6 Structures for Discrete-Time Systems

- We shall be concerned with implementing an LTI discrete-time system, described by LCCD equation.

  We shall consider

$$\text{Filter} \begin{cases} \text{IIR Filter} \\ \text{FIR Filter} \end{cases}$$

$$\begin{array}{c}\text{Basic} \\ \text{Structure}\end{array} \begin{cases} \text{direct form (I, II)} \\ \text{cascade form} \\ \text{parallel form} \\ \text{lattice form} \\ \text{transposed form} \end{cases}$$

- Basic idea behind this topic is to implement a discrete-time LTI system, described by LCCD equation, by an interconnection of the operations of (1) addition, (2) delay, and (3) multiplication by a constant.

6.1 Block-Diagram Representation of LCCD Equations

- Basic Operation Elements:

  1. Addition of Two Sequences:

     Fig. 6.1(a)

  2. Multiplication of a Sequence by a Constant:

     Fig. 6.1(b)

  3. Unit Delay:

     Fig. 6.1(c)

- Ex: Block Diagram Representation of a Difference Equation

  Consider the second-order difference equation

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

  Its corresponding system function is

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

  The block diagram based on $H(z)$ is shown as

  Fig. 6.2

- Consider an LTI system with the N-th order LCCD equation

$$\begin{aligned} y[n] - \sum_{k=1}^{N} a_k y[n-k] &= \sum_{k=0}^{M} b_k x[n-k] \qquad \circledast \\ \Rightarrow \quad y[n] &= \sum_{k=1}^{N} a_k y[n-k] + \underbrace{\sum_{k=0}^{M} b_k x[n-k]}_{v[n]} \\ &= \sum_{k=1}^{N} a_k y[n-k] + v[n] \end{aligned}$$

Fig. 6.3

This form is called the *direct form I* implementation of the general N-th order LTI system.

- Another structure yielding an equivalent system can be obtained as follows:

Taking the z-transform of both sides in $\circledast$, we derive $H(z)$

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

This implies

$$\begin{aligned} Y(z) &\triangleq \mathcal{Z}\{y[n]\} \\ &= H(z) X(z) \\ &= H_2(z) H_1(z) X(z) \\ &= H_2(z) V(z) \end{aligned}$$

where $V(z) \triangleq \mathcal{Z}\{v[n]\}$ and $X(z) \triangleq \mathcal{Z}\{x[n]\}$. That is, the system can be viewed as a cascade of two LTI systems with $H_1(z)$ and $H_2(z)$. Now, mathematically, it is equivalent to express

$$\begin{aligned} Y(z) &= H_1(z) H_2(z) X(z) \\ &= H_1(z) W(z) \end{aligned}$$

where $W(z) \triangleq \mathcal{Z}\{w[n]\} = H_2(z)X(z)$. This implies that

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

Fig. 6.4

If $M = N$, we can use the following structure.

Fig. 6.5

If $M > N$,

Fig. 70-B1

Similarly, for $N > M$, one can derive the structure easily. This form employs the minimum number of delay units and is commonly referred to as the *canonic direct form* or *direct form II* implementation of a general N-th order LTI system.

- Ex: Direct Forms I and II Implementation of an LTI System

  Consider the LTI system with

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

  It has the direct I implementation

Fig. 6.6

  and direct form II implementation

Fig. 6.7

- Note: The implementation block diagram of an LTI system, described by an LCCD equation, is not unique since LCCD equation can not uniquely characterize an LTI system.

6.2 Signal Flow Graph Representation of LCCD Equations

- A signal flow graph is an alternative, more convenient as a matter of fact, approach for representing difference equations.
- A signal flow graph is a network of directed branches connecting nodes.

  1. There is a value associated with each node. (Here, this value represents a sample in a discrete-time sequence at a time instant).
  2. A branch connecting node j with value n_j to node k with value n_k represents a linear system with n_j as input and n_k as output.
  3. Each node has at least one branch connecting it.
  4. *Source nodes* are those nodes without entering branches.
  5. *Sink nodes* are nodes without outgoing branches.

- Example of nodes and branches in a signal flow graph

Fig. 6.8

- Example of a signal flow graph showing source and sink nodes

Fig. 6.9

$$\begin{aligned} w_1[n] &= x[n] + aw_2[n] + bw_2[n] \\ w_2[n] &= cw_1[n] \\ y[n] &= dx[n] + ew_2[n]. \end{aligned}$$

- Example of a first-order digital filter

Fig. 6.10

Fig. 6.11

- Mapping between block-diagram representation and signal flow graph representation:

Fig. 71-B1

Fig. 71-B2

Fig. 71-B3

Fig. 71-B4

- Ex: Consider the signal flow graph

Fig. 6.11

$$\begin{aligned} w_1[n] &= aw_4[n] + x[n] \\ w_2[n] &= w_1[n] \\ w_3[n] &= b_0 w_2[n] + b_1 w_4[n] \\ w_4[n] &= w_2[n-1] \\ y[n] &= w_3[n] \end{aligned}$$

That is, a (linear) signal flow graph represents a set of difference equations!

The above set of difference equations can be sized down to

$$\begin{aligned} w_2[n] &= aw_2[n-1] + x[n] \\ y[n] &= b_0 w_2[n] + b_1 w_2[n-1]. \end{aligned}$$

When there are delay elements in a signal flow graph, it is more convenient to work with z-transform representation.

