CHAPTER 10

Cascade Control

Objectives of the Chapter

- To review classical cascade control.
- To present an alternate way of thinking about cascade control that leads to improved performance.
- To introduce controller design methods that accommodate process uncertainty.

Prerequisite Reading

Chapter 3, "One-Degree of Freedom Internal Model Control"
Chapter 4, "Two-Degree of Freedom Internal Model Control"
Chapter 5, "MSF Implementations of IMC Systems"
Chapter 6, "PI and PID Controller Parameters from IMC Design"
Chapter 7, "Tuning and Synthesis of 1DF IMC Controllers for Uncertain Processes"
Chapter 8, "Tuning and Synthesis of 2DF Control Systems"

10.1 INTRODUCTION

Cascade control can improve control system performance over single-loop control whenever either: (1) Disturbances affect a measurable intermediate or secondary process output that directly affects the primary process output that we wish to control; or (2) the gain of the secondary process, including the actuator, is nonlinear. In the first case, a cascade control system can limit the effect of the disturbances entering the secondary variable on the primary output. In the second case, a cascade control system can limit the effect of actuator or secondary process gain variations on the control system performance. Such gain variations usually arise from changes in operating point due to setpoint changes or sustained disturbances.

A typical candidate for cascade control is the shell and tube heat exchanger of Figure 10.1.

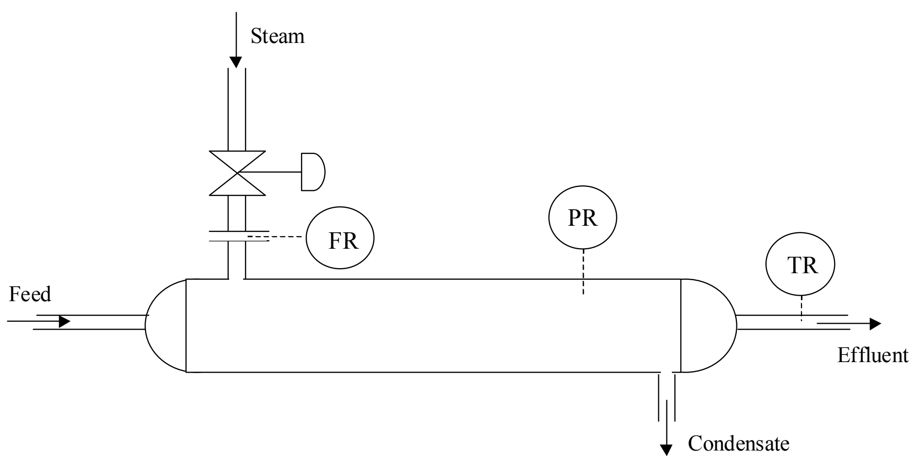

Figure 10.1 A shell and tube heat exchanger.

The primary process output is the temperature of the tube side effluent stream. There are two possible secondary variables, the flow rate of steam into the exchanger and the steam pressure in the exchanger. The steam flow rate affects the effluent temperature through its effect on the steam pressure in the exchanger. The steam pressure in the exchanger affects the effluent temperature by its effect on the condensation temperature of the steam. Therefore, either the steam flow rate or the steam pressure in the exchanger can be used as the secondary output in a cascade control system. The choice of which to use depends on the disturbances that affect the effluent temperature.

If the main disturbance is variations in the steam supply pressure, due possibly to variable steam demands of other process units, then controlling the steam flow with the control valve is most likely to be the best choice. Such a controller can greatly diminish the effect of steam supply pressure variations on the effluent temperature. However, it is still necessary to have positive control of the effluent temperature to be able to track effluent temperature setpoint changes and to reject changes in effluent temperature due to feed temperature and flow variation. Since there is only one control effort, the steam valve stem position, traditional cascade control uses the effluent temperature controller to adjust the setpoint of the steam flow controller, as shown in Figure 10.2.

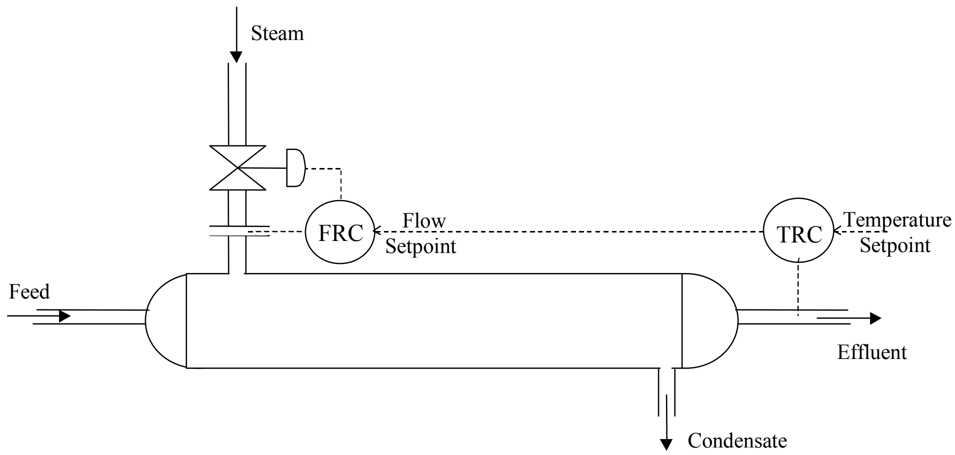

Figure 10.2 Cascade control of effluent temperature via steam flow control.

If feed flow and temperature variations are significant, then these disturbances can be at least partially compensated by using the exchanger pressure rather than the steam flow as the secondary variable in a cascade loop, as shown in Figure 10.3.

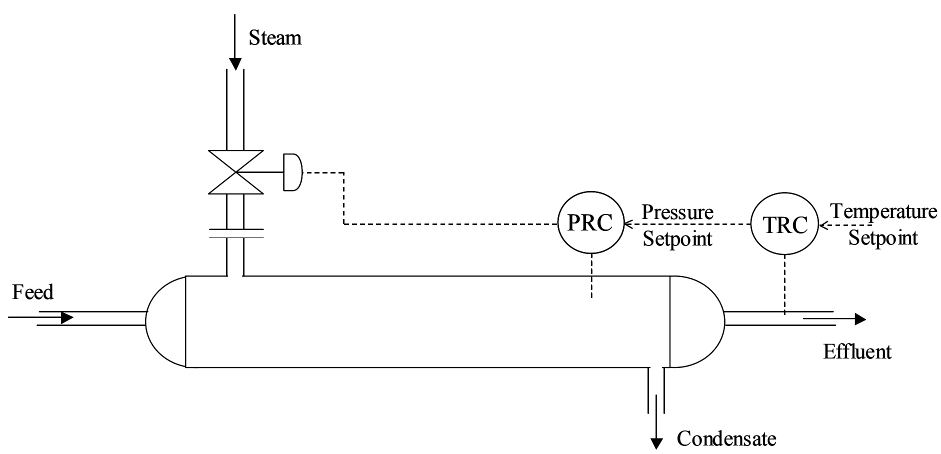

Figure 10.3 Cascade control of effluent temperature via shell side pressure control.

The trade-off in using the configuration of Figure 10.3 rather than that of Figure 10.2 is that the inner control loop from the steam pressure to the valve stem position may not suppress variations in valve gain as well as with an inner loop that uses the valve to control the steam flow rate. This consideration relates to using a cascade control system to suppress the effect of process uncertainty, in this case the valve gain, on the control of the primary process variable, the effluent temperature. We will have a lot more to say about using cascade control systems to suppress process uncertainty in the following sections.

To repeat, cascade control has two objectives. The first is to *suppress the effect of disturbances on the primary process output* via the action of a secondary, or inner control loop around a secondary process measurement. The second is to *reduce the sensitivity of the primary process variable to gain variations* of the part of the process in the inner control loop.

As we shall demonstrate, cascade control can be usefully applied to any process where a measurable secondary variable directly influences the primary controlled variable through some dynamics. We will also demonstrate that despite frequent literature statements to the contrary, inner loop dynamics do not have to be faster than the outer loop dynamics. However, the traditional cascade structure and tuning methods must be modified in order for cascade control to achieve its objectives when the inner loop process has dynamics that are on the order of, or slower than, the primary process dynamics.

10.2 Cascade Structures and Controller Designs

Figure 10.4 shows the traditional PID cascade control system block diagram (Seborg et al., 1989). This is the cascade structure associated with figures 10.2 and 10.3. For Figure 10.2, the secondary process variable y_2 is the steam flow rate, while for Figure 10.3, it is the shell-side steam pressure. In both cases, the primary variable y_1 is the effluent temperature.

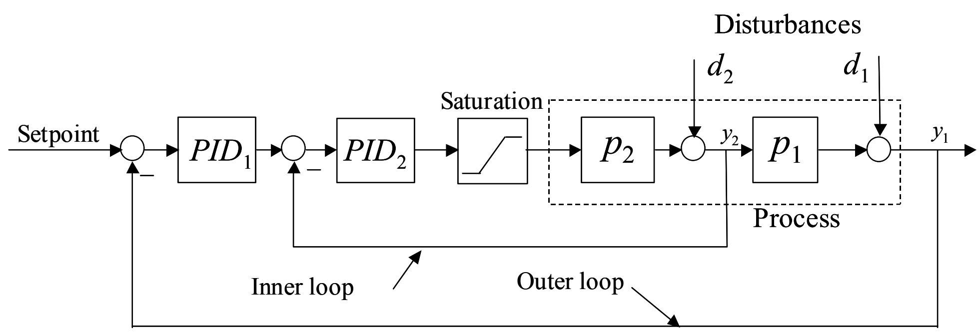

Figure 10.4 Traditional cascade block diagram.

