# 9.1 Introduction

In Chapter 8, we saw that the root locus graphically displayed both transient response and stability information. The locus can be sketched quickly to get a general idea of the changes in transient response generated by changes in gain. Specific points on the locus also can be found accurately to give quantitative design information.

The root locus typically allows us to choose the proper loop gain to meet a transient response specification. As the gain is varied, we move through different regions of response. Setting the gain at a particular value yields the transient response dictated by the poles at that point on the root locus. Thus, *we are limited to those responses that exist along the root locus.*

## Improving Transient Response

Flexibility in the design of a desired transient response can be increased if we can design for transient responses that are not on the root locus. Figure 9.1(*a*) illustrates the concept. Assume that the desired transient response, defined by percent overshoot and settling time, is represented by point *B*. Unfortunately, on the current root locus at the specified percent overshoot, we only can obtain the settling time represented by point *A* after a simple gain adjustment. Thus, our goal is to speed up the response at *A* to that of *B*, without affecting the percent overshoot. This increase in speed cannot be accomplished by a simple gain adjustment, since point *B* does not lie on the root locus. Figure 9.1(*b*) illustrates the improvement in the transient response we seek: The faster response has the same percent overshoot as the slower response.

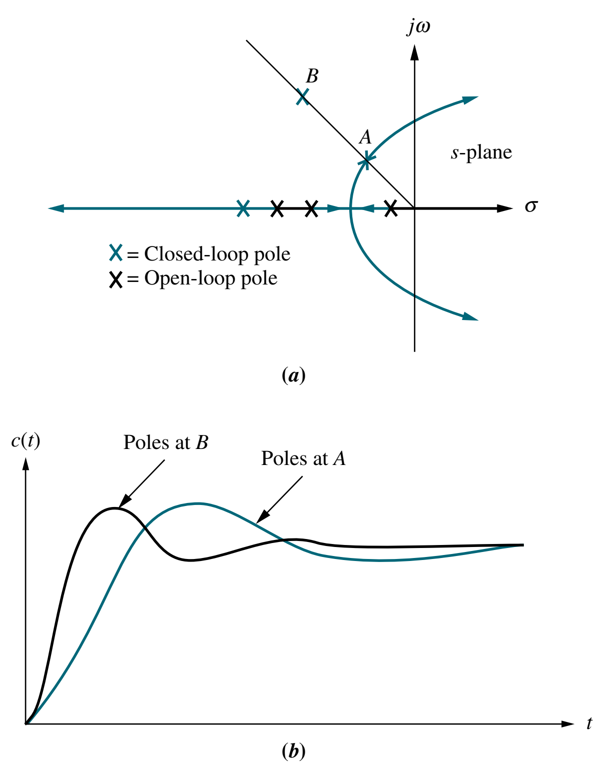


**FIGURE 9.1** **a.** Sample root locus, showing possible design point via gain adjustment ($A$) and desired design point that cannot be met via simple gain adjustment ($B$); **b.** responses from poles at $A$ and $B$

One way to solve our problem is to replace the existing system with a system whose root locus intersects the desired design point, $B$. Unfortunately, this replacement is expensive and counterproductive. Most systems are chosen for characteristics other than transient response. For example, an elevator cage and motor are chosen for speed and power. Components chosen for their transient response may not necessarily meet, for example, power requirements.

Rather than change the existing system, we augment, or *compensate*, the system with *additional* poles and zeros, so that the compensated system has a root locus that goes through the desired pole location for some value of gain. One of the advantages of compensating a system in this way is that additional poles and zeros can be added at the low-power end of the system before the plant. Addition of compensating poles and zeros need not interfere with the power output requirements of the system or present additional load or design problems. The compensating poles and zeros can be generated with a passive or an active network.

A possible disadvantage of compensating a system with additional open-loop poles and zeros is that the system order can increase, with a subsequent effect on the desired response. In Chapters 4 and 8, we discussed the effect of additional closed-loop poles and zeros on the transient response. At the beginning of the design process discussed in this chapter, we determine the proper location of additional *open-loop* poles and zeros to yield the desired second-order *closed-loop* poles. However, we do not know the location of the higher-order *closed-loop* poles until the end of the design. Thus, we should evaluate the transient response through simulation after the design is complete to be sure the requirements have been met.

In Chapter 12, when we discuss state-space design, the disadvantage of finding the location of higher-order closed-loop poles after the design will be eliminated by techniques that allow the designer to specify and design the location of all the closed-loop poles at the beginning of the design process.

One method of compensating for transient response that will be discussed later is to insert a differentiator in the forward path in parallel with the gain. We can visualize the operation of the differentiator with the following example. Assuming a position control with a step input, we note that the error undergoes an initial large change. Differentiating this rapid change yields a large signal that drives the plant. The output from the differentiator is much larger than the output from the pure gain. This large, initial input to the plant produces a faster response. As the error approaches its final value, its derivative approaches zero, and the output from the differentiator becomes negligible compared to the output from the gain.

## Improving Steady-State Error

Compensators are not only used to improve the transient response of a system; they are also used *independently* to improve the steady-state error characteristics. Previously, when the system gain was adjusted to meet the transient response specification, steady-state error performance deteriorated, since both the transient response and the static error constant were related to the gain. The higher the gain, the smaller the steady-state error, but the larger the percent overshoot. On the other hand, reducing gain to reduce overshoot increased the steady-state error. If we use dynamic compensators, compensating networks can be designed that will allow us to meet transient and steady-state error specifications *simultaneously*.[1] We no longer

[1] The word *dynamic* describes compensators with noninstantaneous transient response. The transfer functions of such compensators are functions of the Laplace variable, $s$, rather than pure gain.