- Ex: Determination of the System Function From a Signal Flow Graph

  Consider the signal flow graph

Fig. 6.12

which can be described by the set of difference equations

$$\begin{aligned} w_1[n] &= w_4[n] - x[n] \\ w_2[n] &= \alpha w_1[n] \\ w_3[n] &= w_2[n] + x[n] \\ w_4[n] &= w_3[n-1] \\ y[n] &= w_2[n] + w_4[n]. \end{aligned}$$

Taking z-transform, we have

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

Manipulating the above z-functions, we have

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

The first two equations yield

$$\begin{aligned} W_2(z) &= \frac{\alpha(z^{-1} - 1)}{1 - \alpha z^{-1}} X(z) \\ W_4(z) &= \frac{z^{-1}(1 - \alpha)}{1 - \alpha z^{-1}} X(z) \end{aligned}$$

which result in

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

Thus, with the ROC $|z| > |\alpha|$, the impulse response for the LTI system is

$$h[n] = \mathcal{Z}^{-1}\{\frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}\} = \alpha^{n-1}u[n-1] - \alpha^{n+1}u[n].$$

An equivalent direct form I implementation of the system is

Fig. 6.13

- Notes:

  1. The z-transform can easily simplify the derivation of the system function of the LTI system from a signal flow graph, particularly with delay elements.
  2. The implementation in *Fig. 6.12* requires one multiplication and one delay element, whereas the implementation in *Fig. 6.13* requires two multiplications and two delay elements. Although not illustrated, the direct form II implementation requires two multiplications and one delay element. Thus, different implementations involve different amounts of computational resources.

6.3 Basic Structures for IIR Systems

- An IIR LTI system has a general form of

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

(with $N \geq 1$) which has the corresponding rational system function

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

We shall give its various implementation forms in terms of signal flow graphs.

1. Direct Form I

   Fig. 6.14

2. Direct Form II (assuming $N = M$ and permitting zero coefficients for different combinations of (N, M))

   Fig. 6.15

   Ex: Illustration of Direct Form I and Direct Form II Structures

   Consider the rational system function

$$\begin{aligned} H(z) &= \frac{1 + 2z^{-1} + z^{-2}}{1 - 0.75z^{-1} + 0.125z^{-2}} \\ \Rightarrow\ & a_1 = 0.75,\ a_2 = -0.125 \\ & b_0 = 1,\ b_1 = 2,\ b_2 = 1 \end{aligned}$$

   Direct Form I:

   Fig. 6.16

   Direct Form II:

   Fig. 6.17

3. Cascade Form

   For rational and real-valued $H(z)$, we can factorize $H(z)$ into

$$H(z) = A \frac{\prod_{k=1}^{M_1} \left(1 - f_k z^{-1}\right) \prod_{k=1}^{M_2} \left(1 - g_k z^{-1}\right)\left(1 - g_k^* z^{-1}\right)}{\prod_{k=1}^{N_1} \left(1 - c_k z^{-1}\right) \prod_{k=1}^{N_2} \left(1 - d_k z^{-1}\right)\left(1 - d_k^* z^{-1}\right)} \qquad \circledast$$

   where f_k is a real zero, g_k is a complex zero, c_k is a real pole, d_k is a complex pole, A is a gain factor, $M = M_1 + 2M_2$, and $N = N_1 + 2N_2$.

   Since complex zeros and poles are in conjugate pair, we can form $\left(1 - g_k z^{-1}\right)\left(1 - g_k^* z^{-1}\right)$ and $\left(1 - d_k z^{-1}\right)\left(1 - d_k^* z^{-1}\right)$ into polynomials of degree 2. This implies a *cascade form* realization of $H(z)$ by the cascade of first-order and second-order subsystems.

- Particularly for $M = N$, $H(z)$ can be expressed in general as

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

where $N_s = \lfloor \frac{N+1}{2} \rfloor$ with $\lfloor x \rfloor$ representing the integer part of x. Note that the factor

$$\frac{b_{0k} + b_{1k}z^{-1} + b_{2k}z^{-2}}{1 - a_{1k}z^{-1} - a_{2k}z^{-2}}$$

can be implemented by direct-form structures. Moreover, $H(z)$ can be viewed as a cascade of N_s second-order LTI sections of either Direct-Form I or II structures. For example, consider a sixth-order system with a direct form II realization of each second-order subsystem:

Fig. 6.18

where $N_s = 3$. Note that the difference equations represented by a general cascade of direct form II second-order sections are of the form

$$\begin{aligned} y_0[n] &= x[n] \\ w_k[n] &= a_{1k}w_k[n-1] + a_{2k}w_k[n-2] + y_{k-1}[n] \quad k = 1, 2, ..., N_s \\ y_k[n] &= b_{0k}w_k[n] + b_{1k}w_k[n-1] + b_{2k}w_k[n-2] \quad k = 1, 2, ..., N_s \\ y[n] &= y_{N_s}[n]. \end{aligned}$$

Notes:

1. If there are N_s second-order sections, there are $N_s!$ pairings of the poles with zeros and $N_s!$ orderings of the resulting second-order sections, or a total of $(N_s!)^2$ different pairings and orderings. Although yielding the same overall system function, these different structures may behave quite differently in finite precision realization (see Sections 6.8-6.10 for self reading).
2. Each second-order section has at most five constant multiplications. Thus, if $N_s = N/2$ with an even N, the cascade form structure involves at most $5N/2$ constant multiplications, while both direct form I and II structures require respectively at most $2N + 1$ constant multiplications.
3. Another definition of cascade form is

$$H(z) = b_0 \prod_{k=1}^{N_s} \frac{1 + \widetilde{b}_{1k}z^{-1} + \widetilde{b}_{2k}z^{-2}}{1 - a_{1k}z^{-1} - a_{2k}z^{-2}} \qquad (b)$$

with $\widetilde{b}_{ik} = b_{ik}/b_{ik}$ for $i = 1, 2$ and $k = 1, 2, ..., N_s$ and $b_0 = \prod_{k=1}^{N_s} b_{0k}$. The corresponding cascade structure requires at most $2N + 1$ constant multiplications, which is the same as both direct form structures.

