CHAPTER **8**

IIR Filters

This chapter introduces a new class of LTI systems that have infinite duration impulse responses. For this reason, systems of this class are often called infinite-impulse-response (IIR) systems or IIR filters. In contrast to FIR filters, IIR digital filters involve previously computed values of the output signal as well as values of the input signal in the computation of the present output. Since the output is "fed back" to be combined with the input, these systems are examples of the general class of ***feedback systems***. From a computational point of view, since output samples are computed in terms of previously computed values of the output, the term ***recursive filter*** is also used for these filters.

The z-transform system functions for IIR filters are rational functions that have both poles and zeros at nonzero locations in the z-plane. Just as for the FIR case, we will show that many insights into the important properties of IIR filters can be obtained directly from the pole-zero representation.

We begin this chapter with the first-order IIR system, which is the simplest case because it involves feedback of only the previous output sample. We show by construction that the impulse response of this system has an infinite duration. Then the frequency response and the z-transform are developed for the first-order filter. After showing the relationship among the three domains of representation for this simple case, we consider second-order filters. These filters are particularly important because they can be used to model ***resonances*** such as would occur in a speech synthesizer, as well as many other natural phenomena that exhibit vibratory behavior. The frequency response for the second-order case can exhibit a narrowband character that leads to the definition of bandwidth and center frequency, both of which can be controlled by appropriate choice of the feedback coefficients of the filter. The analysis and insights developed for the first- and second-order cases are readily generalized to higher-order systems.

8-1 The General IIR Difference Equation

FIR filters are extremely useful and have many nice properties, but that class of filters is not the most general class of LTI systems. This is because the output $y[n]$ is formed solely from a finite segment of the input signal $x[n]$. The most general class of digital filters that can be implemented with a finite amount of computation is obtained when the output is formed not only from the input, but also from previously computed outputs.

DEMO: *IIR Filtering*

The defining equation for this class of digital filters is the difference equation

$$y[n] = \sum_{\ell=1}^{N} a_\ell y[n-\ell] + \sum_{k=0}^{M} b_k x[n-k] \tag{8.1}$$

The filter coefficients consist of two sets: $\{b_k\}$ and $\{a_\ell\}$. For reasons that will become obvious in the following simple example, the coefficients $\{a_\ell\}$ are called the ***feedback*** coefficients, and the $\{b_k\}$ are called the ***feed-forward*** coefficients. In all, $N+M+1$ coefficients are needed to define the recursive difference equation (8.1).

Notice that if the coefficients $\{a_\ell\}$ are all zero, the difference equation (8.1) reduces to the difference equation of an FIR system. Indeed, we have asserted that (8.1) defines the most general class of LTI systems that can be implemented with finite computation, so FIR systems must be a special case. When discussing FIR systems we have referred to M as the ***order*** of the system. In this case, M is the number of delay terms in the difference equation and the degree or order of the polynomial system function. For IIR systems, we have both M and N as measures of the number of delay terms, and we will see that the system function of an IIR system is the ratio of an M^{th}-order polynomial to an N^{th}-order polynomial. Thus, there can be some ambiguity as to the order of an IIR system. In general, we will define N, the number of feedback terms, to be the order of an IIR system.

Example 8-1: IIR Block Diagram

As a first step toward understanding the general form given in (8.1), consider the first-order case where $M = N = 1$; i.e.,

$$y[n] = a_1 y[n-1] + b_0 x[n] + b_1 x[n-1] \tag{8.2}$$

The block diagram representation of this difference equation, which is shown in Fig. 8-1, is constructed by noting that the signal $v[n] = b_0 x[n] + b_1 x[n-1]$ is computed by the left half of the diagram, and we "close the loop" by computing $a_1 y[n-1]$ from the delayed output and adding it to $v[n]$ to produce the output $y[n]$. This diagram clearly shows that the terms feed-forward and feedback describe the direction of signal flow in the block diagram. ■

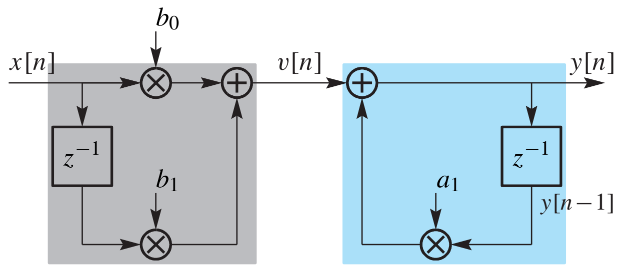

Figure 8-1: First-order IIR system showing one feedback coefficient a_1 and two feed-forward coefficients b_0 and b_1.

We will begin by studying a simplified version of the system defined by (8.2) and depicted in Fig. 8-1. This will involve characterizing the filter in each of the three domains: time domain, frequency domain, and z-domain. Since the filter is defined by a time-domain difference equation (8.2), we begin by studying how the difference equation is used to compute the output from the input, and we will illustrate how feedback results in an impulse response of infinite duration.

8-2 Time-Domain Response

To illustrate how the difference equation can be used to implement an IIR system, we will begin with a numerical example. Assume that the filter coefficients in (8.2) are $a_1 = 0.8$, $b_0 = 5$, and $b_1 = 0$, so that

$$y[n] = 0.8y[n-1] + 5x[n] \tag{8.3}$$

and assume that the input signal is

$$x[n] = 2\delta[n] - 3\delta[n-1] + 2\delta[n-3] \tag{8.4}$$

In other words, the total duration of the input is four samples, as shown in Fig. 8-2.

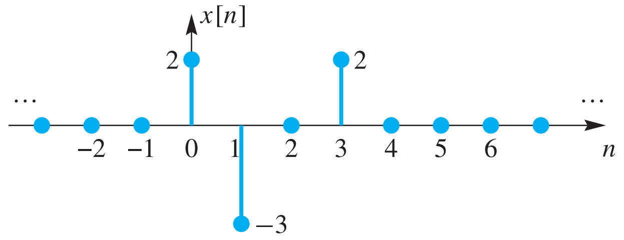

Figure 8-2: Input signal to recursive difference equation.

Although it is not a requirement, it is logical that the output sequence values should be computed in normal order (i.e., from left to right in a plot of the sequence). Furthermore, since the input is zero before $n = 0$, it would be natural to assume that $n = 0$ is the starting time of the output. Thus, we will consider computing the output from the difference equation (8.3) in the order $n = 0, 1, 2, 3, \ldots$. For example, the value of $x[0]$ is 2, so we can evaluate (8.3) at $n = 0$ obtaining

$$y[0] = 0.8y[0-1] + 5x[0] = 0.8y[-1] + 5(2) \tag{8.5}$$

Immediately we run into a problem: The value of $y[n]$ at $n = -1$ is not known. This is a serious problem, because no matter where we start computing the output, we will always have the same problem; at any point along the n-axis, we will need to know the output at the previous time $n - 1$. If we know the value $y[n-1]$, however, we can use the difference equation to compute the next value of the output signal at time n. Once the process is started, it can be continued indefinitely by iteration of the difference equation. The solution requires the following two assumptions, which together are called the ***initial rest conditions***.

Initial Rest Conditions

1. The input must be assumed to be zero prior to some starting time n_0; i.e., $x[n] = 0$ for $n < n_0$. We say that such inputs are *suddenly applied*.
2. The output is likewise assumed to be zero prior to the starting time of the signal; i.e., $y[n] = 0$ for $n < n_0$. We say that the system is *initially at rest* if its output is zero prior to the application of a suddenly applied input.

These conditions are not particularly restrictive, especially in the case of a real-time system, where a new output must be computed as each new sample of the input is taken. Real-time systems must, of course, be

causal in the sense that the computation of the present output sample must not involve future samples of the input or output. Furthermore, any practical device would have a time at which it first begins to operate. All that is needed is for the memory containing the delayed output samples to be set initially to zero.[1]

With the initial rest assumption, we let $y[n] = 0$ for $n < 0$, so now we can evaluate $y[0]$ as

$$y[0] = 0.8y[-1] + 5(2) = 0.8(0) + 5(2) = 10$$

Once we have started the recursion, the rest of the values follow easily, since the input signal and previous outputs are known.

$$\begin{aligned} y[1] &= 0.8y[0] + 5x[1] = 0.8(10) + 5(-3) = -7 \\ y[2] &= 0.8y[1] + 5x[2] = 0.8(-7) + 5(0) = -5.6 \\ y[3] &= 0.8y[2] + 5x[3] = 0.8(-5.6) + 5(2) = 5.52 \\ y[4] &= 0.8y[3] + 5x[4] = 0.8(5.52) + 5(0) = 4.416 \\ y[5] &= 0.8y[4] + 5x[5] = 0.8(4.416) + 0 = 3.5328 \\ &\vdots \qquad\qquad \vdots \end{aligned}$$

This output sequence is plotted in Fig. 8-3 up to $n = 7$.

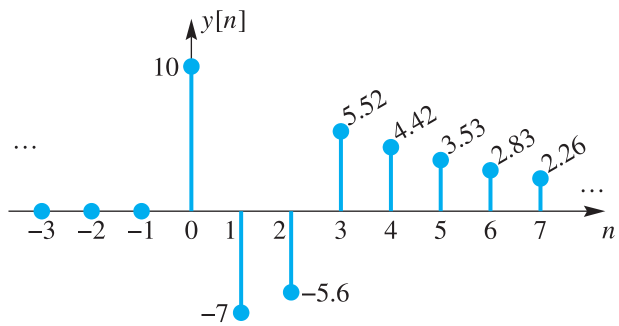

Figure 8-3: Output signal from recursive difference equation of (8.5) for the input of Fig. 8-2. For $n > 3$, the sequence is proportional to $(0.8)^n$, because the input signal ends at $n = 3$.

One key feature to notice in Fig. 8-3 is the structure of the output signal after the input turns off $(n > 3)$. For this range of n, the difference equation becomes

$$y[n] = 0.8y[n-1] \qquad\qquad \text{for } n > 3$$

Thus the ratio between successive terms is a constant, and the output signal decays exponentially with a rate determined by $a_1 = 0.8$. Therefore, we can write the closed form expression

$$y[n] = y[3](0.8)^{n-3} \qquad\qquad \text{for } n \geq 3$$

for the rest of the sequence $y[n]$ once the value for $y[3]$ is known.

[1] In the case of a digital filter applied to sampled data stored in computer memory, the causality condition is not required, but, generally, the output is computed in the same order as the natural order of the input samples. The difference equation could be *recursed* backwards through the sequence, but this would require a different definition of initial conditions.

8-2.1 Linearity and Time Invariance of IIR Filters

When applied to the general IIR difference equation of (8.1), the condition of initial rest is sufficient to guarantee that the system implemented by iterating the difference equation is both linear and time-invariant. Although the feedback terms make the proof more complicated than the FIR case (see Section 5-5.3 on p. 153), we can show that, for suddenly applied inputs and initial rest conditions, the principle of superposition will hold because the difference equation involves only linear combinations of the input and output samples. Furthermore, since the initial rest condition is always applied just before the beginning of a suddenly applied input, time invariance also holds.

EXERCISE 8.1: Assume that the input to the difference equation (8.3) is $x_1[n] = 10x[n-4]$, where $x[n]$ is given by (8.4) and Fig. 8-2. Use iteration to compute the corresponding output $y_1[n]$ for $n = 0, 1, \ldots, 11$ using the assumption of initial rest. Compare your result to the output plotted in Fig. 8-3, and verify that the system behaves as if it is both linear and time-invariant.

8-2.2 Impulse Response of a First-Order IIR System

In Chapter 5, we showed that the response to a unit impulse sequence characterizes a linear time-invariant system completely. Recall that when $x[n] = \delta[n]$, the resulting output signal, denoted by $h[n]$, is by definition the ***impulse response***. Since all other input signals can be written as a superposition of weighted and delayed impulses, the corresponding output for all other signals can be constructed from weighted and shifted versions of the impulse response, $h[n]$. That is, since the recursive difference equation with initial rest conditions is an LTI system, its output can always be represented as the convolution sum

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] \tag{8.6}$$

Therefore, it is of interest to characterize the recursive difference equation by its impulse response.

To illustrate the nature of the impulse response of an IIR system, consider the first-order recursive difference equation with $b_1 = 0$,

$$y[n] = a_1 y[n-1] + b_0 x[n]. \tag{8.7}$$

By definition, the difference equation

$$h[n] = a_1 h[n-1] + b_0 \delta[n] \tag{8.8}$$

must be satisfied by the impulse response $h[n]$ for all values of n. We can construct a general formula for the impulse response in terms of the parameters a_1 and b_0 by simply constructing a table of a few values and then writing down the general formula by inspection. The following table shows the sequences involved in the computation:

n	$n<0$	0	1	2	3	4	$\cdots$
$\delta[n]$	0	1	0	0	0	0	$\cdots$
$h[n-1]$	0	0	b_0	$b_0(a_1)$	$b_0(a_1)^2$	$b_0(a_1)^3$	$\cdots$
$h[n]$	0	b_0	$b_0(a_1)$	$b_0(a_1)^2$	$b_0(a_1)^3$	$b_0(a_1)^4$	$\cdots$

From this table, we see that the general formula is

$$h[n] = \begin{cases} b_0(a_1)^n & \text{for } n \geq 0 \\ 0 & \text{for } n < 0 \end{cases} \tag{8.9}$$

If we recall the definition of the unit step sequence,

$$u[n] = \begin{cases} 1 & \text{for } n \geq 0 \\ 0 & \text{for } n < 0 \end{cases} \tag{8.10}$$

(8.9) can be expressed in the form

$$h[n] = b_0(a_1)^n u[n] \tag{8.11}$$

where multiplication of $(a_1)^n$ by $u[n]$ provides a compact representation of the conditions $n < 0$ and $n \geq 0$.

Example 8-2: Impulse Response

For the example of (8.3) with $a_1 = 0.8$ and $b_0 = 5$, the impulse response is

$$h[n] = 5(0.8)^n u[n] = \begin{cases} 5(0.8)^n & \text{for } n \geq 0 \\ 0 & \text{for } n < 0 \end{cases} \tag{8.12}$$

This is the impulse response of the system in (8.3). ■

EXERCISE 8.2: Substitute the solution (8.11) into the difference equation (8.8) and verify that the difference equation is satisfied for all values of n.

EXERCISE 8.3: Determine the impulse response of the following first-order system:

$$y[n] = 0.5y[n-1] + 5x[n-7]$$

Assume that the system is ***at rest*** for $n < 0$. Plot the resulting signal $h[n]$ as a function of n. Pay careful attention to where the nonzero portion of the impulse response begins.

A slightly more general problem would be to determine the impulse response of the first-order system when a shifted version of the input signal is also included in the difference equation; i.e.,

$$y[n] = a_1 y[n-1] + b_0 x[n] + b_1 x[n-1]$$

Because this system is linear and time-invariant, it follows that its impulse response can be thought of as a sum of two terms as in

$$\begin{aligned} h[n] &= b_0(a_1)^n u[n] + b_1(a_1)^{n-1} u[n-1] \\ &= \begin{cases} 0 & n < 0 \\ b_0 & n = 0 \\ (b_0 + b_1 a_1^{-1})(a_1)^n & n \geq 1 \end{cases} \end{aligned}$$

Notice that the impulse response still decays exponentially with rate dependent only on a_1.

EXERCISE 8.4: Determine the impulse response $h[n]$ of the following first-order system:

$$y[n] = -0.5y[n-1] - 4x[n] + 5x[n-1].$$

Plot the resulting impulse response as a function of n.

8-2.3 Response to Finite-Length Inputs

For finite-length inputs, the convolution sum is easy to evaluate for either FIR or IIR systems. Suppose that the finite-length input sequence is

$$x[n] = \sum_{k=N_1}^{N_2} x[k]\delta[n-k]$$

so that $x[n] = 0$ for $n < N_1$ and $n > N_2$. Then it follows from (8.6) that the corresponding output satisfies

$$y[n] = \sum_{k=N_1}^{N_2} x[k]h[n-k]$$

Example 8-3: IIR Response to General Input

As an example, consider again the LTI system defined by the difference equation (8.3), whose impulse response was shown in Example 8-2 to be $h[n] = 5(0.8)^n u[n]$. For the input of (8.4) and Fig. 8-2,

$$x[n] = 2\delta[n] - 3\delta[n-1] + 2\delta[n-3]$$

it is easily seen that

$$\begin{aligned} y[n] &= 2h[n] - 3h[n-1] + 2h[n-3] \\ &= 10(0.8)^n u[n] - 15(0.8)^{n-1}u[n-1] \\ &\quad + 10(0.8)^{n-3}u[n-3] \end{aligned}$$

To evaluate this expression for a specific time index, we need to take into account the different regions over which the individual terms are nonzero. If we do, we obtain

$$y[n] = \begin{cases} 0 & n < 0 \\ 10 & n = 0 \\ 10(0.8) - 15 = -7 & n = 1 \\ 10(0.8)^2 - 15(0.8) = -5.6 & n = 2 \\ 10(0.8)^3 - 15(0.8)^2 + 10 = 5.52 & n = 3 \\ 5.52(0.8)^{n-3} & n > 3 \end{cases}$$

A comparison to the output obtained by iterating the difference equation (see Fig. 8-3 on p. 255) shows that we have obtained the same values of the output sequence by superposition of scaled and shifted impulse responses.

Example 8-3 illustrates two important points about IIR systems.

1. The initial rest condition guarantees that the output sequence does not begin until the input sequence begins (or later).
2. Because of the feedback, the impulse response is infinite in extent, and the output due to a finite-length input sequence, being a superposition of scaled and shifted impulse responses, is generally (but not always) infinite in extent. This is in contrast to the FIR case, where a finite-length input always produces a finite-length output sequence.

EXERCISE 8.5: Find the impulse response of the first-order system

$$y[n] = -0.5y[n-1] + 5x[n]$$

and use it to determine the output due to the input signal

$$x[n] = \begin{cases} 0 & n < 1 \\ 3 & n = 1 \\ -2 & n = 2 \\ 0 & n = 3 \\ 3 & n = 4 \\ -1 & n = 5 \\ 0 & n > 5 \end{cases}$$

Write a formula that is the sum of four terms, each of which is a shifted impulse response. Assume the initial rest condition. Plot the resulting signal $y[n]$ as a function of n for $0 \le n \le 10$.

8-2.4 Step Response of a First-Order Recursive System

When the input signal is infinitely long, the computation of the output of an IIR system using the difference equation is no different than for an FIR system; we simply continue to iterate the difference equation as long as samples of the output are desired. In the FIR case, the difference equation and the convolution sum are the same thing. This is not true in the IIR case, and computing the output using convolution is practical only in cases where simple formulas exist for both the input and the impulse response. Thus, in general, IIR filters must be implemented by iterating the difference equation. The computation of the response of a first-order IIR system to a unit step input provides a relatively simple illustration of the issues involved.

