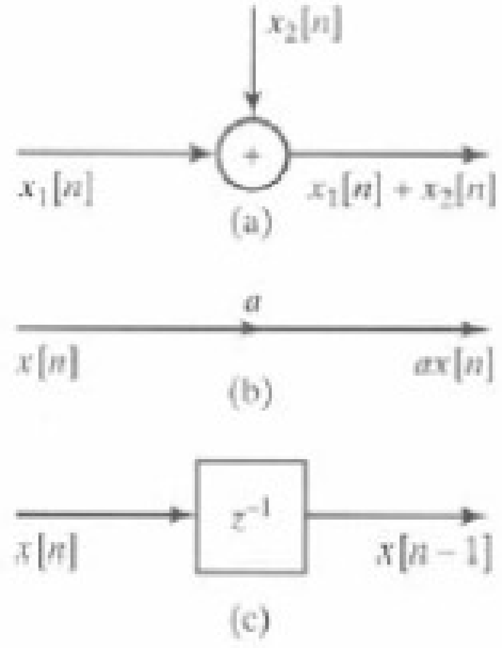



**Figure 6.1** Block diagram symbols. (a) Addition of two sequences. (b) Multiplication of a sequence by a constant. (c) Unit delay.

# Example of block diagram representation

- A second-order difference equation

$$y[n] = a_1 y[n-1] + a_2 y[n-2] + b_0 x[n]$$

$$H(z) = \frac{b_0}{1 - a_1 z^{-1} - a_2 z^{-2}}$$

Demonstrates the complexity, the steps, the amount of resources required.

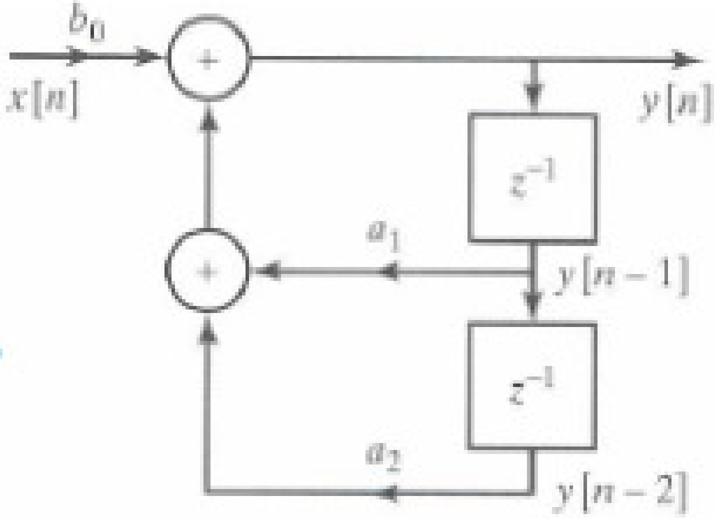
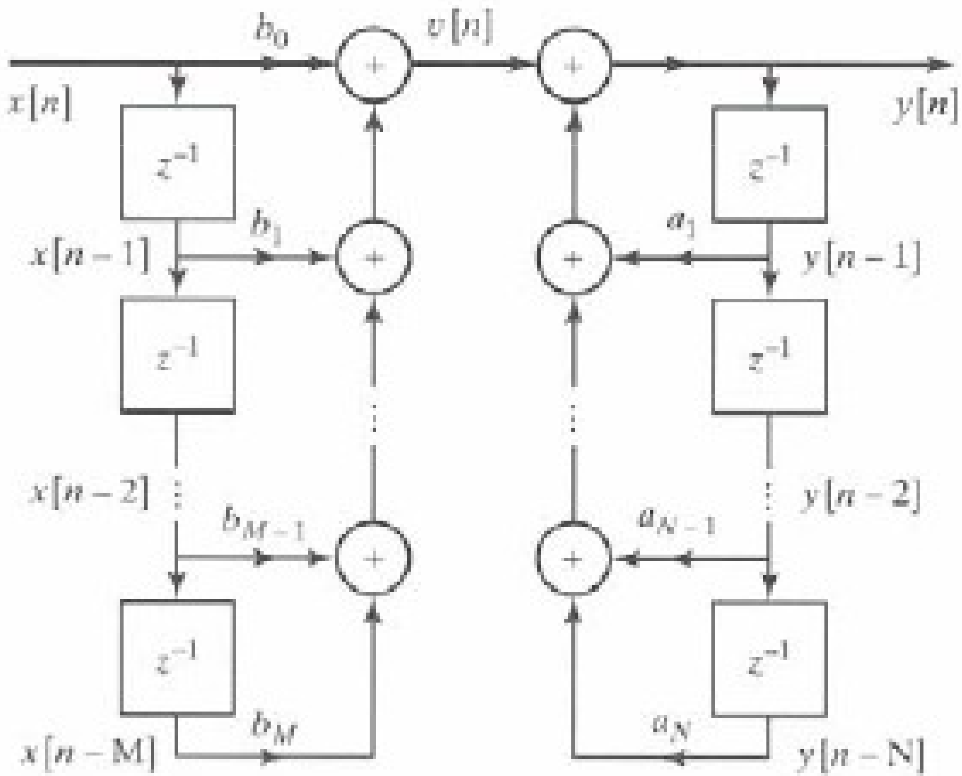



Figure 6.2 Example of a block diagram representation of a difference equation.

# General Nth-order difference equation

$$y[n]=\sum_{k=1}^{N}a_k y[n-k]+\underbrace{\sum_{k=0}^{M}b_k x[n-k]}_{v[n]}$$

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




A cascade of two systems!

X[n]→v[n], v[n]→y[n]

**Figure 6.3** Block diagram representation for a general *N*th-order difference equation.

# Rearrangement of block diagram

- A block diagram can be rearranged in many ways without changing overal function, e.g. by reversing the order of the two cascaded systems.

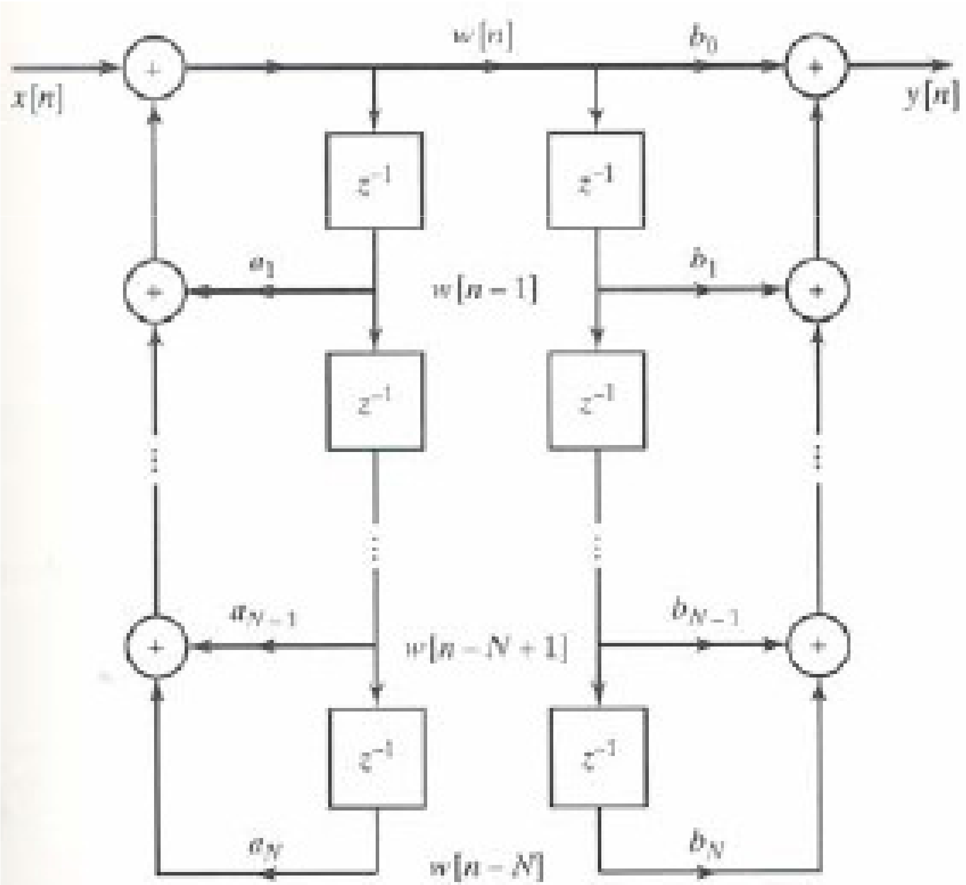



**Figure 6.4** Rearrangement of block diagram of Figure 6.3. We assume for convenience that $N = M$. If $N \neq M$, some of the coefficients will be zero.

