5

Feedforward Design

5.1 Introduction

Feedforward is a simple and powerful technique that complements feedback. Feedforward can be used both to improve the set-point responses and to reduce the effect of measurable disturbances. Use of feedforward to improve set-point response has already been discussed in connection with set-point weighting in Section 3.4. We will now give a systematic treatment of design of feedforward control and also discuss design of model-following systems. The special case of set-point weighting will be discussed in detail, and we will present methods for determining the set-point weights. We will also show how feedforward can be used to reduce the effect of disturbances that can be measured.

5.2 Improved Set-Point Response

Feedforward can be used very effectively to improve the set-point response of the system. By using feedforward it is also possible to separate the design problem into two parts. The feedback controller is first designed to give robustness and good disturbance rejection and the feedforward is then designed to give a good response to set-point changes.

Effective use of feedforward requires a system structure that has two degrees of freedom. An example of such a system is shown in Figure 3.10. It is first assumed that the system has the structure shown in Figure 5.1. Let the process have the transfer function $P(s)$. We assume that a feedback controller $C(s)$, which gives good rejection of disturbances and good robustness, has been designed, and we will consider the problem of designing a feedforward compensator that gives a good response to set-point changes.

The feedforward compensator is characterized by the transfer functions $M_u(s)$ and $M_y(s)$, where $M_y(s)$ gives the desired set-point response. The system works as follows. When the set point is changed the transfer function $M_u(s)$ generates the signal u_{ff}, which gives the desired output when applied as input to the process. The desired output y_m is generated by $M_y(s)$. Under ideal conditions this signal is equal to the process output y. The control error e

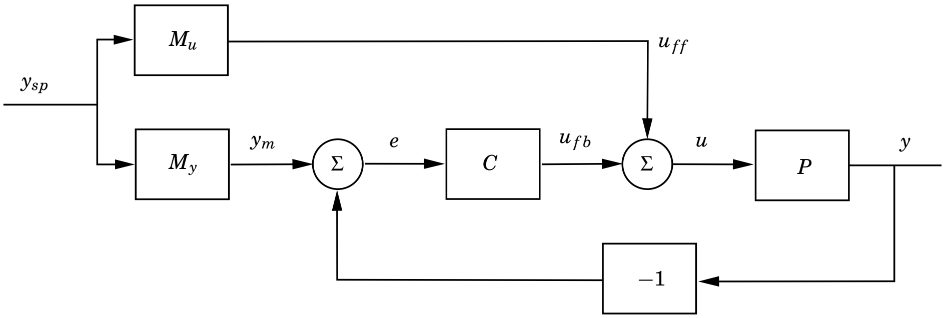

Figure 5.1 Block diagram of a system with two degrees of freedom.

is zero, and the feedback signal u_{fb} remains constant. If there are disturbances or modeling errors the signal y_m and y will differ. The feedback then attempts to bring the error to zero. The transfer function from set point to process output is

$$G_{yy_{sp}}(s) = \frac{P(CM_y + M_u)}{1+PC} = M_y + \frac{PM_u - M_y}{1+PC}. \tag{5.1}$$

The first term represents the desired transfer function. The second term can be made small in two ways. Feedforward compensation can be used to make $PM_u - M_y$ small, or feedback compensation can be used to make the error small by making the loop gain PC large. The condition for ideal feedforward is

$$M_y = PM_u. \tag{5.2}$$

Notice the different character of feedback and feedforward. With feedforward it is attempted to match two transfer functions, and with feedback it is attempted to make the error small by dividing it by a large number. With a controller having integral action the loop gain is very large for small frequencies. It is thus sufficient to make sure that the condition for ideal feedforward holds at higher frequencies. This is easier than to satisfy the condition (5.2) for all frequencies.

System Inverses

From (5.2) the feedforward compensator M_u is

$$M_u = P^{-1} M_y, \tag{5.3}$$

which means that it contains an inverse of the process model P. A key issue in design of feedforward compensators is thus to find inverse dynamics. It is easy to compute the inverse formally. There are, however, severe fundamental problems in system inversion, which are illustrated by the following examples.

EXAMPLE 5.1—INVERSE OF FOTD SYSTEM
The system

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

has the formal inverse

$$P^{-1}(s) = (1 + sT)e^{sL}.$$

This system is not a causal dynamical system because the term e^{sL} represents a prediction. The term $(1 + sT)$ requires an ideal derivative, which also is problematic as was discussed in Section 3.3. Implementation of feedforward thus requires approximations. □

EXAMPLE 5.2—INVERSE OF SYSTEM WITH RHP ZERO
The system

$$P(s) = \frac{s-1}{s+2}$$

has the inverse

$$P^{-1}(s) = \frac{s+2}{s-1}.$$

Notice that this inverse is an unstable system. □

It follows from (5.2) that there will be pole-zero cancellations when designing feedforward. The canceled poles and zeros must be stable and sufficiently fast; otherwise, there will be signals in the system that will grow exponentially or decay very slowly.

The difficulties in computing inverses can be avoided by restricting the choice of M_y. Since $M_u = P^{-1}M_y$ we can require that the transfer function M_y has a time delay that is at least as long as the time delay of P. Further, M_y and P must have the same zeros in the right half plane. To avoid differentiation, the pole excess in M_y must be at least as large as the pole excess in P. One possibility is to approximate process dynamics by a simple model and to choose M_y as a model having the same structure. To design feedforward we thus have to compute approximate system inverses with suitable properties.

Approximate Inverses

Different ways to find approximate process models were discussed in Section 2.8. Here we will give an additional method that is tailored for design of feedforward control.

Let $P^\dagger$ denote the approximate inverse of the transfer function P. A common approximation in process control is to neglect all dynamics and simply take the inverse of the static gain, i.e.;

$$P^\dagger(s) = P(0)^{-1}.$$

A number of results on more accurate system inverses have been derived in system theory. Some of these will be shown here. Note that the inverse transfer function only has to be small for those frequencies where the sensitivity function is large.