4. Both cascade form structures based on (a) and (b) have their own applications, depending on the use of finite precision arithmetic or the limitation of dynamic range. (See Sections 6.7-6.10 for self reading)

Ex: Illustration of Cascade Structures

$$H(z) = \frac{1 + 2z^{-1} + z^{-2}}{1 - 0.75z^{-1} + 0.125z^{-2}} = \frac{(1 + z^{-1})(1 + z^{-1})}{(1 - 0.5z^{-1})(1 - 0.25z^{-1})}$$

Fig. 6.19

4. Parallel Form

Applying partial fraction expansion to a real and rational $H(z)$ yields a form of

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

where $N = N_1 + 2N_2$, A_k, B_k, C_k, c_k, and e_k are real, and $N_p = M - N$ if $M \geq N$ and the first summation is not included otherwise. This system function can be interpreted as representing a parallel combination of first- and second-order IIR systems.

Alternatively, we can express $H(z)$ as

$$H(z) = \sum_{k=0}^{N_p} C_k z^{-k} + \sum_{k=1}^{N_s} \frac{e_{0k} + e_{1k} z^{-1}}{1 - a_{1k} z^{-1} - a_{2k} z^{-2}} \quad (2)$$

where $N_s = \lfloor \frac{N+1}{2} \rfloor$ and the first summation is not included if $M < N$. Since

Fig. 74-B1

both system functions in the form of (1) and (2) can be implemented as a bank of parallel LTI sections (subsystems) of direct-form structures.

- Particularly, a typical example for $M = N = 6$ and $N_s = 3$ is shown in

Fig. 6.20

The general difference equations for the parallel form with second-order direct form II sections are

$$\begin{aligned}
w_k[n] &= a_{1k}w_k[n-1] + a_{2k}w_k[n-2] + x[n] \quad k = 1,2,...,N_s \\
y_k[n] &= e_{0k}w_k[n] + e_{1k}w_k[n-1] \quad k = 1,2,...,N_s \\
y[n] &= \begin{cases} \sum_{k=0}^{N_p} C_k x[n-k] + \sum_{k=1}^{N_s} y_k[n], & \text{if } M \geq N \\ \sum_{k=1}^{N_s} y_k[n], & \text{otherwise} \end{cases}.
\end{aligned}$$

Ex: Illustration of Parallel Form Structures

$$\begin{aligned}
H(z) &= 8 + \frac{-7 + 8z^{-1}}{1 - 0.75z^{-1} + 0.125z^{-2}} \\
& \quad \textit{Fig. 6.21} \\
&= 8 + \frac{18}{1 - 0.5z^{-1}} - \frac{25}{1 - 0.25z^{-1}} \\
& \quad \textit{Fig. 6.22}
\end{aligned}$$

- Feedback in IIR Systems

  1. The signal flow graph of an IIR system contains feedback loops, i.e., it has closed paths that begin at a node and return to that node by transversing branches only in the direction of their arrowheads. Such feedback loops are necessary (but not sufficient) to generate infinitely long impulse responses. For example, consider the first-order system with

  $$\begin{aligned}
  y[n] &= ay[n-1] + x[n] \\
  & \quad \textit{Fig. 6.23(a)} \\
  h[n] &= a^n u[n].
  \end{aligned}$$

  2. If $H(z)$ has poles, a corresponding signal graph will have feedback loops. However, neither poles in $H(z)$ nor loops in a signal flow graph are sufficient for the impulse response to be infinitely long. For example, the feedback loop in a *frequency-sampling* system

  Fig. 6.23(b)

  does not guarantee an infinitely long impulse response, since the corresponding $H(z)$ is

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

  which gives $h[n] = \delta[n] + a\delta[n-1]$.

3. Feedback loops in the signal flow graph may impair the computation in implementation and make the system noncomputable. For example, consider the system with

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

Fig. 6.23(c)

It is noncomputable because the first term in the right-hand side involves $y[n]$ which is what we want to compute in the left-hand side! However, the computation problem can be resolved if we reformulate the system relation to

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

In order to ensure the computability of a signal flow graph with feedback loops, at least one delay element is required in all loops. Thus, no *delay-free* loops are permitted for computable signal flow graphs.

6.4 Transposed Forms

- *Transposition* of a signal graph, or sometimes called *flow graph reversal*, is a network obtained by reverting the directions of all branches in the original network, while keeping the branch again.
- Ex: Transposed Form for a First-Order System With No Zeros

  Consider the first-order system

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

Fig. 6.24(a)

Its transposition is

Fig. 6.24(b)

Fig. 6.24(c)

whose output $y[n]$ is equivalent to

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

$$\Rightarrow \quad H_{trans}(z) = \frac{1}{1-az^{-1}}$$

which is the same as the original!