One of the objectives of this section is to present methods for obtaining the parameters of the PID controllers of Figure 10.4 from a well-designed and well-tuned IMC cascade control system, just as we did for single-loop control systems in Chapter 6.

Figure 10.5 shows an IMC cascade block diagram that accomplishes the same objectives as Figure 10.4. There are other equivalent IMC cascade structures to that given by Figure 10.5 (Morari and Zafiriou, 1989). However, the configuration of Figure 10.5 is convenient because it suggests that *controller q_2 should be designed and tuned solely to suppress the effect of the disturbance d_2 on the primary output y_1*, and also convenient because both controller outputs u_1 and u_2 enter directly into the actuator. As we shall see later, this last point facilitates dealing with control effort saturation. However, for the remainder of this section we shall ignore the saturation block in order to study the design and tuning of IMC controllers. These IMC controllers will then be used to obtain the PID controller parameters in Figure 10.4, as was done in Chapter 6 for single-loop control systems.

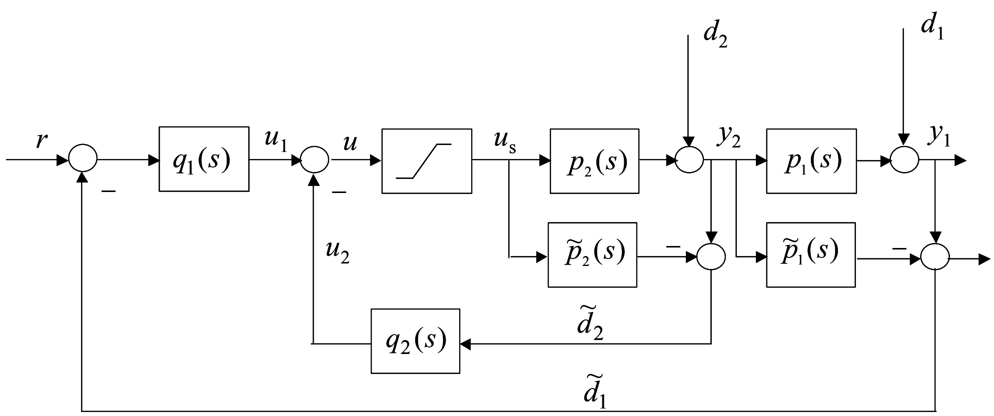

Figure 10.5 IMC cascade structure.

From Figure 10.5, the transfer functions between the inputs to the inner loop, u_1 and d_2, and the secondary process output y_2 are

$$y_2(s) = \frac{p_2(s)u_1(s) + (1 - \tilde{p}_2(s)q_2(s))d_2(s)}{(1 + (p_2(s) - \tilde{p}_2(s))q_2(s)} \,. \qquad (10.1)$$

The transfer functions between the setpoint and disturbances and the primary process output y_1 are

$$y_1(s) = \frac{p_1 p_2 q_1 r(s) + (1 - \tilde{p}_2 q_2) p_1 d_2(s) + (1 - \tilde{p}_1 p_2 q_1 + (p_2 - \tilde{p}_2) q_2) d_1(s)}{(1 + (p_1 - \tilde{p}_1) p_2 q_1 + (p_2 - \tilde{p}_2) q_2)} . \qquad (10.2)$$

In Eq. (10.2) we have suppressed the dependency of all transfer functions on the Laplace variable s to keep the equation on one line. Based on equations (10.1) and (10.2) we observe the following:

(1) If the lag time constants of the primary process $p_1(s)$ are large relative to those of the secondary process $p_2(s)$ then the inner loop controller $q_2(s)$ should be chosen so that the zeros of $(1-\tilde{p}_2(s)q_2(s))$ cancel the small poles (i.e., large time constants) of $\tilde{p}_1(s)$ as outlined in Chapter 4. Otherwise, $q_2(s)$ should simply invert a portion of $\tilde{p}_2(s)$ as described in chapters 3 and 7.

(2) The outer loop controller $q_1(s)$ should approximately invert the entire process model $\tilde{p}_1\tilde{p}_2(s)$, as described in chapters 3 and 7.

(3) The IMCTUNE software can be used to design and tune both $q_1(s)$ and $q_2(s)$.

We recommend tuning $q_2(s)$ with the outer loop open, and then tuning $q_1(s)$ with the inner loop closed. That is, first find the filter time constant ε_2 for $q_2(s)$, and then find ε_1 for $q_1(s)$. According to the denominator of Eq. (10.2), the tunings for $q_1(s)$ and $q_2(s)$ interact. Therefore, some adjustment of ε_2 may be necessary after obtaining ε_1.

Having obtained the IMC controllers for Figure 10.5, we would like to use these controllers to obtain the PID controllers in Figure 10.4 in a manner similar to that for single-loop controllers described in Chapter 6. Unfortunately, however, we can do so only very approximately. Figure 10.5 can be rearranged, ignoring the saturation block, as given by Figure 10.6.

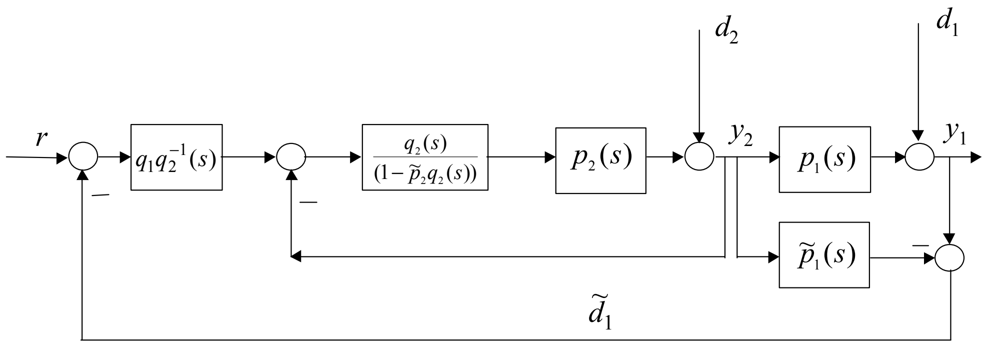

Figure 10.6 IMC cascade control with a simple feedback inner loop.

The controller given by $q_2(s)/(1-\tilde{p}_2(s)q_2(s))$ can often be well approximated by a PID controller, as described in Chapter 6. Again, IMCTUNE can be used to obtain this controller. However, obtaining PID_1 in Figure 10.4 is not so straightforward. Collapsing the feedback loop through $\tilde{p}_1(s)$, while leaving the inner loop alone, yields Figure 10.7.

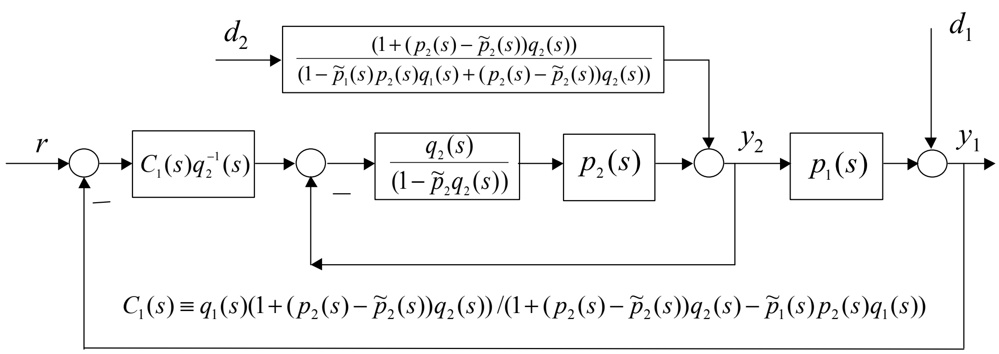

Figure 10.7 Standard feedback form of Figure 10.6.

The controller $C_1(s)$ in Figure 10.7 cannot be realized because it contains the process transfer function $p_2(s)$, which is uncertain and cannot be made part of the controller. We can however approximate $p_2(s)$ with its model $\tilde{p}_2(s)$. In this case $C_1(s)$ becomes

$$C_1(s) \cong q_1(s)/(1-\tilde{p}_1(s)\tilde{p}_2(s)q_1(s)). \qquad (10.3)$$

Another difference between figures 10.6 and 10.7 is that even if the model $\tilde{p}_1(s)$ is a perfect representation of the process, the pulse created by the inner loop response to the disturbance $d_2(s)$ (i.e., $d_2(s)/(1-\tilde{p}_1(s)\tilde{p}_2(s)q_2(s))$ for a perfect model $\tilde{p}_2(s)$) feeds back around the outer loop of Figure 10.7. Since the primary controller cannot suppress this pulse, it continues around the loop until it dies out.

Even using the approximation given by Eq. (10.3) to obtain a PID controller does not reduce Figure 10.7 to the standard PID cascade diagram of Figure 10.4 because $C_1(s)$ in Figure 10.7 is multiplied by $q_2^{-1}(s)$. If $q_2^{-1}(s)$ is a lead (which will generally occur only if the process description is quite uncertain), then $q_2^{-1}(s)$ can be approximated by a polynomial via a Taylor's series. This polynomial can be multiplied into the PID controller obtained from $C_1(s)$ to obtain a new PID controller after dropping higher order terms. Even if $q_2^{-1}(s)$ is a lag, it may still be possible to approximate the term $C_1(s)\,q_2^{-1}(s)$ by a PID controller. However, the necessary approximations will have to be carried out by hand, following procedures in Chapter 6, as the current version of IMCTUNE does not carry out the necessary manipulations.