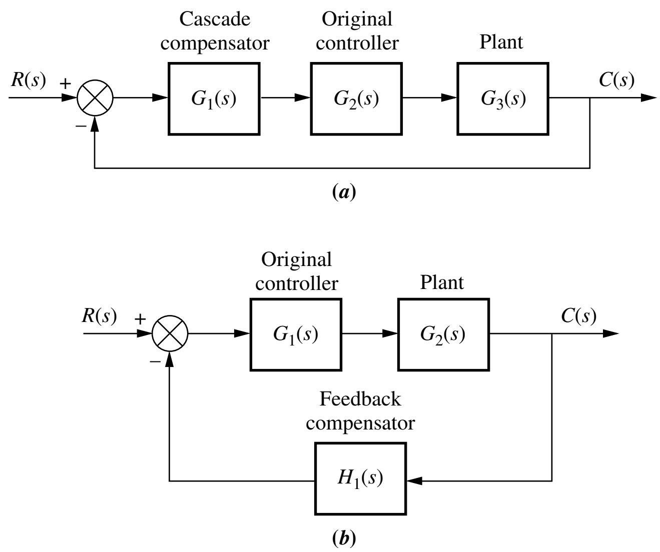


**FIGURE 9.2** Compensation techniques: **a.** cascade; **b.** feedback

need to compromise between transient response and steady-state error, as long as the system operates in its linear range.

In Chapter 7, we learned that steady-state error can be improved by adding an open-loop pole at the origin in the forward path, thus increasing the system type and driving the associated steady-state error to zero. This additional pole at the origin requires an integrator for its realization.

In summary, then, transient response is improved with the addition of differentiation, and steady-state error is improved with the addition of integration in the forward path.

## Configurations

Two configurations of compensation are covered in this chapter: cascade compensation and feedback compensation. These methods are modeled in Figure 9.2. With cascade compensation, the compensating network, $G_1(s)$, is placed at the low-power end of the forward path in cascade with the plant. If feedback compensation is used, the compensator, $H_1(s)$, is placed in the feedback path. Both methods change the open-loop poles and zeros, thereby creating a new root locus that goes through the desired closed-loop pole location.

## Compensators

Compensators that use pure integration for improving steady-state error or pure differentiation for improving transient response are defined as *ideal compensators*. Ideal compensators must be implemented with active networks, which, in the case of electric networks, require the use of active amplifiers and possible additional power sources. An advantage of ideal integral compensators is that steady-state error is reduced to zero. Electromechanical ideal compensators, such as tachometers, are often used to improve transient response, since they can be conveniently interfaced with the plant.

Other design techniques that preclude the use of active devices for compensation can be adopted. These compensators, which can be implemented with passive elements such as resistors and capacitors, do not use pure integration and differentiation and are not ideal compensators. Advantages of passive networks are that they are less expensive and do not require additional power sources for their operation. Their disadvantage is that the steady-state error is not driven to zero in cases where ideal compensators yield zero error.

Thus, the choice between an active or a passive compensator revolves around cost, weight, desired performance, transfer function, and the interface between the compensator and other hardware. In Sections 9.2, 9.3, and 9.4, we first discuss cascade compensator design using ideal compensation and follow with cascade compensation using compensators that are not implemented with pure integration and differentiation.

## 9.2 Improving Steady-State Error via Cascade Compensation

In this section, we discuss two ways to improve the steady-state error of a feedback control system using cascade compensation. One objective of this design is to improve the steady-state error without appreciably affecting the transient response.

The first technique is *ideal integral compensation*, which uses a pure integrator to place an open-loop, forward-path pole at the origin, thus increasing the system type and reducing the error to zero. The second technique does not use pure integration. This compensation technique places the pole near the origin, and although it does not drive the steady-state error to zero, it does yield a measurable reduction in steady-state error.

While the first technique reduces the steady-state error to zero, the compensator must be implemented with active networks, such as amplifiers. The second technique, although it does not reduce the error to zero, does have the advantage that it can be implemented with a less expensive passive network that does not require additional power sources.

The names associated with the compensators come either from the method of implementing the compensator or from the compensator's characteristics. Systems that feed the error forward to the plant are called *proportional control systems*. Systems that feed the integral of the error to the plant are called *integral control systems*. Finally, systems that feed the derivative of the error to the plant are called *derivative control systems*. Thus, in this section we call the ideal integral compensator a *proportional-plus-integral* (*PI*) *controller*, since the implementation, as we will see, consists of feeding the error (proportional) plus the integral of the error forward to the plant. The second technique uses what we call a *lag compensator*. The name of this compensator comes from its frequency response characteristics, which will be discussed in Chapter 11. Thus, we use the name *PI controller* interchangeably with *ideal integral compensator*, and we use the name *lag compensator* when the cascade compensator does not employ pure integration.

### Ideal Integral Compensation (PI)

Steady-state error can be improved by placing an open-loop pole at the origin, because this increases the system type by one. For example, a Type 0 system responding to a step input with a finite error responds with zero error if the system type is increased by one. Active circuits can be used to place poles at the origin. Later in this chapter, we show how to build an integrator with active electronic circuits.

