6.3 BASIC STRUCTURES FOR IIR SYSTEMS

In Section 6.1, we introduced two alternative structures for implementing a linear time-invariant system with system function as in Eq. (6.8). In this section we present the signal flow graph representations of those systems, and we also develop several other commonly used equivalent flow graph network structures. Our discussion will make it clear that, for any given rational system function, a wide variety of equivalent sets of difference equations or network structures exists. One consideration in the choice among these different structures is computational complexity. For example, in some digital implementations, structures with the fewest constant multipliers and the fewest delay branches are often most desirable. This is because multiplication is generally a time-consuming and costly operation in digital hardware and because each delay element corresponds to a memory register. Consequently, a reduction in the number of constant multipliers means an increase in speed, and a reduction in the number of delay elements means a reduction in memory requirements.

Other, more subtle, trade-offs arise in VLSI implementations, in which the area of a chip is often an important measure of efficiency. Modularity and simplicity of data transfer on the chip are also frequently very desirable in such implementations. In multiprocessor implementations, the most important considerations are often related to partitioning of the algorithm and communication requirements between processors. Another major consideration is the effects of a finite register length and finite-precision arithmetic. These effects depend on the way in which the computations are organized, i.e., on the structure of the signal flow graph. Sometimes it is desirable to use a structure that does not have the minimum number of multipliers and delay elements if that structure is less sensitive to finite register length effects.

In this section, we develop several of the most commonly used forms for implementing a linear time-invariant IIR system and obtain their flow graph representations.

6.3.1 Direct Forms

In Section 6.1, we obtained block diagram representations of the direct form I (Figure 6.3) and direct form II, or canonic direct form (Figure 6.5), structures for a linear time-invariant system whose input and output satisfy a difference equation of the form

$$y[n] - \sum_{k=1}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k], \tag{6.26}$$

with the corresponding rational system function

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 - \sum_{k=1}^{N} a_k z^{-k}}. \tag{6.27}$$

In Figure 6.14, the direct form I structure of Figure 6.3 is shown using signal flow graph conventions, and Figure 6.15 shows the signal flow graph representation of the direct form II structure of Figure 6.5. Again, we have assumed for convenience that $N = M$.

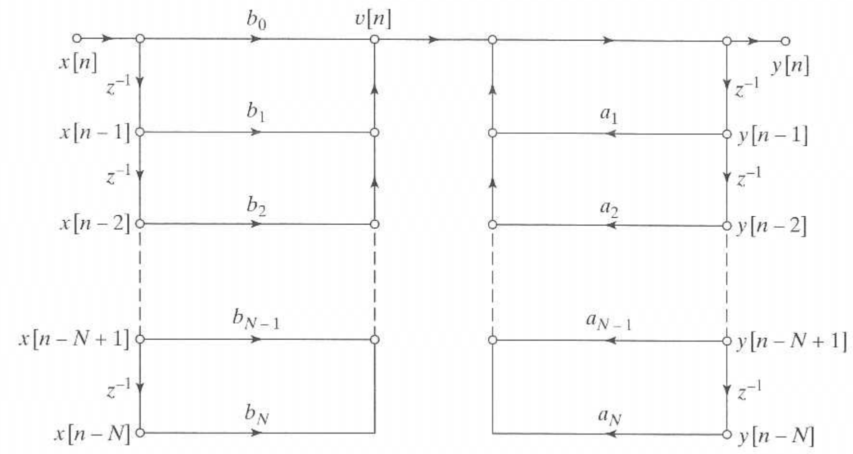

Figure 6.14 Signal flow graph of direct form I structure for an Nth-order system.