Two rather long examples of cascade control of uncertain processes follow. The individual processes in both examples are first-order lags plus dead time and have significant process uncertainty. In the first example, the secondary process output dynamics are significantly faster than the primary process dynamics. In the second example, the primary and secondary process dynamics have similar dynamic behavior.

Example 10.1 Secondary Process has Faster Dynamics than the Primary Process

$$p_1(s) = \frac{K_1 e^{-T_1 s}}{\tau_1 s + 1}; \; 0.8 \le K_1 \le 1.2, \quad 17.5 \le T_1 \le 22.5, \quad 14 \le \tau_1 \le 16 \qquad (10.4a)$$

$$p_2(s) = \frac{K_2 e^{-T_2 s}}{\tau_2 s + 1}; \; 0.6 \le K_2 \le 1.8, \quad 2 \le T_2 \le 4, \quad 1 \le \tau_2 \le 3 \qquad (10.4b)$$

(a) IMC System Design

Following the suggestions in chapters 7 and 8, we use the upper-bound gains and dead times and the lower-bound time constants for the process models.

$$\tilde{p}_1(s) = \frac{1.2 e^{-22.5s}}{14s + 1} \qquad (10.4c)$$

$$\tilde{p}_2(s) = \frac{1.8 e^{-4s}}{s + 1} \qquad (10.4d)$$

Computing the 2DF feedback controller for the inner loop (see Figure 10.5), using IMCTUNE with the outer loop, open gives

$$q_2(s) = \frac{(s+1)(9.05s+1)}{1.8(4.4s+1)^2}. \qquad (10.5a)$$

Figure 10.8 shows the tuning curves, while Figure 10.9 shows typical time responses to a step disturbance in the inner loop. Data for both figures was obtained from IMCTUNE.

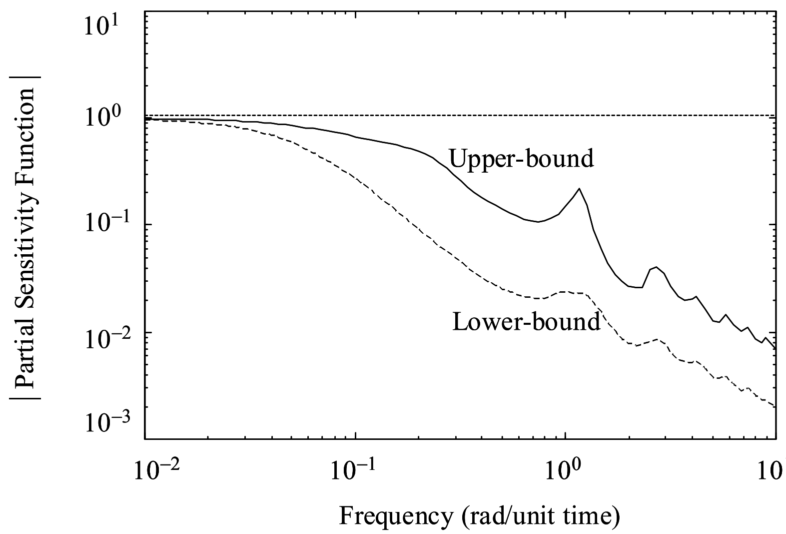

Figure 10.8 Cascade inner loop tuning using controller given by Eq. (10.5a).

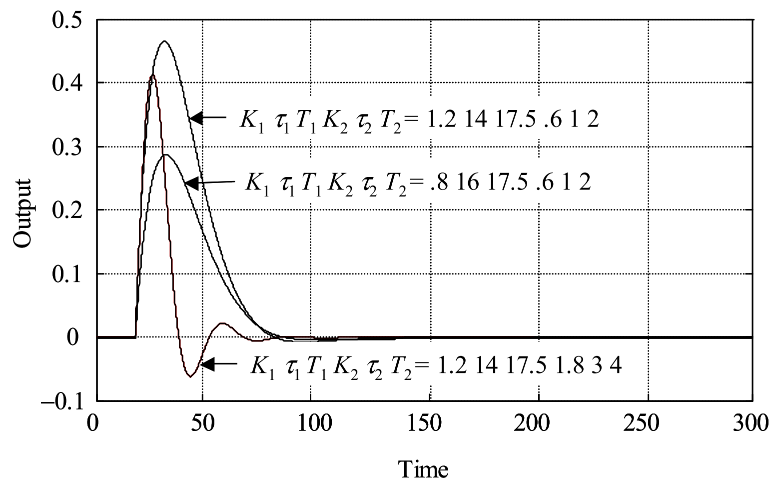

Figure 10.9 Responses to a step inner loop disturbance (d_2) with the outer loop open.

Having obtained the inner loop controller, the outer loop controller can be obtained from the cascade facility of IMCTUNE, and is

$$q_1(s) = \frac{(15s+1)}{2.16(16.87s+1)}. \tag{10.5b}$$

The tuning curves for the outer loop of the cascade, using Eq. (10.5b), are shown in Figure 10.10. Also in this figure are the closed-loop upper-bound and lower-bound curves for a single-loop controller for a model and controller of

$$\tilde{p}(s) = \frac{2.16\,e^{-26.5s}}{15s + 1} \qquad q(s) = \frac{(15s+1)}{2.16\,(14.3s+1)}. \tag{10.6}$$

Recall from equations (10.4a) and (10.4b) that the overall process is

$$p(s) = \frac{Ke^{-Ts}}{(\tau_1 s + 1)(\tau_2 s + 1)} \quad 0.48 \le \mathrm{K} \le 2.16, \quad 19.5 \le \mathrm{T} \le 26.5, \quad 14 \le \tau_1 \le 16 \quad 1 \le \tau_2 \le 3.$$

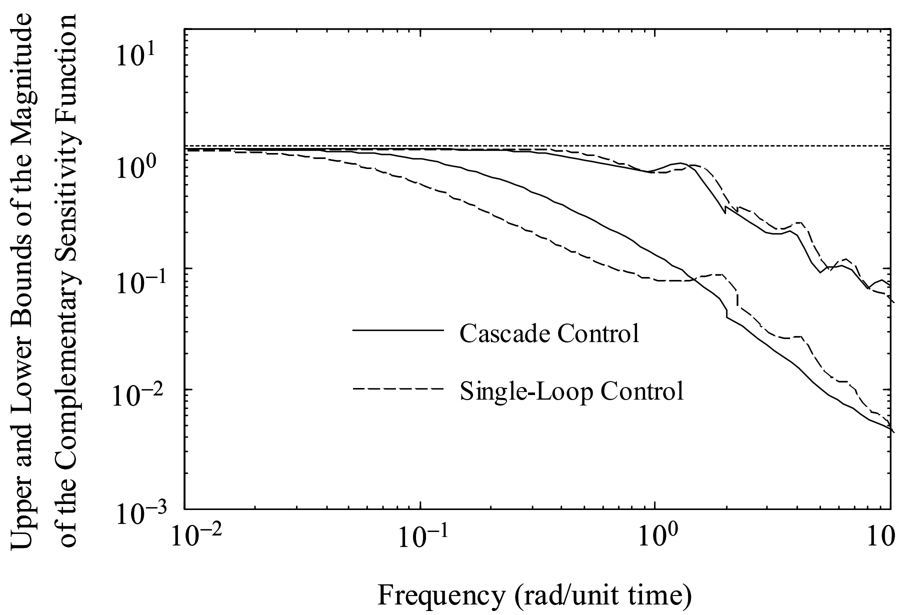

Figure 10.10 Comparison of closed-loop setpoint to output responses.

Based on the closed-loop frequency responses we can conclude that the fastest responses of the single-loop system are slightly faster than those of the cascade system, but more importantly, the slowest responses are significantly slower. Figures 10.11 and 10.12 support these conclusions. Note the different time axes in figures 10.11 and 10.12.

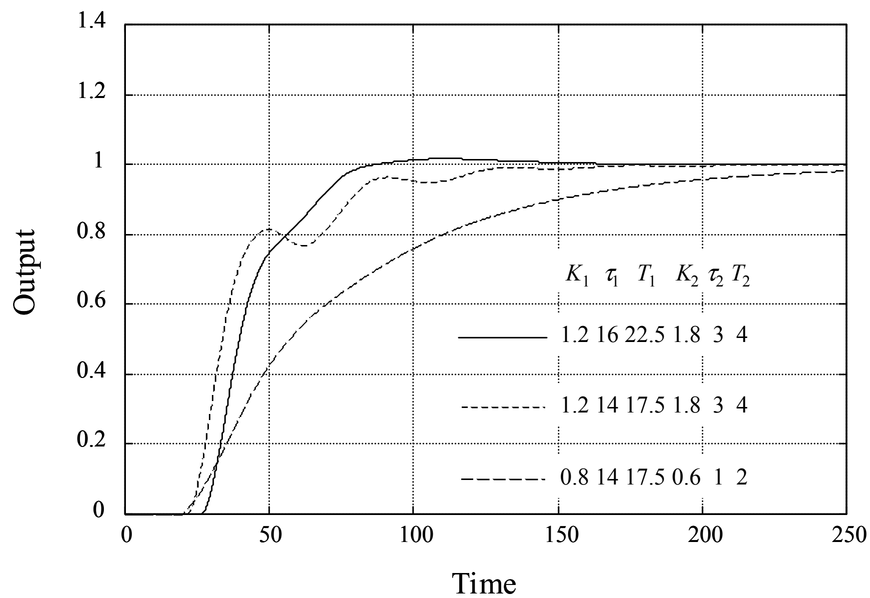

Figure 10.11 Step setpoint responses for the cascade control system of Figure 10.5.