- In general, it can be proven that a transposition of a signal flow graph has the same system function as the original graph.
- Ex*: Transposed Form for a Basic Second-Order System

Fig. 6.25

$$\Rightarrow H(z) = \frac{b_0 + b_1 z^{-1} + b_1 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-1}}.$$

The corresponding difference equations are

$$\begin{aligned} w[n] &= a_1 w[n-1] + a_2 w[n-2] + x[n] \\ y[n] &= b_0 w[n] + b_1 w[n-1] + b_2 w[n-2]. \end{aligned}$$

Its transposition is

Fig. 6.26

$$\begin{aligned} v_0[n] &= b_0 x[n] + v_1[n-1] \\ y[n] &= v_0[n] = b_0 x[n] + v_1[n-1] \\ v_1[n] &= a_1 y[n] + b_1 x[n] + v_2[n-1] \\ v_2[n] &= a_2 y[n] + b_2 x[n] \\ \Rightarrow y[n] &= b_0 x[n] + a_1 y[n-1] + b_1 x[n-1] + v_2[n-2] \\ &= b_0 x[n] + a_1 y[n-1] + b_1 x[n-1] \\ &\quad + a_2 y[n-2] + b_2 x[n-2] \\ \Rightarrow H(z) &= \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-2}} \end{aligned}$$

which is the same as the $H(z)$ of the original graph!

- Notes:

1. The transposed form can be applied to any of the structures discussed so far. For example, the transposed direct form I structure corresponding to Fig. 6.14 is

   Fig. 6.27

   Also, the transposed direct form II structure corresponding to Fig. 6.15 is

   Fig. 6.28

2. The transposed form has the same number of delay branches and the same number of coefficients as the original.
3. Although the same system function is achieved, the transposed form of direct form II structure implements the zeros first and then the poles, an opposite way to the original direct form II (see Ex*). This difference merits the finite-precision digital implementation.
4. Transpositions of cascade or parallel forms consist of a network with individual second-order systems replaced or not replaced by transposed structures. For example, the two systems

   Fig. 76-B1

   are equivalent in that

   Fig. 76-B2

   This implies that a variety of implementation structures are possible for any given rational real system functions.

6.5 Basic Network Structures for FIR Systems

- We shall consider some forms specifically suitable for the FIR systems with the form

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

with

$$h[n] = \begin{cases} b_n, & 0 \leq n \leq M \\ 0, & \text{elsewhere} \end{cases} .$$

1. Direct Form:

   Fig. 6.29

   We call this structure a *tapped delay line* (TDL) structure or a *transversal filter* structure.

   The transposition of this TDL structure is

   Fig. 6.30

   which has the same system function as the original.

2. Cascade Form:

Express real $H(z)$ as a product of the form

$$H(z) = \prod_{k=1}^{M_s} \left(b_{0k} + b_{1k} z^{-1} + b_{2k} z^{-2}\right)$$

where $M_s = \lfloor \frac{M+1}{2} \rfloor$. It has the cascade form

Fig. 6.31

The corresponding transposed form structures can be obtained accordingly.

3. Structures for Linear-Phase FIR Systems:

   Recall that causal FIR systems have generalized linear phase if the corresponding impulse response satisfies the following symmetry condition:

   a. $h[M-n] = h[n]$, $0 \le n \le M$, for type I and II systems;
   b. $h[M-n] = -h[n]$, $0 \le n \le M$, for type III and IV systems.

   Now, for M even, the I/O relationship for type I and III systems can be further simplified as

   (1) Type I:

$$\begin{aligned} y[n] &= \sum_{k=0}^{M} h[k]\, x[n-k] \\ &= \sum_{k=0}^{M/2-1} h[k]\, x[n-k] + h\left[\frac{M}{2}\right] x\left[n - \frac{M}{2}\right] + \sum_{k=M/2+1}^{M} h[k]\, x[n-k] \\ &\quad \left(\sum_{k=M/2+1}^{M} h[k]\, x[n-k] = \sum_{k=0}^{M/2-1} \underbrace{h[M-k]}_{h[k]} x[n-M+k] \right) \\ &= \sum_{k=0}^{M/2-1} h[k]\left(x[n-k] + x[n-M+k]\right) + h\left[\frac{M}{2}\right] x\left[n - \frac{M}{2}\right] \end{aligned}$$

   (2) Type III:

$$y[n] = \sum_{k=0}^{M/2-1} h[k]\left(x[n-k] - x[n-M+k]\right) \quad \left(\text{since } h\left[\frac{M}{2}\right] = 0.\right)$$

Both systems have a similar structure

Fig. 6.32

Next, for M odd, the I/O relationship for type II and IV systems can be further simplified as

(3) Type II:

$$
\begin{aligned}
y[n] &= \sum_{k=0}^{M} h[k]\, x[n-k] \\
&= \sum_{k=0}^{(M-1)/2} h[k]\, x[n-k] + \sum_{k=(M+1)/2}^{M} h[k]\, x[n-k] \\
&= \sum_{k=0}^{(M-1)/2} h[k] \left(x[n-k] + x[n-M+k] \right) \\
& \left(\sum_{k=(M+1)/2}^{M} h[k]\, x[n-k] = \sum_{k=0}^{(M-1)/2} \underbrace{h[M-k]}_{h[k]} x[n-M+k] \right)
\end{aligned}
$$

(4) Type IV:

$$
y[n] = \sum_{k=0}^{(M-1)/2} h[k] \left(x[n-k] - x[n-M+k] \right)
$$

Both systems have a similar structure

Fig. 6.33

Note: The number of coefficient multipliers is essentially halved.