Again, assume that the system is defined by

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

and assume that the input is the unit step sequence given by

$$u[n] = \begin{cases} 1 & \text{for } n \geq 0 \\ 0 & \text{for } n < 0 \end{cases} \tag{8.13}$$

As before, the difference equation can be iterated to produce the output sequence one sample at a time. The first few values are tabulated here. Work through the table to be sure that you understand the computation.

n	$x[n]$	$y[n]$
$n < 0$	0	0
0	1	b_0
1	1	$b_0 + b_0(a_1)$
2	1	$b_0 + b_0(a_1) + b_0(a_1)^2$
3	1	$b_0(1 + a_1 + a_1^2 + a_1^3)$
4	1	$b_0(1 + a_1 + a_1^2 + a_1^3 + a_1^4)$
$\vdots$	1	$\vdots$

From the tabulated values, it can be seen that a general formula for $y[n]$ is

$$\begin{aligned} y[n] &= b_0(1 + a_1 + a_1^2 + \cdots + a_1^n) \\ &= b_0 \sum_{k=0}^{n} a_1^k \end{aligned} \tag{8.14}$$

With a bit of manipulation, we can get a simple closed-form expression for the general term in the sequence $y[n]$. For this we need to recall the formula

$$\sum_{k=0}^{L} r^k = \begin{cases} \dfrac{1 - r^{L+1}}{1 - r} & r \neq 1 \\ L + 1 & r = 1 \end{cases} \tag{8.15}$$

which is the formula for summing the first $L+1$ terms of a geometric series. Armed with this fact, the formula (8.14) for $y[n]$ (when $a_1 \neq 1$) can be manipulated into the form

$$y[n] = b_0 \frac{1 - a_1^{n+1}}{1 - a_1} \qquad \text{for } n \geq 0, \tag{8.16}$$

Three cases can be identified: $|a_1| > 1$, $|a_1| < 1$, and $|a_1| = 1$. Further investigation of these cases reveals two types of behavior.

1. When $|a_1| > 1$, the term a_1^{n+1} in the numerator will dominate and the values for $y[n]$ will get larger and larger without bound. This is called an ***unstable*** condition and is usually a situation to avoid. We will say more about the issue of stability later in Sections 8-5.2 and 8-9.

2. When $|a_1| < 1$, the term a_1^{n+1} will decay to zero as $n \to \infty$. In this case, the system is ***stable***. Therefore, we can determine a limiting value for $y[n]$ as $n \to \infty$

$$\lim_{n\to\infty} y[n] = \lim_{n\to\infty} b_0 \frac{1 - a_1^{n+1}}{1 - a_1} = \frac{b_0}{1 - a_1}$$

3. When $|a_1| = 1$, we might have an unbounded output, but not always. For example, when $a_1 = 1$, (8.14) gives $y[n] = (n+1)b_0$ for $n \geq 0$, and the output $y[n]$ grows as $n \to \infty$. On the other hand, for $a_1 = -1$, the output alternates; it is $y[n] = b_0$ for n even, but $y[n] = 0$ for n odd.

The MATLAB plot in Fig. 8-4 shows the step response for the filter

$$y[n] = 0.8y[n-1] + 5x[n]$$

Notice that the limiting value is 25, which can be calculated from the filter coefficients

$$\lim_{n\to\infty} y[n] = \frac{b_0}{1 - a_1} = \frac{5}{1 - 0.8} = 25$$

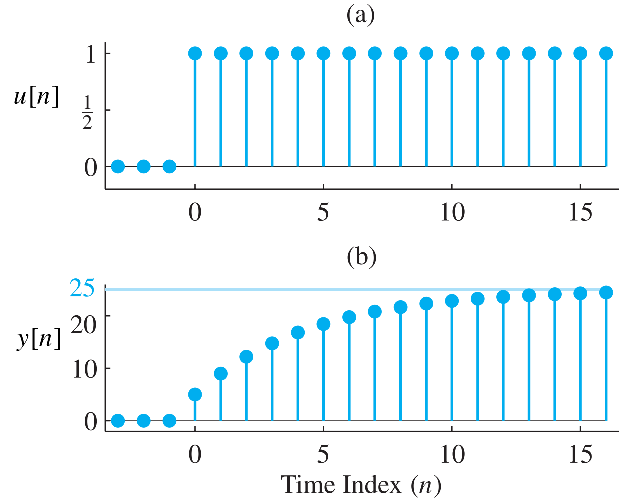

Figure 8-4: (a) Unit-step signal $u[n]$, (b) Step response of a first-order IIR filter.

Now suppose that we try to compute the step response by the convolution sum

$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] \tag{8.17}$$

Since both the input and the impulse response have infinite durations, we might have difficulty in carrying out the computation. However, the fact that the input and output are given by the formulas $x[n] = u[n]$ and $h[n] = b_0(a_1)^n u[n]$ makes it possible to obtain a result. Substituting these formulas into (8.17) gives

$$y[n] = \sum_{k=-\infty}^{\infty} u[k]b_0(a_1)^{n-k}u[n-k]$$

The $u[k]$ and $u[n-k]$ terms inside the sum will change the limits on the sum because $u[k] = 0$ for $k < 0$ and $u[n-k] = 0$ for $n - k < 0$ (or $n < k$). The final result is

$$y[n] = \begin{cases} 0 & \text{for } n < 0 \\ \sum\limits_{k=0}^{n} b_0 (a_1)^{n-k} & \text{for } n \geq 0 \end{cases}$$

Using (8.15) we can write the step response for $n \geq 0$ as

$$\begin{aligned} y[n] &= \sum_{k=0}^{n} b_0 (a_1)^{n-k} = b_0 (a_1)^n \sum_{k=0}^{n} (a_1)^{-k} \\ &= b_0 (a_1)^n \frac{1 - (1/a_1)^{n+1}}{1 - (1/a_1)} \\ &= b_0 \frac{1 - (a_1)^{n+1}}{1 - a_1} \end{aligned} \tag{8.18}$$

which is identical to (8.16), the step response computed by iterating the difference equation. Notice that we were able to arrive at a closed-form expression in this case because of the special nature of the input and impulse response. In general, it would be difficult or impossible to obtain such a closed-form result, but we can always use iteration of the difference equation to compute the output sample-by-sample.

8-3 System Function of an IIR Filter

We saw in Chapter 7 for the FIR case that the system function is the z-transform of the impulse response of the system, and that the system function and the frequency response are intimately related. Furthermore, the following result was shown:

Convolution in the n-domain corresponds to multiplication in the z-domain.

$$y[n] = x[n] * h[n] \quad \overset{z}{\longleftrightarrow} \quad Y(z) = X(z)H(z)$$

The same is true for the IIR case. The system function for an FIR filter is always a polynomial; however, when the difference equation has feedback, it turns out that the system function $H(z)$ is the ratio of two polynomials. Ratios of polynomials are called ***rational*** functions. In this section we will determine the system function for the example of a first-order IIR system, and show how the system function, impulse response, and difference equation are related.

8-3.1 The General First-Order Case

The general form of the first-order difference equation with feedback is

$$y[n] = a_1 y[n-1] + b_0 x[n] + b_1 x[n-1] \tag{8.19}$$

Since this equation must be satisfied for all values of n, we can use the property (7.13) on p. 216 to determine the z-transform of both sides of the equation and obtain the following equation relating the z-transform of the output to the z-transform of the input:

$$Y(z) = a_1 z^{-1} Y(z) + b_0 X(z) + b_1 z^{-1} X(z)$$

Subtracting the term $a_1 z^{-1} Y(z)$ from both sides of the equation leads to the following manipulations:

$$Y(z) - a_1 z^{-1} Y(z) = b_0 X(z) + b_1 z^{-1} X(z)$$
$$(1 - a_1 z^{-1}) Y(z) = (b_0 + b_1 z^{-1}) X(z)$$

Since the system is an LTI system, it should be true that $Y(z) = H(z)X(z)$, where $H(z)$ is the system function of the LTI system. Solving this equation for $H(z) = Y(z)/X(z)$, we obtain

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1}}{1 - a_1 z^{-1}} = \frac{B(z)}{A(z)} \tag{8.20}$$

Thus, we have shown that $H(z)$ for the first-order IIR system is a ratio of two polynomials. The numerator polynomial $B(z)$ is defined by the weighting coefficients $\{b_k\}$ that multiply the input signal $x[n]$ and its delayed versions; the denominator polynomial $A(z)$ is defined by the feedback coefficients $\{a_\ell\}$. That this correspondence is true in general should be clear from the analysis that leads to the formula for $H(z)$. Indeed, the following is true for IIR systems of *any* order:

> *The coefficients of the* ***numerator*** *polynomial of the system function of an IIR system are the coefficients of the* ***feed-forward*** *terms of the difference equation. For the* ***denominator*** *polynomial, the constant term is one, and the remaining coefficients are the negatives of the* ***feedback*** *coefficients.*

In other words, if the system function is

$$H(z) = \frac{\displaystyle\sum_{k=0}^{M} b_k z^{-k}}{1 - \displaystyle\sum_{k=1}^{N} a_k z^{-k}} \tag{8.21a}$$

then the corresponding z-transform is

$$y[n] - \sum_{k=1}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k] \tag{8.21b}$$

The correspondences are evident from a comparison of (8.21a) and 11.84.

In MATLAB, the `filter` function follows this same format. The statement

```
yy = filter(bb,aa,xx)
```

implements an IIR filter, where the vectors `bb` and `aa` hold the filter coefficients for the numerator and denominator polynomials, respectively.

Example 8-4: MATLAB for IIR Filter

The following feedback filter:

$$y[n] = 0.5y[n-1] - 3x[n] + 2x[n-1]$$

would be implemented in MATLAB by

```
yy = filter([-3,2], [1,-0.5], xx)
```

where `xx` and `yy` are the input and output signal vectors, respectively. Notice that the `aa` vector has $-a_1$ for its second element, just like in the polynomial $A(z)$. We can imagine that the filter coefficient multiplying $y[n]$ is 1, so we always have `1` for the first element of `aa`.

EXERCISE 8.6: Find the system function (i.e., z-transform) of the following feedback filter:

$$y[n] = 0.5y[n-1] - 3x[n] + 2x[n-1]$$

EXERCISE 8.7: Determine the system function of the system implemented by the following MATLAB statement:

```
yy = filter(5, [1,0.8], xx).
```

8-3.2 The System Function and Block-Diagram Structures

As we have seen, the system function displays the coefficients of the difference equation in a convenient way that makes it easy to move back and forth between the difference equation and the system function. In this section, we will show that this makes it possible to derive other difference equations, and thus other implementation equations, by simply manipulating the system function.

8-3.2.1 Direct Form I Structure

To illustrate the connection between the system function and the block diagram, let us return to the block diagram of Fig. 8-1, which is repeated in Fig. 8-5 for convenience. Block diagrams such as Fig. 8-5 are called *implementation structures*, or, more commonly, simply ***structures***, because they give a pictorial representation of the difference equations that can be used to implement the system.

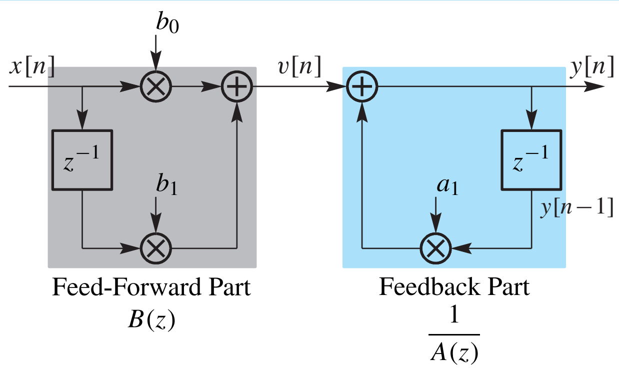

Figure 8-5: First-order IIR system showing one feedback coefficient a_1 and two feed-forward coefficients b_0 and b_1 in Direct Form I structure.

Recall that the product of two z-transform system functions corresponds to the cascade of two systems. The system function for the first-order feedback filter can be factored into an FIR piece and an IIR piece, as in

$$\begin{aligned} H(z) &= \frac{b_0 + b_1 z^{-1}}{1 - a_1 z^{-1}} \\ &= \left(\frac{1}{1 - a_1 z^{-1}}\right)\left(b_0 + b_1 z^{-1}\right) = \left(\frac{1}{A(z)}\right) \cdot B(z) \end{aligned}$$

The conclusion to be drawn from this algebraic manipulation is that a valid implementation for $H(z)$ is the pair of difference equations

$$v[n] = b_0 x[n] + b_1 x[n-1] \tag{8.22a}$$

$$y[n] = a_1 y[n-1] + v[n] \tag{8.22b}$$

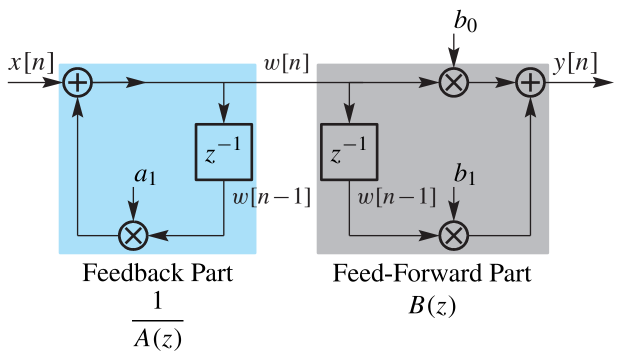

Figure 8-6: First-order IIR system showing one feedback coefficient a_1 and two feed-forward coefficients b_0 and b_1 in a Direct Form II structure.

We see in Fig. 8-5 that the polynomial $B(z)$ is the system function of the feed-forward part of the block diagram, and that $1/A(z)$ is the system function of a feedback part that completes the system. The system implemented in this way is called the ***Direct Form I*** implementation because it is possible to go directly from the system function to this block diagram (or the difference equations that it represents) with no other manipulations than to write the numerator and denominator as polynomials in the variable z^{-1}.

8-3.2.2 Direct Form II Structure

We know that for an LTI cascade system, we can change the order of the systems without changing the overall system response. In other words,

$$H(z) = \left(\frac{1}{A(z)}\right) \cdot B(z) = B(z) \cdot \left(\frac{1}{A(z)}\right)$$

Using the correspondences that we have established, leads to the block diagram shown in Fig. 8-6. Note that we have defined a new intermediate variable $w[n]$ as the output of the feedback part and input to the feed-forward part. Thus, the block diagram tells us that an equivalent implementation of the system is

$$w[n] = a_1 w[n-1] + x[n] \tag{8.23a}$$

$$y[n] = b_0 w[n] + b_1 w[n-1] \tag{8.23b}$$

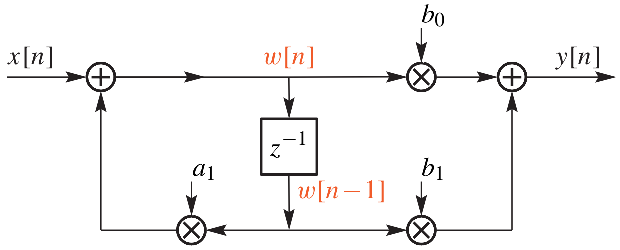

Figure 8-7: First-order IIR system in Direct Form II structure. This is identical to Fig. 8-6, except that the two delays have been merged into one.

Again, because there is such a direct and simple correspondence between Fig. 8-6 and $H(z)$, this implementation is called the ***Direct Form II*** implementation of the first-order IIR system with system function

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

The block-diagram representation of Fig. 8-6 leads to a valuable insight. Notice that the input to each of the unit delay operators is the same signal $w[n]$. Thus, there is no need for two delay operations; they can be combined into a single delay, as in Fig. 8-7. Since delay operations are implemented with memory in a computer, the implementation of Fig. 8-7 would require less memory that the implementation of Fig. 8-6. Note, however, that both block diagrams represent the difference equations (8.23a) and (8.23b).

EXERCISE 8.8: Find the z-transform system function of the following set of cascaded difference equations:

$$w[n] = -0.5w[n-1] + 7x[n]$$
$$y[n] = 2w[n] - 4w[n-1]$$

Draw the block diagrams of this system in both Direct Form I and Direct Form II.

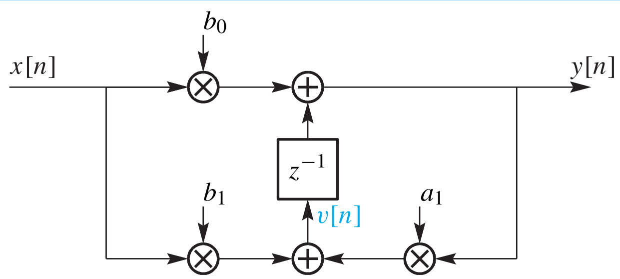

Figure 8-8: Computational structure for a general first-order IIR filter as a transposed Direct Form II structure.

8-3.2.3 The Transposed Form Structure

A somewhat surprising fact about block diagrams like Fig. 8-7 is that if the block diagram undergoes the following transformation:

1. All the arrows are reversed with multipliers being unchanged in value or location.
2. All branch points become summing points, and all summing points become branch points.
3. The input and the output are interchanged.

then the overall system has the same system function as the original system. We will not prove this, but it is true for the kinds of block diagrams that we have just introduced. However, we can use the z-transform to verify that this is true for our simple first-order system.

The feedback structure given in the signal-flow graph of Fig. 8-8 is the ***transposed*** form of the Direct Form II structure shown in Fig. 8-7. In order to derive the actual difference equations, we need to write the equations that are defined by the signal-flow graph. There is an orderly procedure for doing this if we follow two rules:

1. Assign variable names to the inputs of all delay elements. For example, $v[n]$ is used in Fig. 8-8, so the output of the delay is $v[n-1]$.
2. Write equations at all of the summing nodes; there are two in this case.

$$y[n] = b_0 x[n] + v[n-1] \tag{8.24a}$$
$$v[n] = b_1 x[n] + a_1 y[n] \tag{8.24b}$$

The signal-flow graph specifies an actual computation, so (8.24a) and (8.24b) require three multiplications and two additions at each time step n. Equation (8.24a) must be done first, because $y[n]$ is needed in (8.24b).

Owing to the feedback, it is impossible to manipulate these n-domain equations into one of the other forms by eliminating variables. However, we can recombine these two equations in the z-transform domain to verify that we have the correct system function. First, we take the z-transform of each difference equation obtaining

$$\begin{aligned} Y(z) &= b_0 X(z) + z^{-1} V(z) \\ V(z) &= b_1 X(z) + a_1 Y(z) \end{aligned}$$

Now we eliminate $V(z)$ by substituting the second equation into the first as follows:

$$\begin{aligned} Y(z) &= b_0 X(z) + z^{-1}(b_1 X(z) + a_1 Y(z)) \\ (1 - a_1 z^{-1}) Y(z) &= (b_0 + b_1 z^{-1}) X(z) \end{aligned}$$

Therefore, we get

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1}}{1 - a_1 z^{-1}}$$

Thus, because they have the same system function, (8.24a) and (8.24b) are equivalent to the Direct Form I (8.22a) and (8.22b), and to the Direct Form II (8.23a) and (8.23b).

Why are these different implementations of the same system function of interest to us? They all use the same number of multiplications and additions to compute exactly the same output from a given input. However, this is true only when the arithmetic is perfect. On a computer with finite precision (e.g., 16-bit words), each calculation will involve round-off errors, which means that each implementation will behave slightly differently. In practice, the implementation of high-quality digital filters in hardware demands correct engineering to control round-off errors and overflows.

8-3.3 Relation to the Impulse Response

In the analysis of Section 8-3.1 we assumed implicitly that the system function is the z-transform of the impulse response of an IIR system. While this is true, we have demonstrated only that it is true for the FIR case. In the IIR case, we need to be able to take the z-transform of an infinitely long sequence. As an example of such a sequence, consider $h[n] = a^n u[n]$. Applying the definition of the z-transform from equation (7.15) on p. 216, we would write

$$H(z) = \sum_{n=0}^{\infty} a^n z^{-n} = \sum_{n=0}^{\infty} (az^{-1})^n$$

which is the sum of all the terms in a geometric series where the ratio between successive terms is az^{-1}. Thus, we know that if $|az^{-1}| < 1$, then the sum is finite, and in fact is given by the closed-form expression

$$H(z) = \sum_{n=0}^{\infty} a^n z^{-n} = \frac{1}{1 - az^{-1}}$$

The condition for the infinite sum to equal the closed-form expression can be expressed as $|a| < z$. The values of z in the complex plane satisfying this condition are called the ***region of convergence***. From the preceding analysis, we can state the following exceedingly important z-transform pair:

$$\boxed{a^n u[n] \quad \overset{z}{\longleftrightarrow} \quad \frac{1}{1 - az^{-1}}} \tag{8.25}$$

We will have many occasions to use this result in this chapter.