To see how to improve the steady-state error without affecting the transient response, look at Figure 9.3(*a*). Here we have a system operating with a desirable

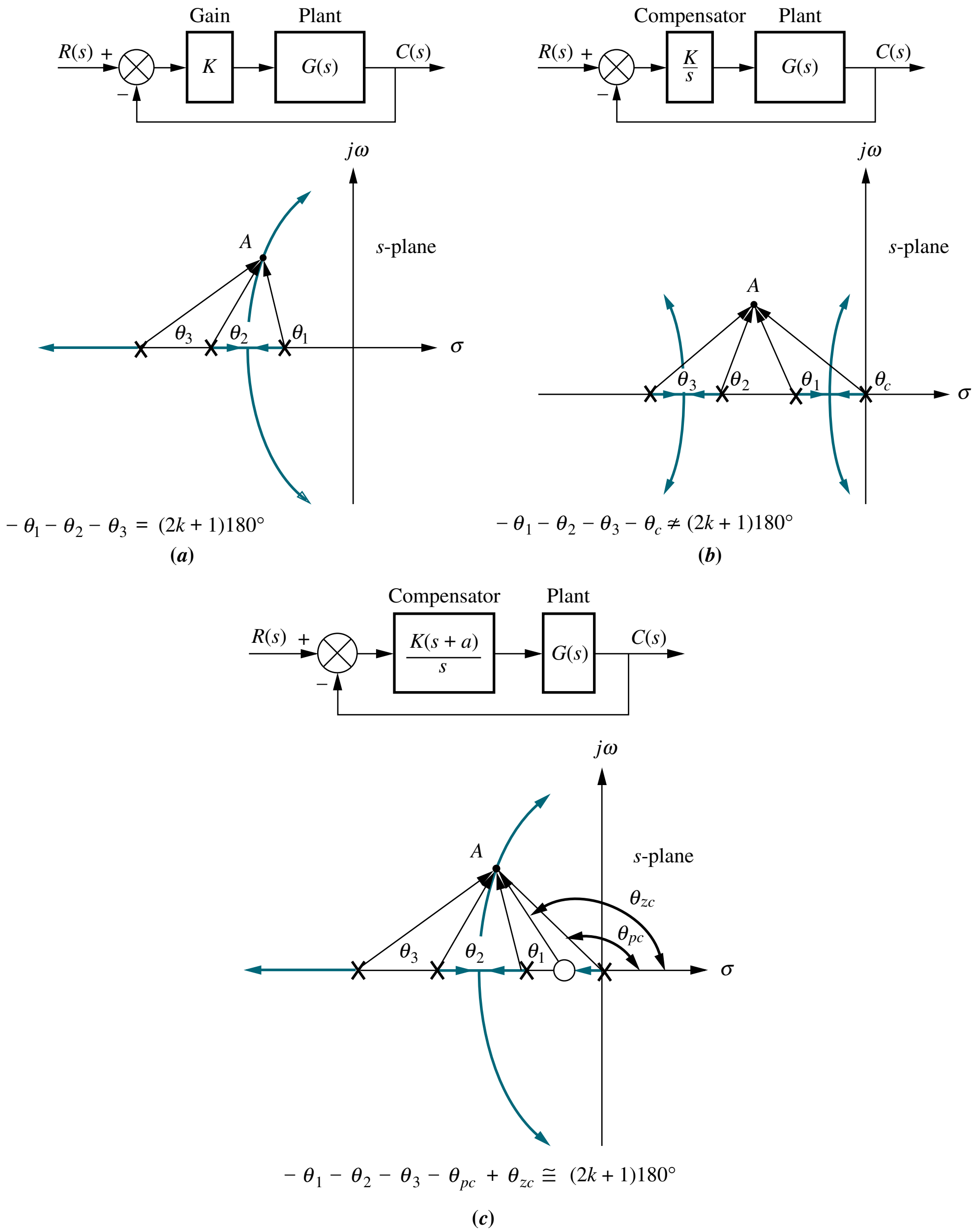


**FIGURE 9.3** Pole at $A$ is **a.** on the root locus without compensator; **b.** not on the root locus with compensator pole added; **c.** approximately on the root locus with compensator pole and zero added

transient response generated by the closed-loop poles at $A$. If we add a pole at the origin to increase the system type, the angular contribution of the open-loop poles at point $A$ is no longer 180°, and the root locus no longer goes through point $A$, as shown in Figure 9.3($b$).

To solve the problem, we also add a zero close to the pole at the origin, as shown in Figure 9.3($c$). Now the angular contribution of the compensator zero and compensator pole cancel out, point $A$ is still on the root locus, and the system type has been increased. Furthermore, the required gain at the dominant pole is about the same as before compensation, since the ratio of lengths from the compensator pole and the compensator zero is approximately unity. Thus, we have improved the steady-state error without appreciably affecting the transient response. A compensator with a pole at the origin and a zero close to the pole is called an *ideal integral compensator*.

In the example that follows, we demonstrate the effect of ideal integral compensation. An open-loop pole will be placed at the origin to increase the system type and drive the steady-state error to zero. An open-loop zero will be placed very close to the open-loop pole at the origin so that the original closed-loop poles on the original root locus still remain at approximately the same points on the compensated root locus.

## Example 9.1

### Effect of an Ideal Integral Compensator

**PROBLEM:** Given the system of Figure 9.4(*a*), operating with a damping ratio of 0.174, show that the addition of the ideal integral compensator shown in Figure 9.4(*b*) reduces the steady-state error to zero for a step input without appreciably affecting transient response. The compensating network is chosen with a pole at the origin to increase the system type and a zero at $-0.1$, close to the compensator pole, so that the angular contribution of the compensator evaluated at the original, dominant, second-order poles is approximately zero. Thus, the original, dominant, second-order closed-loop poles are still approximately on the new root locus.

