REALIZATION OF DIGITAL FILTERS

Mifta Nur Farid

INTRODUCTION

Introduction

- Basic steps to design of digital filters
  - Approximation
  - Realization
  - Study of arithmetic errors
  - Implementation

Approximation

- The process of generating a transfer function that would satisfy the desired specifications, which may concern the amplitude or phase response or even the time-domain response of the filter.
- Can be classified as direct or indirect.

Approximation

- In direct methods, the problem is solved directly in the z domain.
- In indirect methods, a continuous-time transfer function is first obtained and then converted into a corresponding discrete-time transfer function.

ELEKTRO ITK

Institut Teknologi Kalimantan

Realization

- The process of generating a digital-filter network or structure from the transfer function or some other characterization of the filter.
- Can be classified as direct or indirect.
- In direct methods the realization is obtained directly from a given discrete-time transfer function
- In indirect realizations, the filter structure is obtained indirectly from an equivalent prototype analog filter.

Study of arithmetic errors

- During the approximation step, the coefficients of the transfer function are determined to a high degree of precision.
- In practice, however, digital hardware have finite precision that depends on the length of registers used to store numbers; the type of number system used (e.g., signed-magnitude, two's complement); the type of arithmetic used (e.g., fixed-point or floating-point), and so on.

Study of arithmetic errors

- Consequently, filter coefficients must be quantized (e.g., rounded or truncated) before they can be stored in registers.
- When the transfer function coefficients are quantized, errors are introduced in the amplitude and phase responses of the filter, which are commonly referred to as quantization errors.

Study of arithmetic errors

- Under these circumstances, the design process cannot be deemed to be complete until the effects of arithmetic errors on the performance of the filter are investigated and ways are found to mitigate any problems associated with numerical imprecision.

Implementation

- Implementation of a digital filter can assume two forms, namely, software or hardware.
- Software: implementation involves the simulation of the filter network on a general-purpose digital computer, workstation, or DSP chip.
- Hardware: it involves the conversion of the filter network into a dedicated piece of hardware.

Implementation

- In nonreal-time applications where a record of the data to be processed is available, a software implementation may be entirely satisfactory.
- In real-time applications, however, where data must be processed at a very high rate (e.g., in communication systems), a hardware implementation is mandatory.

Realization

- The natural order of the four basic design steps is as stated in the preceding discussion, namely, approximation, realization, study of imperfections, and implementation.
- However, realization is that much easier to learn than the approximation process and for this reason it will be treated first.

REALIZATION

Realization

- The most frequently used direct realization methods of this class are
  - Direct
  - Direct canonic
  - State-space
  - Lattice
  - Parallel
  - Cascade

DIRECT REALIZATION

Realization

- In indirect methods, on the other hand, a given analog-filter network is represented by the so-called wave characterization, which is normally used to represent microwave circuits and systems, and through the use of a certain transformation the analog-filter network is converted into a topologically related digital-filter network

Direct Realization

- A filter characterized by the N th-order transfer function

$$H(z) = \frac{N(z)}{D(z)} = \frac{\sum_{i=0}^{N} a_i z^{-i}}{1 + \sum_{i=1}^{N} b_i z^{-i}}$$

Direct Realization

$$\frac{Y(z)}{X(z)} = H(z) = \frac{N(z)}{D(z)} = \frac{N(z)}{1 + D'(z)}$$

$$H(z) = \frac{N(z)}{D(z)} = \frac{\sum_{i=0}^{N} a_i z^{-i}}{1 + \sum_{i=1}^{N} b_i z^{-i}}$$

$$N(z) = \sum_{i=0}^{N} a_i z^{-i}$$

$$D'(z) = \sum_{i=1}^{N} b_i z^{-i}$$

Direct Realization

$$\frac{Y(z)}{X(z)} = H(z) = \frac{N(z)}{D(z)} = \frac{N(z)}{1 + D'(z)}$$

$$N(z) = \sum_{i=0}^{N} a_i z^{-i}$$

$$D'(z) = \sum_{i=1}^{N} b_i z^{-i}$$

$$Y(z) = N(z)X(z) - D'(z)Y(z)$$

$$Y(z) = U_1(z) + U_2(z)$$

$$U_1(z) = N(z)X(z)$$

$$U_2(z) = -D'(z)Y(z)$$

Direct Realization

$$Y(z) = N(z)X(z) - D'(z)Y(z)$$

$$Y(z) = U_1(z) + U_2(z)$$

$$U_1(z) = N(z)X(z)$$

$$U_2(z) = -D'(z)Y(z)$$

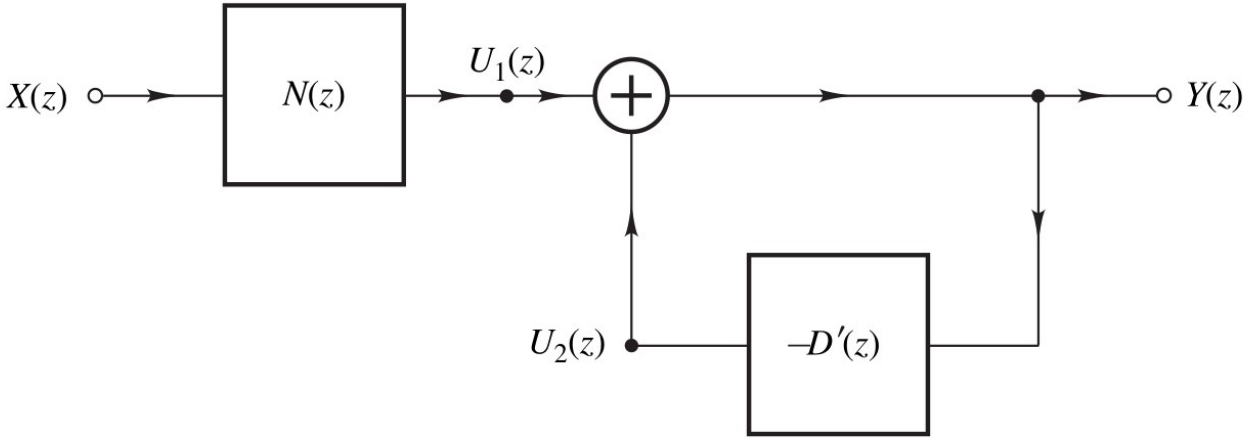


Figure 8.1 Decomposition of $H(z)$ into two simpler transfer functions.

Direct Realization

$$Y(z) = N(z)X(z) - D'(z)Y(z)$$

$$U_1(z) = [a_0 + z^{-1}N_1(z)]X(z)$$

$$Y(z) = U_1(z) + U_2(z)$$

$$N(z) = \sum_{i=0}^{N} a_i z^{-i}$$

$$N_1(z) = \sum_{i=1}^{N} a_i z^{-i+1}$$

ELEKTRO ITK

Institut Teknologi Kalimantan

Direct Realization

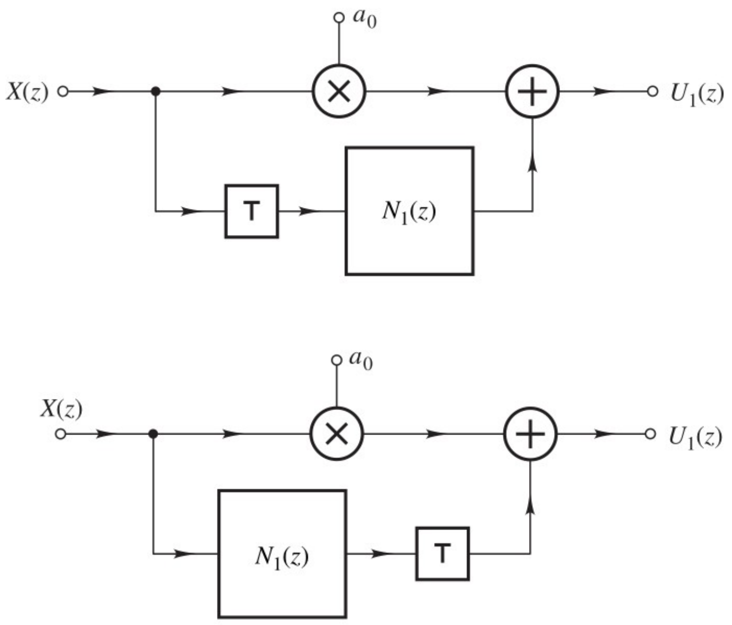


Figure 8.2 Two realizations of $N(z)$.