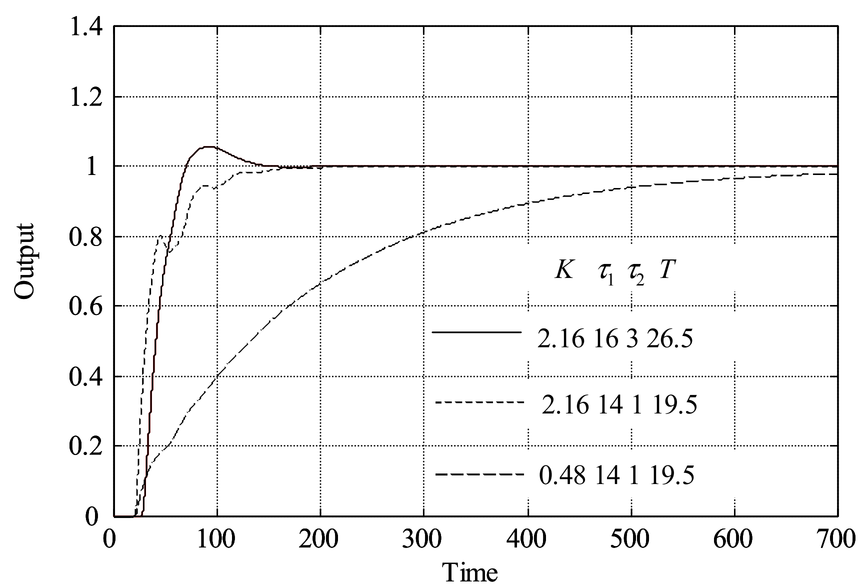

Figure 10.12 Step setpoint responses for the single-loop control system, using Eq. (10.6).

The reason for the improved setpoint response of the cascade system is that the inner loop of the cascade reduces the effect of gain uncertainty in the inner loop process. To show that this is so, Figure 10.13 compares the closed-loop frequency responses of the cascade system with that of a single-loop controller. The process is the same as that given by equations (10.4a) and 10.4b), except that instead of a lower-bound of 0.48, the lower bounds (*lb*) are 1.1 and 1.44. That is, the single-loop process is

$$p(s)=\frac{Ke^{-Ts}}{(\tau_1 s+1)(\tau_2 s+1)}\quad lb\le \mathrm{K}\le 2.16,\quad 19.5\le \mathrm{T}\le 26.5,\quad 14\le\tau_1\le 16\quad 1\le\tau_2\le 3. \quad (10.7)$$

The model and controller for the process of Eq. (10.7) are the same as given in Eq. (10.6) and are repeated for convenience:

$$\widetilde{p}(s)=\frac{2.16\,e^{-26.5s}}{15s+1}\qquad q(s)=\frac{(15s+1)}{2.16\,(14.3s+1)}$$

A lower-bound gain of 1.44 corresponds to a secondary process (i.e., $\widetilde{p}_2(s)$) with a gain of 1.8 and no gain uncertainty. A lower-bound gain of 1.1 corresponds to a secondary process whose gain varies between 1.375 and 1.8. In other words, the effect on the outer loop of the ratio of the maximum to minimum gain variation of the secondary process has been reduced from a ratio of 3 to a ratio of 1.3. The slowest time responses are compared in Figure 10.14.

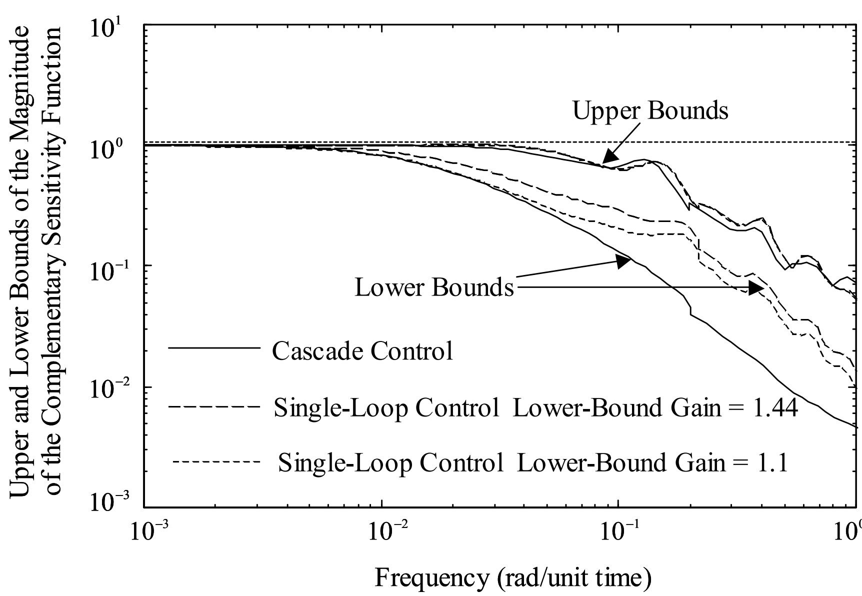

Figure 10.13 Comparison of cascade and single-loop control systems.

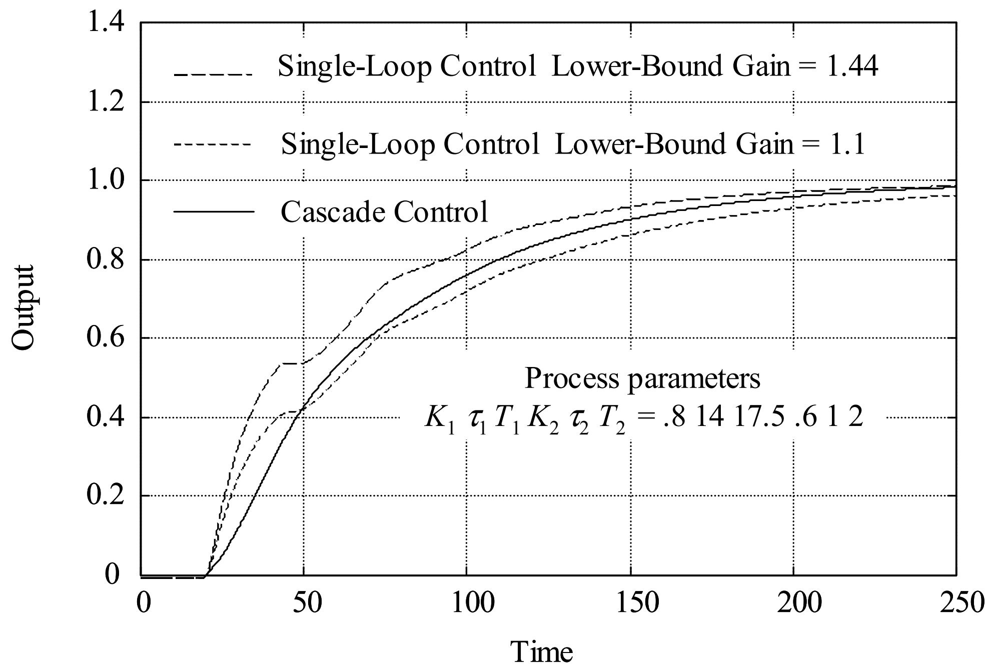

Figure 10.14 Comparison of slowest responses to a step setpoint change.

We now return to the cascade control system responses to a step disturbance to the inner loop, but this time with the outer loop closed. The time responses for the same processes as in Figure 10.9 are shown in Figure 10.15. From this figure we conclude that there is no need to retune the inner loop.

Figure 10.16 shows the effect of using the single-degree of freedom IMC controller given by Eq. (10.8) on the response to a step disturbance in the inner loop. These responses should be compared with those of Figure 10.15.

$$q_2(s) = \frac{(s+1)}{1.8(4.18s+1)}. \tag{10.8}$$

The filter time constant of 4.18 in Eq. (10.8) yields an Mp of 1.05. That is, the controller is tuned so that the worst-case overshoot of the inner loop output y_2 to a step setpoint change to the inner loop is about 10% with the controller q_2 in the forward path. This controller is then used in the feedback path of the inner loop in Figure 10.5.

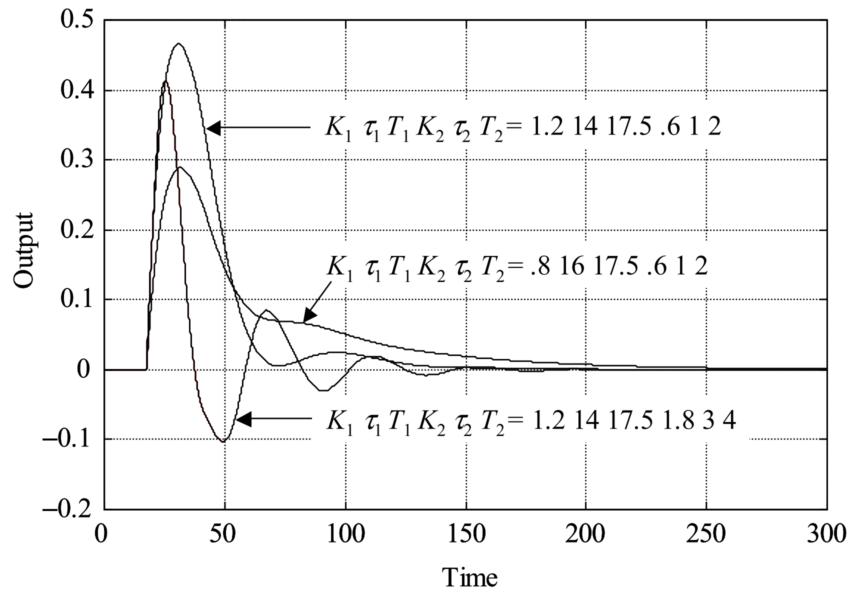

Figure 10.15 Responses to a step inner loop disturbance (d_2) with the outer loop closed.

