Chapter Eight Root Locus Control Design

8.3 Common Dynamic Controllers

Several common dynamic controllers appear very often in practice. They are known as PD, PI, PID, phase-lag, phase-lead, and phase-lag-lead controllers. In this section we introduce their structures and indicate their main properties. In the follow-up sections procedures for designing these controllers by using the root locus technique such that the given systems have the desired specifications are presented. In the most cases these controllers are placed in the forward path at the front of the plant (system) as presented in Figure 8.1.

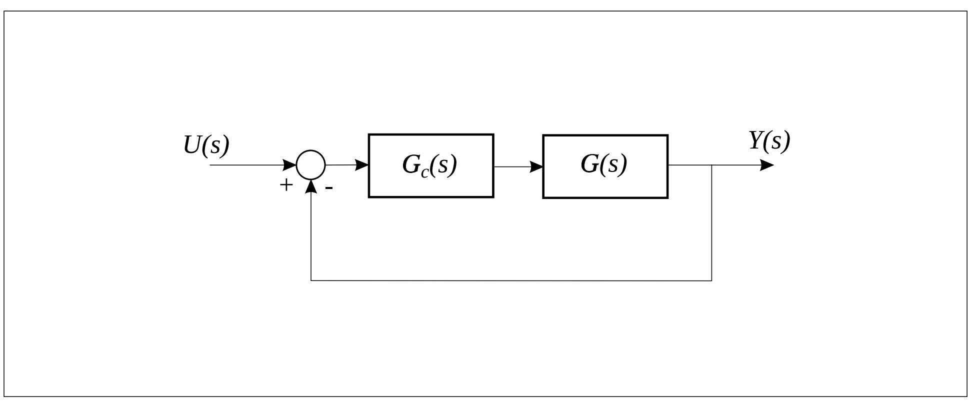

Figure 8.1: A common controller-plant configuration

8.3.1 PD Controller

PD stands for a proportional and derivative controller. The output signal of this controller is equal to the sum of two signals: the signal obtained by multiplying the input signal by a constant gain K_p and the signal obtained by differentiating and multiplying the input signal by K_d, i.e. its transfer function is given by

$$G_c(s) = K_p + K_d s$$

This controller is used to improve the system transient response.

8.3.2 PI Controller

Similarly to the PD controller, the PI controller produces as its output a weighted sum of the input signal and its integral. Its transfer function is

$$G_c(s) = K_p + K_i \frac{1}{s} = \frac{K_p s + K_i}{s}$$

In practical applications the PI controller zero is placed very close to its pole located at the origin so that the angular contribution of this "dipole" to the root locus is almost zero. *A PI controller is used to improve the system response steady state errors* since it increases the control system type by one.

8.3.3 PID Controller

The PID controller is a combination of PD and PI controllers; hence its transfer function is given by

$$G_c(s) = K_p + K_d s + K_i \frac{1}{s} = \frac{K_i + K_p s + K_d s^2}{s}$$

The PID controller can be used to improve both the system transient response and steady state errors. This controller is very popular for industrial applications.

8.3.4 Phase-Lag Controller

The phase-lag controller belongs to the same class as the PI controller. The phase-lag controller can be regarded as a generalization of the PI controller. It introduces a negative phase into the feedback loop, which justifies its name. It has a zero and pole with the pole being closer to the imaginary axis, that is

$$G_c(s) = \left(\frac{p_1}{z_1}\right)\frac{s+z_1}{s+p_1}, \qquad z_1 > p_1 > 0$$

$$\arg G_c(s) = \arg(s+z_1) - \arg(s+p_1) = \theta_{z_1} - \theta_{p_1} < 0$$

where p_1/z_1 is known as the lag ratio. The corresponding angles θ_{z_1} and θ_{p_1} are given in Figure 8.2a. *The phase-lag controller is used to improve steady state errors.*

8.3.5 Phase-Lead Controller

The phase-lead controller is designed such that its phase contribution to the feedback loop is positive. It is represented by

$$G_c(s) = \frac{s+z_2}{s+p_2}, \qquad p_2 > z_2 > 0$$

$$G_c(s) = \arg(s+z_2) - \arg(s+p_2) = \theta_{z_2} - \theta_{p_2} > 0$$

where θ_{z_2} and θ_{p_2} are given in Figure 8.2b. This controller introduces a positive phase shift in the loop (phase lead). *It is used to improve the system response transient behavior.*

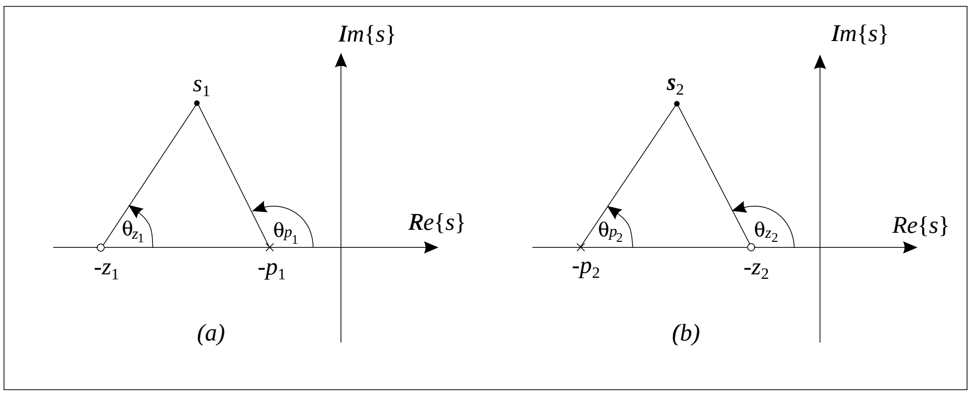

Figure 8.2: Poles and zeros of phase-lag (a) and phase-lead (b) controllers

8.3.6 Phase-Lag-Lead Controller

The phase-lag-lead controller is obtained as a combination of phase-lead and phase-lag controllers. Its transfer function is given by

$$G_c(s) = \frac{(s+z_1)(s+z_2)}{(s+p_1)(s+p_2)}, \quad p_2 > z_2 > z_1 > p_1 > 0, \quad z_1 z_2 = p_1 p_2$$

It has features of both phase-lag and phase-lead controllers, i.e. *it can be used to improve simultaneously both the system transient response and steady state errors*. However, it is harder to design phase-lag-lead controllers than either phase-lag or phase-lead controllers.

8.5 Compensator Design by the Root Locus Method

Sometimes one is able to improve control system specifications by changing the static gain K only. It can be observed that *as* K *increases, the steady state errors decrease (assuming system's asymptotic stability), but the maximum percent overshoot increases*. However, using large values for K may damage system stability. Even more, in most cases the desired operating points for the system dominant poles, which satisfy the transient response requirements, do not lie on the original root locus. Thus, in order to solve the transient response and steady state errors improvement problem, one has to design dynamic controllers, considered in Section 8.3, and put them in series with the plant (system) to be controlled (see Figure 8.1).

In the following we present dynamic controller design techniques in three categories: improvement of steady state errors (PI and phase-lag controllers), improvement of system transient response (PD and phase-lead controllers), and improvement of both steady state errors and transient response (PID and phase-lag-lead controllers). Note that transient response specifications are obtained under the assumption that a given system has a pair of dominant complex conjugate closed-loop poles; hence this assumption has to be checked after a controller is added to the system. This can be easily done using the root locus technique.

8.5.1 Improvement of Steady State Errors

It has been seen in Chapter 6 that the steady state errors can be improved by increasing the type of feedback control system, in other words, by adding a pole at the origin to the open-loop system transfer function. The simplest way to achieve this goal is to add in series with the system a PI controller, i.e. to get

$$G_c(s)G(s) = \frac{K_p s + K_i}{s} G(s)$$

Since this controller also introduces a zero at $-K_i/K_p$, *the zero should be placed as close as possible to the pole.* In that case the pole at $p = 0$ and the zero at $z \approx p$ act as a dipole, and so their mutual contribution to the root locus is almost negligible. Since the root locus is practically unchanged, the system transient response remains the same and the effect due to the PI controller is to increase the type of the control system by one, which produces improved steady state errors. The effect of a dipole on the system response is studied in the next example.