$$U_1(z) = [a_0 + z^{-1} N_1(z)] X(z)$$

$$N_1(z) = \sum_{i=1}^{N} a_i z^{-i+1}$$

Direct Realization

$$U_1(z) = [a_0 + z^{-1} N_1(z)] X(z)$$

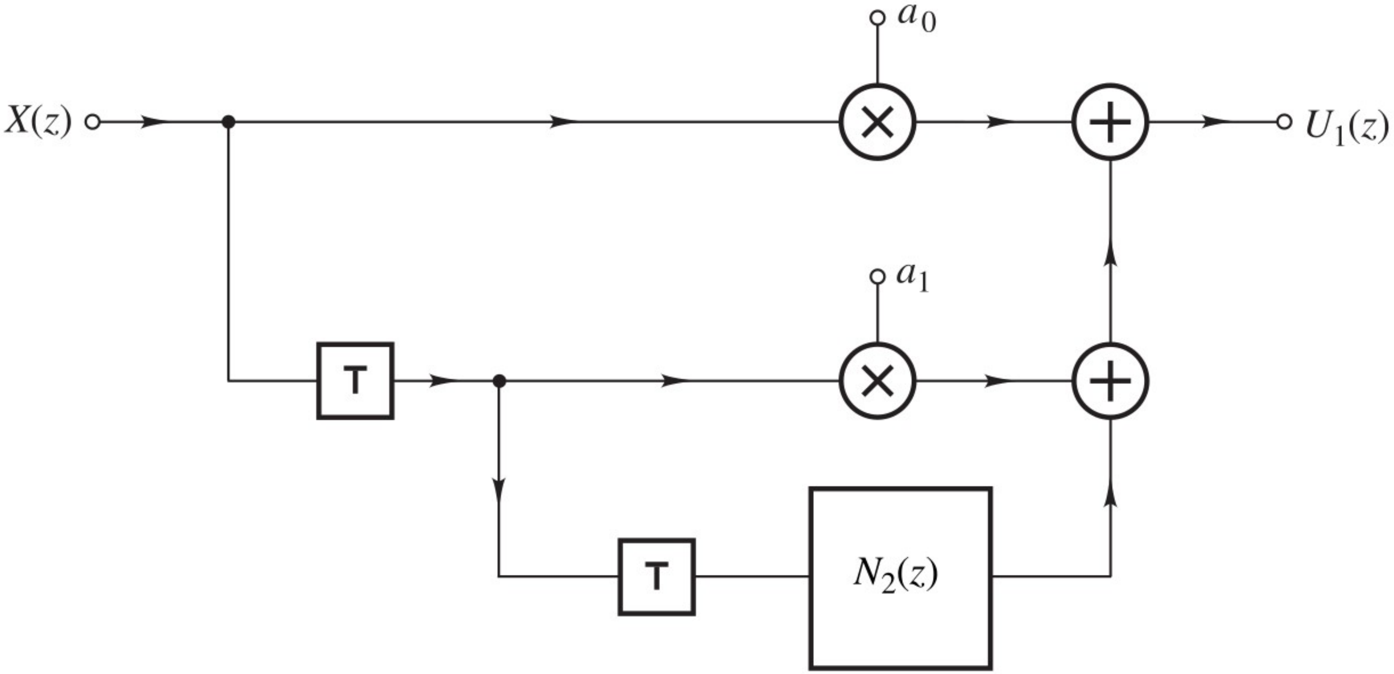
$$N_1(z) = \sum_{i=1}^{N} a_i z^{-i+1}$$

$$N_1(z) = a_1 + z^{-1} N_2(z) \qquad \text{where } N_2(z) = \sum_{i=2}^{N} a_i z^{-i+2}$$