# System function decomposition

$$v[n]$$

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

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

$$w[n]$$

# In the time domain

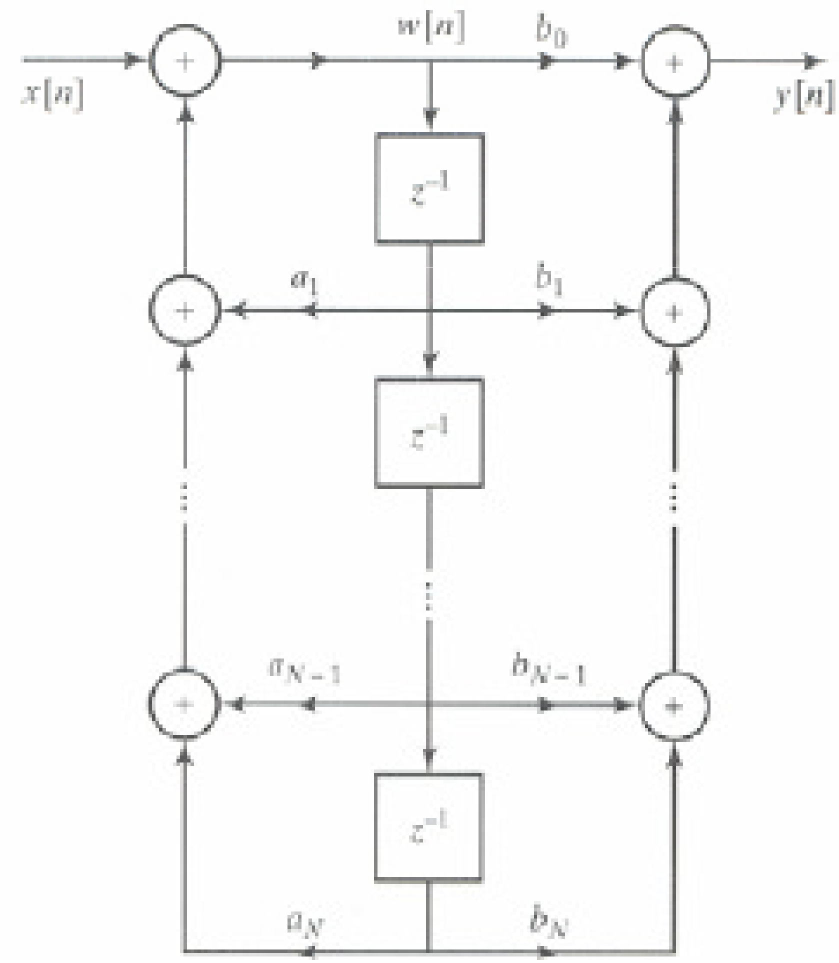
$$y[n]=\sum_{k=1}^{N}a_k y[n-k]+\sum_{k=0}^{M}b_k x[n-k]$$

$$\begin{cases} v[n]=\sum_{k=0}^{M}b_k x[n-k] \\ y[n]=\sum_{k=1}^{N}a_k y[n-k]+v[n] \end{cases}$$

$$\begin{cases} w[n]=\sum_{k=1}^{N}a_k w[n-k]+x[n] \\ y[n]=\sum_{k=0}^{M}b_k w[n-k] \end{cases}$$

# Minimum delay implementation

- One big difference btw the two implementations concerns the number of delay elements




$$N + M$$

$$\max(N, M)$$

**Figure 6.5** Combination of delays in Figure 6.4.

# Direct form I and II

- Direct form I as shown in Fig. 6.3
  - A direct realization of the difference equation
- Direct form II or canonic direct form as shown in Fig. 6.5
  - There is a direct link between the system function (difference equation) and the block diagram

# An example

- Direct form I and direct form II implementation

$$H(z)=\frac{1+2z^{-1}}{1-1.5z^{-1}+0.9z^{-2}}$$

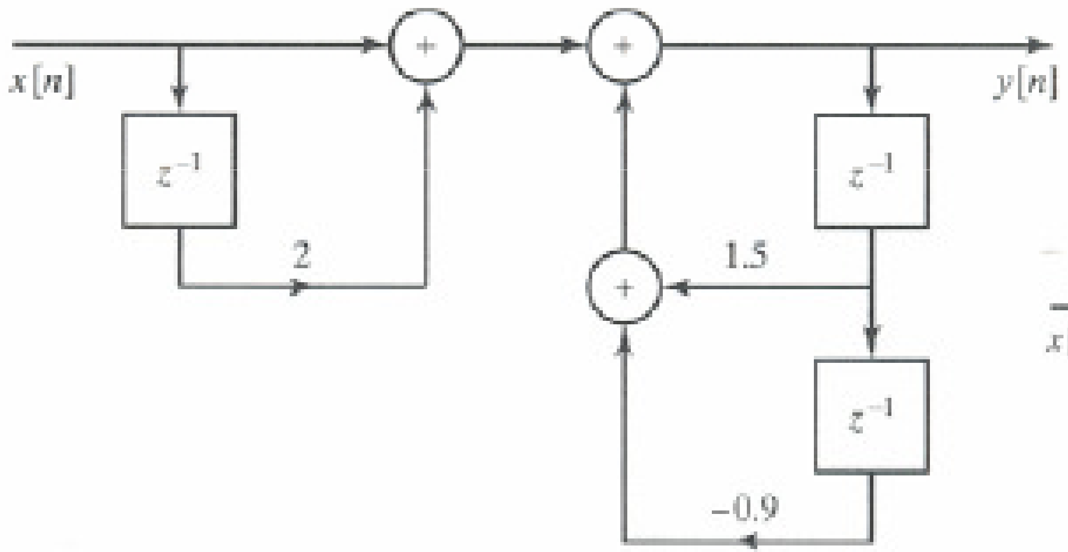



Figure 6.6 Direct form I implementation of Eq. (6.16).

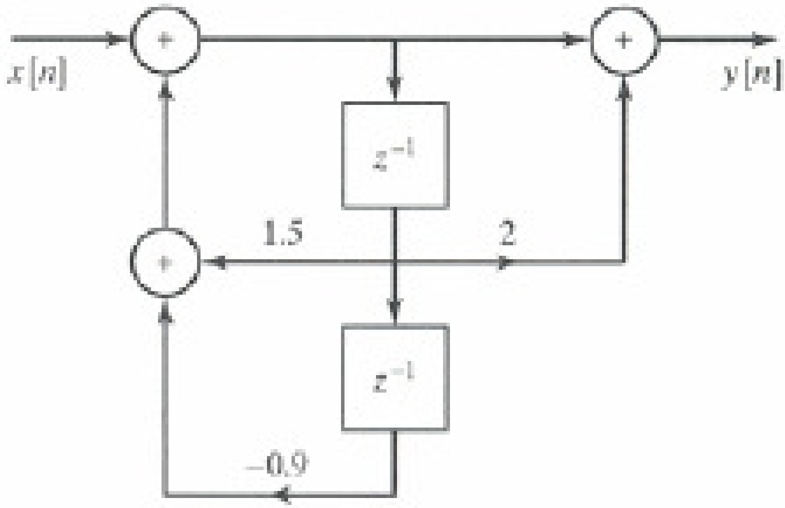



Figure 6.7 Direct form II implementation of Eq. (6.16).