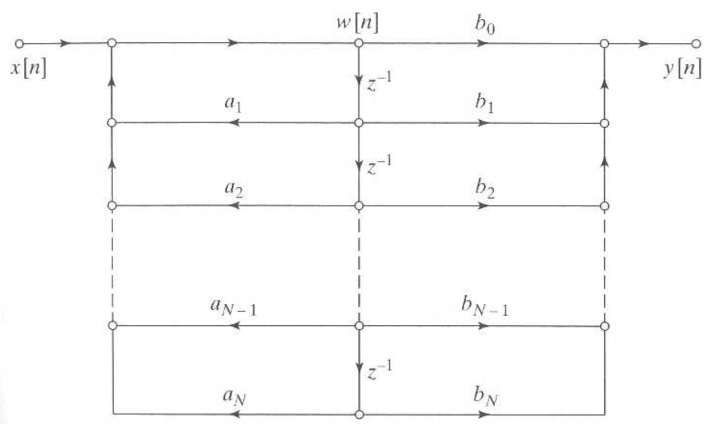

Figure 6.15 Signal flow graph of direct form II structure for an Nth-order system.

Note that we have drawn the flow graph so that each node has no more than two inputs. A node in a signal flow graph may have any number of inputs, but, as indicated earlier, this two-input convention results in a graph that is more closely related to programs and architectures for implementing the computation of the difference equations represented by the graph.

Example 6.4 Illustration of Direct Form I and Direct Form II Structures

Consider the system function

$$H(z) = \frac{1 + 2z^{-1} + z^{-2}}{1 - 0.75z^{-1} + 0.125z^{-2}}. \tag{6.28}$$

Since the coefficients in the direct form structures correspond directly to the coefficients of the numerator and denominator polynomials (taking into account the minus sign in the denominator of Eq. (6.27)), we can draw these structures by inspection with reference to Figures 6.14 and 6.15. The direct form I and direct form II structures for this example are shown in Figures 6.16 and 6.17, respectively.

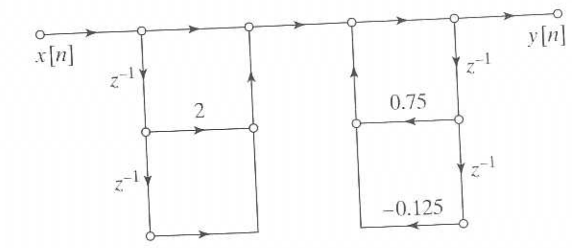

Figure 6.16 Direct form I structure for Example 6.4.

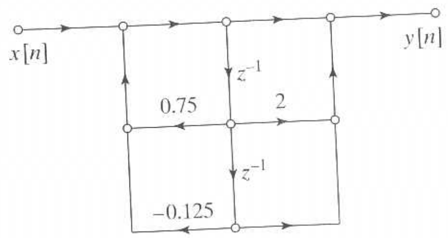

Figure 6.17 Direct form II structure for Example 6.4.

6.3.2 Cascade Form

The direct form structures were obtained directly from the system function $H(z)$, written as a ratio of polynomials in the variable z^{-1} as in Eq. (6.27). If we factor the numerator and denominator polynomials, we can express $H(z)$ in the form

$$H(z) = A\frac{\prod_{k=1}^{M_1}(1 - f_k z^{-1})\prod_{k=1}^{M_2}(1 - g_k z^{-1})(1 - g_k^* z^{-1})}{\prod_{k=1}^{N_1}(1 - c_k z^{-1})\prod_{k=1}^{N_2}(1 - d_k z^{-1})(1 - d_k^* z^{-1})}, \tag{6.29}$$

where $M = M_1 + 2M_2$ and $N = N_1 + 2N_2$. In this expression, the first-order factors represent real zeros at f_k and real poles at c_k, and the second-order factors represent complex conjugate pairs of zeros at g_k and g_k^* and complex conjugate pairs of poles at d_k and d_k^*. This represents the most general distribution of poles and zeros when all the coefficients in Eq. (6.27) are real. Equation (6.29) suggests a class of structures consisting of a cascade of first- and second-order systems. There is considerable freedom in the choice of composition of the subsystems and in the order in which the subsystems are cascaded. In practice, however, it is often desirable to implement the cascade realization using a minimum of storage and computation. A modular structure that is advantageous

