8

Predictive Control

8.1 Introduction

A PI controller only considers present and past data, and a PID controller also predicts the future process behavior by linear extrapolation. There have been many attempts to find other ways of predicting future process behavior and to take this into account when making the control actions. Good predictions can improve controller performance, particularly when the process has time delays, which are common in process control. Time delays can arise from a pure delay mechanism caused by transport or time for computation and communication. Delays may also be caused by measurements obtained by off-line analysis. They also appear when a high-order system or a partial differential equation is approximated with a low-order model as in heat conduction. Time delays appear in many of the models discussed in this book. A new controller that could deal with processes having long time delays was proposed by Smith in 1957. The controller is now commonly known as the Smith predictor. It can be viewed as a new type of controller but it can also be interpreted as an augmentation of a PID controller. There are also many other controllers that have predictive abilities. The model predictive controller is a large class of controller that is becoming increasingly popular.

In this chapter we start by presenting the Smith predictor in Section 8.2. This controller can give significant improvements in the response to set-point changes, but the Smith predictor can also be very sensitive to model uncertainties. This is shown in Section 8.3 where we analyze the closed-loop system when a Smith predictor is used. The analysis also shows that the concepts of gain and phase margin are not sufficient to characterize the robustness of the system. The reason for this is that the Nyquist curve of the loop transfer function can have large loops at frequencies larger than the gain crossover frequency. The robustness is well captured by the properties of the *Gang of Four*, and there is also another classical robustness measure, the delay margin, that gives good insight. A special type of the Smith predictor called the PPI controller is discussed in Section 8.4. This controller is simpler and more robust. Model predictive control, a more general form of prediction that is gaining in popularity, is discussed in Section 8.6.

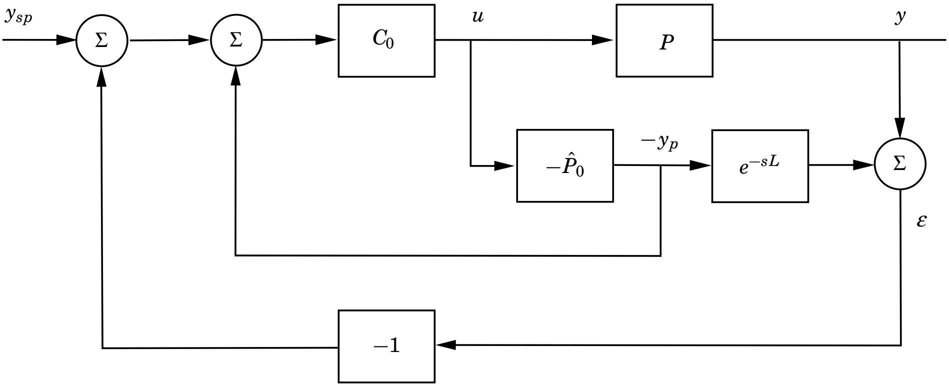

Figure 8.1 Block diagram of a system with a Smith predictor.

8.2 The Smith Predictor

To describe the idea of a Smith predictor we consider a process with a time delay L, and we factor the process transfer function as

$$P(s) = P_0(s)e^{-sL}, \tag{8.1}$$

where the transfer function P_0 does not have any time delays. Figure 8.1 shows a block diagram of a closed-loop system with a Smith predictor. The controller consists of an ordinary PI or PID controller C_0 and a model of the process $\hat{P}$, factored in the same way as the process, connected in parallel with the process. If the model is identical with the process the signal y_p represents the output without the delay or, equivalently, a prediction of what the output would be if there were no delays. By using the model it is thus possible to generate a prediction of the output. The signal y_p is fed back to the controller, and there is also an additional feedback from the process output y to cope with load disturbances. If the model $\hat{P}$ is identical to the process P and if there are no disturbances acting on the process the signal ε is zero. This means that the outer feedback loop gives no contribution, and the input-output relation of the system is given by

$$G_{yy_{sp}} = \frac{PC_0}{1 + P_0C_0} = \frac{P_0C_0}{1 + P_0C_0}e^{-sL}. \tag{8.2}$$

The controller C_0 can thus be designed as if the process has no time delay, and the response of the closed-loop system will simply have an additional time delay.

The system shown in Figure 8.1 can also be represented by the block diagram in Figure 8.2, which is an ordinary feedback loop with a process P and a controller C, where the controller has the transfer function

$$C = \frac{C_0}{1 + C_0(\hat{P}_0 - \hat{P})} = \frac{C_0}{1 + C_0\hat{P}_0(1 - e^{-sL})}. \tag{8.3}$$

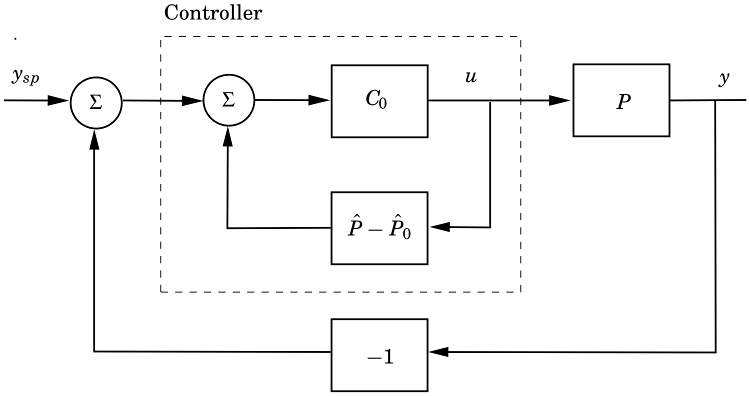

Figure 8.2 Another representation of a system with a Smith predictor.

The transfer function $\hat{P}_0e^{-sL}$ is the transfer function of the process model used to design the controller. The controller C is thus obtained by wrapping a feedback around the controller C_0. The input-output relation of the controller C can be written as

$$U(s) = C_0(s)\big(E(s) - \hat{P}_0(s)(1-e^{-sL})U(s)\big), \tag{8.4}$$

where $U(s)$ and $E(s)$ are the Laplace transforms of the control signal and the error. The term $\hat{P}_0(s)(1-e^{-sL})U(s)$ can be interpreted physically as the predicted effect on the output of control signals in the interval $(t-L, t)$. The Smith predictor can thus be interpreted as an ordinary PI controller where the effects of past control actions are subtracted from the error. The controller can be compared with a PID controller, which predicts by extrapolating the current process output linearly, as is illustrated in Figure 3.5. This type of prediction is less effective for systems with time delays because future process outputs are strongly influenced by past control control actions rather than current inputs.

The properties of the Smith predictor will be illustrated by an example.

EXAMPLE 8.1—FIRST-ORDER SYSTEM WITH TIME DELAY
Consider a process with transfer function

$$P(s) = \frac{K_p}{1+sT}e^{-sL}. \tag{8.5}$$

A PI controller that gives the characteristic polynomial

$$s^2 + 2\zeta\omega_0 s + \omega_0^2$$

for the process without delay is designed as described in Section 6.4. The controller is

$$C_0(s) = K\Big(1 + \frac{1}{sT_i}\Big),$$

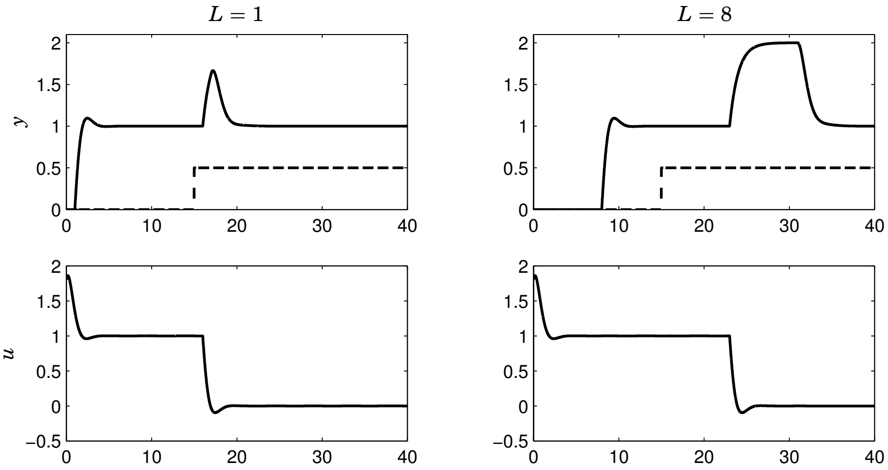

Figure 8.3 Responses of a closed-loop system with Smith predictor. The process has the transfer function $P(s) = e^{-sL}/(s+1)$, and the figure shows response for $L = 1$ and 8. The dashed line is the load disturbance.

where

$$\begin{aligned} K &= \frac{2\zeta\omega_0 T - 1}{K_p} \\ T_i &= \frac{K_p K}{\omega_0^2 T}. \end{aligned} \tag{8.6}$$

Figure 8.3 shows the responses of the system to a unit step change in the set point and a load disturbance in the form of a unit step in the process input. The load disturbance is applied at time $t = 15$ in all cases. The time constant is equal to one in all cases, and the time delay L is changed. The PI controller is designed to give a closed-loop system with $\omega_0 = 2$ and $\zeta = 0.7$ for the process without delays. The figure shows that the responses to set point have the same shape but with a delay that changes with the process delay. The shape is the same as for a system without the time delay. This property of the system is quite remarkable.