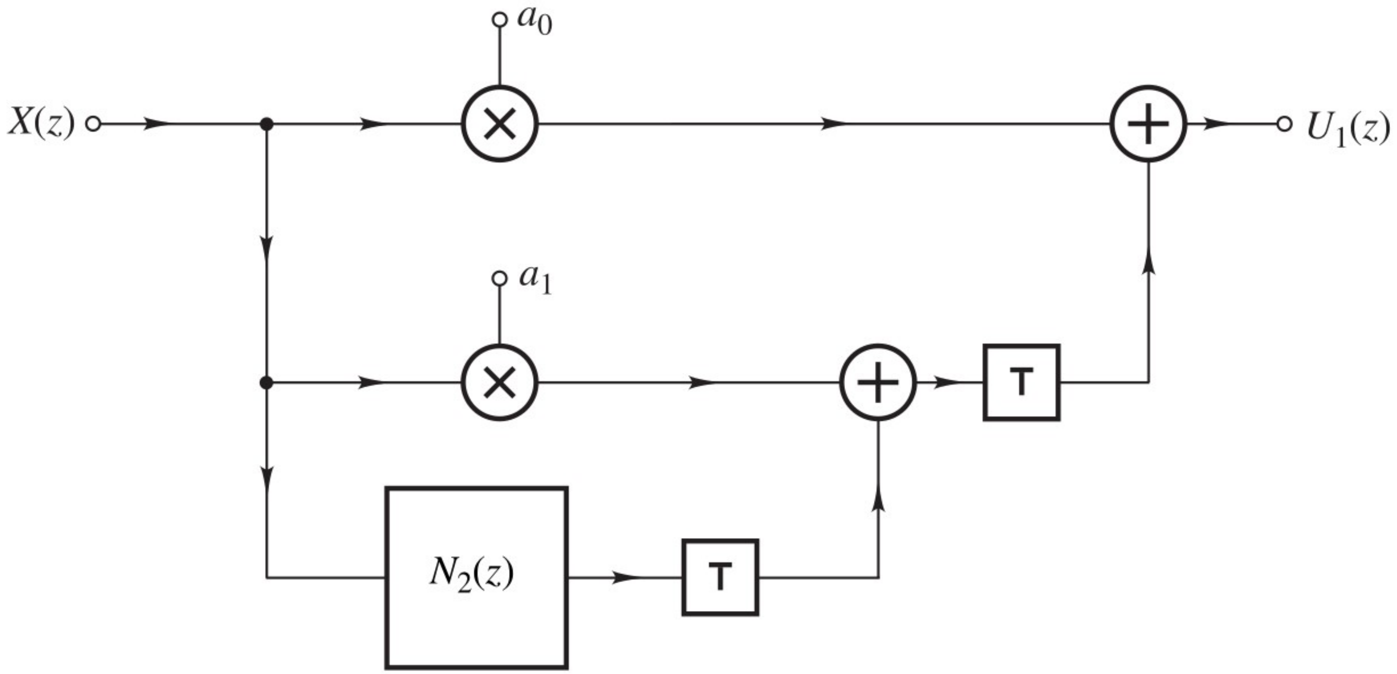
ELEKTRO ITK

Institut Teknologi Kalimantan

Direct Realization



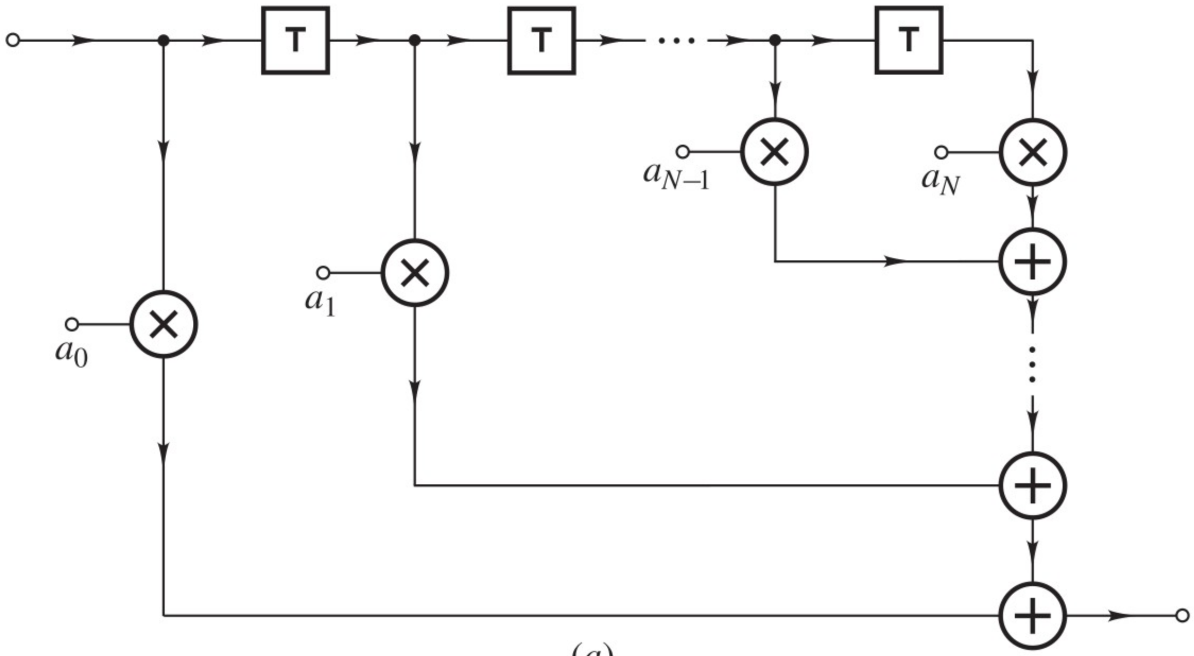
Direct Realization



Direct Realization

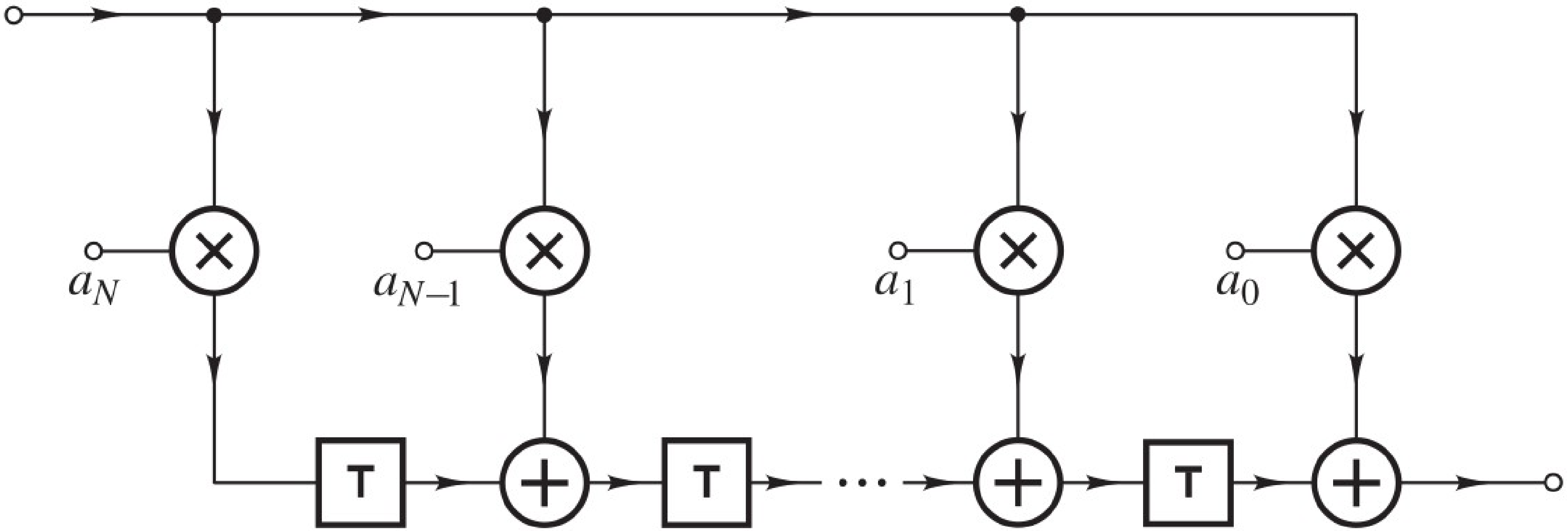
- The above cycle of activities can be repeated N times whereupon $N_N(z)$ will reduce to a single multiplier.
- In each cycle of the procedure there are two possibilities, and since there are N cycles, a total of 2^N distinct networks can be deduced for $N(z)$.

Direct Realization



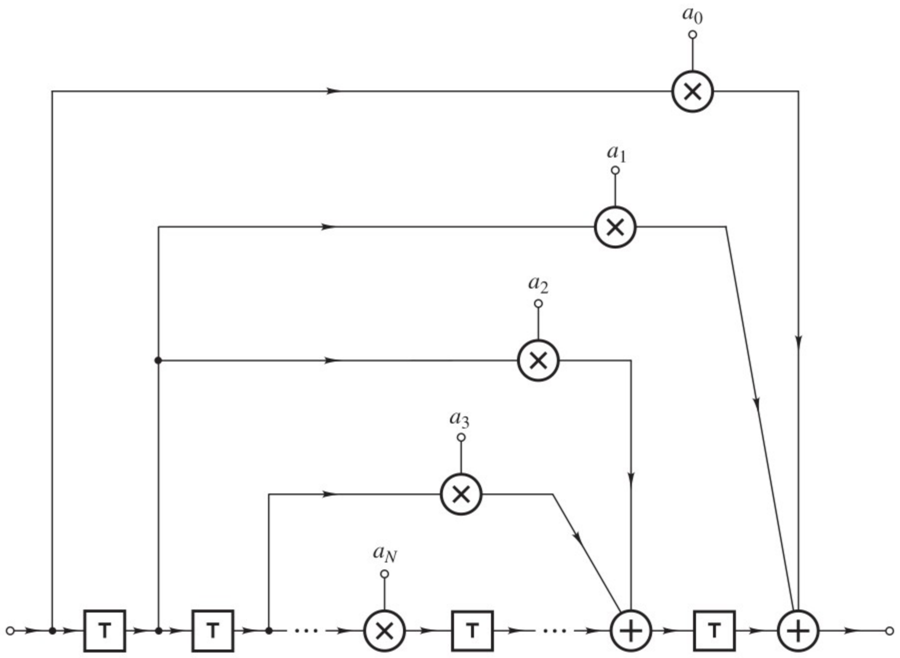
(a)