- Ex: Consider a system with

Fig. 6.34

1. There is a pole of order 9 at $z = 0$ (although not marked). $\Rightarrow$ It is an FIR system.
2. The zeros are either real or in a complex conjugate pair. $\Rightarrow$ It is a real FIR system.

3. All zeros are associated with their conjugate, conjugate reciprocal, and conjugate conjugate reciprocal. $\Rightarrow$ It is a real and linear phase FIR system.
4. Its system function is

$$\begin{aligned}
H(z) &= G\left(1 - z_4 z^{-1}\right) \cdot \left(1 - z_2 z^{-1}\right)\left(1 - z_2^{-1} z^{-1}\right) \\
&\qquad \cdot \left(1 - z_3 z^{-1}\right)\left(1 - z_3^* z^{-1}\right) \\
&\qquad \cdot \left(1 - z_1 z^{-1}\right)\left(1 - z_1^{-1} z^{-1}\right) \\
&\qquad \cdot \left(1 - z_1^* z^{-1}\right)\left(1 - \left(z_1^*\right)^{-1} z^{-1}\right) \\
&= G\left(1 + z^{-1}\right) \cdot \\
&\quad \left(1 - \left(z_2 + z_2^{-1}\right) z^{-1} + z^{-2}\right) \cdot \\
&\quad \left(1 - 2\,\mathrm{Re}\left\{z_3\right\} z^{-1} + z^{-2}\right) \cdot \\
&\quad \left(1 - \left(z_1 + z_1^{-1}\right) z^{-1} + z^{-2}\right) \cdot \\
&\quad \left(1 - \left(z_1^* + \left(z_1^{-1}\right)^*\right) z^{-1} + z^{-2}\right) \\
&= G\left(1 + z^{-1}\right)\left(1 + a z^{-1} + z^{-2}\right) \\
&\quad \cdot \left(1 + b z^{-1} + z^{-2}\right) \cdot \left(1 + c z^{-1} + d z^{-2} + c z^{-3} + z^{-4}\right)
\end{aligned}$$

   where

$$\begin{aligned}
G &= h[0] \\
a &= -\left(z_2 + z_2^{-1}\right) \\
b &= -2\,\mathrm{Re}\left\{z_3\right\} \\
c &= -2\,\mathrm{Re}\left\{z_1 + z_1^{-1}\right\} \\
d &= 2 + \left|z_1 + z_1^{-1}\right|^2
\end{aligned}$$

   are all real. If we use a cascade structure

   Fig. 79-B1

   we need five "different" multipliers. If we use its specific structure ($M = 1 + 2 + 2 + 4 = 9$),

   Fig. 79-B2

   we have five multipliers (may or may not be all different)!

6.6 Lattice Filters

- Recall that both IIR and FIR systems can be implemented in cascade structures if their system functions can be factorized into first- and second-order sections. Here, we are interested in the cascade system structures which are based on connection of the two-port basic structure with *two inputs and two outputs*, shown as

Fig. 6.35

where the two inputs are defined by $a^{(i-1)}[n]$ and $b^{(i-1)}[n]$, and the two outputs are defined by $a^{(i)}[n]$ and $b^{(i)}[n]$.

In general, the cascade system structure of interest is formed as a cascade of M of these basic elements with a termination at each end of the cascade, so that the system has input $x[n]$ and output $y[n]$, related by

Fig. 6.36

Fig. 6.37

where $x[n] = a^{(0)}[n] = b^{(0)}[n]$ and $y[n] = a^{(M)}[n]$. The filters of this cascade form are called *lattice filters*.

6.6.1 FIR Lattice Filters

- An FIR lattice filter with input $x[n]$ and output $y[n]$ can be generally described in a signal flow graph as

  Fig. 6.37

  where the coefficients $k_1, k_2, ..., k_M$ are called the k-parameters of the lattice structure. The node variables $a^{(i)}[n]$ and $b^{(i)}[n]$ are intermediate sequences related by difference equations

$$\begin{aligned} a^{(0)}[n] &= b^{(0)}[n] = x[n] \\ a^{(i)}[n] &= a^{(i-1)}[n] - k_i b^{(i-1)}[n-1] \qquad i = 1, 2, ..., M \\ b^{(i)}[n] &= b^{(i-1)}[n-1] - k_i a^{(i-1)}[n] \qquad i = 1, 2, ..., M \\ y[n] &= a^{(M)}[n]. \end{aligned}$$

  Notes:

  1. These difference equations should be conducted in the order $i = 0, 1, ..., M$.
  2. The lattice filter is LTI since the flow graph is a linear one with only delays and constant branch coefficients, i.e., the current output sample $y[n]$ can be represented as a linear combination of the current and M previous input samples $x[n]$ and $x[n-M], x[n-M+1], ..., x[n-1]$.

3. There are no feedback loops in the flow graph. This renders a finite-duration impulse response.

- Let $x[n] = \delta[n]$. In the case, $a^{(i)}[n]$ and $b^{(i)}[n]$ become the impulse responses observed at the i-th upper and lower nodes, respectively. Thus, if we define the z-transforms $A^{(i)}(z)$ and $B^{(i)}(z)$ by

$$\begin{aligned} A^{(i)}(z) &= \sum_{n=0}^{i} a^{(i)}[n]z^{-n} \\ B^{(i)}(z) &= \sum_{n=0}^{i} b^{(i)}[n]z^{-n} \end{aligned}$$

for $i = 0, 1, ..., M$, then $A^{(i)}(z)$ and $B^{(i)}(z)$ represent the transfer functions between the input and the upper i-th node, and between the input and the lower i-th node, respectively. Because $y[n] = a^{(M)}[n]$, the impulse response for the FIR lattice filter is given by the coefficient sequence of $A^{(M)}(z)$, i.e., $a^{(M)}[0], a^{(M)}[1], ..., a^{(M)}[M]$. Note that $A(z) = A^{(M)}(z)$ is the transfer function of the FIR lattice filter. This is what we shall obtain in the following.