EXAMPLE 5.3—APPROXIMATE INVERSE OF FOTD SYSTEM
The system

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

has the approximate inverse

$$P^{\dagger}(s) = \frac{1+sT}{1+sT/N},$$

where N gives the frequency range where inversion is valid. □

EXAMPLE 5.4—APPROXIMATE INVERSE OF SYSTEM WITH RHP ZERO
The system

$$P(s) = \frac{s-1}{s+2}$$

has the inverse

$$P^{\dagger}(s) = \frac{s+2}{s+1}.$$

Notice that the unstable zero in P gives rise to a pole in $P^{\dagger}$ that is the mirror image of the unstable zero. □

A simple model for systems with monotone step responses has the transfer function

$$P(s) = \frac{K}{(1+sT)^n}e^{-sL}. \tag{5.4}$$

We call this the NOTD model because it has one time delay and n equal lags. The approximation can be made by fitting the transfer functions at a few relevant frequencies. Assuming that we want a perfect fit at $\omega = 0$ and $\omega = \omega_0$ we find that

$$\begin{aligned} P(0) &= K \\ |P(i\omega_0)| &= \frac{1}{(1+(\omega_0 T)^2)^{n/2}} \\ \arg P(i\omega_0) &= -n \arctan \omega_0 T - \omega_0 L. \end{aligned}$$

Solving these equations we find

$$\begin{aligned} K &= P(0) \\ T &= \frac{\sqrt{|P(i\omega_0|^{-n/2} - 1}}{\omega_0} \\ L &= -\frac{\arg P(i\omega_0) + n \arctan \omega_0 T}{\omega_0}. \end{aligned} \tag{5.5}$$

A good fit is required at the frequency ω_{ms} of maximum sensitivity. Since this frequency is known when the feedback controller C has been designed it is natural to choose $\omega_0 = \omega_{ms}$.

We will give an example to illustrate the accuracy of the approximation.

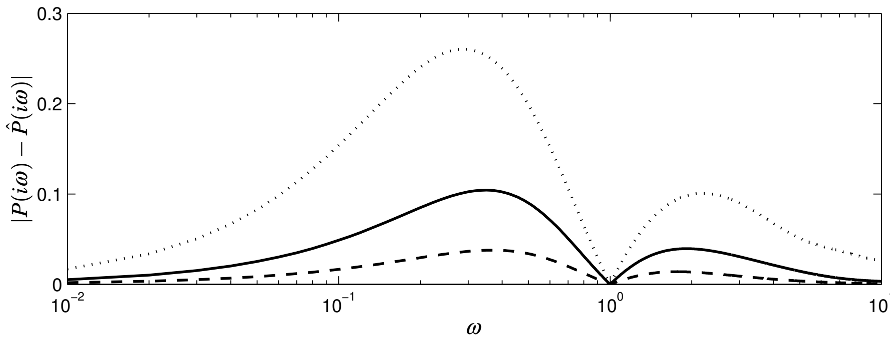

Figure 5.2 Error when fitting NOTD models of different orders to the transfer function $P(s) = 1/(s+1)^4$ for $n = 1$ (dotted), $n = 2$ (solid), and $n = 3$ (dashed).

Table 5.1 Parameters and maximum errors when fitting NOTD models of different orders to the transfer function $P(s) = 1/(s+1)^4$.

n	ω	K	L	T	$e_{\max}$	$\omega_{\max}$
1	0.5	1	1.9566	2.4012	0.1828	1.7400
2	0.5	1	1.1352	1.5000	0.0710	1.4500
3	0.5	1	0.5169	1.1773	0.0255	1.3300
1	1.0	1	1.8235	3.8730	0.2603	0.2800
2	1.0	1	1.0472	1.7321	0.1043	0.3400
3	1.0	1	0.4737	1.2328	0.0378	0.3600

EXAMPLE 5.5—FOUR EQUAL LAGS

Consider a process with the transfer function

$$P(s) = \frac{1}{(s+1)^4}.$$

In Figure 5.2 we show the error $|P(i\omega)-\hat{P}(i\omega)|$ for NOTD models with different n, and in Table 5.1 we give the parameters and the maximum error for different fits. Notice that relatively large errors, 20 to 30 percent, are obtained for first-order models, and significant reductions are obtained by increasing the model order. □

For a process given by (5.4) it is reasonable to choose the response model as

$$M_y = \frac{1}{(1+sT_m)^n} e^{-sL}.$$

It then follows from (5.2) that the feedforward compensator is given by

$$M_u = \frac{1}{K}\Big(\frac{1+sT}{1+sT_m}\Big)^n. \tag{5.6}$$

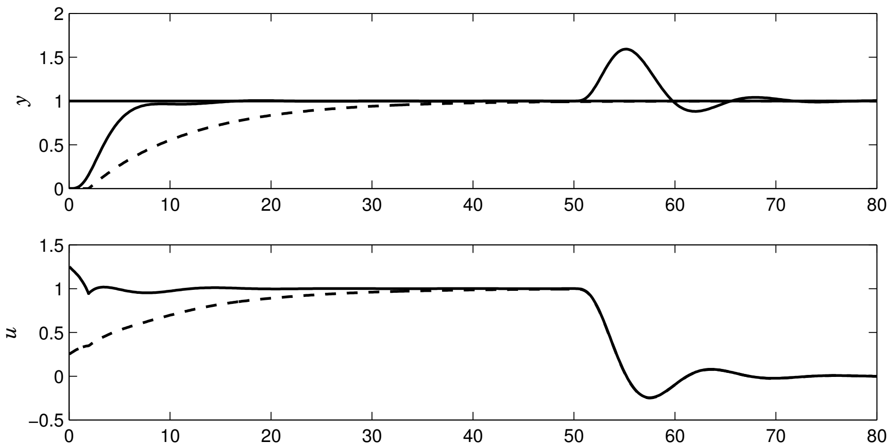

Figure 5.3 Responses to set points and load disturbances of the process $P(s) = 1/(s+1)^4$ with a PI controller and a feedforward based on the FOTD model for desired response times $T_r = 10$ (dashed line) and $T_r = 2$ (solid line).

In this particular case the feedforward compensator thus consists of a process model and a lead-lag or lag-lead network.

There are situations where it is desired that a feedback loop should have a set-point response with specified response time. A typical case is when several substances coming from different sources are mixed. When making production changes it is highly desirable that all systems react to production changes in the same manner. It is very easy to accomplish this when the required process dynamics are slow in comparison to the bandwidth of the feedback, because it follows from (5.1) that the set-point response is not very sensitive to the process model. We illustrate this with an example.