Direct Realization



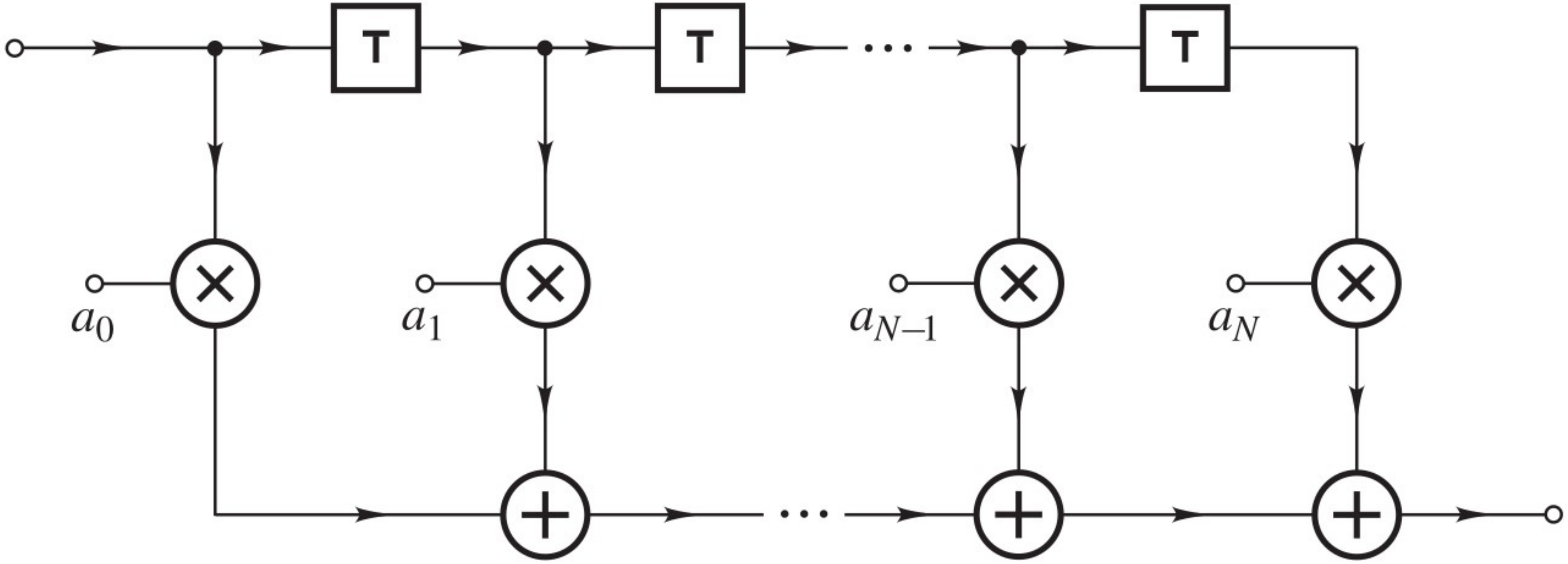
(b)

Direct Realization



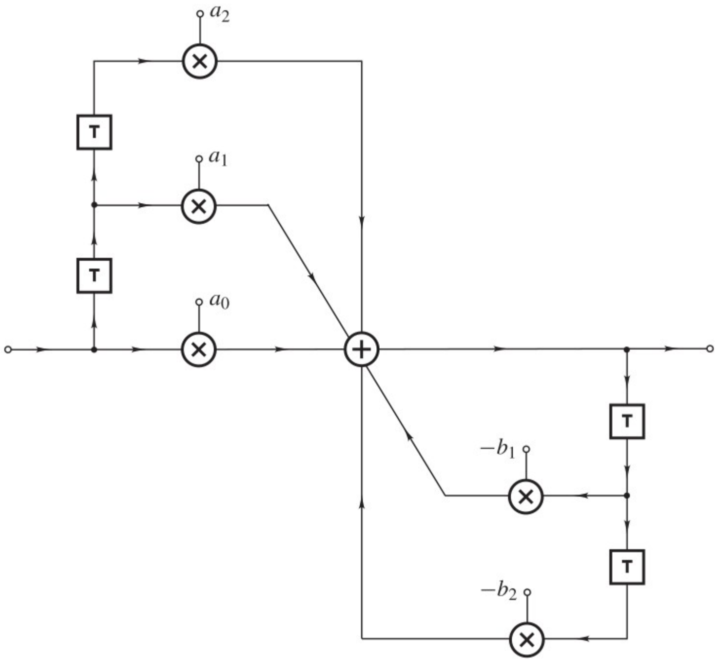
(c)

Direct Realization

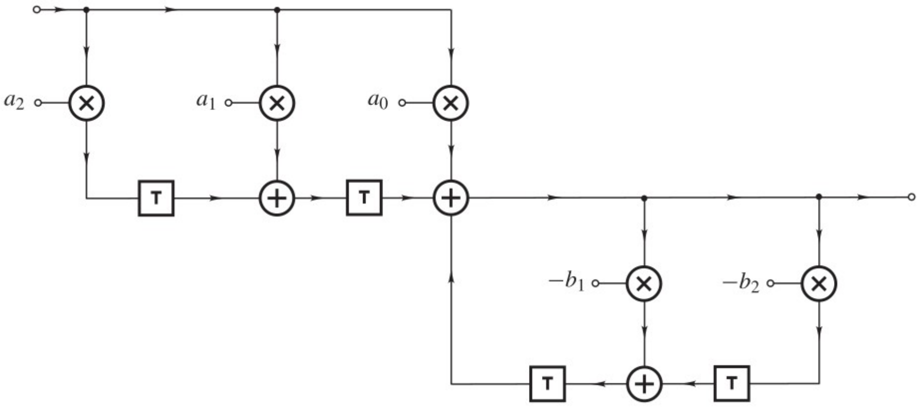


(d)

Direct Realization



(a)



(b)

Example 1

- Realize the transfer function

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}}$$

DIRECT CANONIC REALIZATION

Direct Canonic Realization

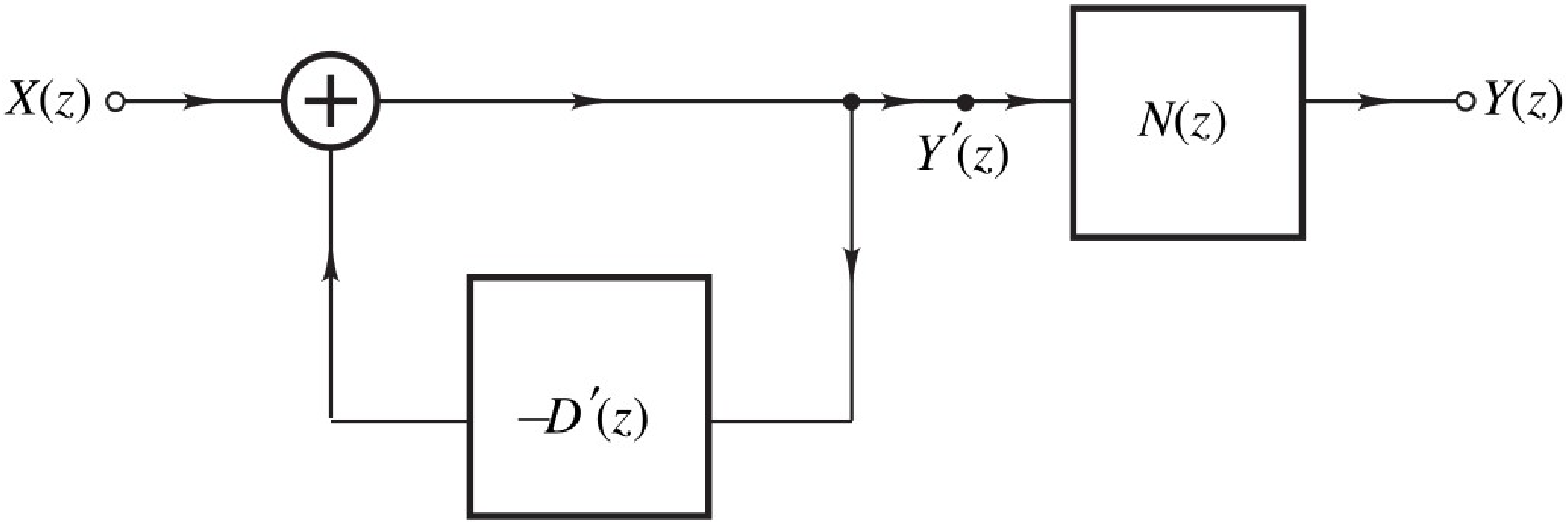
- The smallest number of unit delays required to realize an N th-order transfer function is N.
- An N th-order discrete-time network that employs just N unit delays is said to be canonic with respect to the number of unit delays.