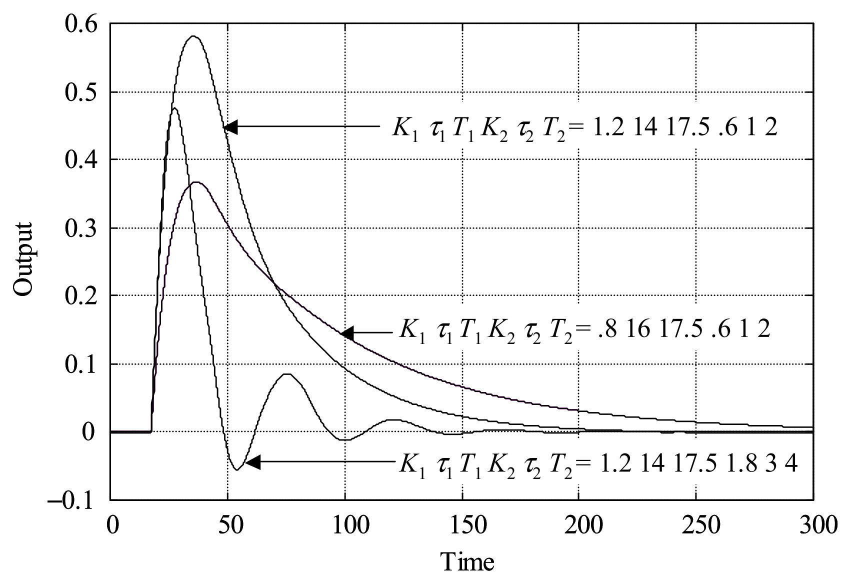

Figure 10.16 Responses to a step inner loop disturbance using the controller given by Eq. (10.8).

While the inner loop disturbance responses using the single-degree of freedom controller Eq. (10.8) are significantly slower than the 2DF controller given by Eq. (10.5a), the responses of the output $y_1(t)$ to setpoint changes to the outer loop are only slightly slower than those given in Figure 10.11.

(b) PID Cascade Controller Designs

Section 10.2 discusses methods for approximating the IMC cascade control system with the traditional cascade system of Figure 10.4. Figure 10.7 shows the IMC equivalent configuration. For convenience, this figure is repeated in Figure 10.17.

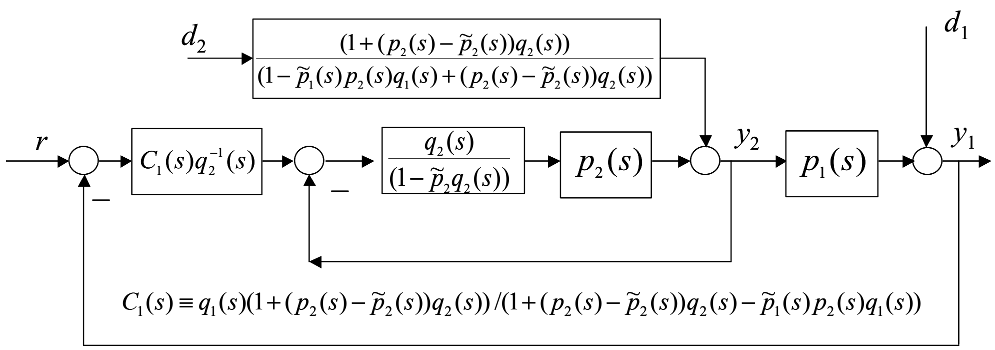

Figure 10.17 Standard feedback form of an IMC cascade control system.

Recall that the controller $C_1(s)$ in Figure 10.7 is not realizable because it contains terms involving the inner loop process $p_2(s)$, which varies within the uncertainty set and cannot be part of the controller. Therefore we suggested replacing $p_2(s)$ with its model, $\tilde{p}_2(s)$. This gives

$$C_1(s) \cong q_1(s)/(1-\tilde{p}_1(s)\tilde{p}_2(s)q_1(s)). \tag{10.9a}$$

IMCTUNE provides the following PID controllers from the IMC controllers obtained previously:

Inner loop:
$$\frac{q_2(s)}{(1-\tilde{p}_2q_2(s))} \cong PID_2 = 1.79(1+1/(12.05s)+1.68s)/(14.7s+1). \tag{10.9b}$$

Outer loop:
$$C_1(s) \cong PID_1 = .234(1+1/(23.77s)+5.35s/(.29s+1)), \tag{10.9c}$$

$$q_2^{-1} = 1.8(4.4s+1)^2/((s+1)(9.05s+1)). \tag{10.9d}$$

Figures 10.18 and 10.19 show the disturbance responses for the configuration of Figure 10.17 using the controllers given in equation sets (10.9) and (10.10). Notice that $q_2(s)$ in Eq. (10.9d) is from a 2DF design, and for this reason the responses are labeled Cascade 2.

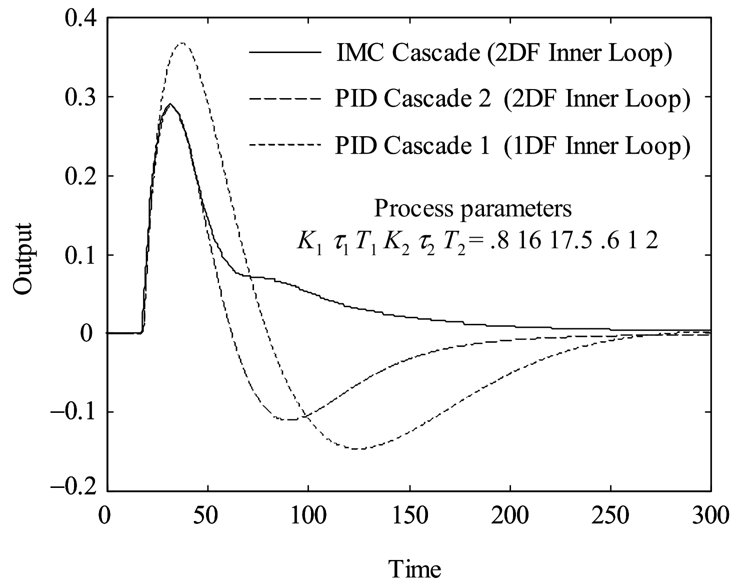

Figure 10.18 Comparison of responses to a step disturbance in the inner loop.

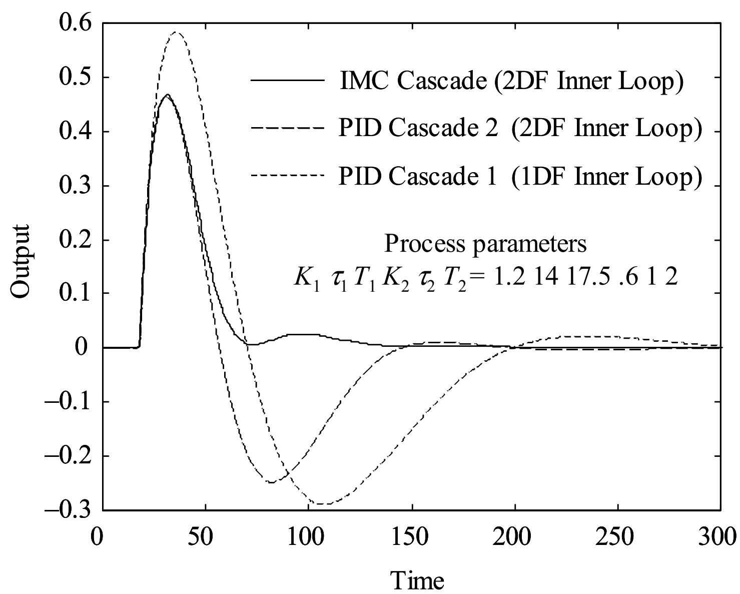

Figure 10.19 Comparison of responses to a step disturbance in the inner loop.

Using the 1DF IMC controller for q_2(s), given by Eq. (10.8) and repeated below, yields the inner loop PID controller given by Eq. (10.10a).

$$q_2(s) = \frac{(s+1)}{1.8(4.18s+1)}$$

Inner loop: $PID_2 = .134(1 + 1/(1.98s) + .319s/(.016s+1)).$ (10.10a)

The outer loop controller remains the same as in Eq. (10.9b) because $q_1(s)$ has not changed. In figures 10.18 and 10.19 the responses using Eq. (10.10a) are labeled Cascade 1. These responses show the benefits of an IMC outer loop over a PID outer loop. The outer loop PID controller in the responses in figures 10.18 and 10.19 is cascaded with the term q_2^{-1}. Since q_2^{-1} is a lead, it can be approximated by the Taylor series as the polynomial $1.8(-3.18s^2 + 3.18s + 1)$. Multiplying this polynomial into Eq. (10.10a) and dropping terms higher than second order gives, after some rearrangement,

Outer loop: $PID_1 = .4956\,(1 + 1/(26.96s) + 7.404s/(.37s+1)).$ (10.10b)

The PID controller given by Eq. (10.10b) can be used in place of equations (10.8) and (10.10a) and gives virtually identical results. The advantage of Eq. (10.10b) is that it can be used in the traditional cascade configuration of Figure 10.4. Of course, the disturbance response will be that of the 1DF cascade of figures 10.18 and 10.19.