- Alternatively, $A^{(i)}(z)$ can be represented for $i = 1, 2, ..., M$ as

$$A^{(i)}(z) = 1 - \sum_{m=1}^{i} \alpha_m^{(i)} z^{-m} \qquad (+)$$

where the coefficients $\alpha_m^{(i)}$ for $m \le i$ are composed of sums of products of $k_1, k_2, ..., k_m$. This means that when $x[n] = \delta[n]$, $a^{(i)}[n]$ for $i = 1, 2, ..., M$ is given by

$$a^{(i)}[n] = \begin{cases} 1, & n = 0 \\ -\alpha_n^{(i)}, & n = 1, 2, ..., i \\ 0, & \text{otherwise} \end{cases} .$$

In what follows, we shall derive $\alpha_m^{(i)}$ for $m = 1, 2, ..., i$ and $i = 1, 2, ..., M$ recursively.

1. Note first that

$$\begin{aligned} A^{(0)}(z) &= B^{(0)}(z) = 1 \\ A^{(i)}(z) &= A^{(i-1)}(z) - k_i z^{-1} B^{(i-1)}(z) \qquad i = 1, 2, ..., M \quad \text{(a)} \\ B^{(i)}(z) &= z^{-1} B^{(i-1)}(z) - k_i A^{(i-1)}(z) \qquad i = 1, 2, ..., M. \quad \text{(b)} \end{aligned}$$

Proof: First, since $a^{(0)}[n] = b^{(0)}[n] = \delta[n]$, $A^{(0)}(z) = B^{(0)}(z) = 1$. Second, for $i = 1, 2, ..., M$,

$$\begin{aligned}
A^{(i)}(z) &= \sum_{n=0}^{i} a^{(i)}[n]z^{-n} \\
&= a^{(i)}[0] + \sum_{n=1}^{i} a^{(i)}[n]z^{-n} \\
&= a^{(i)}[0] + \sum_{n=1}^{i} a^{(i-1)}[n]z^{-n} - k_i \sum_{n=1}^{i} b^{(i-1)}[n-1]z^{-n} \\
&= \sum_{n=0}^{i-1} a^{(i-1)}[n]z^{-n} - k_i z^{-1} \sum_{n=1}^{i} b^{(i-1)}[n-1]z^{-(n-1)} \\
(\text{since } a^{(i-1)}[i] &= 0 \text{ and } a^{(i)}[0] = a^{(i-1)}[0]) \\
&= A^{(i-1)}(z) - k_i z^{-1} B^{(i-1)}(z)
\end{aligned}$$

which proves (a). Similarly, (b) can be proved.

2. Lemma: For $i = 1, 2, ..., M$, $A^{(i)}(z)$ and $B^{(i)}(z)$ are related by

$$\begin{aligned}
B^{(i)}(z) &= z^{-i} A^{(i)}(z^{-1}) \qquad (*) \\
A^{(i)}(z) &= z^{-i} B^{(i)}(z^{-1}).
\end{aligned}$$

Proof: It suffices to prove the first relation since the second can be obtained from the first by replacing z by z^{-1}. Let us prove the first relation by induction.
For $i = 1$, we have from (a) and (b) that

$$\begin{aligned}
A^{(1)}(z) &= 1 - k_1 z^{-1} \\
B^{(1)}(z) &= z^{-1} - k_1 = z^{-1} A^{(1)}(z^{-1})
\end{aligned}$$

which yields (*) for $i = 1$.
For $i = 2$, (a) and (b) give

$$\begin{aligned}
A^{(2)}(z) &= A^{(1)}(z) - k_2 z^{-1} B^{(1)}(z) = 1 - k_1(1-k_2)z^{-1} - k_2 z^{-2} \\
B^{(2)}(z) &= z^{-1} B^{(1)}(z) - k_2 A^{(1)}(z) = z^{-2}(1 - k_1(1-k_2)z - k_2 z^2) \\
&= z^{-2} A^{(2)}(z^{-1})
\end{aligned}$$

which yields (*) for $i = 2$.

Now, consider $i = 3, 4, ..., M$. Assume that the relation (*) holds up to $i-1$. Use the relation $B^{(i-1)}(z) = z^{-(i-1)}A^{(i-1)}(z^{-1})$ and $A^{(i-1)}(z) = z^{-(i-1)}B^{(i-1)}(z^{-1})$ in (b) gives

$$\begin{aligned}
B^{(i)}(z) &= z^{-1}[z^{-(i-1)}A^{(i-1)}(z^{-1})] - k_i[z^{-(i-1)}B^{(i-1)}(z^{-1})] \\
&= z^{-i}[A^{(i-1)}(z^{-1}) - k_i z B^{(i-1)}(z^{-1})] \\
&= z^{-i}A^{(i)}(z^{-1})
\end{aligned}$$

where the last equality follows from (a). This proves (*) by induction.