Direct Canonic Realization

$$Y(z) = N(z)Y'(z)$$

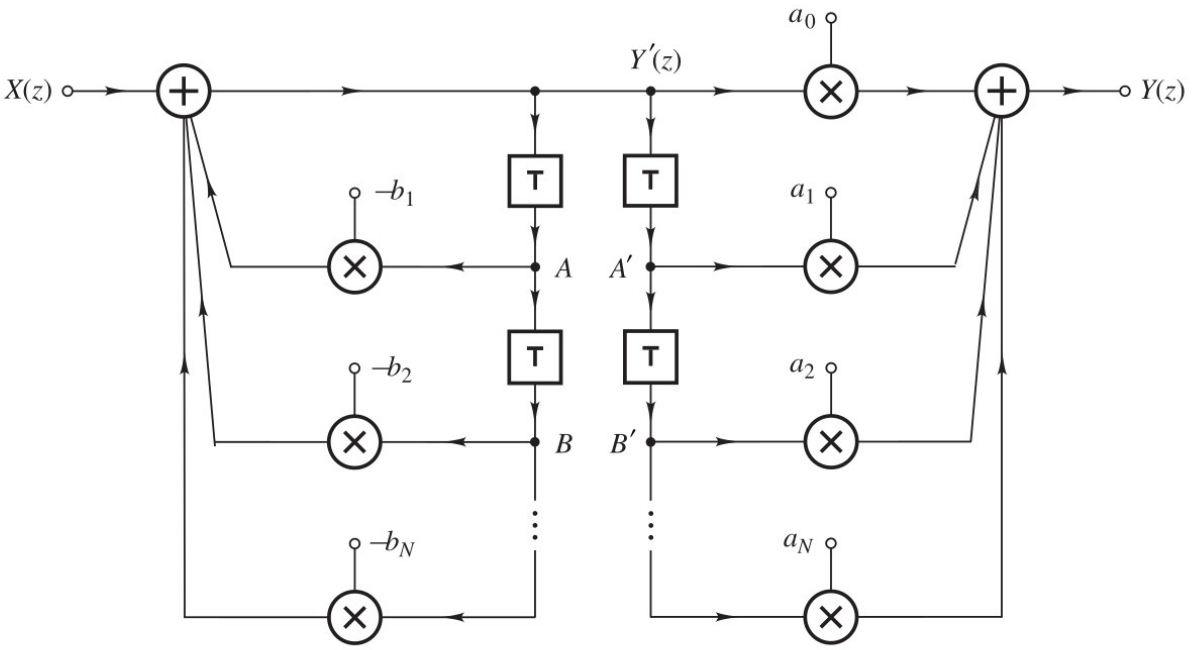
$$Y'(z) = \frac{X(z)}{1 + D'(z)} \qquad \text{or} \qquad Y'(z) = X(z) - D'(z)Y'(z)$$

Direct Canonic Realization



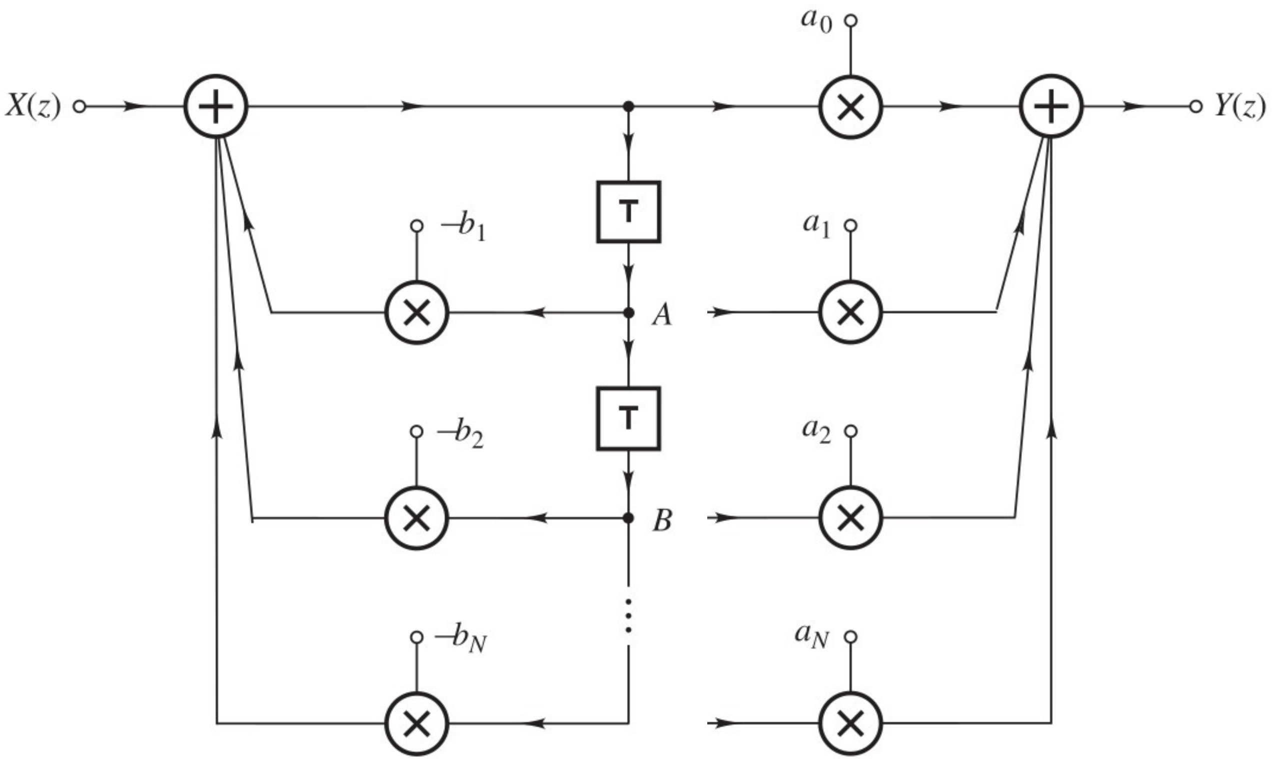
(a)

Direct Canonic Realization



(b)

Direct Canonic Realization



STATE-SPACE REALIZATION

State-Space Realization

- Another approach to the realization of digital filters is to start with the state-space characterization

$$\mathbf{q}(nT + T) = \mathbf{A}\mathbf{q}(nT) + \mathbf{b}x(nT)$$

$$y(nT) = \mathbf{c}^T\mathbf{q}(nT) + dx(nT)$$

State-Space Realization

- Another approach to the realization of digital filters is to start with the state-space characterization

$$\mathbf{q}(nT+T) = \mathbf{A}\mathbf{q}(nT) + \mathbf{b}x(nT)$$

$$y(nT) = \mathbf{c}^T\mathbf{q}(nT) + dx(nT)$$

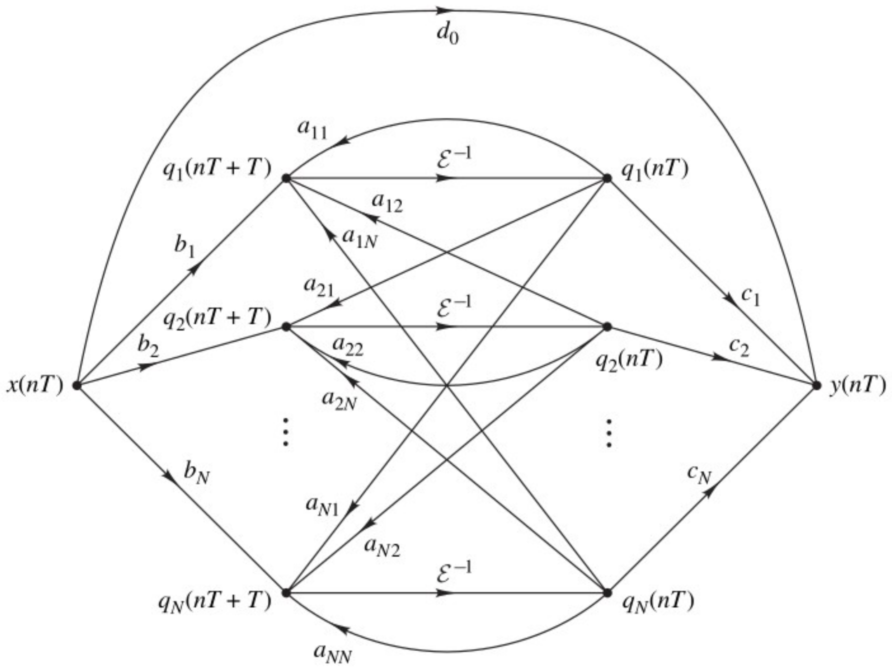
State-Space Realization

- For an N th-order filter

$$q_i(nT + T) = \sum_{j=1}^{N} a_{ij} q_j(nT) + b_i x(nT) \qquad \text{for } i = 1, 2, \ldots, N$$