Example 8.4: Consider the open-loop transfer functions

$$G_1(s) = \frac{(s+2)}{(s+1)(s+3)}$$

and

$$G_2(s) = \frac{(s+2)(s+5)}{(s+1)(s+3)(s+5.1)}$$

Note that the second transfer function has a dipole with a stable pole at -5.1. The corresponding step responses are given in Figure 8.4. It can be seen from this figure that the system with a stable dipole and the system without a stable dipole have almost identical responses. These responses have been obtained by the following sequence of MATLAB instructions.

```
num1=[1 2]; den1=[1 4 3]; num2=[1 7 10];
d1=[1 1]; d2=[1 3]; d3=[1 5.1]; d12=conv(d1,d2);
den2=conv(d12,d3);
```

```
[cnum1,cden1]=feedback(num1,den1,1,1,-1);
[cnum2,cden2]=feedback(num2,den2,1,1,-1);
t=0:0.1:2;
step(cnum1,cden1,t)
step(cnum2,cden2,t)
```

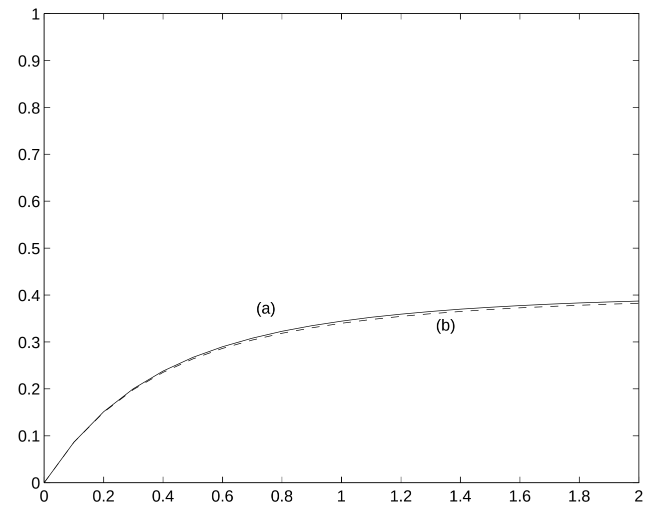

Figure 8.4: Step responses of a system without a stable dipole (a) and with a stable dipole (b)

It is important to point out that in the case of an *unstable dipole* the effect of a dipole is completely different. Consider, for example, the open-loop transfer function given by

$$G_3(s) = \frac{(s+2)(s-5)}{(s+1)(s+3)(s-5.1)}$$

Its step response is presented in Figure 8.5b and compared with the corresponding step response after a dipole is eliminated (Figure 8.5a). In fact, the system without a dipole is stable and the system with a dipole is unstable; hence their responses are drastically different. Thus, we can conclude that *it is not correct to cancel an unstable dipole* since it has a big impact on the system response.

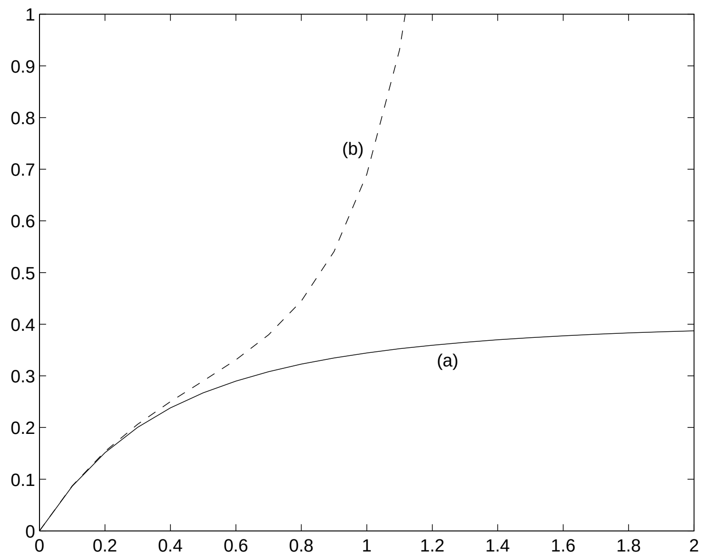

Figure 8.5: Step responses of a system without an unstable dipole (a) and with an unstable dipole (b)

Both the PI and phase-lag controller use this "*stable dipole effect*". They do not change the system transient response, but they do have an important impact on the steady state errors.

PI Controller Design

As we have already indicated, the PI controller represents a stable dipole with a pole located at the origin and a stable zero placed near the pole. Its impact on the transient response is negligible since it introduces neither significant phase shift nor gain change. Thus, the transient response parameters with the PI controller are almost the same as those for the original system, but the steady state errors are drastically improved due to the fact that the feedback control system type is increased by one.

The PI controller is represented, in general, by

$$G_c(s) = K_p \frac{s + \frac{K_i}{K_p}}{s}, \qquad K_i \ll K_p$$

where K_p represents its static gain and K_i/K_p is a stable zero near the origin. Very often it is implemented as

$$G_c(s) = \frac{s + z_c}{s}$$

This implementation is sufficient to justify its main purpose. The design algorithm for this controller is extremely simple.

Design Algorithm 8.1:

1. Set the PI controller's pole at the origin and locate its zero arbitrarily close to the pole, say $z_c = 0.1$ or $z_c = 0.01$.
2. If necessary, adjust for the static loop gain to compensate for the case when K_p is different from one. Hint: Use $K_p = 1$, and avoid gain adjustment problem.

Comment: Note that while drawing the root locus of a system with a PI controller (compensator), the stable open-loop zero of the compensator will attract the compensator's pole located at the origin as the static gain increases from 0 to $+\infty$ so that there is no danger that the closed-loop system may become unstable due to addition of a PI compensator (controller).

The following example demonstrates the use of a PI controller in order to reduce the steady state errors.

Example 8.5: Consider the following open-loop transfer function

$$G(s) = \frac{K(s+6)}{(s+10)(s^2+2s+2)}$$

Let the choice of the static gain $K = 10$ produce a pair of dominant poles on the root locus, which guarantees the desired transient specifications. The corresponding position constant and the steady state unit step error are given by

$$K_p = \frac{10 \times 6}{10 \times 2} = 3 \;\Rightarrow\; e_{ss} = \frac{1}{1+K_p} = 0.25$$

Using a PI controller with the zero at -0.1 ($z_c = 0.1$), we obtain the improved values as $K_p = \infty$ and $e_{ss} = 0$. The step responses of the original system and the compensated system, now given by

$$G_c(s)G(s) = \frac{10(s+0.1)(s+6)}{s(s+10)(s^2+2s+2)}$$

are presented in Figure 8.6.

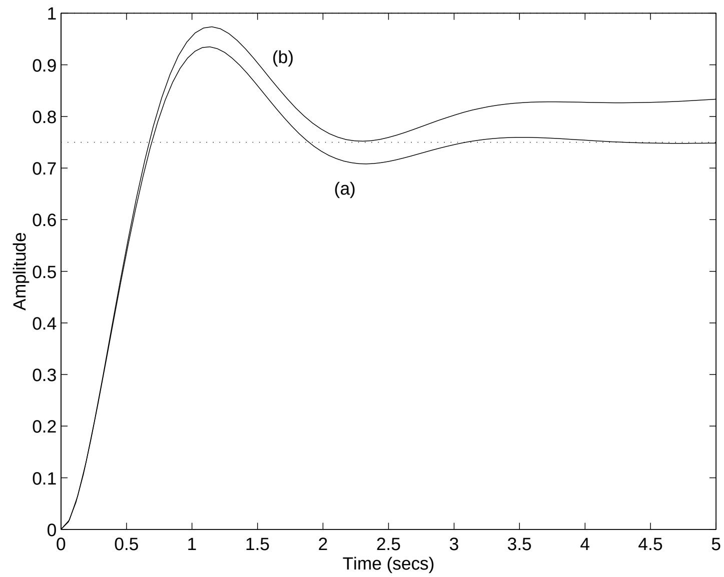

Figure 8.6: Step responses of the original (a) and compensated (b) systems for Example 8.5

The closed-loop poles of the original system are given by

$$\lambda_1 = -9.5216, \qquad \lambda_{2,3} = -1.2392 \pm j2.6204$$

For the compensated system they are

$$\lambda_{1c} = -9.5265, \qquad \lambda_{2c,3c} = -1.1986 \pm j2.6109$$

Having obtained the closed-loop system poles, it is easy to check that the dominant system poles are preserved for the compensated system and that the damping ratio and natural frequency are only slightly changed. Using information about the dominant system poles and relationships obtained from

Figure 6.2, we get

$$\zeta\omega_n = 1.2392, \quad \omega_n^2 = (1.2392)^2 + (2.6204)^2$$
$$\Rightarrow \ \omega_n^2 = 2.9019, \ \zeta = 0.4270$$

and

$$\zeta_c\omega_{nc} = 1.1986, \quad \omega_{nc}^2 = (1.1986)^2 + (2.6109)^2$$
$$\Rightarrow \ \omega_{nc}^2 = 2.8901, \quad \zeta_c = 0.4147$$

In Figure 8.7 we draw the step response of the compensated system over a long period of time in order to show that the steady state error of this system is theoretically and practically equal to zero.

Figures 8.6 and 8.7 are obtained by using the same MATLAB functions as those used in Example 8.4.

The root loci of the original and compensated systems are presented in Figures 8.8 and 8.9. It can be seen from these figures that the root loci are almost identical, with the exception of a tiny dipole branch near the origin.