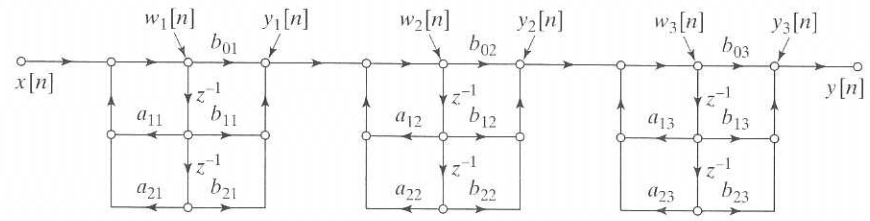

Figure 6.18 Cascade structure for a sixth-order system with a direct form II realization of each second-order subsystem.

for many types of implementations is obtained by combining pairs of real factors and complex conjugate pairs into second-order factors so that Eq. (6.29) can be expressed as

$$H(z) = \prod_{k=1}^{N_s} \frac{b_{0k} + b_{1k}z^{-1} + b_{2k}z^{-2}}{1 - a_{1k}z^{-1} - a_{2k}z^{-2}}, \tag{6.30}$$

where $N_s = \lfloor (N+1)/2 \rfloor$ is the largest integer contained in $(N+1)/2$. In writing $H(z)$ in this form, we have assumed that $M \le N$ and that the real poles and zeros have been combined in pairs. If there are an odd number of real zeros, one of the coefficients b_{2k} will be zero. Likewise, if there are an odd number of real poles, one of the coefficients a_{2k} will be zero. The individual second-order sections can be implemented using either of the direct form structures; however, the previous discussion shows that we can implement a cascade structure with a minimum number of multiplications and a minimum number of delay elements if we use the direct form II structure for each second-order section. A cascade structure for a sixth-order system using three direct form II second-order sections is shown in Figure 6.18. The difference equations represented by a general cascade of direct form II second-order sections are of the form

$$y_0[n] = x[n], \tag{6.31a}$$

$$w_k[n] = a_{1k}w_k[n-1] + a_{2k}w_k[n-2] + y_{k-1}[n], \quad k = 1, 2, \ldots, N_s, \tag{6.31b}$$

$$y_k[n] = b_{0k}w_k[n] + b_{1k}w_k[n-1] + b_{2k}w_k[n-2], \quad k = 1, 2, \ldots, N_s, \tag{6.31c}$$

$$y[n] = y_{N_s}[n]. \tag{6.31d}$$

It is easy to see that a variety of theoretically equivalent systems can be obtained by simply pairing the poles and zeros in different ways and by ordering the second-order sections in different ways. Indeed, if there are N_s second-order sections, there are $N_s!$ (N_s factorial) 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 these all have the same overall system function and corresponding input–output relation when infinite-precision arithmetic is used, their behavior with finite-precision arithmetic can be quite different, as we will see in Section 6.8.

Example 6.5 Illustration of Cascade Structures

Let us again consider the system function of Eq. (6.28). Since this is a second-order system, a cascade structure with direct form II second-order sections reduces to the structure of Figure 6.17. Alternatively, to illustrate the cascade structure, we can use first-order systems by expressing $H(z)$ as a product of first-order factors, as in

$$H(z) = \frac{1 + 2z^{-1} + z^{-2}}{1 - 0.75z^{-1} + 0.125z^{-2}} = \frac{(1 + z^{-1})(1 + z^{-1})}{(1 - 0.5z^{-1})(1 - 0.25z^{-1})}. \tag{6.32}$$

Since all of the poles and zeros are real, a cascade structure with first-order sections has real coefficients. If the poles and/or zeros were complex, only a second-order section would have real coefficients. Figure 6.19 shows two equivalent cascade structures, each of which has the system function in Eq. (6.32). The difference equations represented by the flow graphs in the figure can be written down easily. Problem 6.22 is concerned with finding other, equivalent system configurations.