**SOLUTION:** We first analyze the uncompensated system and determine the location of the dominant, second-order poles. Next we evaluate the uncompensated steady-state error for a unit step input. The root locus for the uncompensated system is shown in Figure 9.5.

A damping ratio of 0.174 is represented by a radial line drawn on the $s$-plane at $100.02°$. Searching along this line with the root locus program discussed in Appendix H at www.wiley.com/college/nise, we find that the dominant poles are $0.694 \pm j3.926$ for a gain, $K$, of 164.6. Now look for the third pole on the root locus beyond $-10$ on the real axis. Using the root locus program and searching for the same gain as that of the dominant pair, $K = 164.6$, we find that the third pole is approximately at $-11.61$. This gain yields $K_p = 8.23$. Hence, the steady-state error is

$$e(\infty) = \frac{1}{1+K_p} = \frac{1}{1+8.23} = 0.108 \tag{9.1}$$

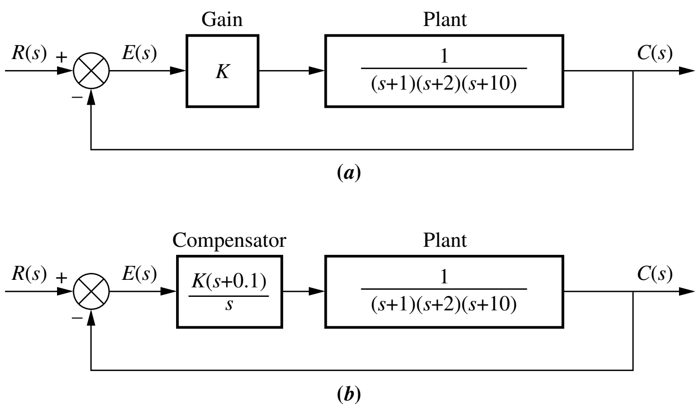


**FIGURE 9.4** Closed-loop system for Example 9.1: **a.** before compensation; **b.** after ideal integral compensation

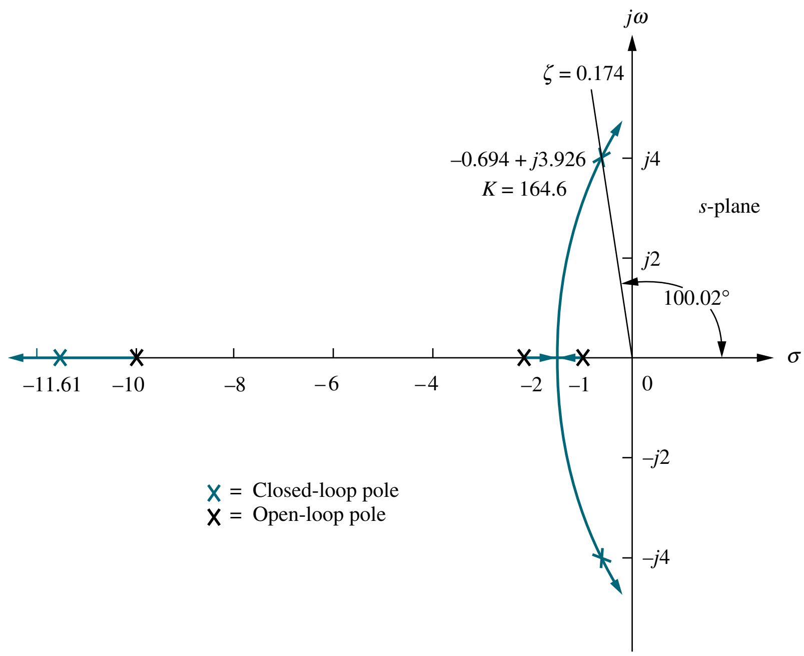


**FIGURE 9.5** Root locus for uncompensated system of Figure 9.4(*a*)

Adding an ideal integral compensator with a zero at $-0.1$, as shown in Figure 9.4(*b*), we obtain the root locus shown in Figure 9.6. The dominant second-order poles, the third pole beyond $-10$, and the gain are approximately the same as for the uncompensated system. Another section of the compensated root locus is between the origin and $-0.1$. Searching this region for the same gain at the dominant pair, $K = 158.2$, the fourth closed-loop pole is found at $-0.0902$, close