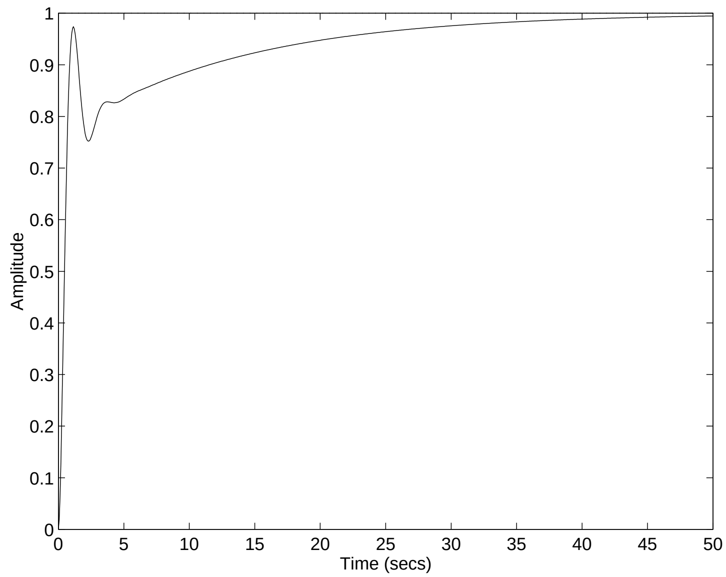

Figure 8.7: Step response of the compensated system for Example 8.5

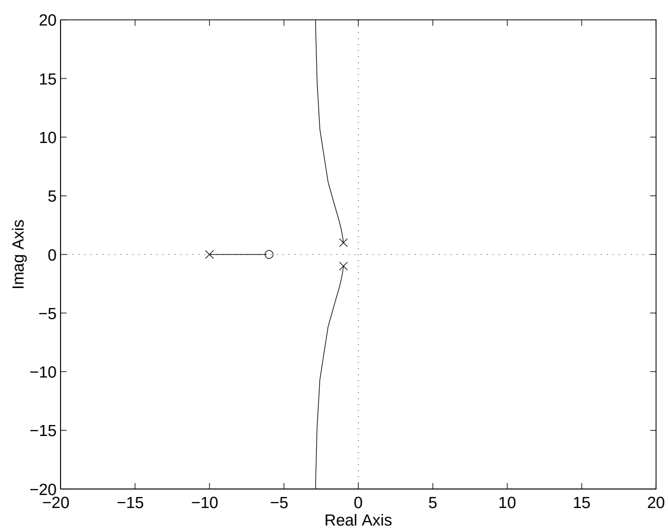

Figure 8.8: Root locus of the original system for Example 8.5

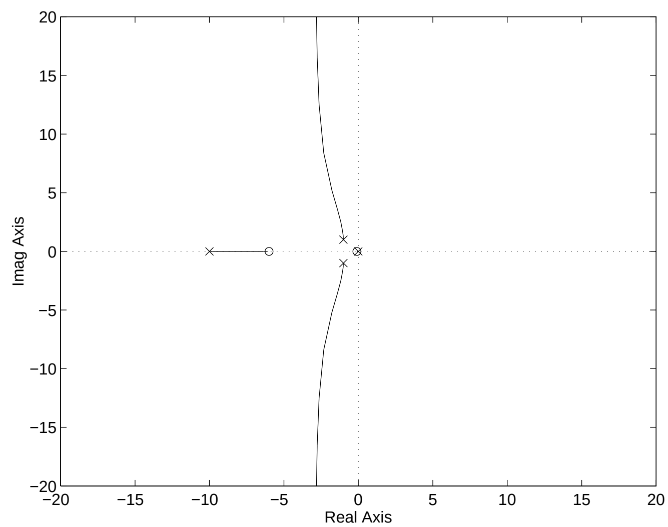

Figure 8.9: Root locus of the compensated system for Example 8.5

Phase-Lag Controller Design

The phase-lag controller, in the context of root locus design methodology, is also implemented as a dipole that has no significant influence on the root locus, and thus on the transient response, but increases the steady state constants and reduces the corresponding steady state errors. Since it is implemented as a dipole, its zero and pole have to be placed very close to each other.

The lag controller's impact on the steady state errors can be obtained from the expressions for the corresponding steady state constants. Namely, we know that

$$K_p = \lim_{s \to 0} \{H(s)G(s)\}, \quad K_v = \lim_{s \to 0} \{sH(s)G(s)\}$$
$$K_a = \lim_{s \to 0} \left\{s^2 H(s)G(s)\right\}$$

and

$$e_{ss_{step}} = \frac{1}{1+K_p}, \quad e_{ss_{ramp}} = \frac{1}{K_v}, \quad e_{ss_{parabolic}} = \frac{2}{K_a}$$

For control systems of type zero, one, and two, respectively, the constants K_p, K_v, and K_a are all given by the same expression, that is

$$K_l = K\frac{z_1 z_2 \cdots}{p_1 p_2 \cdots}, \qquad l = p, v, a$$

Consider, first, a phase-lag compensator of the form

$$G_c(s) = \frac{s+z_c}{s+p_c}, \qquad z_c > p_c > 0$$

If we put this controller in series with the system, the corresponding steady state constants of the compensated system will be given by

$$K_{lc} = K\frac{z_1 z_2 \cdots}{p_1 p_2 \cdots}\frac{z_c}{p_c} = K_l\frac{z_c}{p_c}, \qquad lc, l = p, v, a$$

In order to increase these constants and reduce the steady state errors, the ratio of z_c/p_c should be as large as possible. Since at the same time z_c must be close to p_c (they form a dipole), a large value for the ratio z_c/p_c can be achieved if both of them are placed close to zero. For example, the choice of $z_c = 0.1$ and $p_c = 0.01$ increases the constants $K_l, l = p, v, a,$ ten times and reduces the corresponding steady state errors ten times.

Now consider a phase-lag controller defined by (8.18), that is

$$G_c(s) = \left(\frac{p_c}{z_c}\right)\frac{s+z_c}{s+p_c}, \qquad z_c > p_c > 0$$

This controller will change the value of the static gain K by a factor of p_c/z_c, which will produce a movement of the desired operating point along the root locus in the direction of smaller static gains. Thus, the plant static gain has to be adjusted to a higher value in order to preserve the same operating point. The consequence of using this phase-lag controller is that *the same (desired) operating point is obtained with higher static gain*. We already know that by increasing the static gain, the steady state errors are reduced. In this case, the static gain adjustment has to be done by choosing a new static gain $\widetilde{K} = Kz_c/p_c$. Note that the effects of both phase-lag controllers are exactly the same, since the gain adjustment in the case of controller (8.18) in fact cancels its lag ratio p_c/z_c.

The following simple algorithm is used for phase-lag controller design.

Design Algorithm 8.2:

1. Choose a point that has the desired transient specifications on the root locus branch with dominant system poles. Read from the root locus the value for the static gain K at the chosen point, and determine the corresponding steady state errors.
2. Set both the phase-lag controller's pole and zero near the origin with the ratio z_c/p_c obtained such that the desired steady state error requirement is satisfied.
3. In the case of controller (8.18), adjust for the static loop gain, i.e. take a new static gain as $\widetilde{K} = Kz_c/p_c$.

Example 8.6: The steady state errors of the system considered in Example 8.5 can be improved by using a phase-lag controller of the form

$$G_c(s) = \frac{s + 0.1}{s + 0.01}$$

Since $z_c/p_c = 10$, the position constant is increased ten times, that is

$$K_{pc} = K_p \frac{z_c}{p_c} = 3 \times 10 = 30$$

so that the steady state error due to a unit step input is reduced to

$$e_{ss_{step}} = \frac{1}{1 + K_{pc}} = \frac{1}{31} = 0.03226$$

It can be easily checked that the transient response is almost unchanged; in fact, the dominant system poles with this phase-lag compensator are $-1.2026 \pm j2.6119$, which is very close to the dominant poles of the original system (see Example 8.5).

Example 8.7: Consider the following open-loop transfer function

$$G(s)H(s) = \frac{K(s + 15)}{s(s + 20)(s^2 + 4s + 8)}$$

Let the choice of the static gain $K = 20$ produce a pair of dominant poles on the root locus that guarantees the desired transient specifications. The system closed-loop poles for $K = 20$ are given by

$$\lambda_{1,2} = -0.5327 \pm j2.2024, \quad \lambda_3 = -2.9194, \quad \lambda_4 = -20.0153$$

so that for this value of the static gain K the dominant poles exist, i.e. the absolute value of the real part of the dominant poles (0.5327) is about six times smaller than the absolute value of the real part of the next pole (2.9194), which is in practice sufficient to guarantee poles' dominance. Since we have a type one feedback control system, the steady state error due to a unit step is zero. The velocity constant and the steady state unit ramp error are obtained as

$$K_v = \frac{20 \times 15}{20 \times 8} = \frac{15}{8} \quad \Rightarrow \quad e_{ss_{ramp}} = \frac{1}{K_v} = 0.53$$

Using the phase-lag controller with a zero at -0.1 ($z_c = 0.1$) and a pole at -0.01 ($p_c = 0.01$), we get

$$K_{vc} = K_v \frac{z_c}{p_c} = \frac{150}{8} \quad \Rightarrow \quad e_{ssc_{ramp}} = 0.053$$

It can be easily shown by using MATLAB that the ramp responses of the original and the compensated systems are very close to each other. The same holds for the root loci. Note that even smaller steady state errors can be obtained if we increase the ratio z_c/p_c, e.g. to $z_c/p_c = 100$.

8.5.2 Improvement of Transient Response

The transient response can be improved by using either the PD or phase-lead controllers.