# Signal flow graph (SFG)

- As an alternative to block diagrams with a few notational differences.
- A network of directed branches connecting nodes.

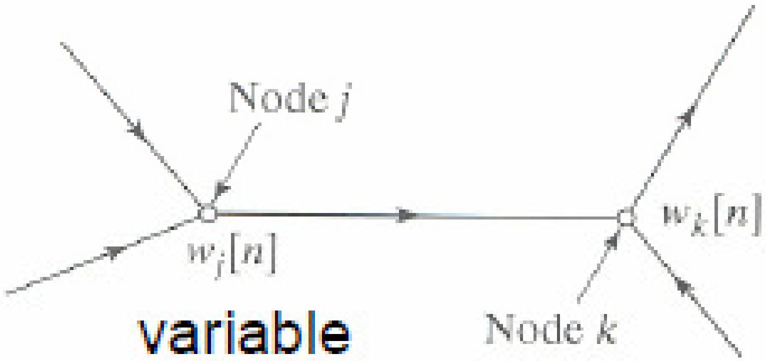



**Figure 6.8** Example of nodes and branches in a signal flow graph.

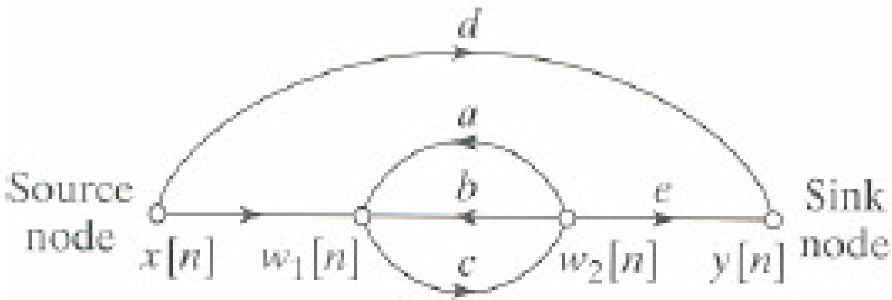
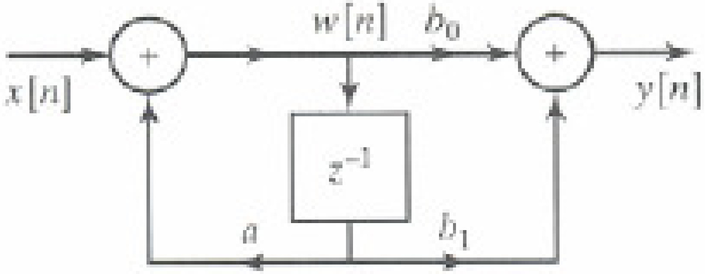
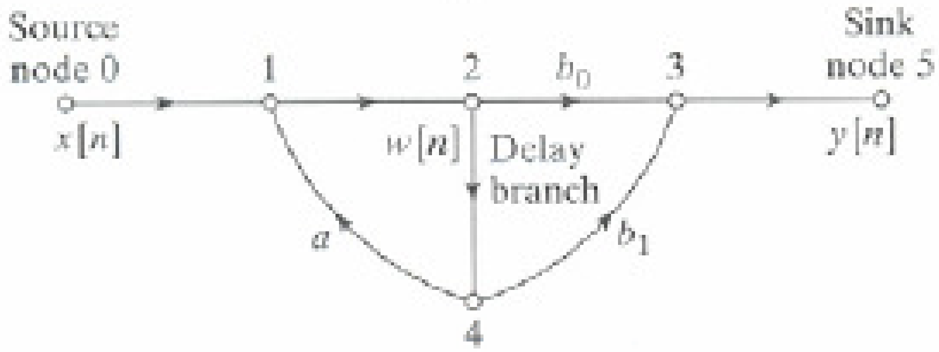


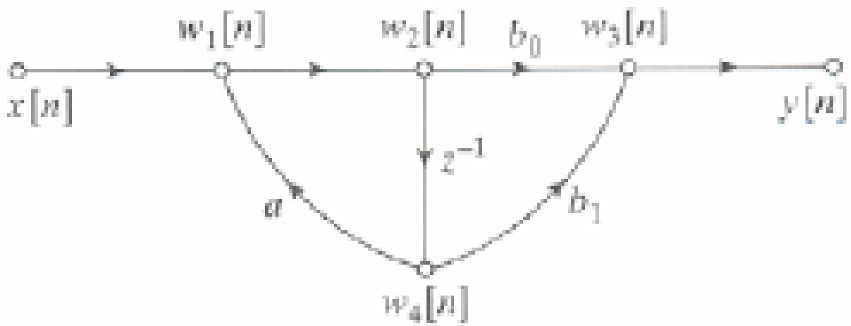

**Figure 6.9** Example of a signal flow graph showing source and sink nodes.




(a)




(b)




Nodes in SFG represent both branching points and adders (depending on the number of incoming branches), while in the diagram a special symbol is used for adders and a node has only one incoming branch.

SGF is simpler to draw.

# From flow graph to system function

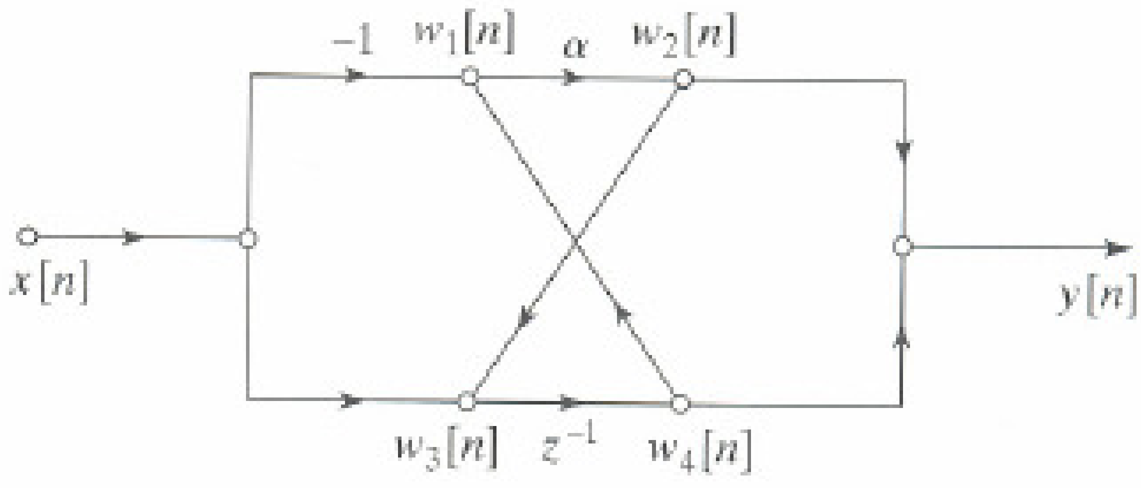



**Figure 6.12** Flow graph not in standard direct form.

$$w_1[n] = w_4[n] - x[n]$$
$$w_2[n] = \alpha w_1[n]$$
$$w_3[n] = w_2[n] + x[n]$$
$$w_4[n] = w_3[n-1]$$
$$y[n] = w_2[n] + w_4[n]$$

- Not a direct form,
  - cannot obtain H(z) by inspection.
  - But can write an equation for each node
    - $w_4[n] = w_3[n-1]$ involve feedback, difficult to solve
    - By z-transform → linear equations

# From flow graph to system function

$$\begin{cases} W_1(z) = W_4(z) - X(z) \\ W_2(z) = \alpha W_1(z) \\ W_3(z) = W_2(z) + X(z) \\ W_4(z) = z^{-1} W_3(z) \\ Y(z) = W_2(z) + W_4(z) \end{cases}$$

$$\begin{cases} W_2(z) = \alpha(W_4(z) - X(z)) \\ \\ W_4(z) = z^{-1}(W_2(z) + X(z)) \\ Y(z) = W_2(z) + W_4(z) \end{cases}$$

$$Y(z) = \left( \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}} \right) X(z)$$

$$H(z) = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}$$

If $\alpha$ is real, the system is ? All-pass

$$h[n] = \alpha^{n-1} u[n-1] - \alpha^{n+1} u[n]$$

Causal!