3. From (+) and (*), we have

$$\begin{aligned}
B^{(i-1)}(z) &= z^{-(i-1)}A^{(i-1)}(z^{-1}) \\
&= z^{-(i-1)}[1 - \sum_{m=1}^{i-1} \alpha_m^{(i-1)} z^m]
\end{aligned}$$

Using this and (+), (a) can be expressed as

$$\begin{aligned}
A^{(i)}(z) &= A^{(i-1)}(z) - k_i z^{-1} B^{(i-1)}(z) \\
&= [1 - \sum_{m=1}^{i-1} \alpha_m^{(i-1)} z^{-m}] - k_i z^{-1}\{z^{-(i-1)}[1 - \sum_{m=1}^{i-1} \alpha_m^{(i-1)} z^m]\} \\
&= 1 - \sum_{m=1}^{i-1} [\alpha_m^{(i-1)} - k_i \alpha_{i-m}^{(i-1)}] z^{-m} - k_i z^{-i} \\
&= 1 - \sum_{m=1}^{i} \alpha_m^{(i)} z^{-m} \qquad \text{(from (+))}
\end{aligned}$$

which shows the recursive relation

$$\begin{aligned}
\alpha_m^{(i)} &= \alpha_m^{(i-1)} - k_i \alpha_{i-m}^{(i-1)} \qquad m = 1, 2, ..., i-1 \qquad \text{(A)} \\
\alpha_i^{(i)} &= k_i \qquad \text{(B)}
\end{aligned}$$

for $i = 1, 2, ..., M$, with the initial setup $\alpha_1^{(1)} = k_1$. Using the recursion (A) and (B), the coefficients of $A^{(i)}(z)$ and $B^{(i)}(z)$ (with the aid of (*)) can be computed.

- If we define the vectors

$$\begin{aligned}
\boldsymbol{\alpha}_{i-1} &= [\alpha_1^{(i-1)}, \alpha_2^{(i-1)}, ..., \alpha_{i-1}^{(i-1)}]^T \\
\widehat{\boldsymbol{\alpha}}_{i-1} &= [\alpha_{i-1}^{(i-1)}, \alpha_{i-2}^{(i-1)}, ..., \alpha_1^{(i-1)}]^T
\end{aligned}$$

where $\widehat{\boldsymbol{\alpha}}_{i-1}$ is obtained by reversing the entries in $\boldsymbol{\alpha}_{i-1}$, $\boldsymbol{\alpha}_i$ can be related to $\boldsymbol{\alpha}_{i-1}$ and $\widehat{\boldsymbol{\alpha}}_{i-1}$ from (A) and (B) as

$$\boldsymbol{\alpha}_i = \begin{bmatrix} \boldsymbol{\alpha}_{i-1} \\ 0 \end{bmatrix} - k_i \begin{bmatrix} \widehat{\boldsymbol{\alpha}}_{i-1} \\ -1 \end{bmatrix} \qquad i = 1, 2, ..., M \tag{V}$$

with the initial setup $\boldsymbol{\alpha}_1 = \widehat{\boldsymbol{\alpha}}_1 = [k_1]$. This shows the recursion of computing $\alpha_m^{(i)}$'s from preassigned k-parameters $k_1, k_2, ..., k_M$ in matrix form.

- With $X(z) = \sum_n x[n]z^{-n}$ and $Y(z) = \sum_n y[n]z^{-n}$ representing the input and the output in z-domain of the FIR lattice filter, they can be related through the system transfer function $A(z)$

$$A(z) = \frac{Y(z)}{X(z)} = 1 - \sum_{m=1}^{M} \alpha_m z^{-m}$$

where $\alpha_m = \alpha_m^{(M)}$ for $m = 1, 2, ..., M$. Using (V) to compute $\boldsymbol{\alpha}_M$ from preassigned k-parameters $k_1, k_2, ..., k_M$ recursively, $A(z)$ can be obtained and represents the equivalent LTI system that describes the FIR lattice filter.

Fig. 6.38

This constitutes the procedure of obtaining the equivalent LTI system structure (with impulse response $h[0] = 1$ and $h[m] = -\alpha_m$ for $m = 1, 2, ..., M$) from a given lattice filter structure (with k-parameters $k_1, k_2, ..., k_M$).

- We are also interested in computing the k-parameters $k_1, k_2, ..., k_M$ from preassigned $\alpha_m = \alpha_m^{(M)}$ for $m = 1, 2, ..., M$.

  1. From (B),
  $$k_M = \alpha_M^{(M)} = \alpha_M.$$
  2. Since $k_i = \alpha_i^{(i)}$ for $i = 1, 2, ..., M$ (from (B)), we need a backward recursion of computing $\alpha_m^{(i)}$'s as follows: Now, replacing m by $i-m$ and multiplying both sides by k_i in (A), we have
  $$k_i \alpha_{i-m}^{(i)} = k_i \alpha_{i-m}^{(i-1)} - k_i^2 \alpha_m^{(i-1)}$$
  Adding this to $\alpha_m^{(i)} = \alpha_m^{(i-1)} - k_i \alpha_{i-m}^{(i-1)}$ yields
  $$\alpha_m^{(i)} + k_i \alpha_{i-m}^{(i)} = \alpha_m^{(i-1)} - k_i^2 \alpha_m^{(i-1)}$$
  $$\Leftrightarrow \quad \alpha_m^{(i-1)} = \frac{\alpha_m^{(i)} + k_i \alpha_{i-m}^{(i)}}{1 - k_i^2} \qquad m = 1, 2, ..., i-1. \tag{W}$$

(W) gives a recursion of computing $\alpha_m^{(i-1)}$'s from $\alpha_m^{(i)}$'s. Thus, $k_i = \alpha_i^{(i)}$ for $i = 1, 2, ..., M-1$ can be obtained recursively.