PD Controller Design

The PD controller is represented by

$$G_c(s) = s + z_c, \qquad z_c > 0$$

which indicates that the compensated system open-loop transfer function will have one additional zero. The effect of this zero is to introduce a positive phase shift. The phase shift and position of the compensator's zero can be determined by using simple geometry. That is, for the chosen dominant complex conjugate poles that produce the desired transient response we apply the root locus angle rule. This rule basically says that for a chosen point, s_d, on the root locus the difference of the sum of the angles between the point s_d and the open-loop zeros, and the sum of the angles between the point s_d and the open-loop poles must be 180°. Applying the root locus angle rule to the compensated system, we get

$$\measuredangle G_c(s_d)G(s_d) = \measuredangle(s_d + z_c) + \sum_{i=1}^{m} \measuredangle(s_d + z_i) - \sum_{i=1}^{n} \measuredangle(s_d + p_i) = 180^\circ$$

which implies

$$\measuredangle(s_d + z_c) = 180^\circ - \sum_{i=1}^{m} \measuredangle(s_d + z_i) + \sum_{i=1}^{n} \measuredangle(s_d + p_i) = \alpha_c$$

From the obtained angle $\measuredangle(s_d + z_c)$ the location of the compensator's zero is obtained by playing simple geometry as demonstrated in Figure 8.10. Using this figure it can be easily shown that the value of z_c is given by

$$z_c = \frac{\omega_n}{\tan \alpha_c}\left(\zeta \tan \alpha_c + \sqrt{1-\zeta^2}\right)$$

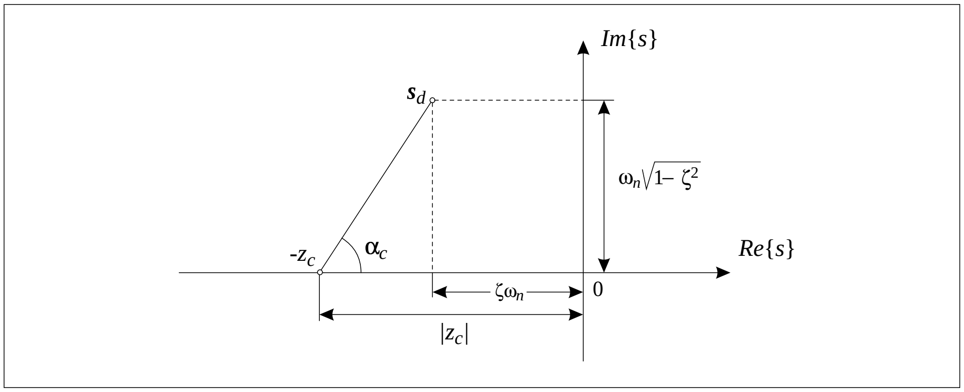

Figure 8:10 Determination of a PD controller's zero location

Design Algorithm 8.3:

1. Choose a pair of complex conjugate dominant poles in the complex plane that produces the desired transient response (damping ratio and natural frequency). Figure 6.2 helps to accomplish this goal.
2. Find the required phase contribution of a PD regulator by using the corresponding formula.
3. Find the absolute value of a PD controller's zero by using the corresponding formula; see also Figure 8.10.
4. Check that the compensated system has a pair of dominant complex conjugate closed-loop poles.

Example 8.8: Let the design specifications be set such that the desired maximum percent overshoot is less than 20% and the 5%-settling time is 1.5 s. Then, the formula for the maximum percent overshoot implies

$$-\frac{\zeta\pi}{\sqrt{1-\zeta^2}} = \ln\{OS\} \;\Rightarrow\; \zeta = \sqrt{\frac{\ln^2\{OS\}}{\pi^2 + \ln^2\{OS\}}} = 0.456$$

We take $\zeta = 0.46$ so that the expected maximum percent overshoot is less than 20%. In order to have the 5%-settling time of 1.5 s, the natural frequency should satisfy

$$t_s \approx \frac{3}{\zeta\omega_n} \quad\Rightarrow\quad \omega_n \approx \frac{3}{\zeta t_s} = 4.348\,\mathrm{rad/s}$$

The desired dominant poles are given by

$$s_d = \lambda_d = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2} = -2.00 \pm j3.86$$

Consider now the open-loop control system

$$G(s) = \frac{K(s+10)}{(s+1)(s+2)(s+12)}$$

The root locus of this system is represented in Figure 8.11a.

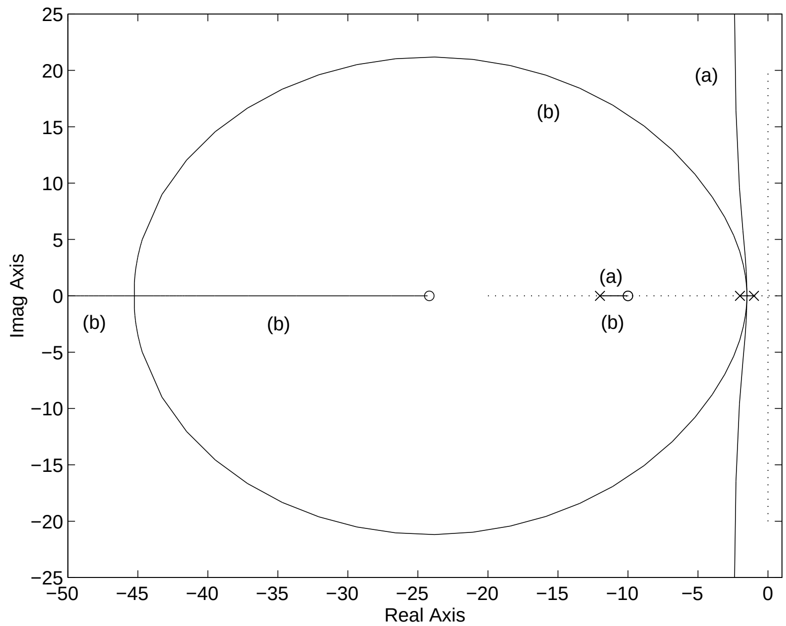

Figure 8.11: Root loci of the original (a) and compensated (b) systems

It is obvious from the above figure that the desired dominant poles do not belong to the original root locus since the breakaway point is almost in the middle of the open-loop poles located at -1 and -2. In order to move the original root locus to the left such that it passes through s_d, we design a PD controller by following Design Algorithm 8.3. Step 1 has been already completed in the previous paragraph. Since we have determined the desired operating point, s_d, we now use the formula for the angles to determine the phase contribution of a PD controller. By MATLAB function `angle` (or just using a calculator), we can find the following angles

$$\measuredangle(s_d + z_1) = 0.4495\,\mathrm{rad}, \quad \measuredangle(s_d + p_1) = 1.8243\,\mathrm{rad}$$
$$\measuredangle(s_d + p_2) = 1.5708\,\mathrm{rad}, \quad \measuredangle(s_d + p_3) = 0.3684\,\mathrm{rad}$$

Note that MATLAB function `angle` produces results in radians. Using the formula for the angles, we get

$$\begin{aligned}\measuredangle(s_d + z_c) &= \pi - 0.4495 + 1.8243 + 1.5708 + 0.3684\\ &= 0.1723\,\mathrm{rad} \;=\; 9.8734^\circ \;=\; \alpha_c\end{aligned}$$

Having obtained the angle α_c, the location of the controller's zero is $z_c = 24.1815$, so that the required PD controller is given by

$$G_c(s) = s + 24.1815$$

The root locus of the compensated system is presented in Figures 8.11b and 8.12b. It can be seen from Figure 8.12 that the point $s_d = -2 \pm j3.86$ lies on the root locus of the compensated system.

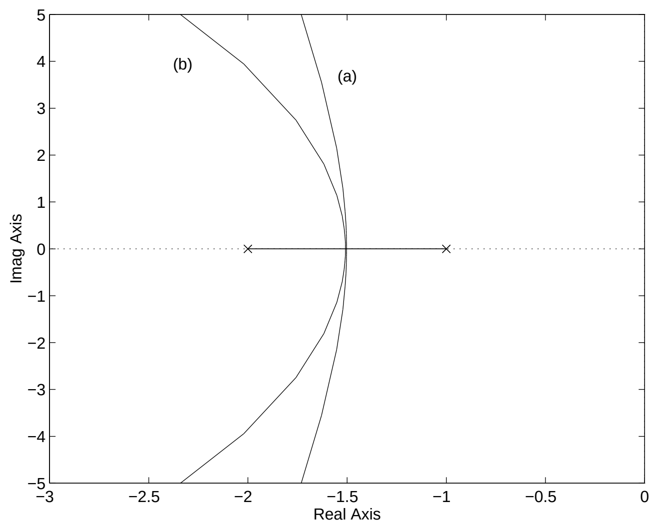

Figure 8.12: Enlarged portion of the root loci in the neighborhood of the desired operating point of the original (a) and compensated (b) systems

At the desired point, s_d, the static gain K, obtained by applying the root locus magnitude rule, is given by $K = 0.825$. This value can be obtained either by using a calculator or the MATLAB function `abs` as follows:

```
d1=abs(sd+p1); d2=abs(sd+p2); d3=abs(sd+p3);
d4=abs(sd+z1); d5=abs(sd+zc);
K=(d1*d2*d3)/(d4*d5)
```

For this value of the static gain K, the steady state errors for the original and compensated systems are given by $e_{ss} = 0.7442$, $e_{ssc} = 0.1074$. In addition, since the controller's zero will attract one of the system poles for large values of K, it is not advisable to choose small values for z_c since it may damage the transient response dominance by the pair of complex conjugate poles closest to the imaginary axis.