EXAMPLE 5.6—SLOW SET-POINT RESPONSE
Consider a process with the transfer function

$$P(s) = \frac{1}{(s+1)^4},$$

controlled with a PI controller with $K = 0.775$ and $T_i = 2.05$. This gives $M_s = 2$ and $\omega_{ms} = 0.559$. Approximating the process model with a first-order FOTD model gives the parameters $K_p = 1$, $T = 2.51$, and $L = 1.94$, see (5.5). Assume that the desired set-point response is given by

$$M_y(s) = \frac{1}{1 + sT_r}.$$

Figure 5.3 shows set-point responses for different values of T_r. The figure shows that the load disturbance response is the same in both cases and that the set-point response has the expected behavior. Notice the distortions of the curves for $T_r = 2$; they are due to the fact that the model does not fit so well for high frequencies. A rule of thumb is that the first-order model is reasonable for $\omega_{ms}T_r > 2$. In this case this gives $T_r > 3.6$. More accurate models are required to get the desired behavior for $T_r = 2$. □

The advantage by using a controller with two degrees of freedom is that the good disturbance attenuation can be maintained while making the set-point response slower.

5.3 Set-Point Weighting

For simple PID controllers it may not be necessary to use a complete system with two degrees of freedom. The desired set-point response can often be maintained simply by adjusting the set-point weights; see Section 3.4. To determine the set-point weights we consider the transfer function from set point to process output, and we choose set-point parameters so that the largest gain of this transfer function is one or close to one. This gives a set-point response without overshoot for most systems.

It follows from Figure 3.10 and Equation 3.20 that the transfer function from set point to process output is

$$G_{yy_{sp}}(s) = \frac{k_i + bks + ck_d s^2}{k_i + ks + k_d s^2} \frac{P(s)C(s)}{1 + P(s)C(s)} = \frac{k_i + bks + ck_d s^2}{k_i + ks + k_d s^2} T(s). \tag{5.7}$$

One possibility to achieve the largest gain close to one is to specify that the maximum sensitivity M_t is close to one. In such a case it may not be necessary to use set-point weighting. For designs with larger values of M_t we can simply compute maximum of $|G_{yy_{sp}}(i\omega)|$ and adjust the values of b and c that give a value close to one. The weight c is often set to zero. In that case, there is only one parameter to choose. If $|G_{yy_{sp}}(i\omega)|$ is larger than one for $b = 0$, a low-pass filtering of the set point may be used to reduce the magnitude of $|G_{yy_{sp}}(i\omega)|$ further. The set-point filter $F_{sp}(s)$ can be determined in the following way. Let m_s be the maximum of the transfer function (5.7) with $b = c = 0$, and let ω_{sp} be the frequency where the maximum occurs. A first-order filter

$$F_{sp} = \frac{1}{1 + sT_{sp}},$$

has the magnitude $1/m_s$ at the frequency ω_{sp} if the time constant is

$$T_{sp} = \frac{1}{\omega_{sp}} \sqrt{m_s^2 - 1}.$$

Feeding the set point through a low-pass filter designed in this way will reduce the magnitude at the frequency ω_{sp} to one.

A drawback with set-point weighting and filtering is that the set-point response may be unnecessarily slow.

5.4 Neutral Feedforward

A very simple choice of feedforward control for systems with monotone step responses that satisfies (5.2) is given by

$$\begin{aligned} M_y &= \frac{P}{P(0)} = \frac{P}{K_p} \\ M_u &= \frac{1}{K_p}. \end{aligned} \tag{5.8}$$

This means that the desired set-point response is the normalized open-loop response of the system. Since $M_u = 1/K_p$ the control signal is proportional to the set point. At a step change in the set point the control signal thus changes stepwise to the constant value that gives the desired steady-state, and remains at that value. The design of a neutral feedforward is thus very simple.

A complicated process model can be replaced by an approximate model. For PID control it is natural to base design of feedforward on the NOTD model. One way to determine appropriate parameters is to match the model at the frequency ω_{ms} where the sensitivity function has its largest value. We illustrate the design procedure with an example.

EXAMPLE 5.7—FOUR EQUAL LAGS
Consider a process with the transfer function

$$P(s) = \frac{1}{(s+1)^4}.$$

A PI controller with a specification on $M_s = 2$ for this system gives the parameters $K = 0.775$, $T_i = 2.05$, and $\omega_{ms} = 0.56$ of an approximate model. Equation 5.5 gives the parameters $K = 1$, $L = 1.94$, and $T = 2.50$. Figure 5.4 shows the response of the system to step and load disturbances. Notice that there is a dip in the control signal around time $t = 2$. The reason is the mismatch between the process and the model used to design the feedforward. This is illustrated in Figure 5.5, which shows the initial responses of the process and the model. Notice that the process responds faster than the model initially. There is then an error, which is compensated for by the feedback.

The set-point response can be improved by using a better approximation of the process model. One possibility is to fit a second-order NOTD model. Such a model has the parameters $K = 1$, $T = 1.52$, and $L = 1.13$. Figure 5.6 shows the responses of the system to step and load disturbances. Compared with Figure 5.4 the control signal is closer to the ideal value $u = 1$ and the set-point response is a little better. Figure 5.7 shows the comparison of the model output y_m and the process output. A comparison with Figure 5.5 shows that the second-order model gives a better fit. A comparison of Figure 5.4 with Figure 5.6 also illustrates that feedforward requires good modeling. □

In temperature control it is often desirable to have a controller without overshoot to step responses. The next example illustrates how neutral feedforward can be used to accomplish this.