Example 8-5: $H(z)$ from Impulse Response

As an example of the use of this result, recall that in Section 8-2.2 we showed by iteration that the impulse response of the system

$$y[n] = a_1 y[n-1] + b_0 x[n] + b_1 x[n-1] \tag{8.26}$$

is

$$h[n] = b_0 (a_1)^n u[n] + b_1 (a_1)^{n-1} u[n-1] \tag{8.27}$$

Thus, using the linearity property of the z-transform, the delay property of the z-transform, and the result of (8.25), the system function for this system is

$$\begin{aligned} H(z) &= b_0 \left(\frac{1}{1 - a_1 z^{-1}} \right) + b_1 z^{-1} \left(\frac{1}{1 - a_1 z^{-1}} \right) \\ &= \frac{b_0 + b_1 z^{-1}}{1 - a_1 z^{-1}} \end{aligned} \tag{8.28}$$

which is what we obtained before in Section 8-3.1 on p. 262 by taking the z-transform of the difference equation and solving for $H(z) = Y(z)/X(z)$. ■

8-3.4 Summary of the Method

In this section, we have illustrated some important analysis techniques. We have seen that it is possible to go from the difference equation (8.26) directly to the system function (8.28). We have also seen that, in this simple example, it is possible by ***taking the inverse z-transform*** to go directly from the system function (8.28) to the impulse response (8.27) of the system without the tedious process of iterating the difference equation. We will see that it is possible to do this, in general, by a process of inverse z-transformation based on breaking the z-transform into a sum of terms like the right-hand side of (8.25). Before developing this technique, which will be applicable to higher-order systems, we will continue to focus on the first-order IIR system to illustrate some more important points about the z-transform and its relation to IIR systems.

8-4 Stability and the Region of Convergence of $H(z)$

Because IIR systems have impulse responses of infinite duration due to the feedback inherent in the difference equations used for their implementation, it is important that the response not "blow up" as the difference equation is iterated indefinitely. This requires that the system be *stable*. The definition of stability that is commonly employed is that the output of a stable system can always be bounded such that $|y[n]| < M_y < \infty$ for all n for any input that is bounded as $|x[n]| < M_x < \infty$. The finite constants M_x and M_y can be different. This definition for stability is called *bounded-input, bounded-output* or BIBO stability; it is a mathematically careful way of saying that "the output doesn't blow up".) To see what this definition applies, note that for an LTI system, the output is related to the input by convolution. In the case of a causal IIR filter, this implies that

$$y[n] = \sum_{m=0}^{\infty} h[m]x[n-m]$$

Therefore, a bound on the size of $|y[n]|$ can be obtained as follows:

$$|y[n]| \leq \sum_{m=0}^{\infty} |h[m]||x[n-m]| < M_x \sum_{m=0}^{\infty} |h[m]| \tag{8.29}$$

where we have used the fact that the magnitude of a sum of terms is less than or equal to the sum of the magnitudes of those terms, and we strengthened the inequality by replacing $|x[n-m]|$ for all m by the bound M_x. Therefore a sufficient condition[2] for $|y[n]| < M_y < \infty$ whenever $|x[n]| < M_x < \infty$ is

Condition for Stability of a Causal LTI System

$$\sum_{m=0}^{\infty} |h[m]| < \infty \tag{8.30}$$

When (8.30) is satisfied, we say that the impulse response is *absolutely summable*, and saying this is equivalent to saying that the system is stable.

In the case of FIR systems, the impulse response has finite duration, and (8.30) is satisfied so long as $|h[n]| < \infty$. Therefore, all practical FIR systems are stable. However, in the case of IIR systems, the impulse response is an infinitely long sequence and not all IIR systems are stable.

[2] It can be shown that (8.30) is also a necessary condition; i.e., any system that violates (8.30) is an *unstable* system.

8-4.1 Example of an Unstable System

Consider the first-order difference equation

$$y[n] = y[n-1] + x[n] \tag{8.31}$$

This system is often called an *accumulator system* because it simply adds the current sample of the input, $x[n]$, to the total of previous samples, $y[n-1]$. The impulse response of this system can be shown by iteration to be

$$h[n] = u[n] \tag{8.32}$$

and it follows from (8.25) with $a = 1$, or by inspection from (8.31) that the system function is

$$H(z) = \sum_{n=0}^{\infty} z^{-n} = \frac{1}{1 - z^{-1}} \tag{8.33}$$

where the associated region of convergence for the infinite sum is $1 < |z|$.

Applying the condition for stability in (8.30) to $h[n] = u[n]$ shows that this is NOT a stable system; i.e., it must be true that some bounded input will produce an unbounded output. On the other hand, for this particular system, there are many bounded inputs that do produce a bounded output. An example is the unit impulse sequence, which produces as output, $h[n]$, which is clearly bounded by one. However, consider the input $x[n] = u[n]$, which is also an input bounded by one.

EXERCISE 8.9: Show by iteration of (8.31) that the output due to an input $x[n] = u[n]$ is

$$y[n] = (n+1)u[n] \tag{8.34}$$

The result in (8.34) is unbounded since the sequence grows linearly with n and we cannot find a value M_y such that $|y[n]| < M_y$ for all n. Even though $y[n] < \infty$ for finite n, the output will be very large for large n. Note that this is enough to prove that the system is *unstable*. On the other hand, observing that one or even many bounded inputs produce corresponding bounded outputs is NOT sufficient to prove that the system is stable. Fortunately, (8.30) is a necessary and sufficient condition for stability of an LTI system with impulse response $h[n]$, and we do not have to test the system for all inputs. We only need the impulse response.

EXERCISE 8.10: A slight change in the accumulator system will yield a stable system. Change (8.31) to

$$y[n] = ay[n-1] + x[n] \tag{8.35}$$

Show that if $|a| < 1$, the system is stable.

8-4.2 The Region of Convergence and Stability

Stability of an IIR system is clearly evident from the system function and its associated region of convergence. If the system is causal, the system function would have the form

$$H(z) = \sum_{n=0}^{\infty} h[n] z^{-n} \tag{8.36}$$

As we have observed, such infinite sums are generally not finite for all values of z. Instead we have found that there would be a set of values of z, called the region of convergence, where the sum converges to finite complex numbers, and, in most cases of practical interest, the value of the infinite sum is identical to the value of a "closed form" system function having the form of a ratio of polynomials in z^{-1}. To determine the values of z such that $H(z)$ is finite, consider the relation

$$|H(z)| = \left|\sum_{n=0}^{\infty} h[n]z^{-n}\right| < \infty \tag{8.37}$$

which gives a condition on the variable z such that the sum converges. Using the fact that the magnitude of a sum of terms is less than or equal to the sum of the magnitudes of those terms, we have

$$|H(z)| \leq \sum_{n=0}^{\infty} |h[n]||z|^{-n} < \infty \tag{8.38}$$

where we have also used the fact that the magnitude of a product of two terms is equal to the product of the magnitudes of those terms. Thus, the region of convergence is defined as those values of $|z|$ such that (8.38) is satisfied. For causal systems, the individual terms in the sum are $|h[n]||z|^{-n}$ for $0 \leq n < \infty$, and for the sum to converge, it is sufficient for the sequence of magnitudes of individual terms to converge exponentially to zero as n approaches infinity. For example, when $h[n] = a^n u[n]$ as in (8.25), we can see that this requires that $|a^n||z^{-n}| \to 0$ as $n \to \infty$, and this will be true if $|a||z^{-1}| < 1$ or equivalently $|a| < |z|$. In general the region of convergence for the system function of a causal IIR system will be of the form $R < |z|$, where R will be the magnitude of the pole of the system function that is farthest from the origin of the z-plane.

To conclude this discussion, consider the two conditions stated in (8.30) and (8.38). Equation (8.30) is the condition for stability of the LTI system with impulse response $h[n]$ and (8.38) defines the region of convergence for the corresponding system function of that system. Observe that if the values $|z| = 1$ (the unit circle) are in the region of convergence of $H(z)$, then it follows that (8.30) is also satisfied. Therefore, we can make the following important statement:

If the region of convergence of the system function of an LTI system includes the unit circle, then the system is stable, and, if the system is stable, the region of convergence of the system function will include the unit circle of the z-plane.

8-5 Poles and Zeros

An interesting fact about the z-transform system function is that the numerator and denominator polynomials have zeros. These zero locations in the complex z-plane are very important for characterizing the system. Although we like to write the system function in terms of z^{-1} to facilitate the correspondence with the difference equation, it is probably better for finding roots to rewrite the polynomials as functions of z rather than z^{-1}. If we multiply the numerator and denominator by z, we obtain

$$H(z) = \frac{b_0 + b_1 z^{-1}}{1 - a_1 z^{-1}} = \frac{b_0 z + b_1}{z - a_1} \tag{8.39}$$

In this form, it is easy to find the roots of the numerator and denominator polynomials.

The numerator has one root at

$$b_0 z + b_1 = 0 \quad \Longrightarrow \quad \text{Root at } z = -\frac{b_1}{b_0}$$

and the denominator has its root at

$$z - a_1 = 0 \quad \Longrightarrow \quad \text{Root at } z = a_1$$

If we consider $H(z)$ as a function of z over the entire complex z-plane, the root of the numerator is a ***zero*** of the function $H(z)$, i.e.,

$$H(z)\big|_{z=-(b_1/b_0)} = 0$$

Recall that the root of the denominator is a location in the z-plane where the function $H(z)$ *blows up*

$$H(z)\big|_{z=a_1} \to \infty$$

so this location $(z = a_1)$ is called a ***pole*** of the system function $H(z)$.

EXERCISE 8.11: Find the poles and zeros of the following z-transform system function

$$H(z) = \frac{3 + 4z^{-1}}{1 + 0.5z^{-1}}$$

EXERCISE 8.12: For the system function $H(z)$ of the following IIR system:

$$y[n] = 0.5y[n-1] - x[n] + 3x[n-1]$$

determine the locations of the pole and zeros.

Since MATLAB is used extensively in the labs and demonstrations that accompany this text, it is worthwhile to point out how MATLAB handles polynomials and their roots. Not surprisingly, the MATLAB format for polynomials is entirely compatible with its representation of the coefficients in the functions `filter( )` and `freqz( )`. For example the system function in (8.39), would be represented in MATLAB by the vectors `b=[b0,b1]` and `a=[1,-a1]` where the coefficients are thought of in terms of powers of z^{-1} moving left-to-right in the vector as in the lefthand form in (8.39). This is the format assumed by the MATLAB functions `filter(b,a,xx)` for implementing the system and `freqz(b,a,omega)` for determining its frequency response. When using the MATLAB function `roots( )`, we can use exactly the same vectors `b` and `a` because the `root( )` function assumes that the first coefficient in a vector of polynomial coefficients is the coefficient of the *highest* power of z as in the righthand form in (8.39); i.e., the MATLAB statements `z=roots(b)` and `p=roots(a)` will compute the zeros and poles for a system function represented as a ratio of polynomials in the variable z.

Several MATLAB-based resources are available to illustrate and demonstrate the pole-zero plot and its relation to the time- and frequency-response of an LTI system. These are indicated by the following links.

DEMO: *PeZ GUI*

LAB: *PeZ - The z, n, and $\hat{\omega}$ Domains*

DEMO: *PeZ Tutorial*

8-5.1 Poles or Zeros at the Origin or Infinity

When the numerator and denominator polynomials have a different number of coefficients, we can have either zeros or poles at $z = 0$. We saw this in Chapter 7, where FIR systems, whose system functions have only a numerator polynomial, had a number of poles at $z = 0$ equal to the number of zeros of the polynomial. If we count all the poles and zeros at $z = \infty$, as well as $z = 0$, then we can assert that ***the number of poles equals the number of zeros***. Consider the following examples.

Example 8-6: Find Poles and Zeros

The system function of the IIR system

$$y[n] = 0.5y[n-1] + 2x[n]$$

is found by inspection to be

$$H(z) = \frac{2}{1 - 0.5z^{-1}}$$

When we express $H(z)$ in positive powers of z

$$H(z) = \frac{2z}{z - 0.5}$$

we see that there is one pole at $z = 0.5$ and a zero at $z = 0$. ■

Example 8-7: Zero at $z = \infty$

The system fuction of the IIR system

$$y[n] = 0.5y[n-1] + 3x[n-1]$$

is found by inspection to be

$$H(z) = \frac{3z^{-1}}{1 - 0.5z^{-1}} = \frac{3}{z - 0.5}$$

The system has one pole at $z = 0.5$, and if we take the limit of $H(z)$ as $z \to \infty$, we get $H(z) \to 0$. Thus, it also has one zero at $z = \infty$. ■

Both of the cases in Examples 8-6 and 8-7 have exactly one pole and one zero, if we count zeros at either $z = 0$ or $z = \infty$.

EXERCISE 8.13: Determine the system function $H(z)$ of the following IIR system:

$$y[n] = 0.5y[n-1] + 3x[n-2]$$

Show that $H(z)$ has a pole at $z = 0$, as well as $z = 0.5$. In addition, show that $H(z)$ has two zeros at $z = \infty$ by taking the limit as $z \to \infty$.

8-5.2 Pole Locations and Stability

The pole location of a first-order filter determines the shape of the impulse response. In Section 8-3.3 we showed that a system having system function

$$\begin{aligned} H(z) &= b_0\left(\frac{1}{1-a_1z^{-1}}\right) + b_1z^{-1}\left(\frac{1}{1-a_1z^{-1}}\right) \\ &= \frac{b_0+b_1z^{-1}}{1-a_1z^{-1}} = \frac{b_0(z+b_1/b_0)}{(z-a_1)} \end{aligned}$$

has an impulse response

$$\begin{aligned} h[n] &= b_0(a_1)^n u[n] + b_1(a_1)^{n-1}u[n-1] \\ &= \begin{cases} 0 & \text{for } n<0 \\ b_0 & \text{for } n=0 \\ (b_0+b_1a_1^{-1})a_1^n & \text{for } n\geq 1 \end{cases} \end{aligned}$$

That is, an IIR system with a single pole at $z=a_1$ has an impulse response that is proportional to a_1^n for $n\geq 1$. We see that if $|a_1|<1$, the impulse response will die out as $n\to\infty$. On the other hand, if $|a_1|\geq 1$, the impulse response will not die out; in fact if $|a_1|>1$, it will grow without bound, and (8.30) tells us that the system is unstable. Since the pole of the system function is at $z=a_1$, we see that the location of the pole can tell us whether the impulse response will decay or grow. Clearly, it is desirable for the impulse response to die out, because an exponentially growing impulse response would produce unbounded outputs even if the input samples have finite size. As we found in Section 8-4, systems that produce bounded outputs when the input is bounded are called ***stable systems***. If $|a_1|<1$, the pole of the system function is ***inside*** the unit circle of the z-plane. It turns out that, for the IIR systems we have been discussing, the following is true in general:

A causal LTI IIR system with initial rest conditions is stable if all of the poles of its system function lie strictly inside the unit circle of the z-plane.

Thus, stability of a system can be seen at a glance from a z-plane plot of the poles and zeros of the system function.

Example 8-8: Stability from Pole Location

The system whose system function is

$$H(z) = \frac{1-2z^{-1}}{1-0.8z^{-1}} = \frac{z-2}{z-0.8}$$

has a zero at $z=2$ and a pole at $z=0.8$. Therefore, the system is stable. Note that the location of the zero, which is outside the unit circle, has nothing to do with stability of the system. Recall that the zeros correspond to an FIR system that is in cascade with an IIR system defined by the poles. Since FIR systems are always stable, it is not surprising that stability is determined solely by the poles of the system function. ■

EXERCISE 8.14: An LTI IIR system has system function

$$H(z) = \frac{2 + 2z^{-1}}{1 - 1.25z^{-1}}$$

Plot the pole and zero in the z-plane, and state whether or not the system is stable.

8-6 Frequency Response of an IIR Filter

In Chapter 6, we introduced the concept of frequency response $H(e^{j\hat{\omega}})$ as the complex function that determines the amplitude and phase change experienced by a complex exponential input to an LTI system; i.e., if $x[n] = e^{j\hat{\omega}n}$, then

$$y[n] = H(e^{j\hat{\omega}})e^{j\hat{\omega}n} \tag{8.40}$$

In Section 7-6 on p. 224 we showed that the frequency response of an FIR system is related to the system function by

$$H(e^{j\hat{\omega}}) = H(z)\big|_{z=e^{j\hat{\omega}}} \tag{8.41}$$

That is, the frequency response is identical to the system function evaluated on the unit circle of the z-plane $(z = ej\hat{\omega})$. This relation between the system function and the frequency response also holds for IIR systems. However, we must add the provision that the system must be stable in order for the frequency response to exist and to be given by (8.41). This condition of stability is a general condition, but all FIR systems are stable, so up to now we have not needed to be concerned with stability.

As was discussed in Section 8-4, in the IIR case, stability is intimately connected to the region of convergence of the system function. That discussion showed that when we perform the substitution in (8.41) asserting that the result is the frequency response of the system, we are assuming implicitly that the region of convergence of $H(z)$ includes the unit circle, and that in turn implies that the system is stable.

Recall that the system function for the general first-order IIR system has the form

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

where the region of convergence of the system function is $|a_1 z^{-1}| < 1$ or $|a_1| < |z|$. If we wish to evaluate $H(z)$ for $z = e^{j\hat{\omega}}$, then the values of z on the unit circle should be in the region of convergence; i.e., we require $|z| = 1$ to be in the region of convergence of the z-transform. This means that $|a_1| < 1$, which was shown in Section 8-5.2 to be the condition for stability of the first-order system. In Section 8-9 we will give another interpretation of why stability is required for the frequency response to exist. Assuming stability in the first-order case, we get the following formula for the frequency response:

$$H(e^{j\hat{\omega}}) = H(z)\big|_{z=e^{j\hat{\omega}}} = \frac{b_0 + b_1 e^{-j\hat{\omega}}}{1 - a_1 e^{-j\hat{\omega}}} \tag{8.42}$$

Since the rational function in (8.42) is a function of $e^{-j\hat{\omega}}$, which is periodic in $\hat{\omega}$ with period 2π, then $H(e^{j\hat{\omega}})$ is a periodic function with a period equal to 2π. This must always be the case for the frequency response of a discrete-time system.

Remember that the frequency response $H(e^{j\hat{\omega}})$ is a complex-valued function of frequency $\hat{\omega}$. Therefore, we can reduce (8.42) to two separate real formulas for the magnitude and the phase as functions of frequency.

For the magnitude of the frequency response, it is expedient to compute the magnitude squared first, and then take a square root if necessary.