The closed-loop step response for this value of the static gain is presented in Figure 8.13.

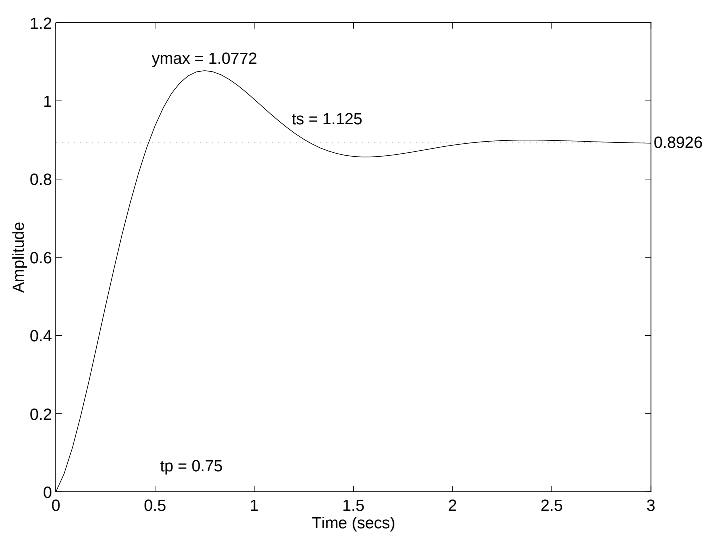

Figure 8.13: Step response of the compensated system for Example 8.8

It can be observed that both the maximum percent overshoot and the settling time are within the specified limits. The values for the overshoot, peak time, and settling time are obtained by the following MATLAB routine:

```
[yc,xc,t]=step(cnumc,cdenc);
% t is a time vector of length i=73;
% cnumc = closed-loop compensated numerator
% cdenc = closed-loop compensated denominator
plot(t,yc);
[ymax,imax]=max(yc);
% ymax is the function maximum;
% imax = time index where maximum occurs;
tp=t(imax)
essc=0.1074;
yss=1–essc;
os=ymax-yss
% procedure for finding the settling time;
delt5=0.05*yss;
i=73;
while abs((yc(i)-yss))<delt5;
i=i-1;
end;
ts=t(i)
```

Using this program, we have found that $t_s = 1.125\,\mathrm{s}$ and $MPOS = 20.68\%$. Our starting assumptions have been based on a model of the second-order system. Since the second-order systems are approximations for higher-order systems that have dominant poles, the obtained results are satisfactory.

Finally, we have to check that the system response is dominated by a pair of complex conjugate poles. Finding the closed-loop eigenvalues we get $\lambda_1 = -11.8251, \quad \lambda_{2,3} = -2.000 \pm j3.8600$, which indicates that the presented controller design results are correct since the transient response is dominated by the eigenvalues $\lambda_{2,3}$.

Phase-Lead Controller Design

The phase-lead controller works on the same principle as the PD controller. It uses the argument rule of the root locus method, which indicates the phase shift that needs to be introduced by the phase-lead controller such that the desired dominant poles (having the specified transient response characteristics) belong to the root locus.

The general form of this controller is given by

$$G_c(s) = \frac{s + z_c}{s + p_c}, \qquad p_c > z_c > 0$$

By choosing a point s_d for a dominant pole that has the required transient response specifications, the design of a phase-lead controller can be done in similar fashion to that of a PD controller. First, find the angle contributed by a controller such that the point s_d belongs the root locus, which can be obtained from

$$\measuredangle G_c(s_d) = 180^\circ - \measuredangle G(s_d)$$

that is

$$\theta_c = \measuredangle(s_d + z_c) - \measuredangle(s_d + p_c) = 180^\circ - \sum_{i=1}^{m} \measuredangle(s_d + z_i) + \sum_{i=1}^{n} \measuredangle(s_d + p_i)$$

Second, find locations of controller's pole and zero. This can be done in many ways as demonstrated in Figure 8.14.

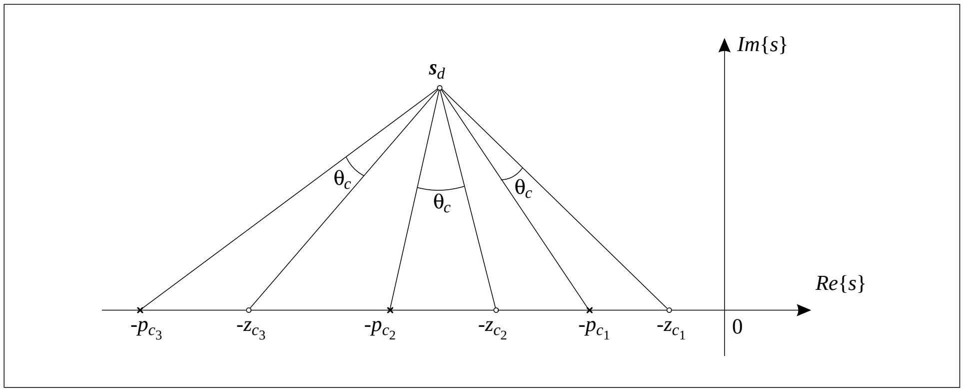

Figure 8.14: Possible locations for poles and zeros of phase-lead controllers that have the same angular contribution

All these controllers introduce the same phase shift and have the same impact on the transient response. However, the impact on the steady state errors is different since it depends on the ratio of z_c/p_c. Since this ratio for a phase-lead controller is less than one, we conclude that the corresponding steady state constant is reduced and the steady state error is increased.

Note that if the location of a phase-lead controller zero is chosen, then simple geometry can be used to find the location of the controller's pole. For example, let $-z_{c3}$ be the required zero, then using Figure 8.14 the pole $-p_{c3}$ is obtained as

$$p_{c3} = \zeta\omega_n + \omega_n\sqrt{1-\zeta^2}\tan\left(\theta_c - \varphi + \pi/2\right)$$

where $\varphi = \measuredangle(s_d + z_{c3})$. Note that $\varphi > \theta_c$.

An algorithm for the phase-lead controller design can be formulated as follows.

Design Algorithm 8.4:

1. Choose a pair of complex conjugate poles in the complex plane that produces the desired transient response (damping ratio and natural frequency). Figure 6.2 helps to accomplish this goal.
2. Find the required phase contribution of a phase-lead controller by using the corresponding formula.
3. Choose values for the controller's pole and zero by placing them arbitrarily such that the controller will not damage the response dominance of a pair of complex conjugate poles. Some authors (e.g. Van de Verte, 1994) suggest placing the controller zero at $-\zeta\omega_n$.
4. Find the controller's pole by using the corresponding formula.
5. Check that the compensated system has a pair of dominant complex conjugate closed-loop poles.

Example 8.9: Consider the following control system represented by its open-loop transfer function

$$G(s) = \frac{K(s+6)}{(s+10)(s^2+2s+2)}$$

It is desired that the closed-loop system have a settling time of $1.5\,\mathrm{s}$ and a maximum percent overshoot of less than 20%. From Example 8.8 we know that the system operating point should be at $s_d = -2 \pm j3.86$. A controller's phase contribution is

$$\begin{aligned}\theta_c &= \pi - 0.7676 + 0.4495 + 1.9072 + 1.7737 \\ &= 6.5044\,\mathrm{rad} = 0.2213\,\mathrm{rad} = 12.6769^\circ\end{aligned}$$

Let us locate a zero at -15 ($z_c = 15$), then the compensator's pole is at $-p_c = -59.2025$. The root loci of the original and compensated systems are given in Figure 8.15, and the corresponding step responses in Figure 8.16.

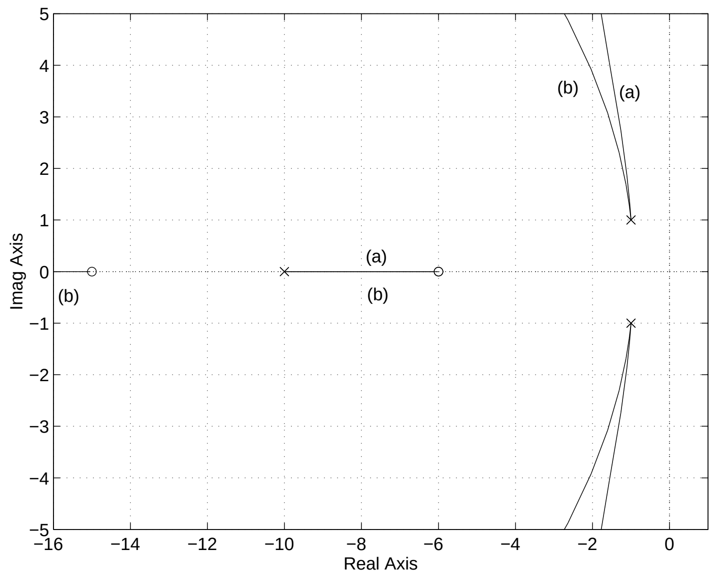

Figure 8.15: Root loci for the original (a) and compensated (b) systems

It can be seen that the root locus indeed passes through the point $-2 \pm j3.86$. For this operating point the static gain is obtained as $K = 101.56$; hence the steady state constants of the original and compensated systems are given by $K_p = 30.468$ and $K_{pc} = K_p(z_c/p_c) = 7.7196$, and the steady state errors are $e_{ss} = 0.0317, e_{ssc} = 0.1147$. Figure 8.16 reveals that for the compensated system both the maximum percent overshoot and settling time are reduced. However, the steady state unit step error is increased, as previously noted analytically.