♦

The purpose of the next example is to demonstrate the advantages of cascade control even when the dynamics of the secondary process are on the same order as the dynamics of the primary process. A common literature fallacy is that the dynamics of the secondary process have to be fast relative to those of the primary process in order to get improved performance from a cascade control system. This fallacy probably arose from the methods used for designing and/or tuning PID cascade control systems. Traditionally, the outer loop controller was designed and tuned assuming that the inner loop is so fast that it can be approximated as a unity gain. When this assumption is not true, the inner and outer loop designs and/or tunings interact, and there existed no good methods of designing and tuning the controller parameters that significantly improved performance over that of a single-loop controller. In an IMC cascade configuration (see Figure 10.5) the inner and outer loops interact mainly by the fact that the inner loop process gain variations are reduced by the action of the inner loop controller. Such interaction is desirable and, as we shall show, does not preclude arriving at controller designs so that cascade performance is significantly better than single-loop performance.

Example 10.2 Primary and Secondary Processes have Similar Dynamics

The following process was obtained by reducing the time constant and dead time of the primary process of Example 10.1 by a factor of five. This gives the following system:

$$p_1(s)=\frac{K_1\mathrm{e}^{-T_1 s}}{\tau_1 s+1};\ .8\le K_1\le 1.2,\quad 3.5\le T_1\le 4.5,\quad 2.8\le \tau_1\le 3.2. \qquad (10.11a)$$

$$p_2(s)=\frac{K_2 e^{-T_2 s}}{\tau_2 s+1};\ .6\le \mathrm{K}_2\le 1.8,\quad 2.0\le \mathrm{T}_2\le 4.0,\quad 1.0\le \tau_2\le 3.0. \qquad (10.11b)$$

(a) IMC System Design

Again following the suggestions in chapters 7 and 8, we use the upper-bound gains and dead times and the lower-bound time constants for the process models.

$$\tilde{p}_1(s)=\frac{1.2e^{-4.5s}}{2.8s+1}\ \text{and}\ \tilde{p}_2(s)=\frac{1.8e^{-4s}}{s+1}. \qquad (10.11c)$$

The controllers associated with the IMC cascade structure of Figure 10.5 are

$$q_2(s)=\frac{(s+1)}{1.8(2.8s+1)}\ \text{and}\ q_1(s)=\frac{(3.8s+1)}{2.16(5.24s+1)}. \qquad (10.11d)$$

An initial attempt at designing a 2DF controller for the inner loop resulted in the filter time constant reaching the primary process model time constant of 2.8 before achieving an Mp of 1.05 for the partial sensitivity function. In such a situation the inner loop feedback controller is chosen as a 1DF controller with the filter time constant tuned using the partial sensitivity function just as for a 2DF design. This controller is given by Eq. (10.11d). Equation (10.11d) also shows the outer loop controller that achieves an Mp of 1.05 for the complementary sensitivity function. Figures 10.20 and 10.21 show the disturbance and setpoint responses of the IMC cascade control system with models and controllers given by equations (10.11c) and (10.11d).

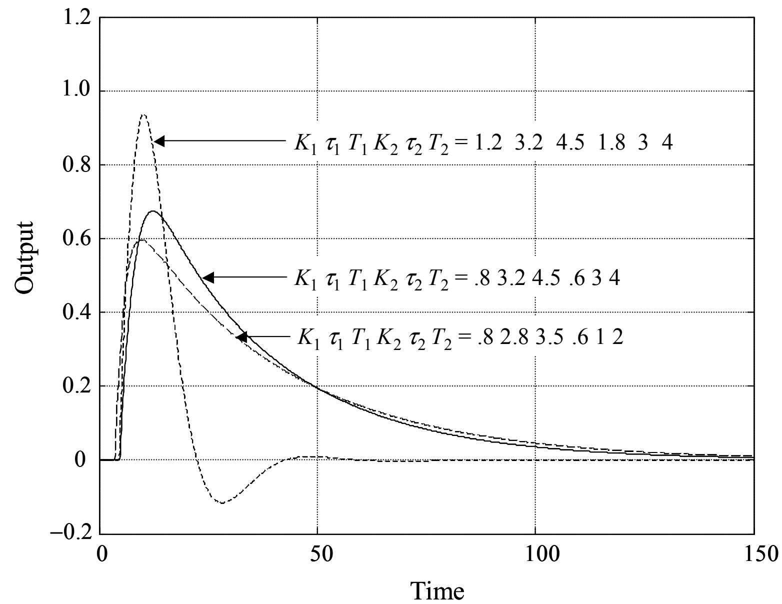

Figure 10.20 Unit step disturbance (d_2) responses for the IMC cascade control system.

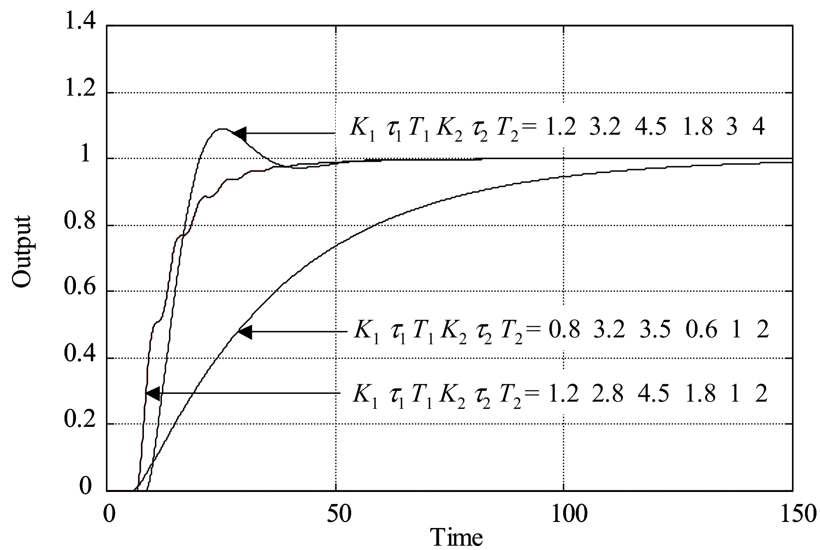

Figure 10.21 Unit step setpoint responses for the IMC cascade control system.

The responses in figures 10.20 and 10.21 should be compared with those of a well-tuned single-loop control system for the process given by equations (10.11a) and 10.11b) and rewritten as equations (10.12a) and (10.12b). Equations (10.12c) and (10.12d) give the associated model and controller:

$$p(s) = \frac{K_1 K_2 e^{-Ts}}{(\tau_1 s + 1)(\tau_2 s + 1)} \quad , \tag{10.12a}$$

where $.8 \le K_1 \le 1.2, .6 \le K_2 \le 1.8,\ 2.8 \le \tau_1 \le 3.2,\ 1 \le \tau_2 \le 3,\ 5.5 \le T \le 8.5$,

$$p_d(s) = \frac{K_1}{(\tau_1 s + 1)}, \tag{10.12b}$$

$$\tilde{p}(s) = \frac{2.16 e^{-8.5s}}{(3.8s + 1)}, \tag{10.12c}$$

$$q(s) = \frac{(3.8s + 1)}{2.16(6.31s + 1)}. \tag{10.12d}$$

Notice that Eq. (10.12b) ignores the disturbance deadtime since this term changes only the effective arrival time of the disturbance and so cannot be distinguished from the disturbance itself. Also, the model given by Eq. (10.12c) approximates the process lags as a first-order system whose time constant is the sum of the time constants of the two first order process lags. Finally, the controller given by Eq. (10.12d) is a 1DF controller because we are using a single loop controller in spite of the fact that the disturbance, d_2, enters into the primary output through the lag given by Eq. (10.12b).

The single-loop responses given in figures 10.22 and 10.23 are roughly twice as slow as those of the cascade control system shown in Figures 10.20 and 10.21. Notice that the time scales in figures 10.22 and 10.23 are from 0 to 300 whereas the time scales in figures 10.20 and 10.21 are from 0 to 150. Also, the disturbance peak heights in Figure 10.22 are higher than those of the cascade control system in Figure 10.20.

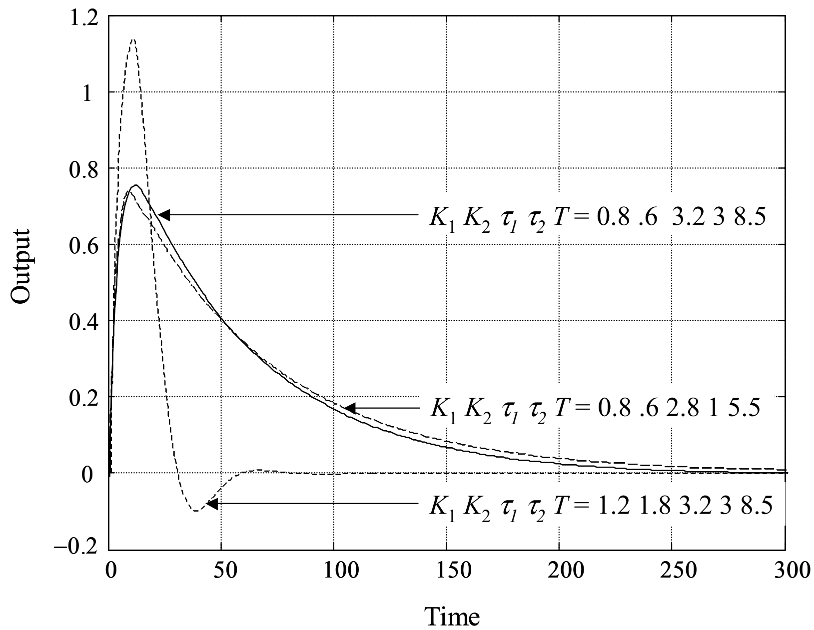

Figure 10.22 Single-loop control system of Eq. (10.12) responses to a step disturbance in d_2.