The shapes of the responses to load disturbances change with the time delay L. With increasing time delay it will take a longer time for the system to react. The initial part of the responses are similar but with different delays. Because of the varying delay the time to recover from the disturbance varies with the time delay. □

Analyzing the results it may appear remarkable that it is possible to obtain such good responses even when the time delay is as long as $L = 8$. In the following we will analyze the systems obtained when using the Smith's predictor to better understand its behavior.

The Predictor

It follows from (8.3) that the Smith predictor can be viewed as the cascade connection of an ordinary controller C_0 and a block with the transfer function

$$C_{pred} = \frac{1}{1 + C_0(\hat{P}_0 - \hat{P})} = \frac{1}{1 + C_0\hat{P}_0(1 - e^{-sL})}. \tag{8.7}$$

To obtain the responses shown in Figure 8.3 the transfer function C_{pred} compensates for the time delay of the process. Intuitively this can be understood in the following way. Assume $C_0\hat{P}_0 \approx -1$; it then follows from (8.7) that

$$C_{pred} \approx e^{sL}.$$

This means that the transfer function $C_{pred}(s)$ acts like an ideal predictor. We can therefore expect that the transfer function C_{pred} behaves like an ideal predictor for frequencies where $C_0(i\omega)\hat{P}_0(i\omega)$ is close to -1. Notice that it is not possible to have $C_0(i\omega)\hat{P}_0(i\omega) = -1$ for any frequency because the transfer function (8.2) is then unstable. The properties of the transfer function (8.7) will be illustrated by an example.

EXAMPLE 8.2—PREDICTOR FOR FIRST-ORDER SYSTEM WITH TIME DELAY
Consider the same system as in Example 8.1. Assuming that there are no modeling errors it follows that $\hat{P} = P = P_0e^{-sL}$. Combined with a PI controller the predictor becomes

$$C_{pred} = \frac{1}{1 + C_0(P_0 - P)} = \frac{1}{1 + \dfrac{K_pK(1 + sT_i)}{sT_i(1 + sT)}(1 - e^{-sL})}. \tag{8.8}$$

It follows from (8.8) that $C_{pred}(i\omega) = 1$ for $\omega L = 2\pi, 4\pi, 6\pi, \ldots$ and that $C_{pred}(s)$ goes to 1 for large s. The transfer function C_{pred} has the series expansion

$$C_{pred}(s) = \frac{T_i}{T_i + K_pKL}\Big(1 + \frac{K_pKL}{T_i + K_pKL}(T + \frac{L}{2} - T_i)s + \ldots\Big).$$

The static gain of C_{pred} decreases with increasing L and is always less than one. Figure 8.4 shows the Bode plot for the transfer function for $L = 8$. The figure shows that the transfer function gives a very large phase advance, more than 800°. A comparison with the phase curve of an ideal predictor shows that the system does approximate an ideal predictor well for certain frequencies. The solid and dashed curves are very close for those frequencies where the gain curve has peaks. Notice, however, that the gain curves are different. The ideal predictor has constant gain, but the gain of the transfer function C_{pred} changes with several orders of magnitude.

We will now investigate how the large phase advance is created. Figure 8.5 shows Nyquist curves of the transfer function C_{pred} for $K_p = 1$, $T = 1$, $K = 1.8$, $T_i = 0.45$, and $L = 1$, 2.5, 4, and 8. For $L = 1$ the largest phase advance is close to 90°. The phase advance increases with increasing L, as is indicated in the curve for $L = 2.5$ where the circular part of the Nyquist curve increases. The Nyquist curve goes to infinity for $L = 2.99$, which indicates that the

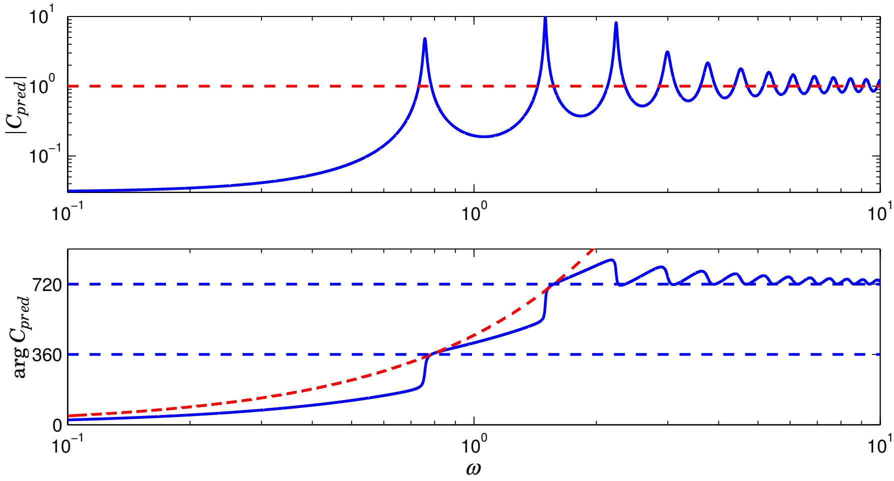

Figure 8.4 Bode plot of the loop transfer functions $C_{pred}(s)$ given by (8.8) for $L = 8$ (solid) and for the ideal predictor e^{sL} (dashed).

transfer function has poles on the imaginary axis. For larger L the Nyquist curve encircles the origin, which means that the phase advance is more than 360°. The curve for $L = 4$ shows that the largest phase advance is more than 450°. As L is increased further the Nyquist curve again goes to infinity for $L = 6.40$, and for larger L there are two encirclements of the origin, indicating that the phase advance is more than 720°. The curve for $L = 8$ shows that the largest phase advance is more than 800°.

To deform the curve for $L = 2.5$ continuously to the curve for $L = 4$ in Figure 8.5 the curve must go to infinity for some intermediate value of L. In the particular case the Nyquist curve of C_{pred} goes to infinity for $L = 2.99$, 6.40, 9.80, 13.40, 17,00, 20.6, ... This means that the transfer function C_{pred} is unstable for some values of L. It has two poles in the right-half plane for $2.99 < L < 6.40$, four poles in the right half plane for $6.40 < L < 9.80$, etc. For the simulation with $L = 10$ in Figure 8.3 the predictor transfer function has six poles in the right half plane. The predictor (8.3) thus achieves very large phase advances through poles in the right half plane. □

There are severe drawbacks with unstable controllers. It follows from Bode's integral (4.28) that poles in the right half plane increase the sensitivity. The remarkable response to set-point changes shown in Figure 8.3 thus comes at a price. Some of these issues will be discussed in the next section.

8.3 Analysis of Smith Predictor Control

The closed-loop system obtained when a process is controlled using a Smith predictor will now be investigated. Let the process transfer function be P, the

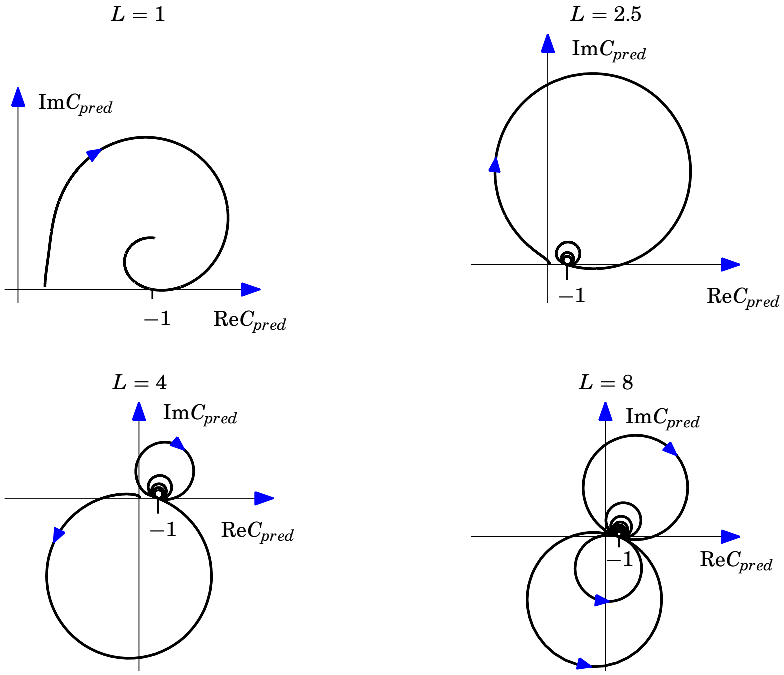

Figure 8.5 Nyquist plots of the transfer function C_{pred} for the system in Example 8.2 with $L = 1$, 2.5, 4, and 8. The plot for $L = 1$ and 2.5 all circles clockwise, the plot for $L = 4$ first makes one counterclockwise loop before making the clockwise loops, the plot for $L = 8$ first makes tow counterclockwise the remaining loops are clockwise.

transfer function of the Smith predictor (8.3), and we find

$$\begin{aligned}
G_{yy_{sp}} &= \frac{PC}{1+PC} = \frac{PC_0}{1+\hat{P}_0C_0+(P-\hat{P})C_0} = \frac{P_0C_0}{1+P_0C_0}e^{-sL}\\
G_{yd} &= \frac{P}{1+PC} = \frac{P\big(1+(\hat{P}_0-\hat{P})C_0\big)}{1+\hat{P}_0C_0+(P-\hat{P})C_0} = P\Big(1-\frac{P_0C_0}{1+P_0C_0}e^{-sL}\Big)\\
-G_{un} &= \frac{C}{1+PC} = \frac{C_0}{1+\hat{P}_0C_0+(P-\hat{P})C_0} = \frac{C_0}{1+P_0C_0}\\
-G_{yn} &= \frac{1}{1+PC} = \frac{1+(\hat{P}_0-P)C_0}{1+\hat{P}_0C_0+(P-\hat{P})C_0} = 1-\frac{P_0C_0}{1+P_0C_0}e^{-sL},
\end{aligned} \tag{8.9}$$

where the last equality is obtained by assuming that the model is perfect, i.e., $\hat{P} = P$. The form of the transfer function from set point to process output $G_{yy_{sp}}$ shows that apart from the time delay the set-point responses are the same as for the system without time delays. The transfer G_{un} from measurement noise to the control signal is the same as the transfer function from set point to controller output. This transfer function is the same as for a system without a delay.