With a zero set at -9, we have $p_c = 15.291$. The root locus of the compensated system with a new controller is given in Figure 8.17.

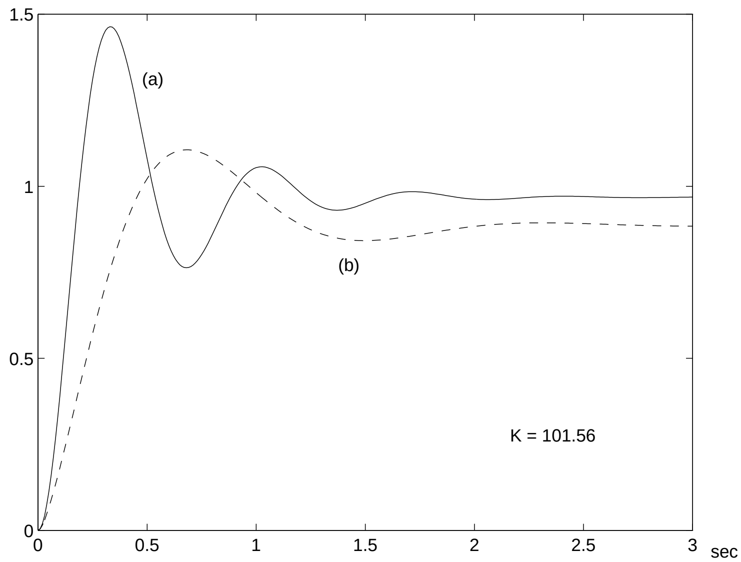

Figure 8.16: Step responses of the original (a) and compensated (b) systems

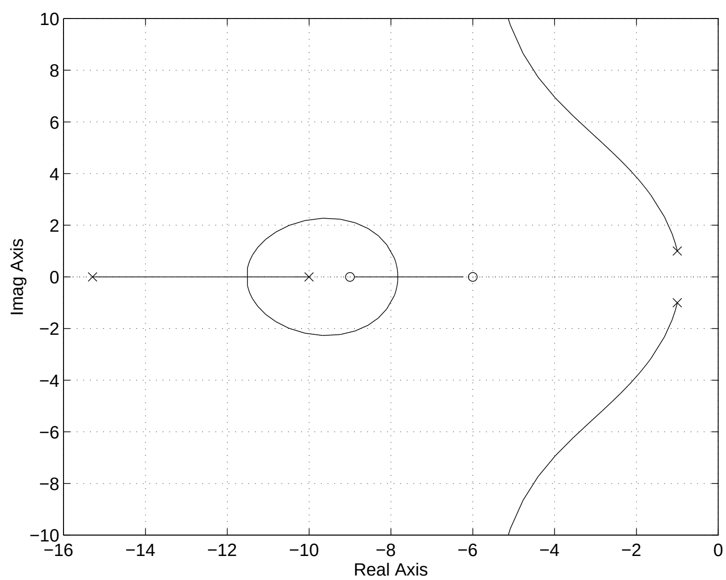

Figure 8.17: Root locus for the compensated system with the second controller

The static gain at the desired operating point $-2 \pm j3.86$ is $K = 41.587$, and hence the steady state errors are $e_{ss} = 0.0742, \ e_{ssc} = 0.11986$. The step responses of the original and compensated systems, for $K = 41.587$, are presented in Figure 8.18.

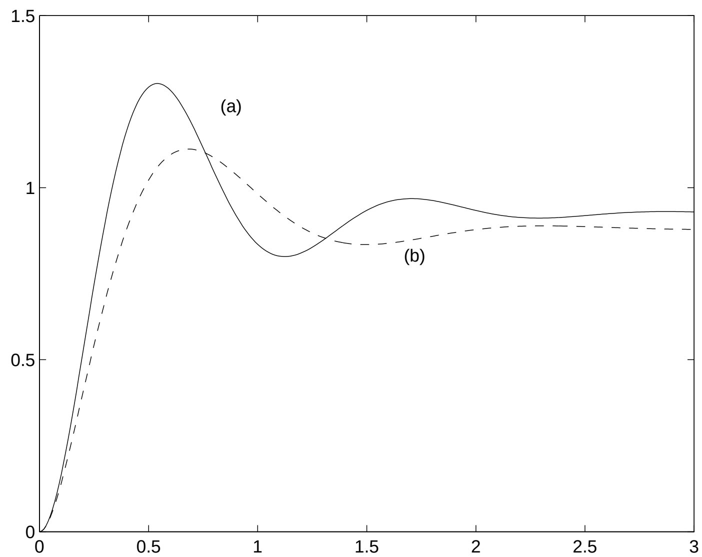

Figure 8.18: Step responses of the original (a) and compensated (b) systems with the second controller for Example 8.9

It can be seen that this controller also reduces both the overshoot and settling time, while the steady state error is slightly increased.

We can conclude that both controllers produce similar transient characteristics and similar steady state errors, but the second one is preferred since the smaller value for the static gain of the compensated system has to be used. The eigenvalues of the closed-loop system for $K = 41.587$ are given by

$$\lambda_{1c} = -12.4165, \quad \lambda_{2c} = -10.8725, \quad \lambda_{2c,3c} = -2.000 \pm j3.8600$$

which indicates that the response of this system is still dominated by a pair of complex conjugate poles.

Remark: In some applications for a chosen desired point, s_d, the required phase increase, θ_c, may be very high. In such cases one can use a *multiple phase-lead controller* having the form

$$G_{lead}^{n}(s)=\left(\frac{s+z_c}{s+p_c}\right)^{n}, \qquad p_c>z_c>0$$

so that each single phase-lead controller has to introduce a phase increase of θ_c/n.

8.5.3 PID and Phase-Lag-Lead Controller Designs

It can be observed from the previous design algorithms that implementation of a PI (phase-lag) controller does not interfere with implementation of a PD (phase-lead) controller. Since these two groups of controllers are used for different purposes—one to improve the transient response and the other to improve the steady state errors—implementing them jointly and independently will take care of both controller design requirements.

Consider first a PID controller. It is represented as

$$\begin{aligned}G_{PID}(s)&=K_p+K_d s+\frac{K_i}{s}=K_d\frac{s^2+\frac{K_p}{K_d}s+\frac{K_i}{K_d}}{s}\\&=K_d(s+z_{c_1})\frac{(s+z_{c_2})}{s}=G_{PD}(s)G_{PI}(s)\end{aligned}$$

which indicates that the transfer function of a PID controller is the product of transfer functions of PD and PI controllers. Since in Design Algorithms 8.1

and 8.3 there are no conflicting steps, the design algorithm for a PID controller is obtained by combining the design algorithms for PD and PI controllers.

Design Algorithm 8.5: PID Controller

1. Check the transient response and steady state characteristics of the original system.
2. Design a PD controller to meet the transient response requirements.
3. Design a PI controller to satisfy the steady state error requirements.
4. Check that the compensated system has the desired specifications.

Example 8.10: Consider the problem of designing a PID controller for the open-loop control system studied in Example 8.8, that is

$$G(s) = \frac{K(s+10)}{(s+1)(s+2)(s+12)}$$

In fact, in that example, we have designed a PD controller of the form

$$G_{PD}(s) = s + 24.1815$$

such that the transient response has the desired specifications. Now we add a PI controller in order to reduce the steady state error. The corresponding steady state error of the PD compensated system in Example 8.8 is $e_{ssc} = 0.1074$. Since a PI controller is a dipole that has its pole at the origin, we propose the following PI controller

$$G_{PI}(s) = \frac{s+0.1}{s}$$

We are in fact using a PID controller with $K_d = 1$, $z_{c1} = 24.1815$, $z_{c2} = 0.1$. The corresponding root locus of this system compensated by a PID controller is represented in Figure 8.19.

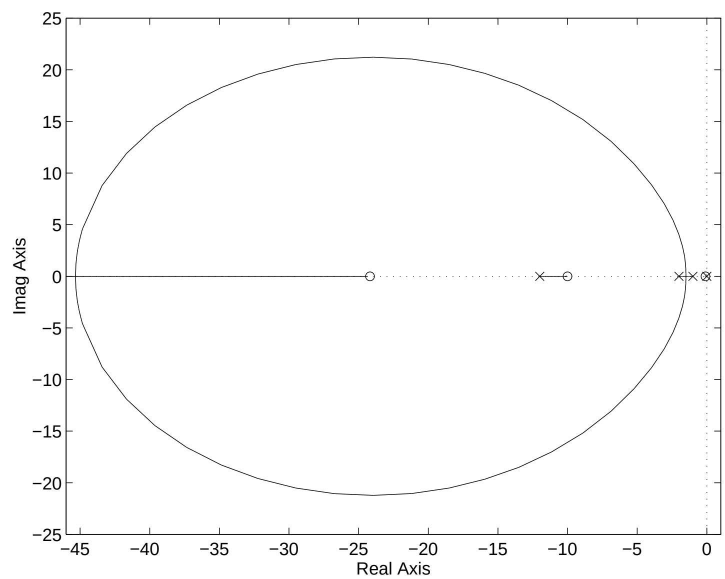

Figure 8.19: Root locus for the system from Example 8.8 compensated by the PID controller

It can be seen that the PI controller does not affect the root locus, and hence Figures 8.11b and 8.19 are almost identical except for a dipole branch.