The magnitude squared can be formed by multiplying the complex $H(e^{j\hat{\omega}})$ in (8.42) by its conjugate (denoted by H^*). For our first-order example,

$$\begin{aligned}
|H(e^{j\hat{\omega}})|^2 &= H(e^{j\hat{\omega}})H^*(e^{j\hat{\omega}}) \\
&= \frac{b_0 + b_1 e^{-j\hat{\omega}}}{1 - a_1 e^{-j\hat{\omega}}} \cdot \frac{b_0^* + b_1^* e^{+j\hat{\omega}}}{1 - a_1^* e^{+j\hat{\omega}}} \\
&= \frac{|b_0|^2 + |b_1|^2 + b_0 b_1^* e^{+j\hat{\omega}} + b_0^* b_1 e^{-j\hat{\omega}}}{1 + |a_1|^2 - a_1^* e^{+j\hat{\omega}} - a_1 e^{-j\hat{\omega}}} \\
&= \frac{|b_0|^2 + |b_1|^2 + 2\Re e\{b_0^* b_1 e^{-j\hat{\omega}}\}}{1 + |a_1|^2 - 2\Re e\{a_1 e^{-j\hat{\omega}}\}}
\end{aligned}$$

This derivation does not assume that the filter coefficients are real. If the coefficients were real, we would get the further simplification

$$|H(e^{j\hat{\omega}})|^2 = \frac{|b_0|^2 + |b_1|^2 + 2b_0 b_1 \cos(\hat{\omega})}{1 + |a_1|^2 - 2a_1 \cos(\hat{\omega})}$$

This formula is not particularly informative, because it is difficult to use it to visualize the shape of $|H(e^{j\hat{\omega}})|$. However, it could be used to write a program for evaluating and plotting the frequency response. The phase response is even messier. Arctangents would be used to extract the angles of the numerator and denominator, and then the two phases would be subtracted. When the filter coefficients are real, the phase is

$$\phi(\hat{\omega}) = \tan^{-1}\left(\frac{-b_1 \sin\hat{\omega}}{b_0 + b_1 \cos\hat{\omega}}\right) - \tan^{-1}\left(\frac{a_1 \sin\hat{\omega}}{1 - a_1 \cos\hat{\omega}}\right)$$

Again, the formula is so complicated that we cannot gain insight from it directly. In a later section, we will use the poles and zeros of the system function together with the relationship (8.41) to construct an approximate plot of the frequency response without recourse to formulas.

8-6.1 Frequency Response using MATLAB

Frequency responses can be computed and plotted easily by many signal processing software packages. In MATLAB, for example, the function `freqz` is provided for just that purpose.[3] The frequency response is evaluated over an equally spaced grid in the $\hat{\omega}$ domain, and then its magnitude and phase can be plotted. In MATLAB, the functions `abs` and `angle` will extract the magnitude and the angle of each element in a complex vector.

Example 8-9: Plot $H(e^{j\hat{\omega}})$ via MATLAB

Consider the example

$$y[n] = 0.8y[n-1] + 2x[n] + 2x[n-1]$$

In order to define the filter coefficients in MATLAB, we put all the terms with $y[n]$ on one side of the equation, and the terms with $x[n]$ on the other.

$$y[n] - 0.8y[n-1] = 2x[n] + 2x[n-1]$$

[3]The function `freqz` is part of MATLAB's Signal Processing Toolbox. In case a substitute is needed, there is a similar function called `freekz` that is part of the SP-First Toolbox on the CD-ROM.

Then we read off the filter coefficients and define the vectors `aa` and `bb` as

```
aa = [ 1, −0.8 ]     bb = [ 2, 2 ]
```

Thus, the vectors `aa` and `bb` are in the same form as for the `filter` function. The following call to `freqz` will generate a 401-point vector `HH` containing the values of the frequency response at the vector of frequencies specified by the third argument, `2*pi*[-1:.005:1]`.

```
HH = freqz( bb, aa, 2*pi*[-1:.005:1] );
```

Plots of the resulting magnitude and phase are shown in Fig. 8-9. The frequency interval $-2\pi \leq \hat{\omega} \leq +2\pi$ is shown so that the 2π-periodicity of $H(e^{j\hat{\omega}})$ will be evident.[4]

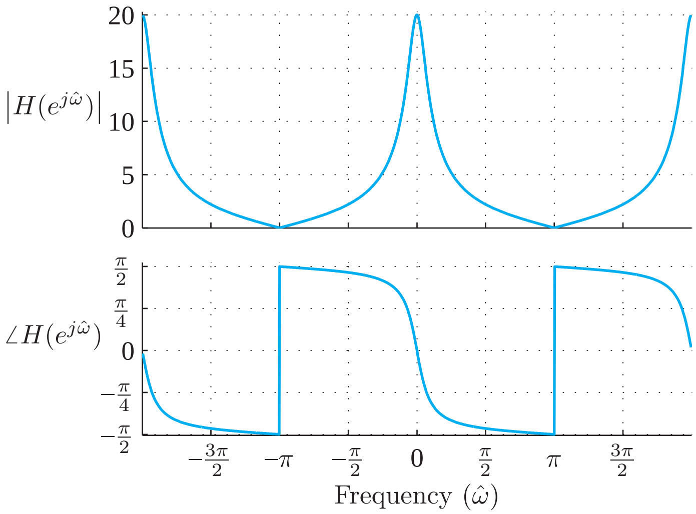

Figure 8-9: Frequency response (magnitude and phase) for a first-order feedback filter. The pole is at $z = 0.8$, and the numerator has a zero at $z = -1$.

In this example, we can look for a connection between the poles and zeros and the shape of the frequency response. For this system, we have the system function

$$H(z) = \frac{2 + 2z^{-1}}{1 - 0.8z^{-1}}$$

which, as shown in Fig. 8-10, has a pole at $z = 0.8$ and a zero at $z = -1$.

The point $z = -1$ is the same as $\hat{\omega} = \pi$ because $z = -1 = e^{j\pi} = e^{j\hat{\omega}}|_{\hat{\omega}=\pi}$. Thus, $H(e^{j\hat{\omega}})$ has the value zero at $\hat{\omega} = \pi$, since $H(z)$ is zero at $z = -1$. In a similar manner, the pole at $z = 0.8$ has an effect on the frequency response near $\hat{\omega} = 0$. Since $H(z)$ blows up at $z = 0.8$, the nearby points on the unit circle must have large values. The closest point on the unit circle is at $z = e^{j0} = 1$. In this case, we can evaluate the frequency response directly from the formula to get

$$\begin{aligned} H(e^{j\hat{\omega}})\Big|_{\hat{\omega}=0} &= H(z)\Big|_{z=1} = \frac{2 + 2z^{-1}}{1 - 0.8z^{-1}}\Bigg|_{z=1} \\ &= \frac{2+2}{1-0.8} = \frac{4}{0.2} = 20 \end{aligned}$$

[4]The labels on the graph were created using special plotting functions.

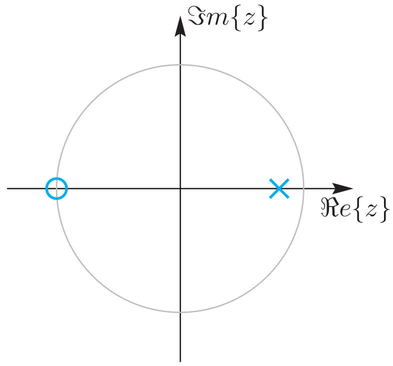

Figure 8-10: Pole-zero plot for a first-order IIR filter. The pole is at $z = 0.8$, and the numerator has a zero at $z = -1$.

8-6.2 Three-Dimensional Plot of a System Function

The relationship between $H(e^{j\hat{\omega}})$ and the pole-zero locations of $H(z)$ can be illustrated by making a three-dimensional plot of $H(z)$ and then cutting out the frequency response. The frequency response $H(e^{j\hat{\omega}})$ is obtained by selecting the values of $H(z)$ along the unit circle (i.e., as $\hat{\omega}$ goes from $-\pi$ to $+\pi$, the equation $z = e^{j\hat{\omega}}$ defines the unit circle).

In this section, we use the system function

$$H(z) = \frac{2 - 2z^{-1}}{1 - 0.8z^{-1}}$$

from Example 8-9 to illustrate the relationship between the system function and the frequency response. Figure 8-11 is a plot of the magnitude of $H(z)$ over the region $[-1.4, 1.4] \times [-1.4, 1.4]$ of the z-plane. Note that since z is complex, we have a two dimensional function, which is a surface over the z-plane. In the magnitude plot of Fig. 8-11, we observe that the pole (at $z = 0.8$) creates a large peak that makes all nearby values very large. At the precise location of the pole, $H(z) \to \infty$, but the grid in Fig. 8-11 does not contain the point $(z = 0.8)$, so the plot stays within a finite scale. Also note how the surface dips sharply in the vicinity of $z = -1$, which is where the zero of $H(z)$ is located.

The frequency response $H(e^{j\hat{\omega}})$ is obtained by selecting the values of the z-transform along the unit circle in Fig. 8-11. Plots of $|H(e^{j\hat{\omega}})|$ and $\angle H(e^{j\hat{\omega}})$ versus $\hat{\omega}$ were given in Fig. 8-9 of Example 8-9. The shape of the frequency response can be explained in terms of the pole and zero location by recognizing that in Fig. 8-11 the pole at $z = 0.8$ pushes $H(e^{j\hat{\omega}})$ up in the region near $\hat{\omega} = 0$, which is the same as $z = 1$. On the other hand, the zero at $z = -1$ forces $|H(e^{j\hat{\omega}})|$ to be zero at precisely $\hat{\omega} = \pi$ and small in the neighborhood of $z = -1$. The unit circle values follow the ups and downs of $H(z)$ as $\hat{\omega}$ goes from $-\pi$ to $+\pi$.

CDROM **DEMO:** *Z to Freq*

8-7 Three Domains

To illustrate the use of the analysis tools that we have developed, we consider the general second-order case.

CDROM **DEMO:** *Three Domains - IIR*

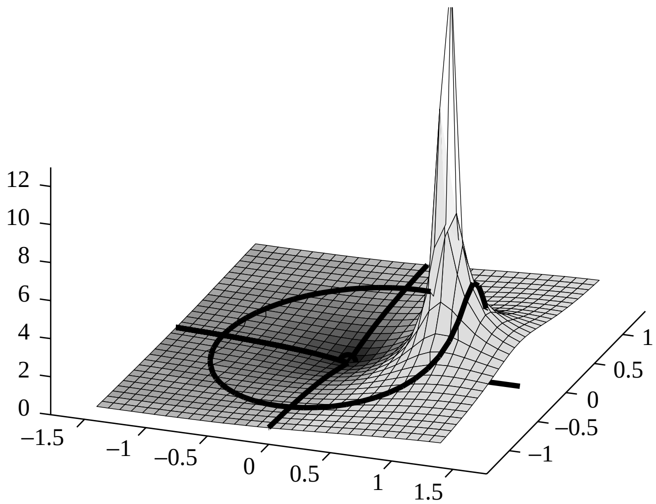

Figure 8-11: z-transform evaluated over a region of the z-plane including the unit circle. Values along the unit circle are shown as a dark line where the frequency response (magnitude) is evaluated. The view is from the fourth quadrant, so the point $z = 1$ is on the right. The first-order filter has a pole at $z = 0.8$ and a zero at $z = 0$. We need to redo the figure here.

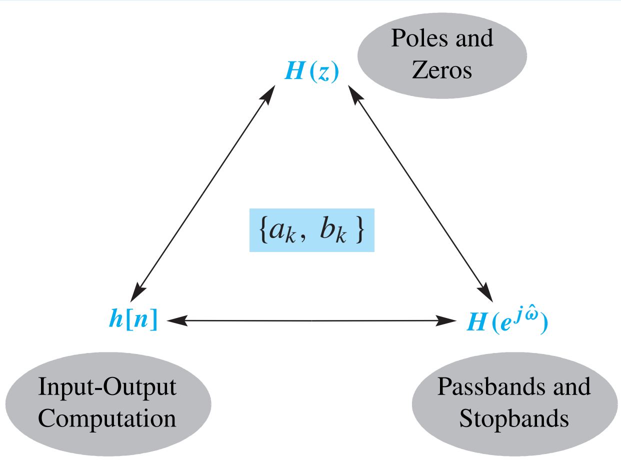

Figure 8-12: Relationship among the n-, z-, and $\hat{\omega}$-domains. The filter coefficients $\{a_k,\ b_k\}$ play a central role.

The three domains: n, z and $\hat{\omega}$ are depicted in Fig. 8-12. The defining equation for the IIR digital filter is the feedback difference equation, which, for the second-order case, is

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

This equation provides the algorithm for computing the output signal from the input signal by iteration using the filter coefficients $\{a_1,\ a_2,\ b_0,\ b_1,\ b_2\}$. It also defines the impulse response $h[n]$.

Following the procedures illustrated for the first-order case, we can also define the z-transform system function directly from the filter coefficients as

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-2}}$$

and we can also obtain the frequency response

$$H(e^{j\hat{\omega}}) = \frac{b_0 + b_1 e^{-j\hat{\omega}} + b_2 e^{-j2\hat{\omega}}}{1 - a_1 e^{-j\hat{\omega}} - a_2 e^{-j2\hat{\omega}}}$$

Since the system function is a ratio of polynomials, the poles and zeros of $H(z)$ make up a small set of parameters that completely define the filter.

Finally, the shapes of the passbands and stopbands of the frequency response are highly dependent on the pole and zero locations with respect to the unit circle, and the character of the impulse response can be related to the poles. To make this last point for the general case, we need to develop one more tool—a technique for getting $h[n]$ directly from $H(z)$. This process, which applies to any z-transform and its corresponding sequence, is called the ***inverse z-transform***. The inverse z-transform is developed in the next section.

8-8 The Inverse z-Transform and Some Applications

We have seen how the three domains are connected for the first-order IIR system. Many of the concepts that we have introduced for the first-order system can be extended to higher-order systems in a straightforward manner. However, finding the impulse response from the system function is not an obvious extension of what we have done for the first-order case. We need to develop a process for inverting the z-transform that can be applied to systems with more than one pole. This process is called the ***inverse z-transform***. In this section we will show how to find the inverse for a general rational z-transform. We will illustrate the process with some examples. The techniques that we develop will then be available for determining the impulse responses of second-order and higher-order systems.

8-8.1 Revisiting the Step Response of a First-Order System

In Section 8-2.4 we computed the step response of a first-order system by both iteration and convolution. Now we will show how the z-transform can be used for the same purpose. Consider a system whose system function is

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

Recall that the z-transform of the output of this system is $Y(z) = H(z)X(z)$, so one approach to finding the output for a given input $x[n]$ is as follows:

1. Determine the z-transform $X(z)$ of the input signal $x[n]$.
2. Multiply $X(z)$ by $H(z)$ to get $Y(z)$.
3. Determine the inverse z-transform of $Y(z)$ to get the output $y[n]$.

Clearly, this procedure will work and will avoid both iteration and convolution if we can determine $X(z)$ and if we can perform the necessary inverse transformation. Our focus in this section will be on deriving a general procedure for step (3).

In the case of the step response, we see that the input $x[n] = u[n]$ is a special case of the more general sequence $a^n u[n]$; i.e., $a = 1$. Therefore, from (8.25) it follows that the z-transform of $x[n] = u[n]$ is

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

so $Y(z)$ is

$$\begin{aligned} Y(z) = H(z)X(z) &= \frac{b_0 + b_1 z^{-1}}{(1 - a_1 z^{-1})(1 - z^{-1})} \\ &= \frac{b_0 + b_1 z^{-1}}{1 - (1 + a_1)z^{-1} + a_1 z^{-2}} \end{aligned} \tag{8.43}$$

Now we need to go back to the n-domain by inverse transformation. A standard approach is to use a table of z-transform pairs and simply look up the answer in the table. Our previous discussions in Chapter 7 and earlier in this chapter have developed the basis for a simple version of such a table. A summary of the z-transform knowledge that we have developed so far is given in Table 8-1. Although more extensive tables can

Table 8-1: Summary of important z-transform properties and pairs.

Short Table Of z-Transforms			
	$x[n]$	$\overset{z}{\longleftrightarrow}$	$X(z)$
1.	$ax_1[n] + bx_2[n]$	$\overset{z}{\longleftrightarrow}$	$aX_1(z) + bX_2(z)$
2.	$x[n - n_0]$	$\overset{z}{\longleftrightarrow}$	$z^{-n_0}X(z)$
3.	$y[n] = x[n] * h[n]$	$\overset{z}{\longleftrightarrow}$	$Y(z) = H(z)X(z)$
4.	$\delta[n]$	$\overset{z}{\longleftrightarrow}$	1
5.	$\delta[n - n_0]$	$\overset{z}{\longleftrightarrow}$	z^{-n_0}
6.	$a^n u[n]$	$\overset{z}{\longleftrightarrow}$	$\dfrac{1}{1 - az^{-1}}$

be constructed, the results that we have assembled in Table 8-1 are more than adequate for our purposes in this text.

Now let us return to the problem of finding $y[n]$ given $Y(z)$ in (8.43). The technique that we will use is based on the partial fraction expansion[5] of $Y(z)$. This technique is based on the observation that a rational function $Y(z)$ can be expressed as a sum of simpler rational functions; i.e.,

$$\begin{aligned} Y(z) &= H(z)X(z) \\ &= \frac{b_0 + b_1 z^{-1}}{(1 - a_1 z^{-1})(1 - z^{-1})} \\ &= \frac{A}{1 - a_1 z^{-1}} + \frac{B}{1 - z^{-1}} \end{aligned} \tag{8.44}$$

[5]The ***partial fraction expansion*** is an algebraic decomposition usually presented in calculus for evaluating certain types of integrals.

If the expression on the right is pulled together over a common denominator, it should be possible to find A and B so that the numerator of the resulting rational function will be equal to $b_0 + b_1 z^{-1}$. Equating the two numerators would give two equations in the two unknowns A and B. However, there is a much quicker way. A systematic procedure for finding the desired A and B is based on the observation that, for this example,

$$Y(z)(1 - a_1 z^{-1}) = \frac{b_0 + b_1 z^{-1}}{(1 - z^{-1})} = A + \frac{B(1 - a_1 z^{-1})}{1 - z^{-1}}$$

Then we can evaluate at $z = a_1$ to isolate A; i.e.,

$$\begin{aligned} Y(z)(1 - a_1 z^{-1})\Big|_{z=a_1} &= \frac{b_0 + b_1 z^{-1}}{(1 - z^{-1})}\Big|_{z=a_1} \\ &= A + \frac{B(1 - a_1 z^{-1})}{1 - z^{-1}}\Big|_{z=a_1} = A \end{aligned}$$

With this result, it follows that

$$A = Y(z)(1 - a_1 z^{-1})\Big|_{z=a_1} = \frac{b_0 + b_1 a_1^{-1}}{1 - a_1^{-1}}$$

Similarly, we can find B by

$$B = Y(z)(1 - z^{-1})\Big|_{z=1} = \frac{b_0 + b_1}{1 - a_1}$$

Now using the superposition property of the z-transform (entry 1 in Table 8-1), and the exponential z-transform pair (entry 6 in Table 8-1), we can write down the desired answer as

$$y[n] = \left(\frac{b_0 + b_1 a_1^{-1}}{1 - a_1^{-1}}\right) a_1^n u[n] + \left(\frac{b_0 + b_1}{1 - a_1}\right) u[n]$$

which, after some manipulation becomes

$$y[n] = \left(\frac{(b_0 + b_1) - (b_0 a_1 + b_1) a_1^n}{1 - a_1}\right) u[n] \tag{8.45}$$

If we substitute the value $b_1 = 0$ into (8.45), we get

$$y[n] = b_0 \left(\frac{1 - a_1^{n+1}}{1 - a_1}\right) u[n]$$

which is the same result obtained in Section 8-2.4 both by iteration of the difference equation (8.16) and by convolution (8.18).

With this example, we have established the framework for using the basic properties of z-transforms together with a few basic z-transform pairs to perform inverse z-transformation for any rational z-transform. We summarize this procedure in the following subsection.

8-8.2 A General Procedure for Inverse z-Transformation

Let $X(z)$ be any rational z-transform of degree N in the denominator and M in the numerator. Assuming that $M < N$, we can find the sequence $x[n]$ that corresponds to $X(z)$ by the following procedure:

Procedure for
Inverse z-Transformation ($M < N$)

1. Factor the denominator polynomial of $X(z)$ and express the pole factors in the form $(1 - p_k z^{-1})$ for $k = 1, 2, \ldots, N$.