Stability

It follows from (8.9) that the closed-loop system has poles at the open-loop process poles and at the zeros of of the function

$$1+\hat{P}_0C_0+(P-\hat{P})C_0 \approx 1+\hat{P}_0C_0,$$

where the approximation is valid when $\hat{P} \approx P$. The zeros of this function can be chosen to be stable by a proper controller C_0. To have a stable closed-loop system it must also be required that the process be stable. This means that the Smith predictor does not work for processes with unstable open-loop dynamics. Modifications to eliminate this difficulty will be given in Section 8.5.

Response to Load Disturbances

When modeling errors are neglected the response to a load disturbance at the process input is given by the transfer function

$$G_{yd}=P\left(1-\frac{P_0C_0}{1+P_0C_0}e^{-sL}\right);$$

see (8.9). The second term has a time delay L. If a disturbance occurs at time 0 it follows that the response in the interval $0 \le t < L$ is the same as the response of the open-loop system. A typical illustration is given in Figure 8.3.

Assume that the process P is stable with static gain K_p and that controller C_0 has integral action with integral gain k_i. A series expansion of G_{yd} for small s gives

$$G_{yd}(s) \approx K_p\left(1-\frac{K_pk_i}{s+K_pk_i}(1-Ls)\right)=K_p\frac{s+K_pk_iLs}{s+K_pk_i} \approx \left(K_pL+\frac{1}{k_i}\right)s. \tag{8.10}$$

Since $G_{yd}(0)=0$ there is no steady-state error for a step change in the load disturbance. Furthermore, the integrated error for a load disturbance in the form of a unit step is

$$IE=K_pL+\frac{1}{k_i}. \tag{8.11}$$

Notice that the first term K_pL only depends on the process and that the second term $1/k_i$ only depends on the controller.

The transfer function P has a pole at the origin for processes that have integral action. For such processes and a controller with integral action we have $P(s) \approx K_v/s$ and $C(s) \approx k_i/s$ for small values of s. This implies that

$$G_{yd}(s) \approx \frac{K_v}{s}\left(1-\frac{K_vk_i}{s^2+K_vk_i}(1-Ls)\right)=\frac{K_v}{s}\frac{s^2+K_vk_iLs}{s^2+K_vk_i} \approx K_vL. \tag{8.12}$$

This means that there will be a steady-state error for processes with integration even if the controller has integral action. The recovery from load disturbances will therefore be very slow for processes with slow dynamics. Notice that the closed-loop system is stable even though P contains an integrator. The reason is that the integrator of P is canceled with a zero of the transfer function $1-PC_0/(1+P_0C_0)$. Several modifications of the Smith predictor have been proposed for processes with integration. This will be discussed in Section 8.5.

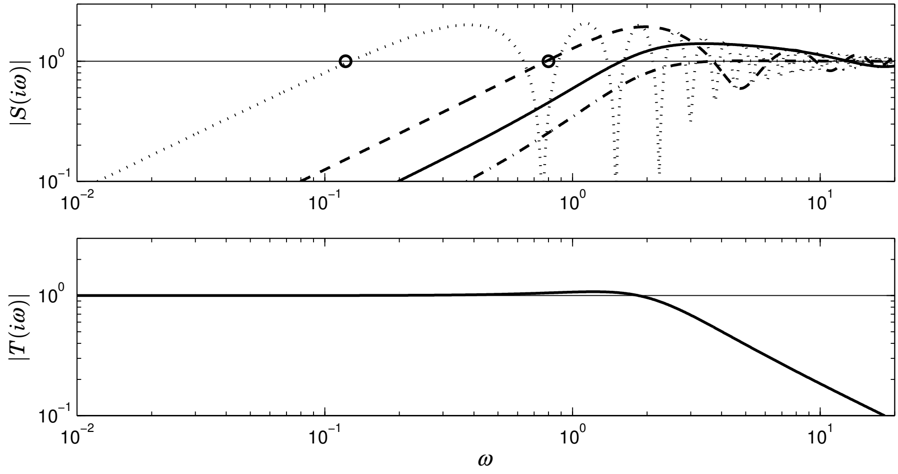

Figure 8.6 Gain curves for the sensitivity functions for the system in Example 8.3 with $L = 0$ (dash-dotted), 0.25 (solid), 1 (dashed), and 8 (dotted).

The Sensitivity Functions

In the ideal case $\hat{P} = P$, it follows from (8.9) that the sensitivity and the complementary sensitivity functions are

$$\begin{aligned} S &= 1 - \frac{PC_0}{1+P_0C_0} = 1 - \frac{P_0C_0}{1+P_0C_0}e^{-sL} = 1 - T_0e^{-sL} \\ T &= \frac{PC_0}{1+P_0C_0} = \frac{P_0C_0}{1+P_0C_0}e^{-sL} = T_0e^{-sL}, \end{aligned} \tag{8.13}$$

where T_0 is the complementary sensitivity function for the nominal system without delay. Notice that the gain curves of T and T_0 are identical. The gain curve of the complementary sensitivity function is independent of L.

EXAMPLE 8.3—SYSTEM OF FIRST ORDER WITH TIME DELAY

For the first-order system in Example 8.1 where the controller C_0 was designed to give $\omega_0 = 2$. The sensitivity functions are

$$\begin{aligned} T(s) &= \frac{K_pK(1+sT_i)}{sT_i(1+sT)+K_pK(1+sT_i)}e^{-sL} = \frac{s\omega_0^2T/(2\zeta\omega_0T-1)+\omega_0^2}{s^2+2\zeta\omega_0s+\omega_0^2}e^{-sL} \\ S(s) &= 1 - T(s). \end{aligned}$$

Figure 8.6 shows the gain curves of the sensitivity functions for $L = 0$, 0.25, 1, and 8, which corresponds to $\omega_0 L = 0$, 0.5, 2, and 16. The largest sensitivity increases rapidly with L; we have $M_s = 1.1$, 1.4 1.6, and 2 for $L = 0$, 0.24, 0.4, and 1.2, respectively. For $L > 1.2$ the maximum sensitivity remains close to $M_s = 2$. Also notice that the sensitivity for low frequencies increases rapidly with increasing L.

The differences between the low-frequency properties of the sensitivity functions in Figure 8.6 are easily explained from (8.10). The low-frequency asymptote of the gain curves of the sensitivity function intersects the unit magnitude line for $\omega = k_i/(1 + k_i K_p L)$. For the system in the figure we have $K_p = 1$ and $k_i = 4$, and the intersections are denoted by circles in Figure 8.6. □

The sensitivity functions shown in Figure 8.6 are typical for systems with Smith predictors. The complementary sensitivity function is close to one for frequencies up to the bandwidth ω_b of the nominal system without time delay. The sensitivity function has the typical oscillatory behavior shown in the figure. It intersects the line $|S| = 1$ several times. For large delays the sensitivity crossover frequency is approximately $\omega_{sc} = k_i/(1 + k_i K_p L)$, reflecting the fact that the attenuation of load disturbances is poor for large L. Also notice that the largest peaks of the sensitivity function are close to $M_s = 2$ in the frequency range where $|T(i\omega)| \approx 1$.