On the other hand, the step responses of the system compensated by the PD controller and by the PID controller (see Figures 8.13 and 8.20) differ in the steady state parts. In Figure 8.13 the steady state step response tends to $\boldsymbol{y_{ss} = 0.8926}$, and the response from Figure 8.20 tends to 1 since due to the presence of an open-loop pole at the origin, the steady state error is reduced to zero. Thus, we can conclude that the transient response is the same one as that obtained by the PD controller in Example 8.8, but the steady state error is improved due to the presence of the PI controller.

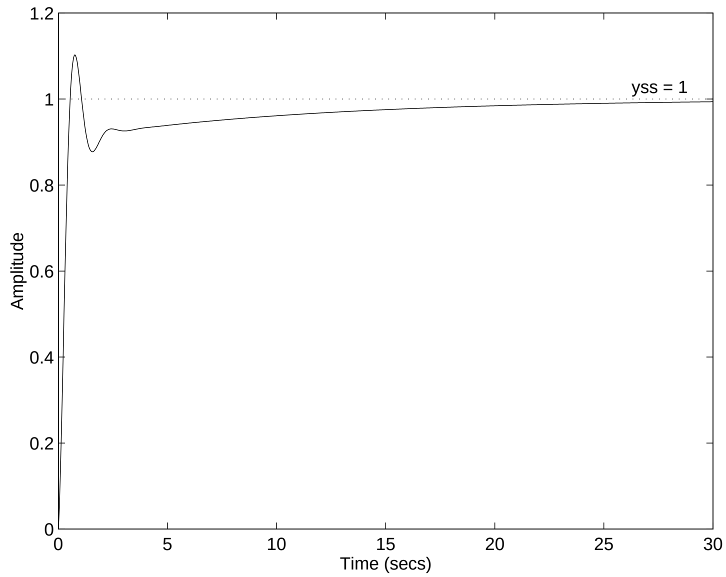

Figure 8.20: Step response of the system from Example 8.8 compensated by the PID controller

Similarly to the PID controller, the design for the phase-lag-lead controller combines Design Algorithms 8.2 and 8.4. Looking at the expression for a phase-lag-lead controller given in formula (8.20), it is easy to conclude that

$$G_{lag/lead}(s) = G_{lag}(s)G_{lead}(s)$$

The phase-lag-lead controller design can be implemented by the following algorithm.

Design Algorithm 8.6: Phase-Lag-Lead Controller

1. Check the transient response and steady state characteristics of the original system.
2. Design a phase-lead controller to meet the transient response requirements.
3. Design a phase-lag controller to satisfy the steady state error requirements.
4. Check that the compensated system has the desired specifications.

Example 8.11: In this example we design a phase-lag-lead controller for a control system from Example 8.9, that is

$$G(s) = \frac{K(s+6)}{(s+10)(s^2+2s+2)}$$

such that both the system transient response and steady state errors are improved. We have seen in Example 8.9 that a phase-lead controller of the form

$$G_{lead}(s) = \frac{s+9}{s+15.291}$$

improves the transient response to the desired one. Now we add in series with the phase-lead controller another phase-lag controller, which is in fact a dipole near the origin. For this example we use the following phase-lag controller

$$G_{lag}(s) = \frac{s+0.1}{s+0.01}$$

so that the compensated system becomes

$$G(s) = G(s)G_c(s) = \frac{K(s+6)}{(s+10)(s^2+2s+2)} \frac{(s+9)}{(s+15.291)} \frac{(s+0.1)}{(s+0.01)}$$

The corresponding root locus of the compensated system and its closed-loop step response are represented in Figures 8.21 and 8.22. We can see that the addition of the phase-lag controller does not change the transient response, i.e. the root loci in Figures 8.17 and 8.21 are almost identical. However, the phase-lag controller reduces the steady state error from $e_{ss,lead} = 0.11986$ to $e_{ss,lag/lead} = 0.01344$ since the position constant is increased to

$$K_{p,lag/lead} = K_{p,lead}\frac{0.1}{0.01} = \frac{41.587 \times 9 \times 0.1}{10 \times 2 \times 15 \times 0.01} = 73.432$$

so that

$$e_{ss,lag/lead} = \frac{1}{1 + K_{p,lag/lead}} = 0.01344$$

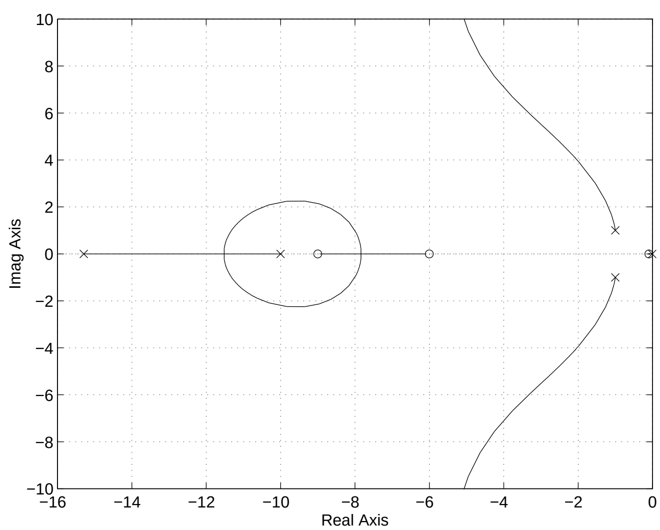

Figure 8.21: Root locus for the system from Example 8.9 compensated by the phase-lag-lead controller

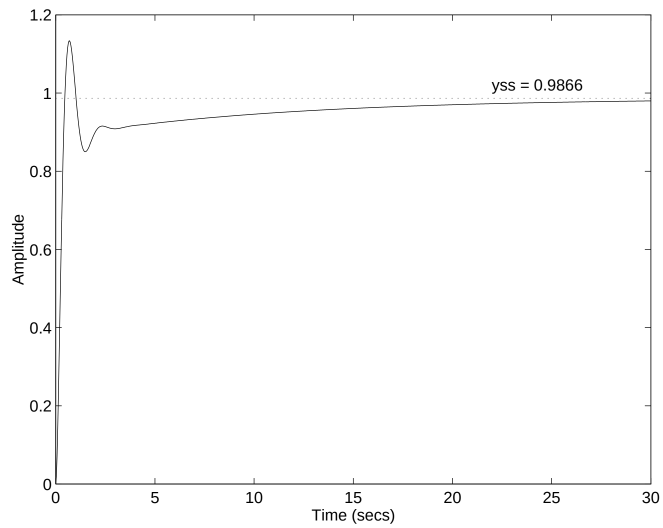

Figure 8.22: Step response of the system from Example 8.9 compensated by the phase-lag-lead controller

8.6 MATLAB Case Studies

In this section we consider the compensator design for two real control systems: a PD controller designed to stabilize a ship, and a PID controller used to improve the transient response and steady state errors of a voltage regulator control system.

8.6.1 Ship Stabilization by a PD Controller

Consider a ship positioning control system defined in the state space form in Problem 7.5. The open-loop transfer function of this control system is

$$G(s) = \frac{0.8424}{s(s+0.0546)(s+1.55)}$$

The root locus of the original system is presented in Figure 8.23a.

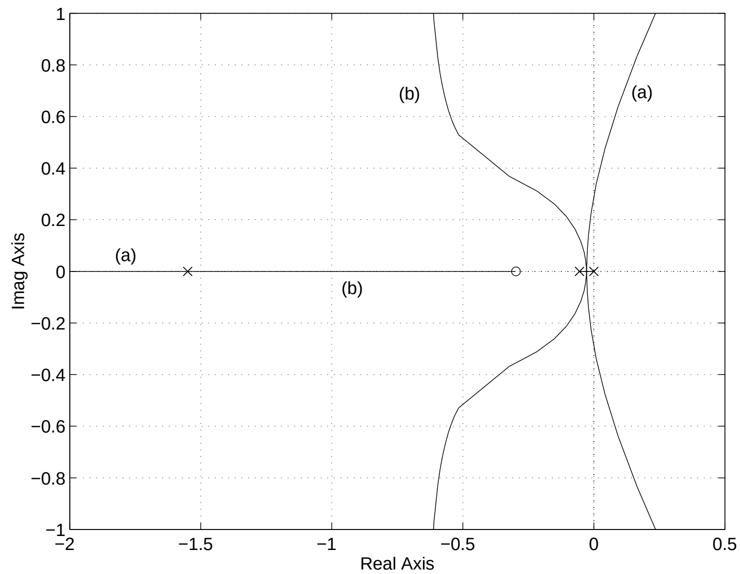

Figure 8.23: Root loci for a ship positioning control problem: (a) original system, (b) compensated system

It can be seen that this system is unstable even for very small values of the static gain. Thus, the system transient response blows up very quickly due to the system's instability. Our goal is to design a PD controller in order to stabilize the system and improve its transient response. Let the desired operating point be located at $s_d = -0.2 \pm j0.3$, which implies $w_n = 0.3606\,\mathrm{rad/s}$ and $\zeta = 0.5547$. We find that the required phase shift is $\alpha_c = 72.0768^\circ$, and the location of the compensator zero is obtained at -0.297. Thus, the PD compensator sought is of the form

$$G_c(s) = s + 0.297$$

It can be seen from Figure 8.23 that the root locus of the compensated system indeed passes through the point $s_d = -0.2 \pm j0.3$ and that the compensated system is stable for all values of the static gain. The static gain at the desired operating point is given by $K_{s_d} = 0.6258$ and the corresponding closed-loop eigenvalues at this operating point are $\lambda_{1c} = -1.2046, \ \lambda_{2c,3c} = -0.2 \pm j0.3$. In Figure 8.24 the unit step response of the compensated system is presented.