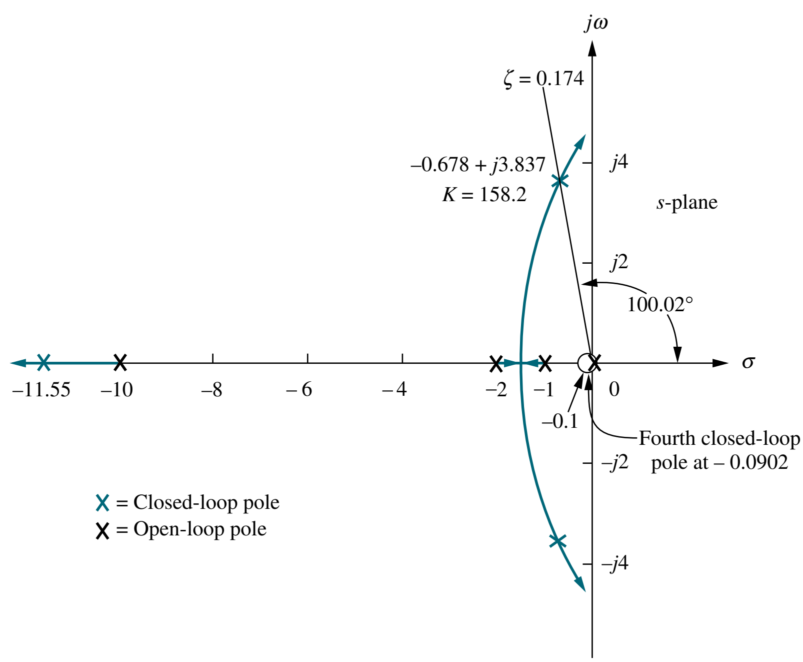


**FIGURE 9.6** Root locus for compensated system of Figure 9.4(*b*)

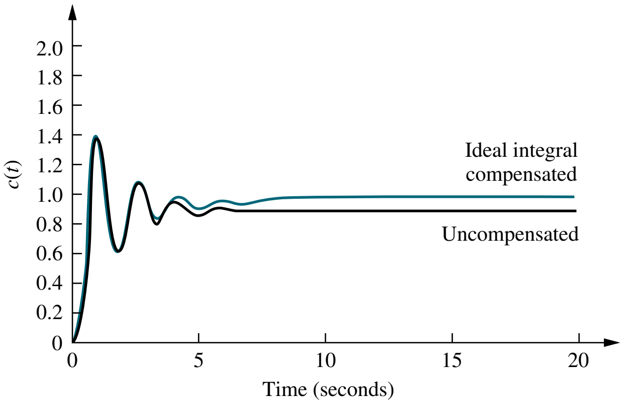


**FIGURE 9.7** Ideal integral compensated system response and the uncompensated system response of Example 9.1

enough to the zero to cause pole-zero cancellation. Thus, the compensated system's closed-loop poles and gain are approximately the same as the uncompensated system's closed-loop poles and gain, which indicates that the transient response of the compensated system is about the same as the uncompensated system. However, the compensated system, with its pole at the origin, is a Type 1 system; unlike the uncompensated system, it will respond to a step input with zero error.

Figure 9.7 compares the uncompensated response with the ideal integral compensated response. The step response of the ideal integral compensated system approaches unity in the steady state, while the uncompensated system approaches 0.892. Thus, the ideal integral compensated system responds with zero steady-state error. The transient response of both the uncompensated and the ideal integral compensated systems is the same up to approximately 3 seconds. After that time the integrator in the compensator, shown in Figure 9.4(*b*), slowly compensates for the error until zero error is finally reached. The simulation shows that it takes 18 seconds for the compensated system to reach to within ±2% of the final value of unity, while the uncompensated system takes about 6 seconds to settle to within ±2% of its final value of 0.892. The compensation at first may appear to yield deterioration in the settling time. However, notice that the compensated system reaches the uncompensated system's final value in about the same time. The remaining time is used to improve the steady-state error over that of the uncompensated system.

A method of implementing an ideal integral compensator is shown in Figure 9.8. The compensating network precedes $G(s)$ and is an ideal integral compensator since

$$G_c(s) = K_1 + \frac{K_2}{s} = \frac{K_1\left(s + \frac{K_2}{K_1}\right)}{s} \tag{9.2}$$

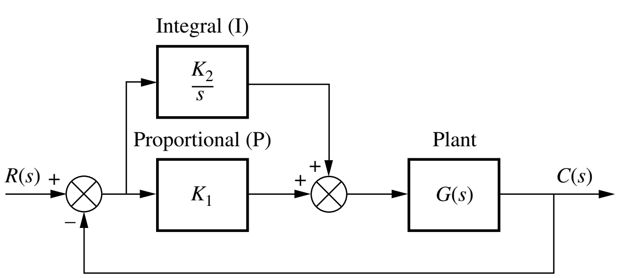


**FIGURE 9.8** PI controller

The value of the zero can be adjusted by varying $K_2/K_1$. In this implementation, the error and the integral of the error are fed forward to the plant, $G(s)$. Since Figure 9.8 has both proportional and integral control, the ideal integral controller, or compensator, is given the alternate name *PI controller*. Later in the chapter we will see how to implement each block, $K_1$ and $K_2/s$.

## Lag Compensation

Ideal integral compensation, with its pole on the origin, requires an active integrator. If we use passive networks, the pole and zero are moved to the left, close to the origin, as shown in Figure 9.9($c$). One may guess that this placement of the pole, although it does not increase the system type, does yield an improvement in the static error constant over an uncompensated system. Without loss of generality, we demonstrate that this improvement is indeed realized for a Type 1 system.