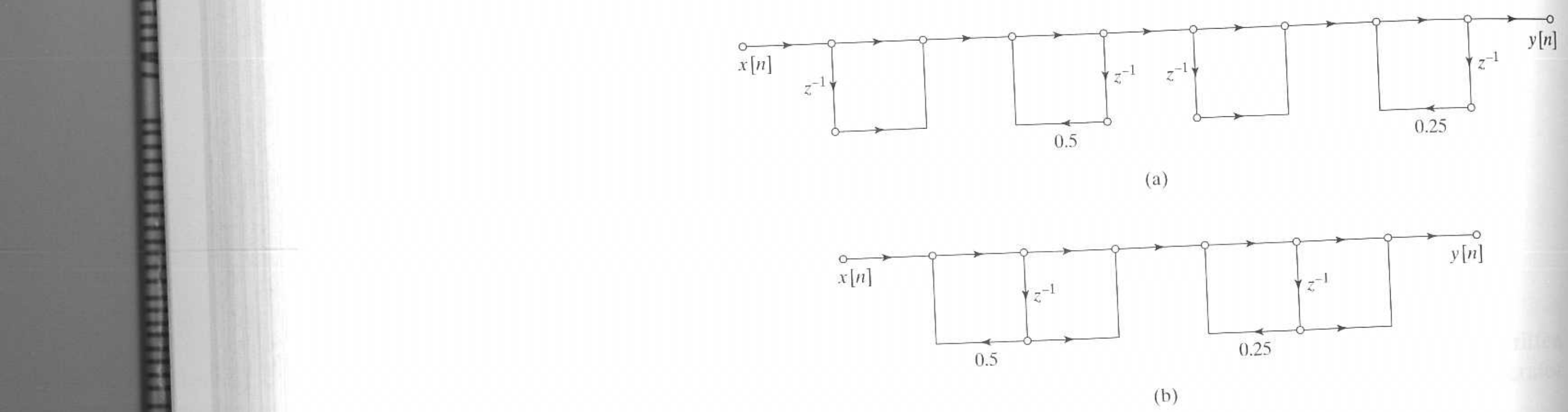

Figure 6.19 Cascade structures for Example 6.5. (a) Direct form I subsections. (b) Direct form II subsections.

A final comment should be made about our definition of the system function for the cascade form. As defined in Eq. (6.30), each second-order section has five constant multipliers. For comparison, let us assume that $M = N$ in $H(z)$ as given by Eq. (6.27), and furthermore, assume that N is an even integer, so that $N_s = N/2$. Then the direct form I and II structures have $2N + 1$ constant multipliers, while the cascade form structure suggested by Eq. (6.30) has $5N/2$ constant multipliers. For the sixth-order system in Figure 6.18, we require a total of 15 multipliers, while the equivalent direct forms would require a total of 13 multipliers. Another definition of the cascade form is

$$H(z) = b_0 \prod_{k=1}^{N_s} \frac{1 + \tilde{b}_{1k}z^{-1} + \tilde{b}_{2k}z^{-2}}{1 - a_{1k}z^{-1} - a_{2k}z^{-2}}, \tag{6.33}$$

where b_0 is the leading coefficient in the numerator polynomial of Eq. (6.27) and

$\tilde{b}_{ik} = b_{ik}/b_{0k}$ for $i = 1, 2$ and $k = 1, 2, \ldots, N_s$. This form for $H(z)$ suggests a cascade of four-multiplier second-order sections, with a single overall gain constant b_0. This cascade form has the same number of constant multipliers as the direct form structures. As discussed in Section 6.8, the five-multiplier second-order sections are commonly used when implemented with fixed-point arithmetic, because they make it possible to distribute the gain of the system and thereby control the size of signals at various critical points in the system. When floating-point arithmetic is used and dynamic range is not a problem, the four-multiplier second-order sections can be used to decrease the amount of computation. Further simplification results for zeros on the unit circle. In this case, $\tilde{b}_{2k} = 1$, and we require only three multipliers per second-order section.