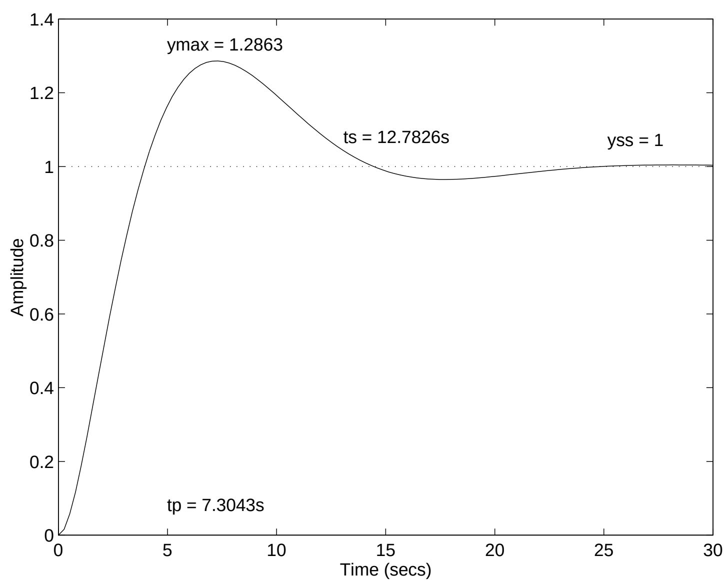

Figure 8.24: Step response of a ship positioning compensated control system

It is found that $y_{max} = 1.2863$, $t_p = 7.3043$ s, and $t_s = 12.7826$ s. From the same figure we observe that the steady state error for this system is zero, which also follows from the fact that the system open-loop transfer function has one pole at the origin.

8.6.2 PID Controller for a Voltage Regulator Control System

The mathematical model of a voltage regulator control system is given in Section 6.7. The open-loop transfer function of this system is

$$G(s) = \frac{154280}{(s+0.2)(s+0.5)(s+10)(s+14.28)(s+25)}$$

The corresponding root locus is presented in Figure 8.25. Since one of the branches goes quite quickly into the instability region, our design goal is to move this branch to the left so that it passes through the operating point selected as $s_d = -1 \pm j1$. For this operating point, we have $w_n = \sqrt{2}$ rad/s and $\zeta = 0.7071$ so that the expected maximum percent overshoot and the 5%-settling time of the compensated system are $MPOS = 4.3214\%,\ t_s = 3$ s. In addition, the design objective is to reduce the steady state error due to a unit step to zero.

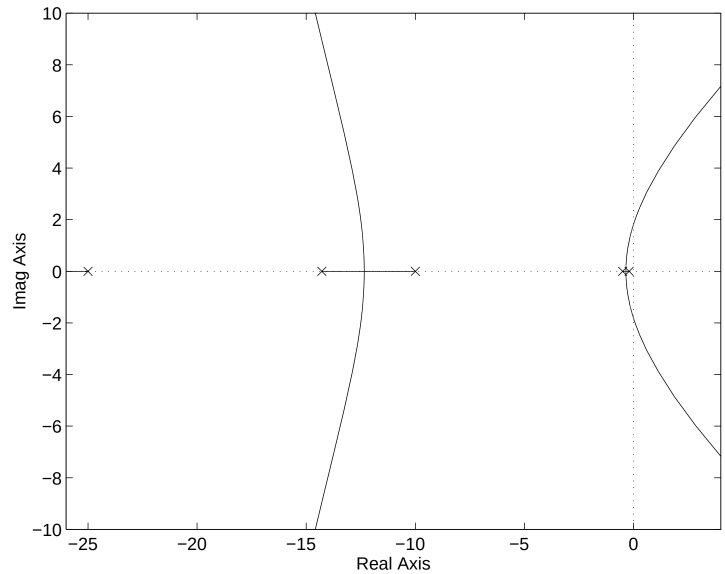

Figure 8.25: Root locus for a voltage regulator system

We use a PID controller to solve the controller design problem defined above. The required phase improvement for the selected operating point is found as $\alpha_c = 1.3658\,\mathrm{rad} = 78.2573^\circ$. The location of the compensator's zero is obtained as $-z_c = -1.2079$, so that the PD part of a PID compensator is

$$G_{PD}(s) = s + 1.2079$$

The branches of the root loci in the neighborhood of the desired operating point of the original and PD compensated systems are presented in Figure 8.26. It can be seen that the compensated root locus indeed passes through the point $s_d = -1 \pm j1$.

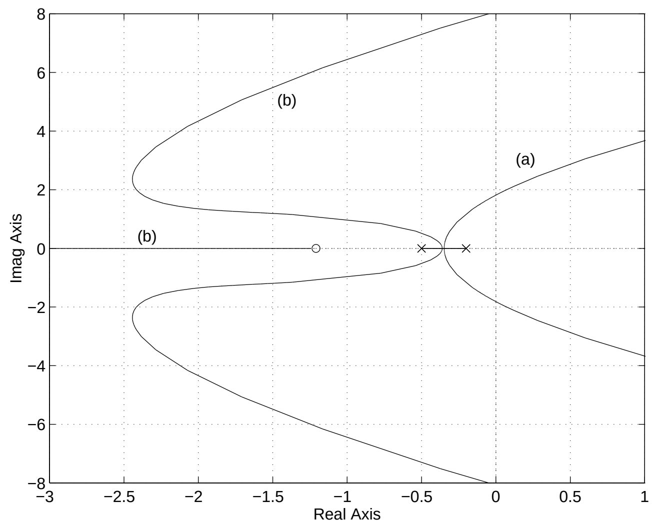

Figure 8:26: Root loci of the original (a) and PD (b) compensated systems

The closed-loop unit step response of the system compensated by the PD controller is represented in Figure 8.27. Using the MATLAB programs given in Example 8.8, gives $MPOS = 6.08\%$, $t_p = 2.1$ s, and $t_s = 3.5$ s, which is quite satisfactory. However, the steady state unit step error is $e_{ssPD} = 0.0808$. Note that the static gain at the operating point, obtained by applying the root locus rule number 9 from Table 7.1, is $K_{s_d} = 4060.8$. The closed-loop eigenvalues at the operating point are

$$\lambda_{1_{PD}} = -23.7027, \quad \lambda_{2_{PD}} = -18.1675, \quad \lambda_{3_{PD}} = -6.1105$$
$$\lambda_{4,5_{PD}} = -0.997 \pm j1.0011$$

which indicates that the system has preserved a pair of dominant complex conjugate poles.

In order to reduce this steady state error to zero we use a PI controller of the form

$$G_{PI}(s) = \frac{s + 0.1}{s}$$

Since the compensated system open-loop transfer function now has a pole at the origin, we conclude that the steady state error is reduced to zero, which can also be observed from Figure 8.27.

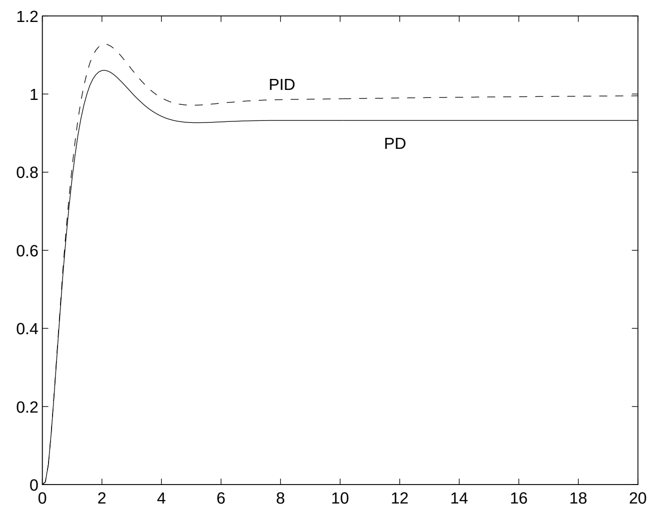

Figure 8.27: Step responses of PD and PID compensated systems

The transient response specifications for the system compensated by the proposed PID controller are $MPOS = 11.277\%$, $t_p = 2.1$ s, and $t_s = 3.1$ s. Thus, the proposed PI controller has slightly worsened the transient response characteristics. It can be checked that the transient response specifications of the compensated system obtained by using PI controllers that have zeros located at -0.01 and -0.001 are improved.