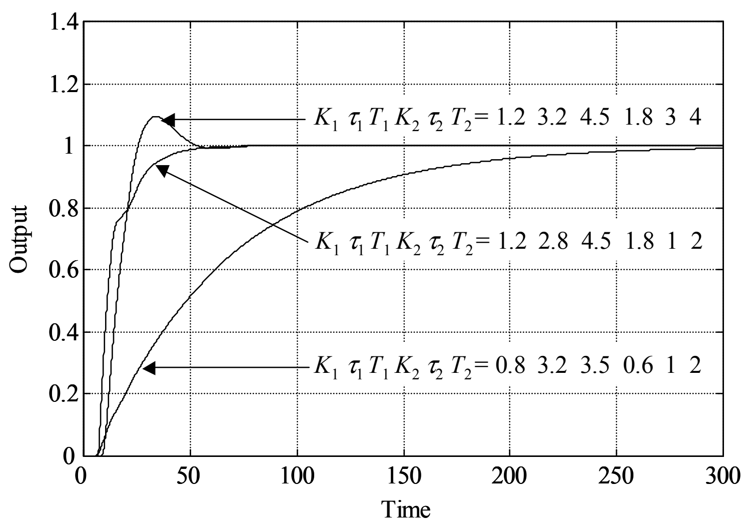

Figure 10.23 Step setpoint responses for the single-loop control system is given by Eq. (10.12).

(b) PID Cascade Controller Designs

Replacing the IMC inner loop with a feedback controller $q_2(s)/(1-\tilde{p}_2 q_2(s))$, as in Figure 10.6, and then approximating the feedback controller with the PID controller given by Eq. (10.13a) does not change the setpoint and disturbance responses of figures 10.20 and 10.21. That is, there is no degradation of the performance of the mixed IMC-PID cascade control system.

$$PID_2(s) = .178(1 + 1/2.18s + .456s/(0.0228s + 1)). \tag{10.13a}$$

Approximating the controller C_1 by Eq. (10.3), multiplying it by the Maclaurin series approximation to $q_2^{-1}(s)$, and finally approximating the term $C_1 q_2^{-1}(s)$ as a PID controller, as in Example 10.1, gives

$$PID_1(s) = .485(1 + 1/8.0s + 2.45s/(.122s + 1)). \tag{10.13b}$$

Figure 10.24 shows the inner loop disturbance d_2 response for the traditional cascade configuration of Figure 10.4, using the PID controllers given by equations (10.13a and (10.13b). The response for a process with upper-bound parameters is too oscillatory. The step setpoint response for the same process shows a 21% overshoot. The reason for this poorer behavior is probably the interaction between the inner and outer loops.

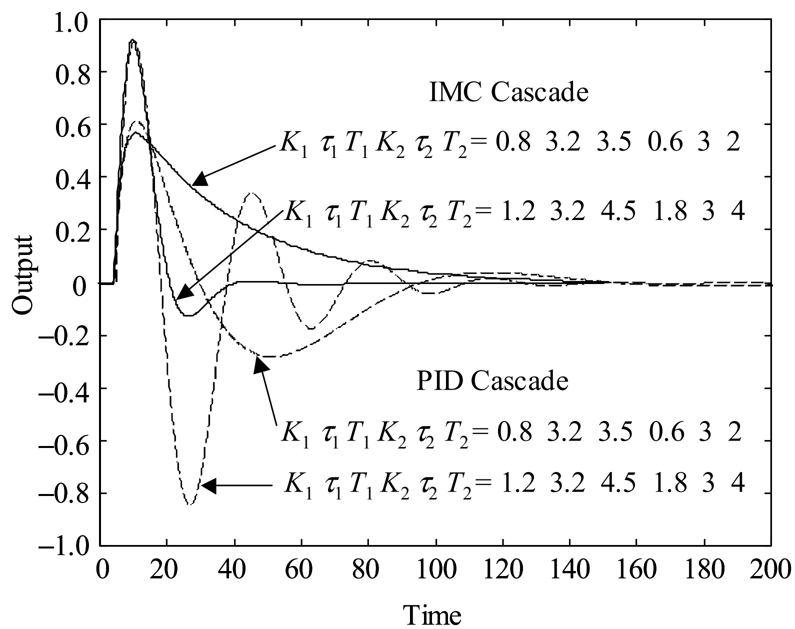

Figure 10.24 Comparison of responses to a step disturbance in the inner loop.

Conceptually, it is possible to extend the cascade feature of IMCTUNE to accommodate the PID cascade configuration of Figure 10.4 and to automatically increase the filter time constant of $q_1(s)$ to tune the outer loop to give a specified Mp. After tuning, IMCTUNE, or any program like it, should be able to provide the PID approximation to the term $C_1\, q_2^{-1}(s)$. Unfortunately, such an extension does not yet exist, and the only method that we can suggest to improve the responses in Figure 10.24 is a rather tedious trial-and-error method wherein one increases the filter time constant of $q_1(s)$, re-computes PID_1, and then checks the responses of the processes with the upper-bound parameters. This assumes that the worst-case responses will always be those for the upper-bound parameters, which of course may not always be true.

♦

10.3 SATURATION COMPENSATION

10.3.1 IMC Cascade

Figure 10.5 provides the simplest starting point for a discussion of control effort saturation in cascade control systems. For convenience, this figure is reproduced in Figure 10.25.

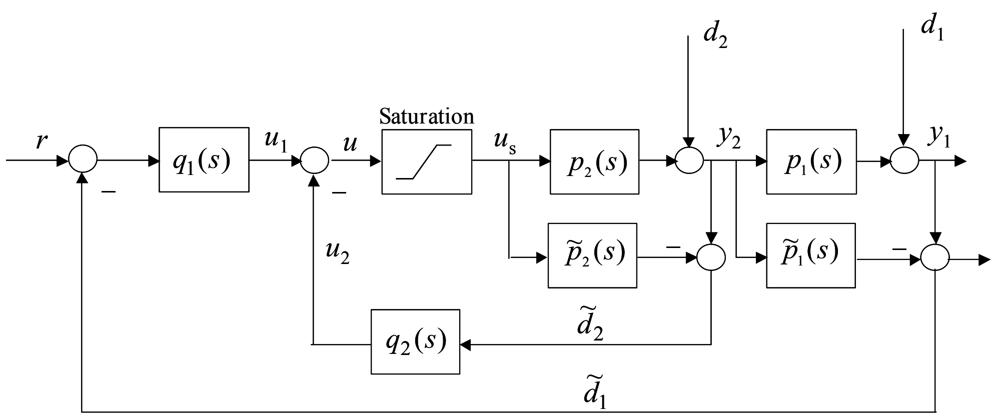

Figure 10.25 IMC cascade structure.

The effect of control effort saturation on the inner loop of Figure 10.25 can be minimized by implementing the inner loop as a model state feedback (MSF) system, as shown in Figure 10.26a.

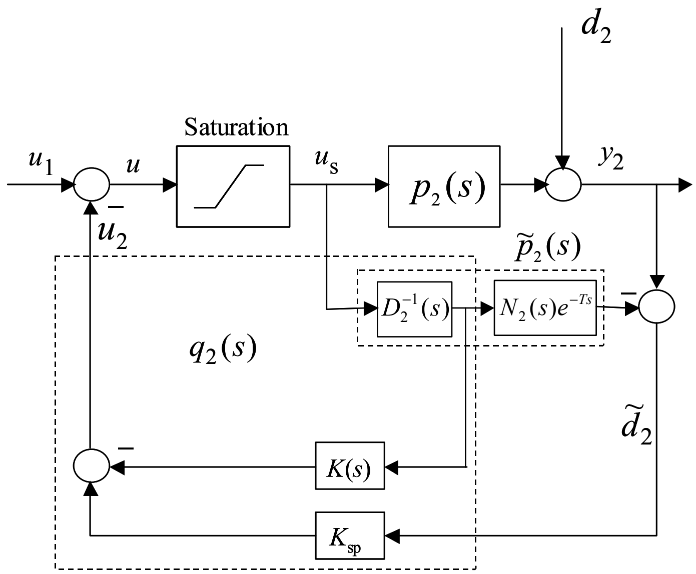

Figure 10.26a An MSF implementation of the inner loop of Figure 10.25.

The only difference between Figure 10.26a and Figure 5.5 of Chapter 5 is that there is no setpoint in Figure 10.26a.

Saturation compensation for the outer loop of Figure 10.25 is not quite so straightforward. One problem is that the outer loop controller is designed to invert portions of both inner loop and outer loop models (i.e., $\tilde{p}_1(s)\tilde{p}_2(s)$). However, there is no such transfer function, since the output of the inner loop model is *not* the input to the outer loop model. One solution is to create a new transfer function, $1/D_1(s)$, where $D_1(s)$ contains the denominator of the transfer function that the controller $q_1(s)$ inverts. Figure 10.26b shows an MSF implementation of the outer loop controller, $q_1(s)$, using this approach. This figure includes inner loop control system of Figure 10.26a, as it is necessary to remove the inner loop control effort u_2 from the signal used to compute the feedback portion of the outer loop control effort u_1.