Fig. 6.39

This constitutes the procedure of obtaining the equivalent lattice filter structure (with k-parameters $k_1, k_2, ..., k_M$) from a given LTI system structure (with impulse response $h[0] = 1$ and $h[m] = -\alpha_m$ for $m = 1, 2, ..., M$).

- Ex: k-Parameters for a Third-Order FIR System

  Now, the FIR LTI system in

  Fig. 6.40(a)

  has the FIR coefficients $\alpha_1 = \alpha_1^{(3)} = 0.9$, $\alpha_2 = \alpha_2^{(3)} = -0.64$, and $\alpha_3 = \alpha_3^{(3)} = 0.576$. It has an equivalent lattice filter with $M = 3$ and

$$\begin{aligned} k_3 &= \alpha_3^{(3)} = \alpha_3 = 0.576 \\ k_2 &= \alpha_2^{(2)} = \frac{\alpha_2^{(3)} + k_3\alpha_1^{(3)}}{1-k_3^2} = -0.182 \\ \alpha_1^{(2)} &= \frac{\alpha_1^{(3)} + k_3\alpha_2^{(3)}}{1-k_3^2} = 0.795 \\ k_1 &= \alpha_1^{(1)} = \frac{\alpha_1^{(2)} + k_2\alpha_1^{(2)}}{1-k_2^2} = 0.673 \end{aligned}$$

Fig. 6.40(b)

6.6.2 All-Pole Lattice Structure

- A lattice structure for the all-pole system with system function $H(z) = 1/A(z)$ can be developed from the FIR lattice with system function $A(z)$, as follows.

  Now, consider an M-stage lattice structure with $a^{(0)}[n] = x[n]$ as *output* and $y[n] = a^{(M)}[n]$ as *input*, where $x[n]$ and $y[n]$ correspond to the input and output, respectively, of the FIR lattice filter with system function $A(z)$. Thus, we want to invert the computation order of $A^{(i)}(z)$'s in the FIR lattice filter with $A(z)$ to obtain the all-pole lattice filter with $H(z)$. Now, treating $a^{(i)}[n]$ as *input* and $a^{(i-1)}[n]$ as *output,* the i-th stage of the FIR lattice filter with $A(z)$ can be described alternatively from (a) and (b) as

$$\begin{aligned} A^{(i-1)}(z) &= A^{(i)}(z) + k_i z^{-1} B^{(i-1)}(z) && \text{(c)} \\ B^{(i)}(z) &= -k_i A^{(i-1)}(z) + z^{-1} B^{(i-1)}(z) && \text{(d)} \end{aligned}$$

for $i = M, M-1, ..., 1$, where only (a) is inverted but (b) remains unchanged.

(c) and (d) can be described by the flow graph

Fig. 6.41

where the signal flow is from i to $i-1$ along the top side and from $i-1$ to i along the bottom side. Using (c) and (d) as a building block, we can construct a cascade of M stages with the flow graph

Fig. 6.42

which can be described by the difference equations

$$\begin{aligned} a^{(M)}[n] &= y[n] \\ a^{(i-1)}[n] &= a^{(i)}[n] + k_i b^{(i-1)}[n-1] \qquad i = M, M-1, ..., 1 \\ b^{(i)}[n] &= b^{(i-1)}[n-1] - k_i a^{(i-1)}[n] \qquad i = 1, 2, ..., M \\ x[n] &= a^{(0)}[n] = b^{(0)}[n]. \end{aligned}$$

This is the all-pole lattice structure with the system function $H(z) = 1/A(z)$.

Notes:

1. There is inherent feedback in the all-pole lattice system. Thus, an initial rest condition is required, e.g., $b^{(i)}[-1] = 0$ for $i = 0, 1, ..., M$.
2. Comparing *Fig. 6.42* with *Fig. 6.37*, one can obtain the lattice structure of the all-pole system with $H(z) = 1/A(z)$ immediately from the lattice structure of the FIR system with $A(z)$.
3. Note that the lattice structure has the same number of delay elements (memory registers) as the previously-mentioned direct form structure, but requires twice the number of multipliers as required by the direct form realization. This impedes the popularity of lattice structures in certain applications.

4. One has to be concerned with the stability of the all-pole lattice filter. As shown in Chapter 13, a necessary and sufficient condition for the stability, i.e., all poles of $H(z)$ locating inside the unit circle, is $|k_i| < 1$ for $i = 1, 2, ..., M$. In general, the lattice realization is *insensitive* to quantization of the k-parameters as long as the stability condition is met. Thus, the lattice structures are still popular in some practical applications, e.g., in speech synthesis applications.

- Ex: Lattice Implementation of an IIR System

  Consider the all-pole system with

$$\begin{aligned} H(z) &= \frac{1}{1 - 0.9z^{-1} + 0.64z^{-2} - 0.576z^{-3}} \\ &= \frac{1}{(1 - 0.8jz^{-1})(1 + 0.8jz^{-1})(1 - 0.9z^{-1})} \end{aligned}$$

  which is the inverse of *Fig. 6.40(a)*. Note that all poles locate inside the unit circle and thus the all-pole system is stable. It has the direct form realization

  Fig. 6.43(a)

  and the equivalent lattice realization

  Fig. 6.43(b)

  Note that the lattice realization has the same number of delay elements (memory registers) as the direct form realization, but requires twice the number of multipliers as required by the direct form realization.

- See Subsection 6.6.3 for the comment on the generalization of lattice structures for systems having both poles and zeros.