Assume the uncompensated system shown in Figure 9.9($a$). The static error constant, $K_{v_O}$, for the system is

$$K_{v_O} = \frac{K z_1 z_2 \cdots}{p_1 p_2 \cdots} \tag{9.3}$$

Assuming the lag compensator shown in Figure 9.9($b$) and ($c$), the new static error constant is

$$K_{v_N} = \frac{(K z_1 z_2 \cdots)(z_c)}{(p_1 p_2 \cdots)(p_c)} \tag{9.4}$$

What is the effect on the transient response? Figure 9.10 shows the effect on the root locus of adding the lag compensator. The uncompensated system's root locus is shown in Figure 9.10($a$), where point $P$ is assumed to be the dominant pole. If the lag compensator pole and zero are close together, the angular contribution of the

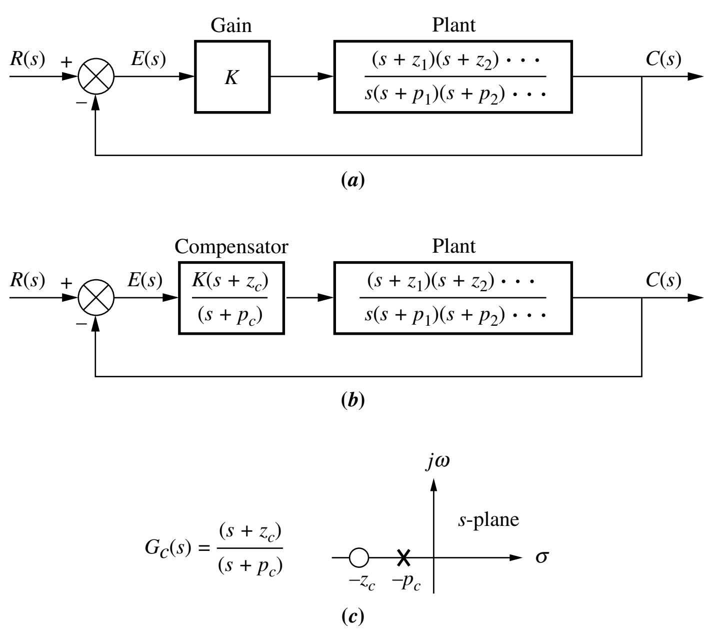


**FIGURE 9.9** **a.** Type 1 uncompensated system; **b.** Type 1 compensated system; **c.** compensator pole-zero plot

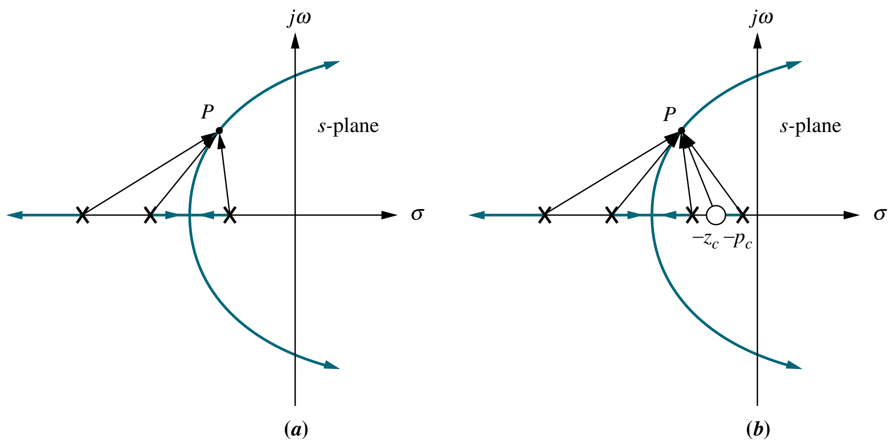


**FIGURE 9.10** Root locus: **a.** before lag compensation; **b.** after lag compensation

compensator to point $P$ is approximately zero degrees. Thus, in Figure 9.10($b$), where the compensator has been added, point $P$ is still at approximately the same location on the compensated root locus.

What is the effect on the required gain, $K$? After inserting the compensator, we find that $K$ is virtually the same for the uncompensated and compensated systems, since the lengths of the vectors drawn from the lag compensator are approximately equal and all other vectors have not changed appreciably.

Now, what improvement can we expect in the steady-state error? Since we established that the gain, $K$, is about the same for the uncompensated and compensated systems, we can substitute Eq. (9.3) into (9.4) and obtain

$$K_{v_N} = K_{v_O} \frac{z_c}{p_c} > K_{v_O} \tag{9.5}$$

Equation (9.5) shows that the improvement in the compensated system's $K_v$ over the uncompensated system's $K_v$ is equal to the ratio of the magnitude of the compensator zero to the compensator pole. In order to keep the transient response unchanged, we know the compensator pole and zero must be close to each other. The only way the ratio of $z_c$ to $p_c$ can be large in order to yield an appreciable improvement in steady-state error and simultaneously have the compensator's pole and zero close to each other to minimize the angular contribution is to place the compensator's pole-zero pair close to the origin. For example, the ratio of $z_c$ to $p_c$ can be equal to 10 if the pole is at $-0.001$ and the zero is at $-0.01$. Thus, the ratio is 10, yet the pole and zero are very close, and the angular contribution of the compensator is small.

In conclusion, although the ideal compensator drives the steady-state error to zero, a lag compensator with a pole that is not at the origin will improve the static error constant by a factor equal to $z_c/p_c$. There also will be a minimal effect upon the transient response if the pole-zero pair of the compensator is placed close to the origin. Later in the chapter we show circuit configurations for the lag compensator. These circuit configurations can be obtained with passive networks and thus do not require the active amplifiers and possible additional power supplies that are required by the ideal integral (PI) compensator. In the following example we design a lag compensator to yield a specified improvement in steady-state error.

## Example 9.2

### Lag Compensator Design

**PROBLEM:** Compensate the system of Figure 9.4($a$), whose root locus is shown in Figure 9.5, to improve the steady-state error by a factor of 10 if the system is operating with a damping ratio of 0.174.