Robustness

For controllers with integral action we have $T(0) = 1$. Let ω_b be a frequency such that $|T(i\omega)|$ is close to 1 for $0 \le \omega \le \omega_b$. If $\omega_b L \ge \pi$ it then follows from (8.13) that the maximum sensitivity is around $M_s = 2$. In order to have smaller sensitivities it is therefore necessary to require that $\omega_b L$ is not too large. It follows from (4.32) that it is possible to have perturbations in the process such that

$$\frac{|\Delta P(i\omega)|}{|P(i\omega)|} < \frac{1}{|T(i\omega)|}$$

without making the system unstable. For frequencies less than ω_b the right-hand side is equal to one. The inequality then implies that the uncertainty region is a circle with center at $P(i\omega)$ that passes through the origin. If we only consider variations in the phase admissible variations are therefore 60° or $\pi/3$ rad. Since the phase change is ωL we find

$$|\omega_b \Delta L| < \frac{\pi}{3},$$

which gives the following estimate of permissible variations in the time delay

$$\frac{|\Delta L|}{L} < \frac{\pi}{3\omega_b L} \approx \frac{1}{\omega_b L}. \tag{8.14}$$

Controllers with large values of $\omega_b L$ thus require that the time delay be known accurately. Consider, for example, the system in Figure 8.3 with $L = 8$. In this case we have $\omega_b L = 16$, which implies that the permissible error in the time delay is at most 6 percent.

The Loop Transfer Function

Analysis of the sensitivity functions indicates that the robustness of a closed-loop system with a Smith predictor may be poor when $\omega_b L$ is large. An analysis of the loop transfer function gives additional insight.

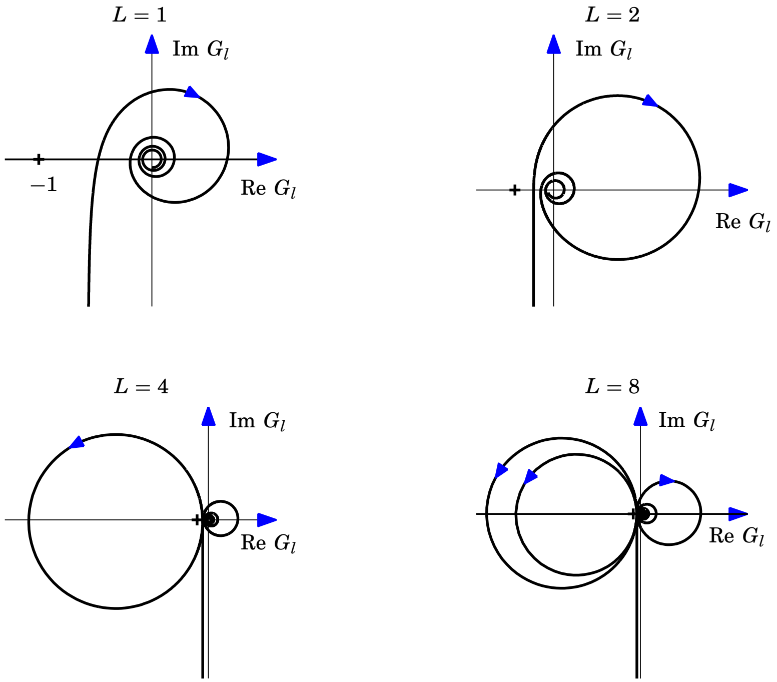

Figure 8.7 Nyquist plots of the loop transfer function for a FODT system, (8.5), with a Smith predictor controller. The critical point -1 is marked with a $+$.

When there are no modeling errors the loop transfer function obtained using a Smith predictor is

$$PC = \frac{PC_0}{1 + C_0(P_0 - P)} = \frac{P_0 C_0 e^{-sL}}{1 + P_0 C_0 (1 - e^{-sL})}. \tag{8.15}$$

Figure 8.7 shows the Nyquist plots of the loop transfer function for different values of L. For $L = 1$ the Nyquist plot has a loop of moderate size. The loop increases with increasing L, as is seen by comparing the cases $L = 1$ and $L = 2$ in Figure 8.7. The loop is almost circular for L larger than 2. For $L = 2.99$ the loop is infinitely large, and for $2.99 < L < 6.40$ the loop transfer function has two encirclements of the critical point, one for positive and another for negative ω. Notice that we have only shown the branch of the Nyquist plot corresponding to $0 \leq \omega < \infty$. The unstable poles are the poles of the predictor transfer function (8.7). The number of encirclements increases as L increases. For $L = 8$ there are four encirclements of the critical point.

Figure 8.8 shows the Bode plots of the loop transfer function for the cases $L = 1$ and $L = 8$. The loop transfer functions change drastically with L. The gain crossover frequency is 0.82 for $L = 1$ and decreases to about 0.13 for $L = 8$. These values agree quite well with the performance limit $\omega_{gc} L \approx 1$ given by (4.57). Notice that the gain curve for $L = 8$ has several crossings at higher frequencies. The gain crossover frequency is smaller for $L = 8$ even if the rise

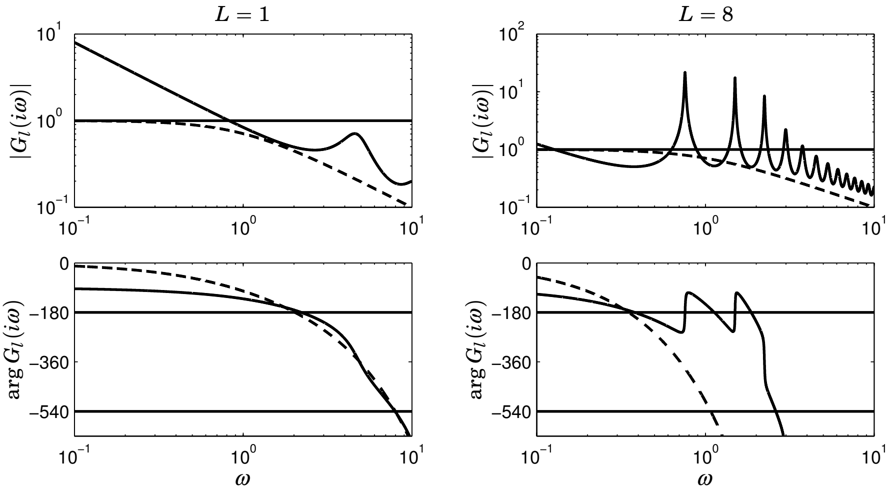

Figure 8.8 Bode plots of the loop transfer function (solid) and the process transfer function (dashed) for a FOTD system (8.5) with a Smith predictor. The curves on the left are for $L = 1$, and those on the right for $L = 8$.

time for set-point changes are the same for both systems. The high peaks of the gain curve correspond to the loops in the Nyquist plot in Figure 8.7.

The Bode plot of the open-loop system is shown in dashed lines in Figure 8.8. Notice that the controller gives a large phase advance at the frequencies corresponding to the first two peaks, which represent the unstable poles of the controller.

The Delay Margin

The classical robustness measures, gain margin and phase margin, do not capture the properties of Nyquist curves of the type shown in Figure 8.7, where the Nyquist curve has large loops. This is illustrated in Figure 8.9, which shows Nyquist plots of the loop transfer function for the case $L = 2$ and for a system where the time delay of the process has been increased with 30 percent. The figure shows that the system becomes unstable when the time delay is increased by 30 percent. Notice that it is the large loop that crosses the critical point -1 and not the part of the Nyquist curve close to the gain margin. The robustness measure called *the delay margin* is introduced to capture this effect. The delay margin is defined as the change in the time delay required to make a system unstable. For the systems with $L = 2$ and $L = 8$ in Figure 8.7 the delay margins are 27 percent and 7 percent, respectively.

Notice that the sensitivity functions also capture the robustness in the cases of loop transfer functions like the ones shown in Figure 8.7. The sensitivity to variations in the time delay can be estimated by (8.14), which gives delay margins of 25 percent and 6 percent for the systems in Figure 8.7 with $L = 2$ and $L = 8$. These numbers are close to the numbers obtained by using the delay margin.

Another way to quantify robustness is to explore the sensitivity of the closed

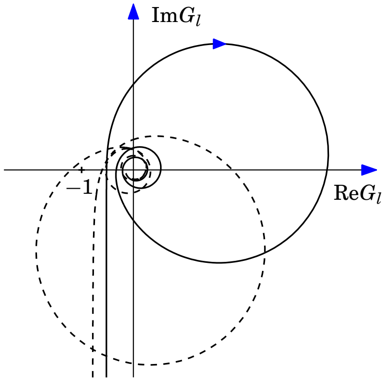

Figure 8.9 Nyquist plots of the loop transfer functions for the system in Example 8.1 with $L = 2$ in the nominal case (solid line) and when the time delay is increased by 30 percent (dashed line).

loop to variations in the process parameters. For the FOTD process we have

$$P(s) = \frac{K_p}{1 + sT} e^{-sL}$$

Hence,

$$\log P = \log K_p - \log(1 + sT) - sL$$

Differentiation this gives

$$\frac{dP}{P} = \frac{dK_p}{K_p} - \frac{sdT_p}{1 + sT_p} - sdL = \frac{dK_p}{K_p} - \frac{sdT_p}{1 + sT_p} - sL\frac{dL}{L}$$

For systems with large time delays the last term is dominating, which means that the sensitivity to time variations in the time delay is the critical constraint. Equation (4.32) then gives

$$\frac{|dL|}{L} < \frac{1}{\omega L |T(i\omega)|},$$

and we obtain the following estimate of the delay margin:

$$d_m = \max \frac{|dL|}{L} < \max \frac{1}{\omega L |T(i\omega)|}.$$

Summary

The Smith predictor makes it possible to obtain dramatic improvements of the set-point response as illustrated in Figure 8.3. The controller is obtained in a very simple way by first designing a controller C_0 for a nominal system P_0 that does not have the time delay. The Smith predictor is then obtained by cascading C_0 with a predictor C_{pred}, which effectively eliminates the time delay. An interesting feature of the Smith predictor is that it uses past control actions for prediction. It is in principle possible to compensate for any delay. The controller may, however, have unstable poles. The product $\omega_b L$, where ω_b is the bandwidth of the nominal closed-loop system $T_0 = P_0 C_0/(1+P_0 C_0)$ and L is the time delay L, are crucial parameters. The number of unstable controller poles grows with $\omega_b L$. Controllers with poles in the right half plane have poor robustness. Admissible variations in the time delay are inversely proportional to $\omega_b L$. To have a robust closed-loop system it is therefore necessary to restrict $\omega_b L$. In Example 8.3 we found, for example, that to have $M_s = 1.4$ it was necessary to have $\omega_b L < 0.5$.