$$y(nT) = \sum_{j=1}^{N} c_j q_j(nT) + d_0 x(nT)$$

State-Space Realization



- state-space signal flow graph

Example 2

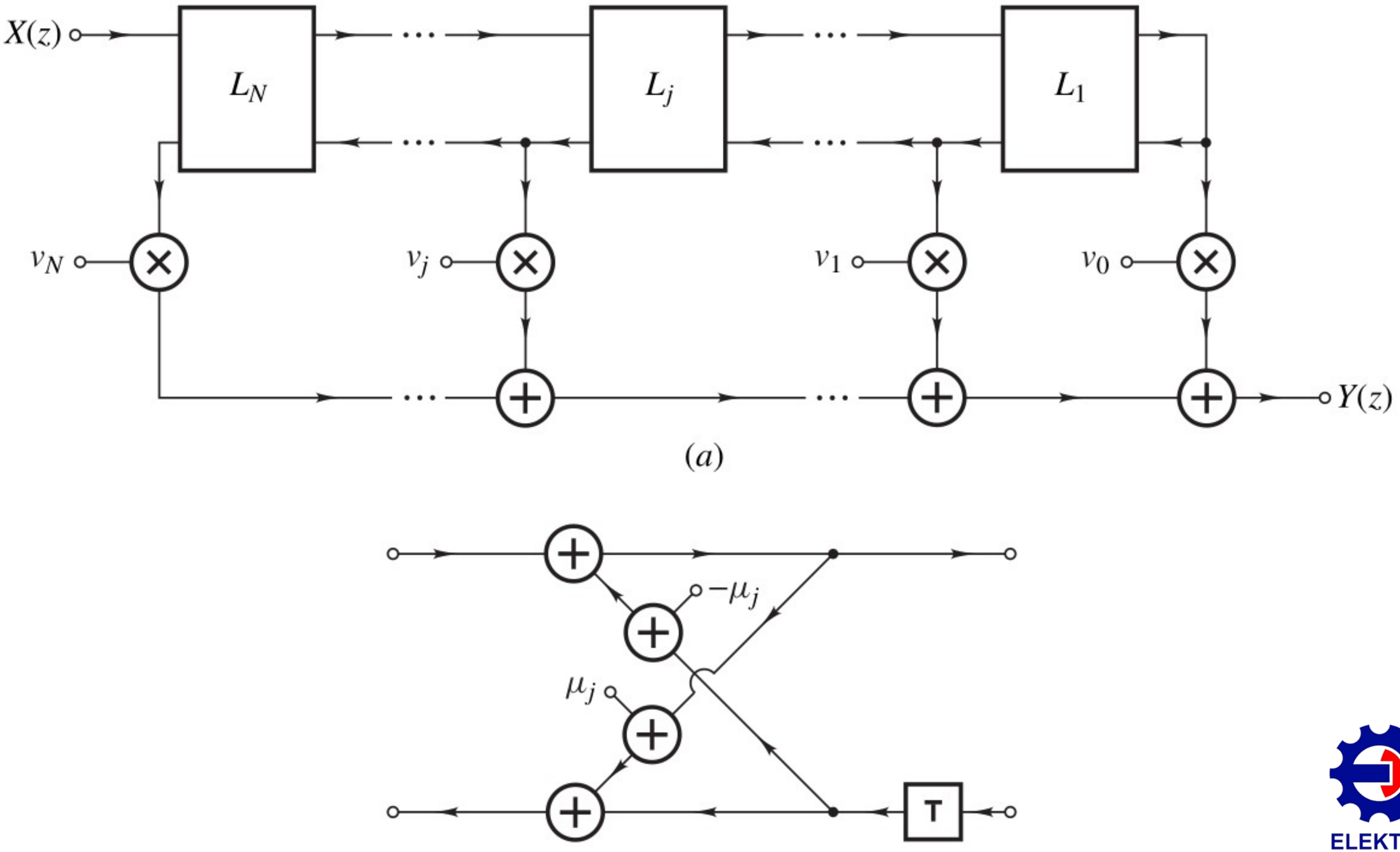
- A digital filter is characterized by the state-space equations

$$\mathbf{A} = \begin{bmatrix} -\frac{1}{2} & -\frac{1}{3} & -\frac{1}{4} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} \qquad \mathbf{c}^T = \begin{bmatrix} -\frac{1}{4} & \frac{1}{6} & \frac{1}{12} \end{bmatrix} \qquad d = 2$$

- Obtain a direct canonic realization.

LATTICE REALIZATION

Lattice Realization



Lattice Realization

- The realization can be accomplished by using a recursive algorithm comprising *N* iterations whereby polynomials of the form

$$N_j(z) = \sum_{i=0}^{j} \alpha_{ji} z^{-i} \qquad D_j(z) = \sum_{i=0}^{j} \beta_{ji} z^{-i}$$

are generated for $j = N, N-1, \ldots, 0$, and for each value of j the multiplier constants ν_j and μ_j are evaluated using coefficients α_{jj} and β_{jj} in the above polynomials

Lattice Realization

Step 1: Let $N_j(z) = N(z)$ and $D_j(z) = D(z)$ and assume that $j = N$, that is

$$N_N(z) = \sum_{i=0}^{j} \alpha_{ji} z^{-i} = \sum_{i=0}^{N} a_i z^{-i}$$

$$D_N(z) = \sum_{i=0}^{j} \beta_{ji} z^{-i} = \sum_{i=0}^{N} b_i z^{-i} \qquad \text{with } b_0 = 1$$

Lattice Realization

Step 2: Obtain ν_j, μ_j, $N_{j-1}(z)$, and $D_{j-1}(z)$ for $j = N, N-1, \ldots, 2$ using the following recursive relations:

$$\nu_j = \alpha_{jj} \qquad \mu_j = \beta_{jj}$$

$$P_j(z) = D_j\left(\frac{1}{z}\right) z^{-j} = \sum_{i=0}^{j} \beta_{ji} z^{i-j}$$

$$N_{j-1}(z) = N_j(z) - \nu_j P_j(z) = \sum_{i=0}^{j-1} \alpha_{ji} z^{-i}$$

$$D_{j-1}(z) = \frac{D_j(z) - \mu_j P_j(z)}{1 - \mu_j^2} = \sum_{i=0}^{j-1} \beta_{ji} z^{-i}$$

Lattice Realization

Step 3: Let $j = 1$ in Eqs. (8.8a)–(8.8d) and obtain ν_1, μ_1, and $N_0(z)$ as follows:

$$\nu_1 = \alpha_{11} \qquad \mu_1 = \beta_{11}$$

$$P_1(z) = D_1\left(\frac{1}{z}\right)z^{-1} = \beta_{10}z^{-1} + \beta_{11}$$

$$N_0(z) = N_1(z) - \nu_1 P_1(z) = \alpha_{00}$$

Lattice Realization

Step 4: Complete the realization by letting

$$\nu_0 = \alpha_{00}$$

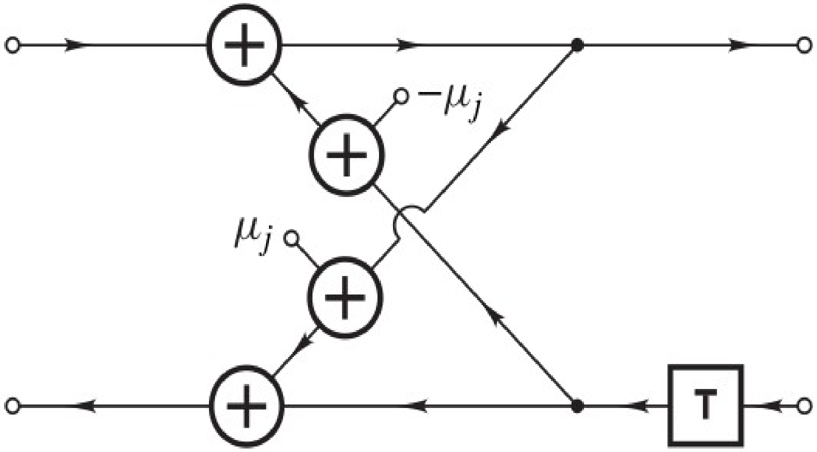
Example 3

- Realize the transfer function’