# From flow graph to system function

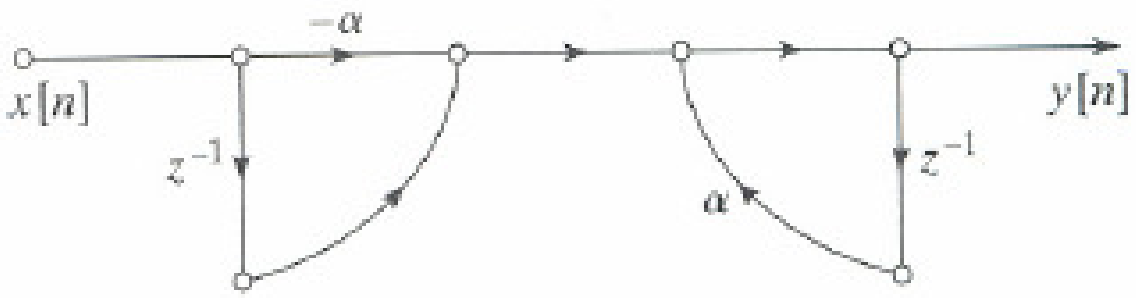



Figure 6.13 Direct form I equivalent of Figure 6.12.

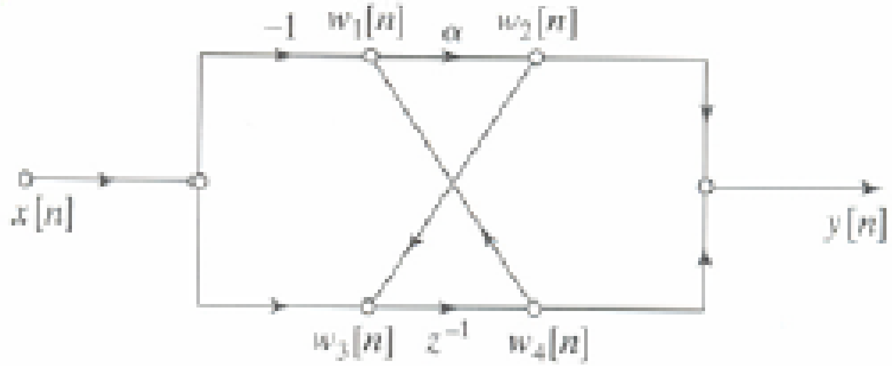



Figure 6.12 Flow graph not in standard direct form.