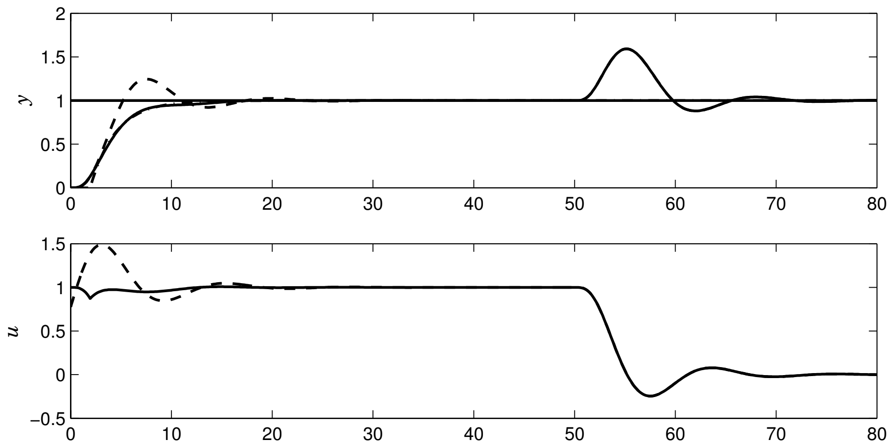

Figure 5.4 Responses to set point and load disturbances of the process $P(s) = 1/(s+1)^4$ with a PI controller (dashed line) and feedforward based on the FOTD model (solid line).

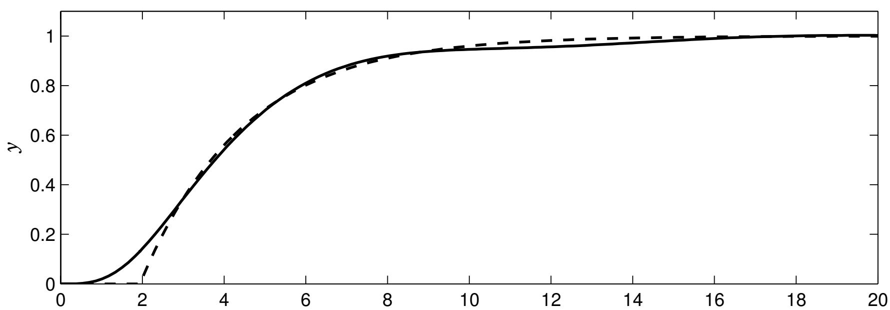

Figure 5.5 Step responses of the process P (solid line) and the model used to design the feedforward (dashed line).

EXAMPLE 5.8—DISTRIBUTED LAGS
Consider a process with the transfer function

$$P(s) = \frac{1}{\cosh\sqrt{s}}.$$

An aggressive PI controller with $M_s = 2$ has $K = 2.66$, $T_i = 0.197$, and $\omega_{ms} = 9.68$. Even with $b = 0$ this controller gives an overshoot as is shown by the dashed curve in Figure 5.8. Fitting a FOTD model at the frequencies 0 and ω_{ms} gives $K = 1$, $T = 0.408$, and $L = 0.0917$. The error in the transfer function is less than 5 percent. Figure 5.8 shows a simulation of the system with neutral feedforward based on that model. The figure shows that neutral feedforward achieves the desired response. □

Oscillatory System

PID control is not the best strategy for oscillatory systems because much better performance can be obtained with more complex controllers. PID control is,

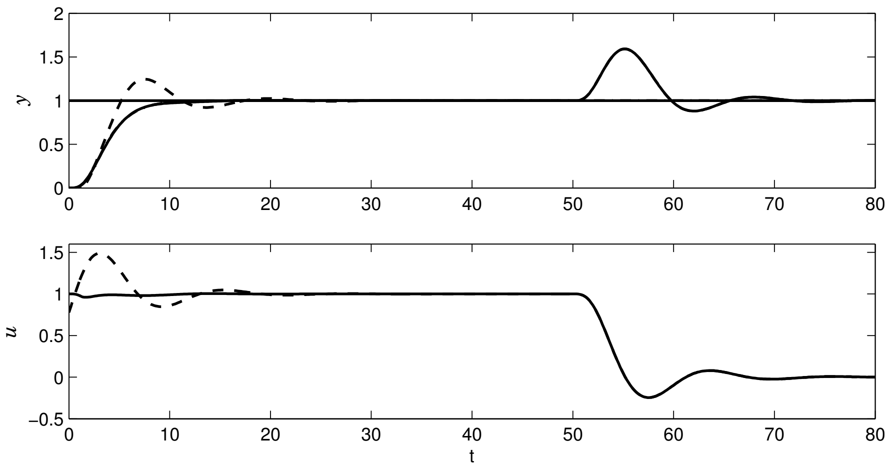

Figure 5.6 Responses to set points and load disturbances of the process $P(s) = 1/(s+1)^4$ with a PI controller (dashed line) and feedforward based on a SOTD model (solid line).

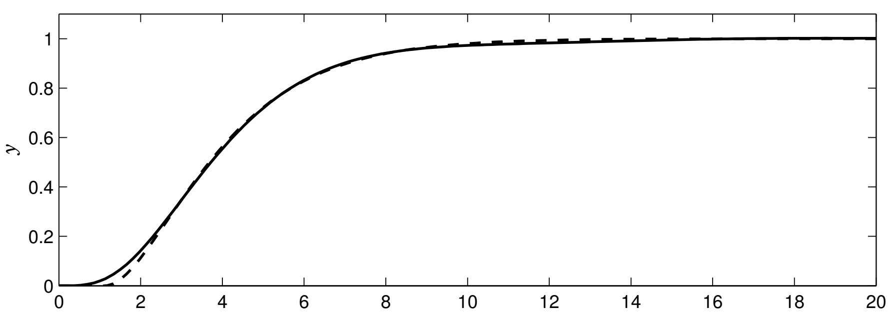

Figure 5.7 Step responses of the process P (solid line) and the model used to design the feedforward (dashed line).

however, sometimes used for such systems, and the performance of a conventional PID controller can often be improved by feedforward. Neutral feedforward, which gives a response similar to the uncontrolled system, can, however, not be used because it will give a response that is too oscillatory. We will illustrate how feedforward can be used by an example.

EXAMPLE 5.9—OSCILLATORY SYSTEM
Consider a system with the transfer function

$$P(s) = \frac{9}{(s+1)(s^2+0.1s+9)}.$$

The oscillatory mode has a relative damping $\zeta = 0.03$, which is quite low.