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}}$$

  using the lattice method

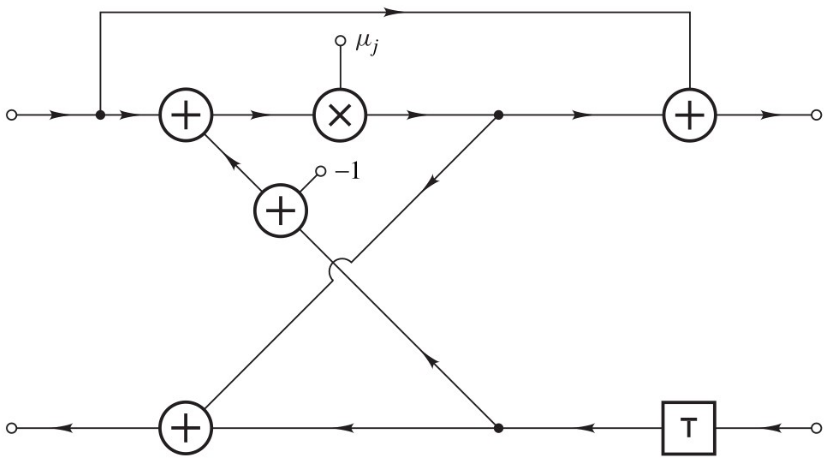
Lattice Realization



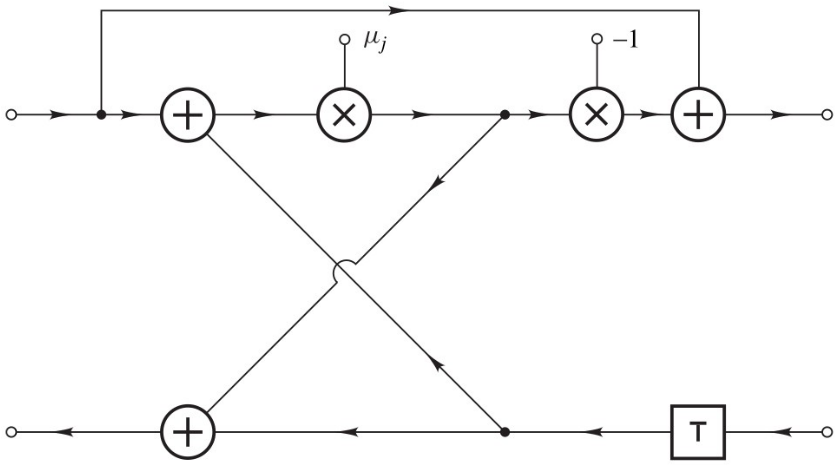
- The 2-multiplier section yields structures that are canonic with respect to the number of unit delays.
- However, the number of multipliers can be quite large, as can be seen in Example 3.

Lattice Realization

- More economical realizations can be obtained by using 1-multiplier first-order sections of the type shown below



(a)



(b)

Lattice Realization

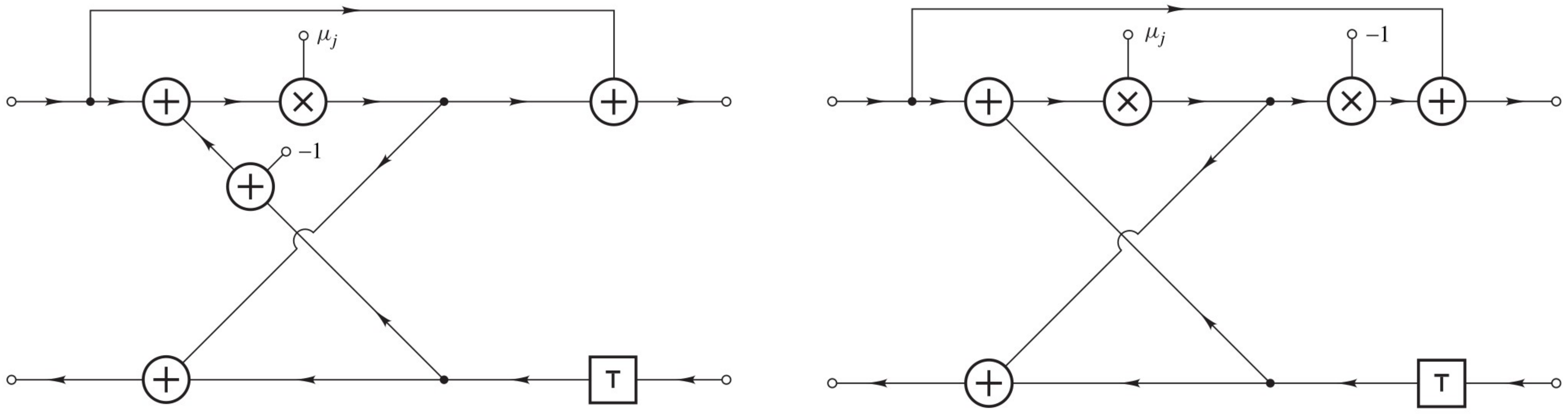
- Such realizations can be obtained by first realizing the transfer function in terms of 2-multiplier sections as described above and then replacing each of the 2-multiplier sections by either of the 1-multiplier sections
- The denominator multiplier constants $\mu_1, \mu_2, \ldots, \mu_N$ remain the same as before. However, the numerator multiplier constants $\nu_0, \nu_1, \ldots, \nu_N$ must be modified as

$$\tilde{\nu}_j = \frac{\nu_j}{\xi_j}$$

$$\xi_j = \begin{cases} 1 & \text{for } j = N \\ \prod_{i=j}^{N-1}(1 + \varepsilon_i \mu_{i+1}) & \text{for } j = 0,\ 1, \ldots,\ N-1 \end{cases}$$

Lattice Realization

- Each parameter ε_i is a constant which is equal to +1 or −1 depending on whether the ith 2-multiplier section is replaced by the 1-multiplier section of Fig. a or that of Fig. b.



(*a*) (*b*)

CASCADE REALIZATION

Cascade Realization

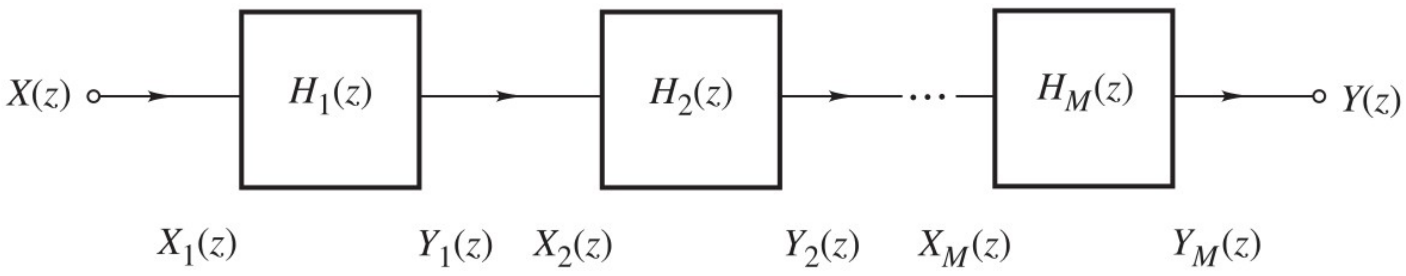
- When the transfer function coefficients are quantized, errors are introduced in the amplitude and phase responses of the filter.
- When a transfer function is realized directly in terms of a single N th-order network using any one of the methods described so far, the sensitivity of the structure to coefficient quantization increases rapidly with N
- Consequently, small errors introduced by coefficient quantization give rise to large errors in the amplitude and phase responses.