**SOLUTION:** The uncompensated system error from Example 9.1 was 0.108 with $K_p = 8.23$. A tenfold improvement means a steady-state error of

$$e(\infty) = \frac{0.108}{10} = 0.0108 \tag{9.6}$$

Since

$$e(\infty) = \frac{1}{1 + K_p} = 0.0108 \tag{9.7}$$

rearranging and solving for the required $K_p$ yields

$$K_p = \frac{1 - e(\infty)}{e(\infty)} = \frac{1 - 0.0108}{0.0108} = 91.59 \tag{9.8}$$

The improvement in $K_p$ from the uncompensated system to the compensated system is the required ratio of the compensator zero to the compensator pole, or

$$\frac{z_c}{p_c} = \frac{K_{p_N}}{K_{p_O}} = \frac{91.59}{8.23} = 11.13 \tag{9.9}$$

Arbitrarily selecting

$$p_c = 0.01 \tag{9.10}$$

we use Eq. (9.9) and find

$$z_c = 11.13 p_c \approx 0.111 \tag{9.11}$$

Let us now compare the compensated system, shown in Figure 9.11, with the uncompensated system. First sketch the root locus of the compensated system, as shown in Figure 9.12. Next search along the $\zeta = 0.174$ line for a multiple of 180° and find that the second-order dominant poles are at $-0.678 \pm j3.836$ with a gain, $K$, of 158.1. The third and fourth closed-loop poles are at $-11.55$ and $-0.101$, respectively, and are found by searching the real axis for a gain equal to that of the dominant poles. All transient and steady-state results for both the uncompensated and the compensated systems are shown in Table 9.1.

The fourth pole of the compensated system cancels its zero. This leaves the remaining three closed-loop poles of the compensated system very close in value to the three closed-loop poles of the uncompensated system. Hence, the transient

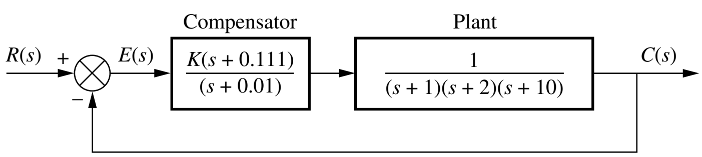


**FIGURE 9.11** Compensated system for Example 9.2

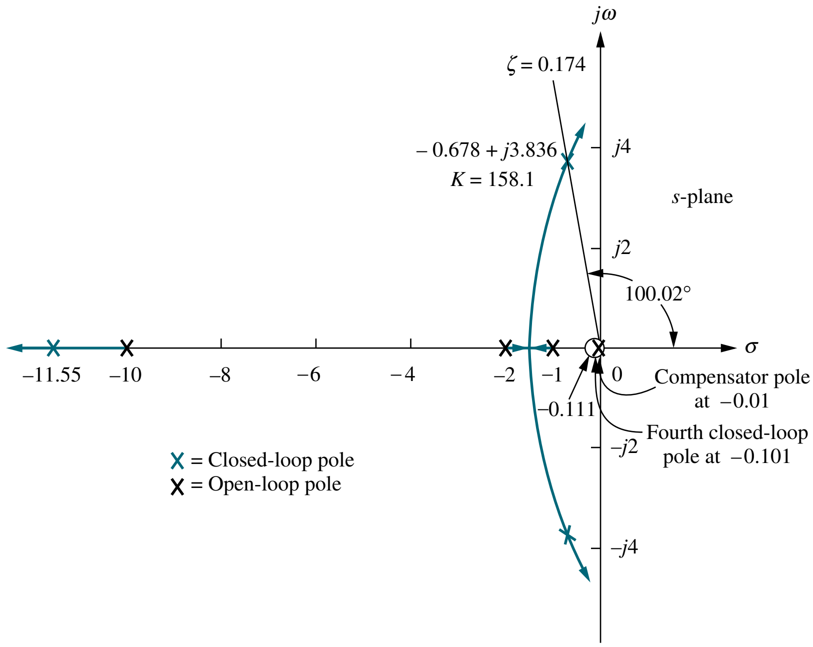


**FIGURE 9.12** Root locus for compensated system of Figure 9.11

response of both systems is approximately the same, as is the system gain, but notice that the steady-state error of the compensated system is 1/9.818 that of the uncompensated system and is close to the design specification of a tenfold improvement.

Figure 9.13 shows the effect of the lag compensator in the time domain. Even though the transient responses of the uncompensated and lag-compensated systems are the same, the lag-compensated system exhibits less steady-state error by approaching unity more closely than the uncompensated system.

We now examine another design possibility for the lag compensator and compare the response to Figure 9.13. Let us assume a lag compensator whose pole and zero are 10 times as close to the origin as in the previous design. The results are compared in Figure 9.14. Even though both responses will eventually reach approximately the same steady-state value, the lag compensator previously designed, $G_c(s) = (s + 0.111)/(s + 0.01)$, approaches the final value faster than the proposed lag compensator, $G_c(s) = (s + 0.0111)/(s + 0.001)$. We can explain this phenomenon as follows. From Table 9.1, the previously designed lag compensator

**TryIt 9.1**

Use the following MATLAB and Control System Toolbox statements to reproduce Figure 9.13.

```
Gu=zpk([],...
 [-1 -2 -10],164.6);
Gc=zpk([-0.111],...
 [-0.01],1);
Gce=Gu*Gc;
Tu=feedback(Gu,1);
Tc=feedback(Gce,1);
step(Tu)
hold
step(Tc)
```