8.4 The PPI Controller

In this section we will discuss special cases of the Smith predictor that give controllers of a particularly simple form. The Smith predictor discussed in Example 8.1 was based on the FOTD model. The design criterion was to find a controller that gives a second-order system with poles having relative damping ζ and frequency ω_0 for the system without delay. Another possible design is to choose a controller that cancels the process pole and makes the other closed-loop pole equal to $s = -1/T_{cl}$, where T_{cl} is the desired response time of the closed-loop system. This design method gives the following controller parameters;

$$K = \frac{T}{T_{cl}K_p}, \quad T_i = T.$$

The loop transfer function of the nominal system without delay is $P_0 C_0 = 1/(sT_{cl})$, and the controller has the transfer function

$$C(s) = \frac{1+sT}{K_p s T_{cl}} \frac{1}{1+\frac{1}{sT_{cl}}(1-e^{-sL})}. \tag{8.16}$$

The loop transfer function is

$$P(s)C(s) = \frac{1}{sT_{cl}} \frac{1}{1+\frac{1}{sT_{cl}}(1-e^{-sL})}. \tag{8.17}$$

Since the process pole is canceled it should be required that the process pole is fast in comparison with the dominant closed-loop dynamics; see Section 6.6. There is one tuning parameter: the closed-loop response time T_{cl}.

The input-output relation of the controller (8.16) can be written as

$$\begin{aligned} U(s) &= \frac{1+sT}{K_p s T_{cl}} E(s) - \frac{1}{sT_{cl}}(1-e^{-sL})U(s) \\ &= \frac{1+sT}{K_p s T_{cl}} \left(E(s) - \frac{K_p}{1+sT}(1-e^{-sL})U(s) \right) = \frac{1+sT}{K_p s T_{cl}} E_p(s), \end{aligned} \tag{8.18}$$

where $E_p(s)$ is the Laplace transform of the predicted error

$$e_p(t) = y_{sp}(t) - y(t) - \tilde{y}(t),$$

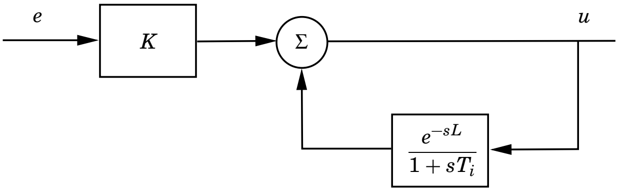

Figure 8.10 Block diagram of an implementation the PPI controller with $T_{cl} = T = T_i$.

and

$$\tilde{Y}(s) = \frac{K_p}{1+sT}\left(1 - e^{-sL}\right)U(s).$$

The term $\tilde{y}(t)$ represents the effect on the output of control actions taken in the interval $(t-L, t)$. The controller can thus be interpreted as a PI controller that acts on a predicted error, which is the actual error compensated for past control actions that have not yet appeared at the output. The controller is called the *predicting PI controller* or the *PPI controller*.

The controller is particularly simple if $T_{cl} = T$. The input-output relation of the controller then becomes

$$U(s) = KE(s) + \frac{e^{-sL}}{1+sT_i}\, U(s).$$

A block diagram describing this equation is given in Figure 8.10. Notice the strong similarity with the PI controller shown in Figure 3.3. There are also versions of this controller where the gain is replaced by a PD controller.

The Predictor

The PPI controller (8.16) is a cascade combination of a PI controller and a predictor with the transfer function

$$C_{pred}(s) = \frac{1}{1 + \frac{1}{sT_{cl}}\left(1 - e^{-sL}\right)}. \tag{8.19}$$

Apart from frequency scaling the predictor is completely characterized by the ratio T_{cl}/L. It can be shown that the predictor does not have poles in the right half plane for any values of T_{cl}. The reason for this is that the loop transfer function of the nominal system without delay has constant phase.

A series expansion of the transfer function (8.19) for small s gives

$$\begin{aligned} C_{pred}(s) &\approx \frac{1}{1 + L/T_{cl} - sT_{cl}(L/T_{cl})^2/2 + \ldots} \\ &\approx \frac{1}{1 + L/T_{cl}}\left(1 + \frac{1}{2}\frac{(L/T_{cl})^2}{1 + L/T_{cl}}T_{cl}s + \ldots\right). \end{aligned} \tag{8.20}$$

The static gain is $C_{pred}(0) = 1/(1 + L/T_{cl})$, and it also follows that C_{pred} goes to 1 as s goes to infinity. Figure 8.11 shows the Bode plot of the predictor (8.19).

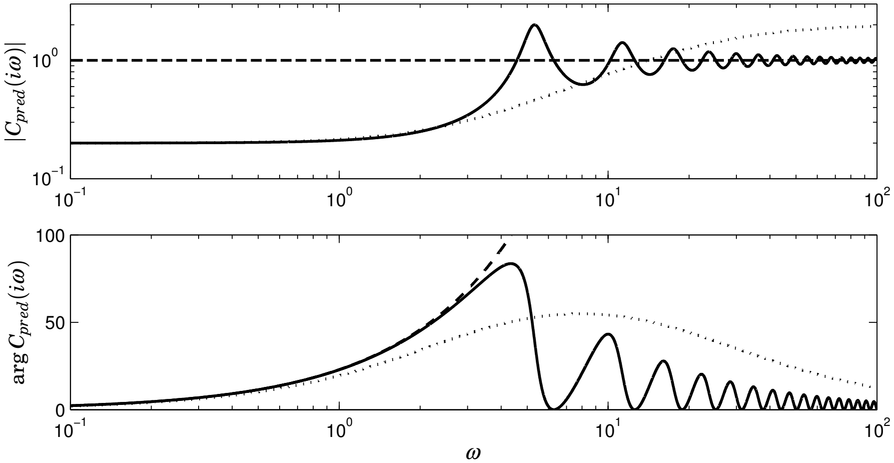

Figure 8.11 Bode plots for the predictor (8.19) (solid), a predictor based on differentiation (dotted), and the ideal predictor e^{sT} (dashed).

For comparison we have also given the Bode plots for an ideal predictor $e^{sT_{pred}}$, where

$$T_{pred} = \frac{1}{2}\frac{(L/T_{cl})^2}{1+L/T_{cl}}, \tag{8.21}$$

and a predictor based on differentiation. The predictor based on differentiation has been adjusted to give the same maximum gain as the predictor (8.19). There are differences between the predictors. The ideal predictor has unit gain for all frequencies; the other predictors have higher gains at high frequencies and lower gains at lower frequencies. The predictor (8.19) provides larger phase advance than the predictor based on differentiation, but the phase advance falls off rapidly for higher frequencies.

Design Choices

The choice of the design parameter T_{cl} is a compromise between robustness and performance. The response time is directly given by T_{cl}; fast response time requires a small T_{cl}. Robustness is governed by the ratio T_{cl}/L. The sensitivity function is given by

$$S = 1 - \frac{e^{-sL}}{1+sT_{cl}}.$$

Figure 8.12 shows the maximum sensitivity as a function of T_{cl}/L. Notice that the largest sensitivity has the property $M_s \leq 2$. To have $M_s \leq 1.6$ requires $T_{cl} > 0.66L$ and $M_s \leq 1.4$ requires $T_{cl} > 1.4L$. To have a reasonable robustness the desired response time cannot be chosen much shorter than L. It follows from (8.14) that the largest relative error in the time delay is given by

$$\frac{|\Delta L|}{L} \leq \frac{T_{cl}}{L}.$$

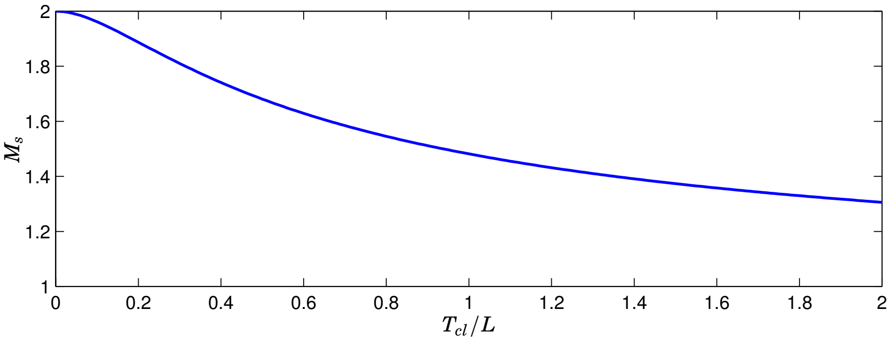

Figure 8.12 Maximum sensitivity M_s for the closed-loop system with the PPI controller (8.16) as a function of T_{cl}/L.