2. Make a partial fraction expansion of $X(z)$ into a sum of terms of the form

$$X(z) = \sum_{k=1}^{N} \frac{A_k}{1 - p_k z^{-1}}$$

$$\text{where} \quad A_k = X(z)(1 - p_k z^{-1})\Big|_{z=p_k}$$

3. Write down the answer as

$$x[n] = \sum_{k=1}^{N} A_k (p_k)^n u[n]$$

This procedure will always work if the poles, p_k, are distinct. Repeated poles complicate the process, but can be handled systematically. We will restrict our attention to the case of non-repeated poles. Furthermore, this procedure can be applied to the inversion of any rational z-transform, including system functions whose inverse z-transform would be $h[n]$. We will illustrate the use of this procedure with two examples.

Example 8-10: Inverse z-Transform

Let a z-transform $X(z)$ be

$$\begin{aligned} X(z) &= \frac{1 - 2.1z^{-1}}{1 - 0.3z^{-1} - 0.4z^{-2}} \\ &= \frac{1 - 2.1z^{-1}}{(1 + 0.5z^{-1})(1 - 0.8z^{-1})} \end{aligned}$$

We wish to write $X(z)$ in the form

$$X(z) = \frac{A}{1 + 0.5z^{-1}} + \frac{B}{1 - 0.8z^{-1}}$$

Continuing the procedure for partial fraction expansion, we obtain

$$\begin{aligned} A &= X(z)(1+0.5z^{-1})\Big|_{z=-0.5} \\ &= \frac{1-2.1z^{-1}}{1-0.8z^{-1}}\Bigg|_{z=-0.5} = \frac{1+4.2}{1+1.6} = 2 \\ \text{and} \quad B &= X(z)(1-0.8z^{-1})\Big|_{z=0.8} \\ &= \frac{1-2.1z^{-1}}{1+0.5z^{-1}}\Bigg|_{z=0.8} = \frac{1-2.1/0.8}{1+0.5/0.8} = -1 \end{aligned}$$

Therefore,

$$X(z) = \frac{2}{1+0.5z^{-1}} - \frac{1}{1-0.8z^{-1}} \tag{8.46}$$

and

$$x[n] = 2(-0.5)^n u[n] - (0.8)^n u[n]$$

Note that the poles at $z = p_1 = -0.5$ and $z = p_2 = 0.8$ give rise to terms in $x[n]$ of the form p_k^n. ■

In Example 8-10, the degree of the numerator is $M = 1$ and the degree of the denominator is $N = 2$. This is important because the partial fraction expansion works only for rational functions such that $M < N$. The next example shows why this is so, and illustrates a method of dealing with this complication.

Example 8-11: Long Divison

Let $Y(z)$ be

$$\begin{aligned} Y(z) &= \frac{2-2.4z^{-1}-0.4z^{-2}}{1-0.3z^{-1}-0.4z^{-2}} \\ &= \frac{2-2.4z^{-1}-0.4z^{-2}}{(1+0.5z^{-1})(1-0.8z^{-1})} \end{aligned}$$

Now we must add a constant term to the partial fraction expansion, otherwise, we cannot generate the term $-0.4z^{-2}$ in the numerator when we combine the partial fractions over a common denominator. That is, we must assume the following form for $Y(z)$:

$$Y(z) = \frac{A}{1+0.5z^{-1}} + \frac{B}{1-0.8z^{-1}} + C$$

How can we determine the constant C? One way is to perform long division of the denominator polynomial into the numerator polynomial until we get a remainder whose degree is lower than that of the denominator. In this case, the polynomial long division looks as follows:

$$\begin{array}{r l} & \qquad\qquad 1 \\ -0.4z^{-2} - 0.3z^{-1} + 1 & \overline{\big) -0.4z^{-2} - 2.4z^{-1} + 2} \\ & \underline{\;\; -0.4z^{-2} - 0.3z^{-1} + 1} \\ & \qquad\qquad - 2.1z^{-1} + 1 \end{array}$$

Thus, if we place the remainder $(1 - 2.1z^{-1})$ over the denominator (in factored form), we can write $Y(z)$ as a rational part (fraction) plus the constant 1; i.e.,

$$Y(z) = \frac{1 - 2.1z^{-1}}{(1 + 0.5z^{-1})(1 - 0.8z^{-1})} + 1$$

The next step would be to apply the partial fraction expansion technique to the rational part of $Y(z)$. Since the rational part turns out to be identical to $X(z)$ in (8.46) from Example 8-10, the results would be the same as in that example, so we can write $Y(z)$ as

$$Y(z) = \frac{2}{1 + 0.5z^{-1}} - \frac{1}{1 - 0.8z^{-1}} + 1 \tag{8.47}$$

Therefore, from Table 8-1,

$$y[n] = 2(-0.5)^n u[n] - (0.8)^n u[n] + \delta[n]$$

Notice again that the time-domain sequence has terms of the form p_k^n. The constant term in the system function generates an impulse, which is nonzero only at $n = 0$ (entry 4 in Table 8-1). ■

The numerical computations for carrying out all the steps of the partial fraction expansion of a rational z-transform can be performed conveniently by the MATLAB function `residuez( )`. The statement

```
[r,p,k]=residuez([2, -2.4, -0.4],[1, -0.3, -0.4])
```

produces the result

```
r =
   -1.0000
    2.0000
p =
    0.8000
   -0.5000
k =
     1
```

where the vector `r` contains the values of the numerator coefficients (called the *residues*) in (8.47), `p` contains the values of the corresponding poles, and `k` gives the coefficients of the quotient polynomial.

8-9 Steady-State Response and Stability

A stable system is one that does not "blow up." This intuitive statement can be formalized by saying that the output of a stable system can always be bounded ($|y[n]| < M_y$) whenever the input is bounded ($|x[n]| < M_x$).[6]

We can use the inverse z-transform method developed in Section 8-8 to demonstrate an important point about stability, the frequency response, and the sinusoidal steady-state response. To illustrate this point, consider the LTI system defined by

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

[6]This definition for stability is called bounded-input, bounded-output stability. The finite constants M_x and M_y can be different.

From our discussion so far, we can state without further analysis that the system function of this system is

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

and that the impulse response is

$$h[n] = b_0 a_1^n \, u[n]$$

We can state also that the frequency response is

$$H(e^{j\hat{\omega}}) = H(z)\big|_{z=e^{j\hat{\omega}}} = \frac{b_0}{1 - a_1 e^{-j\hat{\omega}}}$$

but this is true only if the system is stable ($|a_1| < 1$). The objective of this section is to refine the concept of stability and demonstrate its impact on the response to a sinusoid applied at $n = 0$.

Recall from Section 8-6 and equations (8.40) and (8.42) that the output of this system for a complex exponential input is

$$y[n] = H(e^{j\hat{\omega}_0})e^{j\hat{\omega}_0 n} = \left(\frac{b_0}{1 - a_1 e^{-j\hat{\omega}_0}}\right) e^{j\hat{\omega}_0 n}$$

for $-\infty < n < \infty$. What if the complex exponential input sequence is suddenly applied instead of existing for all n? The z-transform tools that we have developed make it easy to solve this problem. Indeed, the z-transform is ideally suited for situations where the sequences are either finite-length sequences or suddenly applied exponentials. For the suddenly applied complex exponential sequence with frequency $\hat{\omega}_0$

$$x[n] = e^{j\hat{\omega}_0 n} u[n]$$

the z-transform is found from entry 6 of Table 8-1 to be

$$X(z) = \frac{1}{1 - e^{j\hat{\omega}_0} z^{-1}}$$

and the z-transform of the output of the LTI system is

$$\begin{aligned} Y(z) = H(z)X(z) &= \left(\frac{b_0}{1 - a_1 z^{-1}}\right)\left(\frac{1}{1 - e^{j\hat{\omega}_0} z^{-1}}\right) \\ &= \frac{b_0}{(1 - a_1 z^{-1})(1 - e^{j\hat{\omega}_0} z^{-1})} \end{aligned}$$

Using the technique of partial fraction expansion, we can show that

$$Y(z) = \frac{\left(\dfrac{b_0 a_1}{a_1 - e^{j\hat{\omega}_0}}\right)}{1 - a_1 z^{-1}} + \frac{\left(\dfrac{b_0}{1 - a_1 e^{-j\hat{\omega}_0}}\right)}{1 - e^{j\hat{\omega}_0} z^{-1}}$$

Therefore, the output due to the suddenly applied complex exponential sequence is

$$\begin{aligned} y[n] = &\left(\frac{b_0 a_1}{a_1 - e^{j\hat{\omega}_0}}\right)(a_1)^n u[n] \\ &+ \left(\frac{b_0}{1 - a_1 e^{-j\hat{\omega}_0}}\right) e^{j\hat{\omega}_0 n} u[n] \end{aligned} \tag{8.48}$$

Equation (8.48) shows that the output consists of two terms. One term is proportional to an exponential sequence a_1^n that is solely determined by the pole at $z = a_1$. If $|a_1| < 1$, this term will die out with increasing n, in which case it would be called the ***transient component***. The second term is proportional to the input complex exponential signal, and the constant of proportionality term is $H(e^{j\hat{\omega}_0})$, the frequency response of the system evaluated at the frequency of the suddenly applied complex sinusoid. This complex exponential component is the ***sinusoidal steady-state component*** of the output.

Now we see that the location of the pole of $H(z)$ is crucial if we want the output to reach the sinusoidal steady state. Clearly, if $|a_1| < 1$, then the system is stable and the pole is inside the unit circle. For this condition, the exponential a_1^n dies out and we can state that the limiting value for large n

$$y[n] \to \left(\frac{b_0}{1 - a_1 e^{-j\hat{\omega}_0}}\right) e^{j\hat{\omega}_0 n} = H(e^{j\hat{\omega}_0})e^{j\hat{\omega}_0 n}$$

Otherwise, if $|a_1| \geq 1$, the term proportional to a_1^n will grow with increasing n and soon dominate the output. The following example gives a specific numerical illustration.

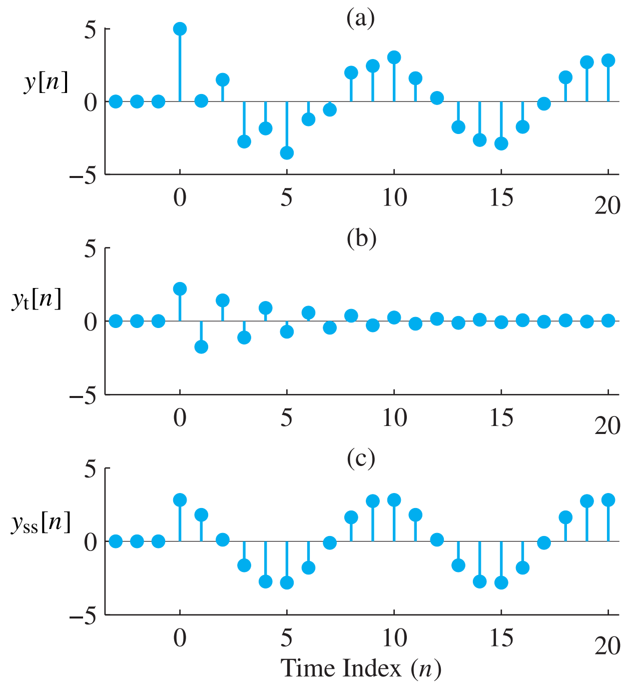

Figure 8-13: Illustration of transient and steady-state responses of an IIR system: (a) Total output $\Re e\{y[n]\}$, (b) decaying transient component $\Re e\{y_t[n]\}$, and (c) steady-state response $\Re e\{y_{ss}[n]\}$.

Example 8-12: Transient and Steady-State

If $b_0 = 5$, $a_1 = -0.8$, and $\hat{\omega}_0 = 2\pi/10$, the transient component is

$$\begin{aligned} y_t[n] &= \left(\frac{-4}{-0.8 - e^{j0.2\pi}}\right)(-0.8)^n u[n] \\ &= 2.3351e^{-j0.3502}(-0.8)^n u[n] \\ &= 2.1933(-0.8)^n u[n] - j0.8012(-0.8)^n u[n] \end{aligned}$$

Similarly, the steady-state component is

$$\begin{aligned} y_{ss}[n] &= \left(\frac{5}{1+0.8e^{-j0.2\pi}}\right) e^{j0.2\pi n} u[n] \\ &= 2.9188 e^{j0.2781} e^{j0.2\pi n} u[n] \\ &= 2.9188 \cos(0.2\pi n + 0.2781)\, u[n] \\ &\quad + j\, 2.9188 \sin(0.2\pi n + 0.2781)\, u[n] \end{aligned}$$

Figure 8-13 shows the real part of the total output (a), and also the transient component (b), and the steady-state component (c). The signals all start at $n = 0$ when the complex exponential is applied at the input. Note how the transient component oscillates, but dies away, which explains why the steady-state component eventually equals the total output. In Fig. 8-13, $y[n]$ in (a) and $y_{ss}[n]$ in (c) look identical for $n > 15$. ■

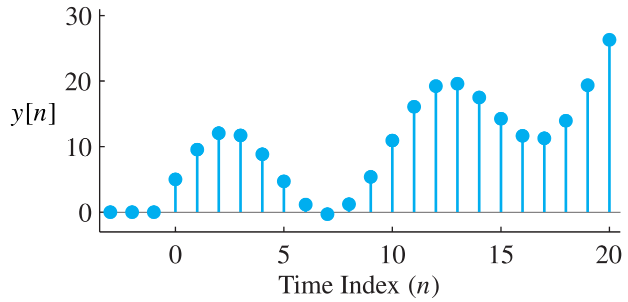

Figure 8-14: Illustration of an unstable IIR system. Pole is at $z = 1.1$.

On the other hand, if the pole were at $z = 1.1$, the system would be unstable and the output would "blow up" as shown in Fig. 8-14. In this case, the output contains a term $(1.1)^n$ that eventually dominates and grows without bound.

The result of Example 8-12 can be generalized by observing that the only difference between this example and a system with a higher-order system function is that the total output would include one exponential factor for each pole of the system function as well as the term $H(e^{j\hat{\omega}_0})e^{j\hat{\omega}_0 n}u[n]$. That is, it can be shown that for a suddenly applied exponential input sequence $x[n] = e^{j\hat{\omega}_0 n}u[n]$, the output of an N^{th}-order IIR $(N > M)$ system will always be of the form

$$y[n] = \sum_{k=1}^{N} A_k (p_k)^n u[n] + H(e^{j\hat{\omega}_0})e^{j\hat{\omega}_0 n}u[n]$$

where the p_ks are the poles of the system function. Therefore, the sinusoidal steady state will exist and dominate in the total response if the poles of the system function all lie strictly inside the unit circle. This makes the concept of frequency response useful in a practical setting where all signals must have a beginning point at some finite time.

8-10 Second-Order Filters

We now turn our attention to filters with two feedback coefficients, a_1 and a_2. The general difference equation (8.1) becomes the second-order difference equation

$$\begin{aligned} y[n] &= a_1 y[n-1] + a_2 y[n-2] \\ &\quad + b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] \end{aligned} \tag{8.49}$$

As before, we can characterize the second-order filter (8.49) in each of the three domains: time domain, frequency domain, and z-domain. We start with the z-transform domain because by now we have demonstrated that the poles and zeros of the system function give a great deal of insight into most aspects of both the time and frequency responses.

DEMO: *PeZ GUI*

8-10.1 z-Transform of Second-Order Filters

Using the approach followed in Section 8-3.1 for the first-order case, we can take the z-transform of the second-order difference equation (8.49) by replacing each delay with z^{-1} (second entry in Table 8-1), and also replacing the input and output signals with their z-transforms:

$$\begin{aligned} Y(z) &= a_1 z^{-1} Y(z) + a_2 z^{-2} Y(z) \\ &\quad + b_0 X(z) + b_1 z^{-1} X(z) + b_2 z^{-2} X(z) \end{aligned}$$

In the z-transform domain, the input-output relationship is $Y(z) = H(z)X(z)$, so we can solve for $H(z)$ by finding $Y(z)/X(z)$. For the second-order filter we get

$$\begin{aligned} Y(z) - a_1 z^{-1} Y(z) - a_2 z^{-2} Y(z) & \\ = b_0 X(z) + b_1 z^{-1} X(z) + b_2 z^{-2} X(z) & \\ (1 - a_1 z^{-1} - a_2 z^{-2}) Y(z) = (b_0 + b_1 z^{-1} + b_2 z^{-2}) X(z) & \end{aligned}$$

which can be solved for $H(z)$ as

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-2}} \tag{8.50}$$

Thus, the system function $H(z)$ for an IIR filter is a ratio of two second-degree polynomials, where the numerator polynomial depends on the feed-forward coefficients $\{b_k\}$ and the denominator depends on the feedback coefficients $\{a_\ell\}$. It should be possible to work problems such as Exercise 8.15 by simply reading the filter coefficients from the difference equation and then substituting them directly into the z-transform expression for $H(z)$.

EXERCISE 8.15: Find system function $H(z)$ of the following IIR filter:

$$\begin{aligned} y[n] &= 0.5 y[n-1] + 0.3 y[n-2] \\ &\quad - x[n] + 3x[n-1] - 2x[n-2] \end{aligned}$$

Conversely, given the system function $H(z)$, it is a simple matter to write down the difference equation.

EXERCISE 8.16: For the system function

$$H(z) = \frac{1 + 2z^{-1} + z^{-2}}{1 - 0.8z^{-1} + 0.64z^{-2}}$$

write down the difference equation that relates the input $x[n]$ to the output $y[n]$.

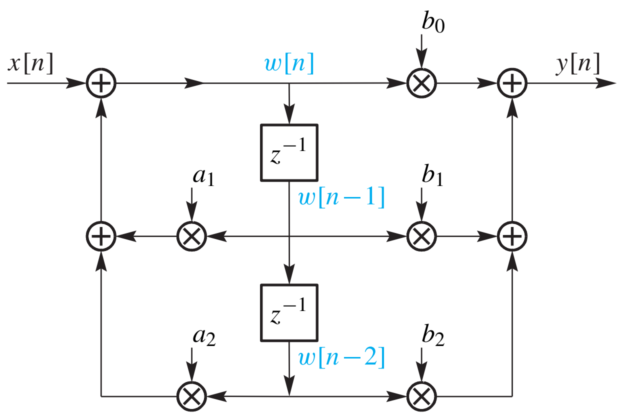

Figure 8-15: Direct Form II (DF-II), an alternative computational structure for the second-order recursive filter.

Example 8-13: Structure for $H(z)$

The connection between $H(z)$ and the difference equation can be generalized to higher-order filters. If we are given a fourth-order system

$$H(z) = \frac{1 - 3z^{-2}}{1 - 0.8z^{-1} + 0.6z^{-3} + 0.3z^{-4}}$$

the corresponding difference equation is

$$\begin{aligned} y[n] = 0.8y[n-1] &- 0.6y[n-3] \\ &- 0.3y[n-4] + x[n] - 3x[n-2] \end{aligned}$$

As before, note the sign change in the feedback coefficients, $\{a_k\}$. ■

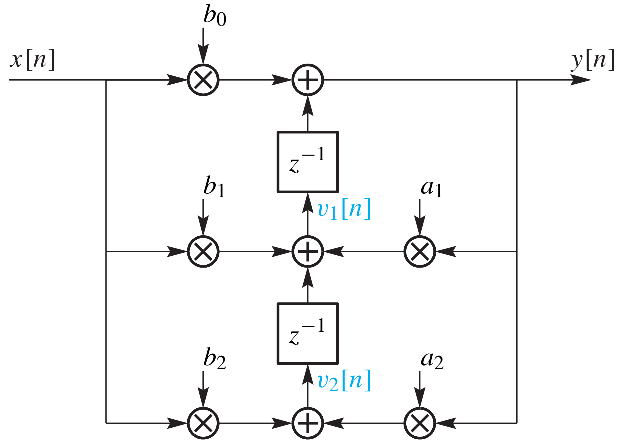

Figure 8-16: Transposed Direct Form II (TDF-II), another computational structure for the second-order recursive filter.