Cascade Realization

- This problem can to some extent be overcome by realizing high-order filters as interconnections of first- and second-order networks.
- In this and the next section, it is shown that an arbitrary transfer function can be realized by connecting a number of first- and second-order structures in **cascade** or in **parallel**.

Cascade Realization

- Consider an arbitrary number of filter sections connected in cascade as

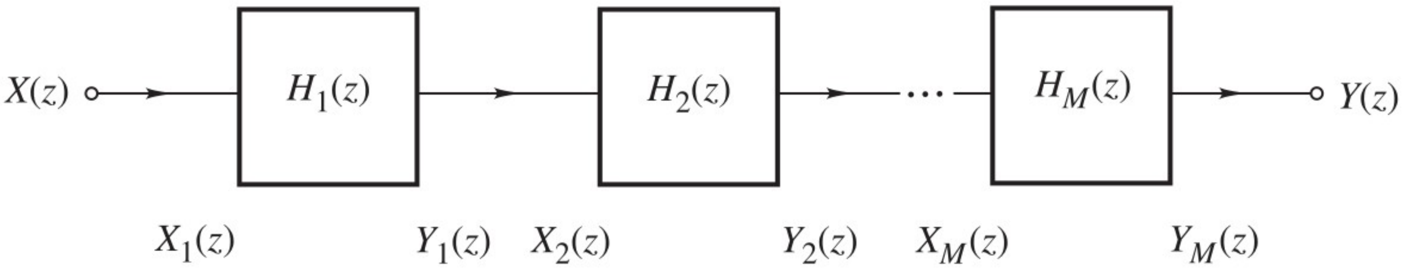


(a)

- and assume that the ith section is characterized by

$$Y_i(z) = H_i(z)X_i(z)$$

Cascade Realization



$$
\begin{aligned}
Y_1(z) &= H_1(z)X_1(z) = H_1(z)X(z) \\
Y_2(z) &= H_2(z)X_2(z) = H_2(z)Y_1(z) = H_1(z)H_2(z)X(z) \\
Y_3(z) &= H_3(z)X_3(z) = H_3(z)Y_2(z) = H_1(z)H_2(z)H_3(z)X(z) \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
Y(z) &= Y_M(z) = H_M(z)Y_{M-1}(z) = H_1(z)H_2(z)\cdots H_M(z)X(z)
\end{aligned}
$$

Cascade Realization

- The overall transfer function of a cascade arrangement of filter sections is equal to the product of the individual transfer functions, that is,

$$H(z) = \prod_{i=1}^{M} H_i(z)$$

- An N th-order transfer function can be factorized into a product of first- and second-order transfer functions of the form

$$H_i(z) = \frac{a_{0i} + a_{1i}z^{-1}}{1 + b_{1i}z^{-1}} \qquad H_i(z) = \frac{a_{0i} + a_{1i}z^{-1} + a_{2i}z^{-2}}{1 + b_{1i}z^{-1} + b_{2i}z^{-2}}$$

ELEKTRO ITK
Institut Teknologi Kalimantan

Example 4

- Obtain a cascade realization of the transfer function

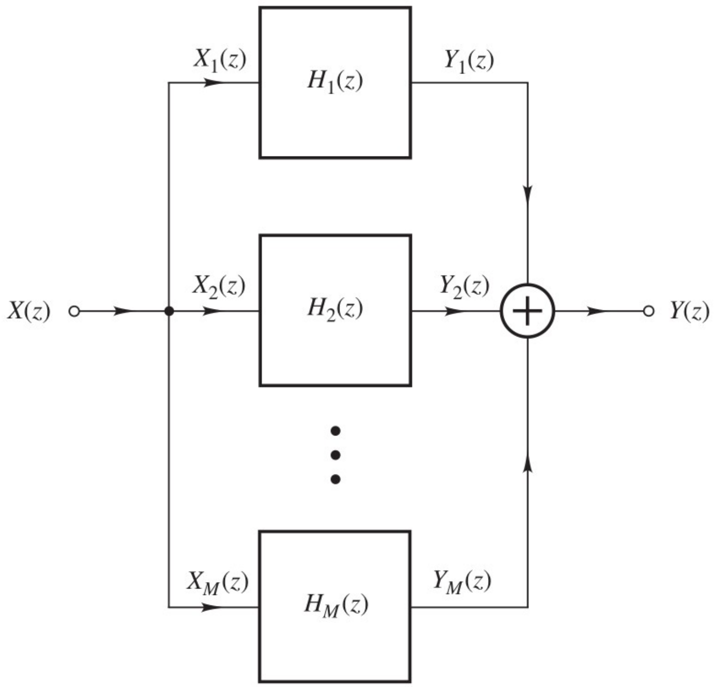
$$H(z) = \frac{216z^3 + 96z^2 + 24z}{(2z+1)(12z^2+7z+1)}$$

using canonic sections.

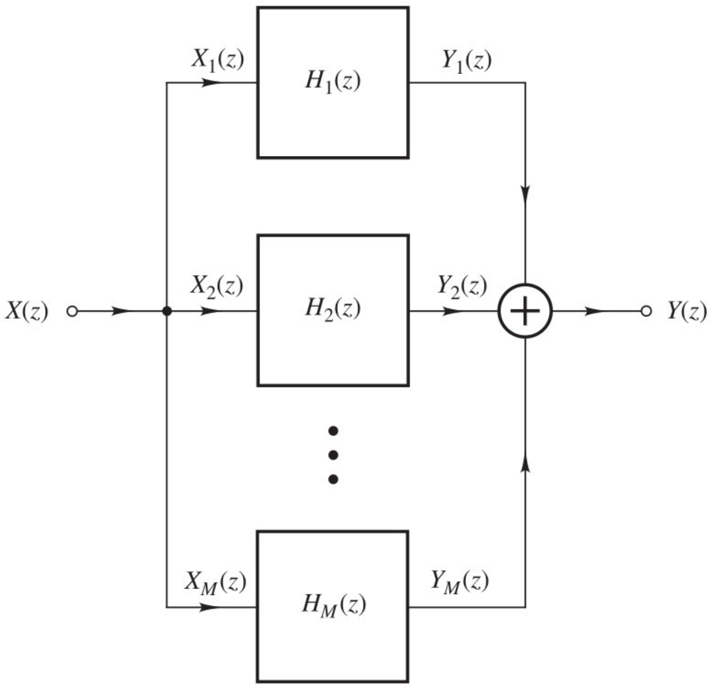
PARALLEL REALIZATION

Parallel Realization

- Another realization comprising first- and second-order filter sections is based on the parallel configuration



Parallel Realization



$$
\begin{aligned}
Y(z) &= Y_1(z) + Y_2(z) + \cdots + Y_M(z) \\
&= H_1(z)X_1(z) + H_2(z)X_2(z) + \cdots + H_M(z)X_M(z) \\
&= H_1(z)X(z) + H_2(z)X(z) + \cdots + H_M(z)X(z) \\
&= [H_1(z) + H_2(z) + \cdots + H_M(z)]X(z) \\
&= H(z)X(z)
\end{aligned}
$$

$$H(z) = \sum_{i=1}^{M} H_i(z)$$

Example 5

- Obtain a parallel realization of the transfer function

$$H(z) = \frac{10z^4 - 3.7z^3 - 1.28z^2 + 0.99z}{(z^2 + z + 0.34)(z^2 + 0.9z + 0.2)}$$

using canonic sections.