If maximum sensitivities as high as $M_s = 2$ are allowed and if the time delay is known precisely it is possible to allow smaller ratios T_{cl}/L.

For well-damped systems the integrated error IE is a performance measure that is easy to compute. From (8.18), the PPI controller in time domain is

$$u(t) = \frac{T}{K_p T_{cl}} e(t) + \frac{1}{K_p T_{cl}} \int_0^t e(t) dt - \frac{1}{T_{cl}} \int_0^t \big(u(t) - u(t-L)\big) dt. \tag{8.22}$$

To compute the integral error for the PPI controller it will be assumed that the system is initially at rest and that a load disturbance in the form of a unit step is applied to the process input. Since the controller has integral action, we have $u(\infty) = 1$. Therefore,

$$\int_0^\infty \big(u(t) - u(t-L)\big) dt = L.$$

After a unit load disturbance, it follows from (8.22) that

$$u(\infty) - u(0) = 1 = \frac{1}{K_p T_{cl}} \int_0^\infty e(t) dt - \frac{L}{T_{cl}}.$$

The integral error thus becomes

$$IE_{\text{PPI}} = K_p(L + T_{cl}).$$

The integrated error consists of two terms. The first term, K_pL, is due to the time delay and cannot be influenced by the controller. The second term, K_pT_{cl}, may be made small by specifying a short closed-loop time constant T_{cl}. A small value of T_{cl} will, however, result in poor robustness.

It is interesting to compare the performance of the PPI controller with the performance of PID controller. In Section 4.9, it was shown that the integral error for a PID controller is

$$IE_{\text{PID}} = \frac{T_i}{K} = \frac{1}{k_i}.$$

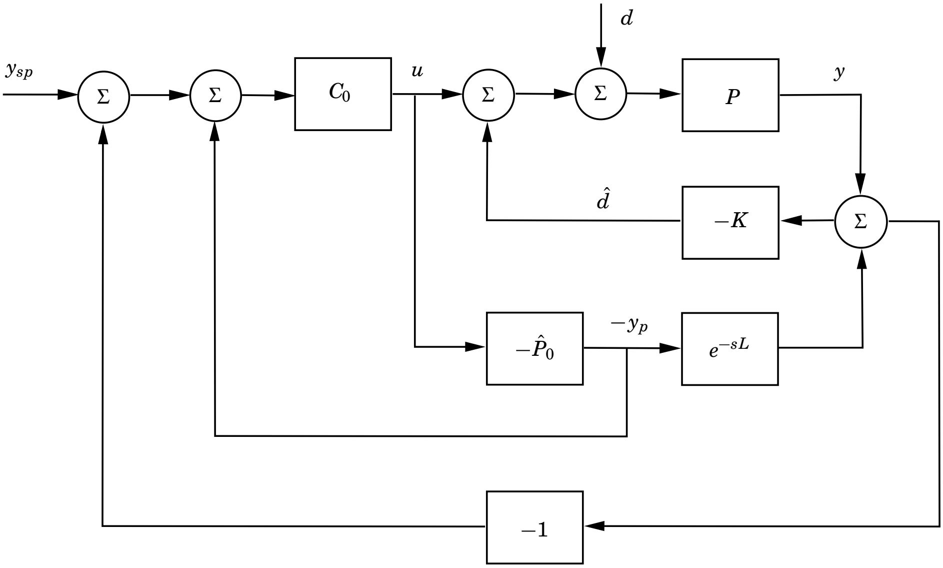

Figure 8.13 Modified Smith predictor for integrating processes.

It follows from (7.7) that a PID controller for delay-dominated processes tuned for $M_s = 1.4$ has $k_i K_p L = 0.5$. This gives $IE = 2K_p L$, which is close to the value $IE = 2.4K_p L$ obtained for the PPI controller. We thus obtain the conclusion that the PPI controller does not give significantly better performance at load disturbances than a PI controller if both controllers have the same robustness. The main advantage of the PPI controller is its ability to improve set-point responses; see Figure 8.3.

8.5 Predictors for Integrating Processes

The basic Smith predictor has useful properties, but it also has some severe drawbacks. It cannot be used for unstable systems, and it gives a steady-state error for load disturbances for processes with integration. Several modifications have therefore been proposed.

For processes with integration it has been suggested to modify the Smith predictor, as shown in Figure 8.13, in order to obtain zero steady-state error for a constant load disturbance. The reason for the modification can be understood from the principle of internal model control. The signal $\hat{d}$ that is fed back is an estimate of the load disturbance.

From Figure 8.13 the transfer functions from set point y_{sp} and load distur-

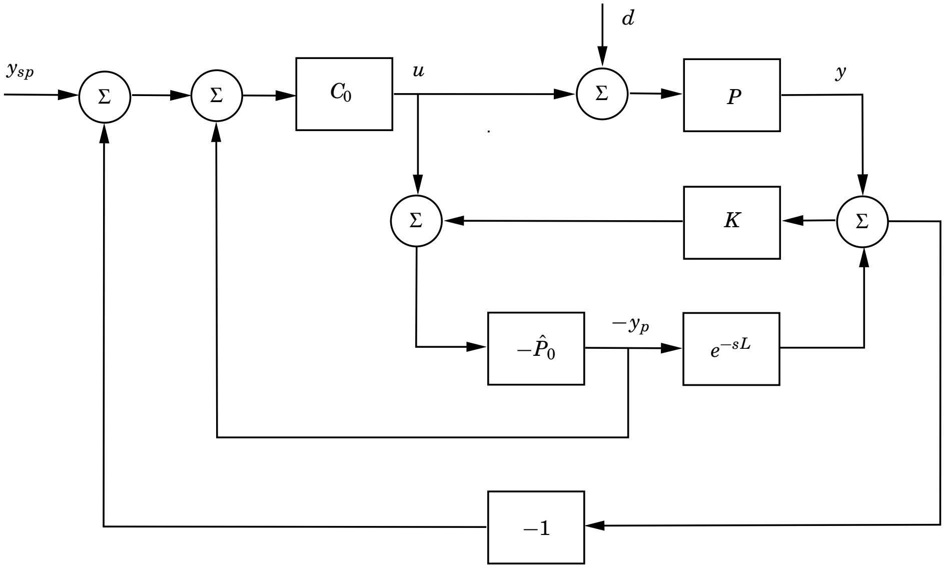

Figure 8.14 Modified Smith predictor for integrating processes.

bance d to output y are given by

$$\begin{aligned} Y = &\frac{PC_0(1+K\hat{P})}{1+C_0(\hat{P}_0-\hat{P})+P(K+KC_0\hat{P}_0+C_0)}Y_{sp} \\ &+\frac{P(1+C_0(\hat{P}_0-\hat{P}))}{1+C_0(\hat{P}_0-\hat{P})+P(K+KC_0\hat{P}_0+C_0)}D. \end{aligned} \tag{8.23}$$

When $s \to 0$, the following approximations hold:

$$C_0 \approx \frac{k_i}{s} \qquad \hat{P}_0 - \hat{P} \approx \frac{K_v}{s}(1-e^{-sL}) \approx K_v L.$$

If we also assume that $\hat{P} = P$, it can be shown that the transfer function between y_{sp} and y becomes one, and the transfer function between d and y becomes zero when $s \to 0$.

Another modification for integrating processes is given in Figure 8.14. The variable y_p is an estimate of the undelayed measurement signal

$$Y = P_0(U+D).$$

The estimation is given by

$$Y_p = \hat{P}_0(U+K(Y-\hat{Y})).$$

When $\hat{P}_0$ is stable, the value $K=0$ can be used, corresponding to the original Smith predictor. For integrating processes, it is, however, necessary to have $K \neq 0$.

From Figure 8.14 the transfer functions from set point y_{sp} and load disturbance d to output y are given by

$$\begin{aligned} Y =& \frac{PC_0(1+KP)}{1+K\hat{P}+\hat{P}_0-\hat{P}+PC_0(1+K\hat{P}_0)}Y_{sp} \\ &+ \frac{P(1+K\hat{P}+\hat{P}_0-\hat{P})}{1+K\hat{P}+\hat{P}_0-\hat{P}+PC_0(1+K\hat{P}_0)}D. \end{aligned} \tag{8.24}$$

Under the assumption that $\hat{P}=P$, it can be shown that the transfer function between y_{sp} and y becomes one, and the transfer function between d and y becomes zero when $s \to 0$.

8.6 Model Predictive Control

Model predictive control is based on the prediction of future process behavior based on a process model and optimization of the process behavior over a finite time horizon. Feedback is obtained by applying the initial part of the control signal and repeating the process over a shifted time horizon. This procedure is called *receding horizon control* or *moving horizon control.* Referring to Figure 8.15 the algorithm can be described as follows:

1: Develop a process model.