Different implementations, different amounts of computational resources

## Basic structures for IIR systems     Direct form I

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

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

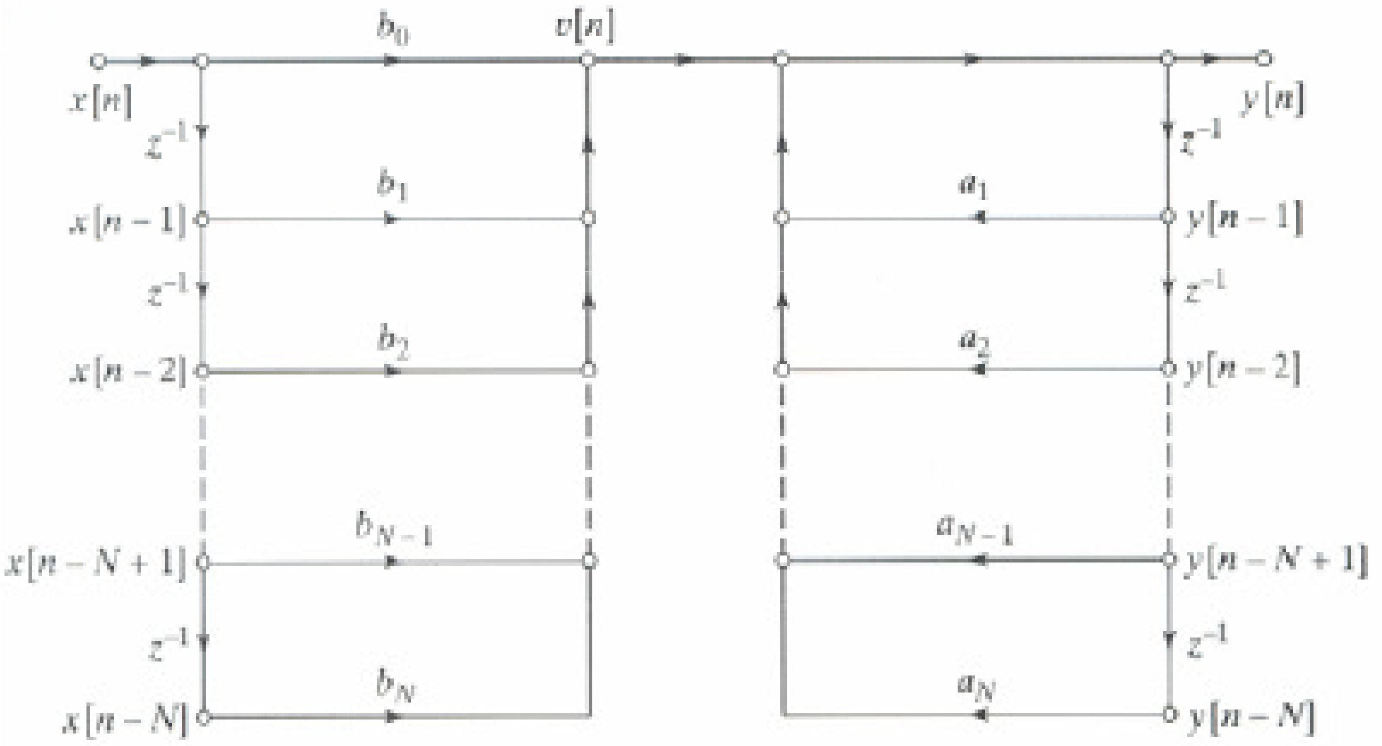



**Figure 6.14** Signal flow graph of direct form I structure for an *N*th-order system.

# Direct form II

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

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

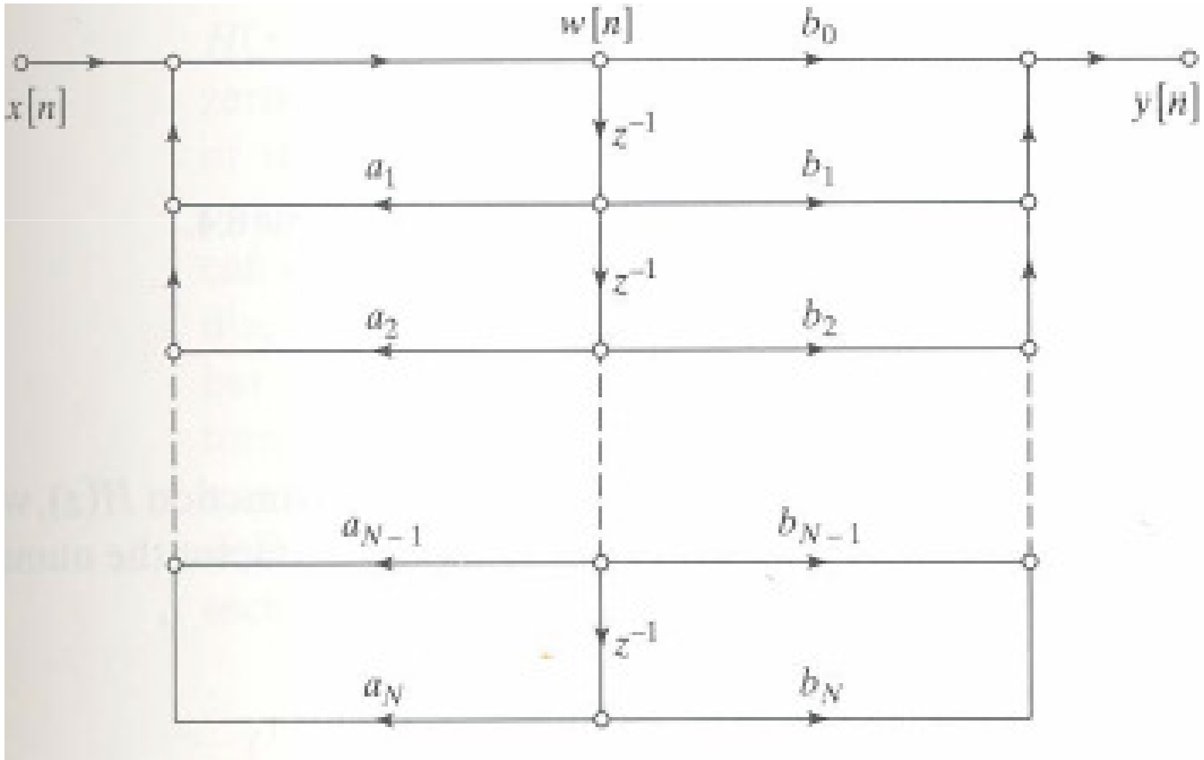



**Figure 6.15** Signal flow graph of direct form II structure for an *N*th-order system.

# Example

$$H(z)=\frac{1+2z^{-1}+z^{-2}}{1-0.752z^{-1}+0.125z^{-2}}$$

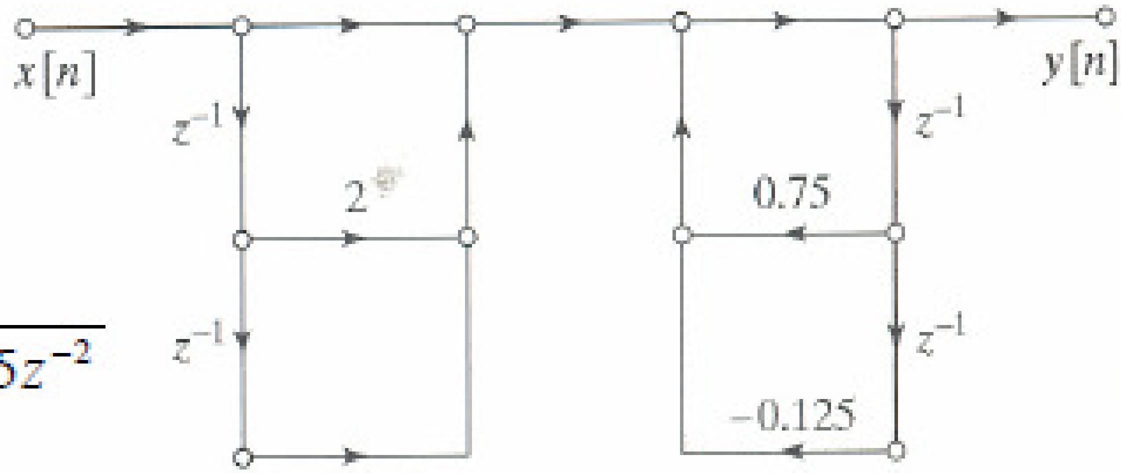



**Figure 6.16** Direct form I structure for Example 6.4.

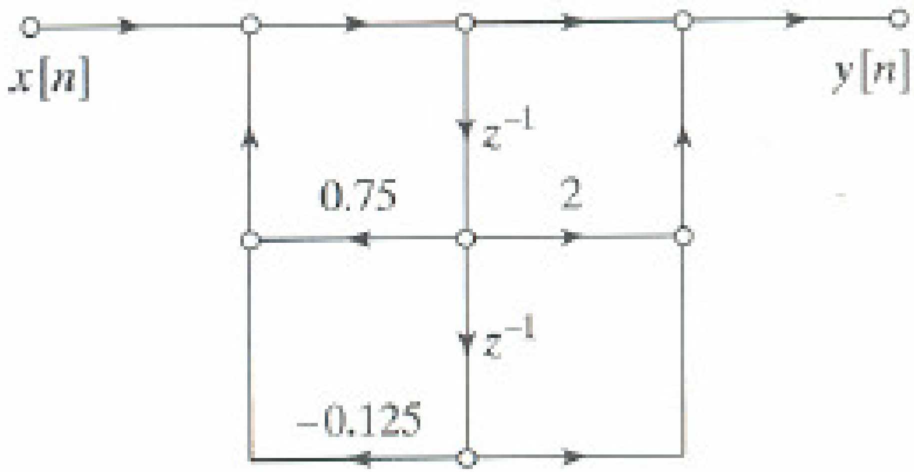



**Figure 6.17** Direct form II structure for Example 6.4.

# Cascade form

Factor the numerator and denominator polynomials

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 - \sum_{k=1}^{N} a_k z^{-k}} = A \frac{\prod_{k=1}^{M_1}(1 - f_k z^{-1}) \prod_{k=1}^{M_2}(1 - g_k z^{-1})(1 - g_k^* z^{-1})}{\prod_{k=1}^{N_1}(1 - c_k z^{-1}) \prod_{k=1}^{N_2}(1 - d_k z^{-1})(1 - d_k^* z^{-1})}$$

$$H(z) = \prod_{k=1}^{N_s} \frac{b_{0k} + b_{1k} z^{-1} + b_{2k} z^{-2}}{1 - a_{1k} z^{-1} - a_{2k} z^{-2}}$$

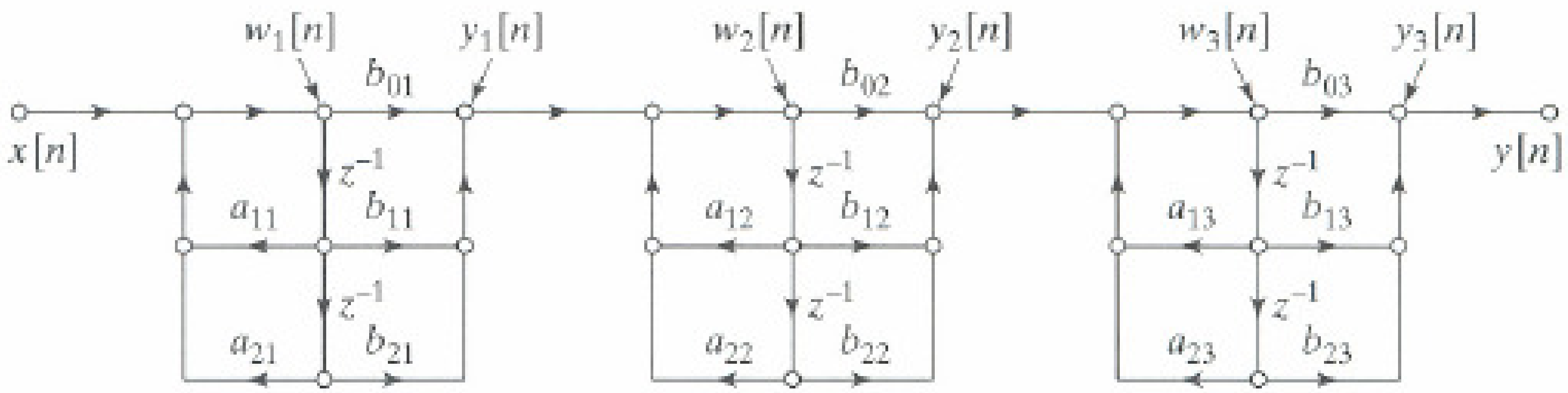
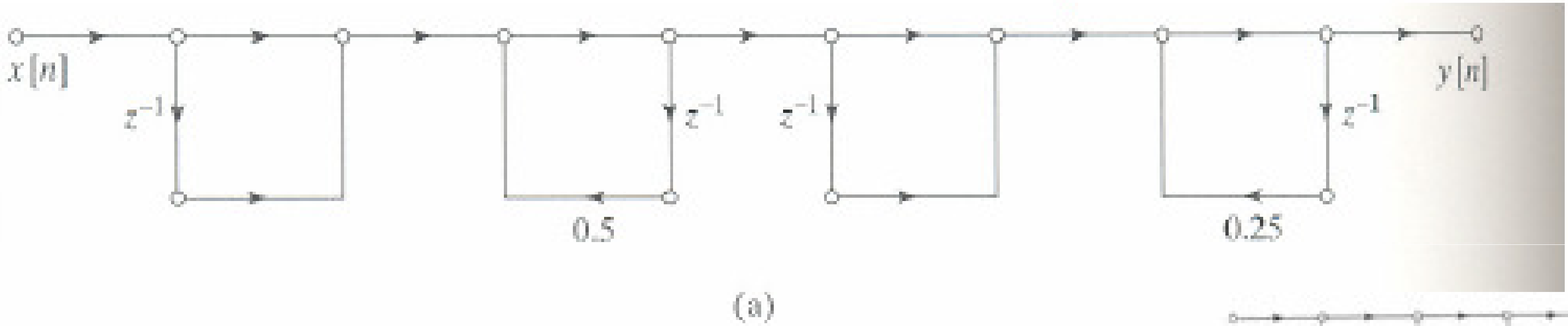