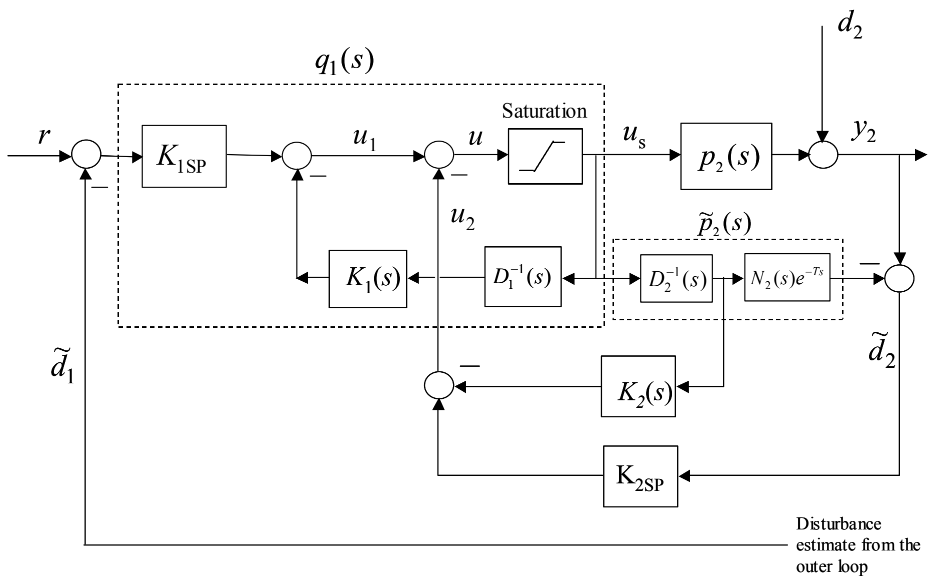

Figure 10.26b MSF implementation of both inner loop and outer loop controllers of Figure 10.25.

10.3.2 IMC/PID Cascade

In the absence of saturation there is usually little performance loss if the inner loop of the IMC cascade structure is replaced by a simple feedback loop, as shown in Figure 10.6. However, the method used in Figure 10.26b to compensate the outer loop for control effort saturation does not readily carry over to the outer loop of the cascade structure in Figure 10.6. The problem is that in the structure of Figure 10.6, there is no explicit calculation of an inner loop control effort, u_2, as there is in figures 10.5 and 10.26b. For this reason, we recommend implementing the outer loop as shown in Figure 10.27. The limits of saturation block in this figure would ideally be set to the limits of the actual control effort less the contribution of the inner loop control effort u_2 to the total control effort. However, since u_2 is not available without additional calculations, we recommend simply setting the limits to those of the actual control effort. This is, of course, equivalent to assuming that u_2 is zero. Notice that the saturation block in Figure 10.27 is not on the outer loop control effort u_1 but rather only on the input to the inverse of the numerator of q_1(s), which is D_1(s). The reason is that the role of the structure in Figure 10.27 is only to attempt to compensate for saturation in the inner loop, and not to limit the setpoint sent to the inner loop. Finally, we recommend replacing the IMC controller, $q_2(s)/(1-\tilde{p}_2 q_2(s))$, in Figure 10.6 with a standard anti-reset windup PID controller, as described in Chapter 6.

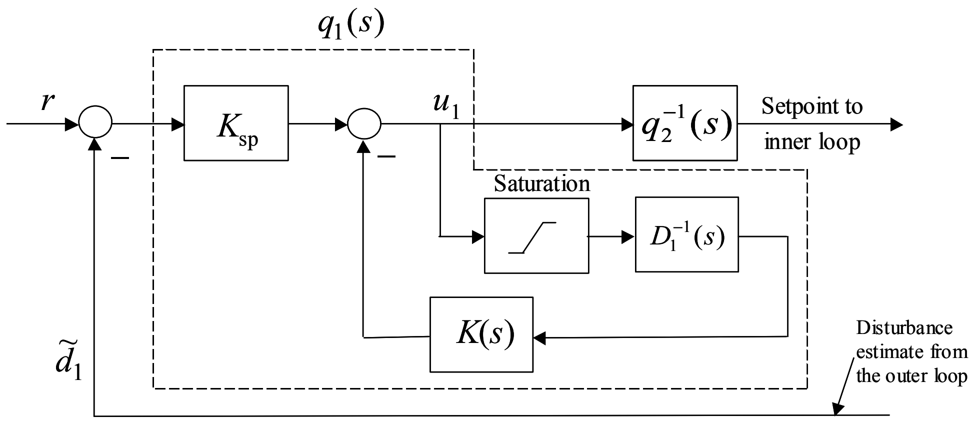

Figure 10.27 Compensating the outer loop of Figure 10.6 for control effort saturation.

10.3.3 PID Cascade

Saturation compensation for the standard PID cascade structure of Figure 10.4 is usually accomplished by either of two methods. The preferred method is to use logic statements that stop the integration in both the inner loop and outer loop PID controllers whenever the control effort reaches a limit, and start it again whenever the error signals change sign or the control effort comes off saturation. The second, and possibly more common, method is to use a standard anti-reset windup controller in the inner loop and implement the integral portion in the outer loop PID controller, as shown in Figure 10.28.

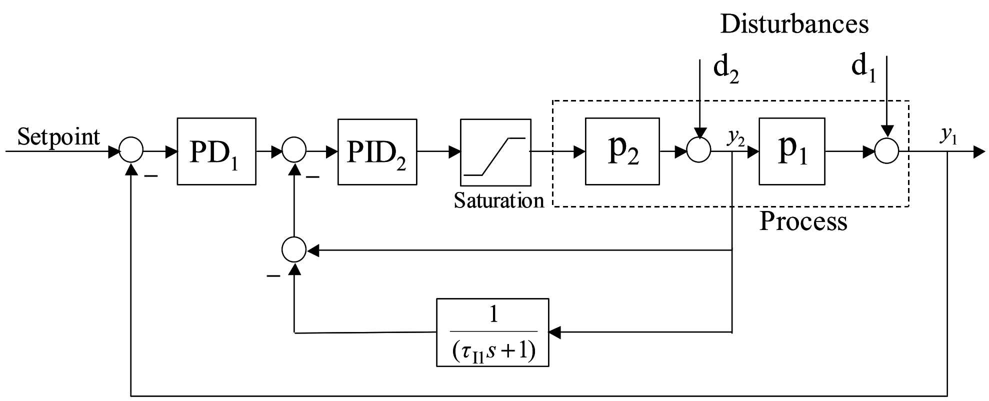

Figure 10.28 Feedback method of outer loop anti-reset windup for PID cascade.

The difficulty with the implementation of Figure 10.28 is that is the outer loop integral time constant is not τ_{I1}, but rather a complicated function of τ_{I1} and the parameters of the inner loop transfer function. This complication can make it inadvisable to use the IMC-generated outer loop PID parameters developed in this section. The first method, which simply stops the integration on control effort saturation, does not have the foregoing drawback.

10.4 SUMMARY

To achieve the best disturbance rejection and setpoint tracking the inner loop of the cascade should be designed and tuned as a 2DF controller. The outer loop process lag plays the role of the disturbance lag in the controller design. The outer loop should be implemented as an MSF IMC system. The inner loop can be implemented as either a PID control system or, in the case of very little process uncertainty, in IMC MSF form.

There is no need for the inner loop process to be faster than the outer loop process in order for a well-designed cascade control system to provide significant performance advantages over a single-loop control system.

The techniques of this chapter can often be used to obtain the PID parameters for the traditional cascade structure. However, anti-reset windup for the outer loop should be implemented by stopping integration when the control effort saturates in order to use the calculated integral time constant. If the anti-reset windup for the outer loop is implemented via a lag around the inner loop, then the lag time constant is not necessarily the same as the computed integral time constant. Outer loop setpoint tracking and disturbance rejection is generally better than that achievable with a single-loop control system because the inner loop serves to reduce the apparent gain uncertainty of the inner loop process.

Problems

Design and tune cascade control systems for each of the following processes. The primary output is y, and the measured secondary output is y_2. Also, compare the performance your cascade control system with that of the feedforward control systems found for the problems of Chapter 9. The problems in Chapter 9 are the same as those below except that all the disturbances were considered to be measured whereas now only the process outputs $y(s)$ and $y_2(s)$ are measured.

10.1 $y(s) = \dfrac{e^{-10s}}{30s + 1} y_2(s) + d_1(s) \quad y_2(s) = \dfrac{e^{-5s}}{4s + 1} u(s) + d_2(s)$

10.2 $y(s) = \frac{K_1 e^{-Ts}}{5s+1} y_2(s) + d_1(s) \qquad 1 \le K_1 \le 3, \quad 4 \le T \le 6$

$y_2(s) = \frac{K_2 e^{-s}}{3s+1} u(s) + d_2(s) \qquad 1 \le K_2 \le 5, \quad 0 \le u(t) \le 10$

10.3 $y(s) = \frac{K_1 e^{-Ts}}{3s+1} y_2(s) + d(s) \quad 1 \le K_1 \le 5, \quad 2 \le T \le 4$

$y_2(s) = \frac{K_2 e^{-s}}{2s+1} u(s) + d_2 \quad 1 \le K_2 \le 5$

10.4 $y(s) = \frac{K_1 e^{-Ts}}{3s+1} y_1(s) + \frac{e^{-Ts} d_1(s)}{3s+1} \qquad 1 \le K \le 5, \quad 2 \le T \le 4$

$y_1(s) = \frac{K_2(-2s+1)}{(2s+1)^2} u(s) + \frac{d_2}{s+1}$

References

Morari, M., and E. Zafiriou, E. 1989. *Robust Process Control*. Prentice Hall, NJ.

Seborg, D. E., T. F. Edgar, and D. A. Mellichamp. 1989. *Process Dynamics and Control.* John Wiley & Sons, NY.