2: Consider the situation at time t. Past process inputs u and past process outputs y are observed; see Figure 8.15. The future behavior of the process is predicted under the assumption that the process model and the future control signals $u_f = u(\tau)$, $t \le \tau < t+t_h$ are known.

3: The control signal u_f is determined to give the desired future behavior.

4: The initial part of control signal u_f is applied over the interval $[t, t+h]$.

5: Change time to $t+h$, and repeat the procedure from Step 2.

The steps can be performed in many different ways, and there are a large number of algorithms. Different process models can be used; physical models, input-output models, and state models. The method can be applied both to single-input single-output systems and to systems with many inputs and many outputs.

The desired behavior can be specified in many ways. A common procedure is to specify the desired future behavior by a mathematical model, for example, one that tells how to approach the set point. The deviation from the desired behavior can be formulated as an optimization problem to minimize the deviation between actual and desired behavior, possibly with a penalty on control actions. Step 2 is an open-loop optimization problem where optimization is carried out over a finite time horizon. Feedback is obtained by only applying the initial part of the control signal. The horizon is then shifted forward, and the optimization is then repeated.

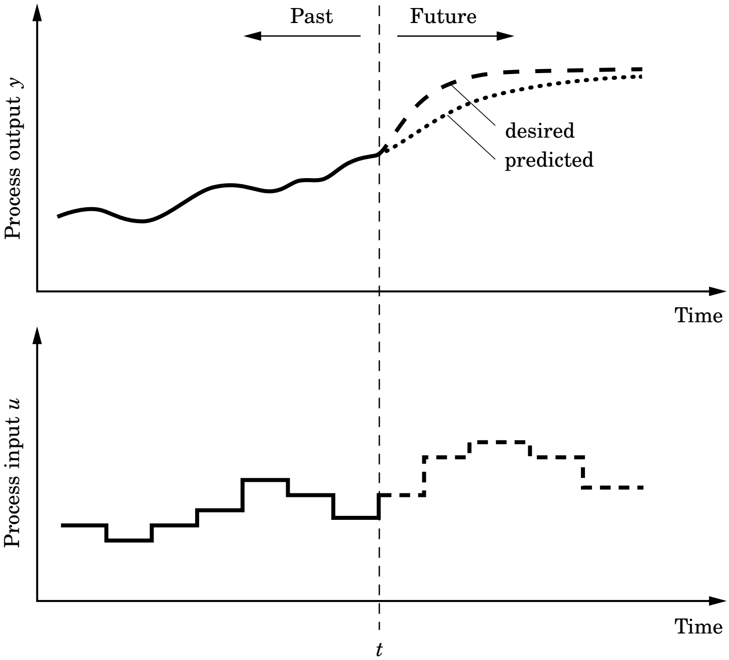

Figure 8.15 Illustration of the model predictive control.

Model predictive control is particularly simple for sampled systems where the control signal is constant over the sampling intervals. Parameter h can then be chosen as the sampling interval, and the prediction horizon t_h is typically chosen as a small number of sampling intervals. Most predictive controllers are also developed for sampled systems.

A very useful property of model predictive control is that constraints on the control signal and the process output can be taken into account. A common choice is to formulate the problem so that efficient algorithms for quadratic programming can be used. A key difficulty with model predictive control is to ensure stability when the prediction horizon is finite. Much research has been devoted to this problem.

A Simple Example

To illustrate the ideas we will give details in a simple case. Consider the sampled process model

$$y(t)+a_1y(t-h)+\cdots+a_ny(t-nh) = b_1u(t-h)+b_2u(t-2h)+\cdots+b_nu(t-nh), \tag{8.25}$$

where y is the process output and u the process input. Consider the situation at time t. The past behavior is completely characterized by

$$\mathcal{Y}_t = (y(t), y(t-h), \cdots, u(t-h), u(t-2h), \cdots). \tag{8.26}$$

Using the model it is straightforward to predict future values of process output as a function of current and future control signals:

$$\mathcal{U}_t = (u(t), u(t+h), \ldots, u(t+Nh)). \tag{8.27}$$

The desired future behavior can be characterized by specifying a reference trajectory for future process outputs, as indicated in Figure 8.15 and giving a loss function that penalizes deviations $e(t) = y(t) - y_d(t)$ from the desired output $y_d(t)$ and the increments of the control actions $\Delta u(t) = u(t) - u(t-h)$

$$J(u(t), u(t+h), \ldots, u(t+Nh)) = \sum_{k=1}^{t+N} e(t+kh)^2 + \rho(\Delta u(t+(k-1)h))^2. \tag{8.28}$$

There may also be constraints on process inputs and outputs and on the increment of the control signal.

Future control signals $\mathcal{U}_t$ are then computed by minimizing J subject to the constraints. The control signal $u(t)$ is then applied, and the whole procedure is repeated. The control signal is a function of past inputs and past outputs

$$u(t) = F(y(t), y(t-h), \ldots, y(t-nh), u(t-h), y(t-2h), \ldots, u(t-nh)),$$

where the function F is obtained implicitly by solving an optimization problem.

A particularly simple case is when the process model is of first order in the increments of process inputs and outputs, which we illustrate by an example.

Example 8.4—MPC for First-Order System

Let the process model be

$$\Delta y(t+h) = -a\Delta y(t) + b\Delta u(t),$$

where $\Delta y(t) = y(t) - y(t-h)$ and $\Delta u(t) = u(t) - u(t-h)$. Let the desired trajectory be a signal $y_d(t)$ which starts at $y(t)$ and approaches the set point y_{sp} exponentially with time constant T_{cl}. The desired process output at time $t+h$ is then

$$y_d(t+h) = y(t) + \left(1 - e^{-h/T_{cl}}\right)(y_{sp} - y(t)).$$

Assuming that there are no penalties on the control actions the desired process output can then be achieved in the next sampling period. Equating $y(t+h)$ with $y_d(t+h)$ gives

$$y(t+h) = y(t) + \Delta y(t+h) = y(t) - a\Delta y(t) + b\Delta u(t)^2 = y(t) + \left(1 - e^{-h/T_{cl}}\right)(y_{sp} - y(t)).$$

Solving this equation for $\Delta u(t)$ gives

$$\Delta u(t) = \frac{1 - e^{-h/T_{cl}}}{b}(y_{sp} - y(t)) + \frac{a}{b}\Delta y(t),$$

which is a PI controller with gains

$$k = \frac{a}{b}$$

$$k_i = \frac{1 - e^{-h/T_{cl}}}{b}.$$

Notice that the proportional gain only depends on the process model and that the integral gains depend on the desired response rate T_{cl}. □

It is straightforward to deal with systems having many inputs and outputs. It is also possible to include constraints. There are many special cases and variants of model predictive control. A few of them will be discussed briefly; for more details we refer to the references.

The Dahlin-Higham Algorithm

One of the earliest model predictive controllers was developed for control of paper machines. The algorithm is based on a process model in terms of the FOTD model

$$P(s) = \frac{K_p}{1+sT} e^{-sL},$$

and the desired response to set-point changes is given by

$$G_{yy_{sp}} = \frac{1}{1+sT_{cl}} e^{-sL}.$$

Assuming that the control signal is constant over sampling intervals of length $h = L/n$, where n is an integer, gives the sampled process model

$$y(t+h) = ay(t) + K_p(1-a)u(t-nh).$$

The desired response to set points is given by the difference equation

$$y_d(t+h) = a_d y_d(t) + (1-a_d)y_{sp}(t-nh).$$

Introducing the backward shift operator q^{-1} defined by

$$q^{-1}y(t) = y(t-h), \tag{8.29}$$

the process model can be written as

$$y(t) = \frac{K_p q^{-(n+1)}}{1-aq^{-1}} u(t) = P(q^{-1})u(t).$$

Let the controller be characterized by

$$u(t) = C(q^{-1})(y_{sp}(t) - y(t)).$$

The input-output relation for the closed-loop system is then

$$y(t) = \frac{P(q^{-1})C(q^{-1})}{1+P(q^{-1})C(q^{-1})} y_{sp}(t).$$

Using the backward shift operator the desired response is given by

$$y_d(t) = \frac{(1-a_d)q^{-(n+1)}}{1-a_d q^{-1}} y_{sp}(t) = G_d(q^{-1})y_{sp}(t),$$

where $a_d = e^{-h/T_{cl}}$. Equating this with the process output gives

$$\frac{P(q^{-1})C(q^{-1})}{1 + P(q^{-1})C(q^{-1})} = G_d(q^{-1}) = \frac{(1-a_d)q^{-(n+1)}}{1 - a_d q^{-1}}.$$

Solving this equation with respect to $C(q^{-1})$ gives

$$C(q^{-1}) = \frac{G_d(q^{-1})}{P(q^{-1})(1 - G_d(q^{-1}))} = \frac{(1-a_d)(1-aq^{-1})}{K_p(1 - a_d q^{-1} - (1-a_d)q^{-(n+1)})}.$$

The controller can then be described by

$$u(t) = \frac{1-a_d}{K_p}(e(t) - ae(t-h)) + a_d u(t-h) + (1-a_d)u(t-(n+1)h).$$

This controller has integral action, and past inputs are used for prediction.