**Figure 6.18** Cascade structure for a sixth-order system with a direct form II realization of each second-order subsystem.

# An example: from 2nd-order to 1st-order cascade

$$H(z) = \frac{1+2z^{-1}+z^{-2}}{1-0.752z^{-1}+0.125z^{-2}} = \frac{(1+z^{-1})\cdot(1+z^{-1})}{(1-0.5z^{-1})\cdot(1-0.25z^{-1})}$$



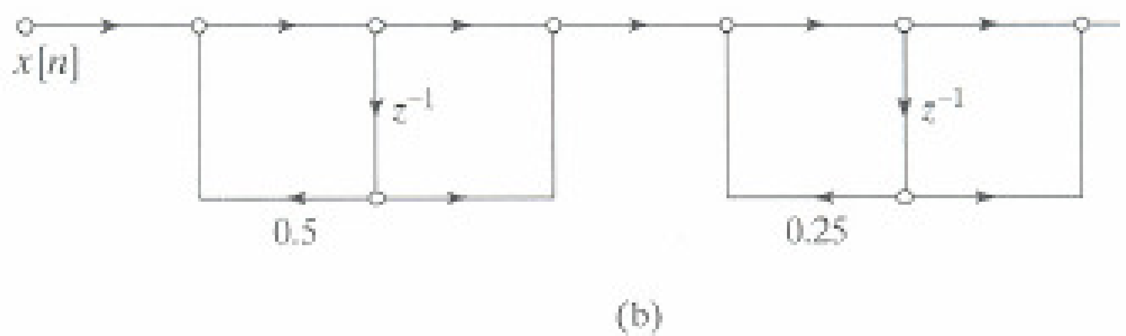




**Figure 6.19** Cascade structures for Example 6.5. (a) Direct form I (b) Direct form II subsections.

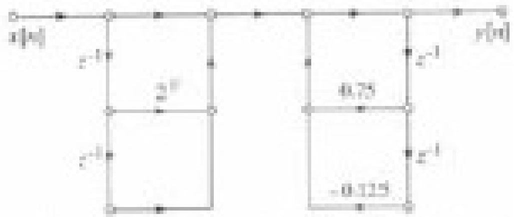



**Figure 6.16** Direct form I structure for Example 6.4.

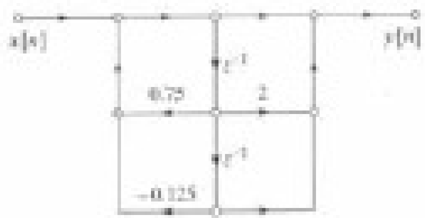



**Figure 6.17** Direct form II structure for Example 6.4.

# Parallel form by partial fraction expansion

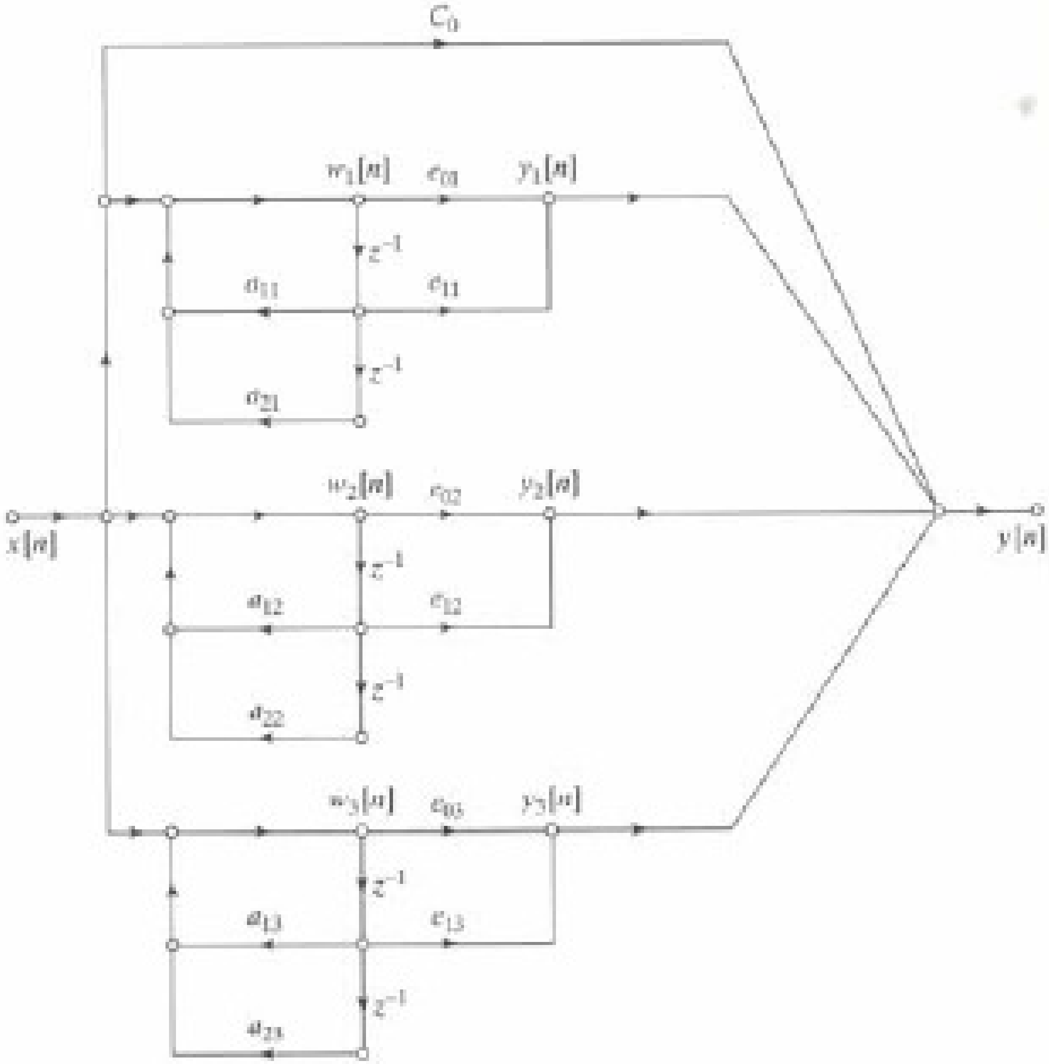
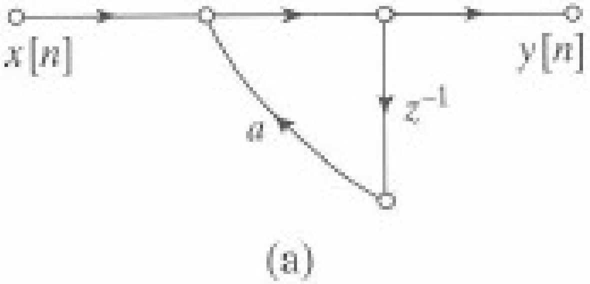