6.3.3 Parallel Form

As an alternative to factoring the numerator and denominator polynomials of $H(z)$, we can express a rational system function as given by Eq. (6.27) or (6.29) as a partial fraction expansion in the form

$$H(z) = \sum_{k=0}^{N_p} C_k z^{-k} + \sum_{k=1}^{N_1} \frac{A_k}{1 - c_k z^{-1}} + \sum_{k=1}^{N_2} \frac{B_k(1 - e_k z^{-1})}{(1 - d_k z^{-1})(1 - d_k^* z^{-1})}, \tag{6.34}$$

where $N = N_1 + 2N_2$. If $M \geq N$, then $N_p = M - N$; otherwise, the first summation in Eq. (6.34) is not included. If the coefficients a_k and b_k are real in Eq. (6.27), then the quantities A_k, B_k, C_k, c_k, and e_k are all real. In this form, the system function can be interpreted as representing a parallel combination of first- and second-order IIR systems, with possibly N_p simple scaled delay paths. Alternatively, we may group the real poles in pairs, so that $H(z)$ can be expressed as

$$H(z) = \sum_{k=0}^{N_p} C_k z^{-k} + \sum_{k=1}^{N_s} \frac{e_{0k} + e_{1k}z^{-1}}{1 - a_{1k}z^{-1} - a_{2k}z^{-2}}, \tag{6.35}$$

where, as in the cascade form, $N_s = \lfloor (N+1)/2 \rfloor$ is the largest integer contained in $(N+1)/2$, and if $N_p = M - N$ is negative, the first sum is not present. A typical example for $N = M = 6$ is shown in Figure 6.20. The general difference equations for the parallel form with second-order direct form II sections are

$$w_k[n] = a_{1k}w_k[n-1] + a_{2k}w_k[n-2] + x[n], \quad k = 1, 2, \ldots, N_s, \tag{6.36a}$$

$$y_k[n] = e_{0k}w_k[n] + e_{1k}w_k[n-1], \quad k = 1, 2, \ldots, N_s, \tag{6.36b}$$

$$y[n] = \sum_{k=0}^{N_p} C_k x[n-k] + \sum_{k=1}^{N_s} y_k[n]. \tag{6.36c}$$

If $M < N$, then the first summation in Eq. (6.36c) is not included.

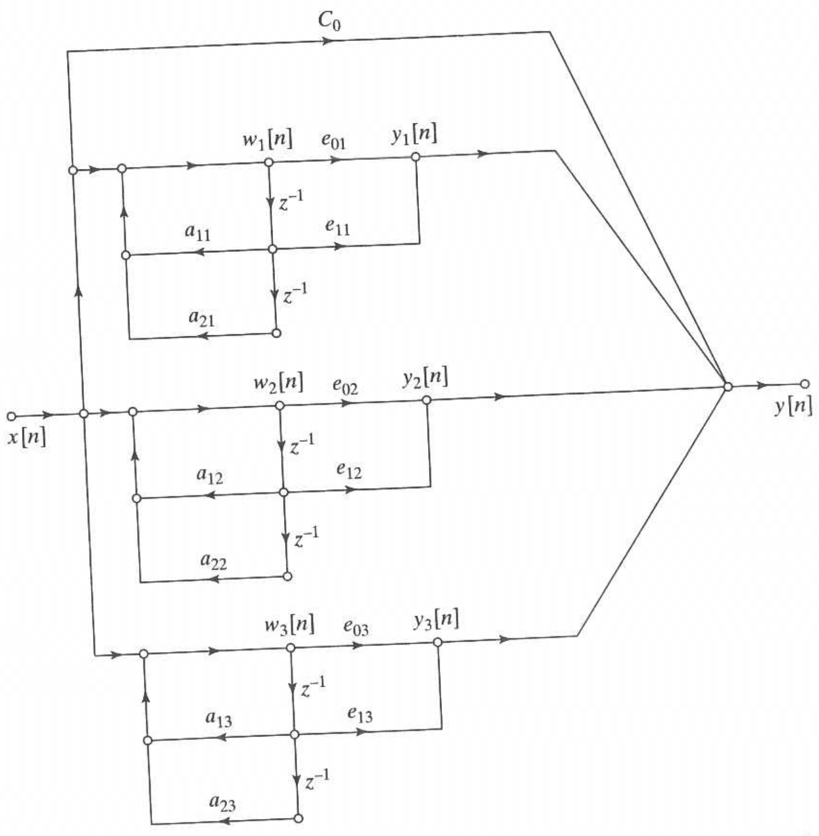

Figure 6.20 Parallel-form structure for sixth-order system ($M = N = 6$) with the real and complex poles grouped in pairs.