Reasonable PI controller parameters for the system are $K = -0.167$ and $T_i = -0.210$. Since the controller has negative gain, set-point weighting with $b = 0$ must be used to get a reasonable response. The overshoot is, however,

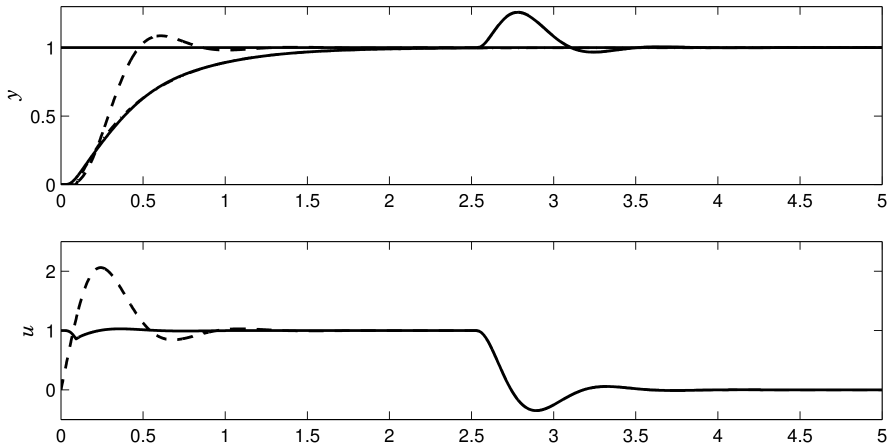

Figure 5.8 Responses to set points and load disturbances of the process $P(s) = 1/\cosh\sqrt{s}$ with a PI controller (dashed lines) and a neutral feedforward based on a first order FOTD model (solid lines).

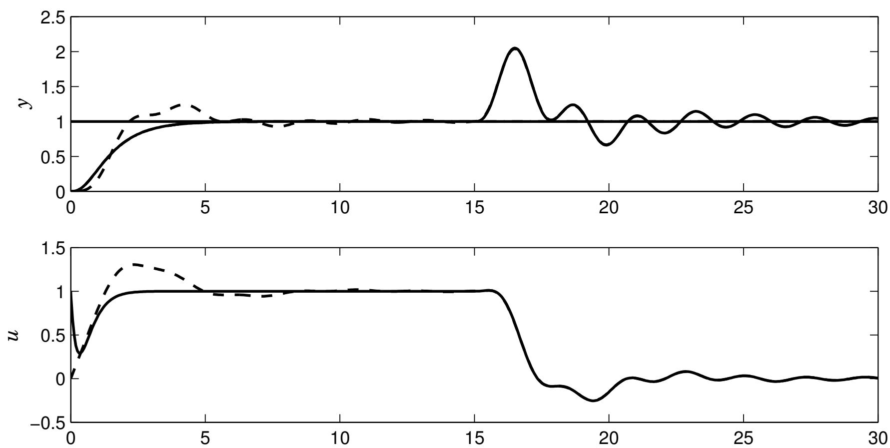

Figure 5.9 Responses to set points and load disturbances of the process $P(s) = 9/(s+1)(s^2+0.2s+9)$ with a PI controller (dashed lines) and a feedforward (solid lines).

still substantial as is seen by the dashed curve in Figure 5.9. To design the feedforward we choose a desired response given by the transfer function

$$M_y = \frac{9}{(s+1)(s^2+6s+9)}.$$

The dynamics of this system is essentially the same as for the process, but the complex poles now have critical damping. It follows from (5.2) that

$$M_u = \frac{s^2+0.1s+9}{s^2+6s+9}.$$

This transfer is close to one for all frequencies except those corresponding to the oscillatory modes where it has low gain. The transfer function thus blocks signals that can excite the oscillatory modes. Figure 5.9 shows the response of the system to set points and load disturbances. It is clear that the set-point response is improved substantially by the use of feedforward. The load disturbance response is still quite poor, which reflects the fact that PI control is not appropriate for a highly oscillatory system. □

5.5 Fast Set-Point Response

With neutral feedforward there is no overshoot in the control signal. It is possible to obtain more aggressive responses if we allow the control signal to overshoot. This is accomplished simply by requiring a faster response. To do this the model must also be accurate over a wider frequency range. The overshoot in the control signal may, however, increase very rapidly with increases in response time as is illustrated by the following example.

EXAMPLE 5.10—FAST SET-POINT RESPONSE
Consider the system

$$P(s) = \frac{1}{(s+1)^4}.$$

Assume that it is desired to have a set-point response given by

$$M_y = \frac{1}{(sT_m+1)^4}.$$

It follows from (5.3) that

$$M_u = \frac{(s+1)^4}{(sT_m+1)^4}.$$

For neutral feedforward we have $T_m = 1$, which gives $M_u = 1$. In general, we have

$$M_u(0) = T_m^{-4}.$$

The controller gain thus increases very rapidly with decreasing values of T_m. This is illustrated in the simulation shown in Figure 5.10, which shows the response for $T_m = 1$ (neutral feedforward), $T_m = 0.5$, and $T_m = 0.2$. The initial values of the control signal are 1, 16, and 625, respectively. Notice that the power 4 in the expressions is due to the fact that the process has a pole excess of 4. In practice, saturation of the actuator determines what can be achieved. □

Time Optimal Control

The example clearly illustrates that feedforward can be used to obtain fast set-point responses but that it requires models that are valid over a wide frequency range and that very large control signals may be required. The size of the