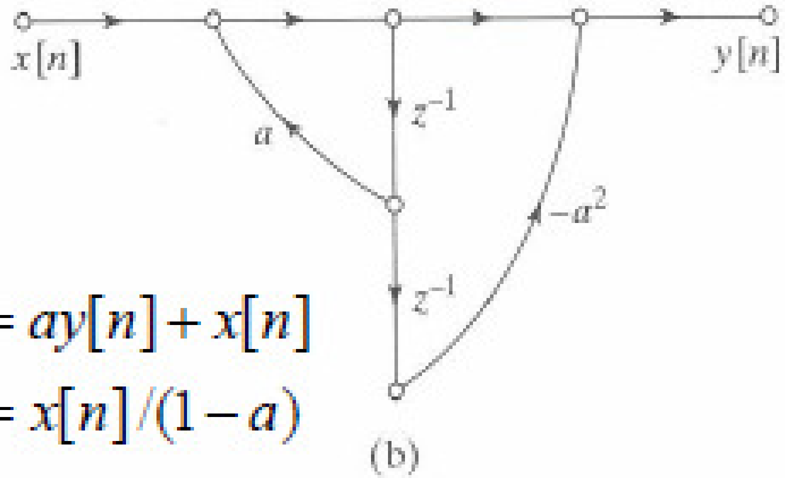

**Figure 6.20** Parallel-form structure for sixth-order system ($M = N = 6$) with the real and complex poles grouped in pairs.

$$H(z) = \sum_{k=0}^{N_p} C_k z^{-k} + \sum_{k=1}^{N_1} \frac{A_k}{1 - c_k z^{-1}} + \sum_{k=1}^{N_2} \frac{B_k(1 - e_k z^{-1})}{(1 - d_k z^{-1})(1 - d_k^* z^{-1})}$$

# Feedback in IIR systems

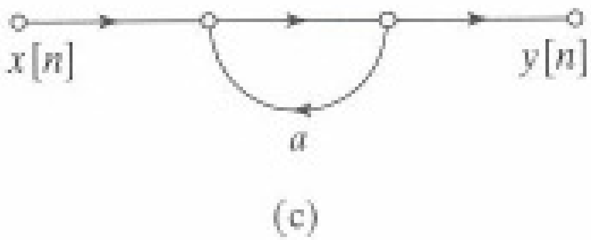



$a^n u[n]$




$y[n] = ay[n] + x[n]$

$y[n] = x[n]/(1-a)$




Feedback loop: a closed path
Necessary but not sufficient condition for IIR system
(Feedback introduced poles could be cancelled by zeros)

$$H(z) = \frac{1-a^2z^{-2}}{1-az^{-1}} = 1 + az^{-1}$$

All loops must contain at least one unit delay element

**Figure 6.23** (a) System with feedback loop. (b) FIR system with feedback loop. (c) Noncomputable system.

# Transposed form for a first-order system

Flow graph reversal or transposition also provides alternatives: reversing the directions of all branches and reversing the input and output

Resulting in same H(z)

$$H(z) = \frac{1}{1-az^{-1}}$$

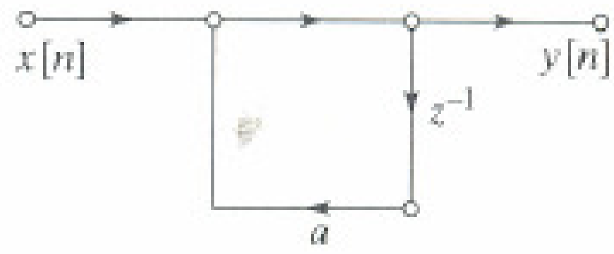



Figure 6.24 Flow graph of simple first-order system.

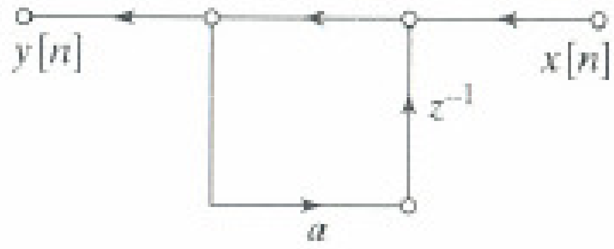



Figure 6.25 Transposed form of Figure 6.24.

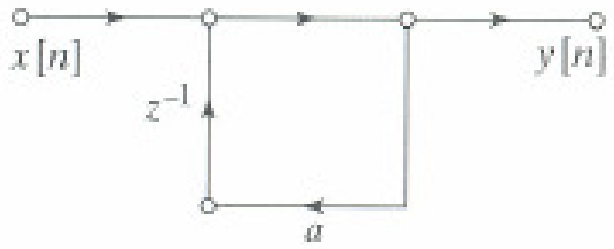



Figure 6.26 Structure of Figure 6.25 redrawn with input on left

# Transposed direct form II and direct form II

The transposed direct form II implements the zeros first and then the poles, being important effect for finite-precision existing

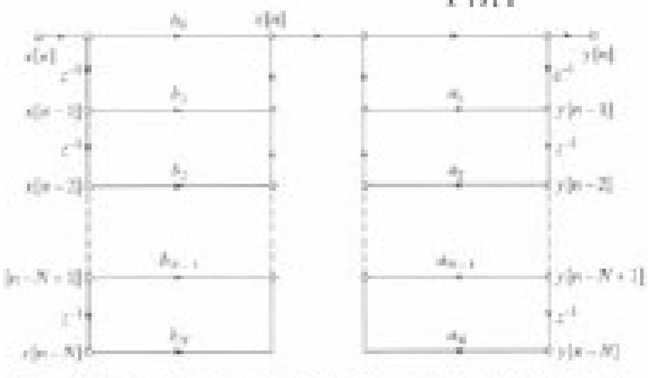
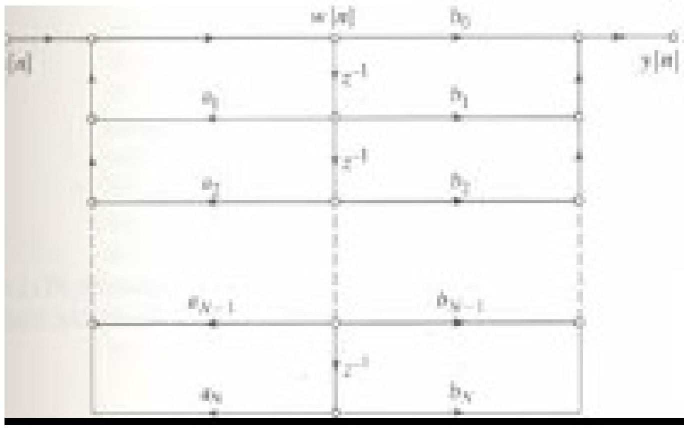
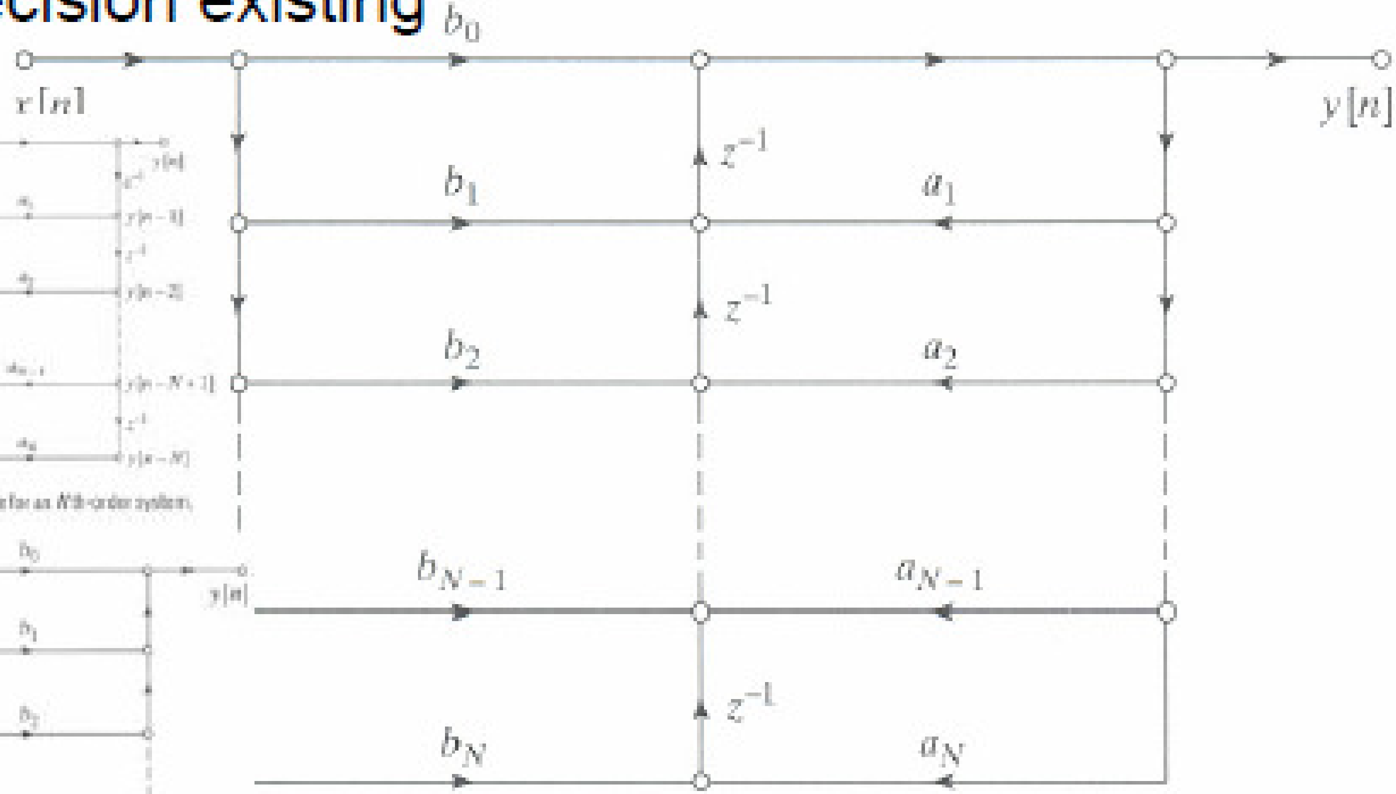