Example 6.6 Illustration of Parallel-Form Structures

Consider again the system function used in Examples 6.4 and 6.5. For the parallel form, we must express $H(z)$ in the form of either Eq. (6.34) or Eq. (6.35). If we use second-order sections,

$$H(z) = \frac{1 + 2z^{-1} + z^{-2}}{1 - 0.75z^{-1} + 0.125z^{-2}} = 8 + \frac{-7 + 8z^{-1}}{1 - 0.75z^{-1} + 0.125z^{-2}}. \tag{6.37}$$

The parallel-form realization for this example with a second-order section is shown in Figure 6.21.

Since all the poles are real, we can obtain an alternative parallel form realization by expanding $H(z)$ as

$$H(z) = 8 + \frac{18}{1 - 0.5z^{-1}} - \frac{25}{1 - 0.25z^{-1}}. \tag{6.38}$$

The resulting parallel form with first-order sections is shown in Figure 6.22. As in the general case, the difference equations represented by both Figures 6.21 and 6.22 can be written down by inspection.

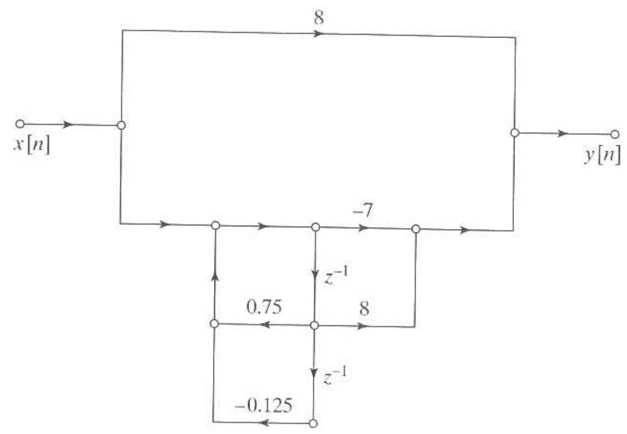

Figure 6.21 Parallel-form structure for Example 6.6 using a second-order system.

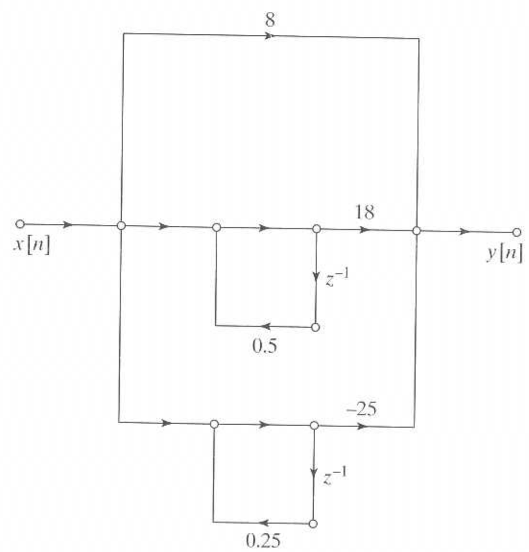

Figure 6.22 Parallel-form structure for Example 6.6 using first-order systems.

6.3.4 Feedback in IIR Systems

All the flow graphs of this section have feedback loops; i.e., they have closed paths that begin at a node and return to that node by traversing branches only in the direction of their arrowheads. Such a structure in the flow graph implies that a node variable in a loop depends directly or indirectly on itself. A simple example is shown in Figure 6.23(a), which represents the difference equation

$$y[n] = ay[n-1] + x[n]. \tag{6.39}$$

Such loops are necessary (but not sufficient) to generate infinitely long impulse responses. This can be seen if we consider a network with no feedback loops. In such a