8-10.2 Structures for Second-Order IIR Systems

The difference equation (8.49) can be interpreted as an algorithm for computing the output sequence from the input. Other computational orderings are possible, and the z-transform has the power to derive alternative structures through polynomial manipulations. Two alternative computational orderings that will implement the system defined by $H(z)$ in (8.50) are given in Figs. 8-15 and 8-16.

In order to verify that the block diagram in Fig. 8-15 has the correct system function, we need to write the equations of the structure at the adders, and then eliminate the internal variable(s). For the case of the Direct Form II in Fig. 8-15, the equations at the output of the summing nodes are

$$\begin{aligned} y[n] &= b_0 w[n] + b_1 w[n-1] + b_2 w[n-2] \\ w[n] &= x[n] + a_1 w[n-1] + a_2 w[n-2] \end{aligned} \tag{8.51}$$

It is impossible to eliminate $w[n]$ in these two equations, unless we work in the z-transform domain. The corresponding z-transform equations are

$$\begin{aligned} Y(z) &= b_0 W(z) + b_1 z^{-1} W(z) + b_2 z^{-2} W(z) \\ W(z) &= X(z) + a_1 z^{-1} W(z) + a_2 z^{-2} W(z) \end{aligned}$$

which can be rearranged into the form

$$\begin{aligned} Y(z) &= (b_0 + b_1 z^{-1} + b_2 z^{-2}) W(z) \\ X(z) &= (1 - a_1 z^{-1} - a_2 z^{-2}) W(z) \end{aligned}$$

Since the system function $H(z)$ is the ratio of $Y(z)$ to $X(z)$, we get

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-2}}$$

Thus we have proved that the Direct Form II (DF-II) structure in Fig. 8-15 which implements the pair of difference equations (8.51) is identical to the system defined by the single difference equation (8.49).

The Transposed Direct Form II (TDF-II) in Fig. 8-16 can be worked out similarly. The difference equations represented by the block diagram are

$$\begin{aligned} y[n] &= b_0 x[n] + v_1[n-1] \\ v_1[n] &= b_1 x[n] + a_1 y[n] + v_2[n-1] \\ v_2[n] &= b_2 x[n] + a_2 y[n] \end{aligned} \tag{8.52}$$

Taking the z-transform of each of the three equations gives

$$\begin{aligned} Y(z) &= b_0 X(z) + z^{-1} V_1(z) \\ V_1(z) &= b_1 X(z) + a_1 Y(z) + z^{-1} V_2(z) \\ V_2(z) &= b_2 X(z) + a_2 Y(z) \end{aligned} \tag{8.53}$$

Using these equations, we can eliminate $V_1(z)$ and $V_2(z)$ as follows:

$$\begin{aligned} Y(z) &= b_0 X(z) + z^{-1}(b_1 X(z) + a_1 Y(z) + z^{-1} V_2(z)) \\ Y(z) &= b_0 X(z) + z^{-1}(b_1 X(z) + a_1 Y(z) \\ &\qquad\qquad + z^{-1}(b_2 X(z) + a_2 Y(z))) \end{aligned}$$

Moving all the $X(z)$ terms to the right-hand side and the $Y(z)$ terms to the left-hand side gives

$$(1 - a_1 z^{-1} - a_2 z^{-2}) Y(z) = (b_0 + b_1 z^{-1} + b_2 z^{-2}) X(z)$$

so we get by division

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-2}}$$

Thus we have shown that the Transposed Direct Form II (TDF-II) is equivalent to the system function for the basic Direct Form-I difference equation in (8.49). Both examples illustrate the power of the z-transform approach in manipulating polynomials that correspond to different structures.

In theory, the system with system function given by (8.50) can be implemented by iterating any of the equations (8.49), (8.51), or (8.52). For example, the MATLAB function `filter` uses the TDF-II structure. However, as mentioned before, the reason for having different block diagram structures is that the order of calculation defined by equations (8.49), (8.51), and (8.52) is different. In a hardware implementation, the different structures will behave differently, especially when using fixed-point arithmetic where rounding error is fed back into the structure. With double-precision floating-point arithmetic as in MATLAB, there is little difference.

EXERCISE 8.17: Draw the block diagram of the Direct Form I difference equation defined by (8.49), and compare it to the other block diagrams in Figs. 8-15 and 8-16.

8-10.3 Poles and Zeros

Finding the poles and zeros of $H(z)$ is less confusing if we rewrite the polynomials as functions of z rather than z^{-1}. Thus, the general second-order rational z-transform would become

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-2}} = \frac{b_0 z^2 + b_1 z + b_2}{z^2 - a_1 z - a_2}$$

after multiplying the numerator and denominator by z^2. Recall from algebra the following important property of polynomials:

> ***Property Of Real Polynomials***
>
> *A polynomial of degree N has N roots.*
> *If all the coefficients of the polynomial are real, the roots either must be real or must occur in* ***complex conjugate pairs****.*

Therefore, in the second-order case, the numerator and denominator polynomials each have two roots, and there are two possibilities: Either the roots are complex conjugates of each other, or they are both real. We will now concentrate on the roots of the denominator, but exactly the same results hold for the numerator. From the quadratic formula, we get two poles at

$$\frac{a_1 \pm \sqrt{a_1^2 + 4a_2}}{2}$$

When $a_1^2+4a_2 \geq 0$, both poles are real; when $a_1^2+4a_2 = 0$, they are real and equal. However, when $a_1^2+4a_2 < 0$, the square root produces an imaginary result, and we have complex-conjugate poles with values

$$\begin{aligned} p_1 &= \tfrac{1}{2}a_1 + j\tfrac{1}{2}\sqrt{-a_1^2 - 4a_2} \\ \text{and} \quad p_2 &= \tfrac{1}{2}a_1 - j\tfrac{1}{2}\sqrt{-a_1^2 - 4a_2} \end{aligned}$$

In polar form, the complex poles can be expressed as $p_1 = re^{j\theta}$ and $p_2 = re^{-j\theta}$, where the radius r is

$$\begin{aligned} r &= \sqrt{(\tfrac{1}{2}a_1)^2 + \tfrac{1}{4}(-a_1^2 - 4a_2)} \\ &= \sqrt{\tfrac{1}{4}a_1^2 - \tfrac{1}{4}a_1^2 - a_2} = \sqrt{-a_2} \end{aligned}$$

and the angle θ satisfies

$$r\cos\theta = \tfrac{1}{2}a_1 \qquad \Longrightarrow \qquad \theta = \cos^{-1}\left(\frac{a_1}{2\sqrt{-a_2}}\right)$$

Example 8-14: Complex Poles

The following $H(z)$ has two poles and two zeros.

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

The poles $\{p_1,\, p_2\}$ and zeros $\{z_1,\, z_2\}$ are

$$\begin{aligned} p_1 &= \tfrac{1}{2} + j\tfrac{1}{2}\sqrt{3} = e^{j\pi/3} \\ p_2 &= \tfrac{1}{2} - j\tfrac{1}{2}\sqrt{3} = e^{-j\pi/3} \\ z_1 &= 0 \\ z_2 &= -1 \end{aligned}$$

The system function can be written in factored form as either of the two forms

$$H(z) = \frac{2z(z+1)}{(z-e^{j\pi/3})(z-e^{-j\pi/3})}$$
$$= \frac{2(1+z^{-1})}{(1-e^{j\pi/3}z^{-1})(1-e^{-j\pi/3}z^{-1})}$$

Since the numerator has no z^{-2} term, we have one zero at the origin. As is our custom, we plot these locations in the z-plane and mark the pole locations with **x** and the zeros with **o**. See Fig. 8-17. ■

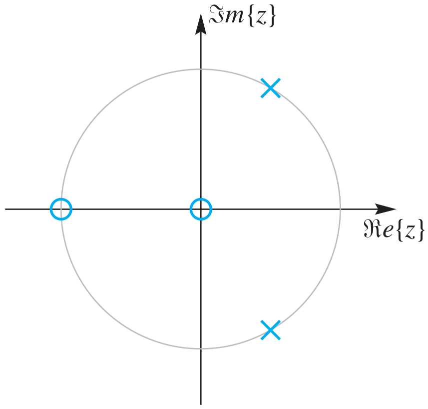

Figure 8-17: Pole-zero plot for system with $H(z) = \dfrac{2+2z^{-1}}{1-z^{-1}+z^{-2}}$. The unit circle is shown for reference.

8-10.4 Impulse Response of a Second-Order IIR System

We have derived the general z-transform system function for the second-order filter

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-2}} \tag{8.54}$$

and we have seen that the denominator polynomial $A(z)$ has two roots that define the poles of the second-order filter. Expressing $H(z)$ in terms of the poles gives

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{(1-p_1 z^{-1})(1-p_2 z^{-1})} \tag{8.55}$$

Using the partial fraction expansion technique developed in Section 8-8, we can express the system function (8.54) as

$$H(z) = (-b_2/a_2) + \frac{A_1}{1-p_1 z^{-1}} + \frac{A_2}{1-p_2 z^{-1}}$$

where A_1 and A_2 can be evaluated by $A_k = H(z)(1-p_k z^{-1})|_{z=p_k}$. Therefore, the impulse response will have the form

$$h[n] = (-b_2/a_2)\delta[n] + A_1(p_1)^n u[n] + A_2(p_2)^n u[n]$$

Furthermore, the poles may both be real or they may be a pair of complex conjugates. We will examine these two cases separately.

8-10.4.1 Real Poles

If p_1 and p_2 are real, the impulse response is composed of two real exponentials of the form p_k^n. This case is illustrated by the following example:

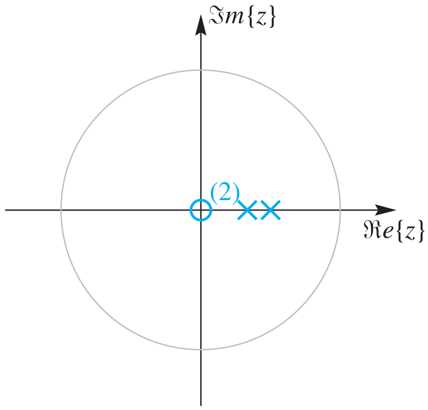

Figure 8-18: Pole-zero plot for system of Example 8-15. The poles are at $z = \frac{1}{2}$ and $z = \frac{1}{3}$; there are two zeros at $z = 0$.

Example 8-15: Second-Order: Real Poles

Assume that

$$\begin{aligned} H(z) &= \frac{1}{1 - \frac{5}{6}z^{-1} + \frac{1}{6}z^{-2}} \\ &= \frac{1}{(1 - \frac{1}{2}z^{-1})(1 - \frac{1}{3}z^{-1})} \end{aligned} \tag{8.56}$$

from which we see that the poles are at $z = \frac{1}{2}$ and $z = \frac{1}{3}$ and that there are two zeros at $z = 0$. The poles and zeros of $H(z)$ are plotted in Fig. 8-18. We can extract the filter coefficients from $H(z)$ and write the following difference equation

$$y[n] = \tfrac{5}{6}y[n-1] - \tfrac{1}{6}y[n-2] + x[n] \tag{8.57}$$

which must be satisfied for any input and its corresponding output. Specifically, the impulse response would satisfy the difference equation

$$h[n] = \tfrac{5}{6}h[n-1] - \tfrac{1}{6}h[n-2] + \delta[n] \tag{8.58}$$

which can be iterated to compute $h[n]$ if we know the values of $h[-1]$ and $h[-2]$, i.e., the values of the impulse response sequence just prior to $n = 0$ where the impulse first becomes nonzero. These values are supplied by the initial rest condition, which means that $h[-1] = 0$ and $h[-2] = 0$. The following table shows the computation of a few values of the impulse response:

n	$n < 0$	0	1	2	3	4	$\cdots$
$x[n]$	0	1	0	0	0	0	$\cdots$
$h[n-2]$	0	0	0	1	$\frac{5}{6}$	$\frac{19}{36}$	$\cdots$
$h[n-1]$	0	0	1	$\frac{5}{6}$	$\frac{19}{36}$	$\frac{65}{216}$	$\cdots$
$h[n]$	0	1	$\frac{5}{6}$	$\frac{19}{36}$	$\frac{65}{216}$	$\frac{211}{1296}$	$\cdots$

In contrast to the simpler first-order case, it is very difficult to guess the general n^{th} term for the impulse response sequence. Fortunately, we can rely on the inverse z-transform technique to give us the general formula. Applying the partial fraction expansion to (8.56), we get

$$H(z) = \frac{3}{1 - \frac{1}{2}z^{-1}} - \frac{2}{1 - \frac{1}{3}z^{-1}}$$

which implies that

$$\begin{aligned} h[n] &= 3(\tfrac{1}{2})^n\, u[n] - 2(\tfrac{1}{3})^n\, u[n] \\ &= \begin{cases} 3(\frac{1}{2})^n - 2(\frac{1}{3})^n & \text{for } n \geq 0 \\ 0 & \text{for } n < 0 \end{cases} \end{aligned}$$

Since both poles are inside the unit circle, the impulse response dies out for n large; i.e., the system is stable.

EXERCISE 8.18: Find the impulse response of the following second-order system:

$$y[n] = \tfrac{1}{4}y[n-2] + 5x[n] - 4x[n-1]$$

Plot the resulting signal $h[n]$ versus n.

8-10.5 Complex Poles

Now let us assume that the coefficients a_1 and a_2 in the second-order difference equation are such that the poles of $H(z)$ are complex. If we express the poles in polar form

$$p_1 = r\, e^{j\theta} \qquad \text{and} \qquad p_2 = r\, e^{-j\theta} = p_1^*$$

it is convenient to rewrite the denominator polynomial in terms of the parameters r and θ. Basic algebra allows us to start from the factored form and derive the polynomial coefficients:

$$\begin{aligned} A(z) &= \left(1 - p_1 z^{-1}\right)\left(1 - p_2 z^{-1}\right) \\ &= \left(1 - r\, e^{j\theta} z^{-1}\right)\left(1 - r\, e^{-j\theta} z^{-1}\right) \\ &= 1 - \left(r\, e^{j\theta} + r\, e^{-j\theta}\right) z^{-1} + r^2 z^{-2} \\ &= 1 - (2r\cos\theta)\, z^{-1} + r^2 z^{-2} \end{aligned} \tag{8.59}$$

The system function is therefore

$$\begin{aligned} H(z) &= \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{(1 - re^{j\theta}z^{-1})(1 - re^{-j\theta}z^{-1})} \\ &= \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 - 2r\cos\theta z^{-1} + r^2 z^{-2}} \end{aligned} \tag{8.60}$$

We can also identify the two feedback filter coefficients as

$$a_1 = 2r\cos\theta \qquad \text{and} \qquad a_2 = -r^2 \tag{8.61}$$

so the corresponding difference equation is

$$\begin{aligned} y[n] &= (2r\cos\theta)y[n-1] - r^2 y[n-2] \\ &\quad + b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] \end{aligned} \tag{8.62}$$

This parameterization is significant because it allows us to see directly how the poles define the feedback terms of the difference equation (8.62). For example, if we want to change the angle of the pole, then we vary the coefficient a_1. Finally, we must remember that (8.61) is valid only for the special case of complex-conjugate poles; when the poles $(p_1,\, p_2)$ are both real, the filter coefficients are

$$a_1 = p_1 + p_2 \qquad \text{and} \qquad a_2 = p_1 p_2$$

Example 8-16: Invert Complex Poles

Consider the following system

$$y[n] = y[n-1] - y[n-2] + 2x[n] + 2x[n-1]$$

whose system function is

$$\begin{aligned} H(z) &= \frac{2 + 2z^{-1}}{1 - z^{-1} + z^{-2}} \\ &= \frac{2(1+z^{-1})}{(1 - e^{j\pi/3}z^{-1})(1 - e^{-j\pi/3}z^{-1})} \end{aligned} \tag{8.63}$$

A pole-zero plot for $H(z)$ was already given in Fig. 8-17. Using the partial fraction expansion technique, we can write $H(z)$ in the form

$$\begin{aligned} H(z) &= \frac{\left(\dfrac{2+2e^{-j\pi/3}}{1-e^{-j2\pi/3}}\right)}{1 - e^{j\pi/3}z^{-1}} + \frac{\left(\dfrac{2+2e^{j\pi/3}}{1-e^{j2\pi/3}}\right)}{1 - e^{-j\pi/3}z^{-1}} \\ &= \frac{2e^{-j\pi/3}}{1 - e^{j\pi/3}z^{-1}} + \frac{2e^{j\pi/3}}{1 - e^{-j\pi/3}z^{-1}} \end{aligned}$$

$$\begin{aligned} \text{so}\quad h[n] &= 2e^{-j\pi/3}e^{j(\pi/3)n}u[n] + 2e^{j\pi/3}e^{-j(\pi/3)n}u[n] \\ &= 4\cos\left(2\pi(\tfrac{1}{6})(n-1)\right)u[n] \end{aligned}$$

The two complex exponentials with frequencies $\pm\pi/3$ combine to form the cosine. The impulse response is plotted in Fig. 8-19. ■

An important observation about the system in Example 8-16 is that it produces a pure sinusoid when stimulated by an impulse. Such a system is an example of a ***sine wave oscillator***. After being stimulated by the single input sample from the impulse, the system continues indefinitely to produce a sinusoid of frequency $\hat{\omega}_0 = 2\pi(\frac{1}{6})$. This frequency is equal to the angle of the poles. A first-order filter (or a filter with all real

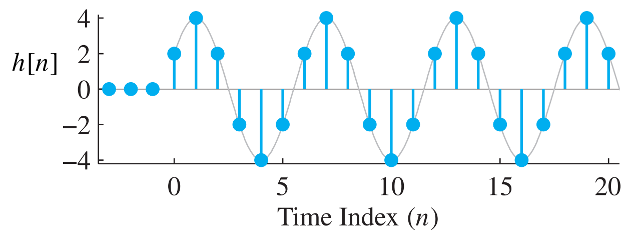

Figure 8-19: Impulse response for system with $A(z) = 1 - z^{-1} + z^{-2}$.

poles) can only decay (or grow) as $(p)^n$ or oscillate up and down as $(-1)^n$, but a second-order system can oscillate with different periods. This is important when modeling physical signals such as speech, music, or other sounds.

Note that in order to produce the continuing sinusoidal output, the system must have its poles on the unit circle[7] of the z-plane, i.e., $r = 1$. Also note that the angle of the poles is exactly equal to the radian frequency of the sinusoidal output. Thus, we can control the frequency of the sinusoidal oscillator by adjusting the a_1 coefficient of the difference equation (8.62) while leaving a_2 fixed at $a_2 = -1$.

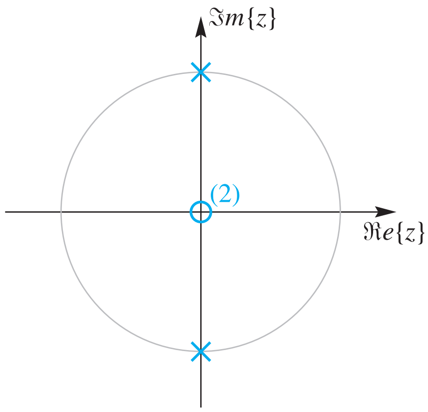

Figure 8-20: Pole-zero plot for system with $H(z) = \dfrac{1}{1 + z^{-2}}$. The unit circle is shown for reference.