**TABLE 9.1** Predicted characteristics of uncompensated and lag-compensated systems for Example 9.2

| Parameter | Uncompensated | Lag-compensated |
|---|---|---|
| Plant and compensator | $\dfrac{K}{(s+1)(s+2)(s+10)}$ | $\dfrac{K(s+0.111)}{(s+1)(s+2)(s+10)(s+0.01)}$ |
| $K$ | 164.6 | 158.1 |
| $K_p$ | 8.23 | 87.75 |
| $e(\infty)$ | 0.108 | 0.011 |
| Dominant second-order poles | $-0.694 \pm j3.926$ | $-0.678 \pm j3.836$ |
| Third pole | $-11.61$ | $-11.55$ |
| Fourth pole | None | $-0.101$ |
| Zero | None | $-0.111$ |

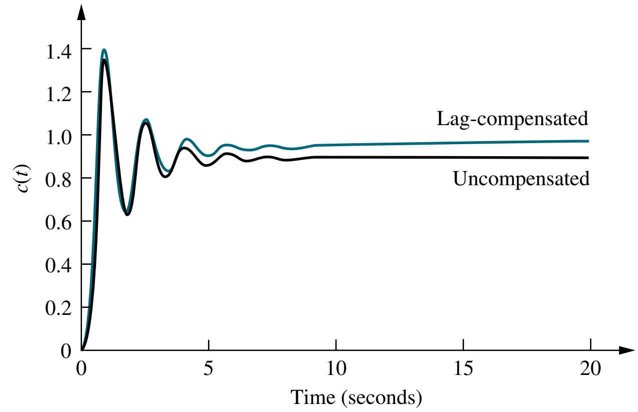


**FIGURE 9.13** Step responses of uncompensated and lag-compensated systems for Example 9.2

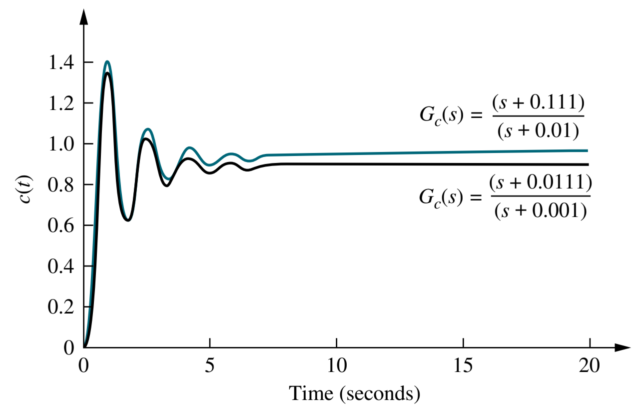


**FIGURE 9.14** Step responses of the system for Example 9.2 using different lag compensators

has a fourth closed-loop pole at $-0.101$. Using the same analysis for the new lag compensator with its open-loop pole 10 times as close to the imaginary axis, we find its fourth closed-loop pole at $-0.01$. Thus, the new lag compensator has a closed-loop pole closer to the imaginary axis than the original lag compensator. This pole at $-0.01$ will produce a longer transient response than the original pole at $-0.101$, and the steady-state value will not be reached as quickly.

## Skill-Assessment Exercise 9.1

WileyPLUS
WPCS
Control Solutions

**PROBLEM:** A unity feedback system with the forward transfer function

$$G(s) = \frac{K}{s(s+7)}$$

is operating with a closed-loop step response that has 15% overshoot. Do the following:

**a.** Evaluate the steady-state error for a unit ramp input.

**b.** Design a lag compensator to improve the steady-state error by a factor of 20.

**c.** Evaluate the steady-state error for a unit ramp input to your compensated system.

**d.** Evaluate how much improvement in steady-state error was realized.

**ANSWERS:**

**a.** $e_{\mathrm{ramp}}(\infty) = 0.1527$

**b.** $G_{\mathrm{lag}}(s) = \dfrac{s + 0.2}{s + 0.01}$

**c.** $e_{\mathrm{ramp}}(\infty) = 0.0078$

**d.** 19.58 times improvement

The complete solution is at www.wiley.com/college/nise.

## 9.3 Improving Transient Response via Cascade Compensation

Since we have solved the problem of improving the steady-state error without affecting the transient response, let us now improve the transient response itself. In this section, we discuss two ways to improve the transient response of a feedback control system by using cascade compensation. Typically, the objective is to design a response that has a desirable percent overshoot and a shorter settling time than the uncompensated system.

The first technique we will discuss is *ideal derivative compensation*. With ideal derivative compensation, a pure differentiator is added to the forward path of the feedback control system. We will see that the result of adding differentiation is the addition of a zero to the forward-path transfer function. This type of compensation requires an active network for its realization. Further, differentiation is a noisy process; although the level of the noise is low, the frequency of the noise is high compared to the signal. Thus, differentiating high-frequency noise yields a large, unwanted signal.

The second technique does not use pure differentiation. Instead, it approximates differentiation with a passive network by adding a zero and a more distant pole to the forward-path transfer function. The zero approximates pure differentiation as described previously.

As with compensation to improve steady-state error, we introduce names associated with the implementation of the compensators. We call an ideal derivative compensator a *proportional-plus-derivative* (*PD*) *controller*, since the implementation, as we will see, consists of feeding the error (proportional) plus the derivative of the error forward to the plant. The second technique uses a passive network called a *lead compensator*. As with the lag compensator, the name comes from its frequency response, which is discussed in Chapter 11. Thus, we use the name *PD controller* interchangeably with *ideal derivative compensator*, and we use the name *lead compensator* when the cascade compensator does not employ pure differentiation.