Figure 6.14 Signal flow graph of direct form I structure for an Nth-order system.



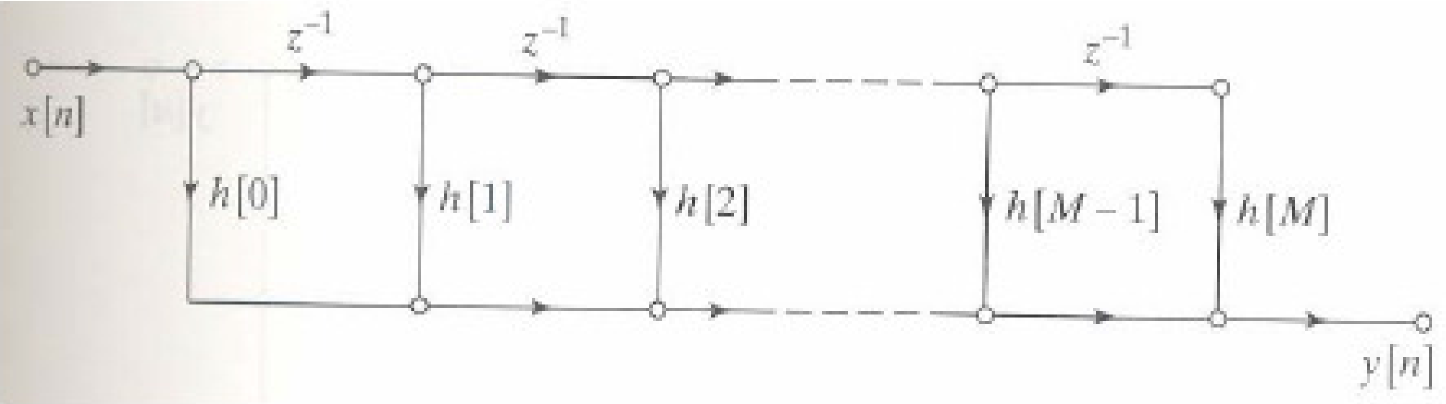


## Basic structures for FIR systems **Direct form**

- So far, system function has both poles and zeros. FIR systems as a special case.
- Causal FIR system function has only zeros (except for poles as z=0)

$$y[n] = \sum_{k=0}^{M} b_k x[n-k] \qquad h[n] = \begin{cases} b_n, & n = 0,1,\ldots,M \\ 0, & \text{otherwise} \end{cases}$$

- Form I and form II are the same.



# Cascade form

Factoring the polynomial system function

$$H(z)=\sum_{n=0}^{M}h[n]z^{-n}=\prod_{k=1}^{M_s}(b_{0k}+b_{1k}z^{-1}+b_{2k}z^{-2})$$

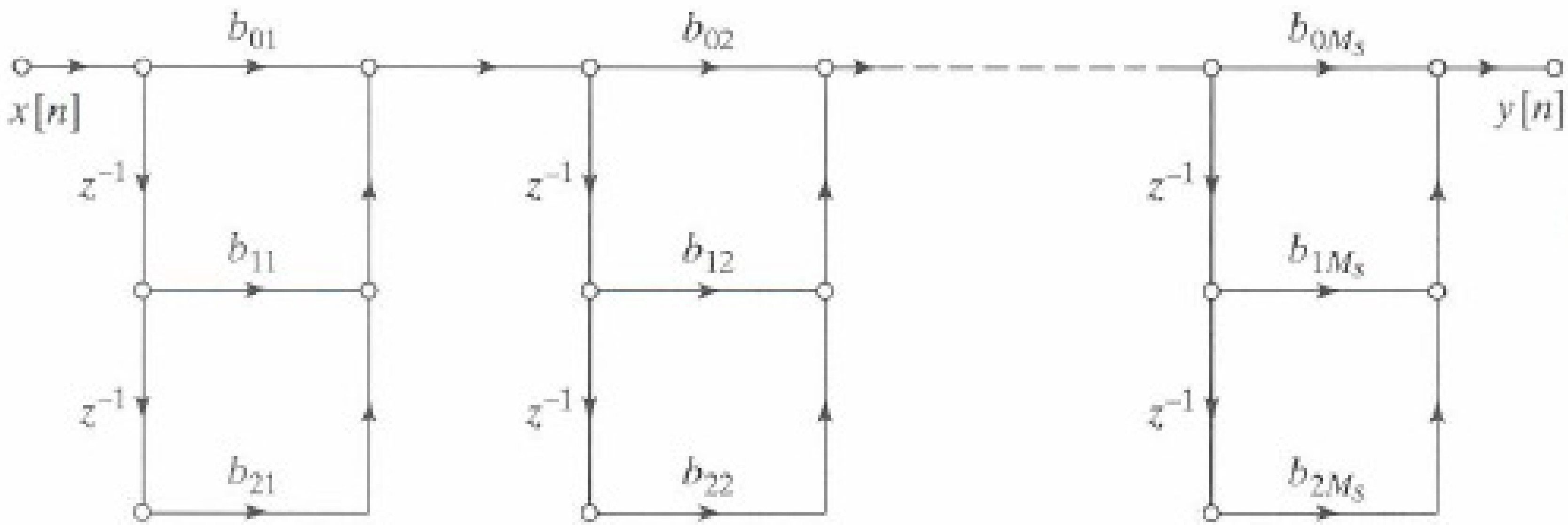



**Figure 6.33** Cascade-form realization of an FIR system.