Example 8-17: Poles on Unit Circle

As an example of an oscillator with a different frequency, we can use (8.62) to define a difference equation with prescribed pole locations. If we take $r = 1$ and $\theta = \pi/2$, as shown in Fig. 8-20, we get $a_1 = 2r\cos\theta = 0$ and $a_2 = -r^2 = -1$.

$$y[n] = -y[n-2] + x[n] \tag{8.64}$$

[7]Strictly speaking, a system with poles on the unit circle is unstable, so for some inputs the output may increase without bound. As in the case of the accumulator system in Section 8-4.1, the impulse response of this second-order system does not blow up.

The system function of this system is

$$\begin{aligned} H(z) &= \frac{1}{1+z^{-2}} \\ &= \frac{1}{(1-e^{j\pi/2}z^{-1})(1-e^{-j\pi/2}z^{-1})} \\ &= \frac{\frac{1}{2}}{1-e^{j\pi/2}z^{-1}} + \frac{\frac{1}{2}}{1-e^{-j\pi/2}z^{-1}} \end{aligned}$$

The inverse z-transform gives a general formula for $h[n]$:

$$\begin{aligned} h[n] &= \tfrac{1}{2}e^{j(\pi/2)n}u[n] + \tfrac{1}{2}e^{-j(\pi/2)n}u[n] \\ &= \begin{cases} \cos(2\pi\left(\frac{1}{4}\right)n) & \text{for } n \geq 0 \\ 0 & \text{for } n < 0 \end{cases} \end{aligned} \tag{8.65}$$

Once again, the frequency of the cosine term in the impulse response is equal to the angle of the pole, $\pi/2 = 2\pi(\frac{1}{4})$.

If the complex conjugate poles of the second-order system lie on the unit circle, the output oscillates sinusoidally and does not decay to zero. If the poles lie outside the unit circle, the output grows exponentially, whereas if they are inside the unit circle, the output decays exponentially to zero.

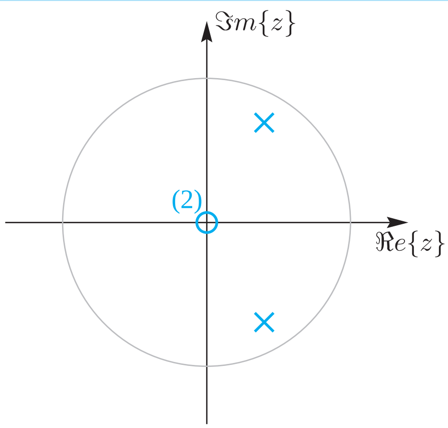

Figure 8-21: Pole-zero plot for system with $H(z) = \dfrac{1}{1-0.8z^{-1}+0.64z^{-2}}$. The unit circle is shown for reference.

Example 8-18: Stable Complex Poles

As an example of a stable system where the poles are inside the unit circle, if we take $r = 0.8$ and $\theta = \pi/3$, as shown in Fig. 8-21, we get $a_1 = 2r\cos\theta = 2(0.8)(\frac{1}{2}) = 0.8$ and $a_2 = -r^2 = -(0.8)^2 = -0.64$, and the difference equation (8.62) becomes

$$y[n] = 0.8y[n-1] - 0.64y[n-2] + x[n] \tag{8.66}$$

The system function of this system is

$$\begin{aligned} H(z) &= \frac{1}{1 - 0.8z^{-1} + 0.64z^{-2}} \\ &= \frac{\frac{1}{\sqrt{3}}e^{-j\pi/6}}{1 - 0.8e^{j\pi/3}z^{-1}} + \frac{\frac{1}{\sqrt{3}}e^{j\pi/6}}{1 - 0.8e^{-j\pi/3}z^{-1}} \end{aligned}$$

and the general formula for $h[n]$ is

$$h[n] = \frac{2}{\sqrt{3}}(0.8)^n \cos\left(2\pi(\tfrac{1}{6})n - \tfrac{\pi}{6}\right) u[n] \tag{8.67}$$

In this case, the general formula for $h[n]$ has a decay of $(0.8)^n$ multiplying a periodic cosine with period 6. The frequency of the cosine term in the impulse response (8.67) is again the angle of the pole, $\pi/3 = 2\pi/6$; while the decaying term is controlled by the radius of the pole, i.e., $r^n = (0.8)^n$. Figure 8-22 shows $h[n]$ for this example. Note the rapid decay of the impulse response. Also note that this rapid decay can be predicted from the pole-zero plot in Fig. 8-21, since, in general, poles that are not close to the unit circle will correspond to rapidly decaying exponential components of the impulse response. The further the poles are away from and inside of the unit circle, the faster the impulse response will decay toward zero.

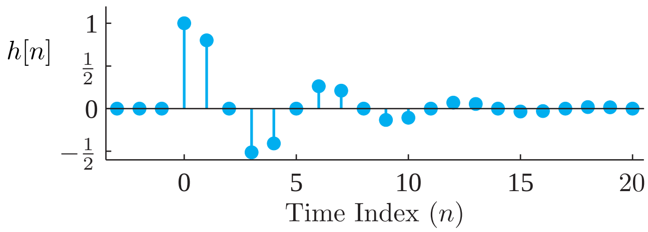

Figure 8-22: Impulse response for system with $A(z) = 1 - 0.8z^{-1} + 0.64z^{-2}$.

DEMO: *IIR Filtering*

8-11 Frequency Response of Second-Order IIR Filter

Since the frequency response of a stable system is related to the z-transform by

$$H(e^{j\hat{\omega}}) = H(z)\big|_{z=e^{j\hat{\omega}}}$$

we get the following formula for the frequency response of a second-order system:

$$H(e^{j\hat{\omega}}) = \frac{b_0 + b_1 e^{-j\hat{\omega}} + b_2 e^{-j2\hat{\omega}}}{1 - a_1 e^{-j\hat{\omega}} - a_2 e^{-j2\hat{\omega}}} \tag{8.68}$$

Since (8.68) contains terms like $e^{-j\hat{\omega}}$ and $e^{-j2\hat{\omega}}$, $H(e^{j\hat{\omega}})$ is guaranteed to be a periodic function with a period of 2π.

The magnitude squared of the frequency response can be formed by multiplying the complex $H(e^{j\hat{\omega}})$ by its conjugate (denoted by $H^*(e^{j\hat{\omega}})$). Rather than work out a general formula, we take a specific numerical example to show the kind of formula that results.

Example 8-19: Frequency Response of a Second-Order System

Consider the case where the system function is

$$H(z) = \frac{1 - z^{-2}}{1 - 0.9z^{-1} + 0.81z^{-2}}$$

The magnitude squared is derived by multiplying out all the terms in the numerator and denominator of $H(e^{j\hat{\omega}})H^*(e^{j\hat{\omega}})$, and then collecting terms where the inverse Euler formula applies.

$$\begin{aligned} |H(e^{j\hat{\omega}})|^2 &= H(e^{j\hat{\omega}})H^*(e^{j\hat{\omega}}) \\ &= \frac{1 - e^{-j2\hat{\omega}}}{1 - 0.9e^{-j\hat{\omega}} + 0.81e^{-j2\hat{\omega}}} \cdot \frac{1 - e^{j2\hat{\omega}}}{1 - 0.9e^{j\hat{\omega}} + 0.81e^{j2\hat{\omega}}} \\ &= \frac{2 + 2\cos(2\hat{\omega})}{2.4661 - 3.258\cos\hat{\omega} + 1.62\cos(2\hat{\omega})} \end{aligned}$$

This formula is useful because it is expressed completely in terms of cosine functions. The procedure is general, so a similar formula could be derived for any IIR filter. Since the cosine is an even function, we can state that any magnitude-squared function $|H(e^{j\hat{\omega}})|^2$ will always be even; i.e.,

$$|H(e^{-j\hat{\omega}})|^2 = |H(e^{j\hat{\omega}})|^2$$

The phase response is a bit messier. If arctangents are used to extract the angle of the numerator and denominator, then the two phases must be subtracted. The filter coefficients in this example are real, so the phase is

$$\begin{aligned} \phi(\hat{\omega}) = \tan^{-1}&\left(\frac{\sin(2\hat{\omega})}{1 - \cos(2\hat{\omega})}\right) \\ &- \tan^{-1}\left(\frac{0.9\sin\hat{\omega} - 0.81\sin(2\hat{\omega})}{1 - 0.9\cos\hat{\omega} + 0.81\cos(2\hat{\omega})}\right) \end{aligned}$$

which is an odd function of $\hat{\omega}$.

This example demonstrates a basic property of the frequency response (or any DTFT) of a real impulse response (or any real sequence). As discussed in Sec. 6-4.3 for the FIR case, the frequency response has conjugate symmetry such that

$$H(e^{-j\hat{\omega}}) = H^*(e^{j\hat{\omega}}) \tag{8.69}$$

whenever the impulse response is real. ■

The formulas obtained in this example are too complicated to provide much insight directly. In a later section we will see how to use the poles and zeros of the system function to construct an approximate plot of the frequency response without recourse to such formulas.

8-11.1 Frequency Response via MATLAB

Tedious calculation and plotting of $H(e^{j\hat{\omega}})$ by hand is usually unnecessary if a computer program such as MATLAB is available. The MATLAB function `freqz` is provided for just that purpose. The frequency response can be evaluated over a grid in the $\hat{\omega}$ domain, and then its magnitude and phase can be plotted. In MATLAB, the functions `abs` and `angle` will extract the magnitude and the angle of each element in a complex vector.

Example 8-20: MATLAB for $H(e^{j\hat{\omega}})$

Consider the system introduced in Example 8-19:

$$y[n] = 0.9y[n-1] - 0.81y[n-2] + x[n] - x[n-2]$$

In order to define the filter coefficients in MATLAB, we put all the terms with $y[n]$ on one side of the equation, and the terms with $x[n]$ on the other.

$$y[n] - 0.9y[n-1] + 0.81y[n-2] = x[n] - x[n-2]$$

Then we read off the filter coefficients and define the vectors aa and bb.

```
aa = [ 1, −0.9, 0.81 ]
bb = [ 1, 0, −1 ]
```

The following call to freqz will generate a vector HH containing the values of the frequency response at the vector of frequencies specified by the third argument, [-pi:(pi/100):pi].

```
HH = freqz(bb, aa, [-pi:(pi/100):pi])
```

A plot of the resulting magnitude and phase is shown in Fig. 8-23. Since $H(e^{j\hat{\omega}})$ is always periodic with a period of 2π, it is sufficient to make the frequency response plot over the range $-\pi \le \hat{\omega} \le \pi$.

For this example, we can look for a connection between the poles and zeros and the shape of the frequency response. For this $H(z)$ we have

$$H(z) = \frac{1 - z^{-2}}{1 - 0.9z^{-1} + 0.81z^{-2}}$$

which has its poles at $z = 0.9e^{\pm j\pi/3}$ and its zeros at $z = 1$ and $z = -1$. Since $z = -1$ is the same as $z = e^{j\pi}$, we conclude that $H(e^{j\hat{\omega}})$ is zero at $\hat{\omega} = \pi$, because $H(z) = 0$ at $z = -1$; likewise, the zero of $H(z)$ at $z = +1$ is a zero of $H(e^{j\hat{\omega}})$ at $\hat{\omega} = 0$. The poles have angles of $\pm\pi/3$ rad, so the poles have an effect on the frequency response near $\hat{\omega} = \pm\pi/3$. Since $H(z)$ is infinite at $z = 0.9e^{\pm j\pi/3}$, the nearby points on the unit circle (at $z = e^{\pm j\pi/3}$) must have large values. In this case, we can evaluate the frequency response directly from the formula to get

$$\begin{aligned}
H(e^{j\hat{\omega}})\big|_{\hat{\omega}=\pi/3} &= H(z)\big|_{z=e^{j\pi/3}} \\
&= \left.\frac{1 - z^{-2}}{1 - 0.9z^{-1} + 0.81z^{-2}}\right|_{z=e^{j\pi/3}} \\
&= \frac{1 - (-\frac{1}{2} - j\frac{1}{2}\sqrt{3})}{1 - 0.9(\frac{1}{2} - j\frac{1}{2}\sqrt{3}) + 0.81(-\frac{1}{2} - j\frac{1}{2}\sqrt{3})} \\
&= \frac{|1.5 + j0.5(\sqrt{3})|}{|0.145 + j0.045(\sqrt{3})|} = 10.522
\end{aligned}$$

This value of the frequency response magnitude is a good approximation to the true maximum value, which actually occurs at $\hat{\omega} = 0.334\pi$. ■

8-11.2 3-dB Bandwidth

The width of the peak of the frequency response in Fig. 8-23 is called the ***bandwidth***. It must be measured at some standard point on the plot of $|H(e^{j\hat{\omega}})|$. The most common practice is to use the 3-dB width, which is calculated as follows:

> Determine the peak value for $|H(e^{j\hat{\omega}})|$ and then find the nearest frequency on each side of the peak where the value of the frequency response is $(1/\sqrt{2})H_{\text{peak}}$. The ***3-dB width*** is the difference $\Delta\hat{\omega}$ between these two frequencies.

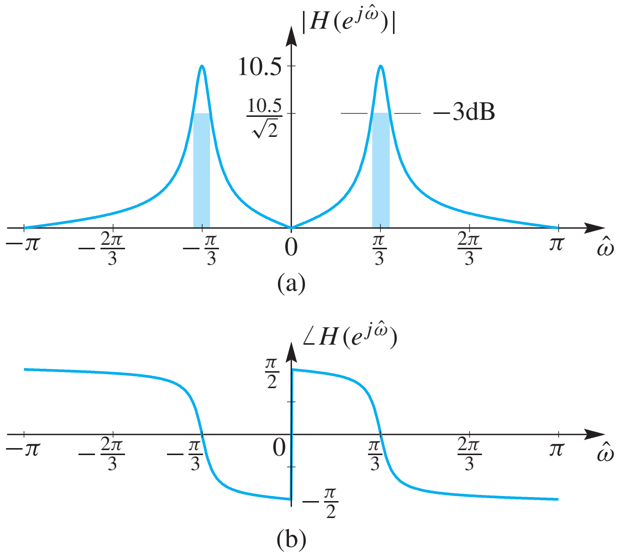

Figure 8-23: Frequency response (magnitude and phase) for a second-order IIR filter. The poles are at $z = 0.9e^{\pm j\pi/3}$ and the numerator has zeros at $z = 1$ and $z = -1$. The shaded region shows the 3-dB bandwidth around the peak at $\hat{\omega} = \pi/3$.

In Fig. 8-23, the true peak value is 10.526 at $\hat{\omega} = 0.334\pi$, so we look for points where $|H(e^{j\hat{\omega}})| = (1/\sqrt{2})H_{\text{peak}} = (0.707)(10.526) = 7.443$. These occur at $\hat{\omega} = 0.302\pi$ and $\hat{\omega} = 0.369\pi$, so the bandwidth is $\Delta\hat{\omega} = 0.067\pi = 2\pi(0.0335) = 0.2105$ rad.

It is common practice to plot $20\log_{10}|H(e^{j\hat{\omega}})|$ when plotting frequency responses that vary over a wide range of amplitudes; e.g., frequency-selective filters where passbands are one and stop bands approximate zero. The units of $20\log_{10}|H(e^{j\hat{\omega}})|$ are defined as decibels, or dB for short. The terminology "3 dB" results from the fact that $20\log_{10}|(1/\sqrt{2})H_{\text{peak}}| = 20\log_{10}|H_{\text{peak}}| - 3.01$. If we plot $20\log_{10}|H(e^{j\hat{\omega}})|$ instead of $|H(e^{j\hat{\omega}})|$, we obtain the plot in Fig. 8-24, where the maximum value is 20.42 dB so the 3 dB bandwidth points are the frequencies where $20\log_{10}|H(e^{j\hat{\omega}})| = 20.42 - 3 = 17.42$. Note how $20\log_{10}|H(e^{j\hat{\omega}})|$ approaches $= -\infty$ at the frequencies corresponding to the zeros of $H(z)$ at $z = \pm 1$ (where $|H(e^{j0})| = 0$ and $|H(e^{j\pi})| = 0$). Other things to note in Fig. 8-24 are that $20\log_{10}|H(e^{j\hat{\omega}})| = 0$ when $|H(e^{j\hat{\omega}})| = 1$. This occurs in Fig. 8-24 at frequencies $\hat{\omega} \approx \pm\pi/8$ and $\hat{\omega} \approx \pm 2\pi/3$. Furthermore, it follows that if $|H(e^{j\hat{\omega}})| < 1$, then $20\log_{10}|H(e^{j\hat{\omega}})| < 0$. In this way, the log function expands the details when $H(e^{j\hat{\omega}})$ is small.

The 3-dB bandwidth calculation can be carried out efficiently with a computer program, but it is also helpful to have an approximate formula that can give quick "back-of-the-envelope" calculations. An excellent

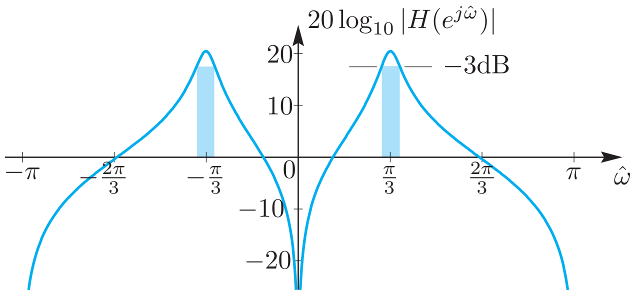

Figure 8-24: Log magnitude of frequency response corresponding to Fig. 8-23(a). The shaded region shows the 3-dB bandwidth around the peak at $\hat\omega = \pi/3$.

approximation for the second-order case with narrow peaks is given by the formula

$$\Delta\hat\omega \approx 2\,\frac{|1-r|}{\sqrt{r}} \tag{8.70}$$

which shows that the distance of the pole from the unit circle $|1-r|$ controls the bandwidth.[8] In Fig. 8-23, the bandwidth (8.70) evaluates to

$$\Delta\hat\omega = 2\,\frac{(1-0.9)}{0.95} = \frac{0.2}{0.95} \approx 0.2108 \ \ \text{rad}$$

Thus we see that the approximation is quite good in this case, where the pole is rather close to the unit circle (radius = 0.9).

8-11.3 Three-Dimensional Plot of System Functions

Since the frequency response $H(e^{j\hat\omega})$ is the system function evaluated on the unit circle, we can illustrate the connection between the z and $\hat\omega$ domains with a three-dimensional plot such as the one shown in Fig. 8-25.

Figure 8-25 shows a plot of the system function $H(z)$ at points inside, outside, and on the unit circle. The peaks located at the poles, $0.85e^{\pm j\pi/2}$, determine the frequency response behavior near $\hat\omega = \pm\pi/2$. If the poles were moved closer to the unit circle, the frequency response would have a higher and narrower peak. The zeros at $z = \pm 1$ create valleys that lie on the unit circle at $\hat\omega = 0, \pi$. I think we should make this right. The unit circle evaluation should be hidden or dotted when it goes behind the pole.