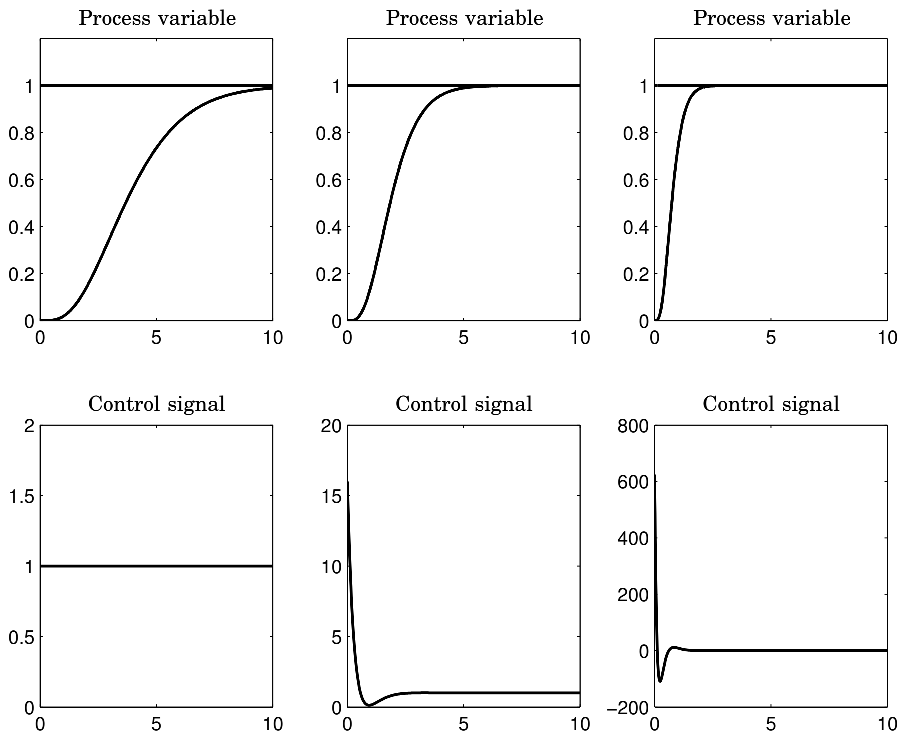

Figure 5.10 Set-point responses of the process $P(s) = (s+1)^{-4}$ with feedforward compensators designed to give $M_y(s) = (sT_m + 1)^{-4}$ for $T_m = 1$ (left), 0.5 and 0.2 (right).

control signal depends critically on the pole excess of the process. In practice, it is also necessary to take account of the fact that the control signals have limited range. It is therefore very natural to look for strategies that bring the process output from one set point to another in minimum time. This problem is solved by optimal control theory. It is known that for linear systems the solution is bang-bang control which means the control signal switches between its extreme values. An example is given in Figure 5.11, which shows the minimum time solution for the process $P = (s+1)^{-2}$ when the control signal is limited to values between 0 and 2. The control is very simple in this case. There can, however, be a large number of switches for high-order systems or for oscillatory systems. Because of its complexity it is not feasible to use optimal control except in very special situations. Approximate methods will therefore be developed.

Pulse Step Control

For stable systems with monotone step responses fast set-point responses can often be achieved with control signals that have the shape shown in Figure 5.11. This means that the maximum control signal is used initially. The control signal is then switched to its lowest value, and the control signal is finally given the value that gives the desired steady state. If the initial pulse is approximated with an impulse we obtain the situation shown in Figure 5.12.

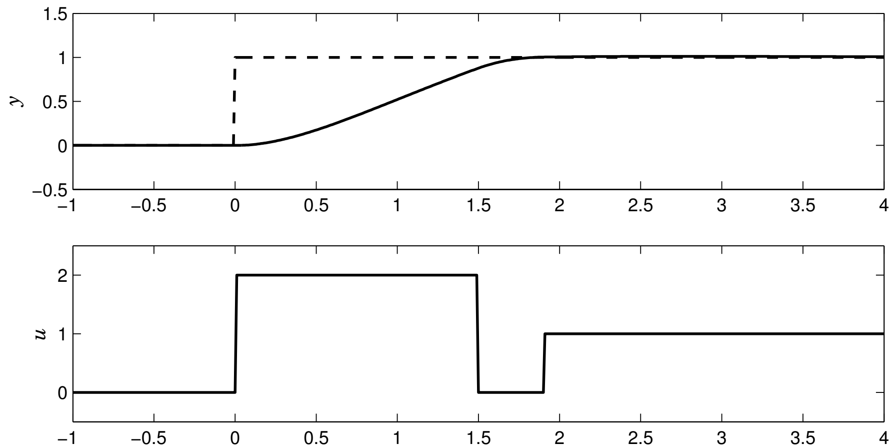

Figure 5.11 Time optimal set-point change for the process $P = (s+1)^{-2}$.

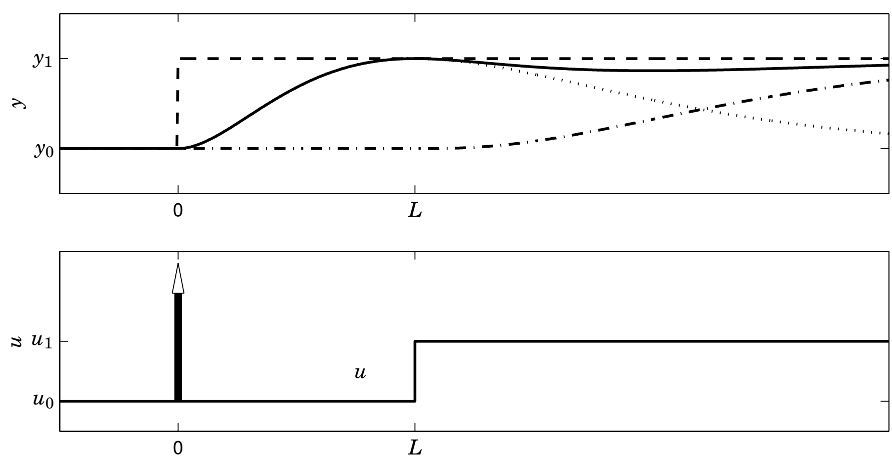

Figure 5.12 Response to an impulse (dotted line) and a delayed step (dashed-dotted line) for a system with monotone step response. The dashed line is the set point, and the solid line is the process output, composed of the sum of the impulse and step responses.