Dynamic Matrix Control (DMC)

In dynamic matrix control the process is modeled by the finite impulse response model

$$y(t) = b_1 u(t-h) + b_2 u(t-2h) + \cdots + b_n u(t-nh), \tag{8.30}$$

and the criterion is to minimize the loss function

$$J\big(u(t), u(t+h), \ldots, u(t+(n-1)h)\big) = \sum_{k=1}^{n} e^2(t+kh),$$

where

$$e(t+kh) = y_d(t+kh) - b_1 u(t+kh-h) + b_2 u(t+kh-2h) + \cdots + b_n u(t+kh-nh).$$

Since e is a linear function of future control variables and the loss function is quadratic the optimization is straightforward. Notice that the model (8.30) also holds if there are many inputs and outputs. The coefficients b_i are then matrices. They were called *dynamic matrices* since they reflect the dynamics of the response in the original paper, which motivated the name DMC. In standard control terminology the parameters are simply the coefficients of the impulse response. In the early use of dynamic matrix control it was common practice to determine the matrices b_i from a simple impulse or step response measurement.

A drawback with DMC is that a large number of parameters may be required if the process dynamics are slow. The DMC algorithm was later generalized to QDMC (Quadratic Dynamic Matrix Control), which also can handle constraints on the control signal.

Minimum Variance Control

The minimum variance controller is a predictive controller for systems with random disturbances where the criterion is to minimize the variance of the fluctuations in process output. The algorithm was originally developed for control of paper machines where the stochastic nature of the disturbances is as important as the process dynamics. We start by a simple example.

EXAMPLE 8.5—MINIMUM VARIANCE CONTROL
Consider a model

$$y(t+h) = -ay(t) + bu(t) + e(t+h) + ce(t),$$

where u is the control variable, y the process output, and e a sequence of independent random variables with zero mean value and standard deviation σ. The sampling period is h.

Consider the situation at time t. The process output $y(t)$ is known and the output at time $t+h$ can be given arbitrary values by choosing the control signal $u(t)$. The random signal $e(t+h)$ is independent of past inputs and outputs $\mathcal{Y}_t$ given by (8.26). Furthermore, $e(t)$ can be computed from past inputs and outputs $\mathcal{Y}_t$. The control law that minimizes the deviation from the set point y_{sp} is given by

$$u(t) = \frac{ay(t) - ce(t)}{b},$$

If this control law is used we find that $y(t) = e(t)$, which means that the output is white noise. The computation of $e(t)$ from past inputs is thus trivial, and the control law becomes

$$u(t) = \frac{a-c}{b}y(t).$$

□

In the general case, the process model is

$$a(q^{-1})y(t) = b(q^{-1})u(t) + c(q^{-1})e(t). \tag{8.31}$$

where u is the process input, y the process output, and e is a sequence of independent Gaussian random variables with zero mean and variance σ. $a(q^{-1})$, $b(q^{-1})$, and $c(q^{-1})$ are polynomials in the backward shift operator

$$\begin{aligned}
a(q^{-1}) &= 1 + a_1q^{-1} + a_2q^{-2} + \ldots + a_nq^{-n}\\
b(q^{-1}) &= \quad b_\ell q^{-\ell} + b_{\ell+1}q^{-\ell-1} + \ldots + b_nq^{-n}\\
c(q^{-1}) &= 1 + c_1q^{-1} + c_2q^{-2} + \ldots + c_nq^{-n}.
\end{aligned}$$

For simplicity we have chosen to let all polynomials be of the same degree. This is no lack of generality because we can allow trailing coefficients to be zero. The coefficient b_ℓ is the first non-vanishing coefficient in the polynomial $b(q^{-1})$. The number ℓ is an important parameter called the input-output delay, and we also introduce the polynomial $b'(q^{-1}) = q^\ell b(q^{-1})$.

It is natural to assume that there are no factors common to all three polynomials $a(q^{-1})$, $b(q^{-1})$, and $c(q^{-1})$. The polynomial $c(q^{-1})$ is assumed to have all its zeros outside the unit disc. The model (8.31) captures the dynamics both of the process and its disturbances.

Minimum variance control is closely related to prediction, and we will therefore first determine a predictor for the process output when the input u is zero. The prediction of y ℓ steps ahead is given by

$$c(q^{-1})\hat{y}(t+\ell) = g(q^{-1})y(t),$$

where the polynomial $g(q^{-1})$ is given by

$$a(q^{-1})f(q^{-1}) + q^{-l}g(q^{-1}) = c(q^{-1}).$$

Notice that the dynamics of the predictor are given by the polynomial $c(q^{-1})$ in the model (8.31). The prediction error

$$\varepsilon(t) = f(q^{-1})e(t)$$

has the variance

$$E\varepsilon^2 = \sigma^2 \sum_0^{\ell-1} f_k^2. \tag{8.32}$$

The simple minimum variance control strategy is given by

$$u(t) = -\frac{s(q^{-1})}{r(q^{-1})}y(t) = -\frac{g(q^{-1})}{b'(q^{-1})f(q^{-1})}y(t), \tag{8.33}$$

and the control error is

$$y(t) = f(q^{-1})e(t). \tag{8.34}$$

The error under minimum variance control is thus equal to the error in predicting the output ℓ steps ahead. The control error is a moving average of order $\ell - 1$. It is thus easy to determine if a process is under minimum variance control simply by computing the correlation function of the output. Since the control error is a moving average of order $\ell - 1$ its covariance function is zero for all lags greater than ℓ.

The robustness of minimum variance control is strongly influenced by the choice of sampling interval. It is good practice to choose h larger than $L/2$.

8.7 Summary

The performance of a PI controller can be improved by adding predictive capability. Derivative action is one possibility, but there are many other alternatives. The Smith predictor and model predictive control are useful for systems with time delays when good models are available. Drastic improvements in the response to set-point changes can be obtained when good models are available. The predictive PI controller is a simple version of the Smith predictor. It has the advantage over a PID controller that the achievable phase advance is larger. A paradox is that the predictive controller only gives modest improvements compared to PI controllers for processes with delay-dominated dynamics but the performance improvements can be significant for lag-dominated processes. Model predictive controllers are more general than Smith predictors, and they can also deal with systems having many inputs and many outputs. Constraints can also be taken into account.

Since predictive controllers are based on mathematical models it is important that the models are accurate. It is particularly important to have a good estimate of the time delay. A fairly complete robustness analysis was given for the Smith predictor. Similar results are available for other predictive controllers. The key result is that sensitivity to modeling errors is closely related to the parameter $\omega_b L$, where ω_b is the closed-loop bandwidth or L/T_{cl} where T_{cl} is the desired closed-loop response time when time delay L is neglected. Robustness required that both parameters are not too small. A reasonable rule of thumb is that the parameters should be larger than 0.5.

8.8 Notes and References

A controller for systems having time delay was proposed by [Smith, 1957]; it is also treated in the book [Smith, 1958]. An explanation of the mechanism that generates the large phase advance is given in [Åström, 1977]. Many modifications of the Smith predictor have been presented; see [Åström *et al.*, 1994; Matausek and Micic, 1996; Matausek and Micic, 1999; Kaya and Atherton, 1999; Kristiansson and Lennartson, 1999]. The controller in [Haalman, 1965], the PPI controller in [Hägglund, 1996], and the PIDτ controller in [Shinskey, 2002] are all special cases of the Smith predictor. The papers [Ross, 1977; Meyer *et al.*, 1976; Ingimundarson and Hägglund, 2002] compare Smith predictors with PID controllers.

Minimum variance control was developed in the early phase of computer control of paper machines as an attempt to find a control strategy that minimizes fluctuations in quality variables. A key result is that the smallest variance that can be achieved is the variance of the error in predicting the output over the time delay of the process. Minimum variance control was first published by [Åström, 1967] and a perspective on its use is given in [Åström, 2001]. Minimum variance control requires a model of disturbances and process dynamics. A method to obtain this information directly from process experiments was developed in [Åström and Bohlin, 1965] and applied to modeling and control of paper machines [Åström, 1970]. The self-tuning controller [Åström and Wittenmark, 1973] can be viewed as automation of system identification and minimum variance control.

The controller presented in [Dahlin, 1968] and [Higham, 1968] can be viewed as a discrete-time version of the Smith predictor. Both the Smith Predictor and the Dahlin-Higham controller, which are early versions of model predictive control [Shinskey, 1991b], were first developed for process control applications. There are many versions of model predictive control; see [Richalet *et al.*, 1976], [Cutler and Ramaker, 1980] and [Garcia and Morshedi, 1986]. There are several recent books on model predictive control [Allgower and Zheng, 2000; Kouvaritakis and Cannon, 2001; Maciejowski, 2002]. The survey papers [Rawlings, 2000; Qin and Badgwell, 2003] contain many references. The papers [Kulhavy *et al.*, 2001], [Downs, 2001] and [Young *et al.*, 2001] and the book [Blevins *et al.*, 2003] give an industrial perspective. Although model predictive control was originally intended for multi-variable systems it has also been suggested to use it as a replacement for PID control; see [Lu, 2004].