DEMO: *Z to Freq*

We can estimate any value of $|H(e^{j\hat\omega})|$ directly from the poles and zeros. This can be done systematically by writing $H(z)$ in the following form:

$$H(z) = G\,\frac{(z-z_1)(z-z_2)}{(z-p_1)(z-p_2)}$$

where z_1 and z_2 are the zeros and p_1 and p_2 are the poles of the second-order filter. The parameter G is a gain term that may have to be factored out. Then the magnitude of the frequency response is

$$|H(e^{j\hat\omega})| = G\,\frac{|e^{j\hat\omega}-z_1|\,|e^{j\hat\omega}-z_2|}{|e^{j\hat\omega}-p_1|\,|e^{j\hat\omega}-p_2|} \tag{8.71}$$

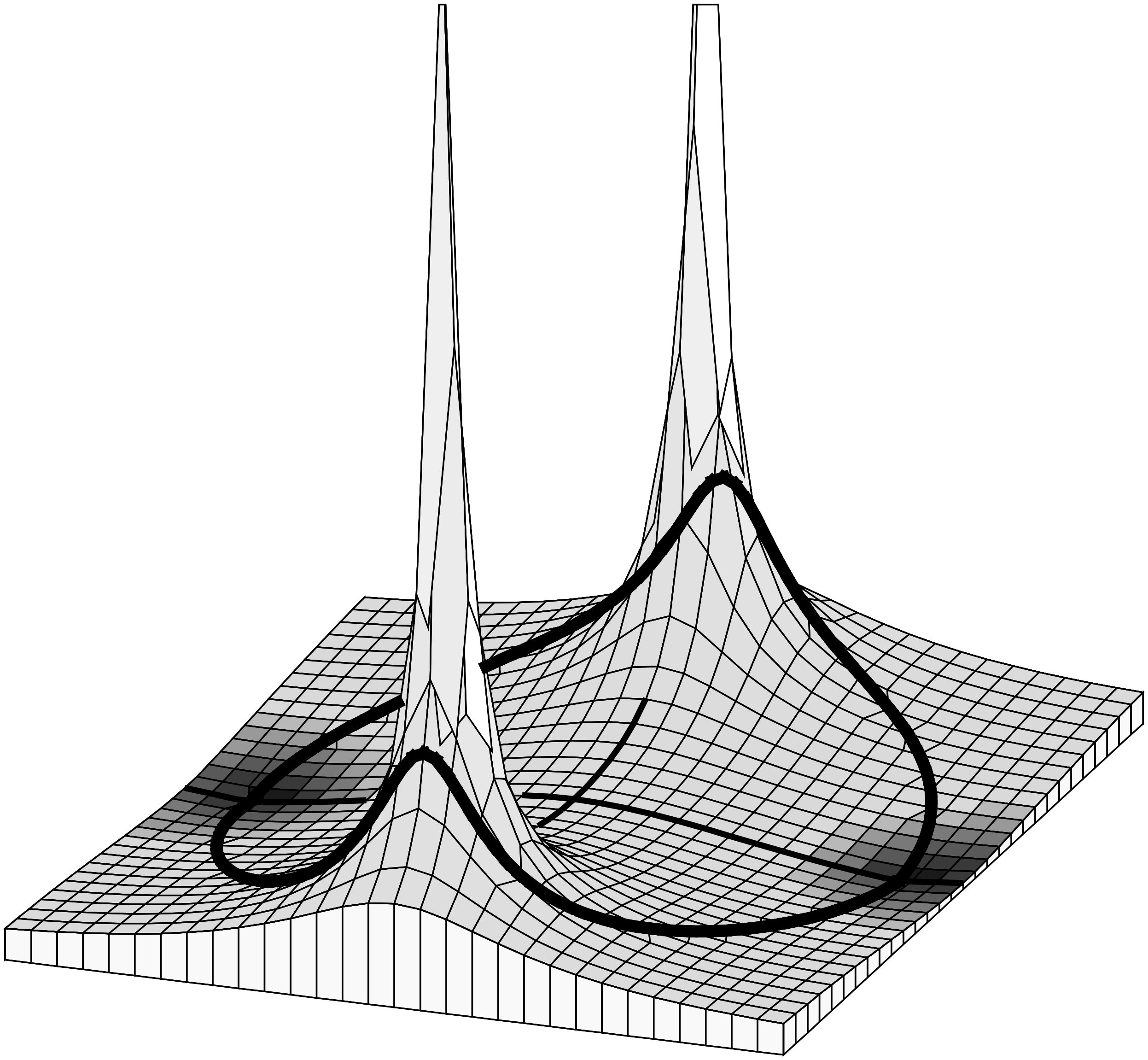

Figure 8-25: z-transform evaluated over a region of the z-plane including the unit circle. The view is from the fourth quadrant, so the $z = 1$ point is at the right. Values along the unit circle are the frequency response (magnitude) for a second-order IIR filter. The poles are at $z = 0.85e^{\pm j\pi/2}$ and the numerator has zeros at $z = \pm 1$.

Equation (8.71) has a simple geometric interpretation. Each term $|e^{j\hat{\omega}} - z_i|$ or $|e^{j\hat{\omega}} - p_i|$ is the vector length from the zero z_i or the pole p_i to the unit circle position $e^{j\hat{\omega}}$, shown in Fig. 8-26. The frequency response at a fixed value of $\hat{\omega}$ is the product of G times the product of the lengths of the vectors to the zeros divided by the product of the lengths of the vectors to the poles.

$$|H(e^{j\hat{\omega}})| = G\frac{\overline{Z_1 Z} \cdot \overline{Z_2 Z}}{\overline{P_1 Z} \cdot \overline{P_2 Z}}$$

[8]This approximate formula for bandwidth is good only when the poles are isolated from one another. The approximation breaks down, for example, when a second-order system has two poles with small angles.

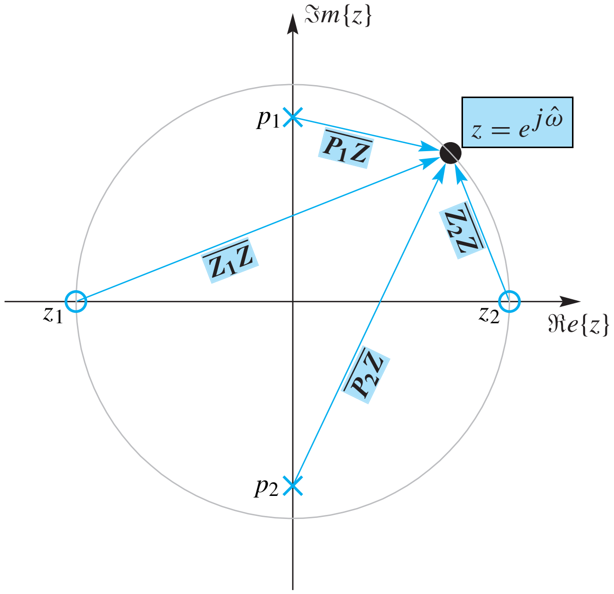

Figure 8-26: z-transform evaluated on the unit circle ($z = e^{j\hat{\omega}}$) by using a product of vector lengths from the poles and zeros. $\overline{\boldsymbol{P_i Z}}$ denotes the vector from the i^{th} pole to ($z = e^{j\hat{\omega}}$), and $\overline{\boldsymbol{Z_j Z}}$ denotes the vector from the j^{th} zero to ($z = e^{j\hat{\omega}}$),

As we go around the unit circle, these vector lengths change. When we are on top of a zero, one of the numerator lengths is zero, so $|H(e^{j\hat{\omega}})| = 0$ at that frequency. When we are close to a pole, one of the denominator lengths is very small, so $|H(e^{j\hat{\omega}})|$ will be large at that frequency.

We can apply this geometric reasoning to estimate the magnitude of $H(e^{j\hat{\omega}})$ at $\hat{\omega} = \pi/2$ in Fig. 8-25. We begin by estimating the lengths of the vectors from the zeros and poles to the point $z = e^{j\pi/2}$, which is the same as $z = j$. The lengths of the vectors from the zeros are then divided by the lengths of the vectors from the poles, so we get

$$\begin{aligned} |H(e^{j\pi/2})| &= |H(j)| \\ &= \frac{|j-1|\,|j+1|}{|j-0.85j|\,|j-(-0.85j)|} \\ &= \frac{2}{0.15 \times 1.85} = 7.207 \end{aligned}$$

The gain G was assumed to be 1.

An excellent way to practice with these ideas is to use the MATLAB GUI called `PeZ` (see Fig. 8-27).

8-12 Example of an IIR Lowpass Filter

First-order and second-order IIR filters are useful and provide simple examples, but, in many cases, we use higher-order IIR filters because they can realize frequency responses with flatter passbands and stopbands and sharper transition regions. The `butter`, `cheby1`, `cheby2`, and `ellip` functions in MATLAB's *Signal*

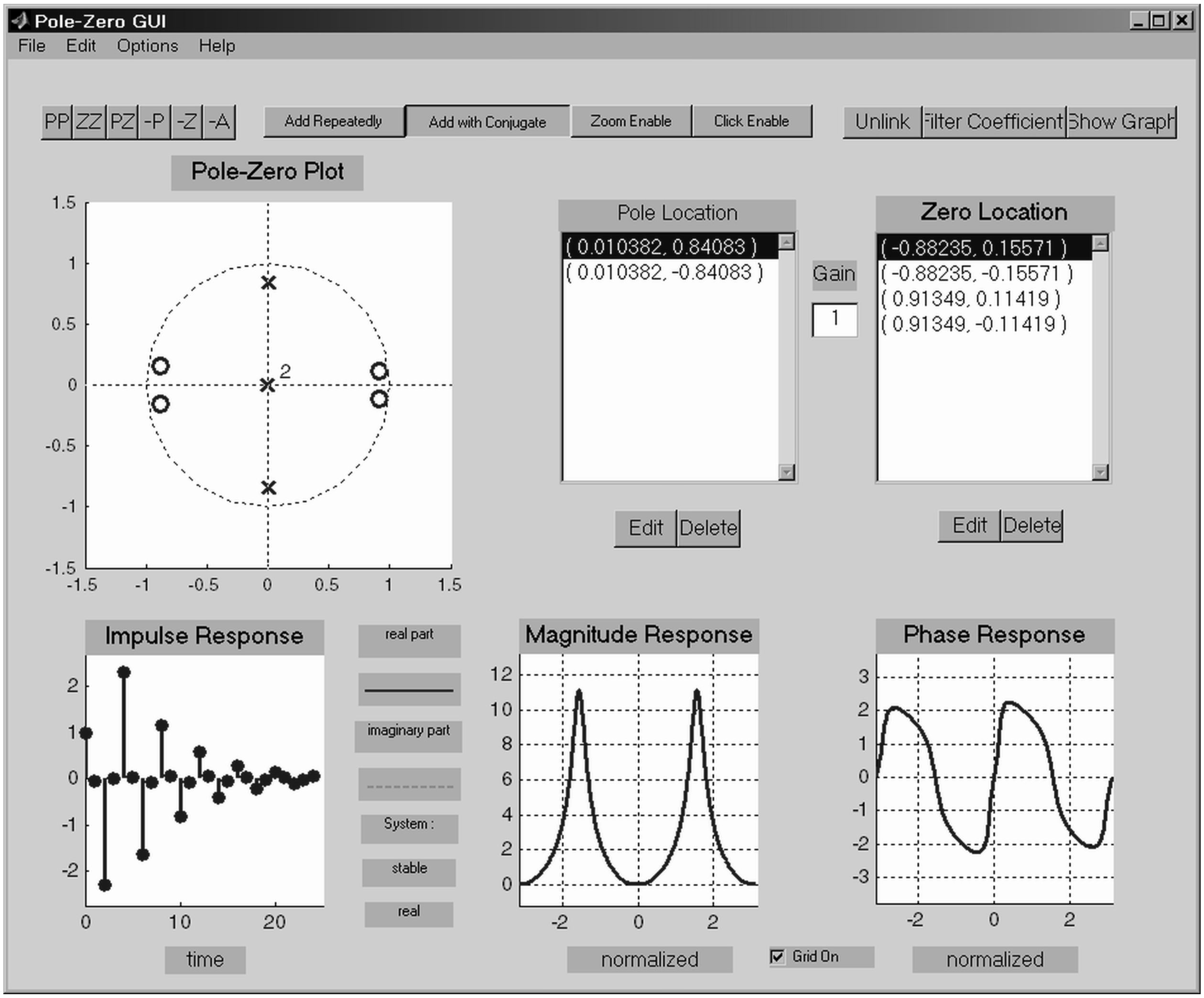

Figure 8-27: Graphical User Interface (GUI) for `PeZ` which illustrates the *three-domains* concept. The user can enter poles and zeros and see the resulting impulse response and frequency response. In addition, the poles/zero can be moved with a mouse pointer and and the other plots will be updated in real-time.

$$H(z) = \frac{0.0798(1+z^{-1}+z^{-2}+z^{-3})}{1-1.556z^{-1}+1.272z^{-2}-0.398z^{-3}} \tag{8.72}$$

$$= \frac{0.0798(1+z^{-1})(1-e^{j\pi/2}z^{-1})(1-e^{-j\pi/2}z^{-1})}{(1-0.556z^{-1})(1-0.846e^{j0.3\pi}z^{-1})(1-0.846e^{-j0.3\pi}z^{-1})} \tag{8.73}$$

$$= -\frac{1}{5} + \frac{0.62}{1-.556z^{-1}} + \frac{0.17e^{j0.96\pi}}{1-.846e^{j0.3\pi}z^{-1}} + \frac{0.17e^{-j0.96\pi}}{1-.846e^{-j0.3\pi}z^{-1}} \tag{8.74}$$

Processing Toolbox can be used to design filters with prescribed frequency-selective characteristics. As an example, consider the system with system function $H(z)$ given by (8.72).

This system is an example of a lowpass ***elliptic filter*** whose numerator and denominator coefficients were obtained using the MATLAB function `ellip`. The exact call was `ellip(3,1,30,0.3)`. Each of the three different forms above is useful: (8.72) for identifying the filter coefficients, (8.73) for sketching the pole-zero plot and the frequency response, and (8.74) for finding the impulse response.[9] Figure 8-28 shows the poles and zeros of this filter. Note that all the zeros are on the unit circle and that the poles are strictly inside the unit circle, as they must be for a stable system.

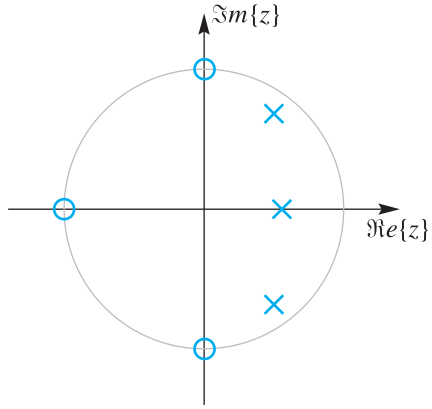

Figure 8-28: Pole-zero plot for a third-order IIR filter (8.73).

EXERCISE 8.19: From (8.72) determine the difference equation (Direct Form I) for implementing this system.

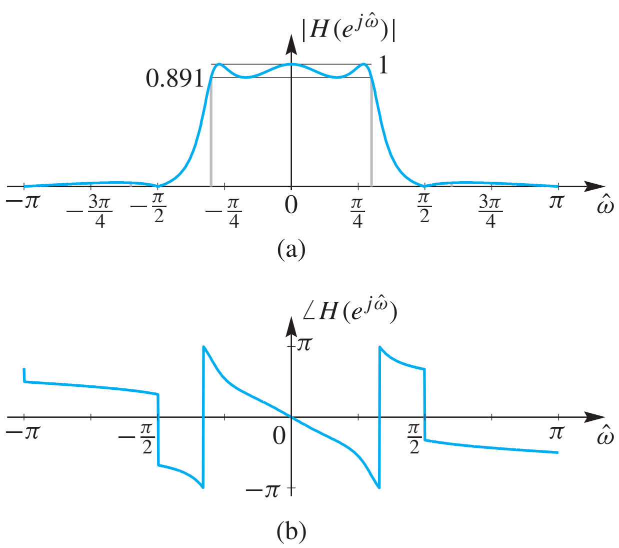

Figure 8-29: Frequency response (magnitude and phase) for a third-order IIR filter.

[9]Factoring polynomials and obtaining the partial fraction expansion was done in MATLAB using the functions `roots` and `residuez`, respectively.

The system function was evaluated on the unit circle using the MATLAB function `freqz`. A plot of this result is shown in Fig. 8-29. Note that the frequency response is large in the vicinity of the poles and small around the zeros. In particular, the passband of the frequency response is $|\hat{\omega}| \leq 2\pi(0.15)$, which corresponds to the poles with angles at $\pm 0.3\pi$. Also, the frequency response is exactly zero at $\hat{\omega} = \pm 0.5\pi$ and $\hat{\omega} = \pi$ since the zeros of $H(z)$ are at these angles and lie on the unit circle.

EXERCISE 8.20: From (8.72) or (8.73), determine the value of the frequency response at $\hat{\omega} = 0$.

Finally, Fig. 8-30 shows the impulse response of the system. Note that it oscillates and dies out with increasing n because of the two complex conjugate poles at angles $\pm 0.3\pi$ and radius 0.846. The decaying envelope is $(0.846)^n$.

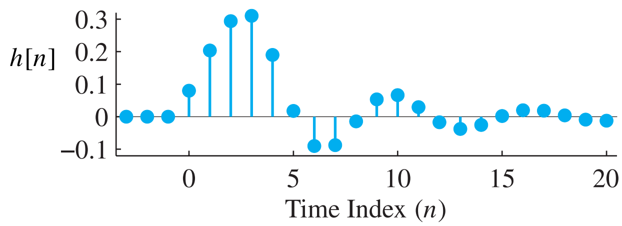

Figure 8-30: Impulse response for a third-order IIR filter.

EXERCISE 8.21: Use the partial fraction form (8.74) to determine an equation for the impulse response of the filter.

Hint: Apply the inverse z-transform.

The elliptic filter example described in this section is a simple example of a practical IIR lowpass filter. Higher-order filters can exhibit much better frequency-selective filter characteristics.

8-13 Summary and Links

The class of IIR filters was introduced in this chapter, along with the z-transform method for filters with poles. The z-transform changes problems about impulse responses, frequency responses, and system structures into algebraic manipulations of polynomials and rational functions. Poles of the system function $H(z)$ turn out to be the most important elements for IIR filters because properties such as the shape of the frequency response or the form of the impulse response can be inferred quickly from the pole locations.

We also continued to stress the important concept of "domains of representation." The n-domain or time domain, the $\hat{\omega}$-domain or frequency domain, and the z-domain give us three domains for thinking about the characteristics of a system. We completed the ties between domains by introducing the inverse z-transform for constructing a signal from its z-transform. As a result, even difficult problems such as convolution can be simplified by working in the most convenient domain (z) and then transforming back to the original domain (n).

Lab #11 is devoted to IIR filters. This lab uses a MATLAB user interface tool called `PeZ` that supports an interactive exploration of the three domains.

LAB: *#11 PeZ - the z, n, and $\hat{\omega}$ Domains*

The PeZ tool is useful for studying IIR systems because it presents the user with multiple views of an LTI system: pole-zero domain, frequency response and impulse response. (See Fig. 8-27.) Similar capabilities are now being incorporated into many commercial software packages (e.g., `sptool` in MATLAB).

The CD-ROM also contains the following demonstrations of the relationship between the z-plane and the frequency domain and time domain:

(a) A set of "three-domain" movies that show how the frequency response and impulse response of an IIR filter change as a pole location is varied. Several different filters are demonstrated.

DEMO: *Three Domains - IIR*

(b) A movie that animates the relationship between the z-plane and the unit circle where the frequency response lies.

DEMO: *Z to Freq*

(c) The PeZ GUI can be used to construct different IIR and FIR filters.

DEMO: *PeZ GUI*

(d) A tutorial on how to use PeZ.

DEMO: *PeZ Tutorial*

(e) A demo that gives more examples of IIR filters.

DEMO: *IIR Filtering*

The reader is again reminded of the large number of solved homework problems on the CD-ROM that are available for review and practice.

NOTE: *Hundreds of Solved Problems*