Assuming that the system is initially at rest its output is then given by

$$y(t) = ag(t) + bh(t - L),$$

where h is the step response and g the impulse response of the system. The parameters a, b, and L should be chosen so that the response matches the desired response as closely as possible. To do this the parameter a should be chosen as $y_{sp}/g_{\max}$, where $g_{\max}$ is the maximum of the impulse response $g(t)$. Parameter b should be chosen so that the desired steady state is obtained. Hence, $b = y_{sp}/K_p$ where K_p is the steady-state process gain. The parameter L should be adjusted to keep the output as close to the set point as possible. These choices imply that the settling time of the system is equal to the time

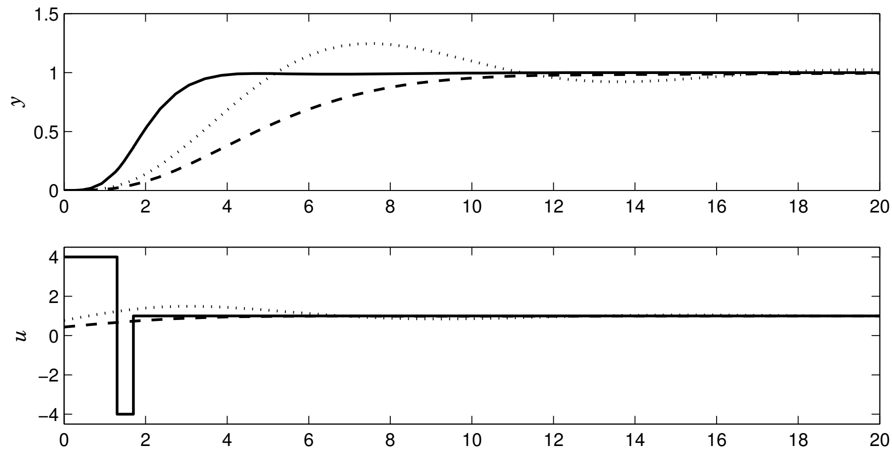

Figure 5.13 Comparison between the fast set-point response strategy (solid) and PI control with $M_s = 1.4$ (dashed) and $M_s = 2.0$ (dotted) for $P(s) = 1/(s+1)^4$.

where the impulse response has its maximum. The closed-loop settling time is thus matched to the natural response time of the system. It is of course not possible to have an impulse as an input. The impulse is therefore approximated by a pulse with an amplitude that corresponds to the maximum value of the control signal. The duration is chosen so that the area under the pulse equals a. The parameters given above can be fine-tuned by optimization. We illustrate the procedure by an example.

EXAMPLE 5.11—FAST SET-POINT RESPONSE
Consider a system with the transfer function

$$P(s) = \frac{1}{(s+1)^4}.$$

Figure 5.13 compares the fast set-point response method with regular PI control with two parameter settings. The fast set-point response has been computed with $u_{max} = 4$ and $u_{min} = -4$, and the resulting rise time and settling time are approximately 4 time units. The controllers have been designed with loop shaping for maximum sensitivities $M_s = 1.4$ and $M_s = 2.0$. The corresponding controller parameters are $K = 0.43$, $T_i = 2.25$, and $b = 1$ for $M_s = 1.4$, and $K = 0.78$, $T_i = 2.05$, and $b = 0.23$ for $M_s = 2.0$. Both PI designs are clearly outperformed by the pulse-step method. The rise times are a factor 2–3 longer, and the settling times approximately 3 times longer. The reason is, of course, that much less of the available control authority is used. If set-point weight b and/or M_s is increased, the size of the control signal will increase. This leads to a faster rise time, but at the expense of larger overshoot, so the settling time may actually be even higher. □

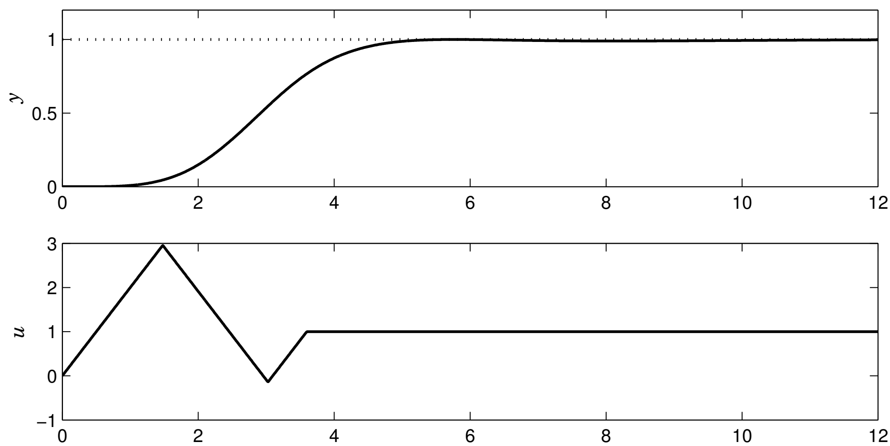

Figure 5.14 Process output and control signal for fast set-point changes with rate limitations $|du/dt| < 2$ for the process $P(s) = 1/(s+1)^4$.

Rate Limitations

The idea of fast set-point response can also be applied to the case when there are rate limitations. This is illustrated in Figure 5.14, which shows a simulation of the process with the transfer function $P = 1/(s+1)^4$ when there are rate limitations $|du/dt| < 2$. It is also possible to combine rate and level limitations.

5.6 Disturbance Attenuation

Disturbances can be eliminated by feedback. With a feedback system it is, however, necessary that there be an error before the controller can take actions to eliminate disturbances. In some situations, it is possible to measure disturbances before they have influenced the processes. It is then natural to try to eliminate the effects of the disturbances before they have created control errors. This control paradigm is called *feedforward*. The principle is illustrated in Figure 5.15.

In Figure 5.15 process transfer function P is composed of two factors, $P = P_1P_2$. A measured disturbance d enters at the input of process section P_2. The measured disturbance is fed to the process input via the feedforward transfer function G_{ff}. The transfer function from load disturbance to process output is

$$G_{yd}(s) = \frac{P_2(1 - P_1G_{ff})}{1 + PC} = P_2(1 - P_1G_{ff})S, \tag{5.9}$$

where $S = 1/(1+PC)$ is the sensitivity function. This equation shows that there are two ways of reducing the disturbance. We can try to make $1 - P_1G_{ff}$ small by a proper choice of the feedforward transfer function G_{ff}, or we can make the loop transfer function PC large by feedback. Feedforward and feedback can also be combined.

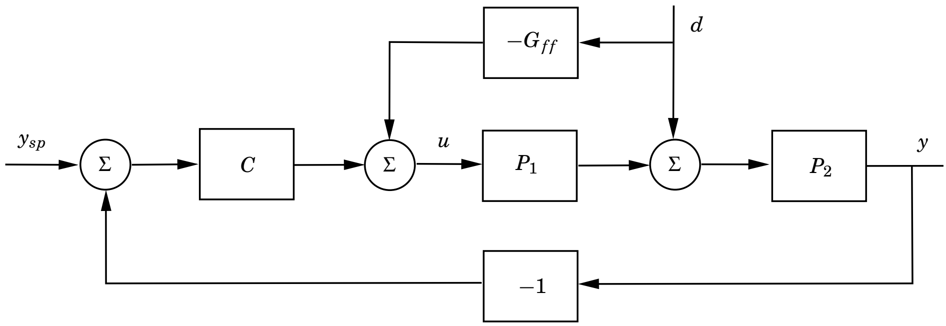

Figure 5.15 Block diagram of a system where a measured disturbance d is reduced by a combination of feedback and feedforward.

Notice that with feedforward we are trying to make the difference between two terms small, but with feedback we simply multiply with a small number. An immediate consequence is that feedforward is more sensitive than feedback. With feedback there is risk of instability; there is no such risk with feedforward. Feedback and feedforward are therefore complementary, and it is useful to combine them.

An ideal feedforward compensator is given by

$$G_{ff} = P_1^{-1} = \frac{P_{yd}}{P_{yu}}, \tag{5.10}$$

where P_{yd} is the transfer function from d to y and $P_{yu} = P$ is the transfer function from u to y. The ideal feedforward compensator is formed by taking the inverse of the process dynamics P_1. This inverse is often not realizable, but approximations have to be used.

Feedforward is most effective when the disturbance d enters early in the process. This occurs when most of the dynamics are in process section P_2. When $P_1 = 1$, and therefore $P_2 = P$, the ideal feedforward compensator is realizable, and the effects of the disturbance can be eliminated from the process output y. On the other hand, when the dynamics enter late in the process, so that $P_1 \approx P$, the effects of the disturbance are seen in the process output y at the same time as they are seen in the feedforward signal. In this case, there is no advantage of using feedforward compared to feedback.

Applications

In many process control applications there are several processes in series. In such cases, it is often easy to measure disturbances and use feedforward. Typical applications of feedforward control are drum-level control in steam boilers, control of distillation columns, and rolling mills. An application of combined feedback and feedforward control follows.

EXAMPLE 5.12—DRUM LEVEL CONTROL

A simplified diagram of a steam boiler is shown in Figure 5.16. The water in the raiser is heated by the burners. The steam generated in the raiser, which

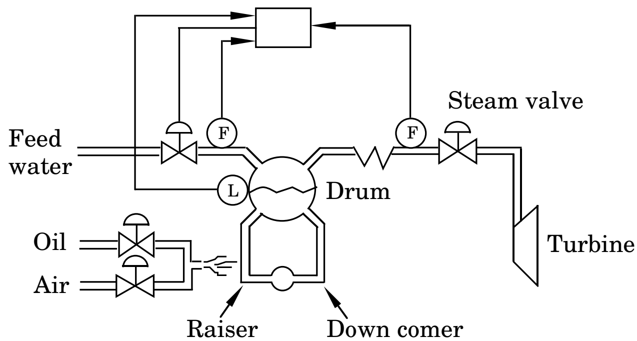

Figure 5.16 Schematic diagram of a drum boiler with level control.

is lighter than the water, rises toward the drum. This causes a circulation around the loop consisting of the raisers, the drum, and the down comers. The steam is separated from the water in the drum. The steam flow to the turbine is controlled by the steam valve.

It is important to keep the water level in the drum constant. Too low a water level gives insufficient cooling of the raisers, and there is a risk of burning. With too high a water level, water may move into the turbines, which may cause damage. There is a control system for keeping the level constant. The control problem is difficult because of the so-called *shrink and swell effect*. It can be explained as follows. Assume that the system is in equilibrium with a constant drum level. If the steam flow is increased by opening the turbine valve, the pressure in the drum will drop. The decreased pressure causes generation of extra bubbles in the drum and in the raisers. As a result, the drum level will initially increase. Since more steam is taken out of the drum, the drum level will of course finally decrease. This phenomenon, which is called the *shrink and swell effect*, causes severe difficulties in the control of the drum level. Mathematically, it also gives rise to right-half plane zero in the transfer function.

The problem can be solved by introducing the control strategy shown in Figure 5.16. It consists of a combination of feedback and feedforward. There is a feedback from the drum level to the controller, but there is also a feedforward from the difference between steam flow and feed-water flow so that the feed-water flow is quickly matched to the steam flow. □

5.7 Summary

Design of feedforward has been discussed in this chapter. Feedforward can be used to reduce the effect of measurable disturbances. Design of feedforward is essentially a matter of finding inverse process models. Different techniques to do this have been discussed. The major part of the chapter has been devoted to set-point response. A structure with two degrees of freedom has been used. This gives a clean separation of regulation and set-point response and of feedback and feedforward. It has been assumed that the feedback controller has been designed. A simple way to modify the set-point response is to use set-point weighting. If the desired results cannot be obtained by zero set-point weighting a full-fledged two-degree-of-freedom can be used. This makes is possible to make a complete separation between load disturbance response and set-point response. The crucial design issue is to decide the achievable response speed. For systems with monotone set-point responses the notion of neutral feedforward has been proposed. Many other variants have also been discussed. Finally, it has been demonstrated that very fast set-point responses can be obtained by using nonlinear methods.

Special care must be taken when implementing feedforward control, otherwise integrator windup may occur. Implementation of feedforward control is discussed in Section 13.4.

5.8 Notes and References

Feedforward is a useful complement to feedback. It was used in electronic amplifiers even before the feedback amplifier emerged as discussed in [Black, 1977]. Use of feedforward in process control was pioneered in [Shinskey, 1963]. The effectiveness of feedforward to improve set-point response using a system structure with two degrees of freedom (2DOF) was introduced in [Horowitz, 1963]. Set-point weighting, which is a simple form of 2DOF, has been used to a limited extent in early PID controllers where the weights have been 0 or 1. The use of continuously adjustable weights appeared in the 1980s. Use of feedforward to reduce the effect of measured disturbances is cumbersome to apply in the process control systems built on separate components but very easy in modern distributed control system; see [Bialkowski, 1995] and [ABB, 2002]. Applications of feedforward are gaining in popularity. Methods for assessment of potential improvements by using feedforward are also emerging; see [Petersson *et al.*, 2001; Petersson *et al.*, 2002; Petersson *et al.*, 2003].